

KAIZEN FINANCE: An Autonomous Protocol and Platform for DeFi Token Launches & Collateralized Token Trading

Technical Whitepaper

Abstract

A novel, secure, and highly flexible autonomous token launch platform for the issuance and lifecycle management of fungible tokens, KAIZEN.FINANCE is introduced. Beyond its ability to reliably swap and stake tokens, the KAIZEN PROTOCOL – a field proven dApp and smart contract authoring program is capable of supporting multi-chain launches and cross-chain commerce including Ethereum, Binance, Solana networks, and others. One of its unique features, the collateralized transaction proxy token, or kTx token, allows enterprises to issue pre-TGE tokens to private buyers and presale investors without any risk of rug pulls by early investor’s prematurely liquidating their position upon listing. Customized for each tranche with unique pricing and separate vesting schedules made in accordance with an offering’s tokenomic model, the kTx token is redeemable for its underlying collateral (e.g. an enterprise, DAO, or project token) only as the collateral unlocks. In this manner the kTx token acts as proof-of-purchase allowing an investor to personally claim their unlocked assets by logging into their secure wallet and swapping kTx for its collateral without troubling the issuer to manage unlocked token distribution or pushing tokens to wallets (and the risks therefrom). Investors can thereby access their unlocked tokens on as need basis either selling them in an exchange or AMM, or staking them in DeFi pool for interest income. The KAIZEN.FINANCE tool suite includes the KAIZEN UI/UX hierarchical menu interface for intuitive trading; an integrated adaptive dApp – the KAIZEN AI ORACLE for repelling arbitrage exploits; a multiple crypto pool able to transact payments in various crypto denominations (WETH, USDC, USDF, DAI, and BUSD); a blockchain based cybersecure wallet CYBERWALLET, and a multifactor identity validation interface HYPERID based on patented technology of world’s most secure communication technology – the HYPERSPHERE. Another novel feature, the kDEX collateralized token trading exchange makes it possible to hypothecate kTx tokens before their collateral vests, either by selling (swapping), lending (staking), or for collateralized borrowing. Together the tool suite of KAIZEN.FINANCE delivers unparalleled performance, security, and flexibility to token issuers and to investors alike.

Table of Contents

A brief history of the cryptocosm	p. 2
What are DeFi tokens?	p. 21
Transacting DeFi tokens	p. 32
Introducing KAIZEN.FINANCE	p. 45
KAIZEN token launch	p. 48
KAIZEN collateralized transaction (kTx) tokens	p. 51
KAIZEN.FINANCE investor user experience (UX)	p. 59
kDEX collateralized token trading	p. 65
KAIZEN.FINANCE token launch lifecycle management	p. 66
Literature cited	p. 67

A Brief History of the Cryptocosm

As excitement toward all things cryptocurrency, decentralized finance, and digital token trading grows and global crypto-exchange markets flourish, an entirely new and transformative ecosphere of business and commerce – the crypto-economy ^{[1][2]} has emerged. The enabling technology driving this cryptoeconomic revolution is the blockchain ^{[3][4]}.

Blockchain transactions today are incredibly diverse. Applications include commerce, business, investing, enterprises, banking & financial services, fintech, corporate & project funding, venture capital, estates, real estate, insurance and trusts, education, manufacturing, construction, supply-chain management, agriculture, pharmaceuticals, clinical trials, research & development, security, IOT, infrastructure, smart highways, communication, IP, creative & artistic works, music, media, gaming, entertainment, and philanthropy.

Together with ancillary services used to support these activities, the blockchain forms a public transactional ecosphere called the cryptocosm ^[5]. Other names for the cryptocosm include the cryptoverse, the cryptosphere, or more broadly as decentralized finance (DeFi).

Enabled by the invention of the blockchain and bolstered by an unprecedented pace of innovation in features, infrastructure, and novel implementations, DeFi is rapidly garnering ubiquitous adoption across diverse markets, applications, and use cases. While the word blockchain describes a singular instance, the term-of-art “the blockchain” more broadly refers to the field of blockchains and its technology rather than to a specific implementation or chain. The nomenclature “cross-chain” refers to transactions spanning more than one blockchain.

In the context of this whitepaper, tokens transactions are inexorably interlinked to blockchains, specifically because digital tokens rely on smart contracts, and smart contracts need blockchains as their hosts. But to understand how smart contracts are used to create launch, lock/vest, and trade tokens we must first understand blockchain processing, decentralized transactions, and the role of cryptography in executing transactions via digital contracts and virtual currencies.

DeFi Token Evolution: In the crypto-economy, businesses operate differently. Decentralized autonomous organizations ^[6] (DAOs) enabled by the gig economy ^[7] easily out-perform lumbering bureaucratic mega-corporations. While starving SMBs cower to shareholder demands while repelling regulatory challenges (many sponsored by Big Tech), DAOs operate safely beyond the corrupting influences of hedge funds, bankers, lobbyists, and politicians.

The flexibility of nimble DeFi “projects” outpace corporate R&D as vanguards of technology innovation. Exchanges and DeFi pools invoke automation rather than costly human-capital to better serve their clients.

With direct access to enthusiastic advocates and avid believers through social media and investor channels on Twitter, Telegram, and social media, TGEs (token generation events)^[8] and IDOs are replacing IPOs as today's hottest investment opportunities. How did this pervasive change in business come to pass? In truth, it took a while. But as demonstrated throughout history, given sufficient time, technology always overcomes political might.

DeFi's pedigree starts with the birth of the Internet^[9] and the first GUI-based personal computers^[10] in the early 1980s, together forming the cloud on which the crypto-economy is based. The next major inflection point was the widespread adoption of the smartphone at the turn of the century, putting supercomputing capability and network connectivity into the hands of increasingly mobile consumers. Through broadband communication protocols based on orthogonal frequency division multiplexing^[11] (OFDM), high-speed multiple-access connectivity to 4G/LTE and 5G cellular networks along with free access to WiFi hotspots became pervasive.

But despite the mobility and high bandwidth offered by smartphones and the global connectivity of public Internet, world commerce at the time remained confidently under the control of central authorities – governments, banks, Big Tech, and other global oligarchs whose interest was (and still is) to ensure all financial transactions must pass through them (so they can take their cut).

That all changed in the housing and financial crisis of 2008 when the public's blind trust in governments and financial institutions was deeply shaken as personal wealth evaporated and banks became insolvent. In the economic panic that followed, people started to look for alternatives to safeguard their money and assets. Nature (and humans) abhor a vacuum...

With karmic irony, on January 3rd, 2009 in the depths of a deepening global financial crisis, the genesis block of bitcoin (block number 0) was mined and the world's first decentralized virtual currency^[12] born. Historically inscribed on its own blockchain, this unpretentious yet seminal event involved the mining of a mere fifty bitcoins by Satoshi Nakamoto, a pseudonym for the anonymous developer (or developers) of version 0.1 of the bitcoin software and author of the whitepaper entitled "*Bitcoin: A Peer-to-Peer Electronic Cash System.*" Preoccupied by loan defaults and mass layoffs, the event passed entirely unnoticed by the financial sector, wholly unaware of the disruptive long-term global commercial impact this unassuming event would foster.

Since that time, and despite governments, banks, financiers, and the powerfully rich attempting to destroy, stop, or otherwise contain it, bitcoin and its blockchain grew unceasingly. Like the gas lighting industry resisting an annoyingly unpredictable entrepreneur^[13] named Edison and his troublesome electric light, traditional banking and financial institutions viewed Bitcoin as an intolerable but inexplicably enigmatic risk that must be contained. They feared it because they couldn't comprehend it.

Despite protestations and self-denials, they gradually came to recognize that with no owner to regulate and no business to sue, traditional banking tricks to smother competition and stifle innovation couldn't stop decentralized finance. Like dinosaurs grazing lazily in the late Cretaceous unaware of impending change, Bitcoin had turned the unthinkable into the inevitable.

The next pivotal event was the birth of the Ethereum blockchain ^[14] in 2015, a second generation blockchain technology able to store, distribute, host, and execute decentralized applications (computer programs) called dApps. Together the Ethereum blockchain and its validators form the Ethereum Virtual Machine ^[15] or EVM, able to function as a singular decentralized computer spread globally across clouds and devices. Unlike its Bitcoin progenitor, the EVM is able to execute cryptographic transactions by invoking transactional dApps called *smart contracts* ^{[16][17]}.

The smart contract was first envisioned ^[18] in 1994 by American computer scientist, inventor, and retro-futurist Nick Szabo. Amazingly, four-years later Szabo conceptualized virtual currency (he called Bit Gold) and even described synthetic assets, ideas preceding bitcoin and its blockchain by over a decade. But despite Szabo's vision and Bitcoin's trailblazing efforts, it took the EVM to bring smart contracts to the world as a viable business tool. As such, Ethereum is considered the pioneer of 2nd generation blockchains – decentralized networks able to process smart contracts.

Since that time, the number of blockchains able to execute smart contracts has expanded greatly to include the Binance Smart Chain (BSC) ^[19], Huobi Eco Chain ^[20], Solana ^[21], Polkadot ^[22] and more... From work dating back to 2012, Verzun and Williams further proposed *fully-decentralized* smart-contract based dynamic directed-acyclic-graphs or DyDAGs ^[23] as the next evolutionary step in blockchain technology in a project called the HYPERSPHERE, a vision continuing apace today.

Irrespective of the blockchain virtual machine used to execute them, the advent of smart contracts gave birth to a new form of commerce called decentralized finance or DeFi and an associated new class of digital assets called tokens. These digital tokens, albeit cryptographically secured, are minted and traded through the execution of smart contracts, not by mining or solving hash-nonce puzzles to create new digital assets.

Because they are based on smart contracts rather than passive data stored on a blockchain, DeFi tokens represent a different class of digital assets ^[24] distinct from Bitcoin, Ether, and other *chain-native* cryptocurrencies including Litecoin (LTC); Cardano (ADA); Polkadot (DOT); Bitcoin Cash (BCH); and others. One principal difference of chain-native cryptocurrencies is that each have their own blockchains while tokens created by smart contracts can share the same blockchain and conceivably can operate cross-chain, i.e. across multiple blockchains. An especially important and unique property of tokens is they function in accordance with rules, i.e. logical conditions, of the smart contract that created them including controlling under what conditions they can be used.

Blockchain Processing in DeFi: In decentralized finance, the blockchain and its network replaces the role of commercial banks ^{[25][26][27]} as transactional fiduciaries. Blockchain features ^[28] include

- Permissionless transactional medium
- Trustworthy asset (or contract) validation mechanism
- Distributed ledger (DLT or blockchain) for record keeping
- Virtual machine for executing smart contracts

As a transactional medium, blockchains enable two parties to engage in commerce without knowing or meeting one another, thereby protecting the privacy and true identity of both. A second key function of a blockchain network is its ability to validate a digital asset, virtual currency, or a smart contract to prevent counterfeiting, double spending and contract fraud.

The third function of a blockchain network is its ability to store data, i.e. to function as a database immune to single point failures or attacks. DLTs, an acronym for *distributed ledger technology* ^{[29][30]} represent a entire class of linear database constructs. The terms DLT and blockchain are often used synonymously. While it is true that a blockchain is a DLT, the converse is not necessarily the case ^{[31][32]}. A more accurate description is that a DLT is a special type of database, and a blockchain is one particular implementation of a DLT.

Specifically, distributed ledger technology (DLT) comprises a distributed database managed by multiple participants spread across multiple nodes. Ostensibly, storing data on different nodes confers an added degree of data security because an attack on any one node (or its local data storage devices) is unable to access the full dataset contained in the diffuse database. But storing data distributed across multiple devices in multiple locations does not ensure a DLT is decentralized.

A truly decentralized database requires both the data content and the database access to be distributed. A DLT database with centralized control access to a distributed database, e.g. a CFO's desktop computer, is not decentralized because focused attacks on the control interface can access the entire content of the DLT regardless of where the contents are held.

The blockchain, a fully decentralized version of a DLT overcomes this vulnerability by removing the central control node entirely. Every block contains content (the payload of the blocks) plus a block header. Contents of the block header vary by blockchain ^{[33][34]}. but generally include a block-number; a time-stamp to ensure blocks are entered sequentially; and the block's unique digital identity needed for cryptographic validation or mining.

The payload of a block may contain passive information such as financial transactions; an identifying “address” of a digital asset or virtual currency; a linear data base or file; a creative work; or other database records.

The payload may also contain software in the form of a computer program or a decentralized application such as a smart contract (described in a subsequent section). The payload may contain both encrypted (ciphertext) and unencrypted (plaintext) data.

Aside from its payload, each block is time-stamped to provide an immutable chronological record of events. Another key element of every entry on a blockchain is a digital identity or signature, some means by which a third party can validate the block is valid and not fraudulent or corrupted.

Cryptography in DeFi: One way users control access to blockchain content and validate its authenticity is through the use of cryptography ^{[35][36][37]}. In broad terms, cryptography employs one of two stratagems,

- Encryption and decryption with cryptographic key exchange
- Hashing (encryption only), no key exchange required

In the reversible process of encryption and decryption, a payload’s content (e.g. transactions, data, files, images) is encrypted using a defined algorithm with a cryptographic key to produce an encrypted file called *ciphertext*. Using a decryption key (uniquely corresponding to the encryption key), another party can then recover original unencrypted (plaintext) content from the ciphertext.

In order to decrypt the file, however, the reader must first obtain a copy of the decryption key in a process called a key exchange. A key exchange safely performed over unsecured channels is referred to as a public key exchange ^{[38][39]}. Although the mechanisms of public key infrastructure (PKI) for exchanging cryptographic keys are many, one common algorithm is a *split key* exchange.

During a split key exchange, the intended recipient of encrypted content (not the sender) creates two keys – a public encryption key and a private decryption key. The cryptographic process is asymmetric ^[40] in that the decryption key can decrypt content encrypted with the encryption key, but the encryption key cannot be used for decryption.

The recipient then sends the encryption key to the sender node to encrypt their content accordingly and the resulting ciphertext is sent to the recipient. Since the encryption key is not useful for decrypting the ciphertext, both the public key and the encrypted file can be communicated publicly over open channels (hence it’s the name “public key”).

In most cases, however, sharing a cryptographic key with a user requires prior approval to access the contents of a block. Such applications are referred to as *permissioned* systems because access must be granted by an administrator. Ironically, not only is PKI used by blockchains for trusted validations, but now blockchains ^[41] are being symbiotically used to improve the integrity of PKIs. While PKI methods can be used to *protect* content in a blockchain, it is unnecessarily complex for validators simply to verify the authenticity of the chain has not been corrupted or the provenance of its blocks altered. Instead an alternative approach called a *permissionless* system ^{[42][43]} is preferred, one where miners don't need to request access to the blockchain and where sufficient information is publicly available to independently verify the veracity of the chain.

This is where the role of a cryptographic *hash* comes in. A hash ^[44] is a cryptographic process whereby a file of any size is encoded into a fixed length ciphertext based on the content of the file being hashed, not on a separate cryptographic key. Properties of a cryptographic hash include:

- Unidirectional – Content can be converted into a hash but the hash cannot be analyzed to reveal the content that created it (even when the hashing algorithm is known)
- Highly non-linear – Small input perturbations cause dramatic changes in hash output affecting over half the bits (sometimes called an avalanche effect)
- Random – The output of a hash cannot be predicted from its input file
- Fixed length – The hash output is of fixed length regardless of the size of the input.
- Deterministic – The same file input to a hash will always produce the same output
- Collision resistant – It is extraordinarily unlikely statistically that two different source files could produce the same hash output. For all practical purposes, every hash output is unique.

Compared to encryption, an important distinction of hashing is that the process is unidirectional – once a file is hashed the original content cannot be recovered or extracted from the hash output. Although at first glance the process of making a file no one can read seem useless, as it turns out hashing has become a critical component in verifying all blockchain transactions today.

The unique property of the hashing process is its deterministic yet highly non-linear behavior. Even the most minute perturbation in the source file produces vastly different unrecognizable hash code outputs. Statistically, this property means the only way the hash of one file will match the hash of another file is if the two files are identical. Metaphorically, a hash is a digital fingerprint ^[45] useful to reject imposters, exclude fraudulent blocks, and prevent content tampering.

A hash's fingerprint feature can be used to confirm a sequence of blocks in a chain are self-consistent and belong together. The hash of a hash of a hash, is called a hash chain ^{[46][47]}, for example:

$$H[H[H[H[x]]]] \equiv H^4[x]$$

Hash chains render sequential verification of a chain simple – the insertion of a fake block will produce a cascade of inconsistency errors easily detected as fraudulent by any third party validator.

Adapting hashing to the process of validating a blockchain for 3rd party inspection, blockchains employ nested hash chains. In nested hash chains, each block includes both a hash $H[x]$ of its predecessor blocks plus new content including transactions, data, images, etc.. For example if $x = n, n-1, n-2, n-3 \dots$ the nesting of a hash chain can be represented by the data structure:

$$\begin{aligned} \text{block } (n) &= \text{content } (n) + H[\text{block } (n-1)] \\ \text{block } (n-1) &= \text{content } (n-1) + H[\text{block } (n-2)] \\ \text{block } (n-2) &= \text{content } (n-2) + H[\text{block } (n-3)] \dots \end{aligned}$$

As depicted in **Figure 1** in a nested hash chain each block is hashed and embedded into the block header of the subsequent block on the chain.. During a block’s validation, if the hash recorded in any specific block does match the result of hashing the previous block, that block doesn’t belong there and the blockchain being evaluated is corrupted or fraudulent.

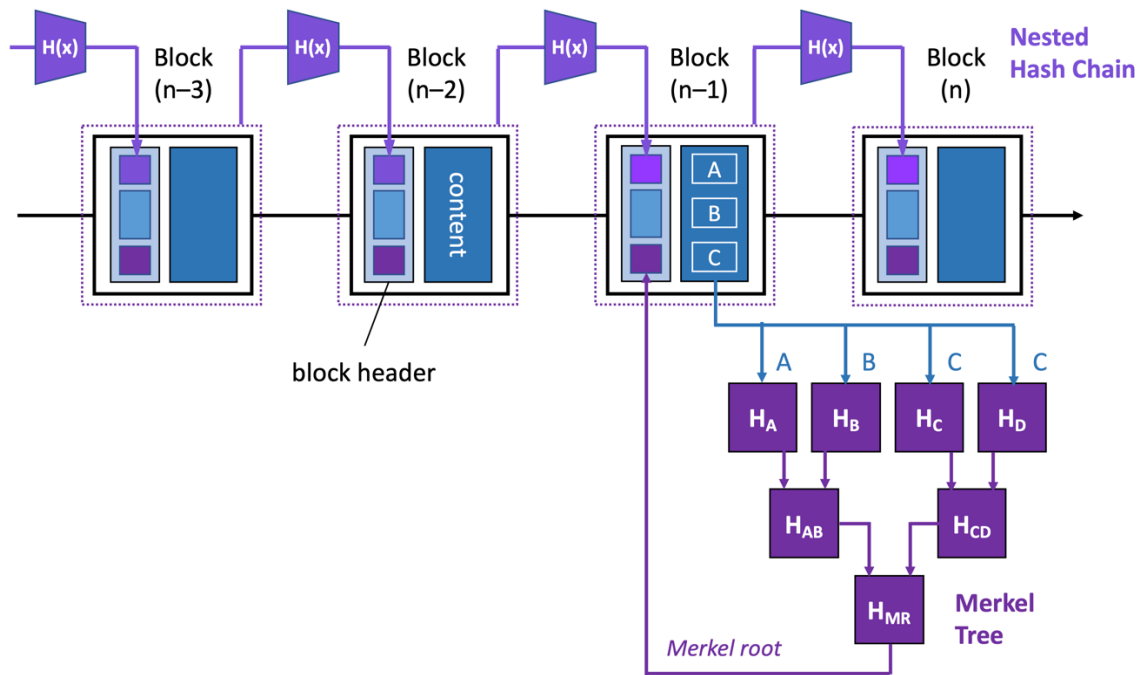


Fig. 1: Blockchain including a nested hash chain and Merkle trees

Aside from using hashing to validate the legitimate sequence of blocks, hashing can also be used to validate the content *within* blocks. Unlike sequenced blocks on the nested hash chain, hashed data of transactions contained *within* a block are not sequential but hierarchical, with each hash linked to its parent following a parent-child tree-like relation with the most populated hashed transactions comprising upper branches of “leaf” entries.

Also depicted in **Figure 1**, this hierarchical arrangement referred to as a Merkle tree ^{[48][49]} comprises binary paired hashes, that can be scaled for 2, 4, 8, 16, 2^{m-1} elements (excluding the root) for any tree having $m \geq 2$ tiers. Beneficial features of a Merkle tree include:

- The ability to determine the integrity and validity of data in a hierarchical database
- The ability to confirm the integrity of a specific tree branch even if portions of the tree are damaged or not yet available
- Lightweight, adding only minimal data overhead
- Allows branches to be verified without inspecting the entire tree
- Requires only small amounts of memory to perform validating proofs
- Results in minimal network traffic during a validation

Merkle trees are created by repeatedly hashing hash pairs until only one hash remains. In the block encoding process, transactions are hashed starting with individual instances (actually at the top of the hash tree), e.g. hashing blocks containing content A, content B, and content C. Because there is no content D to hash (i.e. D is a null entry) and binary hashing requires hash-pairs, to maintain binary leaf symmetry H_D is made a duplicate of the H_C hash.

$$H_A = H[\text{content A}]$$

$$H_B = H[\text{content B}]$$

$$H_C = H[\text{content C}]$$

$$H_D = H[\text{content C}]$$

The hashes are then paired, concatenated (or merged), and hashed again in repeated fashion until a single hash remains. Using $\|$ to denote concatenation, the hash pair H_{AB} and H_{CD} are given by

$$H_{AB} = H[H_A \| H_B]$$

$$H_{CD} = H[H_C \| H_D]$$

This action describes the hash processing executed one layer below the leaf layer. If more than two hashes remain forming an intermediate layer, the process is repeated until only two hash files remain. The final hash, referred to as the Merkle root H_{MR} , hashes these remaining two hashes:

$$H_{MR} = H[H_{AB} \| H_{CD}]$$

The Merkle root therefore represents a singular unique hash of all constituent hashes in the tree. By including this Merkle root in the header in each block of a blockchain, a quick check can be used to confirm block integrity throughout the tree. If the hash of the block doesn't match the Merkle root in the header, the content of the block has been tampered with, meaning the block may contain fraudulent transactions or a corrupted smart contract.

Importantly, hashing provides a means for a validation node or miner to confirm the veracity of a pending block *before* installing it onto the chain. But how can the juror nodes be trusted? Ensuring sincere validation is yet another role of cryptography in blockchain processing.

To avoid the risk of a corrupted authentication, validating a transaction and recording it is performed by a consensus mechanism ^[50], a process of voting by an anonymous jury-of-peers. Since the juror nodes in the blockchain's network are unaware of the identity of the parties involved in any transaction they are validating, there is no logical motivation for the jurors to cheat or vote prejudicially when evaluating the integrity of a new entry.

Although nodes can vote without doing a thorough job vetting hashes, jurors exhibiting a chronic history of disagreeing with other nodes are suspect and ultimately removed from jury consideration.

Another means to discourage improprieties among miners validating transactions is to employ a mechanism of *proof*. In one such concept (Proof of Work), each block header includes the hash of a random number called a *nonce*, a cryptography term meaning a number used once.

Borrowed from the field of trusted communication, by introducing randomness into transactions a nonce ^[51] is used as a form of authentication to ^[51] prevent replay attacks, thereby preventing outsiders from executing a hack by guessing the next action from the last.

In order to gain the opportunity to validate (mine) the block, the validator must first extract the nonce value (a random number having an varying number of leading zeroes) by solving the hash-nonce puzzle ^{[52][53][54]}. This brute force effort requires using energy and spending money thereby discouraging insincere effort or fraudulent results. In PoW implementations, these same hash-nonce puzzles are used to mine new cryptocurrencies ^{[55][56]}.

In addition to validating blocks, cryptography may also be used as credentials for confirming the authenticity of the content itself, e.g. art, music, tickets, etc. a feature especially important in trading non-fungible tokens. By employing cryptography in block confirmations and digital signatures, a blockchain is able to confer to users “trust in a trustless environment” ^{[57][58]}.

Blockchain Properties: Properties of a blockchain include the following key features ^{[28][59]} and characteristics:

- ***Trusted.*** Since every node in a blockchain's network contains a copy of the same digital ledger, there is no means by which to insert a false entry without the fraudulent block being detected and expelled from the network. In this manner, fraudulent transactions and corrupted blocks are never recorded on the blockchain.

- **Immutable.** Once an entry is time-stamped and recorded on the blockchain it cannot be removed or re-sequenced ^[60]. Indelibly recorded in the blockchain there are no means by which history and chronology of the blocks can be rewritten and corrupted *ex post facto*.
- **Decentralized.** By employing a network of anonymous nodes to realize the blockchain's autonomous virtual machine, no central computer or network operator is in control of the blockchain's activities. As such, there is no single device or node vulnerable to cyberattack; no operator to bribe or coerce; and no private information to abscond. Decentralized peer-to-peer networks ^[61] are able to avoid manipulation and repel distributed denial-of-service (DDoS) attacks because would-be hackers cannot identify which servers in the cloud are hosting the various nodes in the blockchain's virtual network.
- **Redundant.** The digital ledger forming the blockchain is shared across multiple nodes in the network eliminating any single point of failure or concentrated vulnerability. This parallel nodal structure provides data, network, and computing redundancy ^{[62][63]}.

Benefits of a redundant distributed virtual machine include timely and confident resolution of transactions and tasks, even when nodes or its communications network go offline. An early example of redundant computing with juror-based decision making was the computer used to control NASA's space shuttle.

A decentralized redundant network should, however, not be confused with distributed computing. In distributed computing, a task is divided into separable pieces and assigned to each node to do its fair share of the work. By paralleling the effort, the time to complete a task is reduced in proportion to the number of nodes contributing to the task. Likewise the total energy consumed to perform the compute job E_{job} is spread across the virtual machine having "n" nodes, where each node consumes an average energy of E_{job}/n per job.

In contrast, in decentralized computing, a single task or job is identically performed in parallel by "n" network nodes operating in unison as a redundant array of compute nodes. As such, the time required to perform a job is not reduced at all by the virtual machine's redundancy. In fact, in decentralized computing the slowest speed node in the network determines the transaction rate of the blockchain's virtual machine. Moreover the energy required by decentralized computing increases from E_{job} consumed by a centralized computer to $n \cdot E_{\text{job}}$ in its decentralized version, meaning decentralized computing is less energy efficient than a central computer, not more so.

- **Consensus Validated.** A key element of redundant systems is the means to arbitrate disagreement among its nodes, i.e. what to do when the nodes cannot come to a unanimous approval on whether a new transaction is valid. Decision making during conflict is

facilitated by voting in a jury-of-peers using a consensus mechanism ^[64] to tally the results of multiple nodes performing the same tasks. The requisite consensus criteria for passing or rejecting a pending transaction varies by a blockchain's implementation.

Consensus attacks ^{[65][66]} (such as a Sybil attack) used by hackers to corrupt voting can be repelled using randomized anonymous nodes and a variety of camouflage techniques to obscure juror nodes.

- **Proof.** A key concept in implementing reliable consensus determinations is the means by which to ensure all participation nodes are sincere, i.e. that their vote is verified to be meaningful. Verifying a node is a sincere juror is called “proof”, although a more insightful description of keeping a node honest should be called proof-of-sincerity. But how can a blockchain virtual machine ensure its nodes are acting sincerely on the best interests of the blockchain's integrity and its user community?

The first step toward ensuring transactional integrity is to protect the identity of the parties participating in a transaction, thereby preventing the validators from engaging in conspiratorial or prejudicial voting, or “gaming” the system, i.e. intentionally committing malfeasance.

Common means by which proof is achieved to thwart corrupted voting is by ensuring that a juror node either has something to lose if they vote insincerely or dishonestly, e.g. that (i) they must have either spent money and effort to participate in the validation process (“Proof of Work”) ^[67], (ii) the value of assets they themselves hold could be adversely affected by their actions (“Proof of Stake”) ^[68], or alternatively (iii) that the juror node holds certain credentials authenticating them as a participant whereby they only earn remuneration by consistently voting with the majority (without knowing *a priori* what the majority vote is).

First deployed as part of the Bitcoin blockchain and then later adopted by Ethereum, Proof of Work (PoW) intentionally wastes electricity as an necessary expense to mine new blocks. Spending discourages malfeasance or mischief by making such exploits cost the hacker money they can only recover so long that they are not disqualified for bad behavior.

Recently, PoW with its horrendous carbon footprint has come under intense global scrutiny as being energy wasteful, environmentally irresponsible, and ecologically unsustainable ^{[69][70]}. Alternative, greener methods such as Proof-of-Stake are now being adopted.

Migration to Proof-of-Stake (PoS) has however, been much slower than projected, in part because users aren't convinced that the *nothing-at-stake* problem can be avoided, a malicious deceit where a node misleads other nodes that it holds significant economic interest in the native cryptocurrency of a blockchain when in fact it doesn't. In PoS

consensus voting, ineligible nodes if undetected can commit fraudulent transactions (such as double spending) or intentionally subvert valid transactions.

Other options include credential-based consensus validations involving time-stamped “Proof of History” (PoH) ^[71] or the HYPERSPHERE’s “Proof of Performance” (PoP) ^[23], a decentralized consensus algorithm employing dynamic node governance and inimitable cryptographic hop-codes (a type of nested hash chain referred to as a transient blockchain).

These newer consensus algorithms, in addition to improving transactional speed, offer substantially smaller carbon footprints and improved resilience to hacking compared to their predecessors. In fact, the HYPERSPHERE hop code generates cryptographic proof of a node’s contributions as part of the network moving data and doing real work – energy that would have been used even if no consensus validation was involved. In other words, PoP validations can be executed as part of packet communication without consuming any additional energy.

But because of their intrinsic differences, these newer consensus mechanisms cannot be retrofitted into extant blockchains, and instead must be built into a blockchain’s architecture at its inception.

- **Security.** As described previously, cryptography provides security to the blockchain and its transactions in numerous ways ^{[72][73]} including authenticating the integrity and provenance of blocks on the chain and facilitating a means by which new blocks, once confirmed, can be appended onto the chain. Cryptography also enables verification of a block’s Merkle tree to ensure its contents have not been altered, thereby preventing double spending, fraud or tampering with a smart contract.

Using hash-nonce puzzles or security credentials issued via a blockchain virtual machine, cryptography is used to confirm the sincerity of juror nodes involved in approving transactions. Through private keys, cryptography protects the assets stored in decentralized wallets and transacted through smart contracts. Using digital signatures, the authenticity of an asset can also be verified, an especially important feature for trading collectables using non-fungible tokens (NFTs).

While blockchain transactions are protected cryptographically, network communications over the Internet are still exposed to surveillance, metadata analysis, phishing exploits, and malware. Transactions executed via Internet browsers, unsecured wallets, and badly designed protocols remain particularly vulnerable to attacks designed to abscond a user’s private encryption keys. To thwart such hacks, an added degree of blockchain protection can be achieved by altering network communication ^{[74][23]}, specifically by obfuscating or

encrypting packet data on OSI layers 3, 6, and/or 7 (i.e. network, presentation, and application layers respectively) – methods unique to the HYPERSPHERE.

Unlike the mining and validation of passive cryptocurrency vulnerable to DDoS and consensus attacks, the hard-coded instructions of a smart contract (once uploaded to the blockchain) are immutable. Thus limited, a cyberhacker can only interfere with smart contract execution by operation of the blockchain's virtual machine (BCVM) hosting it or by disrupting the system's state machine ^{[75][76]}, e.g. using time jacking, reentrancy, etc.

The risk of DeFi cyberattacks is amplified by badly written code, greatly increasing the attack surface of smart contracts exposed to hacking ^[77]. To prevent such risk exposure an exhaustive review by trained dApp and cybersecurity experts is required ^[78].

- **Transparency.** Decentralized unitary blockchains such as Bitcoin and Ethereum are public, providing open inspection of block entries and transactions. Public unitary blockchains facilitate tracing genealogical history of transactionally related addresses including wallets, digital assets, and smart contracts. Downloads of permissionless smart contracts also enable inspection of smart contract code by authors, users, and accredited 3rd party validation services. The downside of the transparency of public unitary blockchains is their susceptibility to various attacks including backtracing of high value wallets. Fortunately wallet addresses, although public do not disclose the wallet's owner's true identity.
- **Transaction Speed.** The speed of a blockchain depends on its validation mechanisms and on its temporal and hierarchical structure ^{[79][80]}. Proof of Work is, for example, significantly slower than Proof of Stake, which in turn is slower than Proof of History and Proof of Performance. Single public unitary blockchains are intrinsically slow and degrade in their transactional throughput the larger (i.e. longer) the chain becomes. Parallel chains like DyDAGs have the ability to deliver higher transaction rates because shorter chains with fewer blocks can be checked more quickly and with a greater degree of parallelism.

When executing a smart contract containing logic and arithmetic calculations, the operations per second (OPS) rate becomes a critical factor in smart contract execution time. In most implementations, blocks are mined, i.e. transacted, in fixed intervals called *block time* specific to the blockchain's network. For example, Ethereum previously ran with a block time of 15 seconds and in 2012 reduced its block time to 12 seconds ^[81]. Newer competing chains are faster, with block times as short as 2 seconds. By contrast bitcoin, a first generation blockchain takes 9 minutes to resolve a block.

To perform the same amount of work in less time requires higher speed computing. But since a blockchain doesn't control the hardware its miners or validators use, it cannot stipulate the hardware or a minimum performance level for nodes in its networks.

Since all transactions are time-stamped upon the block's recording on the chain, a means to measure and monitor a network's evolving performance already exists. A simple statistical analysis of recent chain transactions immediately reveals current processing times of the blockchain's community of nodes. Improvements in computing performance, i.e. miners buying faster processors, are thereby manifested statistically in transaction rates even though the particulars of computer architectures and GPU semiconductor technology nodes being used are unreported. As such, improvements in network performance can be adjusted to reflect ever improving performance for any specific BCVM.

Increasing a decentralized network's performance criteria does however carry risk ^[82]. As blockchain specifications are upgraded, the network unavoidably becomes controlled by an increasingly smaller population of more capable compute nodes that only wealthier miners can afford, thereby threatening node diversity and decentralization.

The other key factor deciding the arithmetic OPS rate is the block size. Rather than measuring blocks in terms of bytes, block size is measured by the maximum amount of gas allocated for a block. Since gas is a unit of measure for the computational effort required to execute certain operations ^[83], the gas limit measures a block's maximum computational effort, where gas required per operation is specified by gas consumption rate G, and where

$$\text{OPS} = \frac{\text{Gas Limit}}{G \cdot \text{Block Time}}$$

For example before its gas limit and OPS rate increase in April of 2021 ^[84], the EVM had a Gas Limit of 12.5M, a 12 second Block Time, and a consumption rate $G = 3$ per fixed integer operation ^[85], so that

$$\text{OPS} = \frac{\text{Gas Limit}}{G \cdot \text{Block Time}} = \frac{12,500,000}{3 \cdot 12 \text{ sec}} \approx 350\text{k ops/sec}$$

This instruction processing rate is quite slow by computing standards, being comparable to that of the IBM System/360 computer of 1965 ^[86]. By contrast, today's readily available graphics processors for gaming perform floating point (not fixed integer) calculations at rates of 30 TFLOPs, eight order-of-magnitude faster than EVM based dApps.

While it may appear obvious to improve throughput, the gas limit can be increased to accommodate larger blocks, the risk that a significant number of nodes will be unable to complete the job within the allocated block time grows exponentially with linear increases in block size. Incomplete block validations must be discarded and repeated, thereby increasing gas fees and delaying the completion of a smart contract's transaction.

- **Smart Contracts (dApps).** Rather than simply storing passive data, 2nd generation blockchains (and beyond) support decentralized applications (dApps) called smart contracts [16][18]. A smart contract comprises software-based *if-then-else* conditional logic able to react to changes in state variables specified in the contract including time & dates, trades, prices, token supplies, and prior transactions. Smart contracts can also perform fixed integer arithmetic operations, but cannot perform floating point calculations.

Blockchains enable smart contracts [17] in several ways including (i) providing a decentralized platform to store and distribute dApps to users, (ii) facilitating a live database recording transactions as updates to the newest most-current instance of the contract, (iii) providing a consensus mechanism to validate new contract instantiations, and (iv) operating as a blockchain virtual machine (BCVM) needed to execute smart contract instructions.

Inspired by the concept of decentralization first demonstrated by the file sharing program BitTorrent [87] combined with a deep appreciation in the frugal application of distributed processing using nodes of limited computing capability (i.e. performance comparable to ES-EVM [88]), the notion of an “unstoppable computer” lacking any dedicated hardware platform was born. This pioneering proof-of-concept in decentralized applications, the Ethereum Virtual Machine [15] or EVM, proved to be far more impactful than anyone could have foreseen.

In fact, the fundamental design of nearly all blockchain virtual machines remains essentially unchanged from its Ethereum blockchain progenitor. Architecturally, as depicted in **Figure 2**, the Ethereum virtual machine (EVM) operates as a state machine [89] comprising a lightweight decentralized application stored on the Ethereum blockchain able to perform varying degrees of computing applications with no dedicated host processor.

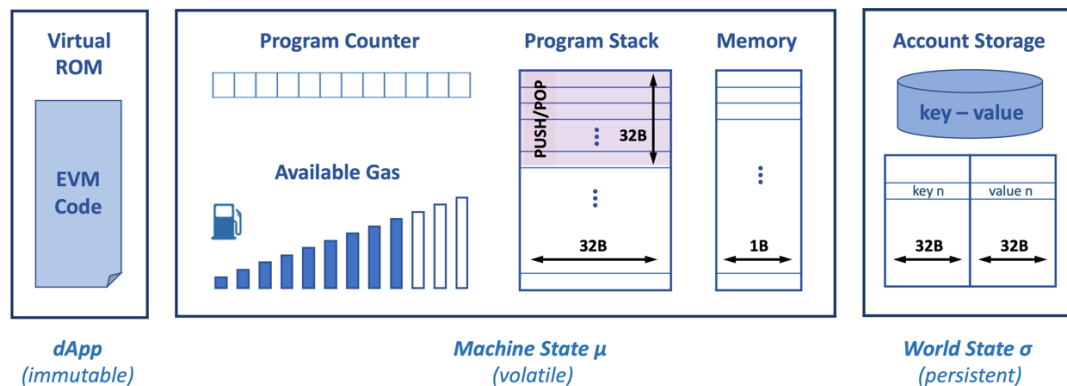


Fig. 2: Decentralized hosting of Ethereum Virtual Machine

As depicted, realization of an Ethereum virtual machine comprises three elements, namely (i) an immutable dApp comprising EVM code stored on the blockchain; (ii) a world state σ maintained in an updated yet persistent state as “account storage”; and (iii) a machine state μ comprising a volatile instance of the Ethereum virtual machine existing temporarily during transactional processing.

The computing kernel of EVM comprises software stored on the Ethereum blockchain as the EVM’s operating system. Using the blockchain as virtual read only memory (virtual ROM), this EVM OS cannot be revised or edited except by launching updates as entirely new instances on the blockchain. Even then, the prior version remains perpetually available. In this manner, the Ethereum blockchain functions as a trustworthy platform to distribute software, in this case the Ethereum Virtual Machine. To invoke the EVM, a smart contract written in the Java-script like language called Solidity make a function call to the EVM OS.

Upon its launch on the Ethereum network of nodes, the EVM OS collects variables data from its designated decentralized account storage representing a world state σ . Loading this data into the EVM allows the system to catch up with the events that occurred during its hibernation, i.e. to make it current. For example, if a number of tokens were sold via a smart contract as the last recorded transaction before going dormant, this available supply data is transferred from account storage into the EVM OS as a starting point for processing.

Data in the account storage is filed sequentially in key-value pairs comprising a cryptographic key 32-bytes wide identifying each state variable and a corresponding current value of the data, also 32B wide. Because this data is always loaded from account storage when booting up the EVM OS and rewritten at the commencement of a job, the world state μ data is described as “persistent”.

At the heart of EVM OS executable code is its machine state μ , a clocked state machine comprising a program counter, a gas gauge, a program stack, and scratch pad memory operating as virtual RAM. During EVM launch, operating instructions are loaded from virtual ROM while present state information is loaded from account storage. Transfers into the state machine occur in 32B data bursts and stored in either 1B wide memory or pushed onto the 32B wide program stack. The machine state μ is volatile, and does not survive beyond execution of a smart contract except via data written to storage as world state σ .

Like RPN-based programs and calculators, the program stack in machine state μ comprises a mix of data and ordered operating instructions, executed sequentially with each advance in the program counter. Completed operations are “popped” off the top of the program stack as execution continues until either the task is complete or gas runs out.

Careful programming of smart contracts for the EVM state machine is warranted to avoid transactions exceeding the maximum allowed block size. Block overflows produce a fault condition leading to permanent loss of the mining efforts for the block, failed program execution, incomplete transactions, and possibly indeterminate states. Looping, hold states, and floating point arithmetic operations are also forbidden as there is no way to determine *a priori* if the program step will compete or exceed the Gas Limit or Block Time.

Finally special care must be given to ensure smart contracts executed on the EVM are confirmed to be bug free. Operations writing data to an improper address or contracts stuck in a suspended state can lead to an unrecoverable loss of assets.

Operation of the EVM is only exemplary. The implementation of any blockchain virtual machine is *chain specific* employing different implementations, languages, protocols, and consensus mechanisms^[90]. Live chains include BSC Binance smart chain^[19], Huobi ECO chain^[20], Solana^[21], Polkadot^[22], Cardano Ouroboros^[91], HyperLedger^[92], Tezos^[93], Stellar^[94], EOS^[95], IOTA Tangle^[96], and others^{[97] [98] [99]}. Some of the foregoing blockchains actually comprise blockchain platforms for hosting *client customized* virtual machines and networks.

One such platform is HyperLedger^[92], an umbrella project for open-source blockchains of the Linux Foundation with major backing from IBM and SAP-Ariba. Frameworks developed atop HyperLedger include Hyperledger Fabric, Hyperledger Iroha, Hyperledger Sawtooth, and Hyperledger Besu plus a large array of developer tools.

Another platform for blockchain development is Polkadot^[22]. Created by Ethereum cofounder and developer of Solidity, Gavin Wood, Polkadot is intended as a development ecosphere for blockchains. An integral component of this system is Relay-Chain, an integration tool supporting network interoperability of parachains, parathreads, and sharding, methods allowing developers to create their own blockchains supporting customized governance and tokens while supporting rapid processing and high transactional throughputs.

One such smart contract platform developed on Polkadot is Moonbeam, an Ethereum compatible network allowing developers to deploy existing Solidity-based smart contracts and associated dApps to Moonbeam without major changes. By porting Solidity dApps to Moonbeam on Polkadot, purported beneficial performance improvements over Ethereum hosting include reduced costs and higher transactional throughput.

Smart contracts also play a special role in creating, distributing, and trading digital assets called tokens including both fungible and nonfungible tokens (NFTs) described here below.

- ***Crypto & DeFi Transactions.*** Blockchains support a variety of digital assets and their transactions. Examples include chain-native cryptocurrencies like BTC and ETH, fungible utility and security tokens such as USD \mathbb{F} for a variety of funding, investing and commercial purposes, and non-fungible tokens (NFTs) used for acquiring and trading collectibles with limited edition content. A smart contract enables unrelated parties to engage in shared activities such as trading (swapping), lending (staking), borrowing, and market making (liquidity providing). It can also be used to autonomously generate new token offerings.

Like a group of individuals pooling money to buy lottery tickets together as a collective, a smart contract can be used to anonymously pool investments from a group of unrelated investors and venture capitalists to jointly fund a project; invest in or fund an index or synthetic fund; or capitalize a DAO or corporation.

A smart contract enabling shared investment from unrelated parties is referred to as a decentralized finance pool, a DeFi pool, or a liquidity pool ^[100]. The DeFi pool is virtual and decentralized, not physically existing in any one server, device, website, storage device, or located in a specific country or domicile. A DeFi pool is therefore simply a smart contract shared by investors.

The platform or website hosting and granting access to the DeFi pool is not a participant in the pool. Existing at arms length from the pool's operation, the host platform is nothing but a link or user interface (UIUX) created to access the pool's smart contract. All terms and conditions for the pool are defined by its smart contract not its platform. As such, no platform or device has any control whatsoever over the terms and conditions of trading in the pool and among its participants. A DeFi pool therefore is peer-to-peer (P2P) commerce.

Tokens created by the smart contract are autonomously generated algorithmically by, for, and on behalf of the pool's participants. Except in the case of a corporate token offering, tokens issued by a DeFi pool in a token generation event (TGE) have no issuer. Since there is no issuer, a TGE does not meet the legal definition of a securities offering. This subtle but important distinction that token issuance without an issuer does not constitute an "offering" will remain a hotly contested legal matter for the foreseeable future.

Because there is no central authority able to regulate autonomously executed smart contracts, investors and token traders are advised to consider DeFi trading in the context of *caveat emptor* (buyer beware) and to take extra precautions to ensure that a smart contract is bug free and its DeFi pool represents a legitimate project or development. In the case of corporate offerings, an appropriate degree of due diligence is advised, the same as suggested in the case of buying private and public equities.

Cryptocosm Summarized: Given the foregoing in-depth discussion, several major high-level take-aways can be surmised.

1. The blockchain provides a decentralized ecosphere for engaging in peer-to-peer financial transactions and record keeping without the need for governing control by a central authority or financial institution. Blockchains are therefore unavoidably disruptive (possibly beneficially or detrimentally) to the banking and financial services sector.
2. Arranged into a linear sequence of finite-sized digital blocks, transactions on the blockchain are indelibly recorded using cryptography to create new entries and consensus mechanisms involving a jury-of-peers to confirm the veracity of existing block entries.
3. Entries on a blockchain may include passive components such as data, ledgers, records, and chain-native cryptocurrency (BTC, ETH, etc.) as well as smart contracts comprising decentralized application dApp software.
4. Decentralized applications and smart contracts can be accessed, downloaded, and executed (called) directly from the public blockchain with no risk of tampering to its source code. Third party expert validation and certification of a smart contract's quality and integrity is commercially available and recommended.
5. The execution of a dApp occurs not on a private website (at risk to hacking) but on a cloud based blockchain virtual machine (such as Ethereum's EVM) able to complete transactions through decentralized networks having no dedicated hardware or devices.
6. Because of risks of typographical errors, bugs, and poorly written logic, arithmetics, and function calls unavoidable in manual smart contract code writing, autonomous dApp software called *protocols* are employed to synthetically generate smart contracts following defined algorithms. Protocols too must be carefully checked and fully vetted.
7. One major class of dApps involves the use of smart contracts to create, issue, trade and hypothecate DeFi tokens. Tokens offer numerous beneficial features not available with passive chain-native cryptocurrency such as those created through crypto mining efforts.
8. DeFi tokens sharing a common smart contract are referred to as a DeFi pool. When sponsored by a token issuer such as a corporation, DeFi pools may be used to launch new tokens into the market. Alternatively DeFi pools funded by liquidity providers operate as autonomous marketplace for trading or lending of pre-existing virtual currencies and non-fungible tokens. A DeFi pool exemplifies peer-to-peer (P2P) commerce.

What Are DeFi Tokens?

Broadly speaking digital tokens are cryptographic tokens created by the execution of a smart contract on a public or private blockchain. Such tokens can be

- Issued as centralized virtual currency (digital money) by governments using a tightly regulated permissioned blockchain,
- Issued by a corporation or NGO (non-governmental organization) as a *non-financial* token for identity, access privilege, or gaming using a private permissioned blockchain,
- Non-fungible tokens (NFTs) issued by artists and gamers as collectables and as virtual assets in the metaverse,
- Issued as fungible DeFi tokens comprising a decentralized tradable asset using a public permissionless blockchain.

The first two examples, government issued digital money and private access tokens are not decentralized and therefore beyond the purview of this whitepaper. The third bullet, non-fungible tokens comprise their own DeFi token class. Unlike fungible tokens which comprise a population of indistinguishable digital tokens of equal value useful in commerce, non-fungible tokens exhibit properties of uniqueness, identity, and ownership (royalty rights).

Trading of NFTs in various blockchain specific marketplaces and communities requires extensive cross-chain transactions whereby gas fees, not utility, become a significant factor. A description of NFTs, their challenges and solutions to minimize gas costs is discussed in detail in an associated technical whitepaper entitled “*Scotty Beam – World’s First Cross-chain Decentralized NFT Teleport,*”^[101] and will not be discussed further in this work.

The fourth bullet, decentralized fungible digital assets known as *DeFi tokens* represents an enormously broad and expanding asset class. Deployed on public permissionless blockchains, DeFi tokens are generated not by puzzle solving, but by executing smart contracts – dApps hosted on one or more blockchain virtual machines (BCVMs). One distinguishing difference between DeFi tokens and cryptocurrency is their “total supply”. Unlike cryptocurrency where new coins can be mined at any time (even at a loss), the total supply of a token at its launch is fixed in quantity and cannot be adjusted afterwards. DeFi tokens come in a variety of forms including:

- Cryptographic wraps of fungible tradable cryptocurrency (such as WETH)
- Stablecoins comprising fungible tradable tokens having value pegged to real world assets such a fiat money, gold, etc.
- Fungible tradable tokens issued by a corporation or entity
- Fungible tradable tokens issued autonomously by and for a DeFi pool

Crypto Cash Equivalents: The first two bullet items, i.e. cryptocurrency wraps and stablecoins, operate as crypto cash equivalents in the cryptocosm. These assets have a defined value at the time of a trade and therefore can be used in swapping a publicly tradable asset of defined value for a private or publicly traded token of speculative future value. The worth of crypto cash equivalents at the time of a DeFi token trade are well defined by high-volume global markets. Some tokens are further collateralized by non-cryptographic assets such as gold or fiat currency to combat volatility. Such tokens, referred to as *stablecoins*, carry value pegged to its fungible counterpart.

For example, Tether is a stable coin pegged to the United States dollar whereby $value\{1\ USD\} \approx value\{1\ \$USD\}$. Stablecoins bridge the token and cryptocurrency worlds as they may be traded on digital currency exchanges and also be used in the autonomous execution of a smart contract. In the commercial press, stablecoins are often incorrectly referred to as cryptocurrency (akin to BTC and ETH) because they perform the functions of storing value and transacting commerce. Stablecoins are not however chain-native cryptocurrencies, they are tokens, which is the reason why they can be used in smart contracts without first wrapping them (when BTC and ETH cannot).

For example, the US dollar based stablecoins USD \mathbb{F} , USDC, DAI and BUSD are pegged to the United State dollar at fixed 1:1 conversion ratio (within some small slippage range). Other stablecoins are now being issued pegged to the Euro, the British pound, the Japanese Yen, and the Singapore dollar. In the case of a highly-traded volatile currency like ETH, its cash equivalent value varies constantly. As such, during any DeFi transaction, the Ether must be valued at its “spot” trade price – its market value frozen at a specific moment or window in time. To avoid any misunderstandings, a particular public digital currency exchange, e.g. Binance, is specified as the exchange on which reference trading price is based.

Chain-native cryptocurrency like ETH suffers several disadvantages compared to tokens. Firstly, they are locked to their own blockchain and are not interoperable across chains. Secondly, they are not ERC-20 compatible meaning they cannot be used to transact trades by smart contract. Thirdly, as a liquid, tradable, and fungible cryptocurrency they cannot be locked or held as collateral except by transferring them to a custodial wallet (wallets having private keys controlled by another party).

The solution to this quandary is tokenization – converting chain-native cryptocurrency into tokens. The token conversion process, turning fungible assets into smart-contract compatible tokens, is performed prior to engaging in any dApp based commerce, and therefore does not affect realtime transactional throughput. The process of tokenizing cryptocurrency is called “wrapping” [102]. Token wrapping involves locking an asset into a smart contract and generating a specified number of corresponding tokens. The aggregate worth of these tokens at inception equals the value of the underlying asset securing the tokens. Token wraps are thereby mirrors of the underlying asset used to create them, whose value depends on the asset collateralizing the token and on the quality of the smart contract used to create them. Token wraps are therefore also called *tokenized cryptocurrency*.

Any crypto asset can be wrapped. Token wraps of cryptocurrency are frequently identified by attaching the prefix “W” to the cryptocurrency’s symbol name. For example WETH is any wrap of Ether, WBTC is Bitcoin wrap, and so on. For convenience, token wraps are generally issued on a 1:1 basis with their underlying asset and maintain a constant price ratio despite market fluctuations in the asset base’s value. In other words, if the price of ETH rises, so too does WETH.

For example, the value of Bitcoin wrap is equal to the real-time value of Bitcoin, $price\{1\ WBTC\} \approx price\{1\ BTC\}$. Similarly on the Ethereum network, $price\{1\ WETH\} \approx price\{1\ ETH\}$. Because the price ratios of tokenized cryptocurrencies are fixed relative to their underlying assets and since it is generally understood that a smart contract can only transact tokens, it is commonplace to exclude the W prefix when referring to wrapped cryptocurrencies, and to casually refer to cryptocurrencies and their token counterparts as equivalent.

The process of tokenizing an asset through wrapping involves execution of a smart contract. In the process of wrapping shown in **Figure 3**, the value of the assets collateralizing the token is first locked by a smart contract and recorded on the blockchain as the new owner address. Thereafter, the smart contract issues the corresponding token or tokens at a fixed ratio, the issuance of which is recorded on the blockchain network hosting the smart contract.

Unwrapping of a token simply reverses the process, surrendering the token and releasing the asset. Since many smart contracts are written in Solidity to be compatible with the Ethereum blockchain and EVM, ETH and numerous versions of genericized WETH form natural token-wrap pairs.

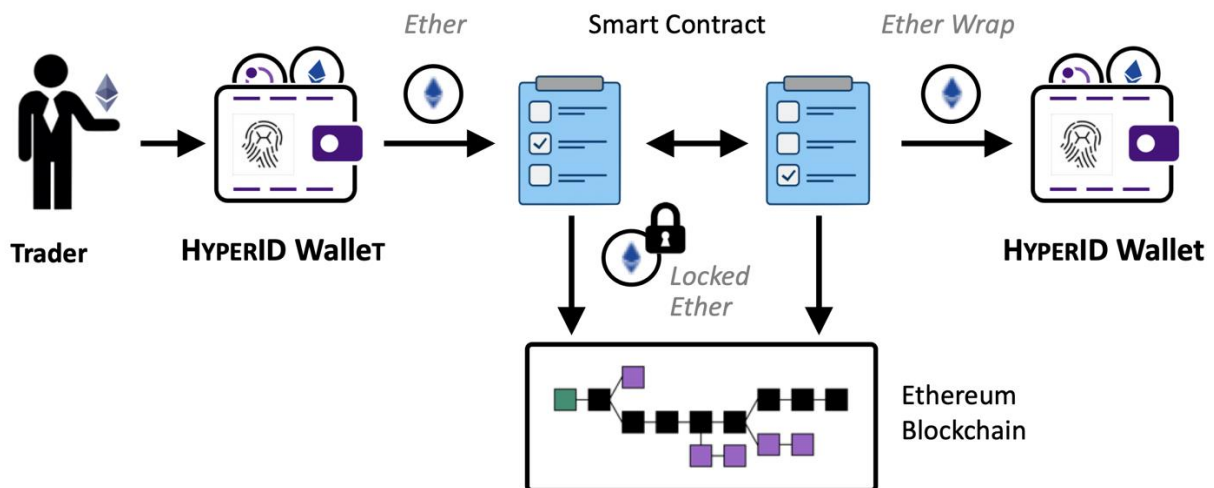


Fig. 3: Token wrapping of cryptocurrency using a smart contract.

Compared to transacting chain-native cryptocurrency (ETH, BTC, etc.), the benefit of using token wraps on a smart-contract enabled blockchain are numerous including:

- *Conditional transactions.* Using smart contracts and blockchain oracles, token wrap based transactions can be executed conditionally in response to changing market and economic conditions.
- *Increased transactional speed.* Smart contract validation of token wrap transfers is simpler and faster than manually executing trades via chain-native cryptocurrency, transactions demanding laborious block-by-block validation and juror consensus by all nodes.
- *Greater transparency.* Using token wraps, smart contracts provide superior transparency facilitated through unrestricted blockchain public access specifying token governance and transactional statistics (e.g. number of tokens outstanding, created, transferred, burned...)
- *Interoperability.* Token wraps are interoperable across a vast range of wallets, exchanges, and dApps and uniquely adapted to transact with assets on any fork (network) of the Ethereum blockchain. WBTC enables Bitcoin payment or swaps to be executed using Ethereum base smart contracts otherwise incompatible with Bitcoin transactions.
- *Portability.* Token wraps can be used to facilitate cross chain trading by launching affiliated smart contracts on two or more blockchains
- *Expanded applications.* Smart contracts facilitate a bridge between token economics and real world applications such as robotics, security, and IoT.
- *Enhanced security.* Tokenization enables users exclusive access of an asset's private keys.
- *Policy enforcement.* Tokenization provides a mechanism to enforce policies on-chain, providing transparent regulation while preventing single-party control abuse, manipulation, malfeasance, or asset misappropriation.

Technical standards for token wraps vary but generally mandate the method in which tokens are transferred and how users can access data regarding a particular token. The standards specify the minimum required token data need to facilitate trade analytics, exchange tokens, or to transfer them into or from a crypto-wallet. The Ethereum blockchain, for example, supports a number of technical standards for tokens, the most common one being ERC-20^[103] comprising a standardized API^[104] used for fungible tokens including transfer, payment, and balance-tracking functionalities. The acronym ERC, meaning Ethereum Request for Comments, refers to information requests, i.e. data exchange, needed to successfully process pending token transactions.

Fungible DeFi Tokens: The generation of fungible tokens by smart contract hosted on a permissionless blockchain provides a means by which new tokens are created without the need for mining and wrapping existing cryptocurrency. In this regard, executing a smart contract to generate tokens is ecologically sustainable (green) compared to PoW cryptocurrency mining. During the token generation process, a smart contract is authored specifying the type and quantity of tokens to be minted, then executed by uploading the smart contract to its intended host blockchain.

Once generated, DeFi tokens are sold (swapped) to investors. Concurrently, or in phases, the purchased tokens are distributed to investors through a process called a *token launch*. DeFi token launches may be executed on behalf of a variety of token sponsors, including

- A corporation, registered business, or DAO
- A project or dApp development
- A trading pool (a consortium of traders)
- An artist or content creator

Proceeds of token sales are directed to its sponsor which in turn has an implicit (or explicit) obligation to deliver a product or service for investment received. For example, a corporation is obligated to deliver a product or service; a project is committed to deliver working software or dApps, a trading pool is obligated to deliver a means for people to swap, stake, or borrow a specified lists of tradable assets, and an artist must deliver their creative works (or rights thereto) to collectors. Token offerings lacking defined deliverables are considered scams or fraud.

Since the tokens are issued via a permissionless blockchain, the crypto-economy considers the token launch as *decentralized finance*. Token distribution during a launch can be executed in a variety of mechanisms, namely

- Through private sales (and/or SAFT agreements) to venture capital (VC) funds
- Through airdrops (free gifts used for marketing purposes)
- Through exclusive IDOs (Initial DEX Offering) on select decentralized exchanges
- Through private presales to accredited investors (pursuant to KYC and regulations)
- Through a token generation event (TGE) on one or more DEXs or centralized DCXs
- Through a government approved whitelisting via a DEX or a centralized DCX

In all of the above cases (but airdrops), token distribution is limited to professional investors who have passed rigid KYC/AML verification generally as executed by a decentralized exchange (DEX) or a centralized digital currency exchange (DCX). General partners and investors in venture capital (VC) funds are also considered as qualified professional investors. Suspiciously absent from the foregoing list is an initial coin offering or ICO, a process analogous to a initial public offering (IPO) for equities where a company sells tokens directly to the unqualified public.

Following a brief but intense period of ICOs between 2017 and 2019, courts ruled that many ICO token issuances represented unlicensed public security offerings ^{[105]-[110]}. It was also determined that many issuers failed to perform sufficient identity verifications to confidently establish candidates as accredited investors. Expectedly, token sales made directly to the general public are now considered *verboden* in many countries and jurisdictions.

Even so, out of a preponderance of caution, many companies issuing tokens restrict their distribution, excluding nations banning or limiting the sale of digital assets and cryptocurrencies such as the United States of America and its territories, Canada, the United Kingdom, and the Peoples Republic of China (including Hong Kong). Investors should be mindful as to applicable laws in their country of citizenship and residence.

As a final point it would be remiss not to mention that the act of creating new tokens does not by itself represent a token generation event. There is no requirement to issue generated tokens at the time of the token's genesis event. In fact, there is no requirement to ever distribute them. For example, an engineering project seeking funding through a DeFi token offering may, for whatever reason, change plans and cancel their offering, generating but never distributing their tokens. If so, the smart contract goes abandoned, and perpetually persists on the blockchain without ever executing a token sale or a single transfer.

DeFi Token Standards: To ensure ubiquitous interchangeability, fungible DeFi tokens must comply with agreed standards specific to a blockchain. Two of the most common standards include Ethereum's ERC-20 standard and Binance's BEP-20 standard. These standards specify key factors determining an offering's tokenomics. The ERC-20 standard for Ethereum ^[111] specifies six mandatory and three optional function calls of argument (*), namely:

- `totalSupply(*)` describes a token's total supply, i.e. its total number in circulation
- `balanceOf(*)` displays the number of tokens in an owner's wallet
- `transfer(*)` specifies the amount of tokens to be sent and to what address
- `transferFrom(*)` enables a smart contract to automate transferring a specified amount of tokens from the owner to the receiver
- `approve(*)` approves withdrawal of a set quantity of tokens from the owners wallet by a receiving address
- `allowance(*)` verifies the owner's wallet contains at least as many validated tokens as specified in the `approve(*)` instruction
- `name(*)` optionally specifies the token's identity
- `symbol(*)` optionally specifies a token's 3 or 4 letter abbreviation
- `decimals(*)` optionally specifies the number of digits to the right of the decimal point when representing integers as fixed-digit numbers, a function required for fractional units of tokens (important when a token appreciates to an unusably high price)

ERC functions provide information needed for smart contracts to reliably perform prescribed tasks (such as transferring assets to wallets or to other smart contracts). Although tokens can be transferred directly using the `transfer(*)` function, sending digital assets manually implicitly carries risks. For example, transferring tokens to a smart contract incapable to receive them will result in irrevocable loss, one where the sent tokens will be credited to the receiver's address and the transaction recorded as "complete, but where the recipient's contract can't recognize them.

For exemplary purposes, numerous ERC-20 compliant tokens include Chainlink (LINK), Shiba Inu (SHIB), OmiseGO (OMG), EOS, Tron (TRX), ICON (ICX), Maker (MKR), Basic attention token (BAT), 0x (ZRX), Quant (QNT), IOT Chain (ITC), and Jasmy. The stablecoins Tether (USD₯), USDC, Dai, and BUSD are also ERC-20 compliant tokens. Ironically, the Ethereum blockchain's native cryptocurrency ETH is not.

Smart contracts using ERC-20 compliant tokens can reduce the risk of errors by invoking certain proofing and compatibility checks before executing a transaction, for example using the function calls `approve(*)` and `transferFrom(*)` transfer tokens pursuant to meeting defined criteria. Even so, ERC-20 compliant tokens are far from foolproof and subject to numerous failure modes.

To circumvent these issues, alternative token standards to ERC-20 have been developed ^{[112][113]} including backward compatible ERC-223 designed to prevent loss of funds during transfers. Other notable options include ERC-721 for non-fungible tokens (NTFs), ERC-809 for renting rival NTFs, and ERC-1238 for non-transferable tokens or badges. Despite offering enhanced features and benefits, widescale adoption of these alternate token standards remains elusive, and ERC-20 retains its unrivaled hegemony on the Ethereum blockchain.

Similar standards, sometimes with extended features exist on other blockchains like BEP-20 on the Binance smart chain, and evolving standards on newer blockchain networks. It should be cautioned, regardless of the asset traded by smart contract transactions on a given blockchain network and BCVM, the gas required to fund the community of validators executing the smart contract is specific to each blockchain. So while execution of a EVM smart contract transacting ERC-20 tokens requires ETH for gas fees, execution of a BSC smart contract transacting BEP-20 tokens requires the Binance BNB token, not ETH for gas.

Features of Tokens: The value and utility of tokens are not simply intended to emulate a fungible replacement for fiat currency. Cryptocurrency can do that. As defined by their role, purpose, and features ^[114], tokens can do things that neither fiat nor cryptocurrency can. Tokens can, for instance, can carry *state* information, passing genetic-like credentials conferring privilege, rights, benefits, preference, privity, privacy, or access despite no direct involvement between transacting parties.

Like an exclusive golf club membership, tokens can enable unique *privileges* to their holders by granting *access* ^{[115]–[118]} to products, content, or services unavailable without the token. For example, tokens issued to an entertainer’s fan club members can enable preferred seating unique to their most faithful followers. Tokens can also be used to provide economic *incentives* to users, allowing them to receive discounts for purchases, earn gifts, or upgrade to a nicer hotel room.

For example, online music purchases using tokens may receive bonus songs not included in an album. Accumulating tokens may earn free online streaming services of movies or music. Similar to airline miles, tokens can also be used to access unused inventory (like an empty business class seat) at discounts or for free. Unlike air miles which are administered by an airline or group of carriers issuing the mileage, however, tokens do not involve staff or costs to administer or distribute. In DeFi, there is no central authority issuing tokens or controlling distribution.

Another uniqueness of tokens is their *indistinguishability* ^[119]. Tokens of a specific type (e.g. USD₯, USDC, aIOT, UNI, YFI, etc.) are all identical in characteristics to all other tokens having the same symbol and therefore are indistinguishable from one another. A crypto wallet containing enterprise tokens cannot differentiate which tokens came from earlier purchases, which ones were acquired later at a higher price, and which ones were received free via a promotional distribution, i.e. a crypto *airdrop* ^[120]. ERC-20 tokens do not carry an identifying code defining their identity or able to register its owner like stock certificates do.

Because of indistinguishability, tokens do not bestow personal ownership rights or involve identity registration. Tokens used for riding a train or bus do not have the rider’s name on them. Anyone can use them. Accordingly, tokens offer the convenience of *transferability*, easily swapped among friends, bartered for favors, or donated for charitable purposes. Indistinguishability has one downside – it is impossible to determine if a token has been stolen and from whom. For this reason, the security and privacy provisions of crypto wallets holding tokens are of paramount importance to protect token assets from cybertheft.

Tokens can also carry economic value. Like chain-native cryptocurrency, ERC-20 compliant tokens created by wrapping a cryptocurrency (generically referred to as WETH), can be used in trading in crypto commerce in lieu of ETH or BTC. Unlike non-token cryptocurrencies, smart contracts enable tokens to facilitate *conditional* properties such as locking, pegged pricing, bonus awards, conditional releases, etc. not possible with pure cryptocurrency. For example, an airdrop of BTC enables its recipient to immediately trade their cryptocurrency while a token airdrop can remain locked for a defined time or until a specific condition is met. As such, tokens can be used to encourage behavior or incentivize long-term participation in a pool or business.

Tokens can also grant *privilege*— facilitating access to a product, service, or preference for its holder. Operating like a membership pass to participating businesses, a token holder can gain access to a venue, receive beneficial pricing, or request special privileges reserved for elite token types, e.g. access to a hotel’s presidential suite, book a private box at a sporting event, schedule a post-show meet-and-greet with a performing artist, request an exclusive VIP table held in reserve at a 5-star restaurant, etc.

As mentioned previously, tokens conferring benefits with undefined economic value are referred to as NFTs, an acronym for *non-fungible tokens* ^[121]. The benefits of NFTs, e.g. *ownership*, are not limited to non-cash transactions. Hybrid tokens may combine NFT features with embedded value.


Another unique capability of tokens compared to cryptocurrency is *awareness*. The trading of dumb cryptocurrency is unconditional. If the crypto coin is valid it can be immediately transferred or sold irrespective of economic conditions or the consequence of their trades. Tokens in contrast can solicit and consider off-chain conditions in managing transactions. By employing blockchain oracles to import information of on-chain and off-chain events, smart contracts can dynamically react, or using artificial intelligence even adapt, to changing economic and environmental conditions including prices, dates, volumes, etc. For example, smart contracts can limit a token’s use for services only in defined GPS regions or on specific cellular networks, thereby ensuring compliance with local legal ordinances. A smart contract may autonomously prevent or limit swapping (buying or selling) in times of volatile trading or price crashes.

Conditional value is of particular benefit in commerce especially in realizing *stable coins* – tokens which maintain a fixed price ratio to a commodity (such as gold) or which are pegged at a pre-defined ratio to the value of a specific fiat currency. USD₯ for example is a token whose commercial price is pegged (within some nominal range) to the United States dollar (USD).


Types of Tokens: Primarily motivated by defining tax implications of various crypto transactions, early purveyors of tokens bifurcated crypto assets ^{[122][123]} into two vague categories— utility tokens and security tokens. Because nearly any token can provide utility and also carry value, such an obtuse categorization provides little clarity into a token’s true purpose or capabilities. Nonetheless, the terms persist even today. For greater insight however, a more functional description of token variants can be obtained by borrowing the parlance of financiers and bankers, described as:

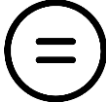
- Tokenized cryptocurrency (crypto wraps, stable coins)
- Index tokens
- Crypto synthetic assets
- Platform tokens (governance, infrastructure, project)
- Enterprise tokens

A brief description of these various token types is summarized below:

<p>Crypto wraps</p> 	<p>As described previously, <i>tokenized cryptocurrency</i> trading comprises stable coins and cryptocurrency wraps, tokenized assets used to transfer value, basically functioning as money in the world of token commerce. Wrapping of cryptocurrency also forms the economic bridge between fiat-based banking and token-based DeFi. And because ERC-20 compliant wraps can transact with one another on the Ethereum blockchain using a common smart contract platform, crypto wraps solve the problem plaguing altcoins isolated on their own blockchain forks, unable to trade beyond their network except through a DCX.</p>
--	---

Tokenized cryptocurrency is necessary to participate in most DeFi pools or to engage in token transactions. It is a common misunderstanding to new crypto traders that a DeFi pool, like a DCX, can accept fiat currency in order for a buyer to procure tokens just like buying cryptocurrency. At the present time, however, DeFi pools do not have the ability to convert fiat into crypto assets as such trades require a securities brokerage license in most countries. Incorporating a payment gateway (e.g. hosted by a DCX) into private wallets represents a potential mechanism to make trading directly from a private wallet possible, thereby eliminating error risk of token transfers.

<p>Index tokens</p> 	<p><i>Index tokens</i> ^[124] are tokens issued by DeFi trading pools and valued based on a specific industry or sector, essentially tokenizing an investment fund and sharing its returns among its investors. Most index tokens start with a liquidity provider funding the pool. The funds are then used to buy the original asset base of the pool. Investors subsequently swap tokens or stake assets to participate in the index, gaining or losing value in proportion to the market pricing of its underlying asset pool.</p>
--	---

<p>Synthetic tokens</p> 	<p>A powerful yet complex alternative to wraps and stable coins, <i>crypto synthetic assets</i> ^[125] represent the equivalent of hedge funds in crypto economy. The term “synthetic asset” refers to a mix of assets that have the same value as another asset. In traditional banking, for example, synthetic products combine derivatives such as options, futures, and credit swaps based a variety of underlying asset classes such as currencies, stocks, bonds, commodities, indexes, interest and so on. The same concept can be adapted for digital assets...</p>
--	---

In crypto synthetic trading, an investor locks collateral into a smart contract used to define a synthetic asset comprising either a premade product or one of their own design. A blockchain oracle monitors market trading of its underlying assets. Depending on whether a stock, interest rate, or future price moves as the investor expects or not, the change in value automatically credits or debits the change on the investor’s collateral balance. Hedging is generally employed to protect against excessive losses for an investor of crypto synthetics (and to protect the DeFi platform).

Platform tokens are tokens issued by a host platform for DeFi pools, issued and traded separately from the hosted pools. These include governance tokens, infrastructure tokens, and project tokens.

Governance tokens



Governance tokens ^[126] comprise crypto tokens issued by DeFi platforms allowing investors to partake in sharing operating profits of its host platform. The term “governance” refers to a token holder’s voting rights, albeit limited, in deciding how the token operates or evolves. Examples of governance tokens include UNI tokens from Uniswap. Since governance tokens are issued by the platform itself, the tokens can be considered as a form of marketing promotion encouraging platform use. Airdrops serve as a thankyou to long term participants of the platform. Asset growth is normally sporadic, made commensurately with a platform’s value appreciation. The value of a governance token is therefore based on the popularity of a platform more than on its profit generation potential.

Because different DeFi platforms focus on specific business concentrations, every platform’s governance token reflects the industry sectors the platform focuses on. For example, a platform may concentrate on tech-centric DeFi markets such as IoT, identity validation, DeFi protocols, cybersecurity, supply-chains, biomed, etc. The governance token of such a platform thereby reflects the aggregated opportunity of the technology sector rather than representing the holdings of any one specific DeFi pool or technology asset. In this way, a governance token functions like a sector-specific index fund, but comprising multiple DeFi pools rather than specific token-pairs.

Infrastructure tokens



Other types of platform-issued tokens include *infrastructure* tokens. Infrastructure tokens help fund technology developments benefiting a DeFi host platform and potentially impacting technology and fintech industries at large ^[127]. Most infrastructure token capitalized efforts involve creating autonomous software as dApps, which once deployed, operate in-the-wild in a permissionless, unregulated, and uncontrolled manner with no owner or system supervisor.

Project tokens



Project tokens are platform specific tokens used to fund various short-term developments and “one-off” projects, similar to crowdfunding ^[128] but through tokenization ^[129]. Although project tokens can be used to facilitate proof-of-concept, fund an event, raise public awareness, or demonstrate new ideas, project tokens do not secure ownership rights in a company or its work product. Funds raised are in essence “donated” to support the project in exchange for a “reward” typically access to special offers, models, or services including special purchase incentives and discount offers exclusively for token holders.

Enterprise tokens



Representing a broad token class, *enterprise* tokens^[130] are multifunctional digital assets, used by corporations and enterprises for a variety of purposes including fundraising, transactional commerce, business operations, marketing and sales, and more. Enterprise tokens are tokens backed and issued by an enterprise, including commercial businesses, non-profits, or governmental agencies. *Tokenized* businesses^{[131][132]} thereby represent any enterprise adapting its business model to employ the use of tokens in various aspects of its operations or phases of its commercial and organizational evolution, including funding.

Another token unique to the KAIZEN protocol, the collateralized transaction (kTx) *proxy* token, will be discussed later in this paper.

Transacting DeFi Tokens

DeFi token transactions occur in three major phases of an token's lifecycle, namely

- Token generation
- Token distribution
- Token trading

Creating a new token, i.e. token generation, involves executing a smart contract defining the type and properties of the issued token. Smart contract authoring is a critical element in properly executing any DeFi offering as the launching of a smart contract live is an irrevocable event – once done it cannot be undone.

Specifically terms of a smart contract cannot be changed after-the-fact and mistakes cannot be removed. The consequences of a badly written smart contract ranges widely from unnecessarily incurring users excessive gas fees to causing a complete loss of invested capital. Failure root-causes include arithmetic errors, logic errors, improper function calls, a stuck program counter (frozen states), incomplete or misdirected transfers, and more.

In essence, the tokenomics of an token offering and its execution are inexorably defined by its smart contract. The number of tokens generated, i.e. the token total supply, cannot be increased at a later date. Locked tokens cannot be unlocked unless the conditions specified in the smart contract occur, e.g. in accordance with a defined vesting schedule. Unlocked tokens cannot be locked after the fact and vesting schedules, once defined, cannot be altered. So the creation of a reliable error-free smart contract is fundamental to a successful token issuance. Smart contracts represent a particular type of software called a decentralized application or “dApp”.

Developing dApps: Writing new smart contracts is the role of a dApp developer, a team of software programmers familiar with blockchain-based decentralized applications. Decentralized application programming is significantly different than programming for a smart phone, personal computer, cloud server, web server, IOT device, or connected vehicle.

In conventional computing, the application program runs on top of an operating system which locally delivers dedicated resources including hardware comprising one or many compute engines, cache memory, and non-volatile storage. The hardware host resources are owned or leased by the software owner. When a user signs a service agreement with Amazon or Google web services, the contract is similar to a car rental in that the client owns the *right to use* the leased hardware for the specified term just as though they bought a server themselves, irrespective of which server in their cloud actually performs the tasks on any given time or day.

When a programmer has unlimited access to dedicated hardware, the software programs it hosts can be incredibly inefficient executing instructions that serve no purpose and storing data that isn't needed, because it simply is too small a concern to care about. Web programmers are particularly notorious for creating inefficient code based on pre-fabricated HTML templates, with compiled programs bloated by all their predecessor's dead-end coding contributions.

Once field of engineering where sloppy programming doesn't work is in software for real-time communication devices and services. In real time systems, inefficient code affects system performance. Doing unnecessary calculations slows the system's ability to transport data adversely impacting the network's latency and throughput, i.e. data bandwidth. In professional radio communication systems, for example, propagation delays not only affect network performance metrics, but may result in complete and irrevocable loss of packets, payloads, and content. Not surprisingly, only the best programmers are capable of real time programming.

Smart contracts may be written in any number of computer languages ^{[90][97]}, varying with both intended use and the targeted blockchain virtual machine host for contract execution. Programming languages used by various blockchain virtual machines and their blockchain users are listed here:

- C++: EOS, HyperLedger Sawtooth, Iota, Wanchain, Solana, Stellar
- C#: Neblio, NEO, Stratis
- F#: NEO
- Go: HyperLedger Fabric, Iota, Neblio, Stellar
- Java: Ardor, Corda, HyperLedger Fabric, HyperLedger Sawtooth, Iota, Neblio, NEO, Smilo, Stellar
- JavaScript: HyperLedger Sawtooth, Iota, Lisk, Neblio, Smilo, Stellar
- Python: HyperLedger Sawtooth, Icon, Neblio, NEO, Smilo
- Objective-C: Neblio

Many in the above list comprise general-purpose high level programming languages derived from C or Java ^{[133]-[135]}. Other commonly used smart contract development ^{[98][99]} include:

- DAML: HyperLedger Sawtooth
- Kotlin: Corda, NEO
- Lidity: Smilo
- LLL: Ethereum
- Michelson: Tezos
- PhP: Neblio
- Plutus: Cardano
- Ride: Waves
- Ruby: Neblio
- Rust: HyperLedger Sawtooth, Iota, Solana
- Scala: Stellar
- Scilla: Zilliqa
- Serpent: Ethereum
- Solidity: Aion, **Binance**, HECO, Enigma, **Ethereum**, Qtum
- WebAssembly: EOS
- Vyper: Ethereum
- Yul: Ethereum

Of the foregoing, Solidity, a Java derivative language, controls the lion share of the smart contract programming market as it is the primary language employed on Ethereum EVM and Binance BSC. Rust appears to be a rapidly growing alternative used by both HyperLedger and Solana. Other blockchains claim to be agnostic to the languages employed in their smart contract execution. They include the Kimodo, Multichain, NEM, OpenChain, and Polkadot networks.

Efficient bug-free dApp development requires extensive experience to circumnavigate programming pitfalls. Accidental arithmetic errors, fixed integer decimal placement, hidden loops, undetected wait states, and incomplete transfers can result in catastrophic losses to investors, affecting the success of a project or the survival of a company. Given the plethora of programming languages, cross chain launches are especially complex and prone to execution risks. Poorly architected code, even if bug-free can still suffer from excessive gas fees or slow execution during high traffic events resulting in opportunity loss for investors.

One way to avoid badly implemented or buggy smart contracts is to employ an autonomous smart contract authoring system. But writing a dApp which authors bug-free smart contracts requires even greater skills than manually writing a smart contract. Only limited enclaves of skilled developers exist across the world able to master such challenges.

Executing dApps: A smart contract launched onto its host blockchain becomes immediately available for distribution and ready for use. As illustrated in **Figure 4**, a function call to the protocol from a UI or API retrieves the most current code instance descended from the original parent block. Accessed through a user interface (UI) over a smartphone app or web browser, this up-to-date template (containing the BCVM's current world state) is downloaded, modified to include the pending transaction (including any token exchange) creating a new instance. The updated contract is then validated and uploaded back onto the chain. This latest uploaded instance is then used as the template for the next transaction, and so on.

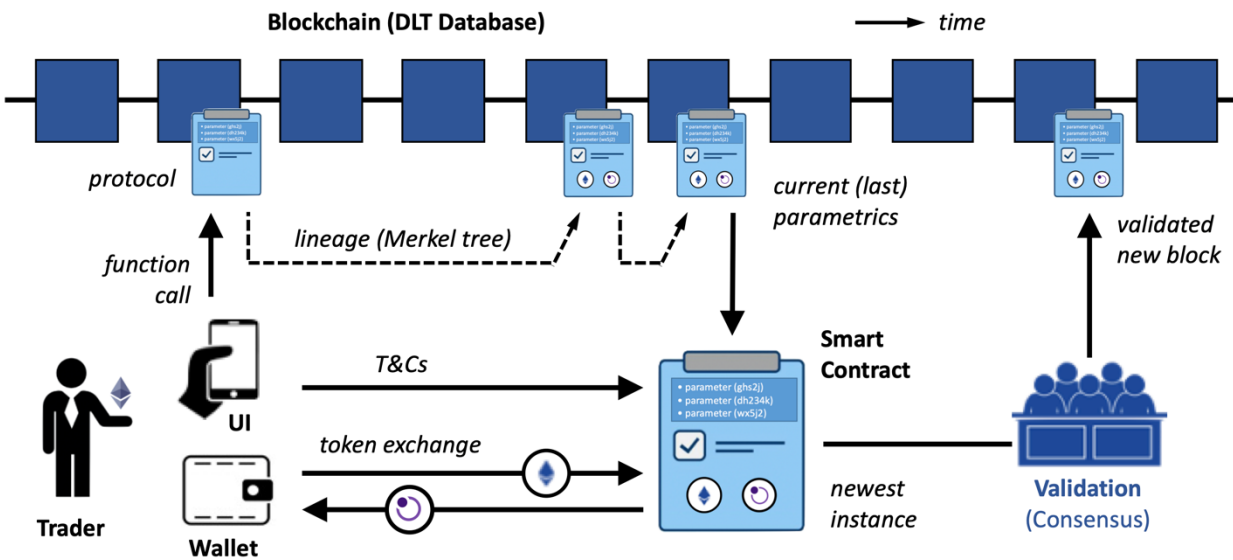


Fig. 4: Based on a common ancestral protocol, smart contract instances containing logic conditionals (if-then-else) and parametrics (quantity, return, price) are distributed, updated, and stored on the blockchain.

In greater detail, the smart contract lineage is traced from its parent to its last entry on the blockchain, i.e. its most current instance, using a peer-validated Merkel tree. Each instance of the smart contract is recorded on the blockchain sequentially using a time-stamped and peer-validated block. Like chain-native cryptocurrency, blockchain processing of smart contracts is executed by jurors to validate each new entry.

Dissimilar from cryptocurrency mining, however, miners validating smart contracts affirm the block's authenticity, but not its content. During validation, jurors do not (and often can not) interpret the meaning or functions contained within the smart contract. For reasons of security and privacy for example, portions of a smart contract can be hashed or encrypted and may not be decipherable by any trader except members within a defined pool. In order to ensure a contract is prepared properly, it is critical that only the most recent version of the smart contract is used as the starting point in writing a new contract.

The latest instance of a smart contract includes the most up-to-date contractual information, i.e. parametrics, needed to synthesize accurate executable code. Parametrics define conditional information that change over time needed to execute the smart contract, including date, time, interest rates, number and type of tokens in the pool, locking mechanisms, etc. For example, before a new contract can be written and recorded for swapping two tokens, the last smart contract must confirm enough tokens exist to execute the transaction. If a buyer tries to purchase more tokens than exist in the pool the contract cannot be executed. If a trader wants to execute a smart contract on the Ethereum blockchain using USD₯, the trader’s wallet must contain sufficient ETH for the gas fee or the transaction cannot be processed.

Although program execution appears as a simple chronological sequence, to maximize throughput of the network smart contracts are constructed in hierarchical layers (subroutine calls) to avoid unnecessary consensus validations. **Figure 5** illustrates the hierarchical representation of a blockchain virtual machine stack. As depicted, a blockchain sits atop the Internet’s TCP/IP protocol stack as OSI Application Layer-7, separately comprising its own 4-layer blockchain stack and BCVM sublayers thereof.

Unlike the Internet’s OSI layers, no standardized nomenclature for blockchains exist at this time. Regardless of terminology, the lowest layer in the blockchain stack is the infrastructure or network Layer 0 used to deliver network and communication services to the blockchain ^{[136][137]}. Blockchain Layer 0 provides services to blockchain Layer 1, the protocol or implementation layer which manages data, the mining and storage of block content and chain-native cryptocurrency; and system operations including the blockchain virtual machine (BCVM) and consensus mechanisms. The Ethereum virtual machine (EVM) and Binance Smart Chain (BSC) are examples of blockchain Layer 1 implementations.

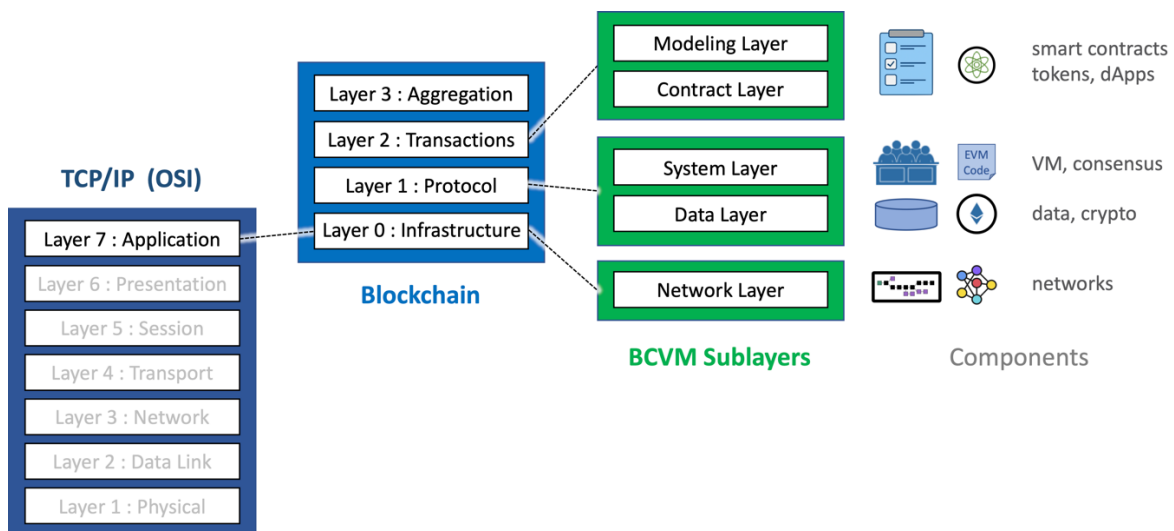


Fig. 5: Blockchain hierarchical layers

Blockchain Layer 1 in turn provides services to blockchain Layer 2, the transaction or application layer ^[138] used to execute dApps, smart contracts, and token generation on the BCVM, providing scalability, and offloading tasks otherwise consuming system bandwidth on blockchain layer 1 bandwidth. Layer 2 implementations may include lightweight computing capability called “state channels” that does not require the same frequency of validation as blockchain Layer 1 processes. In this regard, blockchain Layer 2 is faster but less secure than Layer 1. Note that the term “application layer” for layer 2 in the blockchain stack is not the same thing as Application Layer 7 in the Internet’s TCP/IP communication protocol (the Internet supports all blockchain processing).

Blockchain Layer 2 which relies on Layer 1 similarly provides services to blockchain Layer 3, the so-called Aggregation layer. The purpose of the aggregation layer ^[139] is to bundle applications and services of Layer 2 for the convenience of users. For example despite representing different applications, staking (lending) and borrowing may, for customer convenience be bundled into a unified service sharing a single DeFi pool.

Notably, by processing tasks on a single blockchain layer or spreading them across multiple layers impacts performance, efficiency and security. These tradeoffs led Vitalik Buterin, the founder of Ethereum to coin the phrase “The Blockchain Trilemma” ^[140] in recognition of a developer’s challenge that implementing blockchain processing with fewer validation checks improves speed but invariably compromises decentralization and security, considering if a network hosting a smart contract is

- Decentralized: creating a blockchain system that does not rely on a central point of control
- Scalable: the ability for a blockchain system support a large number of transactions
- Secure: the ability of the blockchain system defend itself from attacks, bugs, and other unforeseen issues to maintain reliable operation.

In this regard, blockchain architecture, virtual machine implementation, and smart contract programming across multiple blockchain layers determine the performance, security, and reliability of any blockchain implemented smart contract. As such, consistent high-quality smart contract programming is necessary to avoid hacks and failures in decentralized finance transactions.

DeFi Pools: As illustrated in **Figure 6**, hierarchical resources form the essential components of decentralized finance transactions in the cryptoeconomy. Based on the application layer of TCIP/IP, the Internet provides the framework for hosting perpetually-live permissionless blockchains even though no dedicated memory is available as non-volatile storage. The blockchain in turn operates as the host platform for a high-uptime blockchain virtual machine (BCVM) despite lacking any dedicated compute resources.

The blockchain virtual machine forms the host platform for the execution of decentralized applications including dApps and smart contracts – transactional processes able to confidently enable financial transactions and commercial business despite lacking any central authority, bank, treasury, or centralized financial clearing house [141].

In one class of smart contract used to issue tokens (called a *token launch*), DeFi tokens are generated and released into the wild in accordance with some defined schedule or set of conditions. Investors may acquire tokens as a direct purchase from the issuer, or alternatively by executing a token swap in a DeFi pool. In the context of a token launch today, a “DeFi pool” is a smart contract facilitating the exchange of two digital assets – the newly generated DeFi token and a crypto cash equivalent (such as USDC, BUSD, WETH, etc.).

Unlike a DeFi trading pool (where either asset in the pool may be bought or sold at will), during a token launch the newly issued token can only be purchased, not sold. Such a DeFi pool may be referred to as a *launch pad* or an *issuing pool*. Two methods exist for an investor to acquire a newly minted token in private token presale – either by a direct purchase or through a DeFi pool. In a direct purchase after completing KYC/AML and pursuant to applicable regulatory statutes, a buyer purchases directly from the issuer.

In DeFi, a prospective investor opens an autonomous transactional interface called a protocol and swaps a specified amount of crypto cash for the token being issued. In high-quality token offerings, demand generally exceeds supply. As such, the maximum purchase quantity is unavoidably limited, i.e. allocated, on a per-person basis.

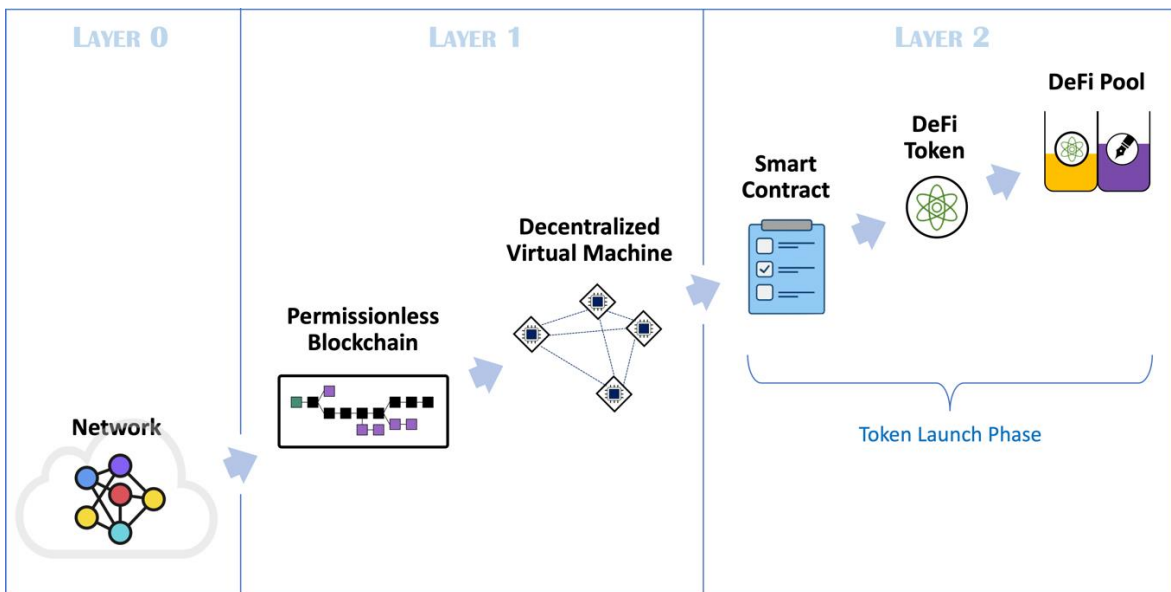


Fig. 6: The hierarchical role of decentralized resources in DeFi pools

Transactionally, a DeFi pool functions like a bank-less bank with “products” offering new issuances to venture capital, lending to stakers, borrowing for debtors, currency exchange (token swapping) for traders, and liquidity providing opportunities to market makers. Although a DeFi pool offers services much like a business entity, the pool itself is ethereal. There is no owner, operator, guarantor, or legal entity controlling its actions or defining its quiddity (its unique essence).

In this regard, the operation of a DeFi pool is autonomous, and its existence, virtual. DeFi is diffuse in nature, with no physical presence within a single device, server, location, country, or legal domicile. If something goes wrong, there is no one to sue, and no one to blame but oneself.

Experientially, users interact with the DeFi pool using what appears to be a web interface. In reality, a DeFi interface (UI) invokes an application program connecting their crypto wallet directly to the Ethereum, Binance, or other host blockchain. For the sake of security and privacy, no web interface should act as a data conduit in any crypto transaction. Web intermediaries cannot be trusted as they can easily capture, steal, and profiteer from access to a client’s security credentials and private keys exchanged during a transaction, including the risk of wallet theft.

Decentralized transactions thereby depend on trustless autonomous operation – not relying on anyone or anything to execute a transaction. Although theoretically, traders can write their own smart contracts to engage in transactions within a DeFi pool, in practice smart contracts are not written from scratch but instead are based on a contractual template called a *protocol* to define the specifics of a pool.

DeFi Protocols: Smart contracts play two major roles in supporting investors and entrepreneurs in today’s rapidly evolving crypto-economy, specifically to facilitate

- DeFi token trading for investors
- DeFi token launches for issuers

Although both of these functions can be performed using a hard-coded smart contract, custom dApp development is slow and prone to errors. The alternative is to employ a transactional protocol for smart contract authoring. A transactional protocol, or simply a *protocol*, refers to software that autonomously creates smart contracts in accordance with user instructions.

Theoretically, a smart contract authoring program for DeFi trading should be able to support a diversity of common DeFi transactions. In practice, most available protocols are designed to operate as single-function automatons, capable of performing only a solitary dedicated task, for example:

- Swapping tokens in a DeFi pool for a single type of cryptocurrency, *or*
- Staking a token in a DeFi pool for interest income, *or*
- Borrowing from a DeFi pool against locked collateral, *or*
- Providing liquidity to a DeFi pool as invested capital to earn fees

Separating investing and trading activities into separate protocols, while easier for developers, creates a more confusing user interface and a degrades user experience. Moreover, numerous inadequacies in the design and programming of present-day DeFi protocols adversely impact traders flexibility in participating in a DeFi pool. These inadequacies has an especially significant impact on companies or projects seeking to launch their own tokens. Important unaddressed issues limiting today’s protocols include the following:

- Inability to accept multi-currency payment (multiple crypto wraps and stablecoins)
- Inability to control early selling and prevent rug pulls by managing locking and vesting
- Inability to concurrently launch multi investor tranches with different vesting schedules
- Inability to support swapping (capital investment) combined with staking (earning)
- Inability to flexibly support new token launches
- Inability to support transactions and launches concurrently on multiple networks

Multi-currency Payments: Ideally investors using a protocol to acquire a token or partake in a new offering would like to pay in any form of crypto cash equivalent token that is convenient for them. In practice, however, as depicted in **Figure 7** each swapping pool comprises a single token-pair comprising the unique combination of one specific DeFi token and a particular cryptocurrency or crypto cash equivalent.

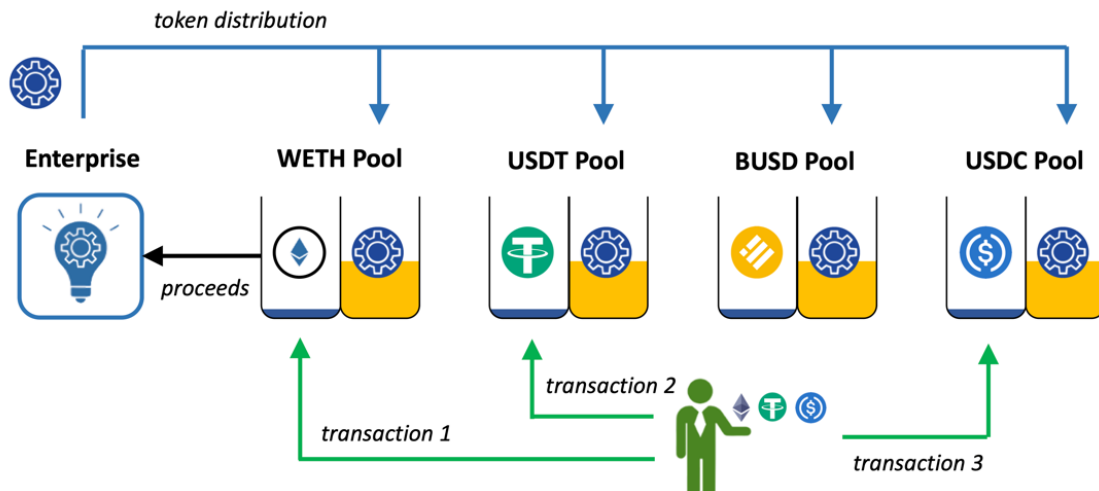


Fig. 7 Purchases via different cryptocurrencies requires trading in multiple pools

For example, purchasing a token such as Jasmy with USDC instead of WETH requires changing DeFi pools. Similarly a SHIB-USDT token pair cannot be mixed with a SHIB-BUSD pair, a SUSHI-WBTC pair cannot be pooled with a SUSHI-DAI pair, and so on. Aside from being terribly inconvenient for investors seeking to acquire tokens using a mix of cryptocurrencies and stablecoins, a rigid single token-pair DeFi pool forces issuers to allocate tokens to inactive or unpopular pools and tokens.

Multi-chain Token Trading: Most protocols are designed to operate on a single blockchain and are unable to support multiple networks, let alone cross-chain transactions. For example, a token offering on Ethereum employs EVM specific smart contracts not usable on the Binance's network or BSC virtual machine. This forces investors to trade exclusively on a single network even if transaction times degrade or gas fees become prohibitive.

Beneficially, cross-chain support as shown in **Figure 8** enables investors to optimize their trades and minimize gas fees and transactions costs. Unfortunately, most protocols are wholly incapable of supporting cross chain transactions or multi-chain token launches. Few developers today have the skill sets or acumen to author dApp protocols for multichain transactions or issuances. Another missing features is the ability to intelligently minimize gas and transaction fees by employing minimum transaction algorithms in multi-swap trades.

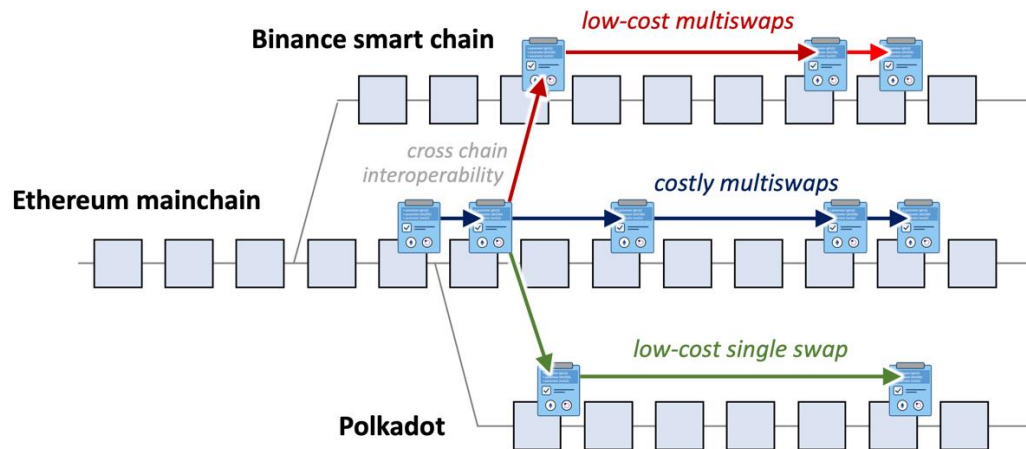


Fig. 8: Cross chain transactions allows investors to lower costs and improve network performance

Unlocked Tokens: A major issue plaguing DeFi investing today is the problem of early selling and insider rug-pulls. The early selling problem is intrinsic to a corrupt or badly-executed token launch where as shown in **Figure 9**, early investors with preferred pricing (e.g. VCs, exchanges, seed investors, and developers), are able to buy unlocked tokens at deeply discounted prices compared to anticipated future market prices of the offering. Similarly any DEX hosting an upcoming listing also acquires token inventory at preferred prices using a volume price agreement.

Later when decentralized exchanges launch the token for sale to its clients and it becomes tradable, e.g. during an initial DEX offering (IDO), the presale token holders secretly can execute a series of unauthorized sales undetectable by the DEX leading to a collapse of the token's price.

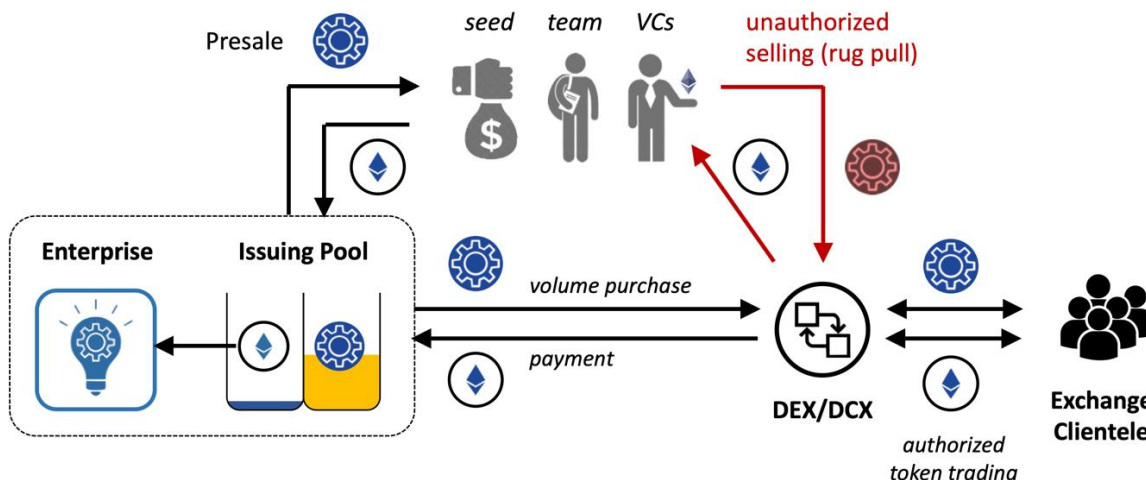


Fig. 9: Mechanisms of a rug-pull exploit comprising unauthorized selling by presale token holders

The problem with the unauthorized selling is there is no way to distinguish tokens approved for sale from those legally traded. Indistinguishability is a fundamental characteristic of a fungible currency, meaning once tokens are unlocked there is no way to restrict their trade. A rug pull is therefore the inevitable consequence of issuing unlocked tokens at the time they were first generated. One alternative means by which an issuer may prevent presale token holders from early selling is through delayed distribution – withholding the issuance of tokens to their buyers till a later date. This option although effective, is problematic in that few investors will agree to a purchase without some collateral protecting their invested capital until the token transaction is fulfilled.

Tiered Vesting Schedules: It is common in the tokenomics of a token launch to offer tokens in several tranches having different prices, lockups, and vesting schedules. Tiered vesting is important to ensure the entire population of investors and token holders don't unlock and try to sell at the same time resulting in a precipitous decline in token value.

In a token offering, tiered vesting is also important to protect the developers from early investors liquidating their position before the engineering team can unlock and be fairly compensated for their efforts. Finally, tiered vesting helps to prevent pump-and-dump (P&D) schemes as shown in **Figure 10**, where an investor accumulates a large quantity of tradable tokens through presales and low volume buying, then executes deceptive purchases in small amounts to artificially inflate the offering's market cap. After pumping the price and attracting momentum traders as buyers, the investor sells a large quantity in a few trades resulting in a landslide of selling. Tiered vesting keeps low cost tokens from providing inventory to pump & dump players. Most protocols however do not support token launches or tiered vesting schedules, so P&D remains a major issue in DeFi.

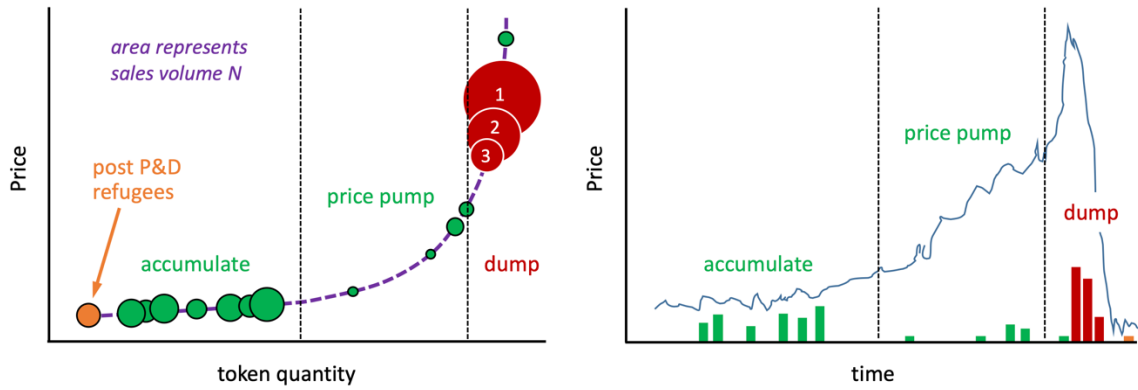


Fig. 10 Mechanisms of a pump & dump exploit using unlocked token trading

Issue-Swap-Stake Pools: One conceptualized (yet unrealized) method to prevent bulk selling of presale token holders is through swap & stake hybrid pools. In an issue-swap-stake hybrid pool investment is split into a combination of swapping (investing) and staking (income). Issue-swap-stake pools can (hypothetically) be implemented in two ways. In one implementation called a *Swap-then-Stake* pool shown in **Figure 11A**, buyers participating in a token offering first receive locked tokens which upon vesting automatically convert into a staking pool for a defined staking term, e.g. six months. At the end of the staking period the investor receives both their unlocked tokens and the interest they earned on the entire invested amount.

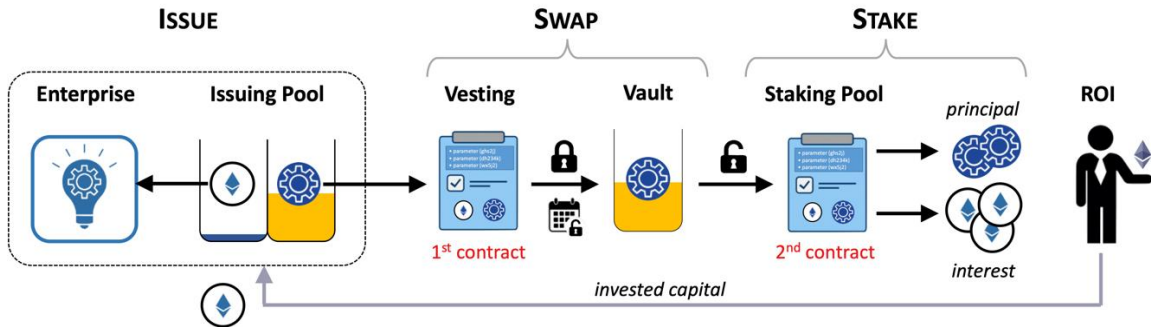


Fig. 11A: Swap-then-Stake pools pay investors principal plus interest at the end of the staking term

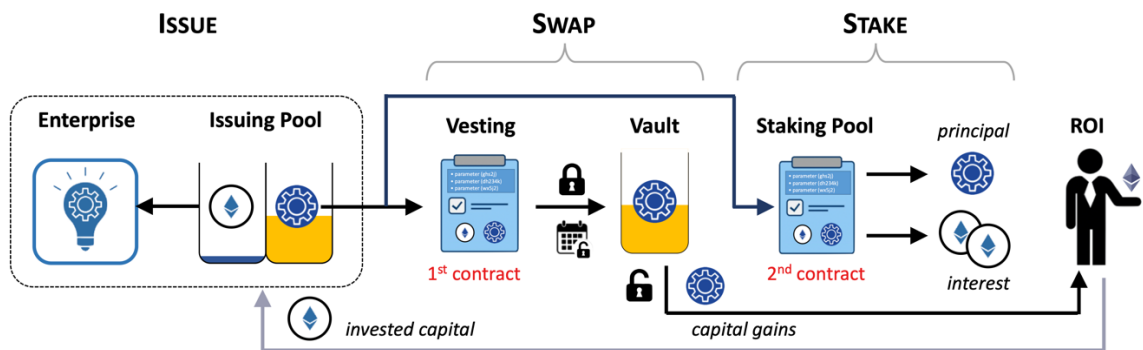


Fig. 11B: Swap-and-Stake pools pay investors concurrently unlocks tokens while earning interest

In an alternative version called a *Swap-and-Stake* pool shown in **Figure 11B**, the investment capital token purchase is divided into two portions, some fraction which unlocks purchased tokens through a defined vesting schedule, and a second portion where the issued tokens are automatically locked in the staking pool. Since the vesting schedule can be designed to unlock tokens at a different time than the expiration of the staking period, the problem of concentrated selling pressure from large private investors is ameliorated. Unfortunately, most protocols are incapable to support issue-swap-staking pools in any form, requiring separate smart contracts for each task.

Token Launch Platforms: As depicted in **Figure 12**, in the launch and listing of a new token the token issuer must support multiple sales channels including distribution through DEXs, DCXs, and DeFi pools. Token selling conditions sold through a token launch should be heterogeneous, able to support a range of prices, lock-up periods, and vesting schedules consistent with tokenomics specified by the issuer or their lead investors. Without the proper tool suite, the offering will invariably be plagued by unresolved issues affecting the financial performance of the offering and the reputation of everyone involved in the process.

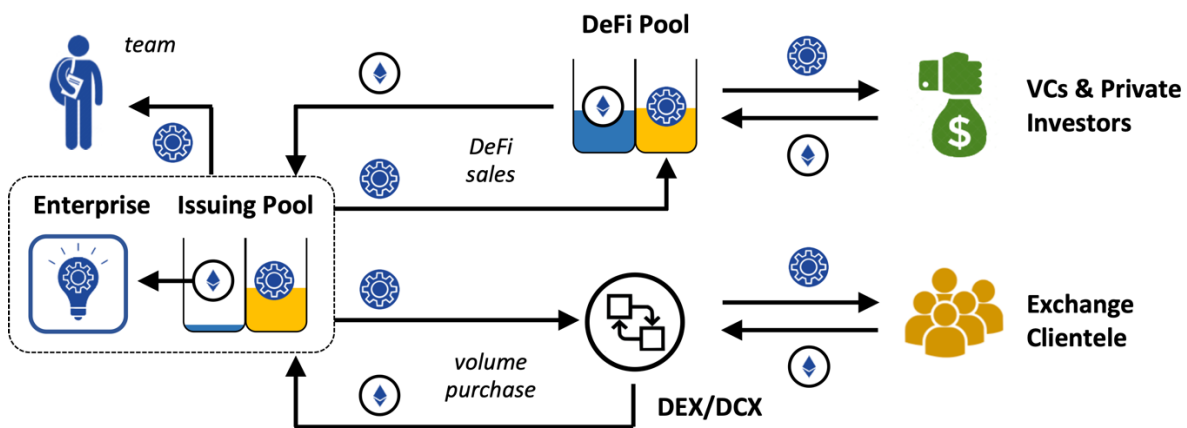


Fig. 12: Enterprise token launch with multi-channel distribution

Considering the limited choices in DeFi protocols today and their intrinsic deficiencies, we are forced to concede that:

- Today's DeFi protocols were created for dedicated token transactional trading in DeFi pools and DEXs, not for token launches.
- Even best-in-class DeFi protocols lack the functions and features needed to support and manage a well-executed token launch.
- Present DeFi protocols have no mechanism to prevent unauthorized selling or to protect IDO- and TGE-stage investors from unscrupulous private and presale holders liquidating their positions in rug pulls, pump-and-dump, and front-running exploits.

- Standard protocols only capable of swapping designated token-pairs constrain prospective investors to specific DeFi pools, cryptocurrencies, and blockchains, reducing the attractiveness and availability of new offerings.
- Most protocols operate on a single blockchain. Unproven protocols attempting cross-chain functionality are plagued by bugs, completion risks, and security vulnerabilities.
- DeFi protocols rely on unsecure crypto wallets with untrustworthy identity verification mechanisms (two party authentication is inadequate).
- No DeFi launch platform anticipates the life cycle management needs of a token offering.

The time is well overdue for a protocol and DeFi platform developed especially to meet the needs and address the concerns of projects, DAOs, and corporations launching their own tokens.

Introducing KAIZEN.FINANCE

KAIZEN.FINANCE represents the world’s first dedicated token launch platform, a token ecosystem especially created to support enterprises, DAOs, and projects seeking to launch their own DeFi tokens, and to provide token lifecycle management tools throughout a token’s entire generation, distribution, and unlocking process. As shown in **Figure 13**, the KAIZEN platform comprises a suite of autonomous KAIZEN dApps including the KAIZEN PROTOCOL, KAIZEN UI/UX, the KAIZEN AI ORACLE, and the world’s first collateralized trading platform (kDEX); along with a number of HYPERSPHERE technology enabled accessories including CYBERWALLET and HYPERID. A guiding principle for KAIZEN.FINANCE is its reliance on decentralization via autonomous applications. Given the financial magnitude of token offerings today, dApp operational autonomy is prudent and necessary to avoid the risk of hacks and exposure to launch misconduct or malfeasance.

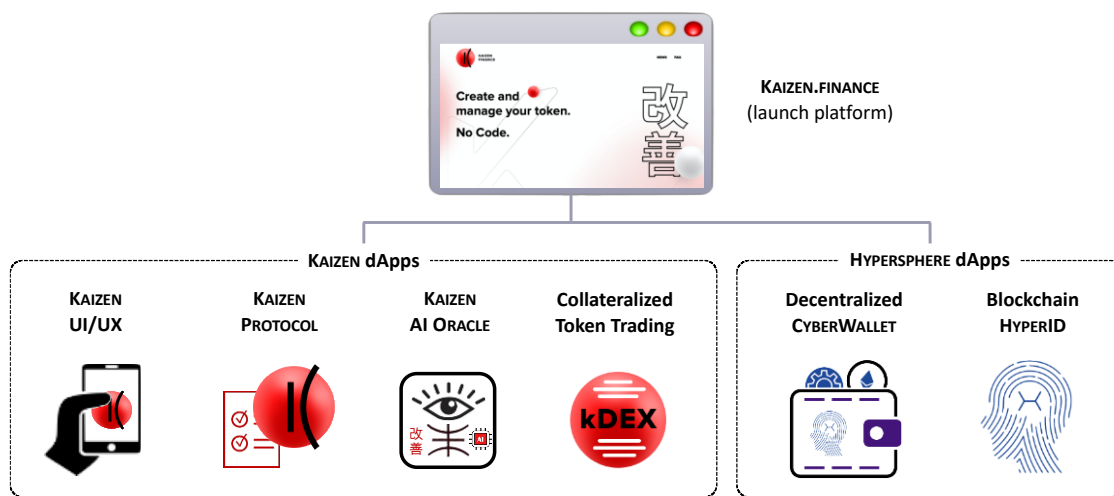


Fig. 13: KAIZEN.FINANCE launch platform combines KAIZEN PROTOCOL with advanced security tools

The key element of KAIZEN.FINANCE important in a token launch is the KAIZEN PROTOCOL, the decentralized application responsible for smart contract authoring, execution, and blockchain processing. As depicted in **Figure 14**, the KAIZEN token launch platform is responsible for issuing tokens in accordance with a terms and conditions (T&Cs) of the tokenomic model as agreed upon between the issuing enterprise and its lead investors, with said T&Cs verified by KAIZEN.FINANCE as executable by smart contract.

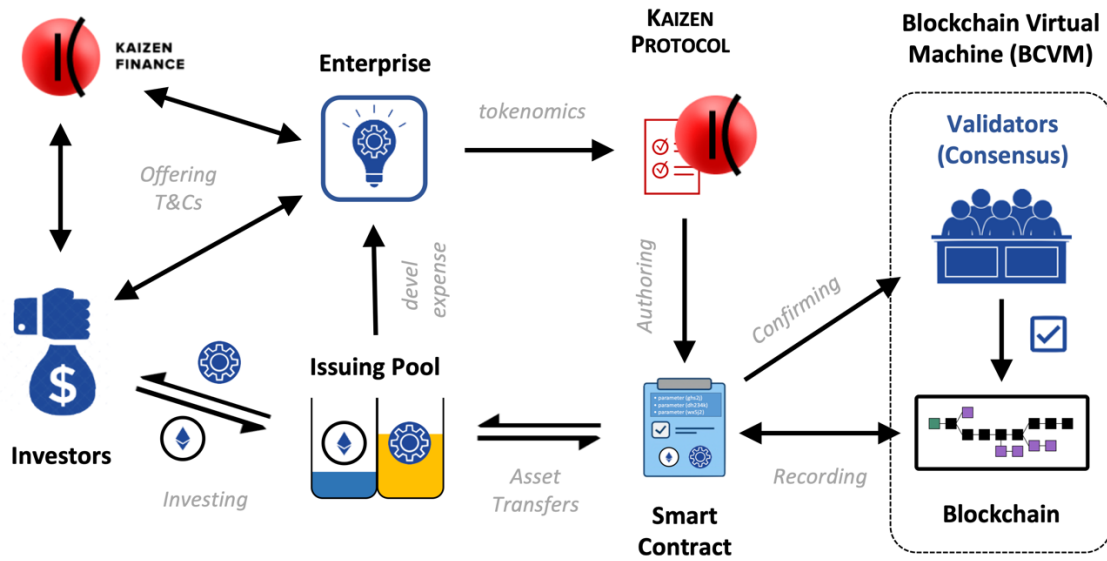


Fig. 14: KAIZEN.FINANCE token launch using KAIZEN PROTOCOL

Once the agreed tokenomics are entered into the KAIZEN PROTOCOL, the dApp autonomously authors a corresponding smart contract. Before it can be executed by the host blockchain virtual machine (BCVM), the smart contract must first be confirmed and validated by consensus and recorded on the block chain. If the contract is to be executed on multiple blockchains in support of cross-chain transactions, the authoring, validation and recording process must be repeated for each blockchain, noting that (as described previously) smart contract programming is BCVM specific. Once launched, the smart contract functions as a DeFi pool able to receive funds from investors and to respond by issuing tokens. Funds received from investment are used to pay for development expense. The DeFi pool itself is virtual, a construct of the smart contract used to manage all incoming and outgoing asset transfers.

Rather than executing the smart contract as a singular simultaneous event of investment and token generation, the execution of a smart contract using KAIZEN PROTOCOL supports sequential transactions based on the most current conditions, i.e. the last world state of the BCVM. The smart contract updating process is illustrated by example in **Figure 15**, showing uploading of the parent smart contract during token launch, followed by a private token sale, and a public presale transaction, recorded and time stamped in sequence on the chain.

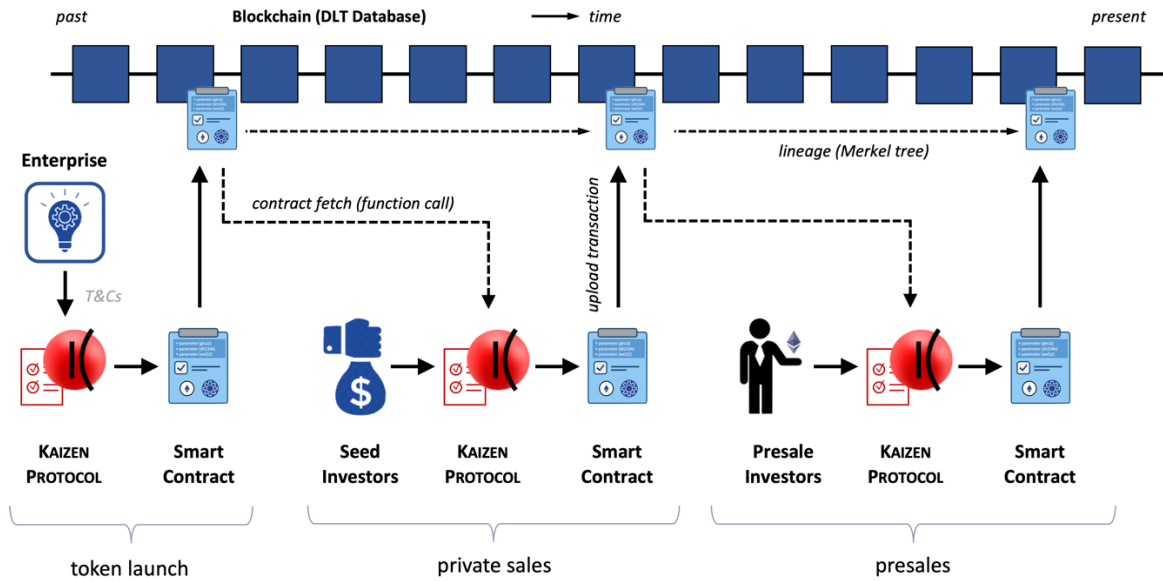


Fig. 15: Sequential execution of token launch smart contracts via multiple blockchain instances

As shown, during the private sales phase the parent smart contract (recorded during the token launch) is downloaded by KAIZEN PROTOCOL to record payment from the seed investors and concurrently issue tokens in response. As part of the contract update, the remaining total supply of tokens is updated and the newly revised instance of the smart contract is recorded on the block chain. A Merkel tree records the lineage of the of the revised contract from the parent contract.

The process is repeated again by presale investors who use KAIZEN PROTOCOL to execute a subsequent set of transactions, each time updating the contract as a new instance on the blockchain as a new leaf of the Merkel tree. In this manner, investor groups receive separate terms and conditions in accordance with their corresponding investment tranches as shown in the exemplary tokenomic model of **Figure 16**.

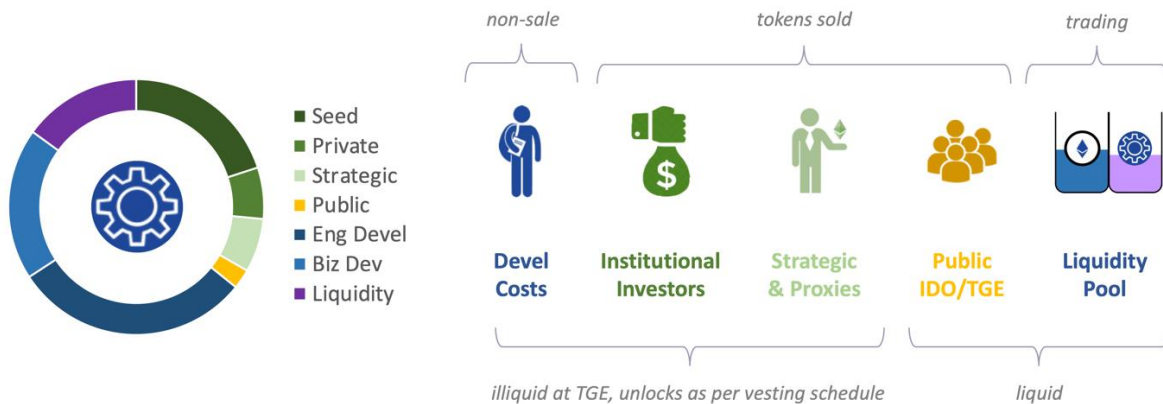


Fig. 16: Example token offering tokenomics arranged by class and tranche

KAIZEN Token Launch

Token launches executed by the KAIZEN PROTOCOL deliver numerous advantages over conventional protocols or hard-coded contracts. KAIZEN PROTOCOL token launch related features include:

- Automated bug-free smart contract authoring
- No coding required, familiarity with Solidity program language is NOT required
- Flexible support of multi-tranche tokenomics with customizable vesting schedules
- Smart contract authoring on multiple blockchains with cross-chain support
- Trustworthy protocol dApp certified by respected 3rd party security auditors
- Automated token minting and token distribution at TGE
- Pre-TGE private and presale token sales
- Tranche specific token locking & vesting features

Smart Contract Authoring: The KAIZEN.FINANCE launch platform enables the automated synthesis and customized authoring of multi-chain smart contracts from BCVM-specific templates. As shown in **Figure 17**, a token offering comprising an agreed upon set of tokenomics can be used generate multiple blockchain-specific smart contracts. In each case, the dApp authoring module downloads a previously generated, 3rd party security-validated KAIZEN PROTOCOL template then populates the blockchain-specific contract template with the offering tokenomics of the sponsoring project. The output of the authoring module is the parent smart contract for the offering, a contract specific to one blockchain and written in a programming languages unique to its BCVM host, e.g. an EVM smart contract written in Solidity for Ethereum, a BSC smart contract in Solidity for the Binance smartchain, an EVM smart contract written in Rust for Solana, etc. The KAIZEN PROTOCOL then uploads the completed smart contract to its target blockchain pursuant to validation.

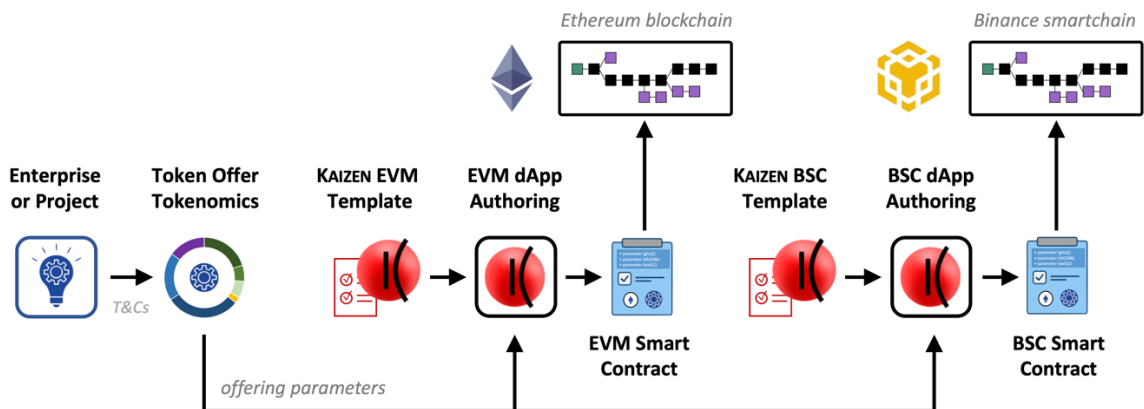


Fig. 17: KAIZEN authoring of multi-chain smart contracts

Token Generation. The execution of a smart contract provides the basis for generating a fixed number of tokens in accordance with the tokenomic offering it implements. Although the KAIZEN.FINANCE launch platform is capable of creating tokens for an enterprise, DAO, or project at any phase in the entity’s lifecycle, the most common implementation is to author a single smart contract executed at the time of the token generation event (TGE). As represented in **Figure 18**, during this TGE, KAIZEN authors a smart contract which in turn mints and concurrently distributes a prescribed number of tokens to a variety of destination addresses including wallets, DeFi pools, and exchanges.

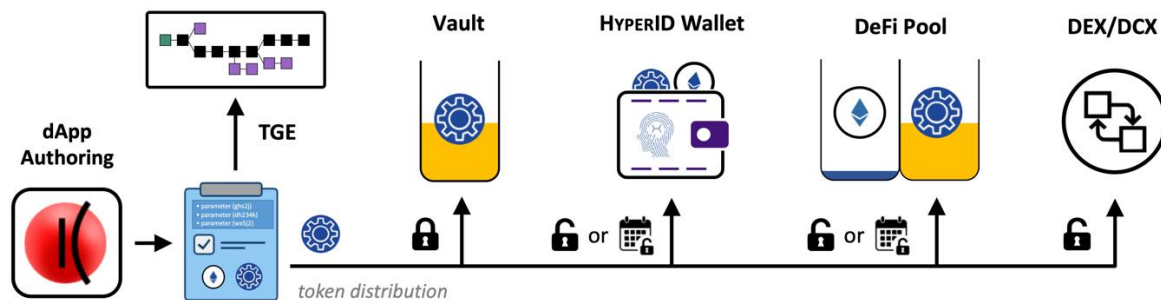


Fig. 18: Token generation and distribution mechanisms

A KAIZEN authored smart contract launch platform provides several key functions in the token issuing process, namely it:

- Defines the total token supply to be issued
- Generates the tokens in accordance with defined standards for fungible tokens
- Functions as a issuing DeFi pool for swapping crypto cash equivalents for unlocked tokens
- Supports token sales (swaps) at distinct prices for separate investor tranches
- Facilitates a treasury function for the issuer (holding crypto cash payments received)
- Distributes unlocked tokens to defined addresses at TGE (pools, wallets, exchanges)
- Retains undistributed tokens in reserve (vault function)
- Provides mechanisms for issuing tokens as per defined vesting schedule (limited)
- Provides a mechanism to vest and distribute non-sale tokens to insiders and team
- Supports optional co-generation of NFTs (as per ERC-721) or governance tokens (as per ERC-1404) for marketing or regulatory purposes.

In essence, the issuing smart contract acts as both a DeFi pool for transacting token sales, and as a treasury vault holding all unissued tokens and all cash equivalencies (cryptocurrency or stablecoins) received. KAIZEN authored smart contracts also unique support cogeneration multiple token types in the same contract using serial execution, e.g. first ERC-20, then ERC-721, etc. In this manner, traders making significant investments can earn non-fungible token rewards, or be granted governance voting rights not available to smaller participants.

Insomuch as vesting and unlocking of purchased tokens can be embedded into the smart contract as part of token generation, it is the nature of *token distribution* that controls a token's circulating supply and thereby its price-per-token (PPT) and correspondingly its market cap.

Token Distribution. Necessarily during a listing, unlocked tokens are supplied to select centralized and decentralized exchanges and to DeFi pools in order to stimulate *market making* – creating trading liquidity through buying and selling of unlocked tokens. If however, all the tokens created by the smart contract are distributed and become tradable at the TGE, the token supply will greatly exceed market demand at that time. Early overselling and profit taking can destroy the prospects for a successful token offering by discouraging new investment and limiting trading to the community of “flippers” present only during the time of the TGE.

To circumvent a price crash at TGE, a token's *circulating supply* is best restricted to comprise a small fraction of its total available supply, a predefined quantity precisely specified in the smart contract used to create it. But how can supply be constrained to prevent overselling? Offering a mix of locked and unlocked tokens at TGE is not as easy as it seems...

At first glance, token locking may appear to trade analogously to restricted stock – equities that cannot be sold until they are released for trading. But equities represent *registered securities* subject to statutory regulations of the Securities & Exchange Commission or other jurisdictional authorities. The unlocking of a restricted stock involves a legal process executed by a law firm or licensed broker to remove the “restricted” stamp from the back of the stock certificate to enable its selling. In other words, stock trading is regulated by central authorities, either the Securities and Exchange Commission or other applicable authorities in various legal jurisdictions.

In contrast, cryptographically generated DeFi tokens are not registered securities. There is no central authority who decides when a token may or may not be traded. There are no lawyers or brokers involved in releasing a generated token for trading. And because fungible tokens are ubiquitous and indistinguishable, there is only one type of token – an unlocked token. So how is it some tokens can be locked and vested over time if all tokens are tradable at the TGE? Locking “unlocked” tokens can be accomplished by several means:

- Not generating the tokens until certain conditions are met, i.e. just-in-time (JIT) issuance.
- Creating tokens in a smart contract but delaying their post purchase transfers.
- Sending purchased tokens to a custodial wallet and not releasing the wallet's private keys until the token unlocks.

All of the foregoing methods are problematic because they involve an element of time – no investor can receive their tokens at the time of purchase but must wait until the TGE to receive them. Even worse, issuance of most pre-TGE private sales must be distributed over a period of many months.

For example, JIT token issuance forces a smart contract to delay issuance of purchased in accordance with vesting. Issuance thereby requires a smart contract to *push* tokens to the buyer's wallet as they vest, costing the token issuer undo gas fees. The second bullet item, where an investor purchases tokens but must wait till they vest to receive anything, is left with the disquieting feeling that they have no receipt proving the money was invested.

The third bullet, withholding the private key of a custodial wallet for an investor's designated investment can be used for cliff vesting but not for executing tiered vesting schedules. In tiered vesting multiple categories of project affiliates and investment tranches each have their own unique vesting schedules. For example, if separate wallets were employed to facilitate unlocking seven categories of token holders each vesting weekly for a year would require an unmanageable 364 separate wallets and 364 PKI key exchanges. If instead, the smart contract executed a weekly push of tokens to each token holder's wallets, the transfer would require 364 transaction and gas fees totaling over thirty thousand dollars of funds wasted.

In conclusion, locking is therefore not a property of a token itself but how is distributed. And since a token generation event (TGE) by definition, enables immediate public trading of all unlocked tokens, care must be executed in token distribution to prevent front-running, rug-pulls, and pump-and-dump exploits. In other words, although a smart contract controls the total supply of a token offering, it is the method of token distribution, not token creation, that determines whether the token is tradable. While existing methods are sufficient for supporting a few large investors, today's protocols are woefully inadequate in handling a larger community of buyers.

As a unique solution to this perpetual quandary, KAIZEN.FINANCE and the KAIZEN PROTOCOL pioneered an alternative unlocking (vesting) and token distribution method called the collateralized transaction (kTx) token. Having successfully distributed well over 10 billion kTx tokens, the novel method has become an invaluable tool in flexibly supporting a wide range of tokenomic models.

KAIZEN Collateralized Transaction (kTx) Tokens

One of KAIZEN's unique innovations is a special category of proxy token called a kTx proxy token, (or kTx token for short) symbolizing a *collateralized transaction token*. To understand kTx tokens and what their used for we must first understand what a proxy token is.

What Are Proxies?: In general, the concept of a *proxy* means a delegated power or authority granting an agent the right to act on its behalf, i.e. a representative. This idea can be adjusted to a variety of applications. In law, a proxy can be a legal agent acting on behalf of a litigant. In a corporate shareholder's meeting, a proxy can be a representative casting votes on behalf of a constituency of shareholders.

In computer networking, a proxy server is any machine that translates traffic between network or protocols such as a NAT (network address translator). By logical extension then, in crypto-economics a proxy token is a crypto token that derives its value from an underlying asset, object, or contract logic that it represents. By the forgoing definition, stablecoins, tokens backed by fiat money or precious metals can be considered as proxy tokens.

In HYPERSPHERE vernacular, security backed tokens are not referred to as proxies, but by a more insightful nominative *mirror coins* because the proxy literally mirrors the market value of the security used to collateralize them, be it fiat money, gold, oil futures, cryptocurrency, or synthetic assets. Once a mirror coin is pegged to its underlying collateral, the worth of the proxy dynamically tracks the original asset's market value fluctuations. Uniquely, the KAIZEN.FINANCE proxy token doesn't represent a convertible value (for example like $\text{value}\{1 \text{ USD}\text{₹}\} = \text{value}\{1 \text{ USD}\text{\$}\}$).

Instead KAIZEN uses a defined swap ratio, i.e. where the collateral swap ratio $k = (\# \text{ of token A}) / \# \text{ of token B}$). In such an instance, the quantity ratio, not the value ratio, is fixed. The proxy is then not a mirror of its collateral asset, it is a new tradable asset. Such an adaptation is more useful in performing transactions (Tx) because the proxy needn't involve the complex financial mechanisms required to peg its value to a market traded asset. By removing price from the equation, the token's value is allowed to float relative to the underlying assets that gives it value. The perceived value of the proxy token can then be valued in accordance to its commercial potential and beneficial trading features, e.g. vesting schedule, staking terms, special offers, etc., not simply its stored value (like mirror coins).

**kTx
tokens**



Accordingly, in the KAIZEN PROTOCOL proxy tokens have been reengineered into a new and entirely different asset class called *collateralized transaction tokens*, or kTx proxy tokens. A kTx token is designed to function like the collateral it represents, yet is not subject to the same limitations, terms, and conditions as its underlying assets. In use, a kTx token functions as a derivative of its base asset, e.g. an enterprise token sold separately under a different and distinct symbol. A kTx token can be issued as part of a pre-TGE private sale or be created for a DeFi pool holding invested assets. It can comprise a single asset or may represent a collection of investments similar to a mutual fund, each with their own vesting schedules and purchase prices.

Unlike currency equivalents, kTx tokens are issued by or for DeFi pools to represent and embody the business potential of its commercial holdings – the enterprise, DAO, project, or infrastructure it sponsors. As such, a kTx token may be issued before, during, or after its underlying asset is created, meaning collateralization of the kTx token is unrelated to its asset's liquidity. And because a kTx token is generated as a proxy on behalf of a DeFi pool, the spot value of the fungible asset retained as collateral in the pool is irrelevant to the kTx token's distribution.

Issuing kTx Tokens: The creation, sale, and distribution of a kTx collateralized token involves the following actions

- Launching an issuing pool generating “enterprise” tokens for an enterprise, DAO or project
- Creating a swap pool comprising a token pair for enterprise tokens and kTx tokens
- Collateralizing the kTx swap pool by transferring unlocked enterprise tokens to the swap pool
- Generating kTx tokens in the swap pool in a precise ratio to the enterprise tokens deposited, typically on a one-to-one basis.
- Launching a kTx trading pool for selling kTx tokens and interlinking it to the swap pool

This process depicted schematically in **Figure 19** shows an enterprise’s issuing pool transferring enterprise tokens to the swap pool, which also contains a calculated number of kTx proxy tokens. As denoted by an icon showing a calendar and lock, the transferred enterprise tokens are locked by the swap pool upon transfer and unlocked in according with some specified vesting schedule. According to a define swap ratio k (typically 1:1) a corresponding number of kTx proxy tokens are made available for sale to qualified investors through a kTx selling pool. The swap pool can be collateralized order-by-order or in a single bulk transfer.

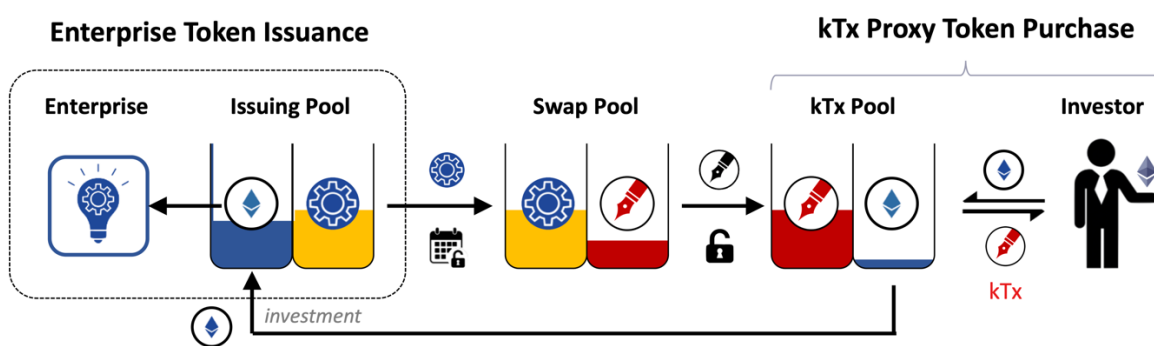


Fig. 19: Issuing kTx collateralized proxy tokens to investors

A third pool, the kTx selling pool is launched to facilitate kTx token sales and purchases. In exchange for swapping crypto cash, the kTx selling pool delivers unlocked kTx tokens to investors comprising kTx tokens issued by the swap pool. The proceeds of the sale are delivered to the enterprise issuer either directly or via the swap pool’s smart contract. Because every kTx token issued has a corresponding enterprise token held in the swap pool, the kTx tokens are *collateralized* by enterprise tokens, having the same prospective future value of the enterprise token itself. But unlike the enterprise token locked in the swap pool, the kTx token are not locked. kTx tokens can be delivered to the investor’s wallet as proof-of-purchase at the time of purchase and exchanged for enterprise tokens at a later date. As described the enterprise token thereby acts as collateral for the kTx token and the kTx token enables investors a means to acquire the enterprise tokens in advance of their issuance or TGE.

Together, the swap pool and the kTx selling pool shown in **Figure 20** operate in concert like a tri-token DeFi pool enabling investors to use crypto cash to acquire unlocked kTx tokens and when vested to convert them into unlocked enterprise tokens. Meanwhile, the pool safely withholds access to unvested (locked) enterprise tokens to prohibit premature trading.

This kTx swap pool thereby protects the enterprise token issuer from unwanted early selling pressure in the exchange markets while providing purchasers with proof of their investment. Because the swap pool and kTx selling pool function transactionally in tandem there is no reason to issue two separate smart contracts to manage their operation (although it can be done so if requested). In other words the enterprise issuing tokens need only launch a kTx swap pool smart contract to distribute locked tokens with prescribed vesting schedules, and without complicating the smart contract used to launch enterprise tokens at TGE.

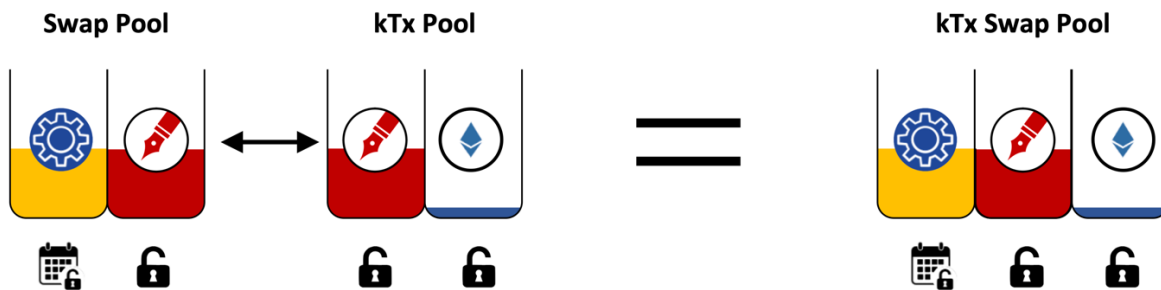


Fig. 20: Swap pool and kTx selling pool function as tri-token DeFi pool

Using the tri-token DeFi representation to illustrate the purchase of kTx collateralized tokens, **Figure 21** illustrates investors are able to use crypto cash to invest in an enterprise, receiving kTx proxy tokens at the time of purchase. Issued kTx tokens are collateralized by enterprise tokens locked in the pool until they become vested, providing investors *comfort* that their token investment has real value and is not a fraudulent transaction.

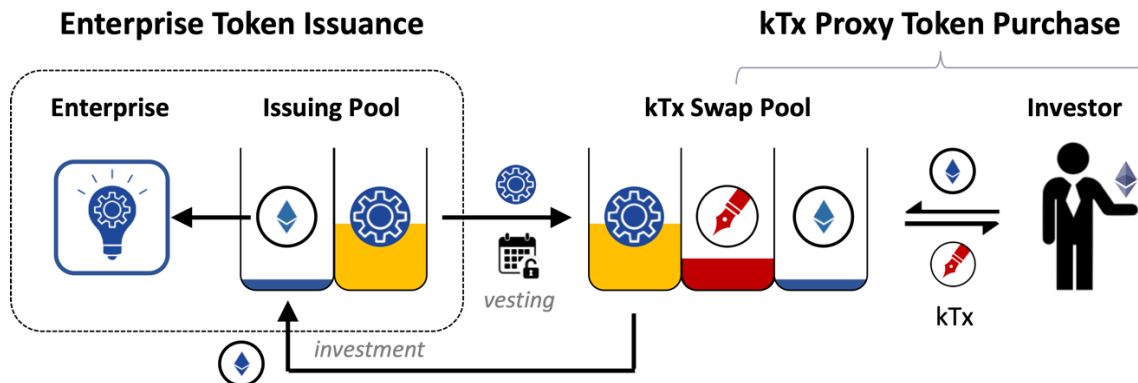


Fig. 21: Tri-token pool representation of kTx collateralized token issuance

Managing Token Unlocking: Contrasting the limited capability of conventional methods for managing token vesting to KAIZEN’s unique approach employing collateralized kTx tokens, the benefits become clear. With conventional platforms, enterprise tokens generated at TGE must contain vesting and unlocking mechanisms coded into the offering’s smart contract, greatly complicating contract execution. Moreover, at vesting, tokens must be pushed to recipient wallets costing an issuer gas fees. The issuer must manage vesting schedules, unlocking, and distribution long before TGE. A better option is for investors to visit a DeFi pool to claim their vested tokens.

Using a kTx swap pool, vesting and unlocking of a new offering’s tokens becomes trivial. During TGE, the total number of generated enterprise tokens to be unlocked over time are transferred *in bulk* to the kTx swap pool as *unlocked tokens*, i.e. as a single transfer comprising one blockchain transaction. Concurrently (assuming a 1:1 swap ratio) the same number of kTx proxy tokens are added to the kTx swap pool for transactional liquidity. The pool itself is created by a smart contract which manages the unlocking and redemption process.

Once an enterprise has transferred its tokens to be vested to the kTx swap pool, the issuer is finished with headaches of managing token distribution. The kTx swap pool then takes over in autonomously deploying tokens in accordance with the tokenomic model of the offering including any required last minute changes.

By moving vesting into the kTx swap pool, tremendous flexibility is possible for implementing complex tokenomic distributions comprising (i) different vesting start dates (TGE, 6 or 12 months), (ii) different vesting periods (0 to 24 months), (iii) different vesting schedules (linear, cliff, cliff-linear, non-linear), and (iv) different vesting intervals (weekly, monthly, quarterly). An example of the cumulative unlocked tokens comprising seven vesting categories along with unlocked allocations for TGE public sales and DEX pool liquidity is represented in **Figure 22**. Despite the complexity of the tokenomics, using a kTx swap pool to manage locked tokens enormously simplifies the process of vesting and token distribution.

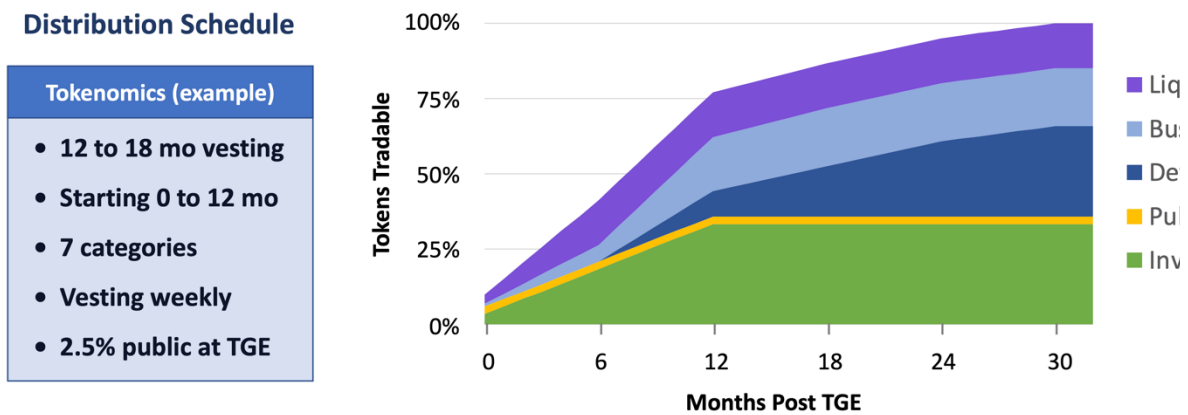


Fig. 22: KAIZEN token distribution as per tokenomic model (example)

Redeeming kTx Tokens: Once a portion of the kTx swap pool’s enterprise tokens become vested, an investor has several options, either to (i) hold their kTx tokens, (ii) redeem kTx tokens for the unlocked portion of enterprise tokens, or (iii) concurrently redeem their kTx tokens and liquidate all (or some portion) of the unlocked enterprise tokens at market prices on a decentralized exchange. Other alternatives (discussed later) involve collateralized token swapping or staking.

As shown in **Figure 23**, investors have two means by which to convert kTx tokens into fungible assets by swapping them for vested enterprise tokens. In one method called kTx token redemption, an investor redeems their kTx tokens for the swap pool’s collateral, i.e. vested enterprise tokens. In a second method called kTx token liquidation, an investor redeems their kTx tokens for vested enterprise tokens to be sold on a DEX, and receives the proceeds therefrom in crypto cash. In both cases the kTx tokens once redeemed, are burned (i.e. destroyed) after the transaction.

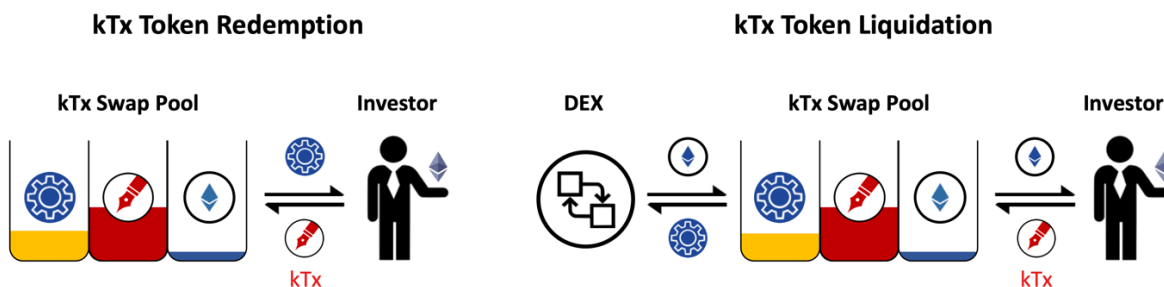


Fig. 23: Converting kTx proxy tokens into fungible assets

Because the kTx swap pool manages token unlocking by smart contract, neither a token issuer or its investors must do anything to take possession of their tokens. The tokens are already reserved only for kTx token holders. Whenever they wish, an investor can personally claim their kTx tokens from inside their wallet simply by opening the KAIZEN PROTOCOL and executing a token swap between their kTx and vested enterprise tokens. As such, transaction expense and gas fees are, rightly so, incurred by the investor, not the issuer.

Moreover, since the protocol automatically limits swaps to only vested tokens, there is no risk that locked enterprise tokens can be prematurely withdrawn and sold or otherwise hypothecated. In this manner the investor has the freedom to monetize their investment on their own terms and the token issuer needn’t worry about managing token vesting or distribution after the TGE. The redemption process also delivers an added degree of security, because only validated kTx token holders can swap their kTx tokens for unlocked enterprise tokens.

In the redemption process kTx tokens are exchanged, i.e. swapped, to claim and download the previously purchased enterprise tokens. Lacking kTx tokens, frauds and hackers have nothing to swap and are unable to even commence the download process. In this sense, kTx tokens provide an additional benefit as a cryptographically unique purchase receipt, i.e. a digital proof-of-purchase.

Issue, Swap, Stake: Besides vesting, another means by which an issuer can discourage *en masse* unlocking and concurrent selling of early private tranche and presale buyers, is by combining token purchases (swapping) with lending (staking). Where swapping involves the irreversible surrender of one digital asset for another (e.g. swapping cryptocurrency for enterprise tokens), staking involves temporarily locking an crypto asset in order to earn interest over some defined interval, often ranging from three months to one year. Early withdrawal (if allowed at all), may result in the complete surrender of all earned interest and possibly involve an early withdrawal penalty assessed against the staked principal.

Although the KAIZEN PROTOCOL supports trading with swapping and staking transactions as stand-alone investment options, the combination of the two offers an advantage to a token issuer to limit and minimize overselling of a new token offering. Two possible “issue, swap, stake” combinations are supported by KAIZEN.FINANCE as part of its comprehensive token launch platform, namely *swap-then-stake*, or *swap-and-stake*. Both options are beneficial to the investor by increasing their investment returns, and good for the token issuer by limiting the risk of early investor selling.

Specifically, in *swap-then-stake*, an investor first swaps crypto cash for enterprise tokens which are locked in a swap pool, delivering kTx tokens to the investor as a redeemable proof of purchase. After the locked enterprise tokens become vested, the enterprise tokens are automatically loaned (staked) to the DeFi pool for an additional staking term with no option for immediate redemption.

In details shown in **Figure 24A** for swap-then-stake purchases all the enterprise tokens (once vested) are automatically staked at a specified APY rate for a defined investment term. At the end of the lending term and upon redemption of the kTx tokens, the purchased enterprise tokens (the loan’s principal) are fully repaid plus the interest earned from staking.

Interest may be paid by the issuance of additional enterprise tokens, by crypto cash, or by another defined token asset. All terms specified in the offer’s description are recorded in the swap-then-stake smart contract. Redemption for principal and interest is verified by kTx tokens (shown by red dashed line) thereby preventing hackers or frauds from absconding the investment proceeds.

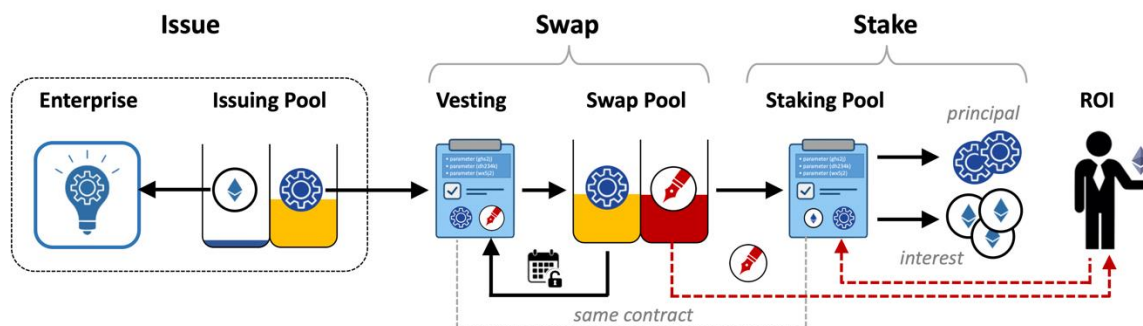


Fig. 24A: Swap-then-stake token issuance plus interest

Conversely, in *swap-and-stake*, an investment of crypto cash is bifurcated into two components – one portion to be swapped for enterprise tokens, and a second portion to be concurrently staked to earn interest. In this manner KAIZEN’s swap-and-stake pool represents a hybrid of token capital and income .

As shown in **Figure 24B**, issued enterprise tokens are swapped for kTx tokens and concurrently staked, with the vesting period for unlocking the enterprise tokens having a different schedule than the staking pool’s investment term. At the end of the investment period, the kTx tokens are used to claim the interest and principal from the staking pool and the vested enterprise tokens from the swap pool.

Interest may be paid by the issuance of additional enterprise tokens, by crypto cash, or by another defined token asset. All terms specified in the offer’s description are recorded in the swap-and-stake smart contract. Redemption for vested tokens, staking principal, and earned interest is verified by swapping of kTx tokens (shown by red dashed lines, one for the swap pool, another for the staking pool) thereby preventing hackers or frauds from absconding the investment proceeds.

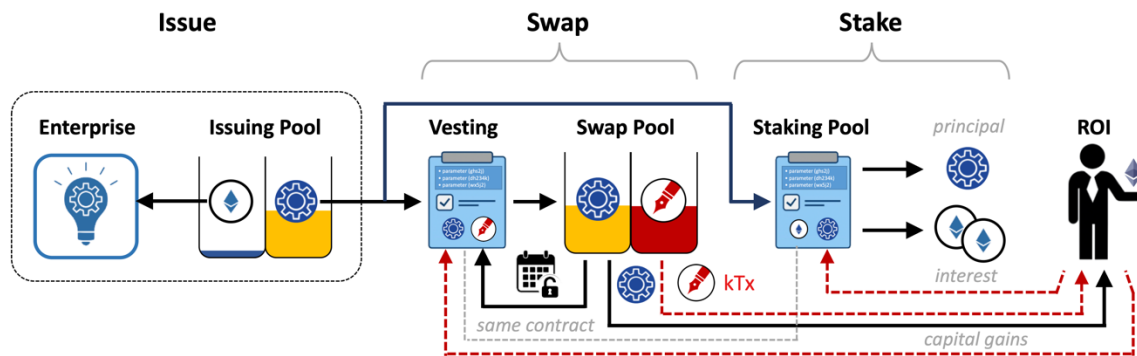


Fig. 24B: Swap-and-stake token issuance plus interest

Other Advantages of kTx Tokens: Aside from the foregoing, other benefits of KAIZEN’s unique kTx tokens is they may be hypothecated even when their underlying asset remains locked. The kTx token allows enterprise tokens to monetize their investment sooner than a vesting schedule or staking period would otherwise allow.

Trading of unlocked kTx tokens backed by locked enterprise tokens as collateral may entail (i) selling the kTx token asset at a perceived value (as a future), staking the kTx token to earn interest, or (iii) using the kTx as collateral for a DeFi loan. Please refer to the Collateralized Token Trading section to follow for more details in the trading options for kTx tokens.

KAIZEN.FINANCE Investor UX

Executing a successful token launch takes more than dApps for generating and distributing tokens. An equally important element to offering and selling an enterprise token is attracting buyers. Aside from marketing and social media, investors must feel confident that they can trust the offering's host platform and the transactional protocol used to perform trades and protect their assets. Not unexpectedly, investor user experience (UX) is important to smoothly transacting numerous trades required in the launch process including private sales, presales, IDOs, and the token generation event. Post-TGE support requires vesting, staking, token redemption, and collateralized token trading.

KAIZEN's superior investor user experience is based on proven dApp tools combined with technological advances in transactional security pioneered and patented by the HYPERSPHERE, including

- KAIZEN UI/UX facilitating intuitive operation for superior quality token transactions
- KAIZEN PROTOCOL for reliable autonomous smart contract authoring
- KAIZEN AI ORACLE real time monitoring of market pricing for accurate rational trading
- CYBERWALLET secure decentralized non-custodial wallet protecting assets
- HYPERID blockchain based identity validation protecting assets and transactional privacy
- Multi-blockchain transactions for cost efficiency with flexible cross chain benefits
- Multi-crypto pools accepting popular cryptocurrency and stablecoins in a single DeFi pool

The interrelationship of KAIZEN.FINANCE features depicted in **Figure 25** highlights that a trader can gain access to KAIZEN's entire financial ecosphere of opportunity through a single interface, the KAIZEN UI/UX, conveniently opened through any HYPERID secured wallet (such as the HYPERSPHERE's unique CYBERWALLET).

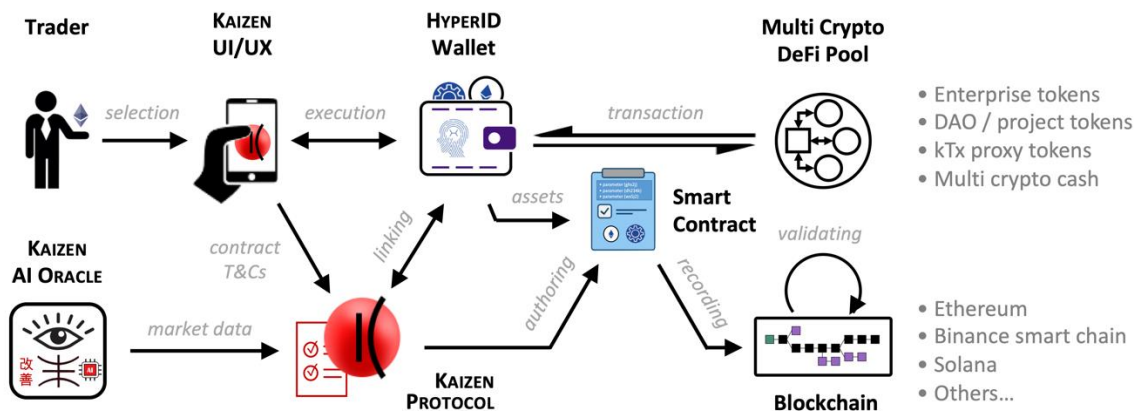


Fig. 25: KAIZEN.FINANCE tool suite flexibly supporting token trading and launches

As the control center for DeFi transactions, KAIZEN UI/UX converts a trader’s instructions into transactional terms and conditions (T&Cs) then passes them to the KAIZEN PROTOCOL for autonomous smart contract authoring. During the contract authoring process, KAIZEN gathers relevant real time market data (such as the current price of ETH or BNB) via its decentralized artificial-intelligence based oracle, the KAIZEN AI ORACLE, and updates all data fields accordingly. By checking current market conditions and trading at the moment of a pending transaction, KAIZEN is able to mitigate investors’ and issuers’ risks from arbitrage trading and front running losses.

The smart contract, once executed, facilitates an autonomous transaction between the trader’s CYBERWALLET and any designated DeFi pool including KAIZEN’s unique multi-crypto DeFi trading pools supporting multiple denominations of crypto cash commingled with a new offering’s issued tokens and kTx tokens as proxies thereof. Issued tokens may comprise enterprise tokens, DAO offerings, or project specific offerings. After its initial launch, each new trade updates the smart contract and Merkle tree with the newest tokenomic trading data, and uploads it to the blockchain for validation.

Agnostic to its BCVM host, the KAIZEN PROTOCOL supports multi-chain and cross-chain transactions on a wide and ever expanding range of networks including the Ethereum blockchain, Binance smartchain, Solana and others. The KAIZEN PROTOCOL is forward compatible with HYPERSPHERE’s ultra-fast high privacy DyDAG architecture, when the network is launched.

Multi-crypto DeFi Pools: Shown in **Figure 26**, as a special benefit to KAIZEN hosted token launches investors are able to purchase either issued enterprise tokens or kTx token proxies via a unique multi-crypto pool, a single smart contract and DeFi pool able to accept payment (i.e. transact swaps) using a variety of cryptocurrencies or stablecoins including Dai, USDC, USD₯, BUSD, and Ether wrap (WETH). Rather than requiring separate token pair pools for every payment option, e.g. USDC-kTx, WETH-kTx, BUSD-kTx, etc., the multi-crypto pool can trade use any listed crypto cash to swap for the offered token, either the enterprise token or its kTx proxy.

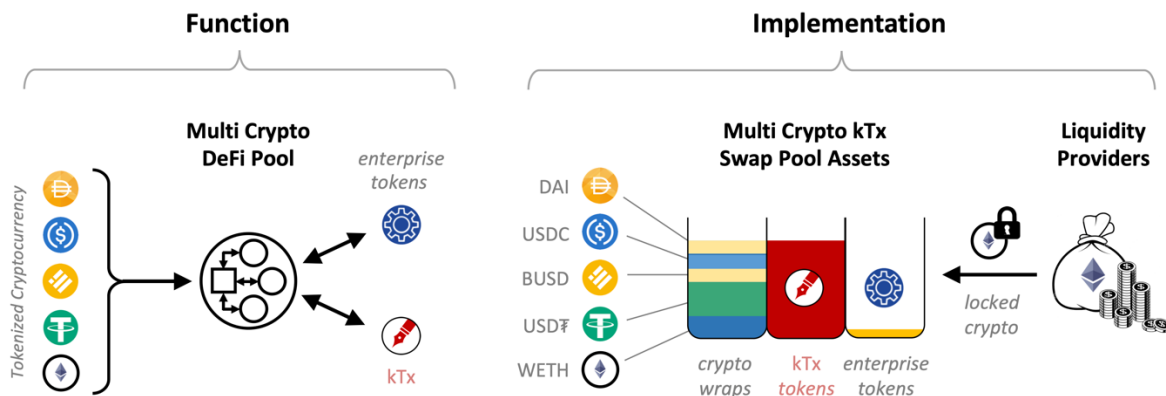


Fig. 26: KAIZEN PROTOCOL enabled multiple DeFi crypto pools enables superior token launch trading

During token launch, as a selling pool for distributing the new issuance, total liquidity requirements for the pool is minimal at least until issued tokens become unlocked and tradable. The liquidity is generally supplied not by a commercial AMM (professional autonomous market maker) by allocating a portion of the offering’s proceeds and generated tokens to the pool. Note that a multi-crypto DeFi pool facilitates swapping crypto cash for enterprise or kTx tokens but not to trade one cash denomination to another. For example in the pool, Dai cannot be traded for USD\$. The pool can, however facilitate swapping between kTx and issued enterprise tokens as per vesting schedules defined in an offering’s tokenomics.

Although the KAIZEN PROTOCOL can also support conventional trading pools comprising publicly traded assets backed by commercial liquidity providers, the subject of transactional trading pools is beyond the scope of this whitepaper. For details regarding the use of KAIZEN PROTOCOL for trading pools comprising swaps, futures, index tokens, yield framing, lending and collateralized borrowing, and synthetic trading please consult a related whitepaper entitled “The KAIZEN PROTOCOL: A Decentralized Application with AI-Based Oracle for Rational DeFi Transactions.” As this whitepaper is directed to token launches, trading pools will not be considered further here.

The Importance of the KAIZEN AI ORACLE: While a majority of protocols do not employ oracles, their role in providing trustworthy trading and rational pricing is critical. Without an oracle, a pool and its token holders can fall prey to arbitrage trading, a method where price incongruities among various DeFi pools, decentralized and centralized exchanges are exploited by traders seeking to profit from buying in one market at slightly lower price and concurrently selling the same asset in another market, stripping all the wealth out of a pool one trade at a time. The arbitrageur make a profit simply by moving money without adding any value to crypto-economy they are exploiting. **Figure 27** illustrates how KAIZEN uses its own on-chain KAIZEN AI ORACLE to combat arbitrage exploits by ensuring it is trading assets at their fair market value (FMV) at all times.

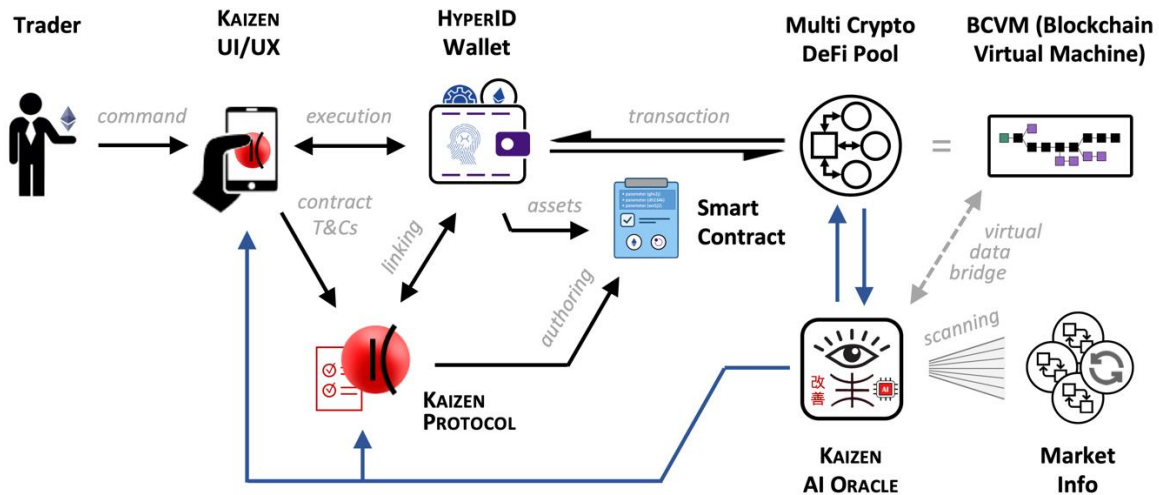


Fig. 27: Using KAIZEN AI ORACLE to ensure real time FMV transactional pricing

In every trade, the KAIZEN AI ORACLE checks the pending transaction details and collects relevant pricing from the market and from the blockchain searching for incongruities in FMV. The resulting data is fed back to both the KAIZEN PROTOCOL and to investor via KAIZEN UI/UX for prudent decision making.

HYPERSPHERE Enabled Security & Privacy: Another key benefit to users participating in a KAIZEN.FINANCE offering is the security and privacy protections made available using the HYPERSPHERE’s patented technology and secure communication methods. Far beyond the simple two-factor authentication used by banks and digital currency exchanges, HYPERID produces a *digital fingerprint* for a user’s identity requiring account information, device ownership, identity validation, and multifactor authentication. If any one of the components are lacking access is not granted. Additional security is provided by linking the HYPERID validated identity to the person’s CYBERWALLET using cryptographic credentials stored on the blockchain. Evening cloning a device will not grant unauthorized access to an owner’s assets or trading capabilities.

As shown in **Figure 28**, user access their CYBERWALLET and KAIZEN dApp feature suite through a login to their HYPERID account after which they can open their wallet’s dApp menu to launch the KAIZEN PROTOCOL and commence with trading or participating in a token offering, either purchasing (swapping), vesting (unlocking), redeeming proxies, selling (swapping), or lending (staking) via multi-crypto DeFi pool. In each case, the user instructs KAIZEN UIUX by commands made on the dApp’s graphical interface, which (i) links the pending transaction to assets stored in the CYBERWALLET, (ii) loads the trade’s T&Cs into KAIZEN PROTOCOL to prepare the smart contract, (iii) checks the FMV of assets associated with the pending trade, and (iv) communicates the details of the pending trade back to user for confirmation. During the UIUX dialog, the KAIZEN user and asset owner must confirm what denomination of crypto cash they intend to pay or receive, specify the amount of tokens they wish to transact, the requested speed in which the transaction will be processed (turtle, car, airplane).

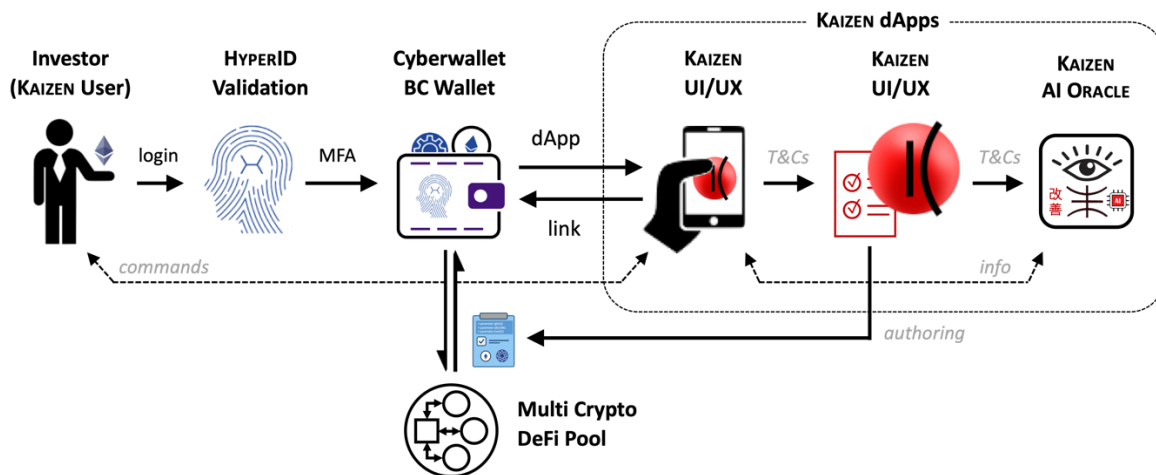


Fig. 28: Using HYPERID to access CYBERWALLET and the KAIZEN dApp feature suite

The user may be requested to enter a discount or access code without which may cost more in fees, not issue bonus tokens, or not be allowed to transact. KAIZEN UI/UX will then calculate the gas fees for the transaction and check the linked CYBERWALLET to confirm the user has adequate funds or tokens for the transaction and adequate crypto cash to pay the gas fees. If the funds are lacking the user will be informed that the transaction cannot be completed.

In some transaction, for example those involving payment using USD₯, the transaction will be performed in two steps, with the user required to approve each step in order to complete the trade. This feature is not a requirement of KAIZEN UI/UX but an artifact of the stablecoin's implementation. The KAIZEN PROTOCOL supports two step transactions with confirmations after each step reported in by KAIZEN UI/UX.

The KAIZEN UI/UX is also able to support multi-chain transactions, for example to transact trades on two or more chains from the same wallet and session provided the issuer support their token launch on more than a single chain. Note that although the KAIZEN PROTOCOL is network agnostic, the tradable assets and gas expense are not interchangeable. Ethereum EVM based trades require ERC-20 compliant tokens for trading and ETH for gas, while Binance smartchain trading required BEP-20 tokens with BNB needed for gas. So for example a BEP-20 compliant USD₯ stablecoin cannot be substituted for a ERC-20 USD₯ stablecoin in an EVM network transaction, even though both are USD₯ tokens.

How KAIZEN Distributes Pre-TGE Tokens: Trading support for a new token launch starts well before a TGE with an enterprise being able to successfully distribute tokens in private tranches and presales as illustrated in **Figure 29**, a task best managed by sale of pre-TGE kTx tokens.

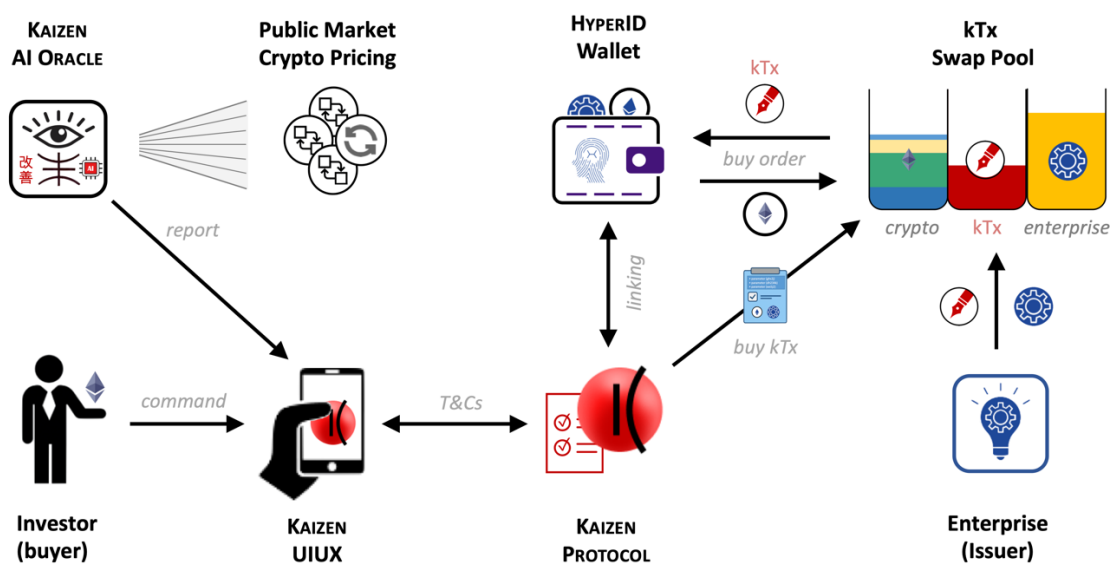


Fig. 29: Using KAIZEN to distribute a new token offering to buyers pre-TGE

During the pre-TGE period, the enterprise issuer and its lead investors negotiate the offerings tokenomics comprising various tranches, each with its own vesting schedule and purchase price. The tranches are then allocated to various investors and if desired legally documented in a term sheet or a SAFT (Simple Agreement for Future Tokens). The tokenomic model may also include a portion dedicated for presales and IDOs planned for the days immediately preceding the TGE. Prior to launching any smart contract, investors in these pre-TGE events may confirm their investment by their sending wallet address and transfers recorded on the blockchain. Alternatively, a separate smart contract can be executed to issue uncollateralized kTx tokens in advance of TGE.

Upon TGE the enterprise adds liquidity to the kTx swap pool as enterprise tokens (which are mostly locked at TGE) plus an equal quantity of kTx tokens are transferred into the swap pool. Buyers, using crypto cash valued by KAIZEN AI ORACLE at the spot price at time place a buy order at the tranche offering price converting the crypto cash into kTx tokens. The kTx tokens are thereby collateralized by enterprise tokens with a defined vesting schedule built into the smart contract. Vesting and automatic unlocking commences upon TGE.

How KAIZEN Redeems kTx Tokens: As shown in **Figure 30**, tokens once vested may be redeemed for enterprise tokens at the prescribed swap ratio, typically 1:1. The redeemed enterprise tokens transferred into the investor’s CYBERWALLET may then either hold them or concurrently sell them on an exchange or through an AMM depending on the terms of the investment agreement or SAFT.

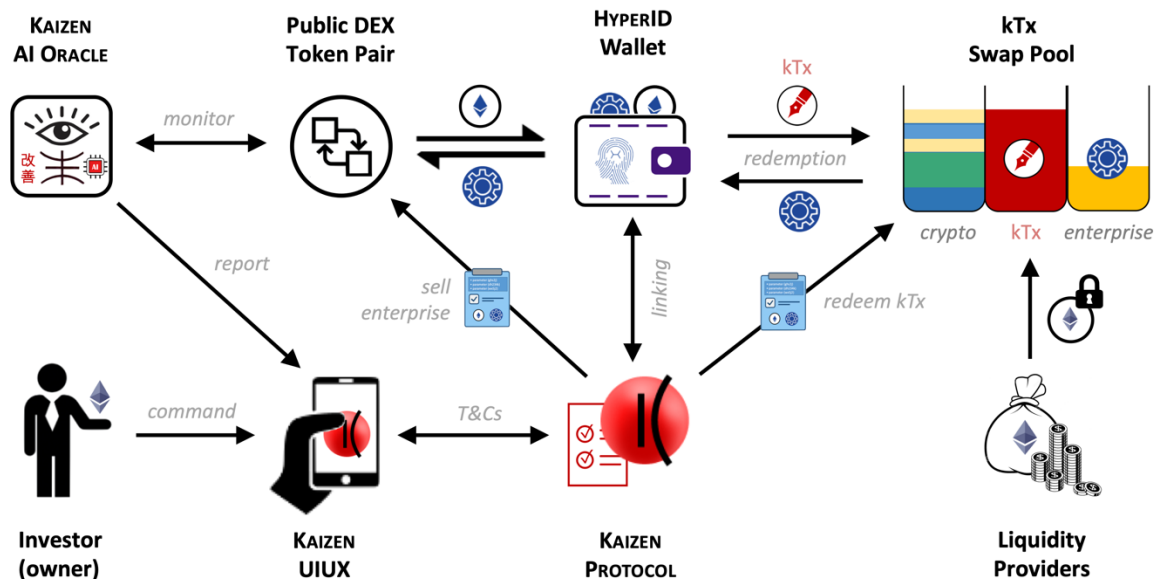


Fig. 30: Using KAIZEN PROTOCOL to redeem kTx tokens for enterprise tokens or sell for crypto cash

In this manner, the kTx token can only be used to claim and sell enterprise tokens once they vested. So even though the enterprise token is listed and publicly tradable, the private purchasers are unable to sell any significant quantities thereby protecting the public investors from rug pulls.

The other means by which an investor may hypothecate a newly unlocked enterprise token without adversely impacting the exchange trading price is by concurrently staking the enterprise tokens for interest income pursuant to the irrevocable lending term of the staking pool, prohibiting early withdrawal.

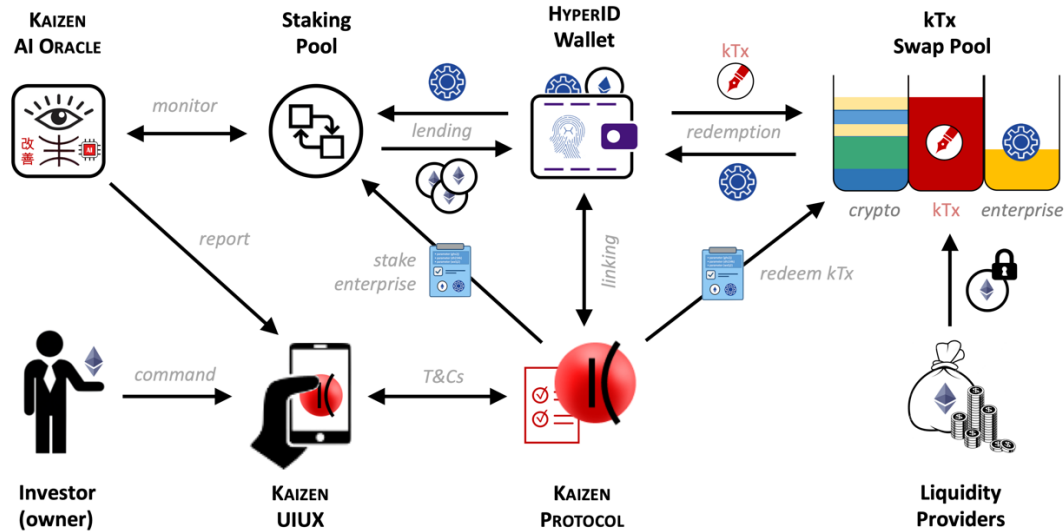


Fig. 31: Using KAIZEN PROTOCOL to redeem kTx tokens and stake enterprise tokens for interest

kDEX Collateralized Token Trading

Although an enterprise token once vested, can be held, sold, or staked as an investor chooses, a unique feature of kTx tokens is their ability to be hypothecated while their underlying collateral remains locked. kTx token trading prior to unlocking their collateral is shown in **Figure 32**.

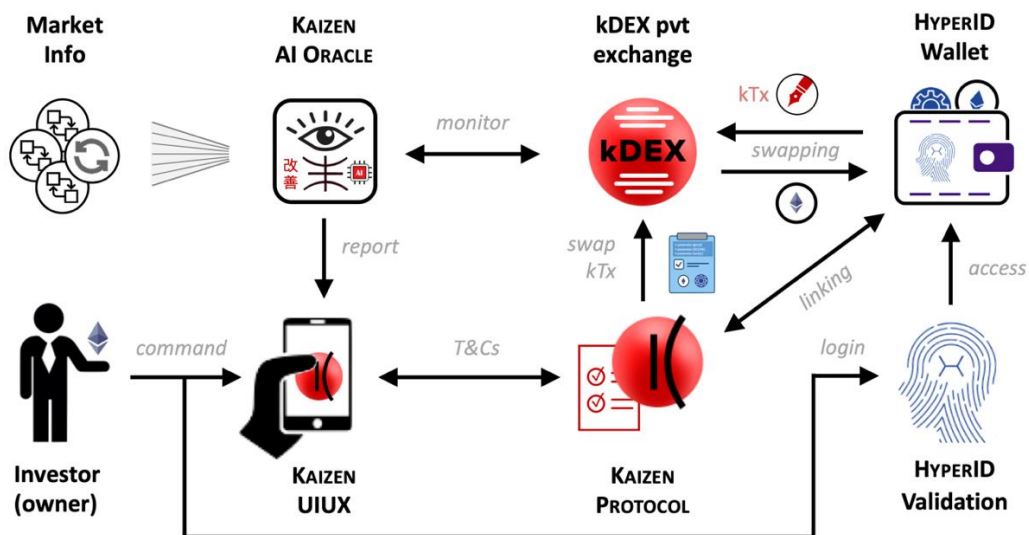


Fig. 32: Using KAIZEN to trade kTx tokens prior to collateral unlocking

In this manner a kTx holds the value of the enterprise token that collateralizes it. Buyers of the kTx token benefit from the speculative value of the enterprise token at a future date when it unlocks. Purchase of the kTx token generally occurs at a discount to provide the benefit of liquidity to its seller and greater profit potential to its buyer. Trading occurs in a special decentralized exchange called a kDEX. The same concept can be extended to staking or collateralized borrowing as shown in **Figure 33**, using a kTx token as the asset locked in the kDEX exchange.

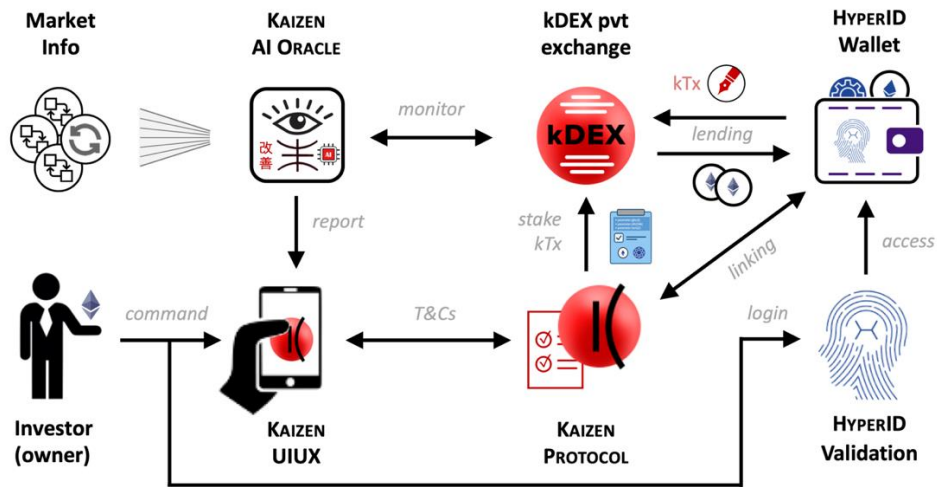


Fig. 33: Using KAIZEN to stake kTx tokens prior to collateral unlocking

KAIZEN.FINANCE Token Launch Lifecycle Management

As shown in **Figure 34**, the role of KAIZEN.FINANCE and its proprietary KAIZEN PROTOCOL is to facilitate a launch platform for tokens capable of supporting the entire lifecycle of token issuance, vesting, unlocking, trading and staking. No other protocol or DeFi platform is capable of such launchpad features and service or delivering the extensive DeFi tool suite that KAIZEN can.

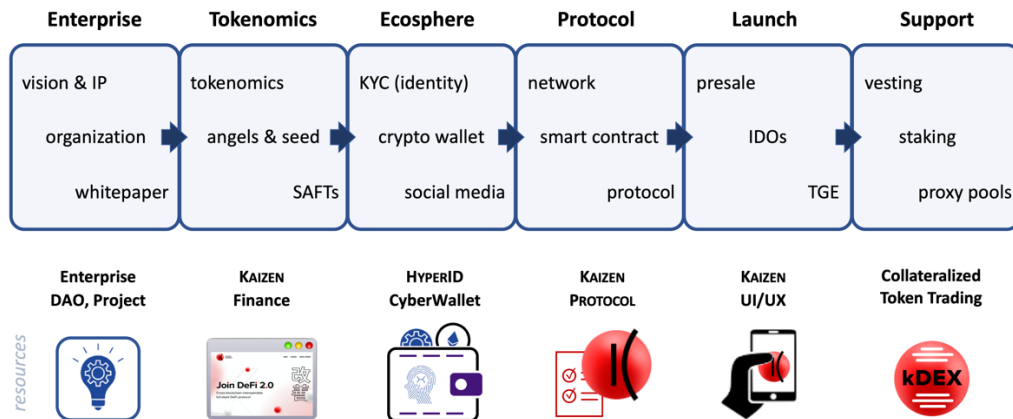


Fig. 34: KAIZEN.FINANCE Token Lifecycle Management

Literature Cited

- [1] “Crypto economy: a new era for the financial markets,” *Medium (NovaMining)*, 2017 Dec 12.
[online] <https://medium.com/@NovaMining/crypto-economy-a-new-era-for-the-financial-markets-573b99a8d17f>
- [2] D. Bryan, A. Virtanen, “What is a crypto economy?” *Medium*, 2018 May 14.
[online] <https://medium.com/econaut/what-is-a-crypto-economy-155bdbc4ab1d>
- [3] “Blockchain. What is blockchain? How does it work?” *Built In*, © 2021
[online] <https://builtin.com/blockchain>
- [4] L. Conway, “Blockchain explained,” *Investopedia*, 2021 May 31.
[online] <https://www.investopedia.com/terms/b/blockchain.asp>
- [5] “Definition: cryptocosm,” *Computer Language*, 2021.
[online] <https://www.computerlanguage.com/results.php?definition=cryptosphere>
- [6] N. Reiff, “Decentralized autonomous organization (DAO),” *Investopedia*, 2021 Sept 24.
[online] <https://www.investopedia.com/tech/what-dao/>
- [7] R. Reshwan, “What to know about the gig economy,” *US News*, 2021 Jul 23.
[online] <https://money.usnews.com/money/blogs/outside-voices-careers/articles/what-is-the-gig-economy>
- [8] M. Hussey, “What are token generation events (TGEs)?” *Decrypt* 2019 Jan 22.
[online] <https://decrypt.co/resources/token-generation-events-what-are-they-guide>
- [9] “A brief history of the Internet,” *Online Lib Learning Center (U Ga)*.
[online] https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml
- [10] A. Powell, “Web 101 – a history of the GUI,” *Wired*, 1997 Dec 19.
[online] <https://www.wired.com/1997/12/web-101-a-history-of-the-gui/>
- [11] “Orthogonal frequency-division multiplexing,” *Wikipedia*.
[online] https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing
- [12] “History of bitcoin,” *Wikipedia*.
[online] https://en.wikipedia.org/wiki/History_of_bitcoin#cite_note-NY2011-18

- [13] G. Clydesdale, “Chapter 4 – Technology, new products and pioneers,” in *Entrepreneurial Opportunity: The Right Place at the Right Time*. Rutledge © 2010:57-59 [online] https://www.researchgate.net/publication/286519077_Entrepreneurial_opportunity_The_right_place_at_the_right_time
- [14] N. Dailey, “Vitalik Buterin says he created Ethereum after his beloved World of Warcraft character was hobbled by the developers, awakening him to the 'horrors centralized services can bring',” *Markets Insider* 2021 Oct 5. [online] <https://markets.businessinsider.com/news/currencies/vitalik-buterin-created-ethereum-following-world-of-warcraft-debacle-2021-10>
- [15] “Ethereum virtual machine,” *Ethereum.org*, 2021 Sep 29. [online] <https://ethereum.org/en/developers/docs/evm/>
- [16] “Smart contract,” *Wikipedia*. [online] https://en.wikipedia.org/wiki/Smart_contract
- [17] “What are smart contracts on blockchain?” *IBM*. [online] <https://www.ibm.com/topics/smart-contracts>
- [18] J. Frankenfield, “Smart contracts,” *Investopedia*. 2021 May 26. [online] <https://www.investopedia.com/terms/s/smart-contracts.asp>
- [19] W. Vermaak, “What is Binance smart chain?” *CoinMarketCap*, © 2021. [online] <https://coinmarketcap.com/alexandria/article/what-is-binance-smart-chain>
- [20] “HECO chain,” *Hecochain.com*, updated 2021 Oct. [online] <https://www.hecochain.com/en-us/>
- [21] “Solana,” *Solano.com*. [online] <https://solana.com>
- [22] “Polkadot,” *Polkadot*. [online] <https://polkadot.network/technology/>
- [23] E. Verzun, R.K. Williams, “The HYPERSPHERE – a real-time cybersecure privacy network with embedded DyDAG blockchain for global e-commerce,” *Hypersphere.ai*, 2018 Jul 31. [online] <https://hypersphere.ai/assets/downloads/HyperSphere%20technical%20whitepaper%20v1.8.0.pdf>
- [24] “Digital assets: cryptocurrencies vs. tokens,” *Cryptopedia*, 2021 May 17. [online] <https://www.gemini.com/cryptopedia/cryptocurrencies-vs-tokens-difference>
- [25] “How blockchain could disrupt banking,” *CBInsights*, 2021 Feb 11. [online] <https://www.cbinsights.com/research/blockchain-disrupting-banking/>

- [26] K. Dwyer, "Will cryptocurrencies and blockchain replace banking and finance?" *coinmarketcap.com*, 2021 May. [online] <https://coinmarketcap.com/alexandria/article/will-cryptocurrencies-and-blockchain-replace-banking-and-finance>
- [27] P. Likos, "How blockchain can transform the financial services industry," *US News & World Report*, 2021, Sep 3.
[online] <https://money.usnews.com/investing/cryptocurrency/articles/how-blockchain-can-transform-the-financial-services-industry>
- [28] G. Iredale, "6 key blockchain features you need to know now," *101 Blockchains*, 2020 Nov 24. [online] <https://101blockchains.com/introduction-to-blockchain-features/>
- [29] "Distributed ledger," *Wikipedia*. [online] https://en.wikipedia.org/wiki/Distributed_ledger
- [30] "What are distributed ledger technologies (DLTs)?" *Hedera.com*.
[online] <https://hedera.com/learning/what-are-distributed-ledger-technologies-dlts>
- [31] "Difference blockchain and DLT," *Marco Polo Network*, 2018 Jan 1.
[online] <https://www.marcopolonetwork.com/articles/distributed-ledger-technology/>
- [32] A. Singh, "Distributed ledger vs blockchain technology: do you know the difference?" *Medium (Brandlitic)*, 2021 Jun 5. [online] <https://medium.com/brandlitic/difference-between-distributed-ledger-and-blockchain-vs-dlt-7969f3837ded>
- [33] J. Frankenfield, "What is a cryptocurrency block header?" *Investopedia*, 2021 Sep 22.
[online] <https://www.investopedia.com/terms/b/block-header-cryptocurrency.asp>
- [34] I. Weber et al, "A platform architecture for multi-tenant blockchain-based systems," *IEEE Intl Conf on Software Arch (ICSA2019, Hamburg, Germany)*, 2019 Mar.
[online] https://www.researchgate.net/publication/330752524_A_Platform_Architecture_for_Multi-Tenant_Blockchain-Based_Systems
- [35] "Cryptography," *Wikipedia*. [online] <https://en.wikipedia.org/wiki/Cryptography>
- [36] "Cryptography," *Britannica*. [online] <https://www.britannica.com/topic/cryptology/Cryptography>
- [37] "Articles on cryptography and cryptoanalysis," *War Dept. (US Gov)*, #A484988, declassified 1950 May 5.
[online] https://www.nsa.gov/Portals/75/documents/news-features/decclassified-documents/friedman-documents/publications/FOLDER_205/41762559080174.pdf

- [38] “Public keys and private keys in public key cryptography,” *Ectigo*, 2020 Jun 9. [online] <https://sectigo.com/resource-library/public-key-vs-private-key>
- [39] O. Pal et al, “Key management for blockchain technology,” *Science Direct (ICT Express)*, 2021, 7:76-80. [online] <https://www.sciencedirect.com/science/article/pii/S2405959519301894>
- [40] K. Brush et al, “Asymmetric cryptography (public key cryptography),” *Tech Target (Search Security)*, © 2021. [online] <https://searchsecurity.techtarget.com/definition/asymmetric-cryptography>
- [41] “Public key infrastructure (PKI) will soon run on blockchain technology,” *Nexus*, 2017, Mar 17. [online] <https://www.nexusgroup.com/public-key-infrastructure-pki-blockchain-technology/>
- [42] D. Yaga et al, “Blockchain technology overview,” *Natl Inst Tech & Standards – NISTR 8202*, 2018 Oct. [online] <https://nvlpubs.nist.gov/nistpubs/ir/2018/nist.ir.8202.pdf>
- [43] “Permissioned vs permissionless blockchains,” *101 Blockchains*, 2020 May 28. [online] <https://101blockchains.com/permissioned-vs-permissionless-blockchains/>
- [44] “What are cryptographic hash functions?” *Synopsis*, 2015 Dec 10. [online] <https://www.synopsys.com/blogs/software-security/cryptographic-hash-functions/>
- [45] S. Wiesner, “Blockchain: how tamper-proofing actually works,” *Medium*, 2019 Jan 28. [online] <https://medium.com/efficient-frontier/blockchains-explained-in-less-than-1000-words-7b54f16135a6>
- [46] “Hash chain,” *Wikipedia*. [online] https://en.wikipedia.org/wiki/Hash_chain
- [47] R. Parashar, “What is the hash chain ?” *LinkedIn*, 2021 Aug 24. [online] https://www.linkedin.com/pulse/what-hash-chain-rajesh-parashar?trk=public_profile_article_view
- [48] S. Ray, “Merkle trees,” *Hackernoon*, 2017 Dec 14. [online] <https://hackernoon.com/merkle-trees-181cb4bc30b4>
- [49] J. Frankenfield, “Merkle tree,” *Investopedia*, 2021 Jul 26. [online] <https://www.investopedia.com/terms/m/merkle-tree.asp>
- [50] V. Chawla, “What are the top blockchain consensus algorithms?” *Analytics India Mag*, 2020 Jan 1 [online] <https://analyticsindiamag.com/blockchain-consensus-algorithms/>
- [51] “What is a cryptographic nonce?” *Security Encyclopedia, HYPR Corp* © 2021. [online] <https://www.hypr.com/nonce/>

- [52] B. Marshall, “How are transactions validated?” *Medium*, 2018 Feb 1. [online] <https://medium.com/@blairmarshall/how-do-miners-validate-transactions-c01b05f36231>
- [53] J. Dorian, “Is it possible to provide a simple example and break down a mathematical puzzle bitcoin miners solve right here on Quora so we can see it under the hood?” *Quora*, 2019. [online] <https://www.quora.com/Is-it-possible-to-provide-a-simple-example-and-break-down-a-mathematical-puzzle-bitcoin-miners-solve-right-here-on-Quora-so-we-can-see-it-under-the-hood>
- [54] D. Drescher, “Blockchain basics” © Apress. 2017 Mar 16. [online] <http://www.blockchain-basics.com/index.html#AboutBook> [online puzzle] <http://www.blockchain-basics.com/HashPuzzle.html>
- [55] N. Akhtar, “Cryptocurrency mining: proof-of-work consensus,” *Blockchain at Berkeley*. [online] <http://www.cse.psu.edu/~trj1/cse543-f17/slides/cse543-blockchains.pdf>
- [56] G.A. Marañón, “A trillion-dollar on offer to the puzzle solver,” *Telefónica Tech*, 2021 Apr 12. [online] <https://business.blogthinkbig.com/trillion-dollar-offer-puzzle-solver/>
- [57] N. Truong et al, “A blockchain-based trust system for decentralised applications: When trustless needs trust,” **124**. 2021 Nov: 68-79. [online] <https://www.sciencedirect.com/science/article/pii/S0167739X21001758?via%3Dihub>
- [58] “What do we mean by “blockchains are trustless?”” *Preethi Kasireddy*, 2018 Feb 10 [online] <https://www.preethikasireddy.com/post/what-do-we-mean-by-blockchains-are-trustless>
- [59] D. Conte de Leon, “Blockchain: properties and misconceptions,” *Asia Pacific J Innovation and Entrepreneurship*, **11**(3). 2017 Sep 27:286-300. [online] https://www.researchgate.net/publication/321811785_Blockchain_properties_and_misconceptions
- [60] K. Doubleday, “Blockchain immutability – why does it matter?” *Medium (Fluree)*, 2018, Nov 27. [online] <https://medium.com/fluree/immutability-and-the-enterprise-an-immense-value-proposition-98cd3bf900b1>
- [61] “Crypto economy: a new era for the financial markets,” *Medium (NovaMining)*, 2018 Dec 5. [online] <https://medium.com/@NovaMining/crypto-economy-a-new-era-for-the-financial-markets-573b99a8d17f>
- [62] B. Felter, “What is network redundancy and why does it matter?” *Vxchnge*, 2020 Dec 17. [online] <https://www.vxchnge.com/blog/network-redundancy-explained>
- [63] V. Joshi, “Redundancy and replication: duplicating in a distributed system,” *Medium (Base)*, 2019 Aug 14. [online] <https://medium.com/baseds/redundancy-and-replication-duplicating-in-a-distributed-system-7ab4322d7378>

- [64] Z. Zheng et al, “An overview of blockchain technology: architecture, consensus, and future trends,” *6th Intl Conf Big Data*, 2017 Jun . **85**:557-564. [online] https://www.researchgate.net/publication/318131748_An_Overview_of_Blockchain_Technology_Architecture_Consensus_and_Future_Trends
- [65] “Blockchain attacks,” *Freeman Law*, © 2021. [online] <https://freemanlaw.com/blockchain-attacks-is-no-one-safe-in-the-world-of-cryptocurrencies/>
- [66] “10 blockchain and new age security attacks you should know,” *Aruba Marketing (HP)*, 2019 Jan 23. [online] <https://blogs.arubanetworks.com/solutions/10-blockchain-and-new-age-security-attacks-you-should-know/>
- [67] A. Gazdecki, “Proof-of-work and proof-of-stake: how blockchain reaches consensus,” *Forbes*, 2019 Jan 28. [online] <https://www.forbes.com/sites/forbestechcouncil/2019/01/28/proof-of-work-and-proof-of-stake-how-blockchain-reaches-consensus/?sh=53ef41368c82>
- [68] “Proof of stake,” *Wikipedia*. [online] https://en.wikipedia.org/wiki/Proof_of_stake
- [69] R. Cho, “Bitcoin’s impacts on climate and the environment,” *State of the Planet – Columbia Climate School*, 2021 Sep 20. [online] <https://news.climate.columbia.edu/2021/09/20/bitcoins-impacts-on-climate-and-the-environment/>
- [70] J. Calma, “The climate controversy swirling around NFTs,” *The Verge*, 2021 Mar 15. [online] <https://www.theverge.com/2021/3/15/22328203/nft-cryptoart-ethereum-blockchain-climate-change>
- [71] A. Yakovenko “How Solana’s proof of history is a huge advancement for block time,” *Medium (Solana)*, 2019 Aug 14. [online] <https://medium.com/solana-labs/how-solanas-proof-of-history-is-a-huge-advancement-for-block-time-178899c89723>
- [72] R. Zhang and R. Xue, “Security and privacy on blockchain,” *Chinese Academy of Sciences*, 2019, Aug 16. [online] <https://arxiv.org/pdf/1903.07602.pdf>
- [73] D. Sanz-Bas et al, “Cryptocurrencies and fraudulent transactions: risks, practices, and legislation for their prevention in Europe and Spain,” *Laws* 2021 Jul 9, **10** (57):1-20. [online] <https://doi.org/10.3390/laws10030057>
- [74] J. Hayes, “Defending blockchains from network attacks,” *Black Ridge Technologies*, 2017 Dec 15. [online] <https://www.blackridge.us/blog/defending-blockchains-from-network-attacks>

- [75] “How secure are smart contracts?” *Gdata*, 2020 Dec 17. [online] <https://www.gdatasoftware.com/blog/2020/12/36570-how-secure-are-smart-contracts>
- [76] S. Sayeed et al, “Smart contract: attacks and protections,” *IEEE Access*, 2020 Jan (99):1-1. [online] https://www.researchgate.net/publication/338926064_Smart_Contract_Attacks_and_Protections
- [77] “DeFi education series: how to protect against smart contract hacks,” *Medium (InsurAce.io Protocol)*, 2021 Jun 1. [online] <https://blog.insurace.io/defi-education-series-how-to-protect-against-smart-contract-hacks-98ca6378f25c>
- [78] “Hacker-powered security and DeFi: how human intelligence improves cryptocurrency security” *Hackerone*, 2021 Aug 27. [online] <https://www.hackerone.com/ethical-hacker/hacker-powered-security-and-defi-how-human-intelligence-improves-cryptocurrency>
- [79] M. Scherer, “Performance and scalability of blockchain networks and smart contracts,” *UMEÅ Univ (theses)*, 2017. [online] <https://www.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf.10>
- [80] P.S. Maharjan, “Performance analysis of blockchain platforms,” *UNLV (theses)*, 2018. [online] <https://digitalscholarship.unlv.edu/cgi/viewcontent.cgi?article=4370&context=thesesdissertations>
- [81] M. Ohtamaa, “Scaling EVM (Ethereum virtual machine),” *Capital Gram*, 2021 Jul 12. [online] <https://capitalgram.com/posts/scaling-ethereum/>
- [82] R. Yang et al, “Empirically analyzing Ethereum’s gas mechanism,” *Univ Melbourne*, 2019 May 2. [online] <https://arxiv.org/pdf/1905.00553.pdf>
- [83] A. Rosic, “What is Ethereum gas? [the most comprehensive step-by-step guide ever!],” *Block Geeks*, 2020 Nov 2. [online] <https://blockgeeks.com/guides/ethereum-gas/>
- [84] C. Harper and C. Kim, “Ethereum gas limit hits 15M as ETH price soars,” *CoinDesk*, 2021 Sep 14. [online] <https://www.coindesk.com/tech/2021/04/22/ethereum-gas-limit-hits-15m-as-eth-price-soars/>
- [85] M. Ohtamaa “What is the Ethereum virtual machine (EVM) total computing power?” *Ethereum Stack Exchange*, 2021 Mar 15. [online] <https://ethereum.stackexchange.com/questions/97798/what-is-the-ethereum-virtual-machine-ethereum-vm-total-computing-power>
- [86] “IBM System/360,” *Wikipedia*. [online] https://en.wikipedia.org/wiki/IBM_System/360

- [87] R. Seeger, "Ethereum: rise of the world computer," *Silicon Valley Data Science*, 2016 Feb 23. [online] <https://www.svds.com/ethereum-the-rise-of-the-world-computer/>
- [88] "ES EVM" *Wikipedia*. [online] https://en.wikipedia.org/wiki/ES_EVM
- [89] T. Takenobu, "Ethereum EVM illustrated," *GitHub*, 2018 Mar.0.01.1 [online] https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf
- [90] A. Makarov, "Top 6 smart contract platforms: a deep dive," *itransition*, 2021 Apr 1. [online] <https://www.itransition.com/blog/smart-contract-platforms>
- [91] "Ouroboros: An environmentally sustainable, verifiably secure proof-of-stake protocol with rigorous security guarantees," *Cardano*. [online] <https://cardano.org/ouroboros/>
- [92] "Hyperledger," *Wikipedia*. [online] <https://en.wikipedia.org/wiki/Hyperledger>
- [93] "What is Tezos?" *Tezos*. [online] <https://tezos.com/learn/what-is-tezos/>
- [94] "Intro to Stellar," *Stellar*. [online] <https://www.stellar.org/learn/intro-to-stellar>
- [95] "Why EOS?" *EOSIO*. [online] <https://eos.io>
- [96] "What is IOTA," *Iota*. [online] <https://www.iota.org/get-started/what-is-iota>
- [97] "A comprehensive list of blockchain platforms," *TechnoDuet* [online] <https://www.technoduet.com/a-comprehensive-list-of-blockchain-platforms/>
- [98] "The best smart contract platforms," *Shrimpy Academy*. [online] <https://academy.shrimpy.io/post/the-best-smart-contract-platforms>
- [99] "Smart contracts platforms," *TrustRadius*. [online] <https://www.trustradius.com/smart-contracts>
- [100] "What are liquidity pools?" *Cryptopedia*, 2021 Sep 3. [online] <https://www.gemini.com/cryptopedia/what-is-a-liquidity-pool-crypto-market-liquidity>
- [101] R.K Williams and E. Verzun, "Scotty Beam – world's first cross-chain decentralized NFT teleport," *HyperSphere Press* © 2021. [online] <https://scottybeam.io>
- [102] A. Dolmatov, "Wrapped Ether (WETH)," *Binance Academy*, updated 2020. [online] <https://academy.binance.com/en/glossary/wrapped-ether>

- [103] T. Falk, “What is an ERC20 token?” *Finder*, updated 17 Dec 2020.
[online] <https://www.finder.com/erc20-tokens>
- [104] T. Tabrizi, “Ethereum token contract ABI in Web3.js for ERC-20 and human standard tokens,” *Shawn Tabrizi*, 8 Nov 2017. [online] <https://www.shawntabrizi.com/ethereum/ethereum-token-contract-abi-web3-erc-20-human-standard-tokens/>
- [105] A. Shome, “Illegal ICOs paid a quarter of SEC’s \$4.68 billion crackdown,” *Finance Magnates* 11 Dec 2020 [online] <https://www.financemagnates.com/cryptocurrency/news/illegal-icos-paid-a-quarter-of-secs-4-68-billion-crackdown/>
- [106] J.R. Silversmith, “United States: SEC settles charges involving ICO, CFTC takes action against crypto investment firm,” *Mondaq*, 25 Feb 2020.
[online] <https://www.mondaq.com/unitedstates/securities/897196/sec-settles-charges-involving-ico-cftc-takes-action-against-crypto-investment-firm>
- [107] “US Securities & exchange commission,” *US SEC* website.
[online] <https://www.sec.gov/news/pressreleases>
- [108] W. Vermaak, “What are dead coins?” *CoinMarketCap*, 21 Dec 2020.
[online] <https://coinmarketcap.com/alexandria/article/what-are-dead-coins>
- [109] D.L. Concannon, “The yellow brick road for consumer tokens: the path to SEC and CFTC compliance – an update,” in *Blockchain and Cryptocurrency Regulation 2020*, 2nd ed., Global Legal Group © 2020: 64-87. [online] <https://www.lw.com/thoughtLeadership/the-yellow-brick-road-for-consumer-tokens-path-to-sec-and-cftc-compliance-an-update>
- [110] J.J. Roberts, “Ripple says it will be sued by the SEC, in what the company calls a parting shot at the crypto industry,” *Fortune* 21 Dec 2020
[online] <https://fortune.com/2020/12/21/ripple-to-be-sued-by-sec-cryptocurrency-xrp/>
- [111] M. William, “ERC-20 tokens, explained,” *Cointelegraph*, 12 May 2018.
[online] <https://cointelegraph.com/explained/erc-20-tokens-explained>
- [112] P. Febrero, “A guide to Ethereum’s ERC standards,” *Coin Rivet*, 17 May 2019.
[online] <https://www.yahoo.com/now/guide-ethereum-erc-standards-150024381.html>
- [113] N. Reiff, “What Is ERC-20 and what does it mean for Ethereum?” *Investopedia*, 6 Feb 2020.
[online] <https://www.investopedia.com/news/what-erc20-and-what-does-it-mean-ethereum/>

- [114] R. Mitra, "Utility tokens vs security tokens: learn the difference - ultimate guide," *Blockgeeks*, 2021. [online] <https://blockgeeks.com/guides/utility-tokens-vs-security-tokens/>
- [115] R. Elder, "Blockchain tokens could transform the music industry," *Insider*, 8 Nov 2017. [online] <https://www.businessinsider.com/blockchain-could-transform-the-music-industry-2017-11>
- [116] M. Casiraghi, "Driving engagement through tokenization: gaming, sports, and entertainment," *Nasdaq*, 4 Jan 2021. [online] <https://www.nasdaq.com/articles/driving-engagement-through-tokenization%3A-gaming-sports-and-entertainment-2021-01-04>
- [117] "Privileged access management," *Secret Security Wiki*. [online] <https://doubleoctopus.com/security-wiki/authentication/privileged-access-management/>
- [118] A. Arriaga et. al, "Private functional encryption: indistinguishability-based definitions and constructions from obfuscation," *Univ. Lux*, 2016. [online] <https://eprint.iacr.org/2016/018.pdf>
- [119] "Indistinguishability," *The Free Dictionary*. [online] <https://www.thefreedictionary.com/indistinguishability>
- [120] "Crypto Airdrops," *DappRadar* 16 Nov 2020. [online] <https://dappradar.com/blog/explained-cryptocurrency-crypto-airdrops>
- [121] W. Vermaak, "What is a non-fungible token (NFT)?" *Alexandria*, 30 Jan 2021. [online] <https://coinmarketcap.com/alexandria/article/what-is-a-non-fungible-token-nft>
- [122] T. Kumar-Sharma, "Security tokens vs. utility tokens," a concise guide," *Blockchain Counsel*. [online] <https://www.blockchain-council.org/blockchain/security-tokens-vs-utility-tokens-a-concise-guide/>
- [123] S.W. Maughan, "Utility token offerings: can a security transform into a non-security?" 21 Aug 2020, 19(4): 1114-1146. [online] <https://digitalcommons.law.byu.edu/cgi/viewcontent.cgi?article=3232&context=lawreview>
- [124] A. Hurling, "Digital currency index and index tokens," *Medium*, 14 Oct 2018. [online] <https://medium.com/dex-top/digital-currency-index-and-index-tokens-3c699d5b8c45>
- [125] N. Kuznetsov, "Crypto synthetic assets, explained," *Cointelegraph*, 18 Mar 2020. [online] <https://cointelegraph.com/explained/crypto-synthetic-assets-explained>
- [126] "What is a governance token?" *CoinMarketCap*, 2021. [online] <https://coinmarketcap.com/alexandria/glossary/governance-token>

- [127] D. Uzsoki, "Tokenization of infrastructure: a blockchain-based solution to financing sustainable infrastructure," *Intl Inst for Sust Dev* (MAVA/IISD.org): 1-34 Jan 2019. [online] <https://www.iisd.org/system/files/publications/tokenization-infrastructure-blockchain-solution.pdf>
- [128] "Crowdfunding," *Wikipedia*. [online] <https://en.wikipedia.org/wiki/Crowdfunding>
- [129] B. Dickson, "Can you trust crypto-token crowdfunding?" *Techcrunch.com*, 12 Feb 2017. [online] <https://techcrunch.com/2017/02/12/can-you-trust-crypto-token-crowdfunding/>
- [130] T. McDonald, "The emergence of enterprise tokens," *Medium*, 2 Jul 2018. [online] <https://medium.com/corda/the-emergence-of-enterprise-tokens-99f6be65a3ba>
- [131] A. Cepka, "How to tokenize an existing business," *Medium*, 15 Apr 2018. [online] <https://medium.com/@acepka/how-to-tokenize-an-existing-business-17897fe3db8e>
- [132] A. Rosen, "How tokenization can democratize your business investments and transform your capital," *Business.com*, 18 Jun 2020. [online] <https://www.business.com/articles/how-tokenization-democratizes-investments/>
- [133] T.K. Sharma, "Best programming languages to build smart contracts," *Blockchain Counsel* © 2020 [online] <https://www.blockchain-council.org/blockchain/best-programming-languages-to-build-smart-contracts/>
- [134] G. Iredale, "Top 5 programming languages to build smart contracts," *101 Blockchains*, 2021 Aug 11. [online] <https://101blockchains.com/smart-contract-programming-languages/>
- [135] "Technical difference between Ethereum, Hyperledger fabric and R3 Corda," *Medium (Micobo GmbH)*, 2018 Mar 16. [online] <https://micobo.medium.com/technical-difference-between-ethereum-hyperledger-fabric-and-r3-corda-5a58d0a6e347>
- [136] L. Sanchez, "Blockchain layers explained: what are they and why do we need layer solutions?" *ZyCrypto*, 2021 Sep 1. [online] <https://zycrypto.com/blockchain-layers-explained-what-are-they-and-why-do-we-need-layer-solutions/>
- [137] T. Sobrado, "What are the six layers of Blockchain technology?" *ToddSobrado.com*, [online] <https://toddsobrado.com/what-are-the-six-layers-of-blockchain-technology/>
- [138] W. Vermaak, "What Are Application Layer Protocols?" *Alexandria (CoinMarketcap)*, 2021 Apr. [online] <https://coinmarketcap.com/alexandria/article/what-are-application-layer-protocols>

[139] R. Sharma, “Decentralized finance (DeFi) definition,” Investopedia, 2021 Mar 4.
[online] <https://www.investopedia.com/decentralized-finance-defi-5113835>

[140] “The blockchain trilemma: decentralized, scalable, and secure? *Medium*, 2019 Oct 4.
[online] <https://medium.com/certik/the-blockchain-trilemma-decentralized-scalable-and-secure-e9d8c41a87b3>

[141] “Clearing house (finance),” *Wikipedia*.
[online] [https://en.wikipedia.org/wiki/Clearing_house_\(finance\)](https://en.wikipedia.org/wiki/Clearing_house_(finance))