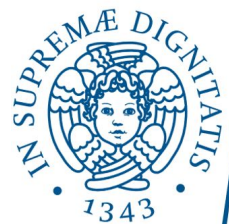




# An introduction to Ethereum data analysis

Matteo Loporchio  
23/4/2026  
matteo.loporchio@di.unipi.it





UNIVERSITÀ  
DI PISA



# Smart contracts and Tokens



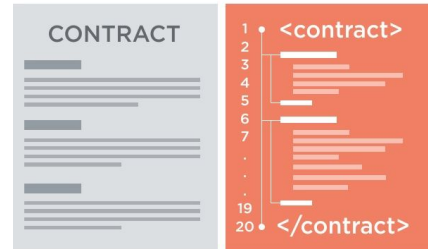
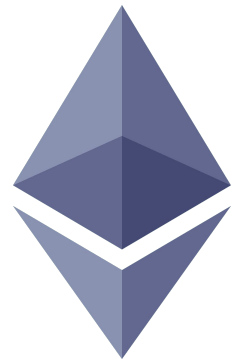
# Smart contracts

A blockchain can be seen as an ordered list of records (i.e., transactions).

Each record represents a *state update* and the list of records defines the *current state*.

A record represents *any* update, such as:

- 1) value transfer in a cryptocurrency;
- 2) code execution.



# Smart contracts

Programs that run on the Ethereum blockchain: execution is replicated by all (full) nodes in the network.

The blockchain stores both their *code* (i.e., functions) and *data* (i.e., state).

Transactions may transfer value or trigger code execution.

```
pragma solidity ^0.8.0;

contract SimpleStorage {
    uint256 public storedData;

    function set(uint256 x) public {
        storedData = x;
    }

    function get() public view returns (uint256) {
        return storedData;
    }
}
```

# Smart contracts

## Compilation



```
pragma solidity ^0.8.0;  
  
contract SimpleStorage {  
    uint256 public storedData;  
  
    function set(uint256 x) public {  
        storedData = x;  
    }  
  
    function get() public view returns (uint256) {  
        return storedData;  
    }  
}
```

Source code

```
608060405234801561001057600080fd  
5b506040516120d33803806120d38339  
81810160405260208110156100335760  
0080fd5b810190808051906020019092  
919050505060016000803373ffffffff  
ffffffffffffffffffffffffffffffff  
1673ffffffffffffffffffffffffffff  
ffffffffffff16815260200190815260  
20016000208190555060405180806120  
81605291396052019050604051809103  
90206040518060400160405280600e81  
526020017f44616920537461626c6563  
6f696e000000000000000000000000  
0000000008152508051906020012060  
4051806  
...
```

Bytecode

```
PUSH1 0x80  
PUSH1 0x40  
MSTORE  
PUSH1 0x0  
DUP1 PUSH1  
0x14 PUSH1  
0x0 CODECOPY  
PUSH1 0xAC  
PUSH1 0x23  
PUSH1 0x0  
CODECOPY  
PUSH1 0x0  
RETURN  
...
```

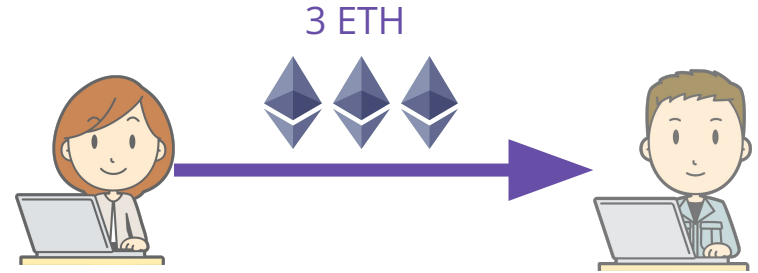
Opcodes

# Smart contracts

- As code execution requires computational resources, users must pay a fee proportional to the complexity of the operation.
- Each instruction consumes a predefined amount of *gas*.
- For each TX the user specifies a *gas limit* (i.e., the maximum amount of gas that can be consumed by it).
- The user also sets the gas price in Gwei (1 Gwei =  $10^9$  wei =  $10^{-9}$  ETH) to pay per unit of gas.

# Ethereum transactions

**Regular transactions:** a transaction transferring currency (*Ether*) from one account to another.



**Contract deployment:** a transaction without a 'to' address that writes the contract code on the blockchain.



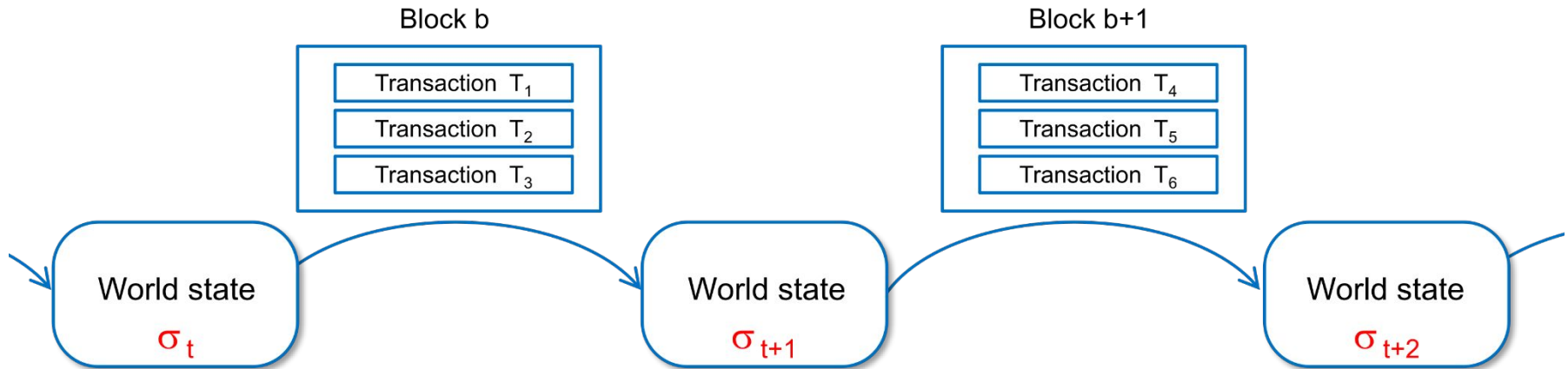
**Contract execution:** a transaction that interacts with a deployed smart contract.



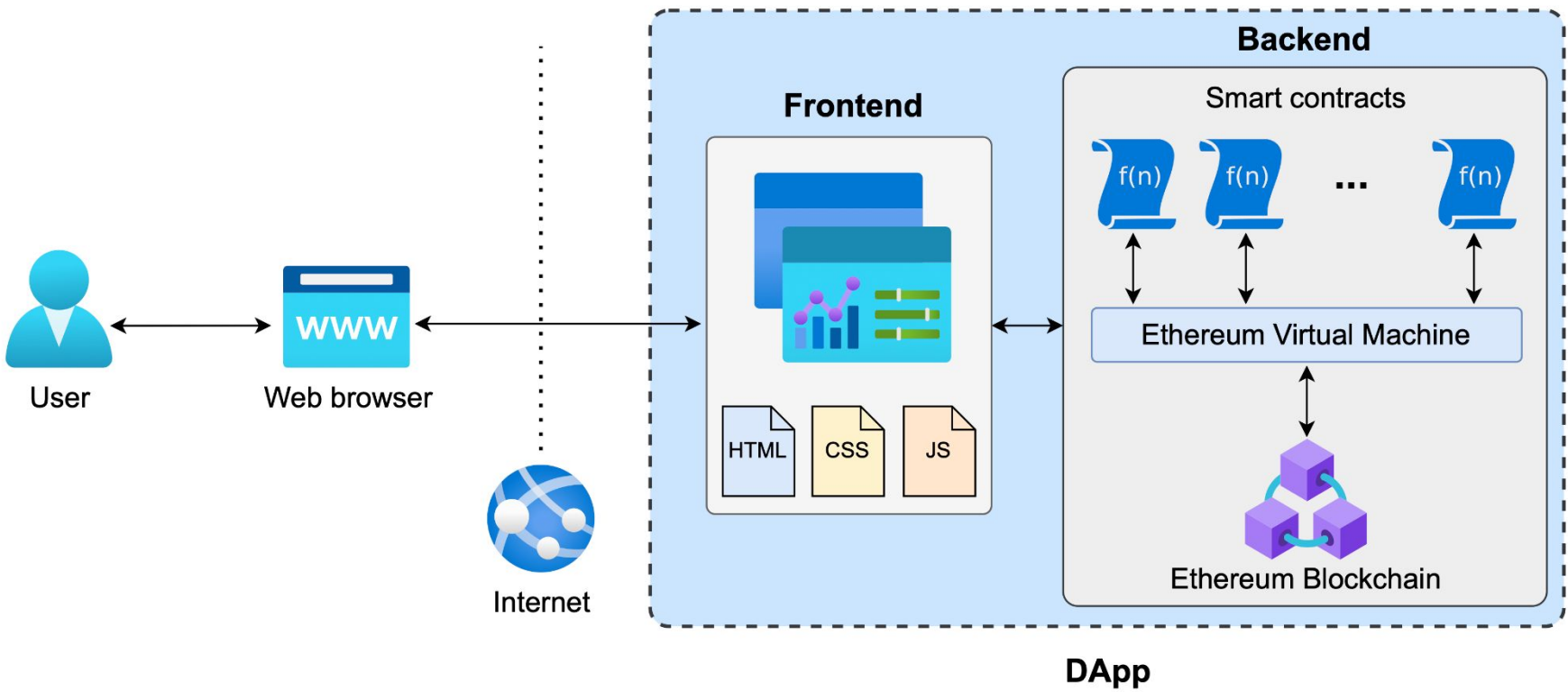
# Ethereum Virtual Machine

Ethereum is a *distributed state machine* whose state changes from block to block.

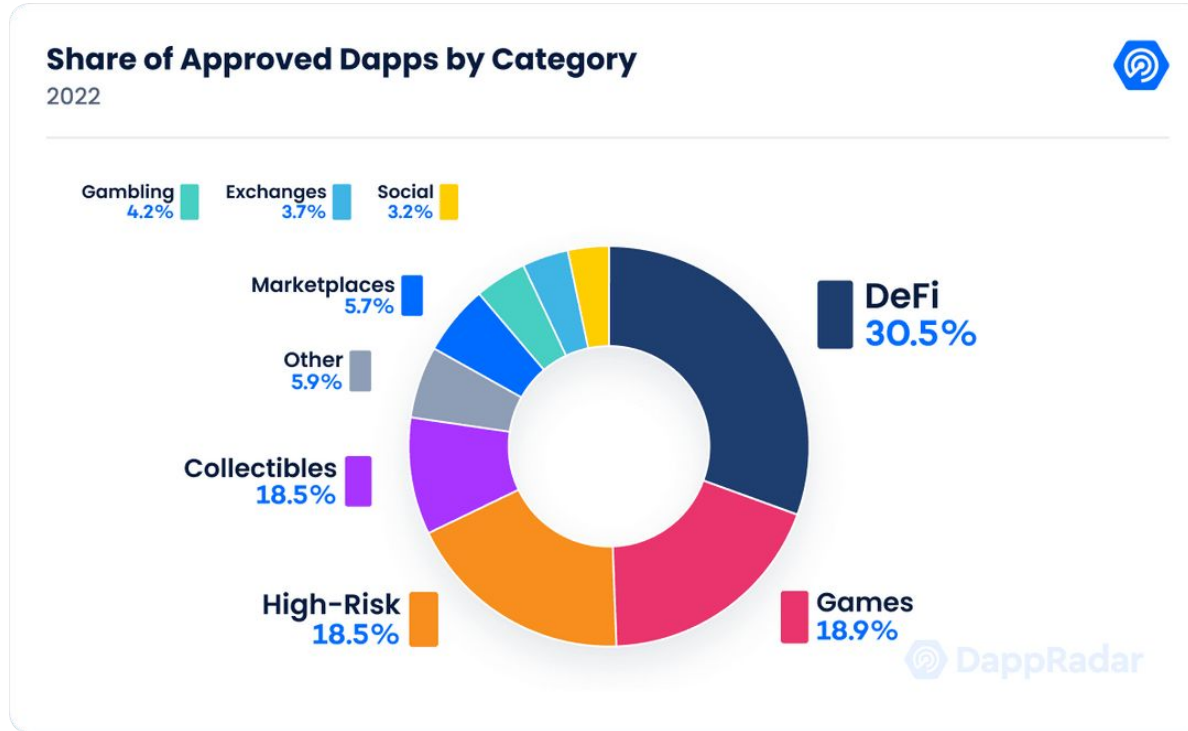
Transition from state  $t$  to  $t+1$  happens by executing transactions in a block.



# Decentralized Applications



# Decentralized Applications



Source: DappRadar. Dapp Industry Report 2022.

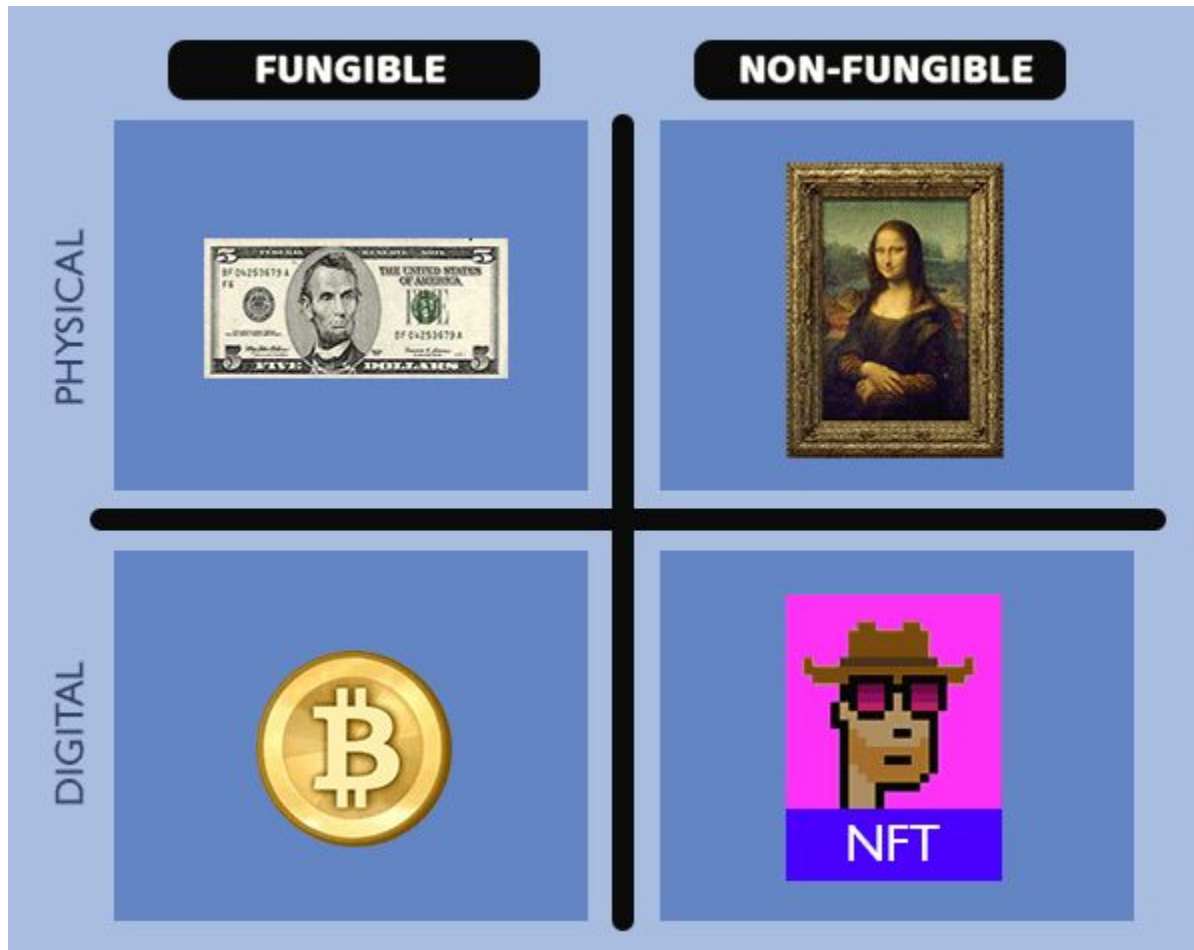
<https://dappradar.com/blog/dapp-industry-report-2022-dapp-industry-proves-resilient-in-crypto-winter>

# Tokens

Represent *digital assets* (i.e., any form of content or value stored in a digital format) that can be owned and transferred between users.

Tokens are implemented through smart contracts.

Transfers (i.e., ownership changes) are recorded securely on the blockchain.



# Fungible tokens



USD Coin  
(USDC)



Chainlink  
(LINK)



Maker  
(MKR)



Basic Attention Token  
(BAT)



Decentraland  
(Mana)



Aave  
(AAVE)



Ox  
(ZRX)



Enjin Coin  
(ENJ)

# Fungible tokens

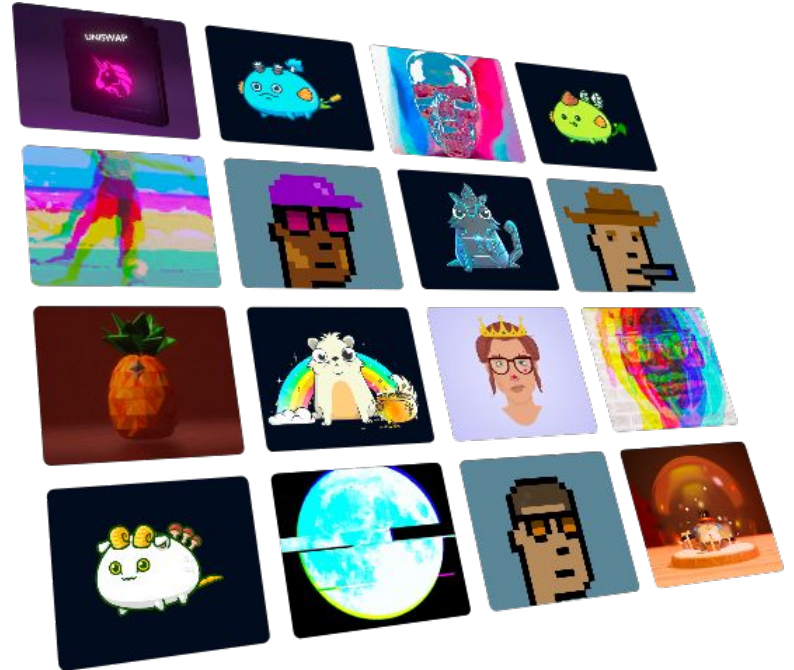
- **Stablecoins**
  - Pegged to a fiat currency to provide price stability.
- **Utility**
  - Used to pay for services on decentralized networks.
- **Governance**
  - Give voting rights in DAOs.
- **Metaverse**
  - Currencies for purchasing virtual land, items, and services.
- **Games**
  - In-game currencies or rewarding mechanisms.

# Non-Fungible Tokens

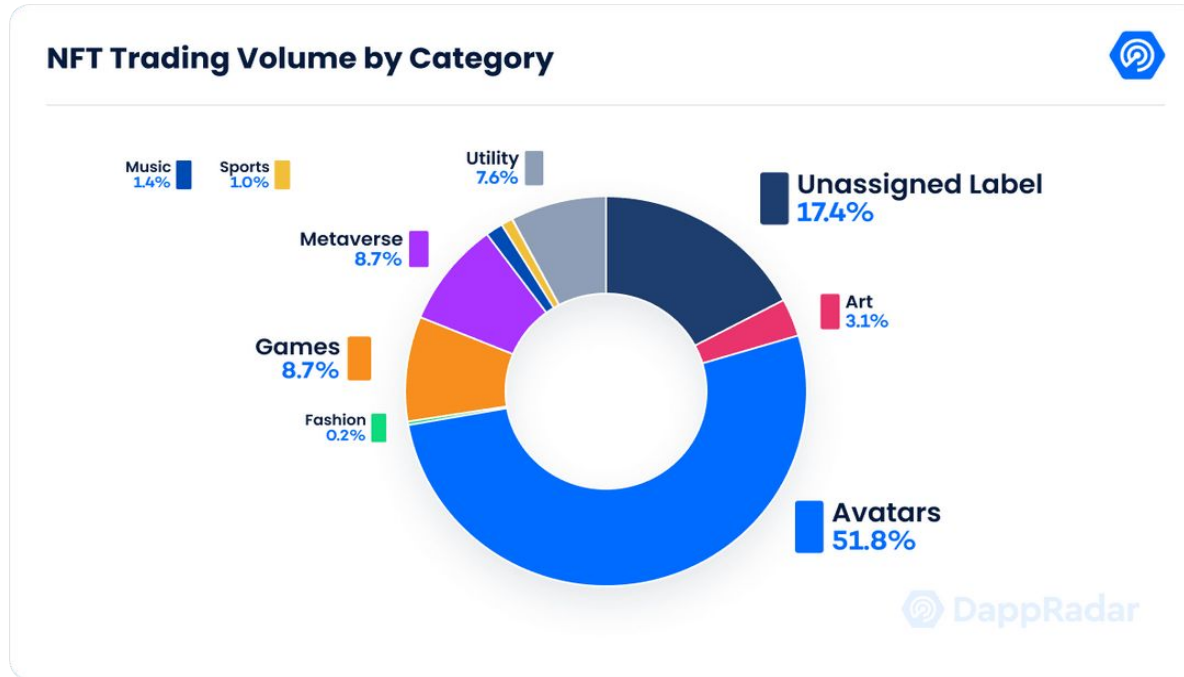
Often abbreviated as **NFTs**.

Represent the ownership of assets with **unique and distinguishable** features.

Linked to a (off-chain) digital representation of the asset (e.g., a picture).



# Non-Fungible Tokens



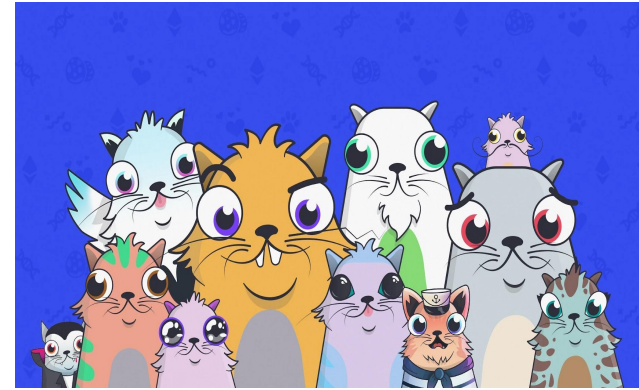
Source: DappRadar (Q3 2022)



Meebits



Bored Apes



CryptoKitties

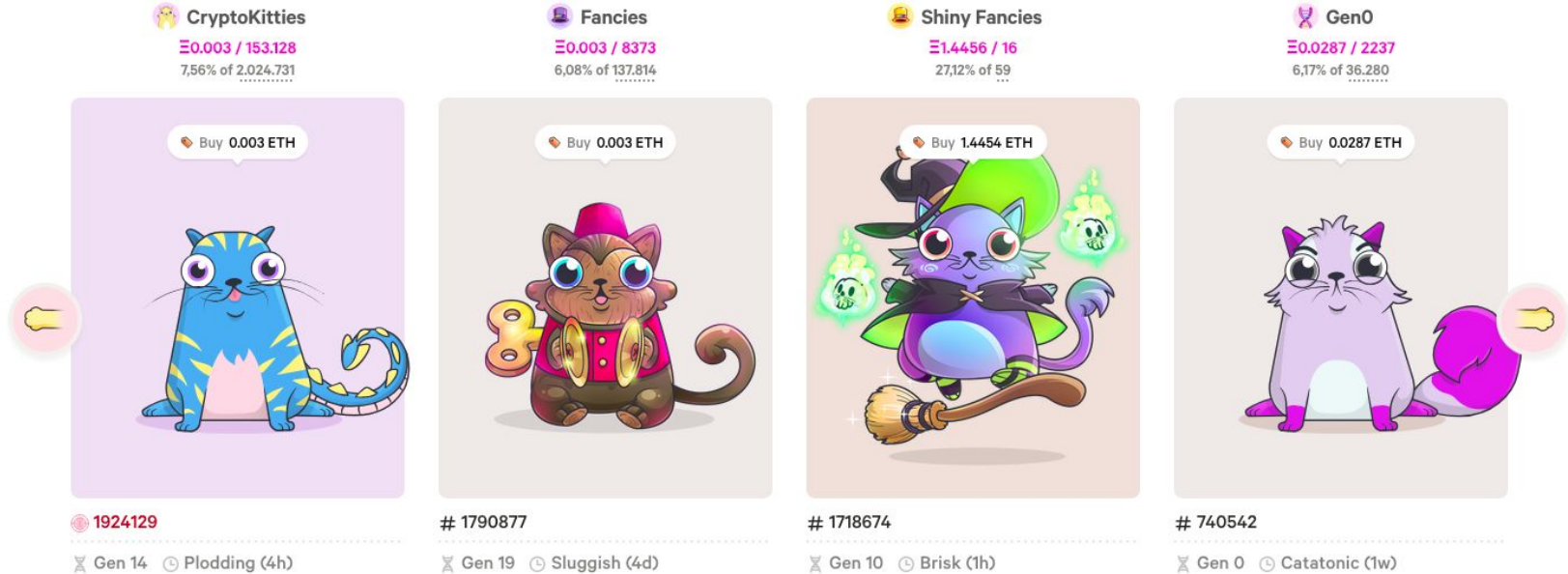


Decentraland



Axie Infinity

# CryptoKitties (2017)



- First project to showcase NFT potential and employ the ERC-721 standard.
- Collectible and breedable digital representations of cats.
- Each CryptoKitty has a unique combination of *attributes*.

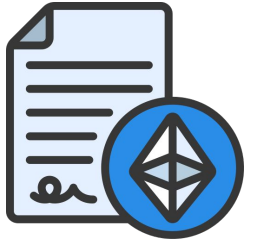


**Snoop Dogg** launched his 10,000 NFT avatars collection on The Sandbox (metaverse game) in November 2024.

# Token standards



# Token standards



A **token contract** keeps:

- a mapping of accounts with token balances;
- a set of functions that define how these balances can be changed.

## **Token standards:**

- 1) specify basic interfaces and allow third party contracts to use tokens in a standardized way;
- 2) set minimum requirements, but do not restrict the design beyond that.

# ERC - 20



## Fungible Tokens

Most basic token standard, used to create interchangeable tokens

Trade-able virtual currencies  
Governance/voting tokens  
Staking tokens



WBTC



USDC



CRV

# ERC - 721



## Non-Fungible Tokens

Basic NFT standard, used to create unique tokens, distinguishable from others in the same collection

Collectable art  
Digital items and property  
Tickets (events, seats, lottery)



CryptoPunks



WMVG



Audioglyph

# ERC - 1155



## Multi-Token Standard

A single interface that manages any combination of multiple token types (fungible, non-fungible, etc).

Alternate to ERC-20 and ERC-721  
Video game items  
Memorabilia



Enjin Tokens



Beanstalk  
Fertilizer



Adidas  
Metaverse

# Token standards

While ERC-20 contracts track the *balances* that belong to each address, ERC-721 contract track the *owner* of a specific token identifier.

ERC20	
address	amount
0x061F...	10
0x25ed...	5
0xab3B...	20

ERC721	
tokenId	address
0	0x0901...
1	0x7A25...
2	0x0901...

# Token standards

## ERC-20

### FUNCTIONS

```
totalSupply()  
balanceOf(account)  
transfer(recipient, amount)  
allowance(owner, spender)  
approve(spender, amount)  
transferFrom(sender, recipient, amount)
```

### EVENTS

```
Transfer(from, to, value)  
Approval(owner, spender, value)
```

## ERC-721

### FUNCTIONS

```
balanceOf(owner)  
ownerOf(tokenId)  
safeTransferFrom(from, to, tokenId)  
transferFrom(from, to, tokenId)  
approve(to, tokenId)  
getApproved(tokenId)  
setApprovalForAll(operator, _approved)  
isApprovedForAll(owner, operator)  
safeTransferFrom(from, to, tokenId, data)
```

---

```
supportsInterface(interfaceId)
```

### EVENTS

```
Transfer(from, to, tokenId)  
Approval(owner, approved, tokenId)  
ApprovalForAll(owner, operator, approved)
```

# Token standards

## ERC-20

### FUNCTIONS

```
totalSupply()  
balanceOf(account)  
transfer(recipient, amount)  
allowance(owner, spender)  
approve(spender, amount)  
transferFrom(sender, recipient, amount)
```

### EVENTS

```
Transfer(from, to, value)  
Approval(owner, spender, value)
```

## ERC-721

### FUNCTIONS

```
balanceOf(owner)  
ownerOf(tokenId)  
safeTransferFrom(from, to, tokenId)  
transferFrom(from, to, tokenId)  
approve(to, tokenId)  
getApproved(tokenId)  
setApprovalForAll(operator, _approved)  
isApprovedForAll(owner, operator)  
safeTransferFrom(from, to, tokenId, data)  


---

supportsInterface(interfaceId)
```

### EVENTS

```
Transfer(from, to, tokenId)  
Approval(owner, approved, tokenId)  
ApprovalForAll(owner, operator, approved)
```

Events are the main focus  
of our analyses.

# Ethereum events

Ethereum smart contract utilize **events** to notify a state update or a particular condition being met during their execution.

Whenever a contract triggers an event, it also generates **logs** (which are included in transaction receipts).

Logs can be used by DApps to perform an action or update their frontend.

# Ethereum events

**Transfer:** emitted every time a token transfer occurs between two addresses.

**Approval:** triggered when a user allows another participant to transfer tokens on their behalf.

Tokens are not actually transferred upon the issuance of the Approval event. In fact, this only occurs through the subsequent issuance of a Transfer event.

# ERC-20 standard (2015)

Solidity signatures of **Transfer** and **Approval** events.

```
event Transfer(address indexed _from, address indexed _to,  
uint256 _value);
```

```
event Approval(address indexed _owner, address indexed _spender,  
uint256 _value);
```

# ERC-20 standard

```
{  
  ...  
  "logs": [{  
    "address": "0xdac17f958d2ee523a2206206994597c13d831ec7",  
    "topics": [  
      "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",  
      "0x000000000000000000000000000000003a9e6cf4e3157670a3b991c25d6f4fcbd9419c03",  
      "0x000000000000000000000000000000002e5c78023c283dde16b6feab95d7d8d8ba1f785e"  
    ],  
    "data": "0x0000000000000000000000000000000000000000000000000000000000000000241b02f00",  
    "blockNumber": "0x8c4731",  
    "transactionHash": "0xb213c5e15ba8c8d86c482fc11ad8e0521f11f2eb464f75ac72a5dc4520f33b53",  
    ...  
  }],  
  ...  
}
```

# ERC-20 standard



# ERC-721 standard (2018)

Solidity signatures of **Transfer** and **Approval** events.

```
event Transfer(address indexed _from, address indexed _to, uint256  
indexed _tokenId);
```

```
event Approval(address indexed _owner, address indexed _approved,  
uint256 indexed _tokenId);
```

# ERC-721 standard (2018)

Solidity signatures of **Transfer** and **Approval**

Third event parameter is marked as indexed (unlike ERC-20).

```
event Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId);
```

```
event Approval(address indexed _owner, address indexed _approved, uint256 indexed _tokenId);
```

# ERC-1155 standard (2018-2019)

- Also known as the **Multi-Token standard**.
- Allows for managing multiple fungible and non-fungible tokens within the same contract.
- Supports **single** and **batch** transfer mode.
- Benefits:
  - reduce transaction costs;
  - simplify DApp development (e.g., need to deploy only one contract);
  - more efficient use of blockchain resources.

# ERC-1155 standard (2018-2019)

```
event TransferSingle(address operator, address from, address to,  
    uint256 id, uint256 value);  
  
event TransferBatch(address operator, address from, address to,  
    uint256 [] ids, uint256 [] values);
```

**Listing 1** Signatures of the ERC-1155 TransferSingle and TransferBatch events in Solidity.

# ERC-1155 standard (2018-2019)

`id[i]` = id of the i-th token being transferred

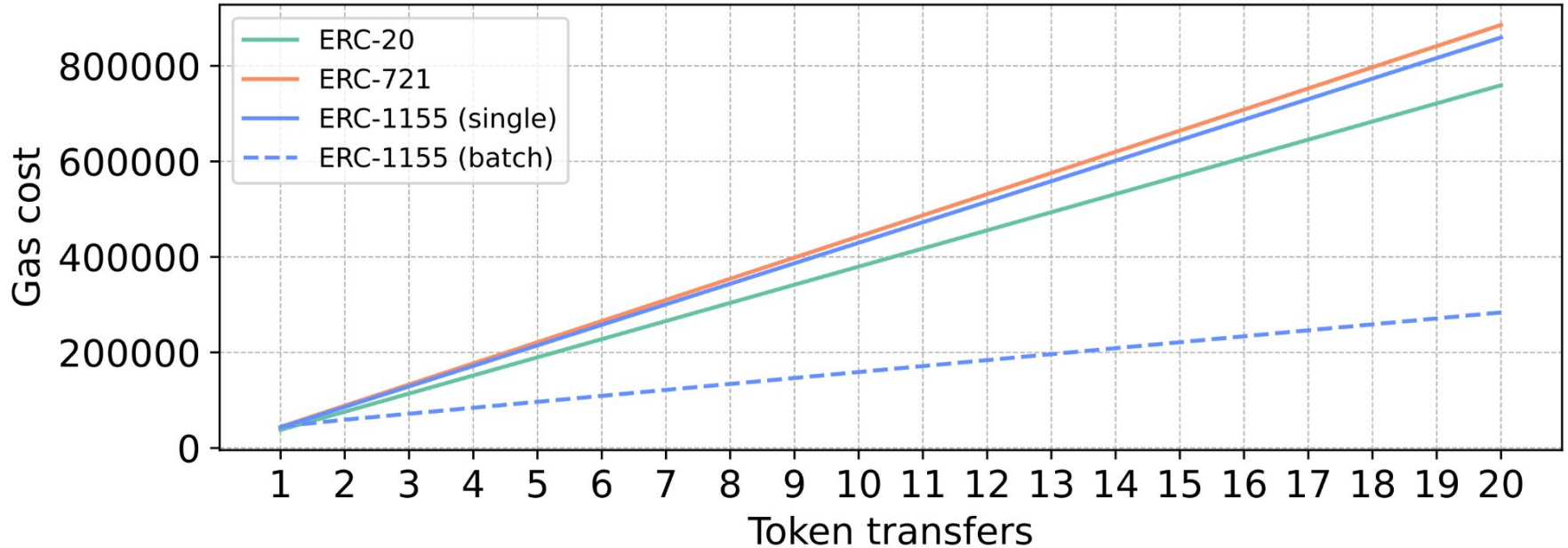
`values[i]` = amount of the i-th token being transferred.

```
event TransferSingle(address operator, address from, address to,  
    uint256 id, uint256 value);  
  
event TransferBatch(address operator, address from, address to,  
    uint256 [] ids, uint256 [] values);
```

**Listing 1** Signatures of the ERC-1155 `TransferSingle` and `TransferBatch` events in Solidity.

# ERC-1155 standard (2018-2019)

Transfer gas cost comparison



# Ethereum token analysis



# Ethereum token analysis

## Dataset

- First 15M blocks of the Ethereum blockchain with TX receipts (i.e., from height 0 to 14999999).
- Time period from July 30th, 2015 03:26:13 PM UTC, to June 21st, 2022 02:28:10 AM UTC.

# Ethereum token analysis

**Table 1** Total number of contracts and triggered transfer events for the ERC-20 and ERC-721 standards

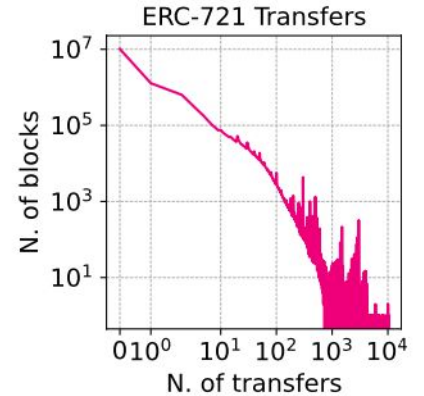
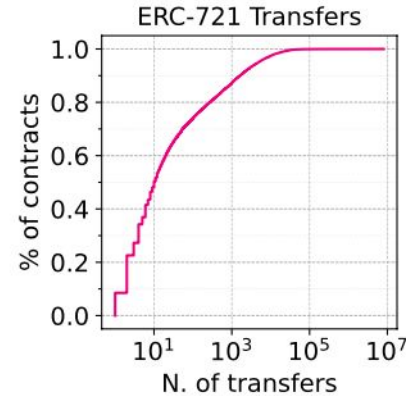
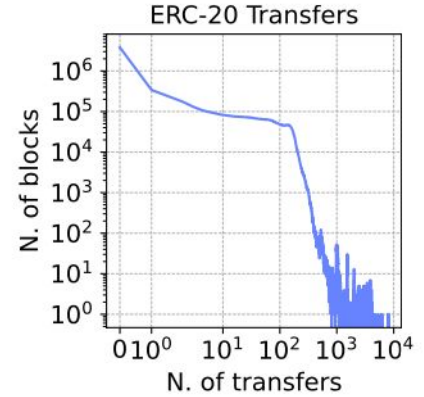
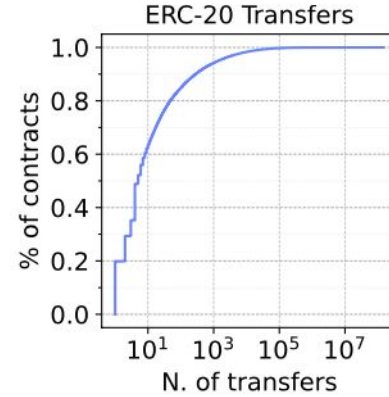
	<b>N. of transfers</b>	<b>N. of contracts</b>
ERC-20	961 603 795	386 615
ERC-721	107 384 863	83 041

ERC-20 tokens are more “liquid” w.r.t. non-fungible assets.

# Ethereum token analysis

Most ERC-20 and ERC-721 contracts trigger a relatively low number of transfer events.

60% (resp. 50%) of ERC-20 (resp. ERC-721) have at most 10 events.

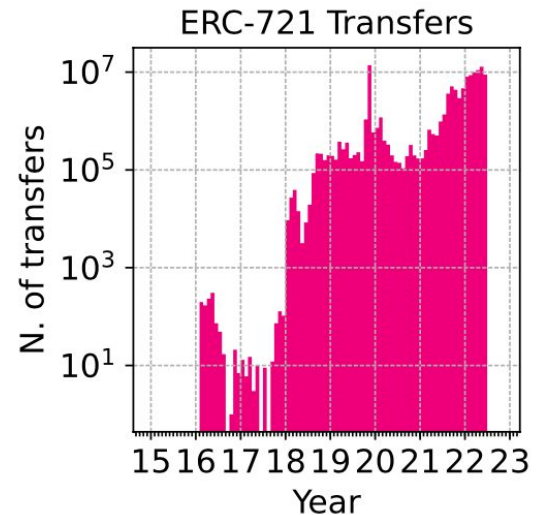
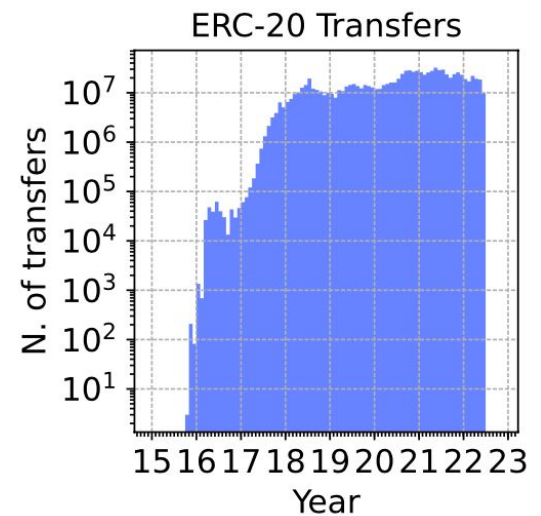


# Ethereum token analysis

Counting the number of token transfers for each month.

ERC-721:

- initially made popular by CryptoKitties (end of 2017);
- significant spike in November 2019.



# Ethereum token analysis



**Table 2** The top 5 services by number of raised ERC-721 transfer events during November 2019

Contract address	Token name	N. of transfers
0e3a2a1f2146d86a604adc220b4967a898d7fe07	Gods unchained cards	6 962 634
629cdec6acc980ebeebea9e5003bcd44db9fc5ce	Gods unchained cards	6 680 552
8853b05833029e3cf8d3cbb592f9784fa43d2a79	Codex record	44 698
bd13e53255ef917da7557db1b7d2d5c38a2efe24	DozerDoll	27 591
fac7bea255a6990f749363002136af6556b31e04	Ethereum name service	25 050
Total (top 5)		13 740 525
Total (global)		13 850 302

Notice how the total number of events raised by these 5 contracts alone is about 99% of all events produced in the same period

The launch of the Gods Unchained Marketplace caused ~ 13% of all Transfers in the data set (7 years).

# Ethereum token analysis



# Ethereum token analysis

Many fungible and non-fungible token contracts (i.e., ~ 60% for ERC-20 and 50% for ERC-721 contracts) show signs of low activity ( $\leq 10$  transfers).

=> Analysis of the top 100 ERC-20 and ERC-721 contracts by number of Transfers to focus on the more flourishing economies.

# Ethereum token analysis

## ERC-20

Token name	N. of transfers
Tether USD	149 408 698
Wrapped Ether	104 183 120
USD coin	42 601 224
Dai stablecoin	14 387 573
ChainLink token	11 388 177
More gold coin	8 947 669
SHIBA INU	7 781 424
Strong	6 964 935
Livepeer token	6 025 932
Cotoken	5 370 855
<b>Total</b>	<b>357 059 607</b>

## ERC-721

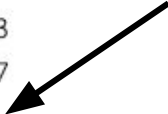
Token name	N. of transfers
Ethereum Name Service	1 837 127
Gods unchained cards	1 028 936
Sorare	815 331
Art blocks	254 442
Foundation	234 387
LucidSight-MLB-NFT	234 247
Sandbox's LANDs	233 665
DusktopiaNFT	201 541
DozerDoll	170 642
Wizards & dragons game	157 118
<b>Total</b>	<b>5 167 436</b>

# Ethereum token analysis

## ERC-20

Token name	N. of transfers
Tether USD	149 408 698
Wrapped Ether	104 183 120
USD coin	42 601 224
Dai stablecoin	14 387 573
ChainLink token	11 388 177
More gold coin	8 947 669
SHIBA INU	7 781 424
Strong	6 964 935
Livepeer token	6 025 932
Cotoken	5 370 855
<b>Total</b>	<b>357 059 607</b>

Known spamming attempt (clogged network in 2019)\*



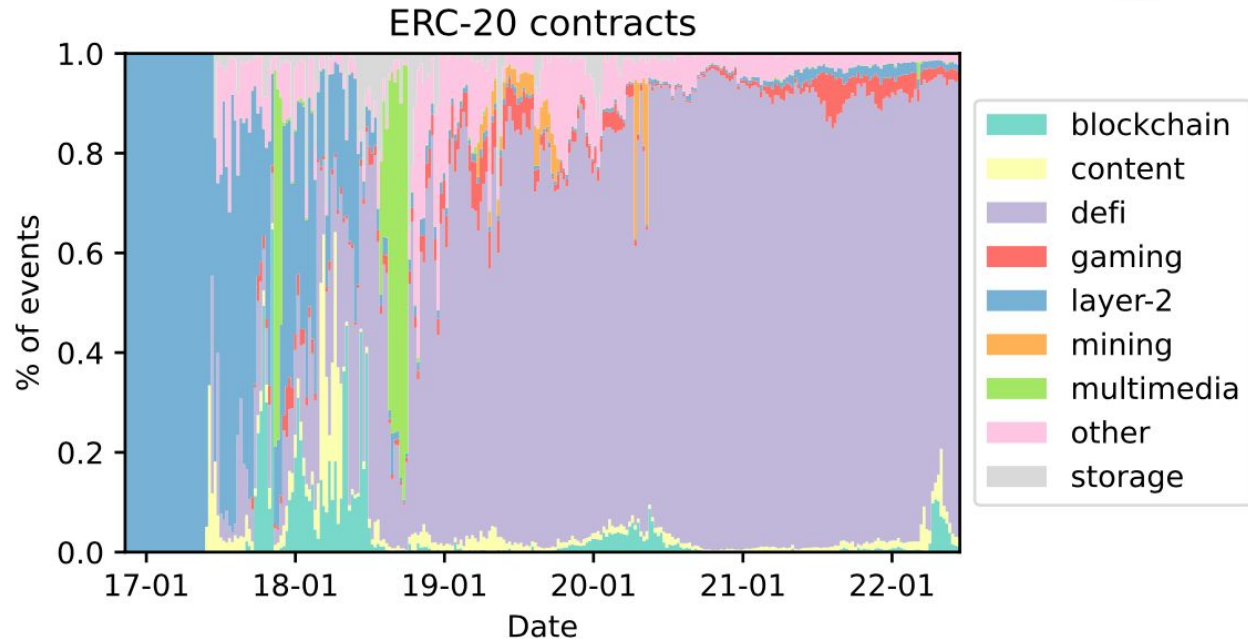
## ERC-721

Token name	N. of transfers
Ethereum Name Service	1 837 127
Gods unchained cards	1 028 936
Sorare	815 331
Art blocks	254 442
Foundation	234 387
LucidSight-MLB-NFT	234 247
Sandbox's LANDs	233 665
DusktopiaNFT	201 541
DozerDoll	170 642
Wizards & dragons game	157 118
<b>Total</b>	<b>5 167 436</b>

\*Source: <https://beincrypto.com/scam-more-gold-coin-clogs-ethereum-network-causing-gas-price-to-spike/>

# Ethereum token analysis

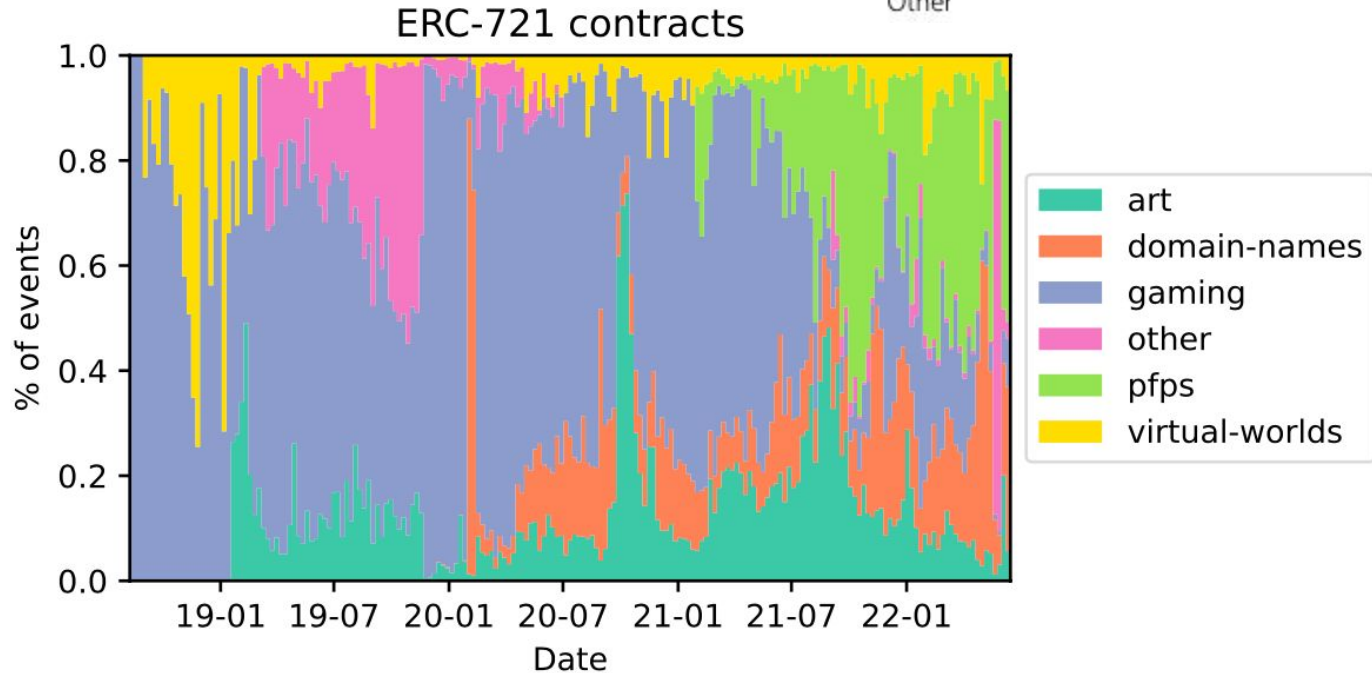
Defi	54
Gaming	9
Blockchain	5
Content	4
Layer-2	4
Storage	4
Mining	3
Multimedia	2
Other	15



Manual classification based on contract application domain.

# Ethereum token analysis

PFPs	48
Gaming	18
Art	14
Virtual-worlds	5
Domain-names	2
Other	13



Contract application domains retrieved from <https://moralis.io>.

# Ethereum token analysis

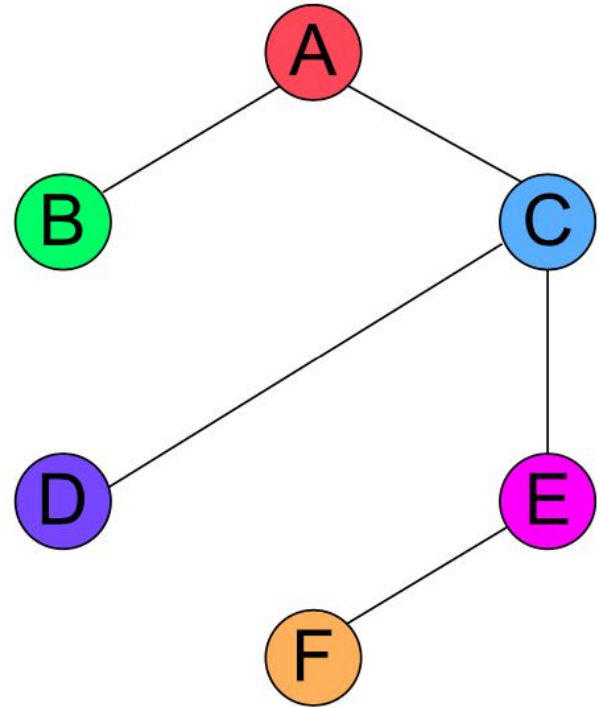
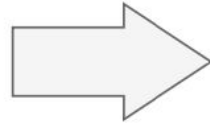
Each token economic community can be represented as a **graph**.

**Transfer event graph:** simple **undirected** graph (unweighted, no self-loops, no repeated edges).

- A node represents an Ethereum address.
- An edge  $(u, v)$  represents a token transfer between addresses  $u$  and  $v$  (in either direction).

# Transfer event graph

Transfer(A, B, 12000)  
Transfer(A, C, 104000)  
Transfer(C, D, 5250000)  
Transfer(E, C, 140500)  
Transfer(E, C, 25250)  
Transfer(E, F, 2000000)



# Ethereum token analysis

Each token network is associated with **5 numerical features** describing the basic topological properties.

<b>Coverage</b>	Percentage of nodes in GCC
<b>Relative diameter</b>	Diameter / Log of n. of nodes
<b>Relative path length</b>	Average shortest path length / Log of n. of nodes
<b>Density</b>	N. of edges / Max possible n. of edges
<b>Transitivity</b>	Global clustering coefficient

# Ethereum token analysis

Graph diameter and average shortest path length have been computed using the **WebGraph** framework [BV04].

- Diameter: **SumSweep** [BCHK+15]

<https://webgraph.di.unimi.it/docs/it/unimi/dsi/webgraph/algo/SumSweepUndirectedDiameterRadius.html>

- Avg. shortest path length: **HyperBall** [BV13]

<https://webgraph.di.unimi.it/docs/it/unimi/dsi/webgraph/algo/HyperBall.html>

[BCHK+15] M. Borassi, P. Crescenzi, M. Habib, W.A. Kusters, A. Marino, F.W. Takes. "Fast diameter and radius BFS-based computation in (weakly connected) real-world graphs: With an application to the six degrees of separation games." Theoretical Computer Science 586 (2015): 59-80.

[BV04] P. Boldi and S. Vigna. "The webgraph framework I: compression techniques". In: Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004. ACM, 2004, pp. 595-602.

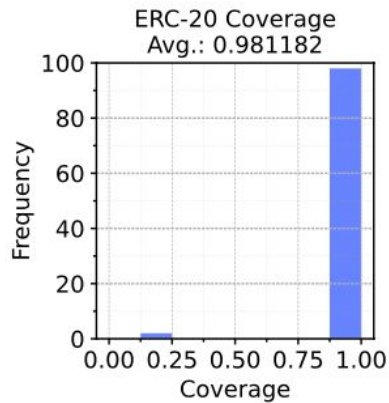
[BV13] P. Boldi and S. Vigna. "In-core computation of geometric centralities with hyperball: A hundred billion nodes and beyond". In: 2013 IEEE 13th International Conference on Data Mining Workshops. IEEE. 2013, pp. 621-628.

# Ethereum token analysis

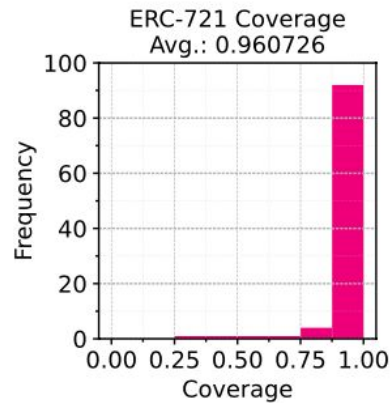
## **Average graph sizes:**

- ERC-20: 759004 nodes and 1701879 edges
- ERC-721: 21283 nodes and 36711 edges

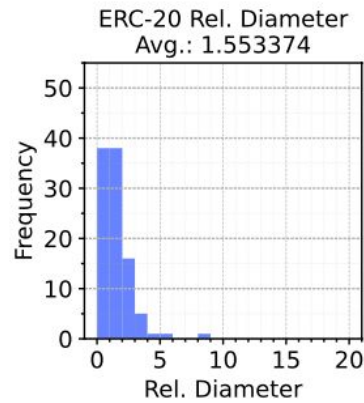
ERC-20 tokens are more liquid (and more numerous) while non-fungible tokens, which usually represent unique and collectible assets, tend to be exchanged less frequently.



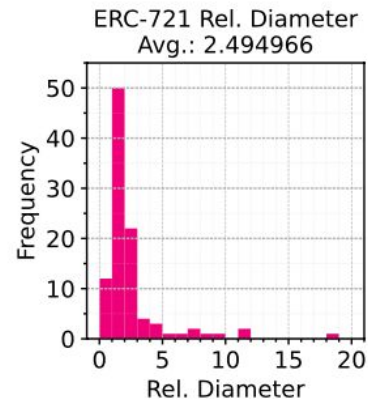
(a)



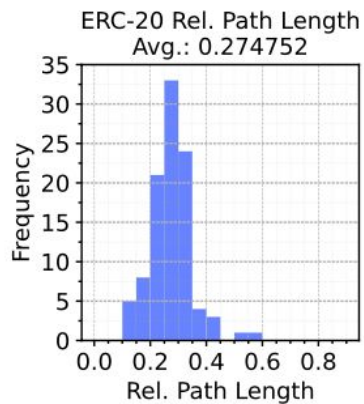
(b)



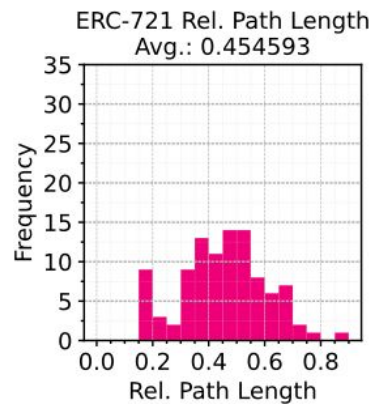
(c)



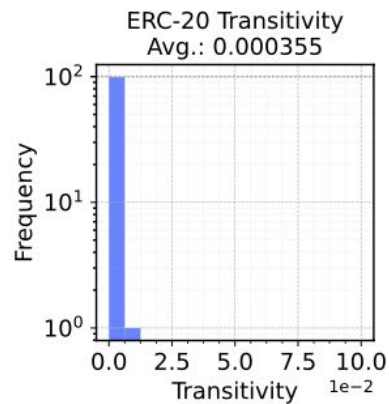
(d)



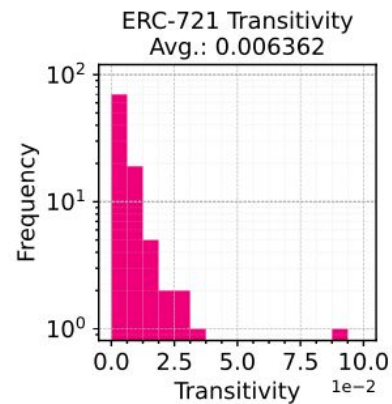
(e)



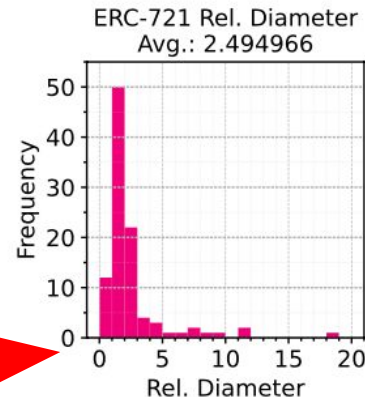
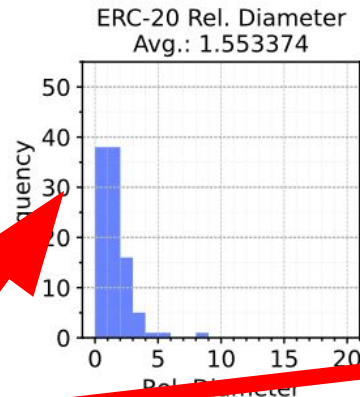
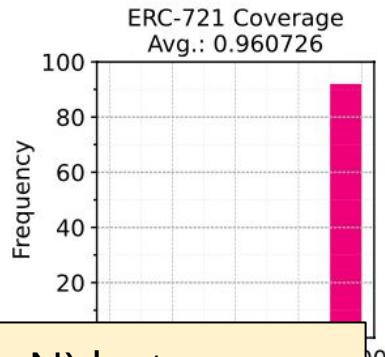
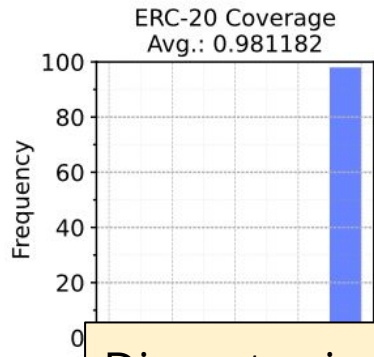
(f)



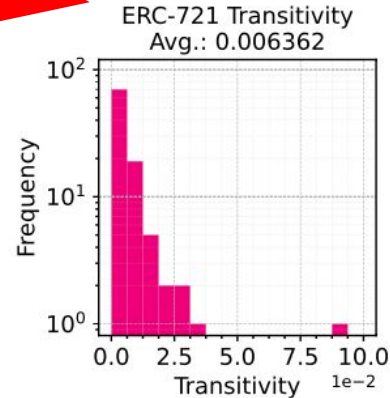
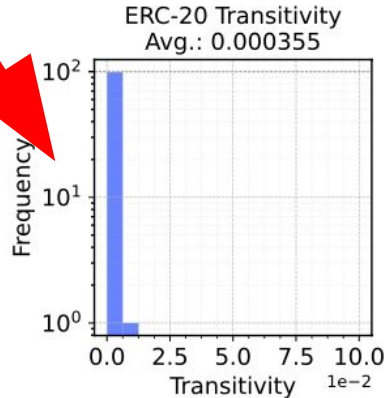
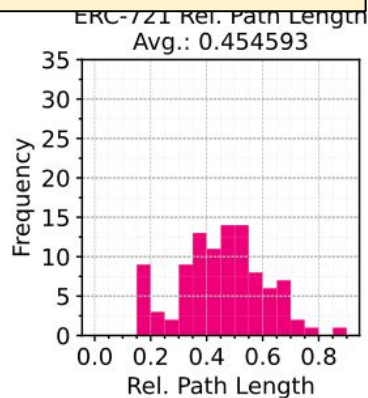
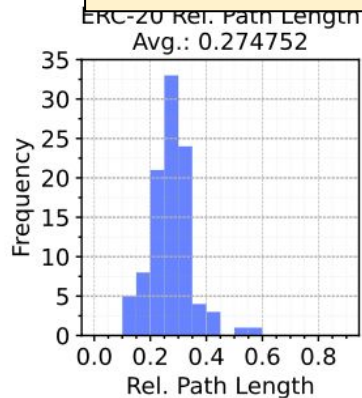
(g)



(h)



Diameter is  $O(\log N)$  but  
transitivity is low  $\Rightarrow$  not a small  
world.



(e)

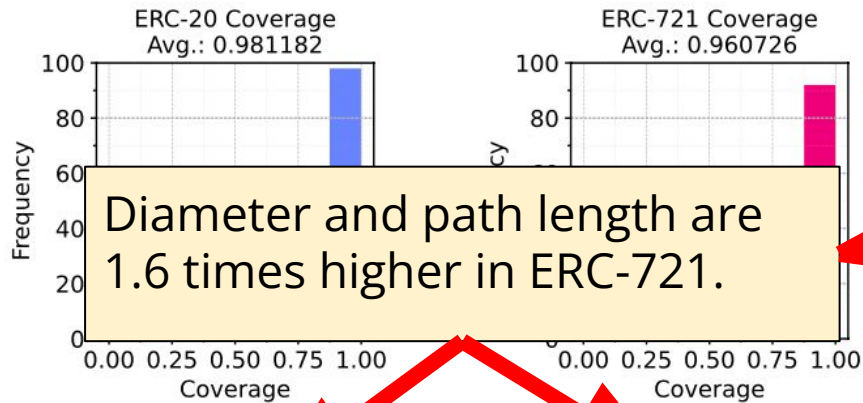
(f)

(g)

(h)

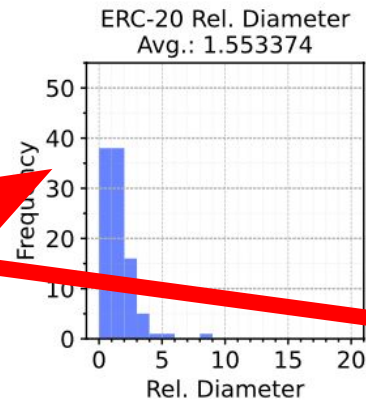
(c)

(d)

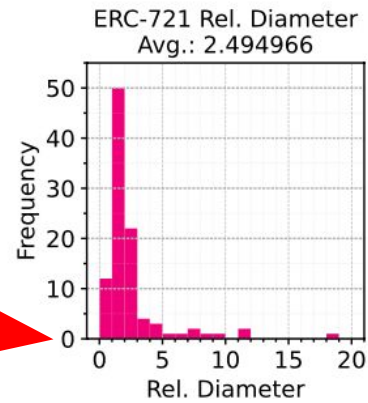


(a)

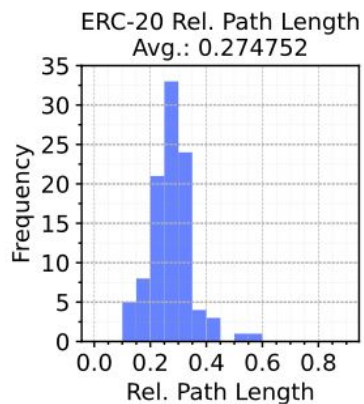
(b)



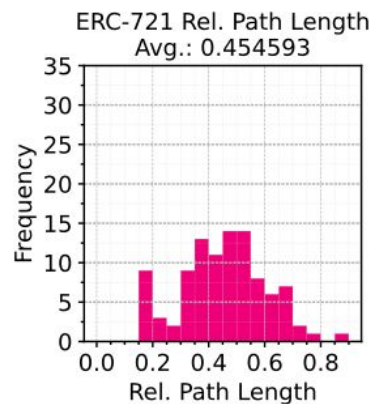
(c)



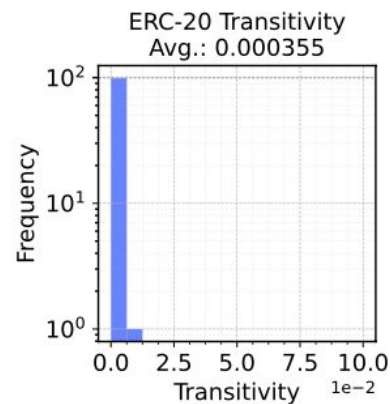
(d)



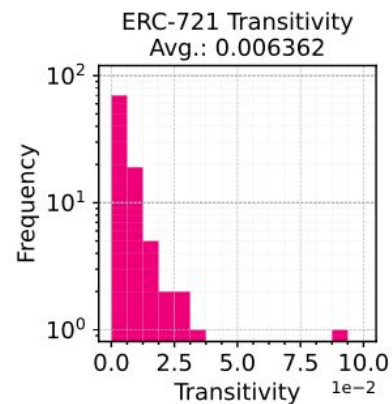
(e)



(f)



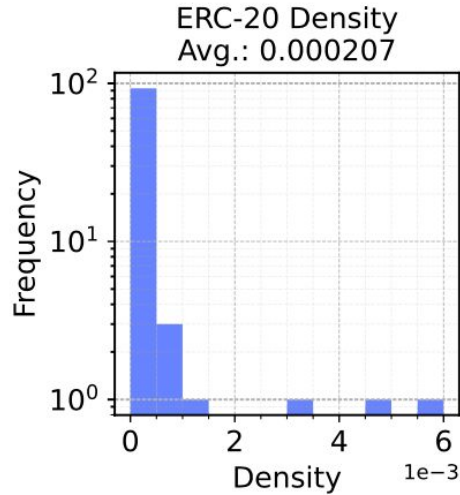
(g)



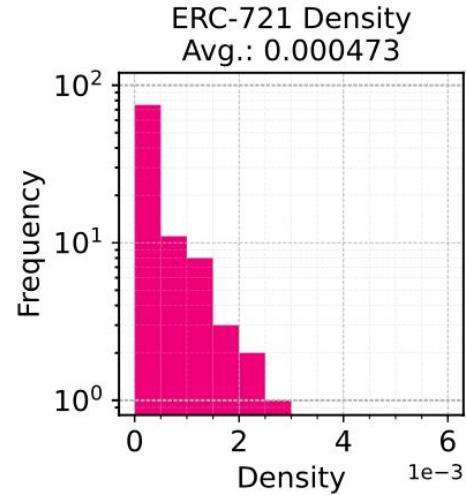
(h)

# Ethereum token analysis

Low density in both scenarios (but 2x for NFTs).



(i)



(j)

# Ethereum token analysis

Two coverage outliers for ERC-20: “Etheal Promo” and “INS Promo”.

Both launched on the market through *airdropping* (free giveaway to multiple users).

# Ethereum token analysis

## Degree distribution analysis

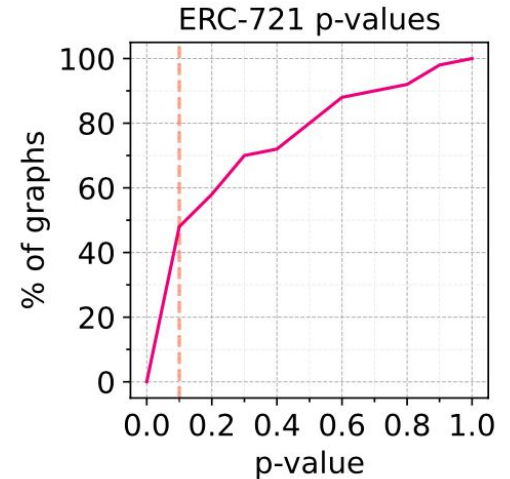
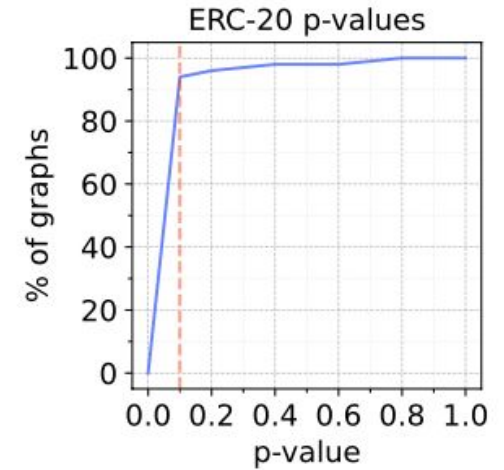
- 1) **Scale-free property** (i.e., degree distribution follows a power law).
- 2) **Degree assortativity**: quantifies the tendency of high-degree nodes (resp. low-degree nodes) to connect with other high-degree (resp. low-degree) nodes.

# Ethereum token analysis

Fitting power laws on degree distribution to test scale-free property.

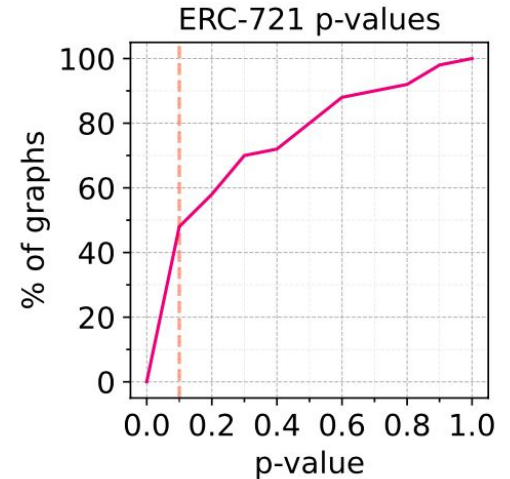
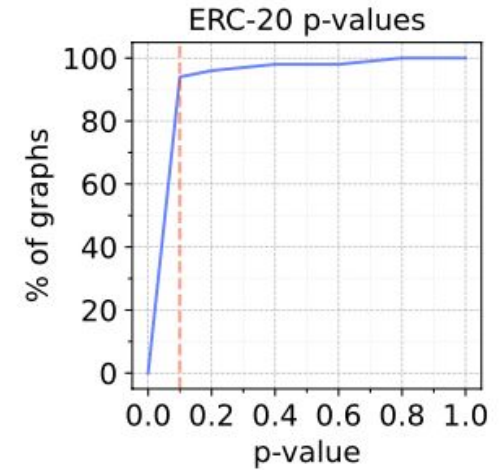
Kolmogorov-Smirnov (KS) test:

- if p-value  $> 0.1$  the power law hypothesis is accepted;
- otherwise it is rejected.



# Ethereum token analysis

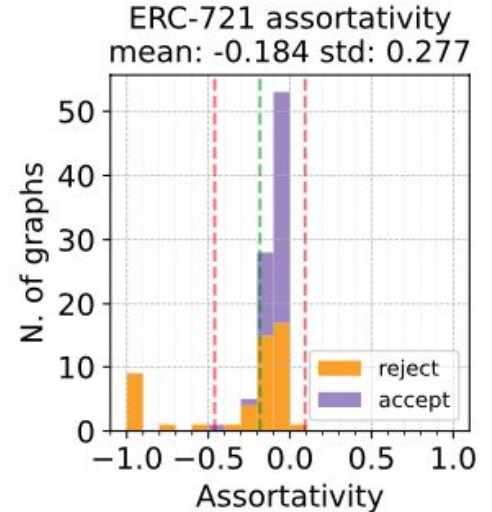
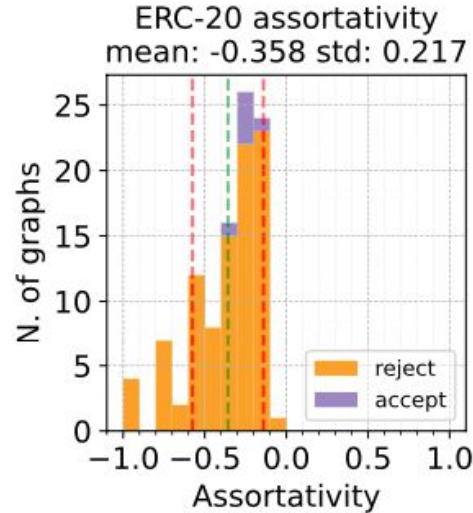
- Only 6% of all ERC-20 networks seem to follow a power law.
- For 51% of ERC-721 graphs the power law hypothesis is acceptable.



# Ethereum token analysis

ERC-20 networks are more disassortative and have a star-like topology (hub-and-spoke).

=> more centralized economies



# Ethereum token analysis


## Similarities:

- Lack of small-world effect (diameter is of the order of  $\log(N)$  but clustering coefficient is generally low).
- Nodes tend to interact with “strangers” (i.e., previously unconnected nodes). Token exchanges are driven by economic factors (whether two users have previously transacted with each other or not is not relevant).
- Low density (interactions tend to be infrequent, weak community structure).


# Ethereum token analysis

## Differences:

- ERC-20 graphs are larger due to the more liquid (and more numerous) nature. Non-fungible tokens tend to be exchanged less frequently.
- ERC-721 diameter and path length are 1.6 times higher, suggesting the existence of longer interaction chains.
- ERC-20 graphs show higher disassortativity due to presence of hubs (e.g., exchanges). NFTs are often transferred directly between users, creating more decentralized economies.



ERC-1155 standard



# ERC-1155 standard



## Multi-token standard

A single contract:

- can manage more than one token;
- can manage tokens with different nature (i.e., fungible and NFT).
- can transfer tokens in **batch**.

Drafted in June 2018 by the Enjin team (a platform for building Web3 games) and approved in June 2019.

# ERC-1155 analysis

## Dataset:

- All Ethereum blocks from height 0 to 21,525,890 (included).
- Time period: from July 30th, 2015 03:26:13 PM UTC to December 31st, 2024 11:59:59 PM UTC.

## Sources:

[LDBR24] M. Loporchio, D. Di Francesco Maesa, A. Bernasconi, and L. Ricci, "Analyzing ERC-1155 Adoption: A Study of the Multi-token Ecosystem," *Studies in Computational Intelligence*. Springer Nature Switzerland, pp. 385–397, 2025.

[LDBR26] M. Loporchio, D. Di Francesco Maesa, A. Bernasconi, and L. Ricci, "ERC-1155 under the lens: a graph-based analysis of the Ethereum multi-token standard," *Appl Netw Sci*, Jan. 2026.

# ERC-1155 analysis

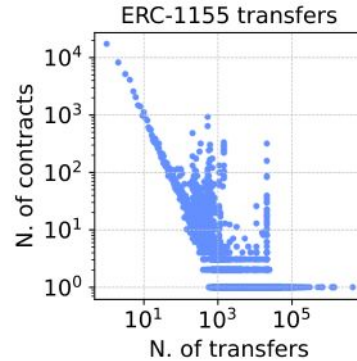
- The analyzed dataset includes 64,252,168 ERC-1155 transfer events produced by 78,526 distinct smart contracts, which manage a total of 8,680,312 different tokens.
- 97% (resp. 3%) of all transfer events are of the TransferSingle (resp. TransferBatch) type.
- All TransferBatch induce 6,126,787 transfers (i.e., an average of 3.7 transfers per event), as opposed to the 62,603,221 produced by TransferSingle events.
- About 91% of the transfers are made in single mode, while only 9% come from batch events.

**=> The batch functionality is used less frequently compared to its counterpart.**

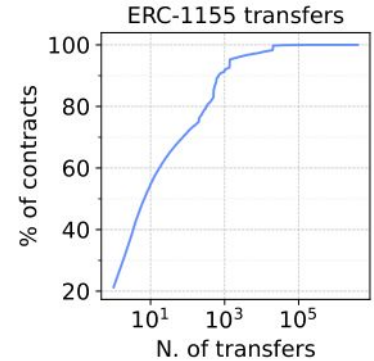
# ERC-1155 analysis

About 20% of all contracts produce only one transfer.

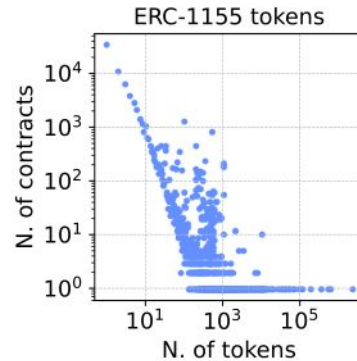
About 40% of all contracts manage a single token (as in ERC-20 and ERC-721).



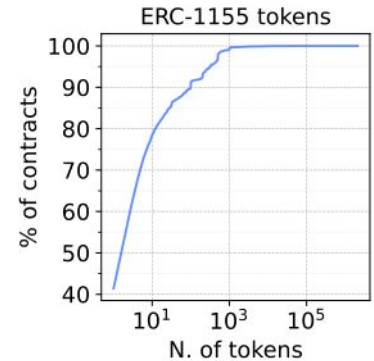
(a)



(b)

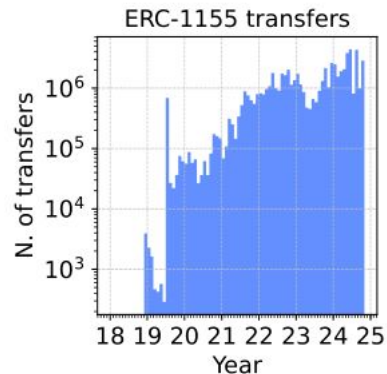


(c)

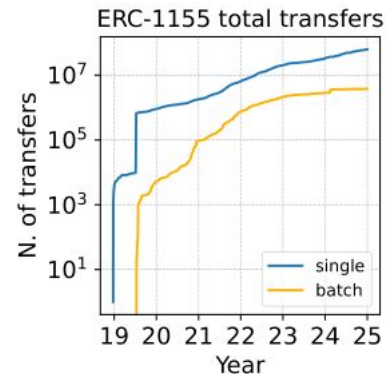


(d)

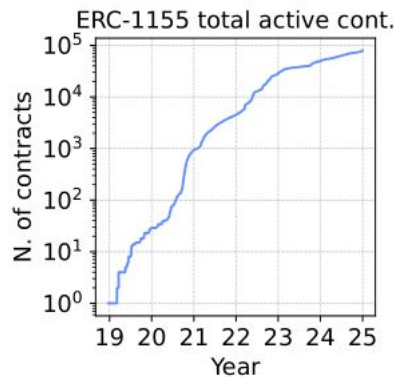
# ERC-1155 analysis



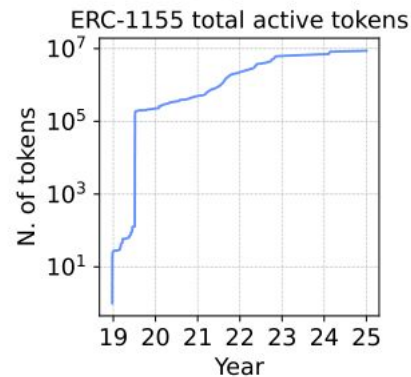
(a)



(b)



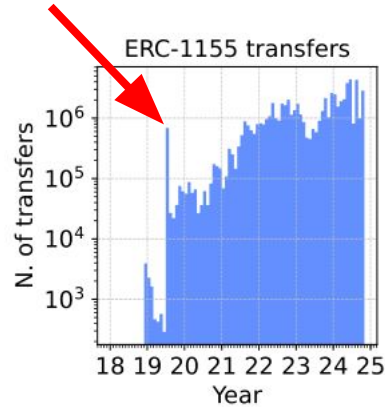
(c)



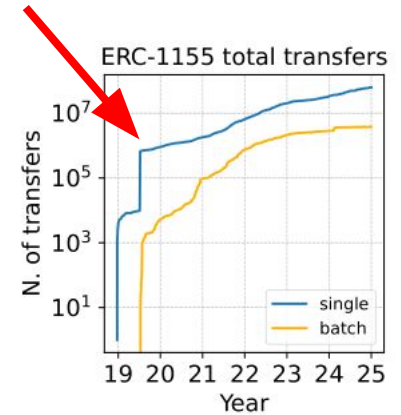
(d)

# ERC-1155 analysis

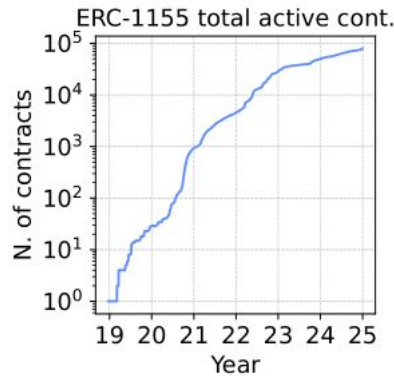
Official launch of Enjin platform (July 2019)



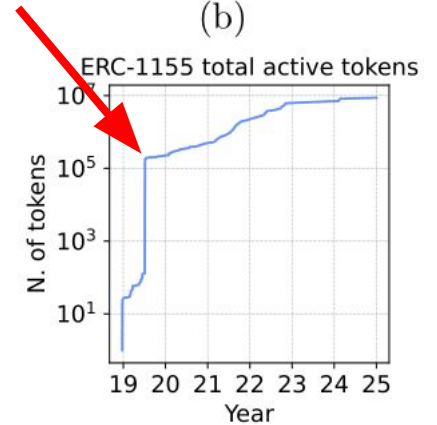
(a)



(b)



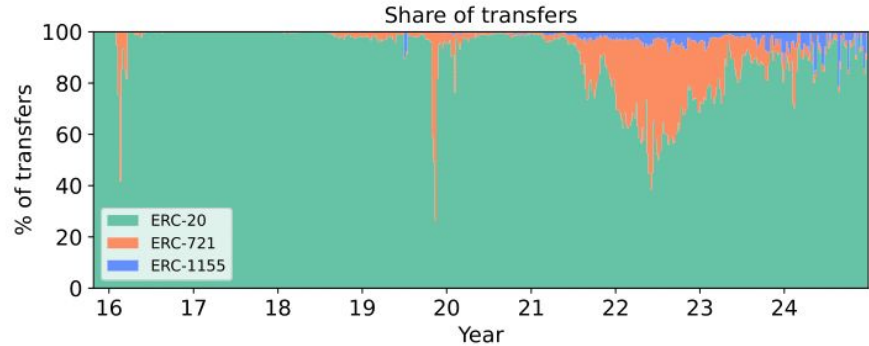
(c)



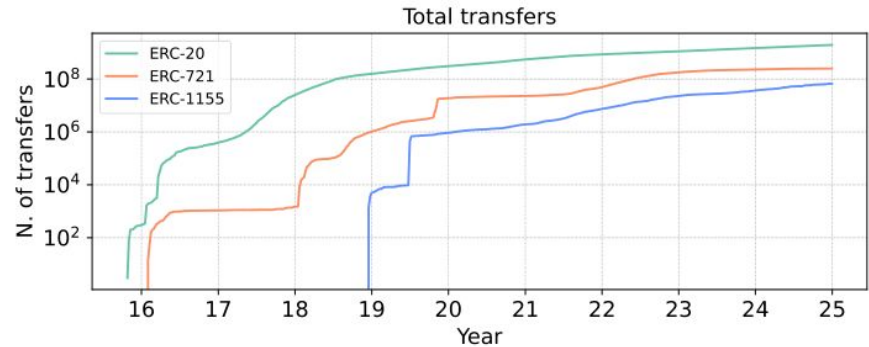
(d)

# ERC-1155 analysis

- ERC-20 transfers are predominant.
- The launch of the Enjin platform in July 2019 boosted ERC-1155 adoption.
- ERC-721 transfer share reflects the NFT boom of 2021.



(a)

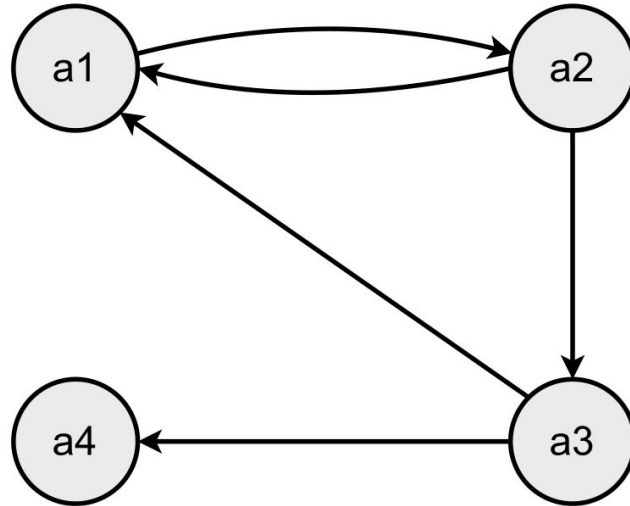


(b)

# ERC-1155 analysis

```
TransferSingle(a1, a1, a2, 10, 100)
TransferSingle(a4, a2, a3, 20, 1)
TransferBatch(a4, a2, a3, [10, 20, 30, 40], [200, 1, 1, 1000])
TransferBatch(a3, a3, a1, [10, 10, 15, 40], [200, 200, 1, 500])
TransferSingle(a2, a2, a1, 10, 1000)
TransferBatch(a3, a3, a4, [10, 20], [50, 50])
```

**Token Transfer Graph:**  
directed and unweighted  
graph model.



# ERC-1155 analysis

## **Normalized degree centralization index:**

1. Sum of the deviations from the maximum degree (either in-degree or out-degree).
2. Result is normalized by dividing the obtained score by that of the most centralized structure with the same number of vertices.

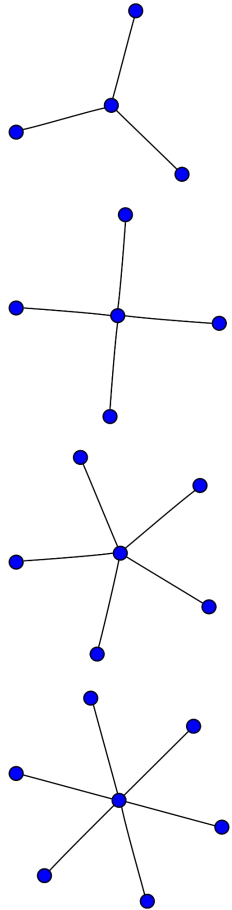
Higher centralization index => more centralized network.

# ERC-1155 analysis

**Star graphs** have the highest centralization score.

If the in-degree (resp. out-degree) centralization index is 1, there is a single central node connected with edges incoming from (resp. outgoing to) all other nodes in the graph.

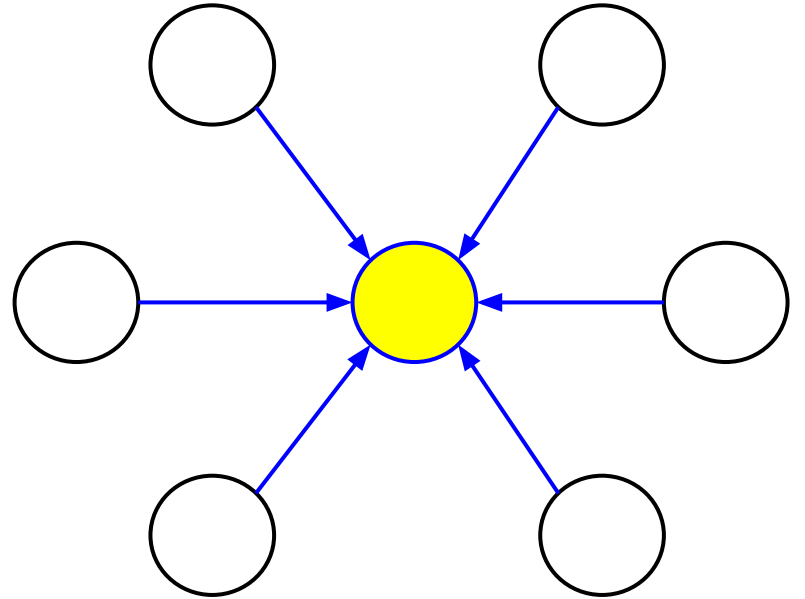
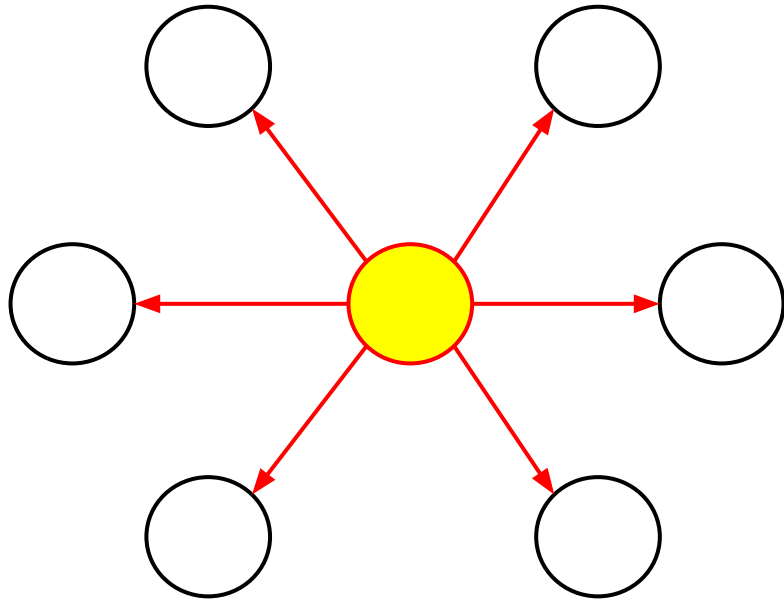
Such nodes do not have any other connections besides those with the central node.



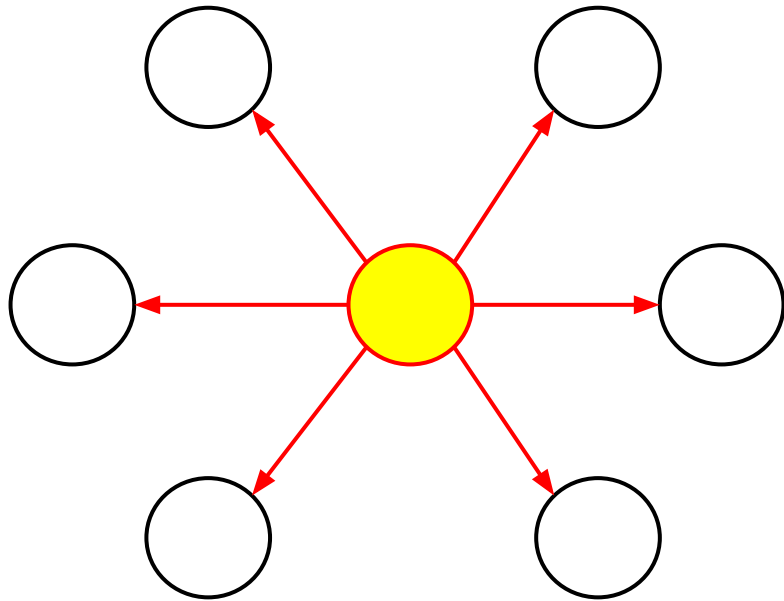
Source: [https://upload.wikimedia.org/wikipedia/commons/thumb/7/7d/Star\\_graphs.svg/3840px-Star\\_graphs.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/7/7d/Star_graphs.svg/3840px-Star_graphs.svg.png)

Source: [Fre78] L. C. Freeman, "Centrality in social networks conceptual clarification," Social Networks, vol. 1, no. 3, pp. 215–239, Jan. 1978.

# ERC-1155 analysis



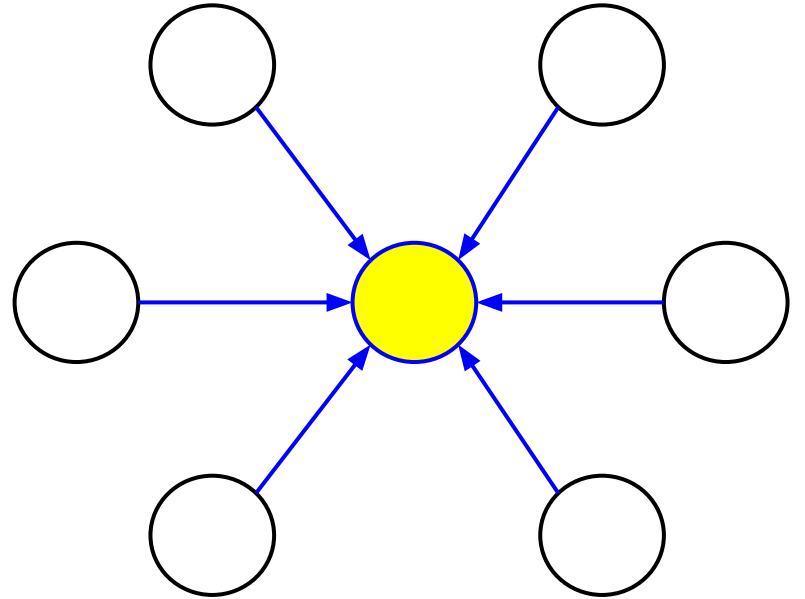
# ERC-1155 analysis



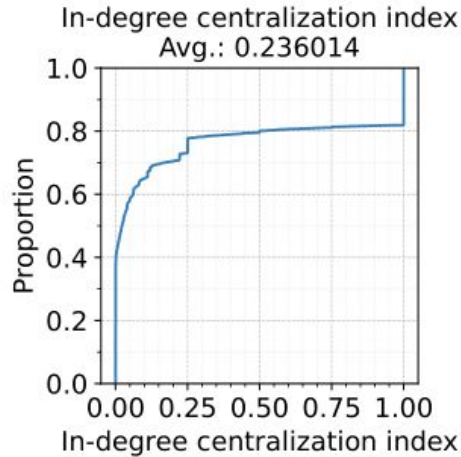
**Out-stars** (i.e., star graphs w.r.t. out-degree) have an out-degree centralization index = 1.

# ERC-1155 analysis

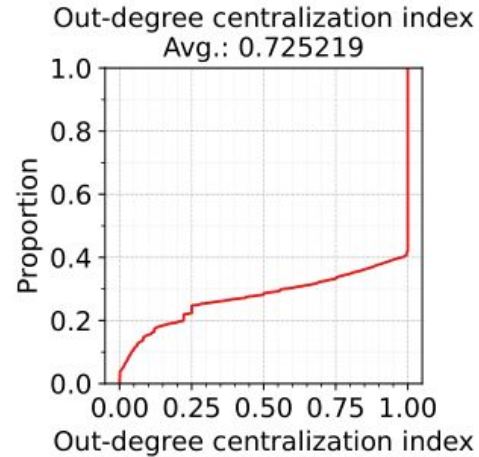
**In-stars** (i.e., star graphs w.r.t. in-degree) have an in-degree centralization index = 1.



# ERC-1155 analysis



(a)



(b)

- 20% of the graphs have an in-degree centralization index = 1.
- 60% of the graphs have an out-degree centralization index = 1.

# ERC-1155 analysis

Category	N. of graphs	Percentage
Stars (in-degree)	157	0.40%
Stars (out-degree)	15 441	39.82%
Degenerate stars	6 864	17.70%
Non-stars	16 319	42.08%
Total	38 781	100.00%

**Table 1** Classification of TTGs based on degree centralization indices.

# ERC-1155 analysis

**Airdrop campaigns** are marketing strategies where users receive free tokens from a single central entity, usually to promote a new project or reward current community members.

Could also be conducted for malicious purposes, such as phishing.

# ERC-1155 analysis

**Airdrop campaigns** are marketing strategies where users receive free tokens from a single central entity, usually to promote a new project or reward current community members.

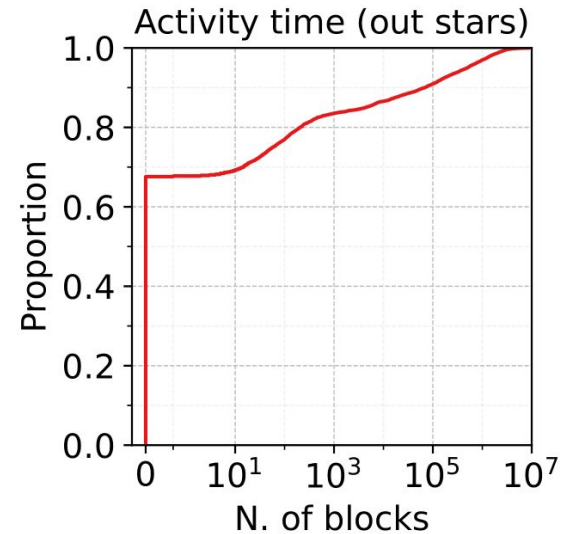
Could also be conducted for malicious purposes, such as phishing.

=> Are ERC-1155 star networks related to airdrop campaigns?

# ERC-1155 analysis



- Alchemy's service\* enables the identification of smart contracts linked to suspicious or fraudulent activities.
- About 80% of the contracts inducing stars for the out-degree are classified as “spam”.
- Activity time of contract: n. of blocks between its first and last recorded transfer in the dataset.
- About 90% of the contracts have an activity time of zero blocks.



\*Source: <https://www.alchemy.com/docs/reference/nft-api-endpoints/nft-api-endpoints/nft-spam-endpoints/is-spam-contract-v-3>

# Ethereum Data Access



How to access Ethereum data?

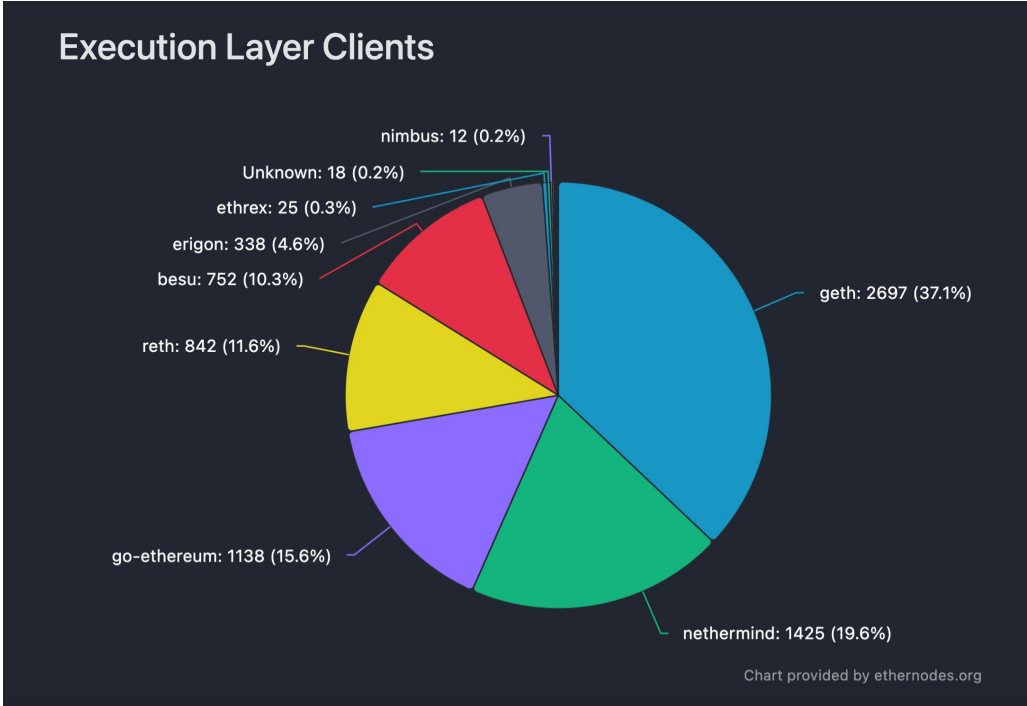


# Ethereum nodes

An Ethereum node is a client that connects to the Ethereum network and maintains a copy of the blockchain.

- 1) **Full nodes:** store all blockchain data and participate in block validation (i.e., keeping track of new blocks and computing and maintaining state changes).
- 2) **Archive nodes:** full nodes that store *historical* blockchain states. While full nodes can calculate such states, archive nodes readily have such information in local storage.

# Ethereum nodes

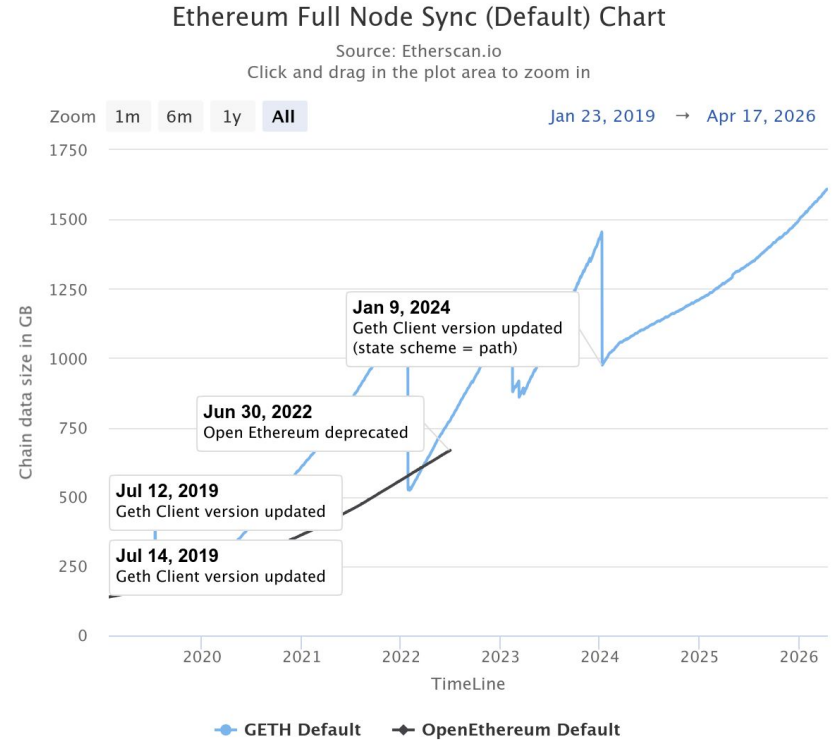


# Ethereum nodes

Running a full/archive Ethereum node today requires significant storage.

- Full node: up to 2 TB of disk space.
- Archive node: over 10 TB of disk space.

**=> Impractical for many users!**



# Third-party services



Etherscan

## Block Explorer

Ethereum block explorer and analytics platform



## Node Providers

Access to the Ethereum node and development tool



## JSON/RPC APIs

API libraries to Interact with the Ethereum node

# AWS Public Blockchain Data

Amazon Web Services offers free access to blockchain data.

Datasets are maintained by Amazon (or other parties) in an S3 bucket.

- Bitcoin
- Ethereum
- Arbitrum
- Aptos
- ...

Reference URL:

<https://registry.opendata.aws/aws-public-blockchain/>



# AWS Public Blockchain Data

- Data are organized in **tables**.
- Stored using the binary and column-based Apache Parquet format (<https://parquet.apache.org>).
- Organized by date.
- Can be easily read using Pandas (or Polars).



# AWS Public Blockchain Data



Ethereum (s3://aws-public-blockchain/v1.0/eth/)

- blocks/date={YYYY-MM-DD}/{id}.snappy.parquet
- transactions /date={YYYY-MM-DD}/{id}.snappy.parquet
- logs/date={YYYY-MM-DD}/{id}.snappy.parquet
- token\_transfers/date={YYYY-MM-DD}/{id}.snappy.parquet
- traces/date={YYYY-MM-DD}/{id}.snappy.parquet
- contracts/date={YYYY-MM-DD}/{id}.snappy.parquet



Full schema documentation (for Ethereum data):

<https://github.com/aws-solutions-library-samples/guidance-for-digital-assets-on-aws/blob/main/analytics/consumer/schema/eth.md>



Boto3 is the AWS SDK for Python (<https://docs.aws.amazon.com/boto3/latest/>).

Can be used to access and download data from the S3 bucket.

```
1 import boto3
2 from botocore import UNSIGNED
3 from botocore.config import Config
4 from datetime import date, timedelta
5 import os
6 import time
7
8 bucket_name = 'aws-public-blockchain'
9 prefix = 'v1.0/eth/logs/'
10 DOWNLOAD_DIR = 'download'
11 start_date = date(2025, 1, 1)
12 end_date = date(2025, 12, 31)
13 delta = timedelta(days=1)
14
15 # Initialize a session using Boto3 with unsigned requests.
16 session = boto3.Session()
17 s3 = session.client('s3', config=Config(signature_version=UNSIGNED))
18
19 # Create download directory if it does not exist.
20 if not os.path.exists(DOWNLOAD_DIR):
21     os.makedirs(DOWNLOAD_DIR)
22
23 current_date = start_date
24 while current_date <= end_date:
25     try:
26         print(f"Processing date: {current_date}...")
27         current_date_str = current_date.strftime('%Y-%m-%d')
28         current_prefix = prefix + 'date=' + current_date_str + '/'
29         response = s3.list_objects_v2(Bucket=bucket_name, Prefix=current_prefix)
30         if 'Contents' in response:
31             # Create a directory for the current date if it doesn't exist
32             current_download_path = os.path.join(DOWNLOAD_DIR, current_date_str)
33             if not os.path.exists(current_download_path):
34                 os.makedirs(current_download_path)
35             for obj in response['Contents']:
36                 object_key = obj['Key']
37                 current_file_path = os.path.join(current_download_path, os.path.basename(object_key))
38                 print(f"Downloading {object_key} to {current_file_path}...")
39                 s3.download_file(bucket_name, object_key, current_file_path)
40         else:
41             print(f"No objects found under the specified prefix for date {current_date}.")
42     except Exception as e:
43         print(f"Error for date {current_date}: {e}")
44     current_date += delta
45     time.sleep(1)
46
```

# Connecting to a third-party full node



ChainList (<https://chainlist.org/>) lists several publicly available Ethereum full nodes.

The number of requests within a given timeframe is often limited.

Ethereum Mainnet RPC URL List					II Sorting
RPC Server Address	Height	Latency	Score	Privacy	
<a href="https://eth.llamarpc.com">https://eth.llamarpc.com</a>			✖	✔	Connect Wallet
<a href="wss://ethereum-rpc.publicnode.com">wss://ethereum-rpc.publicnode.com</a>	24899805	0.062s	✔	✔	
<a href="https://ethereum-rpc.publicnode.com">https://ethereum-rpc.publicnode.com</a>	24899805	0.084s	✔	✔	Connect Wallet
<a href="https://ethereum-mainnet.gateway.tatum.io">https://ethereum-mainnet.gateway.tatum.io</a>	24899805	0.085s	✔	✖	Connect Wallet
<a href="https://eth-mainnet.rpcfast.com?api_key=xbhWBl...">https://eth-mainnet.rpcfast.com?api_key=xbhWBl...</a>	24899805	0.085s	✔	⦿	Connect Wallet
<a href="https://eth.rpc.blxrbdn.com">https://eth.rpc.blxrbdn.com</a>	24899805	0.085s	✔	✖	Connect Wallet
<a href="wss://mainnet.gateway.tenderly.co">wss://mainnet.gateway.tenderly.co</a>	24899805	0.092s	✔	⦿	
<a href="wss://eth.drpc.org">wss://eth.drpc.org</a>	24899805	0.104s	✔	⦿	
<a href="https://0xrpc.io/eth">https://0xrpc.io/eth</a>	24899805	0.109s	✔	✔	Connect Wallet
<a href="wss://0xrpc.io/eth">wss://0xrpc.io/eth</a>	24899805	0.125s	✔	✔	
<a href="https://gateway.tenderly.co/public/mainnet">https://gateway.tenderly.co/public/mainnet</a>	24899805	0.136s	✔	✖	Connect Wallet
<a href="https://mainnet.gateway.tenderly.co">https://mainnet.gateway.tenderly.co</a>	24899805	0.174s	✔	✖	Connect Wallet
<a href="https://ethereum.public.blockpi.network/v1/rpc/pu...">https://ethereum.public.blockpi.network/v1/rpc/pu...</a>	24899805	0.174s	✔	⚠	Connect Wallet
<a href="https://eth.api.onfinality.io/public">https://eth.api.onfinality.io/public</a>	24899805	0.178s	✔	⚠	Connect Wallet
<a href="https://rpc.eth.gateway.fm">https://rpc.eth.gateway.fm</a>	24899805	0.178s	✔	✖	Connect Wallet
<a href="https://api-ethereum-mainnet-reth.n.dwellir.com/2...">https://api-ethereum-mainnet-reth.n.dwellir.com/2...</a>	24899805	0.203s	✔	⚠	Connect Wallet
<a href="https://eth1.lava.build">https://eth1.lava.build</a>	24899805	0.204s	✔	✖	Connect Wallet

# Connecting to a third-party full node

It is possible to connect to a full node using the web3.py Python library.

Documentation:

<https://web3py.readthedocs.io/en/stable/>

```
▶ from web3 import Web3

PROVIDER_URL = "https://eth.blockrazor.xyz"

w3 = Web3(Web3.HTTPProvider(PROVIDER_URL))

# Check connection
if not w3.is_connected():
    raise RuntimeError("Provider not reachable")

# Get client version and chain ID
client = w3.client_version
chain_id = w3.eth.chain_id

print(f"Connected: {client}")
print(f"Chain ID: {chain_id}")
print(f"Latest block: {w3.eth.block_number}")
```

```
... Connected: Geth/v1.16.7-stable-4bf6c387/linux-amd64/go1.24.5
Chain ID: 1
Latest block: 24900488
```

```

1  from web3 import Web3
2  import time
3
4  INPUT_FILE = "opensea_token_list.tsv"
5  OUTPUT_FILE = "opensea_token_supply.tsv"
6
7  w3 = Web3(Web3.HTTPProvider("https://eth.blockrazor.xyz"))
8
9  abi = "[{"inputs":[{"internalType":"uint256","name":"id","type":"uint256"}], " + \
10 "\name":"totalSupply","outputs":[{"internalType":"uint256","name":"","type":"uint256"}], " + \
11 "\stateMutability":"view","type":"function"}]"
12
13 input_fh = open(INPUT_FILE, "r")
14 output_fh = open(OUTPUT_FILE, "w")
15 lines = input_fh.readlines()
16 input_fh.close()
17
18 try:
19     output_fh.write("address\ttoken_id\ttotal_supply\n")
20     count = 1
21     total_lines = len(lines)
22     for line in lines:
23         parts = line.strip().split("\t")
24         if len(parts) != 2:
25             continue
26         address = Web3.to_checksum_address(parts[0])
27         token_id = int(parts[1])
28         contract_instance = w3.eth.contract(address=address, abi=abi)
29         print(f"Querying address: {address}, token: {token_id}... ({count}/{total_lines}")
30         total_supply = contract_instance.functions.totalSupply(token_id).call()
31         output_fh.write(f"{address}\t{token_id}\t{total_supply}\n")
32         count += 1
33         time.sleep(0.2)
34 except Exception as e:
35     print(f"Error: {e}")
36 finally:
37     input_fh.close()
38     output_fh.close()
--

```

Thank you  
for your attention



This presentation is based on previous materials by the following authors, whom I sincerely thank:

- Dr. Damiano Di Francesco Maesa.
- Dr. Natkamon Tovanich.

# References

- [BCHK+15] M. Borassi, P. Crescenzi, M. Habib, W.A. Kosters, A. Marino, F.W. Takes. "Fast diameter and radius BFS-based computation in (weakly connected) real-world graphs: With an application to the six degrees of separation games." *Theoretical Computer Science* 586 (2015): 59-80.
- [BV04] P. Boldi and S. Vigna. "The webgraph framework I: compression techniques". In: *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*. ACM, 2004, pp. 595–602.
- [BV13] P. Boldi and S. Vigna. "In-core computation of geometric centralities with hyperball: A hundred billion nodes and beyond". In: *2013 IEEE 13th International Conference on Data Mining Workshops*. IEEE. 2013, pp. 621–628.
- [Fre78] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, Jan. 1978.
- [LDBR24] M. Loporchio, D. Di Francesco Maesa, A. Bernasconi, and L. Ricci, "Analyzing ERC-1155 Adoption: A Study of the Multi-token Ecosystem," *Studies in Computational Intelligence*. Springer Nature Switzerland, pp. 385–397, 2025.
- [LDBR26] M. Loporchio, D. Di Francesco Maesa, A. Bernasconi, and L. Ricci, "ERC-1155 under the lens: a graph-based analysis of the Ethereum multi-token standard," *Appl Netw Sci*, Jan. 2026.