

The World of Tokenization

What are the key factors to evaluate and develop on-chain token economies?

—

Binance Research - August 23rd 2019

KEY TAKEAWAYS

- Tokens have brought significantly higher utility for the blockchain industry as a whole, allowing the technology to be built into almost all existing industries.
- **Ethereum (ETH)** is currently the **most used blockchain** worldwide for developers to **issue new tokens**. Specifically, there are **hundreds of thousands of tokens** already issued on Ethereum, along with many token standards either fully developed or in progress, such as ERC-1400/ERC-1410 for security tokens. On top of this, a wide variety of companies are creating **security branded tokens** like Polymath or Securitize.
- There are dozens of token standards already on the Ethereum blockchain and introduced as **“EIPs”**. These include numerous customizable elements such as **fungibility status, minting/burning features, and potential transfer restrictions**.
- Partially owing to **Ethereum’s scalability constraints, competing programmable blockchains also support their own token standards**, either natively or constructed (i.e., through the deployment of smart contracts).
- A native standard refers to a blockchain that natively supports the creation of specific tokens on its chain, with **Binance Chain being a prime example**. On the other hand, a constructed standard is a blockchain whose tokens are supported as part of a smart-contract function, Ethereum being the most famous example.
- Other popular competing blockchains include **EOS, Ontology**, and **Tron** or **second layers** running on blockchains like **Simple Ledger Protocol for Bitcoin Cash**.

- In the expanding tokenization of assets, this competition occurs on different aspects such as:
 - **DApp availability or growing use cases:** the more applications a blockchain has, the greater the value proposition is.
 - **Transactions and pace:** the faster a blockchain is, the more appealing it generally becomes. On-chain transaction frequency also reflects the popularity of a network.
 - **Blockchain fees:** the lower on-chain and token issuance fees are, the higher the incentives to interact within the on-chain ecosystem.
 - **Easiness to build:** elements like a test-net, EVM-compatibility & number of smart-contract languages play a key factor in attracting users.
 - **Security and blockchain development:** general activity on the blockchain, reflected by the number of improvement proposals or the count of developers on layer 1, is generally an accurate measure of the health of a blockchain. In addition, blockchain security remains essential with vector attacks and past attacks being focal points.
 - **(De)centralization:** the degree of centralization impacts the value proposition of a chain, particularly from the perspective of potential token stakeholders.
- Specifically, these above key aspects are considered by different stakeholders, such as **investors and token-holders, prospective and existing application developers, and projects when looking to issue tokens.**
- Yet, **“impossible trade-offs” remain**, which ultimately stem **from the blockchain triangle** (i.e., scalability, security, and decentralization cannot be achieved simultaneously). As a result, key drivers such as **on-chain metrics, level of adoption, or the size of the developer pools** remain as some of the core factors to consider.
- In the long-run, a **wide variety** of programmable blockchains **will likely coexist** if interoperability solutions across chains develop and prove to be secure and usable.

Since 2016, hundreds of thousands of tokens have been created on various blockchains, with the vast majority deployed on the Ethereum blockchain. Amidst the launches of newer blockchains that support the minting of new tokens with supplementary token standards, this report discusses and analyzes the diversity of token standards across blockchains.

The report begins with an overview of token standards across more than twenty blockchains and second-layer architectures, before focusing on ten of the largest and most-used blockchains. Finally, blockchains are analyzed through a **multi-dimensional prism** in order to distinguish **key important factors** creating a compelling proposition from the perspective of developers, projects, and token-holders worldwide.

1. Tokenization standards on Ethereum

Ethereum¹ was the first programmable blockchain, designed to serve as the “world computer”.

In 2013, the whitepaper for Ethereum was written by Vitalik Buterin². It described an open-source, public, blockchain-based distributed computing platform that could run smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference.

Ethereum makes it possible for developers **to build and deploy smart contracts**, as well as **issue their own cryptocurrency** directly on the Ethereum blockchain removing the need for developers to create their own new blockchain for their service. It saves developers not just the time it takes to create a blockchain but also allows them to leverage the security and decentralization of Ethereum.

As a result, **Ethereum has become the “go-to” blockchain** standard for creating utility tokens and raising capital. Furthermore, new decentralized applications such as DeFi applications have led to **positive network effects**, attracting most of the blockchain developers to build on Ethereum.

We can distinguish utility tokens from security tokens at the code level, with security tokens referring to tokens whose built-in features are intended to make them **compliant with both existing and future securities regulations**. Specifically, security token standards introduce new methods for issuers, such as whitelisting of wallet addresses, transfer and ownership restrictions, and the creation of central authorities.

1.1 Utility tokens

Regarding utility tokens on Ethereum, the most prevalent standards are **ERC-20** for fungible tokens and **ERC-721** for non-fungible tokens. This subsection discusses many token standards: adopted, in development or drafted versions.

[ERC-20](#)

ERC-20 is a technical standard used for smart contracts on the Ethereum blockchain for implementing tokens. Specifically, it refers to a common set of rules that an Ethereum token needs to implement, in order to allow developers to program exactly how tokens function within the Ethereum ecosystem. Thanks to these rules, it allows for greater predictability in how tokens are moved from one address to another.

¹ For a complete introduction to Ethereum, please visit the Binance Academy.

<https://www.binance.vision/blockchain/what-is-ethereum>

² <https://github.com/ethereum/wiki/wiki/White-Paper>

Before the introduction and the actual adoption of this set of rules by all Ethereum developers, tokens could not be transferred with full predictability, leading to interoperability issues.

[ERC-223](#)

ERC-223, built on top of ERC-20, refers to additional standard functions token-related contracts can implement to prevent accidental sending of tokens to contract addresses. It also makes token transactions behave like Ether transactions. Specifically, it extends the ERC-20 standard to address issues that could lead to some funds being lost forever.

[ERC-777](#)

Similarly to the ERC-223 standard discussed previously, ERC-721 refers to additional standard features a token contract can implement to prevent potential loss of tokens.

These features allow pre-checks to verify that a contract has all the required functions to support tokens received via the use of certain functions. In a nutshell³, operators can send tokens on behalf of another address, whether it is a contract or a regular account while token-holders obtain greater control over their tokens. Furthermore, it *de facto* supports the black-list of specific addresses as operators can now be white-listed.

[ERC-721](#)

ERC-721 refers to an open standard that describes how to construct and deploy [non-fungible](#) or unique tokens on the Ethereum blockchain. While most tokens are fungible, ERC-721 tokens are all unique. One of the first popular use-cases of non-fungible tokens were [CryptoKitties](#) in late 2017⁴.

[According to Binance Academy](#), the creation of blockchain-based non-fungible tokens allows for:

- **Physical property** (e.g., houses, artwork, and vehicles)
- **Virtual Collectibles** (e.g., CryptoKitties or collectible cards)
- **Assets with a negative value** (e.g., loans)

All ERC-721 tokens must also comply with the [ERC-165 interface](#), which standardizes the method by which smart contracts interact with tokens that follow other standards (i.e., non ERC-20).

³ <https://eips.ethereum.org/EIPS/eip-777>

⁴ <https://www.bbc.com/news/technology-42237162>

[ERC-998](#)

ERC-998 is a standard extension for any non-fungible token to own another non-fungible ERC-721 or fungible ERC-20 tokens. Specifically, transferring the ownership of the token composition means transferring the entire hierarchy of items.

This standard extension for any non-fungible token could allow for new creations, such as:

- Meta NFTs (e.g., NFTs as part of other NFTs)
- Bundles and tailor-made portfolios for digital asset managers (e.g., sets in the Set Protocol⁵)

[ERC-1155](#) (Enjin)

ERC-1155 refers to a standard interface for contracts that manage multiple token types. It allows a single contract to include any combination of fungible tokens, non-fungible tokens, or other configurations such as partially fungible tokens.

This standard interface allows the use of a token ID to differentiate items part of the same contract. Each token ID represents a new configurable token type, which can have its unique metadata, supply, and other specific attributes. Specifically, the design from this interface offers several benefits, such as the ability to transfer multiple token types at once (e.g., game items).

1.2 Security tokens

ERC-20 was the main type of token standard used in the “2017 ICO-mania”⁶, which saw many market participants taking advantage of the global regulatory uncertainty surrounding cryptocurrencies and ICOs in order to raise capital from investors.

Since then, the SEC and other financial authorities have been establishing various regulatory frameworks for security tokens. As a result, Ethereum developers and researchers have been working on different token standards to be compliant with current and forthcoming regulations across the globe.

Security tokens are designed to represent “*complete or fractional ownership interests in assets and/or entities*”. While **utility tokens have no limitation** on who can send or receive the token, **security tokens are subject to greater restrictions** based on identity, jurisdiction, and asset category.

⁵ https://www.setprotocol.com/pdf/set_protocol_whitepaper.pdf

⁶ <https://venturebeat.com/2018/01/18/2017-ico-mania-430-offerings-raised-4-6-billion-through-november/>

Specifically, these key features include the ability to differentiate token ownership, the right to freeze some tokens by a central keeper or the inclusion of document references (e.g., KYC documents).

Regarding security token standards on Ethereum, one of the most prominent standards is **ERC-1400**, in conjunction with **ERC-1410** for partially non-fungible tokens (PFT).

[ERC-1400/ERC-1410](#)

ERC-1400 represents **a library of standards** for security tokens on Ethereum⁷. This set of standards were conceptualized, designed and developed by Ethereum core developers, namely Adam Dossa, Pablo Ruiz, Stephane Gosselin, and Fabian Vogelsteller.

These standards are an umbrella of several other standards (briefly discussed below), which are all **backward compatible** with ERC-20 and ERC-777 interfaces.

[ERC-1410](#) : Partially Fungible Token Standard

ERC-1410 refers to both differentiated ownership and transparent restrictions. This interface supports an owner's tokens to be grouped into partitions, with each of them being represented by an identifying key and a balance.

Some of these partitions can be fungible while others are non-fungible. For instance, a non-fungible token partition may have specific covenants (e.g., vesting period specific to the security-holders).

[ERC-1594](#) : Core Security Token Standard

This standard provides an interface, which introduces checks for potential on-chain restriction, off-chain data injection for transfer restrictions, and issuance/redemption semantics.

Data injection refers to rules, which are defined off-chain, that can be applied and updated by the contract administrator, to define the set of applicable issuances and redemption mechanisms and potential transfer restrictions between addresses.

[ERC-1643](#) : Document and Legend Management

This standard allows documents to be associated with a smart contract and provides a standard interface for querying or modifying these contracts, as well as receiving updates (via events) to changes on these documents.

⁷ <https://thesecuritytokenstandard.org/>

[ERC-1644](#) : Controller Token Operation Standard

This standard allows “a token to transparently declare whether or not a controller can unilaterally transfer tokens between addresses”.

A controller refers to a program that manages or directs the flow of data between two addresses.

[ERC-884](#)

ERC-884 token is an ERC-20 compatible token, which was developed by David Sag to comply with the Delaware General Corporate Law.

Delaware corporations can use blockchain technologies to create a tradable ERC-20 token and maintain shares issued by a Delaware corporation.

[ERC-1404](#)

ERC-1404 is an addition to ERC-20-compatible tokens, which includes an additional function to allow restrictions on the transfer of tokens. This token standard was created by [TokenSoft](#), a technology provider for companies seeking to issue and manage digital securities on the blockchain while being compliant with regulations.

Specifically, it adds the following features on top of existing functions of ERC-20 compatible tokens:

- **Know Your Token Holders (KYTH):** maintenance of a whitelist of accredited investors who are allowed to own the tokens.
- **Enforcement of Complex Restrictions: country-specific limitations** or restrictions on either a maximum amount of token-holders or a maximum amount of shares per token-holder.
- **Support for Branded Standards:** integration with proprietary token standards such as ST20 or R Token, which are described.

[ERC-1450](#)

ERC-1450 (also referred to as LDGRToken) refers to an ERC-20 compatible token that complies with the new Securities Act Regulations: Regulation Crowdfunding, Regulation D, and Regulation A. This token standard was developed by [Start Engine](#).

1.3 Proprietary token standards (branded standards)

In this subsection, the main branded token standards for security tokens are discussed.

Branded token standards refer to **token standards, developed in-house, by companies** such as Polymath, Securitize or Harbor.

[ST-20](#) (Polymath)

Built by Polymath, ST-20 is an ERC-20 compliant token standard that incorporates the option to restrict the transfer of assets.

It is designed to be in line with ERC-1400 set of standards, which was also created by Polymath.

[DS Token](#) (Securitize)

The DS Token (Digital Security Token) was developed by Securitize, with part of its platform aiming to enable Digital Securities issuance on the blockchain, while satisfying compliance requirements.

The DS Token is an ERC-20 compliant token that implements specific features from the DS Protocol. DS Token addresses regulatory compliance through additional checks, via a Compliance Service, which verifies whether a transfer between two addresses should be accepted.

Furthermore, the DS Protocol adds methods for security issuances to either block wallets or freeze tokens in order to match regulatory requirements. In addition, it offers specific functions for security issuers to execute specific services, such as dividend payment directly to a list of investors (referred to by their wallet addresses)

[R-Token](#) (Harbor)

R-Token is developed by Harbor as part of its decentralized compliance protocol to standardize how crypto-securities are issued and traded on blockchain networks.

R-Token is an open-source standard, which defines a set of rules for crypto-securities to be transferred among addresses while being in compliance with existing rules. R-Tokens are permissioned ERC-20 tokens, on the Ethereum blockchain, which must clear operations with an on-chain Regulator Service before any approval.

Specifically, the Regulator Service introduces specific rules that are designed specifically for each type of security to comply with relevant regulations, KYC policies and AML requirements and tax laws.

[S3 \(Open Finance\)](#)

Part of their proprietary ecosystem (i.e., OpenFinance Network), S3 is a library built by many modular contracts, which aim at offering a library with the needs of specific regulation. Specifically, this library addresses restriction about transfer compliance, AML/KYC, investor accreditation, and malicious actor checks (through blacklisting).

This library also covers many offerings of registered & restricted securities, such as “Regulation D, Regulation S, Regulation A+, and Regulation CF”. These offerings allow issuers to easily and compliantly create a security token.

[Atomic-DSS \(Atomic Capital\)](#)

Atomic DSS tokens are developed by Atomic Capital, offering an extension of the ERC-20 token standard for “digital security issuance and automated regulatory compliance”.

Atomic DSS (Digital Security Standard) tokens are a permissioned, ERC-20 tokens, on the Ethereum blockchain, designed to be digital securities, which enforce transfer restrictions based on a “Regulator Service” contract that can be upgraded over time to match changes in the regulatory and compliance environment. Specifically, this standard is designed to enforce “KYC and AML requirements, accredited investor checks, trading lock-up periods, tax laws, and other contractual agreements”.

1.4 Ethereum limitations

Despite the dominance of Ethereum as the go-to blockchain for the creation of new tokens, other programmable blockchains have been developed and have recently gained traction.

As of today, one of the main issues with Ethereum relates to the **scalability of its base layer**, which sometimes results in **high gas fees and slow transactions**. While this can be interpreted as a sign of the popularity of the blockchain, newer blockchains have begun to compete in different segments.

Despite the full range of promises from ICO projects in 2017 to build payment systems, gaming, and other utility systems on the base Ethereum layer, Ethereum’s most significant use case during this period remained as the actual raising of funds for these projects. A vast majority of those “promises” have since never been realized, simply due to the scalability issues of Ethereum.

With Ethereum being seen as a generalist blockchain where almost anything could be created, albeit with relatively slow and inefficient transactions, other blockchains sought to build out more specialized and niche areas, for example TRON with more efficient distributed storage solutions and transactions leading to better support for Gaming, and Binance Chain with transaction speed and order matching for transfers and trading.

Layer-2 scaling solutions (e.g., [Celer Network](#) or [Matic](#)) or the [Plasma upgrade of Ethereum](#) may potentially turn out to be solutions for these scalability issues, though new blockchains have been increasingly aggressive in competing for market share in on-chain tokenization.

The next section will analyze token standards on many blockchains before focusing further on a select set of blockchains.

2. Multiple tokens on various blockchains

This section discusses the core characteristics of programmable blockchains that support native tokens running on-chain. It also compares the activity, the number of developers, and tokens between chains.

2.1 An overview of the blockchain token standards

This first subsection introduces all the main blockchains (both through the deployment of smart contracts and natively) and extra layers (e.g., Omni Layer, Simple Ledger Protocol) which support the creation of tokens.

A native standard refers to a blockchain that natively supports the creation of specific tokens on its chain, Binance Chain being a prime example with the BEP-2 standard. For example - from a code perspective - tokens running on the Binance Chain as BNB (the Binance Chain Native Token, behaves in a similar manner as all other tokens issued under the BEP-2 standard, which all run on the Binance Chain.

On the other hand, a constructed standard refers to a blockchain whose tokens are supported as part of a smart-contract function with Ethereum and Tron being two familiar examples. For instance, ERC-20 tokens are not recognized at the blockchain level and run on a Virtual Machine (the Ethereum Virtual Machine).

Table 1 - Blockchains and main layers supporting tokens

Blockchain	Token support	Standard names	Deployment / Language(s) supported	Token Status **
Ethereum	Constructed	ERC-20, ERC-223, ERC-721, ERC-777, ERC-1400, ERC-1410...	EVM / Solidity and Vyper	Active
Simple Ledger Protocol - Bitcoin Cash based on UTXO	Native (2nd layer)	SLP - Fungible Simple NFT SLP - NFT1 (group of NFTs)	Native (Script in OP_RETURN) / JavaScript	Active
Liquid Sidechain - Bitcoin based on Lightning Network	Native (2nd layer on-chain)	L-tokens	Native / Python, C#, Golang, Ruby, Java, Perl	Active
Binance Chain	Native	BEP-2 - Fungible BEP-7 - Non-fungible	Native / Binance Chain CLI , JavaScript, Golang, etc.	Active
EOS	Native and constructed	Tokens EOS-21 Tokens - ERC-20 equivalent	Native / CLEOS - Tokens WASM Virtual Machine / C++	Active
Omni Layer - Bitcoin based on wallet addresses	Native	Omni tokens	Native	Active (but likely to disappear)
Stellar	Native	Stellar tokens	Native / Java, JavaScript	Active
Tron	Native and Constructed	TRC-10 - Native fungible TRC-20 - Constructed Fungible	Native / CLI (TRC-10) Tron Virtual Machine / Solidity (TRC-20)	Active
Cardano	Constructed	-	EVM / Solidity Virtual Machine / Haskell	In development (presumably)
Tezos	Constructed	TZIP - Fungible	-	In development
NEO	Constructed	NEP-5 - Fungible NEP-11 - Non-fungible (WIP)	NeoVM / C#, VB.Net, F#, Java, Kotlin, Python, Golang, JavaScript	Active
Ethereum Classic	Constructed	ERC-20 - Fungible (other contracts can be implemented)	EVM / Solidity	Active but not used
NEM	Native	Mosaic tokens	Native / "in the NANO wallet"	Active but not used

Ontology	Constructed	OEP-4 - Fungible OEP-5 - Non-fungible OEP-506 - Security tokens	WASM Virtual Machine / C#, VB.Net, F#, Java, Kotlin, Python, C, C++	Active
QTUM	Constructed	QRC-20 - Fungible	EVM / Solidity	Active
WAVES	Native	Waves Token	Native	Active
VeChain	Constructed	VIP-180 - Fungible VIP-181 - Non-fungible	EVM / Solidity	Active but not used
IOST	Constructed	IRC-20 - Fungible IRC-21 - Self-deployed tokens IRC-721 - Non-fungible	Virtual Machine / JavaScript, Golang	Active
ICON	Constructed	IRC-2 - Fungible IRC-16 - Security tokens	SCORE / Python	Active but not used
Bytom	Constructed	BAP-2 - Fungible (WIP)	Virtual Machine / Equity Language	In development
STEEM	Constructed	SRC-20 - Fungible (Steem Engine side-chain)	STEEM Engine proprietary software	Active
Stratis	Constructed	SRC-20 - Fungible	Cirrus Sidechain / C#	Active but not used
Nebulas	Constructed	NRC-20 - Fungible NRC-721 - Non fungible	Nebulas Virtual Machine / JavaScript, TypeScript	Active but not used
NULS	Constructed	NRC-20 - Fungible NRC-721 - Non fungible	Nuls Virtual Machine / Java	Active but not used
Tomochain	Constructed	TRC-20 - Fungible TRC-21 - Fungible (can be used to pay for gas fees) TRC-721 - Non-fungible	EVM / Solidity and Vyper	Active
CyberMiles	Constructed	CRC-20 - Fungible CRC-721 - Non-fungible	Lity VM / Lity language EVM / Solidity	Active but not used
WaykiChain	Constructed	WRC-20 - Fungible	EVM / Solidity (WIP) Lua language	Active but not used
Wanchain	Constructed	WRC-20 - Fungible (Cross chain)	EVM / Solidity	Active but not used
MOAC	Constructed	ERC-20 - Fungible ERC-721 - Non-fungible	EVM / Solidity	Active but not used

Sources: Binance Research, GitHub

** Each blockchain that is EVM-compatible such as MOAC or Tomochain, can be used with Solidity, Vyper or other high-level languages.

*** Token status is defined as:

(1) Active - Ability to issue on the chain, with more than 3 tokens with value and/or a stablecoin issued on the chain

(2) Active but not used - Ability to issue on the chain, doesn't have 3 tokens with value or a stablecoin issued on the chain or any transactions

(3) In development - Development to create a first token standard

The next two subsections focus on both blockchains supporting decentralized applications (DApps) as classified by [DApp.Review](#), and the main blockchains by market cap. As a result, the rest of this report will only focus on the following nine blockchains: Ethereum (ETH), Binance Chain (BNB), Tron (TRX), EOS (EOS), IOST (IOST), Steemit (STEEM), Ontology (ONT), NEO (NEO) and Tomochain (TOMO).

2.2 Comparison of activity and developers between chains

Table 2 - Blockchains, constructed standards and main layers supporting on-chain tokens

Blockchain	Native asset	Genesis block	Count - Tokens ⁸	Market Cap ⁹	7-Day Avg. daily transactions ¹⁰	GitHub Commits ¹¹	Count - Accepted Improvement Proposals
Ethereum	ETH	July 30th 2015	200,000+	\$21.6 B	695.9 K	11,134	201
Binance Chain	BNB	April 23th 2019	130+	\$4.5 B	251.3 K	Private repository	10
EOS	EOS	May 25th 2018	4200+	\$3.4 B	3.5 M	11,597	6 (105 on-chain governance proposals)
Tron	TRX	May 31st 2018	2800+	\$1.2 B	3.5 M	12,077	11
NEO	NEO (GAS)	October 17th 2016	120+	\$713 M	14.6 K	724	10
Ontology	ONT (ONG)	June 30th 2018	15+	\$429 M	23.4 K	2,042	6
IOST	IOST	February 25th 2019	80+	\$107 M	563.5 K	8,547	N/A
Steemit	STEEM	March 24th 2019	490+	\$60 M	913.5 K	5,187	N/A

⁸ Data as of August 12th 2019.

⁹ Data as of August 12th 2019.

¹⁰ 7-day average between August 14 2019 and August 20 2019. Data from Coinmetrics and Blockchair.

¹¹ Data as of August 12th 2019. GitHub commits reflect modifications “committed” in a public repository. The main repository for each of these blockchains was selected (e.g. go-ethereum for Ethereum). The quality of each commit was not assessed and the absolute count of commits do not reflect the general quality of a project. Some blockchains keep their most of their development private, such as the Binance Chain.

TomoChain	TOMO	December 14th 2018	20+	\$31 M	25.8 K	10,348	N/A
-----------	------	--------------------	---------------------	--------	--------	------------------------	-----

Sources: *Binance Research, GitHub*

Without many surprises, Ethereum has the highest count of tokens running on its blockchain. However, the vast majority of these tokens are worthless owing to the issuance cost limited to the deployment of a smart contract.

On other competing blockchains, most of the tokens are also worthless, in spite of absolute numbers being lower. On the opposite, the Binance Chain has the second largest number of valuable tokens after Ethereum, with more than 50 positively-valued tokens¹². The third blockchain for utility tokens is NEO, with more than 30 tokens with a positive value¹³. The next subsection discusses issuance fees and on-chain transaction fees.

2.3 Comparison of issuance fees, transaction/gas fees, etc.

Table 3 - Issuance costs, transaction, and gas fees (estimated)

Blockchain	Issuance Cost	Median gas fee	Transfer cost between addresses
Ethereum	Gas fees (≈ 0.05 ETH ¹⁴)	1-2 gwei	Same as gas fee
Binance Chain	500 BNB	Same as transfer cost	0.000375 BNB
Tron	1024 TRX (for TRC-10)	0.0025 TRX	Same as gas fee
EOS	No gas fee or transaction fee but it needs resources. Two resources, CPU and Network, come from EOS staking while RAM needs to be purchased.		
IOST	Gas fees	1 gas ¹⁵ (works through a token pledge)	Same as gas fee.
Steemit	≈ 100 STEEM	0. STEEM relies on bandwidth.	
Ontology	Gas fees (≈ 0.02 ONG)	0.01 ONG	Same as gas fee
NEO	Gas fees (≈ 490 GAS)	0.00 ¹⁶	Same as gas fee

¹² Sources: *Binance DEX, Binance Chain explorer*

¹³ Sources: *CoinMarketCap, Switcho DEX trading pairs, NEO explorer*

¹⁴ Complex contracts cost more Ether to deploy in the main network. Please refer to for details: <https://bitfalls.com/2017/12/05/ethereum-gas-and-transaction-fees-explained/>

¹⁵ Pledging 1 IOST returns 100,000 gas immediately and 100,000 gas per day.

¹⁶ [For transaction > 1024 bytes, it s a function of the transaction size.](#)

TomoChain	Gas fees (≈ 10 TOMO)	0.00015 TOMO	Same as gas fee
-----------	-------------------------	--------------	-----------------

Source: *Binance Research*

Issuance costs are relatively **cheap** on most blockchains. As a general rule, the more complex the contract to deploy is, the more expensive it gets.

Furthermore, despite the Binance Chain being the most expensive to issue a token, it has the second largest amount of **tokens with positive value** (more than 50 tokens traded on the DEX) after Ethereum. NEO has more than 30 tokens with a positive value.

Transfer costs and median gas fees vary greatly across chains. Blockchains such as STEEM or EOS rely on a contribution model with the need for resources. On the other hand, blockchains like Ethereum, Tron or TomoChain rely on **gas fees** to transfer tokens from one address to another. The price of gas fees is subject to real-time market demand/supply dynamics.

Eventually, some blockchains like Binance Chain **have fixed transaction fees**. The next subsection will discuss the use of DApp on these blockchains.

2.4 The state of decentralized applications (DApps)

In this subsection, decentralized applications are analyzed using volume figures. It relies on data provided by [DApp Review](#), one of the leading websites for DApp data and metrics.



On DApp Review, the following categories are used: **Game** (e.g., CryptoKitties), **Social** (e.g., PeepETH), **Casino** (e.g., TronBet), **Finance** (e.g., Compound), **Exchange** (e.g., Switcheo), **Others** (e.g., Triip Protocol) and **High-Risk**.

However, the **High-Risk** category refers to [Ponzi schemes](#) and other dubious decentralized applications. As a result, volumes from this category were **omitted** from our analysis.

Table 4 - Median daily volumes on DApps (January 2019 - July 2019)

Blockchain	Median daily volume*	Game	Social	Casino	Finance	Exchange	Others
Ethereum	22,910 ETH (5 million USD)	0.75%	0.00%	31.33%	28.01%	39.75%	0.16%
Tron	414,061,257 TRX (9.3 million USD)	0.19%	0.00%	89.17%	0.00%	10.64%	0.00%
EOS **	3,533,870 EOS (15.6 million USD)	0.10%	0.00%	81.38%	0.29%	19.19%	0.09%

* Adjusted based on July 31st close price (UTC)

** Irregularities in volume with clear evidence of washtrading.

Ethereum's focus is larger than Tron and EOS with its volume split between exchange (39.75%), finance (28.01%) and casino (31.33%) categories. Conversely, Tron and EOS DApps are mostly related to gambling activities, with **80%** of their respective on-chain DApp volumes accounting for services such as online casino services, etc.

For Ethereum, the trend has moved toward greater Finance use-cases, as discussed in our [previous report about DeFi](#).

Table 5 - Median daily volumes on DApps (over July 2019)

Blockchain	Median daily volume *	Game	Social	Casino	Finance	Exchange	Others
Ethereum	35,619 ETH (7.8 million USD)	0.37%	0.00%	3.84%	73.01%	22.76%	0.02%
Binance Chain**	251,641 BNB 7.3 million USD	-	-	-	-	100%	-
Tron	407,980,000 TRX (9.1 million USD)	0.07%	0.00%	93.26%	0.00%	6.67%	0.00%
EOS ***	2,133,989 EOS (9.5 million USD)	0.05%	0.00%	80.52%	1.27%	17.86%	0.09%
IOST ***	1,220,161 IOST (12K USD)	2.96%	0.00%	53.65%	43.39%	0.00%	0.00%
Steemit	36,470 STEEM (9K USD)	42.39%	1.60%	3.22%	31.74%	3.52%	17.53%
Ontology ***	320,994 ONT (321K USD)	99.93%	0.00%	0.07%	0.00%	0.00%	0.00%
NEO	162,858 NEO (1.9 million USD)	0.40%	0.00%	0.00%	0.00%	99.60%	0.00%
TomoChain	360,076 TOMO (210K USD)	0.00%	0.00%	37.94%	0.00%	0.85%	61.21%

* Adjusted based on July 31st 2019 close price (UTC)
** Binance Chain includes the volume on Binance DEX.
*** Irregularities in volumes such as clear evidence of washtrading.
Sources: DApp Review, Binance DEX

All three largest blockchains started to diversify further: **finance use-cases for Ethereum**, while **Tron and EOS are mostly used for gambling**.

From table 5, the **most used blockchains are Ethereum, Binance Chain, EOS, Tron, and NEO** with median daily volumes **above USD 1 million**.

It is also worth noting that were reported evidence of **data manipulation** from some DApps, especially **on IOST and EOS**¹⁷. As a result, these numbers shall be taken with a grain of salt. In addition to this, other elements such as “DApp competitions” can potentially incentivize activities on a decentralized application. Indeed, some DApp owners have incentives to manipulate on-chain activities to earn prizes and financial rewards.

Despite having the choice to use stablecoins, many of these decentralized applications use their native tokens within their App. One potential reason is the ability to reach a broader user-base without stablecoins. In fact, the cumulative market capitalization of all stablecoins (across all blockchains) **remains below USD 5 billion**, which is nearly four times lower than the market capitalization of the Ethereum alone. It remains to be seen whether or when this trend would revert¹⁸.

3. Blockchain differentiation criteria

This section discusses how blockchains compete to attract developers and resources to build token economies on their respective blockchain.

Specifically, this section addresses these questions from the perspective of investors & token-holders, developers, and projects looking to develop a token.

3.1 Transactions and scalability

Transaction count and scalability are key factors to evaluate whether the time required to create, issue, and develop a token on a blockchain is worth it. As a general rule, the faster a blockchain is, the more appealing it appears. Yet, the speed of transactions can actually be interpreted from two opposite perspectives:

¹⁷<https://medium.com/@AnChain.AI/our-ai-detects-your-ai-revealing-the-secret-blockchain-dapp-world-of-bots-ed8884a07>

¹⁸ Future research reports will cover more in-depth the world of decentralized applications.

- A blockchain with fast processing and confirmation time is appealing, hence provides, at first, an impression of higher scalability.
- In certain cases, **longer transaction times** also indicate that a **blockchain is widely used**, as it was the case during some of Ethereum's past congestion periods, which then illustrates a general wider adoption of the blockchain.

Furthermore, the **amount of on-chain transactions also reflects the popularity and current adoption of a network**. From a developer's perspective, transactions indicate the health of a blockchain. Hence, a highly used blockchain can bring a larger audience to interact with the new token. Other indicators such as **count of individual addresses or new active addresses** also reflect the overall popularity of a blockchain.

From the perspective of third-party developers and projects looking to issue a token, they should look to issue new tokens or/and develop third-party on widely used blockchains, hence blockchains with a high amount of transactions.

3.2 Blockchain fees

Blockchain fees are a key element to attract users, projects, and developers. The lower on-chain and token issuance fees are, the higher are the incentives to interact within the on-chain ecosystem.

Blockchain fees relate to **on-chain fees**, such as the **cost to issue a new asset or transaction/gas fees**, along with the uncertainty regarding fees over time.

The cost to issue a token on a blockchain can also be interpreted with two conflicting lenses:

- If the cost to deploy a new smart contract is virtually null (e.g., Ethereum with only gas fees), a lot of worthless tokens can be created, which represent **spam on the network**.
- From the perspective of projects looking to issue their token, **a cost too prohibitive could potentially hinder the growth** of the on-chain token economy. As a result, blockchain would require some bootstrapping to incentivize new actors to create their utility tokens if the issuance cost is high.

In short, there is a **“sweet point”** between cheap to-issue tokens, which potentially lead to the creation of many worthless tokens (along with scams designed to fraud users e.g., similar names, etc.), and expensive-to-issue tokens, which may discourage projects to issue tokens on the chain.

Transaction and gas fees are also a key element to determine **each blockchain's attractiveness**. Some of the **gas fees on Ethereum** are often above USD 30 cents for a single transaction, which can **hinder greater network participation ratios**, i.e., prevent the on-boarding of new participants.

Furthermore, other factors, such as the need to pay gas fees in Ether (ETH), may potentially create issues for participants. For instance, a user on the Ethereum blockchain wanted to transact in USDC (or other stablecoins) could not send funds to another wallet without owning Ether. Some competing blockchains like TomoChain (TRC-21) or Binance Chain (natively) have allowed these fees to be paid in any valuable asset.

Yet, the **growing development of 2nd layer solutions**, such as Matic, should provide **greater scalability** for legacy blockchains (e.g., Ethereum) while **reducing on-chain fees**, i.e., allowing **users to transact at near-zero fees**.

Eventually, some EIPs (like [EIP-865](#)) have been drafted to offer similar features on Ethereum which could lead to gas fees, payable in different units than Ether.

3.3 DApps availability and use-cases

DApps availability and use-cases refer to the amount of existing and prospective applications that run on a blockchain. The greater the number of use-cases on a blockchain, the greater the value proposition is.

Some of the key elements to determine use-cases are:

- **Existence of at least one stablecoin** on the blockchain.
- **DApps variety**: the **diversity of decentralized applications** built on a blockchain.
- **Third-party developers** working on platforms (either centralized or decentralized) including assets, running on the blockchain. On the other hand, **interoperability features** across blockchains could potentially help to generate **positive network effects** that would lead to higher adoption of both chains.

Regardless, based on the findings from the previous section (2.4), **both the number of use-cases and the count of single users remain quite low**. As a result, first-mover advantages are still uncertain as no critical mass has yet been achieved. As interoperability and composability have increasingly been developed between dApps, we could see some compounding network effects and growth for the games with early users. Recent examples like Waterloo, a bridge between EOS and Ethereum by Kyber Network, illustrate the ongoing development of solutions for cross-chain interoperability.¹⁹

3.4 Security and blockchain development

Security and blockchain development must be considered from the perspective of projects and companies.

¹⁹

<https://blog.kyber.network/waterloo-a-decentralized-practical-bridge-between-eos-and-ethereum-c25b1698f010>

Several factors related to blockchain development are to-be-considered:

- **Amount of active developers: the total amount of active developers can be analyzed** with a breakdown between full-time, part-time. The **number of active full-time layer-1 developers** measures the health of a blockchain.
- **Commits:** the frequency, the quality, and the absolute amount of commits on GitHub often illustrate whether development is healthy.
- **Number of third-party developers** actively **working** on blockchain improvement solutions such as scalability and security patches.

On the other hand, security remains a key aspect for blockchains:

- **Vector attacks.** An **in-depth analysis of vector attacks** along with potential risks related to blockchain can reveal potential security outcomes
- **Existence of past attacks.** The outcome of past attacks may reflect some interesting elements. Whether an attack turned out to be successful or not, how the blockchain continues to fix, iterate, and develop in response to previous exploits represents **are critical points of evaluation for developers and stakeholders**
- **Public security policies.** For instance, bug bounty programs represent major key aspects regarding the importance of security from the perspective of the blockchain itself.

In general, blockchain development and security are **highly correlated**. A blockchain with high active development is more likely to follow security best practices and to react swiftly to patch any security bug. In a similar fashion, there is an incentive alignment between token-holders, third-party developers, and layer-1 developers to protect the security as an unsecured network would lead to a lower value in the long-term.

3.5 Easiness to build

Public blockchains compete to create the most inviting developer ecosystem to engage and retain global developer communities who will build applications on top of these chains, thereby facilitating additional transaction flows, network growth and coin / token use cases.

Specifically, projects that provide proper tooling, SDKs/APIs, and documentation along with open technical support that allows developers to quickly ramp up the learning curve and interact with the respective blockchains will have competitive advantages.

While there are many ways to build a thriving developer community and experience, several key elements stand out as necessary components:

- **Smart contract languages.** The introduction of new languages can potentially be daunting for new users as it introduces a new learning curve, which may potentially harm a blockchain adoption. Language could be a key component as EVM-compatible language Solidity has become, de facto, the standard in the industry. For instance, EVM-compatible blockchains potentially have a lower learning curve, to onboard developers on Ethereum, at the price of lower flexibility to address specific existing and future problems.
- **Detailed developer documentation** for running node software, using SDKs, issuing tokens, writing smart contracts, etc. (e.g., [Perlin Network](#))
- **Functioning test-nets** where developers can directly interact with the chain and test functionality (e.g., [Harmony Testnet](#) and [Block Explorer](#)). Some projects additionally have multiple versions of testnets with different features, leaving developers to not rely on only one testing playground. For example, Ethereum has four major ones: [Kovan](#), [Rinkeby](#), [Ropsten](#), and [Goerli](#). It is also easy to deploy an EVM-compatible private local network (e.g., Truffle/Ganache).
- **Technical focused social communities** where developers can directly interact with project teams and one another.
- **High quality, open-source code** that can be easily reviewed and audited. Once again, it relates to the number of developers and the overall development process from core developers along with third-party developers.

3.6 (De)centralization

(De)centralization refers to the overall degree of centralization a chain has from the validator perspective.

In general, blockchains whose validators, nodes, mining pools, and other contributors are centralized may induce **greater censorship risk**.

Censorship risk has different components for different stakeholders:

- **For on-chain trading on DEXs:** if traders are executing trades on-chain, there could be an **execution risk** with the trader being exposed either to any change in the price of the underlying assets featured in their strategies or be forbidden to trade.
- **For transfers:** there is potential **settlement risk** as there may be censorship preventing the transfer between two wallets. Censorship at the validator level may prevent transactions from ever actually being processed, rather than limitations or restrictions levied at the token level.

Moving along the spectrum of decentralization is not easy, as chains that begin at a certain spot between fully centralized and fully decentralized will attract a certain initial audience - shifting away from their initial degree of centralization may alienize or turn away new users that the chain otherwise would have attracted.

On the other hand, the parameter of decentralization for blockchains cannot be considered solely in a vacuum; generally, a greater degree of centralization may potentially increase the speed to achieve on-chain consensus, and hence, it reflects in faster finality (when a transaction can be safely considered as “done”) and a potentially **higher “effective TPS”**.

4. Conclusion

Ethereum remains the **“go-to” blockchain** to issue tokens. Specifically, it has the largest amount of issued tokens, despite the existence of a lot of “on-chain spam” (worthless tokens). Ethereum also has the largest pool of developers, on layer-1 core development, layer-2 solutions, along with many active developers working on third-party applications (DApps). Furthermore, many new token standards are also actively developed on Ethereum, such as security token standards (e.g., ERC-1400) and other proposals like ERC-998 or ERC-1155.

However, many competing blockchains also offer a compelling value proposition to issue tokens, which may challenge the existing leadership of Ethereum for on-chain tokenization. Some of these blockchains allow the creation of tokens natively (e.g., Binance Chain) while others rely on the deployment of smart-contracts, following Ethereum’s practices.

Regardless, the number of actual use-cases and single users remain fairly low across the entire blockchain space and in spite of the current Ethereum’s dominance, it is too early to rule out any potential competing blockchains which allow the issuance of tokens.

In general, there are key elements that projects, coin-stakeholders, and developers should consider to decide what blockchain to consider and invest in the future:

- **DApp availability or growing use-cases:** the more applications a blockchain has, the greater the value proposition is.
- **Transactions and pace:** the faster a blockchain is, the more appealing it generally becomes. On-chain transaction frequency also reflects the popularity of a network.
- **Blockchain fees:** the lower on-chain and token issuance fees are, the higher the incentives to interact within the on-chain ecosystem.
- **Easiness to build:** elements like a test-net, EVM-compatibility & number of smart-contract languages supported play a key factor in attracting users.
- **Security and blockchain development:** general activity on the blockchain, reflected by elements such as the number of improvement proposals and developers on layer 1, is generally an accurate measure of the health of a blockchain. In addition, blockchain security remains essential with vector attacks and past attacks being focal points.
- **(De)centralization:** the degree of centralization impacts the value proposition of a chain, particularly from the perspective of potential token-stakeholders.

Eventually, a wide variety of programmable blockchains will likely coexist **if interoperability solutions across chains develop** and prove to be both secure and usable. The rise of blockchain agnosticism, in conjunction with the convergence of best practices and programming languages, could further lead to a range of blockchains with different use-cases and communities at different scales.

References

- 101Blockchains. ERC Standards (2018). <https://101blockchains.com/erc-standards/>
- 101Blockchains. ERC-20 vs ERC-223 vs ERC-777 (2018). <https://101blockchains.com/erc20-vs-erc223-vs-erc777/>
- Antonopoulos, Andreas. Mastering Ethereum: Building Smart Contracts and DApps (2018)
- Atomic Capital (Alexander S. Blum). State of the Standards: A Technical Review of Current Digital Securities Standards (2018). <https://medium.com/atomic-capital/state-of-the-standards-a-technical-review-of-current-digital-securities-standards-3bc79f5aaf73>
- Electric Capital. Developer Report H1 2019 (2019). <https://medium.com/@ElectricCapital/electric-capital-developer-report-h1-2019-7d836d68fecb>
- Kwok Angela, Sun Weimin, Wu Xun (Brian). Security Tokens and Stablecoins Quick Start Guide (2019).
- Lockyer, Matt. The Tokens of Babel - Token Standards on the Blockchain (2018). <https://hackernoon.com/the-tokens-of-babel-token-standards-on-the-blockchain-745c87c640c1>
- Marks, Howard. The Security Token Standard (2018). <https://hackernoon.com/the-security-token-standard-bc07409947ae>

This report has been prepared solely for informative purposes and should not be the basis for making investment decisions or be construed as a recommendation to engage in investment transactions or be taken to suggest an investment strategy in respect of any financial instruments or the issuers thereof. This report released by Binance Research is not related to the provision of advisory services regarding investment, tax, legal, financial, accounting, consulting or any other related services and are not recommendations to buy, sell, or hold any asset. The information contained in this report is based on sources considered to be reliable, but not guaranteed, to be accurate or complete. Any opinions or estimates expressed herein reflect a judgment made as of this date and are subject to change without notice. Binance Research will not be liable whatsoever for any direct or consequential loss arising from the use of this publication/communication or its contents.

Binance Research provides in-depth analysis and data-driven insights of digital assets by generating unbiased, institutional-grade research reports for investors in the crypto space.

