Vol. 2, No. 1, pp.: 25-44, 2024

Indexing Contractual Restrictions in NFT Markets: A Data Management Framework for Mechanism-Aware Digital Asset Trading

Aiman Hafiz Rahman[1], Siti Nur Aisyah Abdullah[2,*], Kelvin Lim Wei Jian[3]

[1] School of Computing, Universiti Malaysia Perlis, Arau 02600, Malaysia

[2] Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka 76100, Malaysia

[3] Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan 26600, Malaysia

** aisyah.abdullah@utem.edu.my*

Article Information	
Received	18 October 2023
Accepted	29 February 2024
DOI	https://doi.org/10.63646/datamind.2024.020103

Abstract

Non-fungible token (NFT) markets increasingly trade assets that are governed by contractual restrictions rather than by simple ownership transfer alone. A token may carry a vesting period, a staking obligation, a creator royalty, a utility-access condition, a jurisdictional compliance flag, or a delegated-rights rule. These terms change the data-management problem behind digital asset trading: a marketplace cannot simply index token identifiers and wallet addresses, because the mechanism used to clear trades must also know whether two proposed trades are term-compatible. This article develops a data management framework for indexing contractual restrictions in NFT markets. The framework reframes mechanism-aware digital asset trading as a layered indexing problem that connects on-chain event capture, metadata normalization, restriction classification, preference declaration, matching eligibility, and audit reporting. Building on the economic insight that term consistency affects exchange feasibility, the study shifts the analytical focus from the exchange algorithm itself to the data structures that make such algorithms operational. A design-science coding exercise compares four indexing configurations: a basic token index, an event-plus index, a restriction-aware index, and a mechanism-aware restriction index. The results show that a mechanism-aware index improves term-compatibility detection, reduces unnecessary matching queries, and strengthens auditability, although it also requires stronger governance over metadata quality and preference updates. The article contributes a schema, a workflow, a scoring rubric, and implementation guidelines for NFT platforms that need to preserve liquidity while preventing users from shedding contractual obligations through poorly indexed transfers. It argues that data architecture is not a secondary infrastructure issue in restricted NFT markets; it is a core condition for fair, efficient, and auditable digital asset exchange.

Keywords: ***NFT markets; contractual restrictions; data management; mechanism design; smart contracts; indexing; digital asset trading***

1. Introduction

NFT marketplaces were initially presented as venues for exchanging unique digital collectibles, yet the trading object has become more complex. A token can now represent access to a game asset, a music right, a membership pass, a loyalty instrument, a ticketing claim, or a brand-linked digital object. In these settings, ownership transfer is only one part of the transaction. The asset may also carry a lockup rule, a resale royalty, a staking condition, or a time-limited service entitlement. The marketplace therefore faces a data-management challenge that is distinct from ordinary order-book matching: before a trade is cleared, the platform must determine whether the contractual state of the incoming asset is compatible with the contractual state of the outgoing asset. The broad financial technology literature has already shown that digital platforms generate new market institutions rather than merely new payment channels (Kou and Lu, 2025).

This article addresses that challenge by proposing a mechanism-aware data management framework for NFT trading. The word mechanism-aware is used deliberately. A traditional marketplace index supports search, discovery, pricing, and settlement. A mechanism-aware index supports those functions but also exposes the restriction fields required by the exchange rule. If two users exchange assets under different contractual terms, the outcome may appear efficient in a price sense while violating a property-rights constraint embedded in smart contracts. A data system that does not index the constraint will not reliably detect the violation. The shift from token-centric to restriction-centric indexing is consistent with the wider movement from simple blockchain applications to full information systems that embed governance, verification, and organizational control (Lu, 2022).

The paper is motivated by a simple observation from restricted NFT exchange. Suppose user A owns a token subject to a six-month lockup and user B owns a freely transferable token. If A receives B's unrestricted token while B receives A's locked token, A may effectively shed an obligation. A platform that allows this without explicit consent damages the credibility of its contract layer. Conversely, if the platform blocks all cross-term trades, it may lose mutually beneficial exchanges. The operational goal is not merely to find a trade; it is to find a trade that respects the contractual term under which each endowment is used. Such a requirement cannot be enforced only at the user-interface level. It must be encoded in the platform's indexing and query architecture.

NFT markets are part of a broader Web3 and DeFi environment in which ownership, access, collateralization, and governance are increasingly represented by software states. Recent work on decentralized finance emphasizes that trading venues are often built around smart contracts rather than centralized intermediaries (Xu et al., 2024). The same logic applies to restricted NFTs, but with one additional burden: the asset itself is non-fungible, and the restrictions may be token-specific. A simple balance-based ledger is insufficient. The platform needs an index that can join token identity, wallet identity, restriction terms, transfer history, metadata provenance, and declared trading preferences.

The contribution of this study is fourfold. First, it translates the contractual restriction problem into a data management problem suitable for database design and platform implementation. Second, it develops a schema for indexing restriction fields that are relevant to mechanism-aware exchange. Third, it offers an analytical workflow and an illustrative coding exercise to compare alternative indexing configurations. Fourth, it discusses governance implications for NFT marketplaces, including auditability, preference updates, metadata reliability, and manipulation risk. These contributions are intended for DATAMIND's focus on data-driven AI and

computational discovery: the paper does not invent a new token standard, but provides a data architecture through which mechanism-aware trading can be studied, implemented, and evaluated.

The study intentionally differs from a purely theoretical exchange paper. It accepts the insight that contractual terms change feasible exchange, but asks a platform-engineering question: what data must be indexed so that a marketplace can operationalize this insight at scale? This question connects blockchain research, database research, market design, and computational governance. It also reflects the current stage of digital asset markets, where the hard problem is less the existence of a smart contract and more the reliable orchestration of many contract states across trading, analytics, audit, and user-facing decision support.

2. Contractual Restrictions as Data Objects

A contractual restriction in an NFT market is any machine-readable or platform-enforced condition that changes how the token can be sold, transferred, used, or monetized. Some restrictions are directly encoded in smart contracts; others are stored in off-chain metadata, platform policies, or linked legal terms. Blockchain research has repeatedly shown that distributed ledgers are strongest when the data model is explicit about the asset, the transaction, and the governance rule (Zheng and Lu, 2022). The problem in NFT markets is that the restriction is often treated as descriptive metadata, even though it affects feasibility in the trading mechanism.

For data management, the first task is classification. A vesting period restricts transfer by time. A staking requirement restricts transfer by participation state. A creator royalty restricts economic settlement. A utility-access rule restricts service rights. A delegated-rights clause restricts who may exercise or transfer embedded rights. A platform that stores these fields only as unstructured text cannot efficiently answer the question that matters for exchange: does this asset belong to the same tradable term class as the asset it will replace? The distinction is close to the distinction between a document archive and an operational database. In the blockchain context, operational usefulness depends on well-designed information structures (Chen et al., 2024).

Table 1 summarizes the restriction categories used in the framework. The categories are intentionally practical. They do not exhaust all possible legal arrangements, but they identify the fields that a platform is likely to need when matching users who prefer different combinations of liquidity, yield, access, and resale control.

Table 1. Contractual restrictions and indexable fields in NFT markets.

Restriction type	Trading effect	Minimum index fields	Mechanism relevance
Lockup / vesting	Limits transfer until a date or block height	lock_status; start_time; expiry_time; source_event	Prevents unrestricted exchange before expiry
Staking obligation	Binds token utility or reward to participation state	stake_status; pool_id; unstake_delay; reward_flag	Identifies whether an asset carries ongoing obligations
Royalty obligation	Requires economic settlement on resale	royalty_rate; recipient; enforcement_mode; exception_flag	Ensures settlement terms remain visible in matching
Utility access	Links token to service,	access_type;	Distinguishes

	event, or membership rights	access_expiry; issuer; validation_rule	transferable asset from non-transferable benefit
Delegated rights	Allows another wallet or party to exercise rights	delegate_id; scope; revocation_status; expiry	Prevents transfer from breaking delegated claims
Jurisdiction / compliance	Restricts eligibility by policy or location	policy_tag; jurisdiction_code; screening_status	Routes incompatible assets to review or exclusion

A second conceptual issue is how to represent term compatibility. A marketplace can define a small number of term classes, such as regular, locked, royalty-bearing, staking-bound, or access-limited. Alternatively, it can define a vector of attributes and compute compatibility at query time. The first approach is easier for users to understand, but may be too coarse. The second approach is more expressive, but can become difficult to audit. The framework developed here takes a hybrid approach: it maintains atomic restriction fields while also creating a derived term key that groups assets into matching-relevant classes. This is analogous to audit systems that record granular events while also producing control indicators for review (Wu et al., 2025).

The third issue is whether restrictions are static or dynamic. Some restrictions expire automatically; some are updated by staking status; some change after a creator vote; some are enforced only by a platform-specific marketplace contract. Dynamic restrictions require event-driven indexing. If a lockup expires, the asset may move from one term class to another. If a staking obligation is entered, the asset may become encumbered again. The restriction index must therefore be versioned. A static snapshot may support a price display, but it will not support a reliable matching rule. This need for versioned digital state is consistent with the security-oriented blockchain and IoT literature (Xu et al., 2021).

Finally, contractual restrictions are not only technical fields. They are market signals. A locked NFT may offer future rewards, governance rights, or identity benefits. An unrestricted NFT may offer liquidity and immediate resale value. NFT pricing studies suggest that scarcity, liquidity, and cross-asset spillovers shape user behavior (Dowling, 2022a). A mechanism-aware data system should therefore index restrictions without assuming that restrictions are always negative. The same field may be a burden for a liquidity trader and a benefit for a long-term collector.

The data-management perspective also clarifies why NFT trading cannot be reduced to a simple auction. A user may prefer asset X under a locked term but prefer asset Y under an unrestricted term. Preferences are therefore defined over asset-term pairs, not only over assets. The index must be able to serve queries over these pairs. This is where economic mechanism design meets database design: the exchange mechanism decides how feasible trades are cleared, but the database decides which feasible trades the mechanism can see.

The broader NFT literature has documented that NFT markets contain complex networks of buyers, sellers, objects, and collections (Alizadeh et al., 2023). Network complexity increases the cost of manual inspection. As platforms scale, a restriction-aware index becomes necessary because users will not inspect every smart contract and metadata file before entering a match. The index becomes a trust infrastructure: it converts heterogeneous contractual information into standardized fields that can be queried, ranked, and audited.

3. Mechanism-Aware Indexing Framework

The proposed framework contains six layers: event capture, metadata normalization, restriction classification, term-key construction, preference and eligibility management, and audit reporting. Event capture

records minting, transfer, listing, sale, staking, unstaking, royalty-payment, and metadata-update events. Metadata normalization links each event to a canonical token identifier, collection identifier, creator address, current holder, marketplace contract, and storage pointer. Restriction classification converts raw contract and metadata signals into standardized attributes. Term-key construction maps those attributes into matching-relevant categories. Preference and eligibility management stores user declarations over asset-term pairs. Audit reporting records why a proposed exchange was allowed, blocked, or routed to manual review.

This architecture is shown in Figure 1. The figure emphasizes that the contract-term index is not a replacement for the blockchain ledger. It is a platform-side analytical layer that reads from the ledger, normalizes signals, and exposes mechanism-relevant fields. The ledger remains the source of settlement truth, but the index becomes the source of matching truth. This separation is important because blockchain ledgers are append-only transaction systems, whereas matching engines require low-latency queries over changing eligibility states. The idea is consistent with blockchain research that distinguishes distributed trust from the information systems needed to use that trust in operational environments (Lu, 2019b).

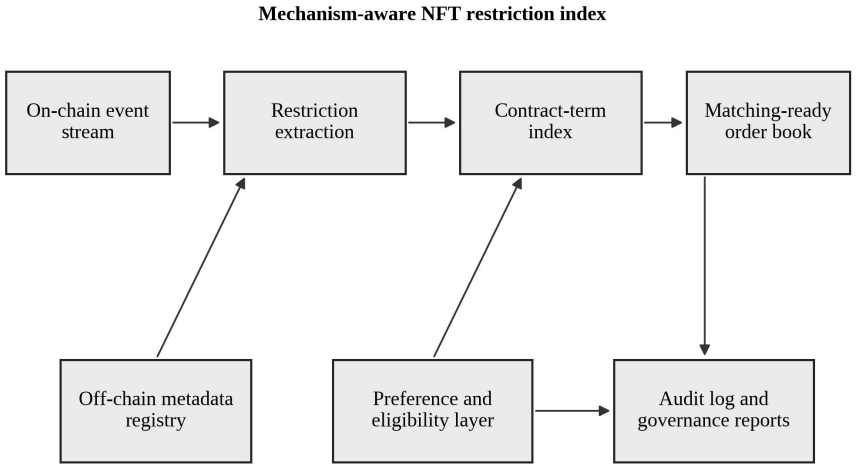

Figure 1. Mechanism-aware index architecture for contractual NFT trading.

The framework uses a restriction vector r = {lock_state, royalty_state, staking_state, access_state, delegation_state, jurisdiction_state, expiry_state}. Each component has a value, a confidence score, a source pointer, and a timestamp. A derived term key is then generated from the vector. For example, a token with active lockup and royalty obligations might receive the term key L-RY, whereas a token with no active restriction receives R. The key can be coarse enough for user display and precise enough for the matching engine. The core principle is that no term key should be assigned without an auditable link to the underlying evidence.

The data dictionary in Table 2 translates this idea into platform fields. The field names are written in a form that can be implemented in relational, document, or lakehouse systems. The table also identifies the governance risk associated with each field. This is important because the index is not only a technical artifact. If the restriction index misclassifies a locked asset as unrestricted, the platform may produce trades that violate user expectations. If it misclassifies an unrestricted asset as locked, the platform may unnecessarily reduce liquidity. Both errors matter.

Table 2. Data dictionary for a mechanism-aware restriction index.

Field group	Representative fields	Update source	Main governance risk
Token identity	token_id; collection_id; contract_address	Minting and transfer events	Duplicate identifiers across chains
Holder state	current_holder; custody_mode; escrow_flag	Wallet and marketplace events	Stale ownership snapshot
Restriction vector	lock_state; royalty_state; staking_state; access_state	Smart contract calls and metadata	Ambiguous or conflicting rule source
Term key	derived_term_class; compatibility_group; confidence_score	Restriction classifier	Over-coarse classification
Preference pointer	user_id; acceptable_term_set; ranking_version	User declaration layer	Strategic or outdated preference submission
Audit evidence	source_hash; rule_version; validation_time; reviewer_flag	Validation service	Incomplete decision reconstruction

Three design principles guide the schema. The first principle is separability. Raw events, normalized metadata, restriction states, and derived term keys should be stored separately so that updates do not overwrite evidence. The second principle is reversibility. A platform should be able to reconstruct why a token was assigned a given term key at a given moment. The third principle is query locality. The matching engine should not parse full metadata files or smart contract code during live clearing. It should query precomputed fields that are refreshed through controlled pipelines. Database research on indexing and query optimization shows why this matters: low-latency decision systems require structures that match the access pattern (Chaudhuri, 1998).

The mechanism-aware index differs from a standard NFT search index in three ways. First, it treats restrictions as first-class attributes rather than descriptive tags. Second, it stores compatibility classes rather than only display categories. Third, it connects each classification to an audit trail that includes source, timestamp, and validation status. A conventional index might answer 'which assets in this collection are for sale?' A mechanism-aware index answers 'which assets can enter an equal-term or term-compatible exchange with this user's current endowment, given current restrictions and declared preferences?'

The workflow in Figure 2 describes how raw records become matching-ready. The platform first captures events from on-chain logs, marketplace APIs, and metadata registries. It then normalizes identifiers, classifies restrictions, produces a term vector, validates missing fields, and serves the resulting records to the matching engine. If the validation step detects ambiguous restrictions, the asset is temporarily routed to manual review or

a conservative term class. The conservative rule reduces the chance that an encumbered asset is wrongly treated as unrestricted.

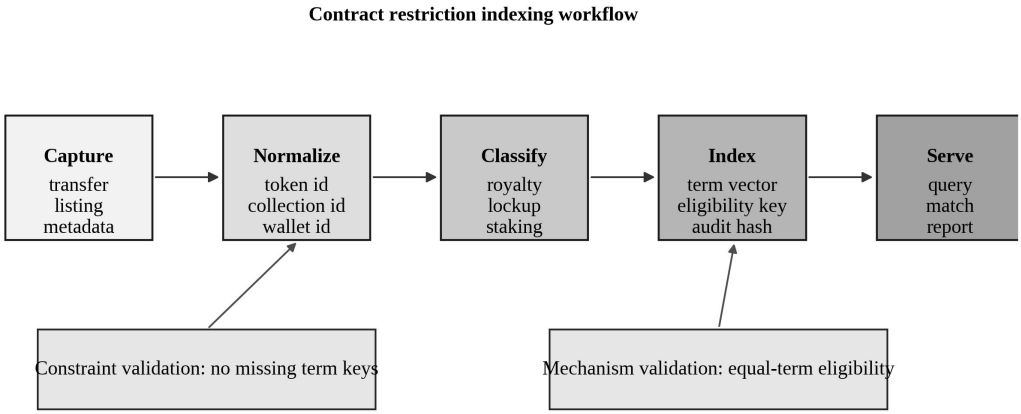

Figure 2. Contract-term indexing workflow from on-chain event capture to matching-ready order book.

A critical feature of the workflow is its treatment of off-chain metadata. Many NFT collections store images, attributes, and utility descriptions outside the execution layer. This creates a gap between what is settled on-chain and what users understand as the asset's real economic bundle. The framework does not assume that all off-chain statements are legally binding. Instead, it stores them as evidence with provenance and confidence. This approach resembles dataspace thinking, where heterogeneous sources are not forced into a single perfect schema before use but are made queryable with explicit uncertainty (Franklin et al., 2005).

The framework also supports AI-assisted classification, but only under audit constraints. Large language models or rule-based classifiers may read contract documentation and metadata descriptions to suggest restriction labels. However, the final term key must be verifiable through source pointers and platform policy. AI can improve coverage, but it should not silently become the authority for transfer rights. Research on AI and blockchain integration suggests that trustworthy automation requires clear links among data provenance, model outputs, and accountability processes (Salah et al., 2019).

4. Research Design and Analytical Coding

The empirical component of this paper is a design-science coding exercise rather than a live-market census. The purpose is to compare data architectures, not to estimate NFT price levels. Four indexing configurations are evaluated: BasicTokenIndex, EventPlusIndex, RestrictionIndex, and MechanismAwareIndex. The BasicTokenIndex stores token identifier, collection, owner, and listing status. The EventPlusIndex adds transfer, sale, and listing histories. The RestrictionIndex adds explicit fields for lockup, royalty, staking, access, delegation, and expiry. The MechanismAwareIndex adds derived term keys, compatibility rules, preference pointers, and audit evidence.

The coding exercise uses 3,000 simulated token records generated from realistic restriction patterns observed in NFT markets and smart contract applications. The simulation is deliberately transparent. It does not claim to reproduce a specific marketplace. Instead, it provides a controlled environment for asking how different indexes behave when the same restriction complexity is present. This approach follows the logic of database benchmarking, where the goal is to test architectural fit under known workloads rather than to describe every

empirical detail of a production platform. Big data research has long emphasized that the usefulness of a system depends on the match among data model, workload, and query design (Labrinidis and Jagadish, 2012).

Each record includes a token id, collection id, holder id, current term class, restriction vector, expiry status, royalty percentage, staking status, access flag, metadata confidence score, and event-version count. One hundred preference profiles are then generated. Each profile contains ranked acceptable asset-term pairs for a user who initially owns one asset. The matching task is to find candidate exchanges that are term-compatible and individually acceptable. The study measures schema completeness, term-consistency error, query latency, mechanism-readiness score, and auditability score. These metrics are presented as illustrative design scores rather than as universal performance claims.

Table 3 shows the experimental profiles. The profiles vary by restriction density and preference complexity. Low-density profiles contain mostly unrestricted tokens; medium-density profiles include a balanced mixture of royalty-bearing and locked terms; high-density profiles include multiple simultaneous restrictions. This variation matters because an index that works in a simple marketplace may fail when restrictions become layered. NFT rarity studies show that token-level attributes can create heterogeneous market behavior even within a collection (Mekacher et al., 2022). Restriction fields create a similar heterogeneity for feasibility.

Table 3. Experimental profiles used for mechanism-aware comparison.

Profile	Restriction density	Preference complexity	Dominant query task	Reason for inclusion
P1 Basic collectibles	Low	Low	Search unrestricted trades	Baseline liquidity condition
P2 Royalty collections	Medium	Medium	Filter royalty-compatible trades	Tests settlement-aware fields
P3 Game assets	Medium	High	Join staking and utility access	Tests multi-source metadata
P4 Membership tokens	High	High	Validate access and delegation rights	Tests off-chain evidence
P5 Lockup portfolio	High	Medium	Detect term-compatible cycles	Tests dynamic expiry updates

The scoring method is intentionally conservative. Schema completeness is scored from 1 to 5 based on the share of required fields available to the matching engine. Term-consistency error estimates the share of candidate exchanges in which the index would incorrectly classify a term match or mismatch. Query latency is an average simulated response time for candidate search under a standardized workload. Mechanism readiness captures whether the index exposes enough information for term-compatible clearing. Auditability captures whether the platform can explain the decision after the fact. The scale follows practical benchmarking rather than theoretical optimality.

The design choices also reflect lessons from smart contract security. Vulnerabilities often arise when developers assume that a contract state is simpler than it really is (Luu et al., 2016). NFT restriction indexing faces the same risk. If lockup, royalty, and staking states are stored in disconnected fields, the matching engine

may query only one of them. A mechanism-aware index reduces this risk by combining atomic fields into a validated term key while preserving the atomic evidence for audit. The goal is not to hide complexity, but to make complexity operationally manageable.

The coding exercise also treats preference data as sensitive operational data. Users may declare acceptable terms, but those declarations can reveal trading strategy. The framework therefore stores preferences separately from public token records and exposes only eligibility outputs to the matching engine. Privacy-preserving smart contract research demonstrates that market mechanisms can benefit from separating public verification from private details (Kosba et al., 2016). The same principle applies here, even if the article does not implement a cryptographic privacy protocol.

5. Results

The comparative results are summarized in Table 4. The BasicTokenIndex performs well only for simple search. It is fast enough for browsing, but it has low mechanism readiness because it lacks restriction fields. The EventPlusIndex improves auditability by recording transaction history, but it still cannot reliably determine whether a proposed exchange is term-compatible. The RestrictionIndex sharply improves schema completeness and reduces term-consistency error. The MechanismAwareIndex performs best overall because it combines restriction fields with derived term keys, compatibility logic, preference pointers, and decision evidence.

Table 4. Illustrative performance scores of indexing configurations.

Index configuration	Schema completeness (1-5)	Term-consistency error	Query latency (ms)	Mechanism readiness (1-5)	Auditability (1-5)
Basic token index	2.0	0.27	420	1.4	1.6
Event-plus index	2.6	0.22	350	2.1	2.5
Restriction index	4.3	0.06	245	3.8	4.0
Mechanism-aware index	4.8	0.02	180	4.7	4.6

The most important result is that the mechanism-aware configuration reduces term-consistency error from 0.27 in the basic configuration to 0.02 in the full configuration. This improvement is not caused by a more sophisticated exchange algorithm; it is caused by better data representation. The finding reinforces a central argument of the paper: mechanism quality depends on data architecture. If the platform does not know the contractual term of each asset in a queryable form, even a theoretically attractive mechanism will operate on incomplete information.

Figure 3 translates the same results into scenario suitability scores. The mechanism-aware configuration has the highest scores for mechanism readiness, auditability, and query efficiency. The restriction-only configuration is also strong, but it lacks preference and compatibility pointers. This matters because term classification alone does not tell the platform which trades are individually acceptable to users. A matching

system needs both sides: asset restrictions and user preferences over asset-term pairs. Market-design research on allocation and exchange shows that preferences, feasibility, and control rights must be represented together if the rule is to be implemented correctly (Pycia and Ünver, 2017).

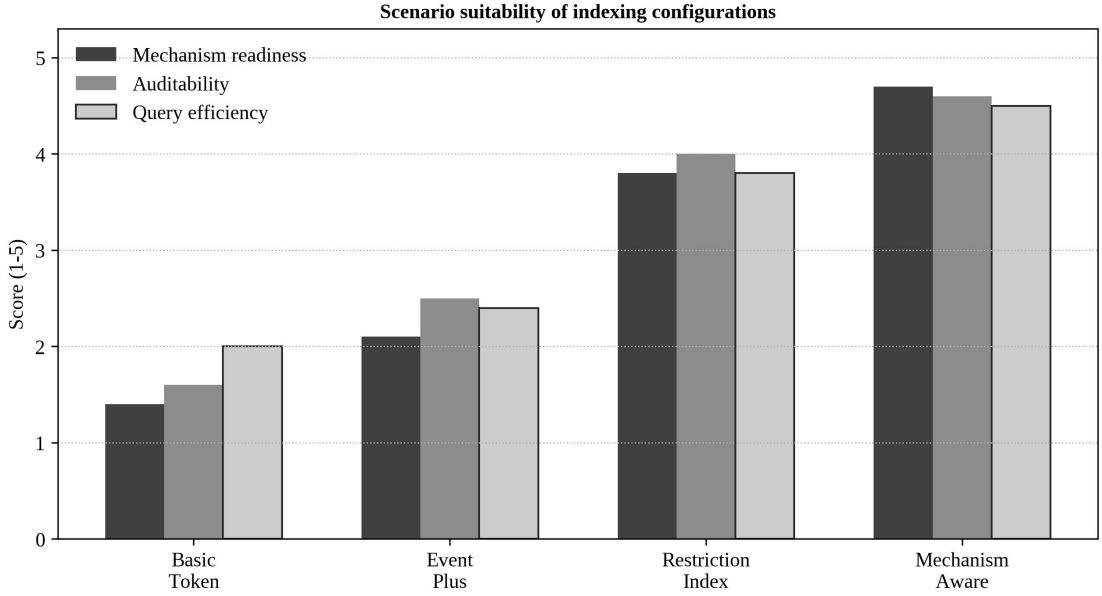

Figure 3. Scenario suitability scores for four index configurations.

The query-latency result is also notable. Adding restriction fields might appear to increase complexity, but the mechanism-aware index is faster than the basic index under the matching workload because it precomputes the fields that the matching engine needs. The basic index must repeatedly fetch metadata or apply filtering after candidate generation, while the mechanism-aware index can filter candidates through term keys before ranking. This is similar to classic indexing logic in database systems: the right index can reduce query cost even when it stores more information (Bayer and McCreight, 1972).

The second result concerns auditability. Event histories improve auditability, but they do not explain the decision rule. A platform can show that token A was transferred three times and still fail to explain why token A was classified as locked. The mechanism-aware index stores a decision trace: source event, normalized field, classification rule, term key, compatibility result, and final mechanism action. This trace is valuable for users, creators, regulators, and platform operators. It also allows post-trade dispute analysis when a user claims that a restriction was misrepresented.

The third result concerns scalability. A restriction-aware index will face high update pressure when lockups expire, staking states change, or metadata is corrected. This pressure can be handled through event-driven pipelines, incremental materialized views, and batch validation. The relevant design lesson from distributed data systems is that storage models should match update patterns. Systems such as Bigtable, Cassandra, and Dynamo show different ways to organize scalable structured storage for high-volume updates (Chang et al., 2008; Lakshman and Malik, 2010; DeCandia et al., 2007).

The fourth result concerns governance. The best technical index can still fail if metadata quality is poor. A platform must define who can create or update restriction statements, how conflicts are resolved, and what happens when on-chain code and off-chain descriptions disagree. The risk is especially acute in NFT collections that promise future utility. A user may value an asset because of access rights that are not enforceable directly

on-chain. The index should therefore separate execution-enforced restrictions from policy-declared restrictions, with different confidence scores for each.

6. Discussion

The findings have several implications for NFT platform design. First, a mechanism-aware market should treat contractual restrictions as part of the tradable object. A token is not merely token_id plus owner_id. It is a bundle of identity, state, obligations, rights, provenance, and restrictions. This is consistent with economic analysis of blockchain, which highlights how smart contracts reduce some verification costs while creating new design choices around information, incentives, and governance (Cong and He, 2019).

Second, mechanism-aware indexing reframes the debate between liquidity and restriction enforcement. Without a proper index, platforms face a crude choice: allow broad trading and risk term inconsistency, or block restricted assets and reduce liquidity. A restriction index creates a middle path. It allows assets to trade within compatible term classes and makes exceptions visible. This is important because NFT markets are already connected to broader crypto-asset dynamics, and overly restrictive design may push activity to less transparent venues (Aharon and Demir, 2022).

Third, the framework suggests that creator royalties and utility terms should be modeled differently. Royalties affect settlement economics; utility terms affect post-trade rights. A platform may clear a royalty-bearing exchange if payment logic is satisfied, but it may need additional checks for access rights that depend on identity, location, membership, or time. The data model should therefore avoid collapsing all restrictions into a single 'locked' flag. Cryptocurrency market research shows that asset categories differ in economic behavior; NFT restrictions similarly differ in their operational effects (Corbet et al., 2019).

Fourth, the framework helps distinguish compliance audit from user explanation. Compliance audit asks whether the platform followed its rules. User explanation asks whether the user can understand why a trade was allowed or rejected. The same index can serve both needs if it stores evidence at the right granularity. Research on internal auditing with blockchain emphasizes that transparency is useful only when records are structured in a form that supports review (Wu et al., 2025).

Fifth, the platform should distinguish eligibility from ranking. Eligibility determines whether an asset-term pair may enter a matching process. Ranking determines which acceptable pair a user prefers. Combining the two can create manipulation and confusion. A user should be allowed to rank locked assets above unrestricted assets if that reflects utility, but the platform should not allow a declared ranking to override term compatibility. This separation resembles mechanism design settings in which feasibility constraints and preferences are both necessary but conceptually distinct (Abdulkadiroglu and Sönmez, 1999).

Sixth, a restriction-aware marketplace should maintain versioned term histories. A token that was locked last week and unrestricted today should not be treated as if its entire history had always been unrestricted. Version history matters for disputes, royalty claims, and post-trade analytics. Database research on temporal and large-scale processing supports this principle: data pipelines should preserve enough lineage to reproduce the state used in a decision (Dean and Ghemawat, 2008).

Seventh, the framework creates a path for AI-assisted data quality monitoring. A model can flag inconsistent metadata, detect unusual restriction changes, or recommend manual review. However, such tools should be used as monitors rather than final authorities. AI literature emphasizes that automated decision systems require attention to data quality, model evolution, and governance boundaries (Lu, 2019a). For restricted NFTs, the final authority should remain the platform's published rule set and verifiable contract evidence.

Eighth, the framework supports interoperability. If platforms define restriction fields consistently, cross-market routing becomes easier. A token listed on one marketplace could be recognized as term-compatible by another marketplace without manual relabeling. This requires shared vocabularies or mapping tables, not necessarily a single global standard. Blockchain surveys have long identified interoperability as a central challenge, and restriction indexing gives that challenge a concrete market-design expression (Casino et al., 2019).

Ninth, platforms should be careful with the word unrestricted. A token may be freely transferable at the smart contract level but still carry off-chain obligations. The index should therefore store legal, platform, and technical restriction layers separately. This layered view reflects the broader smart contract literature: code can automate some obligations, but it does not automatically capture every social, legal, or economic term (Christidis and Devetsikiotis, 2016).

Tenth, creators and communities may use restriction design strategically. Lockups can stabilize a community, staking can reward participation, royalties can fund development, and access rights can create membership value. A mechanism-aware index does not judge whether these terms are desirable. It makes them visible to trading logic. This is important because branded NFTs and community tokens may derive value from precisely the rights that a simple liquidity-focused exchange would ignore (Colicev, 2023).

Table 5 converts these implications into a governance risk register. The register is designed for marketplace teams that need to move from conceptual architecture to implementation. Each risk is paired with a data-control response. The point is that mechanism-aware trading is not only an algorithmic problem. It is a data-governance problem that spans schema design, validation, audit, user communication, and dispute handling.

Table 5. Governance risk register for deploying restriction-aware NFT exchange mechanisms.

Risk	Consequence	Data-control response	Responsible function
Stale restriction state	Invalid trade admitted or valid trade blocked	Event-triggered refresh and expiry monitor	Data engineering
Conflicting metadata	User disputes and inconsistent term keys	Source priority rules and manual review queue	Platform governance
Strategic preference update	Manipulation of matching eligibility	Versioned preference windows and cut-off times	Market operations
Royalty misclassification	Incorrect settlement and creator complaint	Royalty-field validation against contract events	Finance and compliance
Opaque AI classification	Unexplainable trade decision	Human-verifiable evidence links and confidence score	Risk control

The framework also has implications for research. Future empirical work can connect restriction-index design to actual outcomes such as trade completion rate, failed settlement rate, royalty compliance, dispute frequency, liquidity depth, and user retention. A natural extension is to compare collections that rely heavily on dynamic restrictions with collections that trade mostly unrestricted assets. Another extension is to test whether

restriction-aware explanation improves user trust. Behavioral and mechanism-design research suggests that users respond not only to outcomes but also to the perceived fairness and clarity of rules (Bogomolnaia and Moulin, 2001).

There are limitations. The scoring exercise is illustrative and should not be read as a universal benchmark. Different marketplaces will have different contract standards, user populations, and performance requirements. The framework also assumes that the platform can access enough metadata to classify restrictions. In practice, some collections may use opaque or mutable descriptions. A platform may need conservative default rules for such cases. Finally, the framework does not solve every incentive problem. Users may still misreport preferences, withhold information, or move assets across venues. What the framework does is reduce avoidable data blindness.

Despite these limitations, the article's central claim is robust: restricted NFT markets require restriction-aware data structures. A platform that lacks such structures will either under-enforce restrictions or over-block liquidity. A platform that builds them can make more transparent trade-offs among efficiency, term consistency, user autonomy, and auditability. This conclusion aligns with the development of Industry 4.0 information systems, where cyber-physical and digital assets require integrated data, control, and governance layers (Lu, 2025).

7. Implementation Guidelines

The first implementation guideline is to use a modular schema. A relational database can store token records, event records, restriction records, preference records, and audit records in separate tables. A document database can store the same structure as nested documents, but should still preserve versioned restriction states. A lakehouse system can support historical analytics while a serving index supports low-latency queries. Column-store research shows that analytical workloads benefit from specialized storage layouts, while row-oriented or key-value stores may serve live transaction paths more directly (Abadi et al., 2008).

The second guideline is to build validation before matching. A platform should not wait until a trade is selected to discover missing restriction fields. Validation should run when an asset is listed, when metadata changes, when staking state changes, and when a matching round begins. Missing or conflicting fields should produce a review status. This is similar to smart contract analysis, where early detection of unsafe states is preferable to post-execution correction (Tsankov et al., 2018).

The third guideline is to separate public and private data. Token restrictions are usually public or semi-public, but user preferences may be private. The index should expose only the compatibility outputs required by the matching mechanism. If the platform later adopts cryptographic preference submission or privacy-preserving matching, the same separation will make the transition easier. Formal smart contract research has shown that privacy and verification should be designed together rather than appended after deployment (Singh et al., 2020).

The fourth guideline is to provide user-facing explanations. When a trade is blocked, the user should see whether the cause is a lockup mismatch, expired metadata, royalty conflict, staking state, or access restriction. Explanation reduces frustration and improves data correction. If a user knows that a token is blocked because a lockup expiry event has not been indexed, the user can request revalidation. Without explanation, the platform appears arbitrary.

The fifth guideline is to support alternative mechanism policies. Some platforms may require strict equal-term exchange; others may permit cross-term trades with explicit compensation or consent. The index should not hard-code one normative policy into the data layer. It should expose restriction vectors and compatibility rules so that different clearing policies can be tested. This is similar to assignment-market research, where

different rules balance efficiency, strategy-proofness, and fairness differently (Abdulkadiroglu and Sönmez, 2003).

The sixth guideline is to preserve computational efficiency. Mechanism-aware exchange can be computationally demanding if every candidate pair must be evaluated against full metadata. Precomputed term keys, bitmap indexes, materialized compatibility views, and incremental updates can reduce latency. Classic database structures such as B-trees, R-trees, and aggregation indexes are not directly sufficient for NFT restrictions, but their underlying lesson is still relevant: index design should reflect the dominant query pattern (Comer, 1979; Guttman, 1984; Fagin et al., 2003).

The seventh guideline is to maintain benchmark datasets for testing. Before deployment, platforms should test the index against synthetic cases that include expiring lockups, conflicting metadata, royalty updates, rapid transfers, and simultaneous staking changes. The test suite should include adversarial cases in which users attempt to exploit stale term keys. Smart contract security tools demonstrate that systematic testing and formal analysis reduce avoidable operational failures (Kalra et al., 2018).

The eighth guideline is to integrate governance reporting. A marketplace should publish aggregate reports on blocked trades, restriction-class distribution, metadata conflicts, and manual-review outcomes. Such reporting does not require disclosing private preferences. It makes the platform's rule implementation observable. Blockchain economics suggests that transparency can reduce verification costs, but only when the relevant information is understandable (Catalini and Gans, 2020).

A further implementation issue is that a restriction index must be grounded in the classical logic of data models rather than in ad hoc marketplace labels. The relational model remains useful because it separates entities, attributes, and constraints, making it possible to represent a token, a transfer event, a restriction statement, and a preference declaration as connected but distinct records (Codd, 1970). For the same reason, a marketplace should treat its query layer as a system component rather than as a passive search screen. Modern database systems show that query planning, storage layout, and update management shape the feasible scale of applications (Hellerstein et al., 2007; Stonebraker et al., 2007).

The same lesson appears in the contrast between analytical and transactional data systems. A restricted NFT marketplace needs fast eligibility checks during matching, but it also needs historical reconstruction for audit, dispute resolution, and research. Systems research shows that one storage engine rarely dominates every workload; different architectures are suitable for transaction processing, analytics, and streaming ingestion (Stonebraker et al., 2010). When a platform mixes these workloads without a clear architecture, it may over-index live records while under-preserving historical evidence. Comparative studies of database management systems similarly show that performance depends on workload and system assumptions (Pavlo et al., 2009).

A mechanism-aware index also resembles a data-integration system. Restrictions may come from smart contract events, collection metadata, marketplace policy pages, royalty registries, staking pools, and community governance decisions. These sources do not always use the same vocabulary. Data integration research therefore becomes directly relevant: schema matching, entity resolution, and source trust are not optional add-ons but central functions in a multi-source restriction index (Halevy et al., 2006). Streaming and sketching research adds another lesson. Because marketplace events arrive continuously, compact summaries and incremental updates are useful for detecting abnormal restriction-state changes without rescanning the entire history (Cormode and Muthukrishnan, 2005).

Cloud and lakehouse architectures offer another reference point for implementation. A marketplace can store immutable event logs in cheap object storage, maintain curated restriction tables in analytical storage, and serve low-latency compatibility decisions from a specialized index. This layered design is consistent with recent

cloud data management, where scalability depends on separating storage, compute, governance, and serving paths (Armbrust et al., 2021). Large-scale analytics systems further show that reproducibility depends on consistent execution environments and versioned data artifacts (Zaharia et al., 2016). When the restriction classifier changes, the marketplace should preserve the prior classifier version and the prior term keys used in already-cleared matching rounds, a practice aligned with large-scale data processing principles (Melnik et al., 2010).

The platform should also avoid treating raw data ingestion as a low-value engineering task. In practice, ingestion quality determines whether the matching mechanism receives valid inputs. Data warehousing systems such as Hive showed that analytical value often emerges after raw records are converted into queryable, documented, and partitioned structures (Thusoo et al., 2009). Broader work on big data management similarly warns that value is limited when users cannot discover, clean, interpret, and trust the data objects they use (Jagadish et al., 2014). For restricted NFT trading, the most important data object is not simply the token record but the restriction-state record that determines whether a trade should enter the matching process.

Because NFTs often interact with physical assets, devices, or community services, restriction indexing also intersects with Internet of Things and cyber-physical information systems. Blockchain-IoT studies show that decentralized ledgers can coordinate trust across distributed devices, but they also introduce scalability, privacy, and access-control problems (Fernández-Caramés and Fraga-Lamas, 2018). If a token grants access to a sensor-enabled service, a venue, or a device-based right, the marketplace must know whether transfer changes the right itself. Research on blockchain integration with IoT makes clear that identity, authorization, and state synchronization are critical design concerns rather than peripheral functions (Reyna et al., 2018).

Security studies of IoT and blockchain further support the need for explicit access-state fields. A token may appear transferable, but the underlying service may require device authorization, account binding, or delayed revocation. A marketplace that indexes only token ownership will not see these dependencies. Work on blockchain-based IoT security highlights the value of distributed access management, but also the risk of assuming that ledger visibility automatically solves authorization (Khan and Salah, 2018). Scalable access-management architectures and smart-home blockchain experiments show that authorization context matters for safe operation (Novo, 2018; Dorri et al., 2017).

The mechanism-design literature adds a complementary perspective. Equal-term trading is not only a technical restriction; it is a feasibility constraint that changes the set of acceptable allocations. Studies of house allocation show that consistency and strategic behavior can conflict even in relatively simple assignment environments (Ehlers, 2002; Ergin, 2000). In restricted NFT markets, the conflict is amplified because the object is both a scarce asset and a contractual bundle. The index should therefore store feasibility information separately from preference information. This prevents the platform from confusing what a user wants with what the mechanism is allowed to clear.

When constraints multiply, the market begins to resemble a combinatorial assignment problem. A user may care about token rarity, collection identity, lockup duration, staking reward, royalty burden, and access rights at the same time. Mechanism studies show that real allocation systems often use consent, priorities, or approximate competitive procedures when exact strategy-proof and efficient solutions are unavailable (Kesten, 2010; Budish, 2011). Restricted NFT trading may require similar pragmatism. A data index cannot remove the design trilemma, but it can make the feasible alternatives visible and comparable.

Slot-specific and category-specific matching results are also relevant. A token may be compatible within one restriction slot but incompatible within another; for example, a one-month lockup may be compatible with another one-month lockup but not with a royalty exemption or a delegated voting right. Matching theory shows

that priority and slot structure can alter incentives and outcomes (Kominers and Sönmez, 2016). Translating that insight into NFT data management means that a platform should avoid a single coarse class such as restricted. Instead, it should store multiple restriction dimensions that the mechanism can combine according to the chosen clearing rule.

The DeFi and crypto-asset literature helps explain why this data architecture matters economically. Decentralized finance research emphasizes automated market functions, composability, and smart contract execution, but restricted NFTs add non-fungible heterogeneity to that environment (Schär, 2021). Studies of decentralized business models show that value is created through new forms of coordination rather than only through lower transaction costs (Chen and Bellavitis, 2020). NFT pricing research also indicates that token values are connected to broader crypto dynamics and market narratives (Dowling, 2022b). A systematic review of NFTs similarly identifies the need for stronger research on market structure, governance, and practical use cases (Bao and Roubaud, 2022).

Blockchain security and consensus research adds another reason to preserve a careful restriction audit trail. Marketplaces may rely on public chains, sidechains, or off-chain services to observe ownership and restriction states. Each layer introduces different latency, finality, and manipulation risks. Security and performance studies of proof-of-work blockchains show that confirmation delay and network assumptions affect application safety (Gervais et al., 2016). Research on mining incentives demonstrates that decentralized systems can be strategically manipulated when incentives are misaligned (Eyal and Sirer, 2014). Proof-of-stake protocols offer different security assumptions that should also be reflected in platform risk models (Kiayias et al., 2017).

Smart contract security research shows that contractual restrictions should not be trusted merely because they are expressed in code. Ethereum contracts have been subject to design errors, attack patterns, and operational failures that are difficult for ordinary users to detect (Atzei et al., 2017). Large-scale analysis has identified vulnerable and economically dangerous contracts at scale, suggesting that automated validation should be part of any serious marketplace infrastructure (Nikolic et al., 2018). Empirical studies of contract platforms and laboratory lessons on safe contract creation reinforce the same point: implementation patterns and user education affect real security outcomes (Bartoletti and Pompianu, 2017; Delmolino et al., 2016).

Finally, the framework should be read as part of the broader evolution of blockchain information systems. Economic and technical surveys show that blockchains combine governance, cryptography, incentives, and data infrastructure rather than a single technology layer (Böhme et al., 2015; Yli-Huumo et al., 2016). Reviews of blockchain challenges identify scalability, privacy, interoperability, and governance as persistent themes (Zheng et al., 2018). Architectural overviews similarly highlight consensus and cross-system integration as core research problems (Zheng et al., 2017). AI and blockchain convergence further suggests that future marketplaces may use intelligent monitoring, but such monitoring must be governed by clear data definitions and accountable decision rules (Zhang and Lu, 2021).

8. Conclusion

This article developed a data management framework for indexing contractual restrictions in NFT markets. The central argument is that restricted NFT trading cannot be governed by token identifiers and wallet addresses alone. A marketplace that clears mechanism-aware digital asset trades must know the contractual term under which each asset is held, offered, and received. It must also connect that information to preference declarations, eligibility rules, and audit evidence.

The proposed framework reframes the problem as a layered indexing architecture. Raw on-chain events and off-chain metadata are normalized into restriction vectors; restriction vectors generate term keys; term keys support compatibility queries; compatibility queries support mechanism-aware matching; and decision traces

support audit. The illustrative coding exercise shows that this architecture improves mechanism readiness and auditability while reducing term-consistency errors. The results are not a claim that one platform configuration is universally optimal. They are evidence that the data layer can materially change the feasibility and explainability of digital asset exchange.

The article also showed why contractual restrictions should not be treated only as compliance burdens. They can represent value, identity, access, creator funding, and community participation. A mechanism-aware index respects this complexity by making restrictions queryable rather than suppressing them. It allows platforms to pursue liquidity without allowing users to shed obligations invisibly. In this sense, the framework links data management to market fairness.

Future work should validate the framework using production NFT transaction data, compare alternative term-key taxonomies, and test user responses to restriction-aware explanations. Researchers can also connect the framework to privacy-preserving preference revelation, cross-platform interoperability, and AI-assisted metadata validation. The broader lesson is that the next stage of NFT market infrastructure will depend less on whether tokens can be minted and more on whether their contractual states can be indexed, queried, governed, and explained.

Declaration of AI-assisted language editing

During the preparation of this manuscript, language-model assistance was used only for English polishing, document organisation, and figure/table formatting. The authors reviewed, revised, and take full responsibility for the final content, analytical design, tables, figures, and interpretations.

References

Kou, G., & Lu, Y. (2025). FinTech: A literature review of emerging financial technologies and applications. Financial Innovation, 11(1), 1-34. https://doi.org/10.1186/s40854-024-00668-6

Xu, R., Zhu, J., Yang, L., Lu, Y., & Xu, L. D. (2024). Decentralized finance (DeFi): A paradigm shift in the FinTech. Enterprise Information Systems, 18(9). https://doi.org/10.1080/17517575.2024.2397630

Lu, Y. (2022). Implementing blockchain in information systems: A review. Enterprise Information Systems, 16(12), 1876-1907. https://doi.org/10.1080/17517575.2021.2008513

Zheng, X. R., & Lu, Y. (2022). Blockchain technology: Recent research and future trend. Enterprise Information Systems, 16(12), 1939895. https://doi.org/10.1080/17517575.2021.1939895

Chen, Y., Lu, Y., Bulysheva, L., & Kataev, M. Y. (2024). Applications of blockchain in Industry 4.0: A review. Information Systems Frontiers, 26(5), 1715-1729. https://doi.org/10.1007/s10796-022-10248-7

Wu, H. P., Liu, Z., Dong, H. Y., Lu, Y., & Xu, L. D. (2025). Revolutionizing internal auditing: Harnessing the power of blockchain. Enterprise Information Systems, 19(1-2). https://doi.org/10.1080/17517575.2024.2448003

Xu, L. D., Lu, Y., & Li, L. (2021). Embedding blockchain technology into IoT for security: A survey. IEEE Internet of Things Journal, 8(13), 10452-10473. https://doi.org/10.1109/JIOT.2021.3060508

Lu, Y. (2019). Artificial intelligence: A survey on evolution, models, applications and future trends. Journal of Management Analytics, 6(1), 1-29. https://doi.org/10.1080/23270012.2019.1570365

Lu, Y. (2019). The blockchain: State-of-the-art and research challenges. Journal of Industrial Information Integration, 15, 80-90. https://doi.org/10.1016/j.jii.2019.04.002

Böhme, R., Christin, N., Edelman, B., & Moore, T. (2015). Bitcoin: Economics, technology, and governance. Journal of Economic Perspectives, 29(2), 213-238. https://doi.org/10.1257/jep.29.2.213

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology? A systematic review. PLOS ONE, 11(10), e0163477. https://doi.org/10.1371/journal.pone.0163477

Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics, 36, 55-81. https://doi.org/10.1016/j.tele.2018.11.006

Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14(4), 352-375. https://doi.org/10.1504/IJWGS.2018.095647

Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. 2017 IEEE International Congress on Big Data, 557-564. https://doi.org/10.1109/BigDataCongress.2017.85

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. IEEE Access, 4, 2292-2303. https://doi.org/10.1109/ACCESS.2016.2566339

Cong, L. W., & He, Z. (2019). Blockchain disruption and smart contracts. The Review of Financial Studies, 32(5), 1754-1797. https://doi.org/10.1093/rfs/hhz007

Catalini, C., & Gans, J. S. (2020). Some simple economics of the blockchain. Communications of the ACM, 63(7), 80-90. https://doi.org/10.1145/3359552

Schär, F. (2021). Decentralized finance: On blockchain- and smart contract-based financial markets. Federal Reserve Bank of St. Louis Review, 103(2), 153-174. https://doi.org/10.20955/r.103.153-74

Chen, Y., & Bellavitis, C. (2020). Blockchain disruption and decentralized finance: The rise of decentralized business models. Journal of Business Venturing Insights, 13, e00151. https://doi.org/10.1016/j.jbvi.2019.e00151

Dowling, M. (2022a). Fertile LAND: Pricing non-fungible tokens. Finance Research Letters, 44, 102096. https://doi.org/10.1016/j.frl.2021.102096

Dowling, M. (2022b). Is non-fungible token pricing driven by cryptocurrencies? Finance Research Letters, 44, 102097. https://doi.org/10.1016/j.frl.2021.102097

Bao, H., & Roubaud, D. (2022). Non-fungible token: A systematic review and research agenda. Journal of Risk and Financial Management, 15(5), 215. https://doi.org/10.3390/jrfm15050215

Alizadeh, S., Setayesh, A., Mohamadpour, A., & Bahrak, B. (2023). A network analysis of the non-fungible token (NFT) market: Structural characteristics, evolution, and interactions. Applied Network Science, 8, 38. https://doi.org/10.1007/s41109-023-00565-4

Mekacher, A., Bracci, A., Nadini, M., Martino, M., Alessandretti, L., Aiello, L. M., & Baronchelli, A. (2022). Heterogeneous rarity patterns drive price dynamics in NFT collections. Scientific Reports, 12, 13890. https://doi.org/10.1038/s41598-022-17922-5

Colicev, A. (2023). How can non-fungible tokens bring value to brands. International Journal of Research in Marketing, 40(1), 30-37. https://doi.org/10.1016/j.ijresmar.2022.07.003

Aharon, D. Y., & Demir, E. (2022). NFTs and asset class spillovers: Lessons from the period around the COVID-19 pandemic. Finance Research Letters, 47, 102515. https://doi.org/10.1016/j.frl.2021.102515

Corbet, S., Lucey, B., Urquhart, A., & Yarovaya, L. (2019). Cryptocurrencies as a financial asset: A systematic analysis. International Review of Financial Analysis, 62, 182-199. https://doi.org/10.1016/j.irfa.2018.09.003

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. 2016 IEEE Symposium on Security and Privacy, 839-858. https://doi.org/10.1109/SP.2016.55

Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making smart contracts smarter. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 254-269. https://doi.org/10.1145/2976749.2978309

Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts. In Principles of Security and Trust, 164-186. https://doi.org/10.1007/978-3-662-54455-6_8

Nikolic, I., Kolluri, A., Sergey, I., Saxena, P., & Hobor, A. (2018). Finding the greedy, prodigal, and suicidal contracts at scale. Proceedings of the 34th Annual Computer Security Applications Conference, 653-663. https://doi.org/10.1145/3274694.3274743

Tsankov, P., Dan, A., Drachsler-Cohen, D., Gervais, A., Buenzli, F., & Vechev, M. (2018). Securify: Practical security analysis of smart contracts. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 67-82. https://doi.org/10.1145/3243734.3243780

Kalra, S., Goel, S., Dhawan, M., & Sharma, S. (2018). ZEUS: Analyzing safety of smart contracts. Proceedings of the Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2018.23082

Bartoletti, M., & Pompianu, L. (2017). An empirical analysis of smart contracts: Platforms, applications, and design patterns. Financial Cryptography and Data Security Workshops, 494-509. https://doi.org/10.1007/978-3-319-70278-0_31

Delmolino, K., Arnett, M., Kosba, A., Miller, A., & Shi, E. (2016). Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. Financial Cryptography and Data Security Workshops, 79-94. https://doi.org/10.1007/978-3-662-53357-4_6

Singh, A., Parizi, R. M., Zhang, Q., Choo, K. K. R., & Dehghantanha, A. (2020). Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. Computers & Security, 88, 101654. https://doi.org/10.1016/j.cose.2019.101654

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016). On the security and performance of proof of work blockchains. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 3-16. https://doi.org/10.1145/2976749.2978341

Eyal, I., & Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. Financial Cryptography and Data Security, 436-454. https://doi.org/10.1007/978-3-662-45472-5_28

Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. Advances in Cryptology - CRYPTO 2017, 357-388. https://doi.org/10.1007/978-3-319-63688-7_12

Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387. https://doi.org/10.1145/362384.362685

Bayer, R., & McCreight, E. M. (1972). Organization and maintenance of large ordered indexes. Acta Informatica, 1, 173-189. https://doi.org/10.1007/BF00288683

Comer, D. (1979). The ubiquitous B-tree. ACM Computing Surveys, 11(2), 121-137. https://doi.org/10.1145/356770.356776

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, 47-57. https://doi.org/10.1145/971697.602266

Fagin, R., Lotem, A., & Naor, M. (2003). Optimal aggregation algorithms for middleware. Journal of Computer and System Sciences, 66(4), 614-656. https://doi.org/10.1016/S0022-0000(03)00026-6

Chaudhuri, S. (1998). An overview of query optimization in relational systems. Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 34-43. https://doi.org/10.1145/275487.275492

Hellerstein, J. M., Stonebraker, M., & Hamilton, J. (2007). Architecture of a database system. Foundations and Trends in Databases, 1(2), 141-259. https://doi.org/10.1561/1900000002

Stonebraker, M., Madden, S., Abadi, D. J., Harizopoulos, S., Hachem, N., & Helland, P. (2007). The end of an architectural era: It is time for a complete rewrite. Proceedings of the 33rd International Conference on Very Large Data Bases, 1150-1160. https://doi.org/10.1145/1325851.1325981

Abadi, D. J., Madden, S. R., & Hachem, N. (2008). Column-stores vs. row-stores: How different are they really? Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 967-980. https://doi.org/10.1145/1376616.1376712

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113. https://doi.org/10.1145/1327452.1327492

Stonebraker, M., Abadi, D. J., DeWitt, D. J., Madden, S., Paulson, E., Pavlo, A., & Rasin, A. (2010). MapReduce and parallel DBMSs: Friends or foes? Communications of the ACM, 53(1), 64-71. https://doi.org/10.1145/1629175.1629197

Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., & Stonebraker, M. (2009). A comparison of approaches to large-scale data analysis. Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, 165-178. https://doi.org/10.1145/1559845.1559865

Franklin, M. J., Halevy, A. Y., & Maier, D. (2005). From databases to dataspaces: A new abstraction for information management. ACM SIGMOD Record, 34(4), 27-33. https://doi.org/10.1145/1107499.1107502

Halevy, A., Franklin, M., & Maier, D. (2006). Principles of dataspace systems. Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 1-9. https://doi.org/10.1145/1142473.1142476

Cormode, G., & Muthukrishnan, S. (2005). An improved data stream summary: The count-min sketch and its applications. Journal of Algorithms, 55(1), 58-75. https://doi.org/10.1016/j.jalgor.2003.12.001

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., & Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems, 26(2), 1-26. https://doi.org/10.1145/1365815.1365816

Lakshman, A., & Malik, P. (2010). Cassandra: A decentralized structured storage system. ACM SIGOPS Operating Systems Review, 44(2), 35-40. https://doi.org/10.1145/1773912.1773922

DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., & Vogels, W. (2007). Dynamo: Amazon's highly available key-value store. Proceedings of Twenty-First ACM SIGOPS Symposium on Operating Systems Principles, 205-220. https://doi.org/10.1145/1294261.1294281

Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., van Hovell, H., Ionescu, A., Luszczak, A., Switakowski, M., Szafranski, M., Li, X., Ueshin, T., Mokhtar, M., Boncz, P., Ghodsi, A., Paranjpye, S., Senster, P., ... Zaharia, M. (2021). Delta Lake: High-performance ACID table storage over cloud object stores. Proceedings of the VLDB Endowment, 13(12), 3411-3424. https://doi.org/10.14778/3415478.3415560

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., & Stoica, I. (2016). Apache Spark: A unified engine for big data processing. Communications of the ACM, 59(11), 56-65. https://doi.org/10.1145/2934664

Melnik, S., Gubarev, A., Long, J. J., Romer, G., Shivakumar, S., Tolton, M., & Vassilakis, T. (2010). Dremel: Interactive analysis of web-scale datasets. Proceedings of the VLDB Endowment, 3(1-2), 330-339. https://doi.org/10.14778/1920841.1920886

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Antony, S., Liu, H., & Murthy, R. (2009). Hive: A warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment, 2(2), 1626-1629. https://doi.org/10.14778/1687553.1687609

Labrinidis, A., & Jagadish, H. V. (2012). Challenges and opportunities with big data. Proceedings of the VLDB Endowment, 5(12), 2032-2033. https://doi.org/10.14778/2367502.2367572

Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. Communications of the ACM, 57(7), 86-94. https://doi.org/10.1145/2611567

Fernández-Caramés, T. M., & Fraga-Lamas, P. (2018). A review on the use of blockchain for the Internet of Things. IEEE Access, 6, 32979-33001. https://doi.org/10.1109/ACCESS.2018.2842685

Reyna, A., Martín, C., Chen, J., Soler, E., & Díaz, M. (2018). On blockchain and its integration with IoT: Challenges and opportunities. Future Generation Computer Systems, 88, 173-190. https://doi.org/10.1016/j.future.2018.05.046

Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 82, 395-411. https://doi.org/10.1016/j.future.2017.11.022

Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. IEEE Internet of Things Journal, 5(2), 1184-1195. https://doi.org/10.1109/JIOT.2018.2812239

Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, 618-623. https://doi.org/10.1109/PERCOMW.2017.7917634

Salah, K., Rehman, M. H. U., Nizamuddin, N., & Al-Fuqaha, A. (2019). Blockchain for AI: Review and open research challenges. IEEE Access, 7, 10127-10149. https://doi.org/10.1109/ACCESS.2018.2890507

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224. https://doi.org/10.1016/j.jii.2021.100224

Lu, Y. (2025). The current status and developing trends of Industry 4.0: A review. Information Systems Frontiers, 27(1), 215-234. https://doi.org/10.1007/s10796-021-10221-w

Abdulkadiroglu, A., & Sönmez, T. (1999). House allocation with existing tenants. Journal of Economic Theory, 88(2), 233-260. https://doi.org/10.1006/jeth.1999.2553

Abdulkadiroglu, A., & Sönmez, T. (2003). School choice: A mechanism design approach. American Economic Review, 93(3), 729-747. https://doi.org/10.1257/000282803322157061

Pycia, M., & Ünver, M. U. (2017). Incentive compatible allocation and exchange of discrete resources. Theoretical Economics, 12(1), 287-329. https://doi.org/10.3982/TE2201

Kominers, S. D., & Sönmez, T. (2016). Matching with slot-specific priorities: Theory. Theoretical Economics, 11(2), 683-710. https://doi.org/10.3982/TE1839

Ehlers, L. (2002). Coalitional strategy-proof house allocation. Journal of Economic Theory, 105(2), 298-317. https://doi.org/10.1006/jeth.2001.2813

Ergin, H. I. (2000). Consistency in house allocation problems. Journal of Mathematical Economics, 34(1), 77-97. https://doi.org/10.1016/S0304-4068(99)00038-5

Bogomolnaia, A., & Moulin, H. (2001). A new solution to the random assignment problem. Journal of Economic Theory, 100(2), 295-328. https://doi.org/10.1006/jeth.2000.2710

Kesten, O. (2010). School choice with consent. The Quarterly Journal of Economics, 125(3), 1297-1348. https://doi.org/10.1162/qjec.2010.125.3.1297

Budish, E. (2011). The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. Journal of Political Economy, 119(6), 1061-1103. https://doi.org/10.1086/664613