CodexField

The Next-Generation Web3 Content Assetization Platform

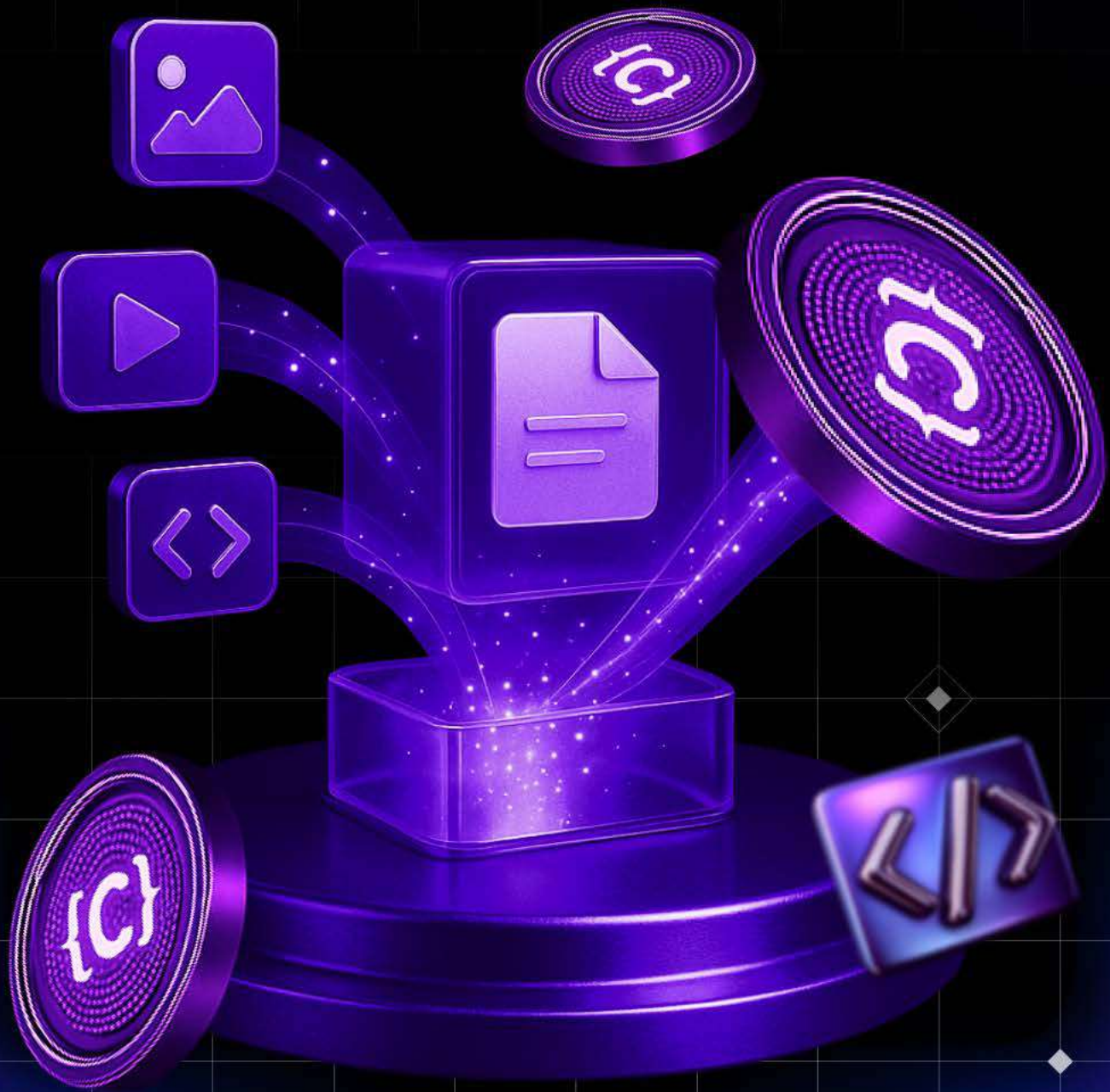


Table of Contents

01 PREFACE

1.1 The Rise of the AI Creator Economy

With the rapid development of large models and AIGC tools, the creator economy is currently undergoing a profound paradigm shift. The role of content is fundamentally migrating from "filler for platform traffic" to "native asset units in a chain-based system." Especially in the context of AI, content is not only an output result but also a fundamental resource for model fine-tuning, corpus support, and interaction logic construction. This enhancement of resource attributes endows content itself with asset value comparable to computing power and bandwidth.

According to estimates by Wall Street analyst Doug Shapiro, the global market size of the creator economy reached $250 billion in 2024 and is expected to double to $480 billion by 2027. The structural momentum behind this rapid growth stems not only from the commercialization volume of content on Web2 platforms like TikTok, YouTube, and Patreon but also from the deeper driving force of the continuous strengthening of the trends of content "atomization" and "assetization." In other words, while platforms used to monetize content, we are now gradually transitioning to content itself possessing an independent financial life.

Of particular note is the rise of structured content assets—including high-frequency reusable content units such as model call records, Prompt templates, code snippets, algorithm modules, training corpora, etc.—which are being regarded as one of the most promising asset forms on-chain, following stablecoins and RWA. On platforms like Hugging Face, OpenAI, and Midjourney, the licensing, calling, and reuse of such content have become de facto business processes, but they still rely on centralized gateways for resource scheduling and revenue distribution, lacking transparent, standard, and composable underlying protocols. This non-open state requires the intervention of Web3's standardized infrastructure to form a universal model that is predictable, licensable, and capable of revenue mapping.

Furthermore, from the perspective of macro policies and capital signals, global content infrastructure is also attracting capital pursuit. AIGC infrastructure platforms represented by Stability AI and RunwayML have successively completed financing rounds exceeding hundreds of millions of dollars, while emerging projects focusing on content architecture and licensing protocols, such as Story Protocol and Lit Protocol, have also gained favor from top institutions like a16z and Hashed. When Story Protocol completed its $54 million financing in 2023, it clearly stated, "In the future, content will no longer be files, but decomposable, recombinable, incentivizable asset tree structures."

However, despite the explosive growth of the creator economy and the increasingly significant trends of content structuring and assetization, there still exists a series of institutional and technical foundational gaps between "potential assets" and "verifiable, tradable on-chain assets."

1.2 Pain Points in the Web3 Content Market Space

At this stage, the mechanisms for rights confirmation and value capture of content itself in the on-chain world still lag severely. In the traditional Web2 model, content ownership and revenue distribution highly rely on platform centrality, and creators lack independent and controllable means of permission expression and financialization paths. In the current immature state of the Web3 content ecosystem, the lack of structural infrastructure has become a core obstacle restricting the content assetization process.

We see that the current process for putting content and code on-chain is still complex, lacking unified format standards and confirmation protocols. Structured content such as algorithm libraries, model weights, Prompt templates, and graphic-text corpora are all highly fragmented in terms of data structure, metadata tagging, and calling interfaces, creators often find it difficult to complete low-cost, verifiable, cross-platform pre-authorization processes.

According to a GitHub report, among over 100 million registered developers worldwide, less than 1% of projects possess complete on-chain intellectual property credentials. Emerging assets like AIGC content and AI datasets, which frequently appear in Web3, are almost completely outside the rights confirmation system. The vacuum at the property rights level not only hinders content circulation but also prevents creators from truly controlling the boundaries and rights of their output.

At the same time, the revenue mapping path between content and its usage behavior remains incomplete in Web3 today. The incentive mechanisms provided by current mainstream content platforms are mostly platform subsidies or preset rewards, rather than dynamic profit-sharing based on real user behaviors such as "access, subscription, citation, calling." This disconnect in incentives directly leads to a lack of collaborative motivation between users and creators, and the use value of content cannot be fed back to the original author in a timely manner. On Web3 content platforms like Mirror and Lens, although the barrier to content creation is significantly lowered, content still falls into an inefficient cycle of "visible but not monetizable" because the revenue mechanism is not established.

More critically, the storage, computation, authorization, and transaction links of content assets are still in a fragmented state, lacking unified composable protocol standards. Underlying storage networks such as Arweave, Filecoin, and Greenfield provide diverse decentralized storage solutions, but they lack collaboration with the calling side, unable to form a closed-loop system of "content as a service." Simultaneously, the lack of native smart contract authorization, on-chain call records, and billing mechanisms also makes it difficult for content assets to possess the "measurability, financializability, and composability" required by RWA. This not only causes content assets to remain stagnant and illiquid but also limits their potential to enter more complex financial structures such as staking, unpacking, and revenue sharing.

In such an environment lacking standards, suffering from severe disconnects, and broken revenue streams, content, as one of the most dynamic and potential elements in the Web3 ecosystem, has long been in a "non-assetized" state. This not only misses the precise capture of content value but also affects the further evolution of high-growth channels such as AI model training, data markets, and the creator economy. It can be said that for content assets to truly step into the on-chain world, the primary prerequisite is to build a closed-loop mechanism that covers "from rights confirmation, to use, and then to revenue," enabling all structured content to possess native financial attributes and standardized circulation capabilities.

Against this backdrop, CodexField is committed to solving the underlying bottleneck problems faced by the current content economy in the Web3 context by building a set of native rights confirmation, calling, and financialization infrastructure for structured content assets. As a protocol stack that supports the full-process on-chain and usage of content types such as code, models, and Prompts, we are also a middleware system connecting content creation and on-chain value capture.

CodexField will build a closed-loop network from rights confirmation, usage to revenue based on unified data structures, programmable authorization mechanisms, and native billing paths, enabling every content access, citation, and training call to form traceable and distributable on-chain revenue mapping. By connecting the mapping link between content and value, CodexField is providing a truly asset-oriented infrastructure for the next-generation creator economy.

02 CODEXFIELD: BUILDING THE NEXT-GENERATION WEB3 CONTENT ASSETIZATION PLATFORM

2.1 About CodexField

CodexField is a Web3 native asset infrastructure for content creators and developers, dedicated to realizing the rights confirmation, calling, and financialization of structured content. The project takes high-reuse content units such as code, models, Prompts, corpora, and graphics-text as its core, building unified data encapsulation and authorized calling standards, enabling them to have on-chain rights confirmation, usage tracking, and revenue mapping capabilities, thereby promoting the standardization process of "content as asset."

CodexField provides a complete closed-loop system from storage, authorization to billing, and profit-sharing, compatible with multi-chain ecosystems (BSC, ETH, Solana, Greenfield, etc.) and mainstream storage networks, supporting the definition of content access and commercial strategies through smart contracts. All content interaction behaviors, such as subscription, calling, training, etc., can achieve automated settlement and distribution on-chain, protecting the revenue rights and interests of creators and collaborators.

As a key middleware layer in the content assetization path, CodexField serves Web3 creators, AI model developers, and platform parties simultaneously, opening standard interfaces for "content as a service," building a transparent and composable content economy system, and providing a universal basic protocol for the integration of Web3 and AI.

2.2 Our Vision

The vision of CodexField is to build a globally unified on-chain content asset protocol, so that every piece of code, every model, and every call can be ights-confirmed, measured, and receive due value return.

We hope to break down the collaboration barriers between creators and developers through standardized data structures, open authorization protocols, and automated revenue mechanisms, making structured content truly a fundamental resource and value carrier in the Web3 and AI world. By launching a content asset circulation infrastructure with well-developed incentive mechanisms and sustainable growth, we aim to propel the creator economy into a new cycle of "content as capital."

2.3 Key Features

The core advantage of CodexField lies in its comprehensive system for executable, measurable, and settleable content and model assetization. Our features can be summarized as follows:

Programmable Authorization & Mandatory Verification

All content and model assets can clearly define access and usage rules through a standardized licensing language (LexDL), coupled with capability tokens (CapToken) carrying quotas, validity periods, and region/usage constraints. At the entry point, the authorization is enforced by the system's internal verification engine (Policy VM), ensuring every call is legal, revocable, and auditable.

Verifiable Execution & Unified Metering

The platform provides an integrated execution and metering environment. Call behaviors, whether in model inference, training, or content retrieval, generate unique usage receipts (UR). All resource consumption is standardized into unified metering units (MU), and supports post-facto reconciliation and recalculation when necessary, ensuring transparent and reliable settlement.

Settlement & Revenue Routing

Real revenue is automatically mapped to the on-chain settlement system (SRP). Through the Royalty Graph, revenue can be finely split based on relationships such as citation, derivation, training, or RAG retrieval. Meanwhile, the PoA consensus mechanism ensures that revenue distribution is tied to real usage, with long-term profit-sharing and access weights dynamically bound to user staking, calling activity, and content quality.

Intelligent Discovery & Indexing

All content and model assets are uniformly indexed, forming a complete graph of content and call relationships. The platform supports semantic search and structured search, providing dynamic ranking and recommendations combined with the Execution Index (CEI), helping users quickly find high-quality content and models, and providing factual input for pricing and revenue distribution.

Cross-Chain Consistency

CodexField has two types of built-in cross-chain channels. The Receipt Bridge (RB) ensures consistency of usage receipts and settlement results across multiple chains; the Mirror Bridge (MB) ensures real-time synchronization of rights confirmation and licensing status across different chain environments. Together, they provide a low-cost, strongly consistent foundation for cross-chain collaboration.

Developer & Ecosystem Tools

CodexField provides a complete toolchain: the Gitd toolchain allows developers to directly connect Git repositories with on-chain assets; the Marketplace provides asset listing, subscription, and trading interfaces; APIs, SDKs, and CLI cover the end-to-end closed loop from asset packaging, authorization to calling and settlement, helping developers quickly enter the ecosystem.

Security & Compliance

The platform builds privacy protection and compliance verification into its architecture. Through zero-knowledge proofs and verifiable archives, it protects user data while meeting audit requirements. Coupled with behavioral risk control mechanisms, it can effectively identify and defend against brushing, self-calling, or abnormal patterns; revocation and freeze statuses take effect instantly through cross-chain channels, ensuring system security and compliance boundaries.

2.4 Ecosystem Development Status

Since its inception, the CodexField project has reached several阶段性 (phase) milestones in development delivery, on-chain deployment, and partnership resources. The following are key achievements currently completed or in progress:

2.4.1 Ecological Progress & Platform Integration

A BNB Chain Hackvolution Hackathon First Prize: In September 2024, CodexField won first prize in the Infrastructure track of the Hackvolution hackathon hosted by BNB Chain. The event focused on on-chain innovative foundational capabilities, and the project was selected for review and awarded for its Gitd protocol and content rights construction trading mechanism.

B Selected for BNB Chain MVB Season 8 Acceleration Program: CodexField has been selected for the eighth round of review of the Most Valuable Builder program led by Binance Labs and listed as an official acceleration project.

2.4.1 Ecological Progress & Platform Integration

C

Deployed on BNB Greenfield, and currently one of the most used content applications: As of August 2025, CodexField is one of the top projects in terms of activity on the Greenfield storage layer, maintaining activity in platform traffic, on-chain call data, user contributions, and other dimensions.

D

Testnet Phase User Scale Data:

- Telegram Mini App **active users exceed 1,000,000**
- Cumulative wallet **connection addresses exceed 1,000,000**
- Total contract **calls exceed 2,000,000**
- Users are mainly distributed in **Southeast Asia, the Middle East, Europe, and America.**

2.4.2 Partners & Institutional Participation

Ecosystem Partners/Investors

- Gate.io Labs (Gate.io Ecosystem Lab)
- Web3Labs (Hong Kong Web3 Policy Access Accelerator)
- CoinMarketCap (CoinMarketCap Ecosystem Lab)
- YZiLabs (Former Binance Labs branch)
- KUCOIN LABS (KuCoin Investment Research Department)

Cooperation Expansion in Middle East & Asia-Pacific

CodexField has established cooperative relationships with several Middle Eastern family offices and financial advisory institutions.

The project team participated in the Hong Kong Web3 acceleration program and engaged with Hong Kong Legislative Council members and regulatory advisors.

Currently advancing resource connection and module compliance review with OSL (Hong Kong Licensed Exchange).

03 SYSTEM ARCHITECTURE

3.1 Overview

The technical architecture of CodexField has "content as asset" and the nativization of AI capabilities as its core goals, building a complete system covering rights confirmation, authorization, execution, metering, and settlement. We call the overall architecture Codex Mesh, and extend it with two major networks, Model Fabric and Agent Fabric, enabling the platform to simultaneously support the rights-confirmed circulation of structured content assets, and the native execution and value capture of AI models and agents.

In design, CodexField divides the system into five layers:

Content Layer

Encapsulates code, models, data, and multi-modal assets through unified Content Capsules, achieving cross-chain storage and verifiable rights confirmation.

Licensing & Identity Layer

Provides a programmable, revocable authorization mechanism based on DIDs and capability tokens combined with the license description language (LexDL).

Execution & Metering Layer

Integrates zk, TEE, Committee, and ML-Executor to ensure that calls to content and models generate verifiable usage receipts.

Settlement & Revenue Layer

Transparently distributes real call revenue to authors, referrers, execution nodes, and the protocol through the Royalty Graph and revenue routing.

Indexing & Discovery Layer

Establishes a unified content and model index, supports semantic retrieval, quality measurement, and recommendation, forming the basis for governance and incentives.

On this basis, we built two cross-chain mechanisms to provide interoperability for the architecture: the Receipt Bridge for cross-chain synchronization of usage receipts, and the Mirror Bridge to ensure that rights confirmation and licensing status remain consistent across multiple chains.

Ultimately, CodexField aims to achieve an integrated, verifiable economic closed loop for content and AI models, meaning any content or model can be rights-confirmed, called, and obtain traceable revenue mapping on-chain, providing a unified underlying foundation for the Web3 creator economy and the AI service market.

3.2 Content Layer

The responsibility of the content layer is to abstract code, model weights, datasets, Prompts, and multi-modal materials into atomic objects that can be rights-confirmed, referenced, measured, and settled, providing a stable entry point for subsequent authorization verification, verifiable execution, and on-chain profit sharing. For this purpose, we propose the "Content Capsule" (CC) as a unified carrier, which is encapsulated once, available across chains, addressable by fragment, and embeds license hints and metering anchor points at the encapsulation stage, transitioning content from a "data object" to a "financialized object."

Upon creation, the CC obtains an on-chain AssetId, binds the author's DID signature and content root hash, and records version pointers and derivation relationships, forming a traceable content DAG. Encapsulation is described by a human-readable manifest and a standardized blockmap: the former covers metadata, LexDL license summary, dependency and citation graph, type tags, and integrity digest; the latter maintains shard indexes, checksums, and redundancy strategies.

Different asset forms are expressed using unified subtypes such as:

And through the URI scheme Codex://<AssetId>@<version>#<fragment>, it achieves sub-fragment and range-level addressing, supporting fine-grained citation and statistics for "pay-per-use" billing.

For cross-chain availability and retrieval performance, the content layer performs multi-copy distribution and hot/cold tiering across networks like Greenfield, Arweave, Filecoin, and CESS through storage orchestration; hot fragments are cached on edge nodes and routed based on geography and latency. Upload and read procedures have built-in lightweight PoR digest verification and retry fallback, ensuring stable and reliable performance for the calling side even during network jitter or backend degradation. Shard-level deduplication and streaming pull further reduce the throughput cost of large objects, supporting progressive acquisition and verification from fragment to whole.

Integrity, privacy, and compliance are incorporated during the encapsulation period. Author signatures and Merkle Roots establish immutability; watermarks/fingerprints and perceptual hashes can be optionally enabled for multi-modal and weight assets to support post-facto forensics and infringement comparison; sensitive data uses envelope encryption and minimizes plaintext exposure; the manifest explicitly labels PII and region/industry restrictions, providing a basis for enforcement by the licensing/identity layer. For compliance scenarios requiring "deletability," the content DAG retains base tokens and invalidation pointers to maintain historical verifiability while meeting deactivation requirements.

The content layer is linked to the upper-layer economic and governance logic through graph relationships and metering. Any import, citation, or derivation forms an attributed edge on the DAG; this edge carries the citation scope and fingerprint similarity, providing fine-grained attribution evidence for the Royalty Graph at the settlement end; the execution end parses and pulls fragments using the Codex:// URI at the metering layer, normalizes the usage scope and resource consumption into MU, and generates usage receipts (UR), which become the sole credential for subsequent splitting and auditing. The asset's type, tags, quality and popularity signals, and dispute status are continuously written to the Indexing & Discovery layer (DIP), inversely affecting retrieval, recommendation, and pricing strategies.

Meanwhile, the content layer provides minimal viable specifications and a toolchain to reduce integration costs. CC v1 uses the manifest and blockmap as the protocol baseline; the SDK/CLI covers key paths for packaging, uploading, parsing, and pulling; shard verification is enforced by default, while progressive verification and end-to-end encryption are optional; version evolution supports differential synchronization and read-only snapshots; old versions are marked with base tokens to retain verifiable history. Through this unified primitive, CodexField can seamlessly integrate any content and model assets into the authorization system, verifiable execution system, and on-chain revenue system under the premise of the same format, same addressing, and same confirmation, forming a long-term stable bearing layer for Web3 and AI.

3.3 Licensing & Identity Layer

The licensing and identity layer is responsible for translating "who uses which asset, under what conditions, and in what manner" into verifiable, machine-executable rules. It weaves identity (DID), capability tokens, and the license description language into a closed loop from issuance, verification to revocation, enabling any code, model, data, or multi-modal content to be precisely constrained before calling, and seamlessly connected with metering and settlement after calling.

In the object model, an author or organization first establishes a signing subject and organizational space using a DID; for a specific asset, a license ticket is generated, using LexDL to declare permitted operations (e.g., view / clone / derive / train / infer / commercial) and their boundary conditions (time/region/industry, request frequency and token limits, budget cap, visibility, redistribution and derivative attribution, minimum quality guarantee, etc.). LexDL is compiled by the license compiler into policy bytecode and executed in a hybrid on-chain/off-chain Policy VM.

For specific callers, we designed a mechanism to issue capability tokens with constrained scopes, using the EIP-712 signature format:

- Carrying license summary
- Delegable scope
- Expiration time
- Rate and budget quota
- Call entry binding, and other information

For enterprise accounts or contract accounts, this layer supports EIP-1271 and threshold signatures (M-of-N).

When calling:

The client first submits an intent message and capability token.

The entry gateway and Policy VM perform a pre-check: verify the signature and validity period, compare the license summary, evaluate rate/budget quotas, and parse contextual restrictions in LexDL (e.g., "inference only within the EU" or "only for offline training, not commercial use"). After the pre-check passes, the execution layer (including ML-Executor) pulls the target fragment using the Codex:// URI and completes the computation.

When the metering layer generates the UR, fields such as captoken_id / policy_digest / constraints.applied are written into the receipt, forming a traceable binding between license, execution, and receipt.

If the call exceeds the license boundaries (e.g., unauthorized region or exceeded quota), the policy will directly reject it at the pre-check stage; if sensitive interfaces or derivative writes are triggered during execution, the UR will record an "out-of-bounds intent" for the settlement and governance layer to handle (rejection/arbitration/penalty).

To ensure controllability and revocability, licenses and credentials have complete lifecycle management: issuance, renewal, delegation, freezing, revocation. Revocation information is synchronized across chains through the mirroring bridge, ensuring consistency of rights confirmation and license status; high-frequency verification paths use a Permit Cache and short TTL tickets to reduce call latency, which become globally invalid once revocation or quota changes are detected. The organization side supports combined authorization with RBAC/ABAC/ReBAC:

Express team permissions through roles and attributes, and cross-entity collaborative authorization through relationships (upstream suppliers/subsidiaries/project spaces).

Critical operations (e.g., delegable commercial license, training derivative rights) require multi-signature thresholds and audit annotations.

Compliance and privacy are internalized in the licensing layer. PII and region/industry tags from the content layer are enforced by the policy; training licenses inherit the LexDL terms of the dataset, prohibiting data marked "inference only" from being used for training; for privacy or compliance-restricted scenarios, it supports zero-knowledge proof language (e.g., "prove the call is from a KYC'd financial institution" without exposing identity details), and only records the proof digest in the UR. For "deletability" requirements, the licensing layer provides freeze/invalidation pointers and a dispute arbitration entry point: once arbitration is upheld, a new policy snapshot blocks subsequent access, while historical receipts are retained to satisfy audit and recalculation needs.

The licensing and identity layer provides minimal viable components, including:

LexDL v0.9
Covers view/derive/infer/train/commercial and common restriction subsets

Policy VM
Deterministic evaluation + rate/budget counters + region/industry whitelist

CapToken
EIP-712 structured signature, supports delegation and revocation lists

DID/Org
Individual and organizational spaces, key rotation, threshold signatures

On this basis, the SDK/CLI provides common operations: issue | delegate | renew | revoke | inspect, and is deeply integrated with gateway pre-check and metering receipts.

Through this layer, CodexField translates "readable license agreements" into "executable access control" and precisely aligns every legal call with subsequent metering and profit sharing.

3.4 Execution & Metering Layer

The execution and metering layer is responsible for translating "authorized usage intent" into verifiable execution and producing a unique, trustworthy metering credential. It uniformly schedules multiple execution domains (including AI model execution), normalizes real resource consumption into standard metering units, and uses the Usage Receipt (UR) as the source evidence for settlement and governance, ensuring that "who did what, to which assets, when, and how" can be independently audited and replayed.

In the object model, any call carries the license and identity context (DID, CapToken, LexDL policy digest) into the execution entry. The system selects the execution domain based on SLA, cost, and compliance constraints:

First is the **zk Domain**

Which produces zero-knowledge proofs for deterministic subgraphs, suitable for lightweight inference and rule-based tasks.

Second is the **TEE Domain**

Which produces remote attestations for large models or non-deterministic processes, covering acceleration scenarios like GPU/NPU.

Third is the **Committee domain**

Which forms a consensus receipt through multi-node threshold signatures, serving as a low-cost, widely covered fallback path.

Fourth is the **ML-Executor**

Which brings inference and training into the same metering system, generating Infer-UR and Train-UR respectively.

For complex tasks orchestrated by Agents, the system generates fine-grained URs for each step and aggregates them into a batch root at the task level, ensuring the entire chain is auditable.

To avoid metering fragmentation across different loads, the layer introduces a unified metering unit (MU), normalizing GPU memory-seconds, operator steps, LLM tokens, I/O bytes, and memory residency time into a configurable cost model. The coefficients of the metering formula are automatically adjusted based on regional cloud prices, hardware generations, and SLA tiers, and the hash of the current "Pricing Surface" is recorded in the UR to support post-facto reconciliation and recalculation.

Standard fields of the UR include: asset_refs (AssetId/fragment), model_id/dataset_ids, captoken_id, policy_digest, proof_type/digest, μ (MU), tokens_in/out, latency, exec_domain, region, and timestamp; if Retrieval-Augmented Generation (RAG) is involved, it also includes the fingerprint and scope of the content fragments referenced this time, used for subsequent royalty attribution.

All raw URs are first aggregated by the Receipt Collector on the caller side into a Merkle tree per user and time window, and the root is calculated; the raw details are written to a persistent backend (e.g., Greenfield/Arweave); the root and proof digest are submitted to the settlement chain via the Receipt Bridge (RB). This ensures the determinism of on-chain settlement while avoiding moving large amounts of detail directly on-chain. In case of disputes, the system can replay and verify based on the UR root and path: zk tasks re-verify the circuit, TEE tasks check the attestation and measured values, committee tasks trigger re-signing and sampled recalculation; if the dispute is upheld, the relevant executor is slashed and revenues are re-settled.

To further ensure scheduling and stability, we dispatch requests to the most suitable execution domain and geographically proximate resource pool; semantic caching and deduplication mechanisms automatically cache highly repetitive Prompts; the retrieval/rewriting/assembly process of RAG is also metered. Elastic scaling and degradation paths are enabled under burst traffic, and the UR marks the QoS tier. Quotas and budgets are issued by the licensing layer; rate limiting and budget consumption are jointly counted by the entry gateway and Policy VM; requests exceeding limits are rejected or queued at the pre-check stage. To guard against attack vectors like "contract self-calling usage," "circular references," and "forged commercial usage," the layer internally deploys synthetic traffic detection and account clustering; abnormal URs are down-weighted or filtered out, and their characteristics are written to the discovery index for subsequent governance reference.

Privacy and compliance requirements are also guaranteed during execution: training tasks mandatorily verify dataset license terms, data marked "inference only" cannot be used for training; region and industry whitelists are enforced at entry; when privacy protection is needed, only the digest of the zero-knowledge assertion is put on-chain for the UR, while the full proof and audit materials are kept in verifiable storage. For scenarios requiring "deletability," the system provides freeze/invalidation pointers to block subsequent access, while already generated URs are retained as historical facts to satisfy regulatory and audit requirements.

The execution and metering layer provides standardized SDK/CLI and gateway protocols, covering key paths such as intent signing, license pre-check, fragment parsing (Codex://), execution submission, UR query, and reconciliation export; the MVP version prioritizes launching the Committee domain and ML-Executor's inference metering, reserving slots for zk and TEE, ensuring the main path from authorization to settlement is established as soon as possible under the premise of "same metering, same receipt." Through this layer, CodexField transforms "calls" into "measurable, provable economic events," providing a unified factual benchmark for subsequent revenue splitting, risk governance, and ecosystem incentives.

3.5 Settlement & Revenue Layer

The role of the settlement and revenue layer is to transform the verified call facts from the execution and metering layer into on-chain auditable value distribution. This layer takes the Usage Receipt (UR) as the only trusted input, maps the real commercial revenue generated by users into the Royalty Graph, and completes profit splitting and flow payments through the Revenue Router. The entire process ensures that call behavior is not only traceable and verifiable but also directly translated into revenue signals, providing creators, model developers, data contributors, and execution nodes with a stable, transparent, and anti-malicious economic closed loop.

In design principles, the settlement and revenue layer emphasizes three core elements:

1. **Anchored by Real Revenue** – All distributions are based on actual settleable revenue generated by real calls, avoiding traditional patterns of "brushing calls" and "idling for control."

2. **Auditable & Recalculable** – Each settlement cycle has a clear data structure (UR Root, Pricing Surface hash, Royalty Graph edge weights), allowing external independent verification and replay calculation.

3. **Anti-Malicious & Compliant** – Through dispute mechanisms and reconciliation systems, it resists fake calls, malicious citations, and unauthorized use, ensuring fund safety and compliance.

The core tool for revenue attribution is the Royalty Graph. It uses content assets, models, and datasets as nodes, and citations, derivations, training, RAG calls, or Agent orchestration as edges, forming a weighted directed graph. Edge attributes include not only the citation relationship but also similarity index, coverage scope, call depth, and license terms.

In actual calculation, the revenue generated by calls goes through the following steps:

Deduct execution cost

Calculate the cost based on the MU (Metering Unit) recorded in the UR, the execution domain (zk, TEE, Committee, ML-Executor), and the current Pricing Surface.

Profit Sharing Split

The remaining distributable revenue is split among different entities according to a formula. The creator, as the direct contributor, receives the core share; citation and derivation relationships receive attenuated distributions based on depth and similarity; dataset contributors receive profit shares based on weights; execution nodes receive execution compensation; finally, the protocol charges a certain percentage as a system fee.

This design ensures incentives for direct authors while preserving continuous revenue rights for upstream creators and data providers in the citation chain.

To adapt to different types of call scenarios, the settlement and revenue layer provides multiple payment modes:

Instant Settlement

Suitable for high-frequency, low-value calls, such as API inference requests; the system completes aggregated payments within short cycles.

Stream Payment

Subscriptions and long-term calls are settled through continuous payments; users can pause or renew at any time, and payment is automatically cut off upon arrears.

Batch Settlement

Enterprise customers can settle in bulk after confirming an EIP-712 invoice according to daily, weekly, or monthly billing periods, facilitating integration with ERP and invoicing systems.

Additionally, the system supports multi-asset payments, including both mainstream stablecoins and $CODEX. If $CODEX is used as the payment unit, the settlement contract can provide instant exchange or netting accounting functions.

To avoid short-term incentives that rely solely on traffic, the settlement and revenue layer introduces a new mechanism into the profit-sharing logic: users or institutions can obtain priority access bandwidth by staking $CODEX or mainstream assets and receive weight bonuses in revenue distribution.

The revenue weight is determined by three parts:

Staking Share

The size of the staked funds.

Usage Score

The number of URs generated by actual calls and retention behavior.

Quality Factor

Quality evaluation from the Discovery and Indexing Layer (DIP), audit results, and complaint rate.

Through this mechanism, while ensuring real revenue, the platform can also guide long-term participants and high-quality contributors to receive more sustainable incentives.

The settlement layer provides enterprise-grade reconciliation capabilities. Each billing period generates an EIP-712 invoice with the UR Root, Pricing Surface hash, and split results, facilitating verification by institutions with their internal systems. All UR details and split evidence are archived in verifiable storage (e.g., Greenfield/Arweave), while only events and root digests need to be kept on-chain.

This "two-tier storage" solution ensures that settlement can be handled lightly while providing complete review materials in case of disputes. For call batches that fail to meet SLA, the revenue router supports delayed payment or partial deductions, and a "dispute withholding ratio" can be set to ensure fund safety during disputes.

The settlement and revenue layer designs a complete dispute mechanism. After a UR is on-chain, it enters a fixed dispute window; anyone can initiate a challenge during this period, requiring them to stake a bond and submit evidence.

The system selects different review methods based on the execution domain:

- zk domain calls are reviewed by re-verifying the circuit.
- TEE calls are reviewed by comparing attestation and measured values.
- Committee domain calls trigger multi-node sampled recalculation.
- RAG and derivative relationships are reviewed by comparing fingerprints and scopes.

If the challenge is upheld, the system will re-settle the batch, slash the violating party, and reward the challenger. The challenge result is recorded in the Discovery and Indexing Layer (DIP) as a negative signal for subsequent recommendations and quality scoring.

The core parameters of the settlement and revenue layer (such as split weights α, β, γ, δ, τ, depth decay λ, minimum threshold, and dispute window T, etc.) are all adjusted through the governance mechanism. Upgrades use proxy contracts and versioned identifiers to ensure compatibility and recalculability of new logic with old URs. All parameters, votes, and change events are publicly written to the DIP for community and external audit use.

The settlement and revenue layer is CodexField's value distribution engine. Through the combination of UR, Royalty Graph, and Revenue Router, it achieves a verifiable path from call behavior to revenue split; through PoA, it binds access incentives to long-term participation; through reconciliation, audit, and dispute mechanisms, it provides transparent, fair, and anti-malicious revenue guarantees for enterprise and community users. This layer not only ensures the value realization of content and model assets but also establishes a reliable economic foundation for the entire ecosystem.

3.6 Discovery & Indexing Layer

The discovery and indexing layer is the information hub of CodexField, responsible for unifying scattered content assets, models, datasets, evaluation results, and call behaviors into a network graph that is searchable, measurable, and recommendable. It is not only the first entry point for developers and users into the platform but also an important input source for governance, incentives, and pricing. Through continuous updates, the discovery and indexing layer keeps CodexField transparent, interpretable, and capable of dynamic evolution.

Architecturally, the core is the content graph index, which treats all Content Capsules, model versions, datasets, and evaluation results (EvalId) as nodes, while mapping citations, derivations, training, RAG retrievals, and Agent calls as edges, thus forming a multi-dimensional relationship graph. Each node has clear attributes, such as version information, license summary, quality tags, and compliance status; each edge records call frequency, citation scope, and similarity index, giving the entire graph complete traceability.

At the user interaction level, the index provides two main types of retrieval:

The first is semantic retrieval, based on vector index and semantic tags, allowing cross-language and multi-modal natural language queries, helping users quickly find target content or models.

The second is structured retrieval, aimed at institutions and professional users, enabling filtering based on license terms, execution cost, regional restrictions, and dependencies. For example, an enterprise can directly search for "models commercially usable in the Asia-Pacific region with inference cost below a specified threshold," thereby reducing the complexity of compliance and performance evaluation. Ranking combines quality score, execution cost, compliance, and historical stability to ensure recommendation results are both efficient and reliable.

The discovery and indexing layer also undertakes the function of quality measurement. The system aggregates multi-dimensional signals such as call data, SLA achievement rate, disputes and audit results, and user feedback into an Execution Index (CEI). This index is not only the core basis for recommendations and pricing but also the quality factor in the PoA incentive mechanism. Through the CEI, CodexField can continuously identify high-quality content and models, highlight them in ranking and incentives, which in turn encourages contributors to continuously optimize asset quality.

Beyond basic retrieval and measurement, this layer also provides pricing and decision support. For any search result, the system directly displays license restrictions (LexDL summary), required CapToken, estimated execution cost, and suggestions for the optimal calling path. In more complex scenarios, such as RAG or Agent calls, the indexing layer can generate combined recommendations of "model + dataset + content fragment" and provide estimated execution index and cost hints. This allows users not only to "find resources" but also to "know how to use resources optimally."

As part of platform governance, the discovery and indexing layer ensures transparency and reviewability. All ranking factors, CEI updates, arbitration results, and governance parameter changes are recorded and open for external query. External participants can recreate the ranking logic at a certain point in time based on UR roots and index snapshots, thereby verifying the fairness of the recommendation process. Furthermore, when the system detects patterns like brushing, self-citation loops, or malicious similarity forgery, the indexing layer sends signals back to the licensing and execution layers, triggering down-weighting or quota restrictions, ensuring the health of the ecosystem.

To further ensure compliance and privacy, the discovery and indexing layer follows the principle of "minimum disclosure." For sensitive or private assets, only the license summary and necessary tags are exposed, without leaking call details. If upstream uses zero-knowledge assertions (e.g., KYC compliance), the index only stores the proof digest and validity period, not identity information. External interfaces provide standardized GraphQL/SQL-like query and explanation APIs, supporting event subscriptions for quick integration by developers and institutions. In the MVP stage, the system will prioritize implementing the basic graph index, semantic and structured retrieval, CEI v1 index, and the closed loop of cost hints and combined recommendations, ensuring early users can benefit immediately.

3.7 Cross-Chain Channels

CodexField pursues two types of consistency in a multi-chain environment: consistency of value settlement and consistency of rights confirmation/licensing view. The former requires that usage facts generated by different execution domains can be uniformly and auditably aggregated on the same settlement plane; the latter requires that any call location can instantly obtain the latest and reliable confirmation and licensing status. For this purpose, we split cross-chain interoperability into two independent yet cooperative channels: the Receipt Bridge (RB) and the Mirror Bridge (MB).

- RB focuses on transmitting the UR batch reports aggregated by the metering layer and necessary proof digests.
- MB focuses on mirroring the minimal valid state of AssetId / LicenseId / CapToken.

Both follow the design principles of data minimization and strong reviewability, ensuring that settlement and authorization remain deterministic and explainable across domains.

3.7.1 Receipt Bridge (RB)

The semantics of the RB are very clear:

Converge the call details that have completed license verification and execution metering over a period of time into a verifiable report and securely deliver it to the settlement mainchain.

The Receipt Collector on the metering side aggregates the raw URs into a Merkle tree per user/business dimension within each settlement cycle, calculating the UR batch root; it also generates a summary object in an agreed format to solidify metadata such as the hash of the current Pricing Surface, execution domain identifiers (zk/TEE/Committee/ML-Executor), sample size, and statistics.

During the cross-chain process, only the "root + summary" is carried on-chain; the raw UR details are archived long-term in verifiable storage (e.g., Greenfield/Arweave) indexed by the report, thus achieving a cost/security balance between on-chain determinism and off-chain reviewability.

The RB provides two switchable paths at the verification layer:

When the batch amount and audit requirements are high, the settlement chain verifies the source chain's block headers and event inclusion using a light client, with finality and security guaranteed by the source chain's consensus.

When pursuing throughput and cost, the RB adopts an optimistic path: it first accepts the batch root and opens a challenge window T; any participant can initiate a challenge by providing a Merkle branch and counterevidence; if upheld, the batch is rolled back and the submitter relay is slashed.

To handle edge cases, the RB maintains a monotonically increasing epoch/nonce and a "finality threshold" for each source chain; batches not meeting the threshold remain in a pending state and do not enter the settleable set; if the source chain undergoes a reorganization, the settlement contract automatically invalidates and replaces the corresponding batches; duplicate or reverted submissions are rejected, fundamentally blocking replay and double-spending.

At the operational level, the throughput and latency of the RB are determined by multiple factors:

Batch size and submission rate, the value of challenge window T, and the gas status of the target chain. To keep the system available during congestion and volatility, the RB has built-in backpressure and throttling logic.

When resources are tight, the Collector automatically reduces batch size and extends T, ensuring settlement continuity on a "slower but safer" curve.

For institutional billing periods and audit needs, every state transition of the RB's root transactions is written to the Discovery and Indexing Layer (DIP); external auditors can use this to trace back the root set and verifiable evidence of any billing period. When a dispute occurs, only the archived details need to be retrieved off-chain according to regulations, and sampling replay can complete the review. zk tasks undergo secondary circuit verification, TEE tasks check attestation measured values, committee tasks trigger re-signing and sampled recalculation; the review results are written back to the DIP and drive the re-settlement and penalty process at the settlement layer.

3.7.2 Mirror Bridge (MB)

Unlike the RB, which focuses on "facts," the MB focuses on "state." Its goal is to enable the call location to instantly and reliably query the currently valid rights confirmation and licensing view without moving large objects and historical traces.

For this purpose, the MB only synchronizes the minimal valid digest of three types of objects across chains:

For **AssetId**, synchronize create/freeze/invalidate flags, version pointers, and parent relationship digest, so the caller can confirm whether the asset can be referenced and whether its version lineage meets license constraints.

For **LicenseId**, synchronize policy digest, effective/expiration time, and region/usage tags, enabling the Policy VM to complete most pre-check vetos locally.

For **CapToken**, synchronize events such as issuance, renewal, delegable scope, freeze/revocation, and quota changes, enabling rate limits and budget constraints to take effect instantly at the call location.

The consistency strategy of the MB is centered around versioned merging:

Each mirrored object carries a monotonically increasing version and updated_at; the destination chain merges them upon receipt using a "higher version priority rule, ensuring that late arrivals do not overwrite newer updates.

For revocation/freeze events, the system adopts a revocation-first safety semantics; any subsequently arriving old state will be directly discarded.

On the verification side, similar to the RB, the MB also supports both light client and optimistic modes, prioritizing the former in high-value scenarios. To balance privacy, compliance, and cost, the MB never transmits the full policy text or data details across chains; when the caller needs finer-grained license terms, the Policy VM can perform a source check based on the policy_digest. For enterprise private domains or restricted assets, the MB allows enabling a "private mirroring domain," exposing only necessary boolean bits and fingerprints, satisfying compliance audits while avoiding leakage of policies and organizational structures.

The coupling between the MB and the Licensing/Identity Layer (LIL) is manifested at the very front of the call path. When a request arrives at the entry gateway, the local mirror snapshot is queried first; if the snapshot shows the CapToken is revoked, quota exhausted, or region/usage mismatch, the request is directly rejected at an early stage, suppressing unauthorized access and brushing risks at the root. To reduce consistency latency, the MB is complemented by an optional push channel and short-term TTL cache on top of on-chain events: when a revocation or freeze occurs on the source chain, the event is quickly pushed to subscribed destination chain instances, then solidified by a mirror transaction; if the mirror transaction is not seen after the TTL expires, the system reverts to source verification mode and raises the audit level, trading a small amount of latency for strong consistency.

3.7.3 Coordination

In overall coordination, the RB and MB govern separately the value closed loop and the control closed loop:

The UR batch report delivered by the RB contains policy_digest and captoken_id digests; the settlement layer can optionally sample and compare against the MB's latest mirror before attribution to confirm the call's license view and filter out possible unauthorized usage.

When the MB handles revocation/freezing, it writes the event to the DIP and triggers risk control switches in the execution and metering layers, blocking subsequent requests from entering the execution plane.

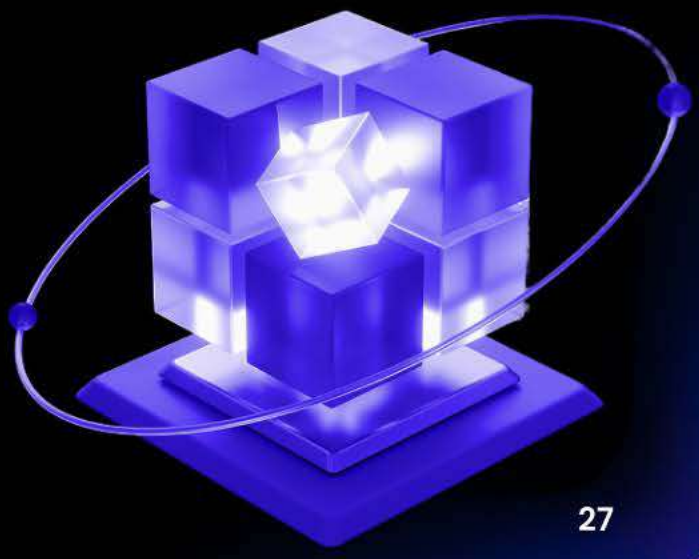
Under abnormal conditions (source chain instability, relay anomaly, or target chain congestion), both channels enter a safety-first degradation mode:

The RB switches to a more conservative optimistic path and extends T, pausing the effectiveness of new settleable roots if necessary.

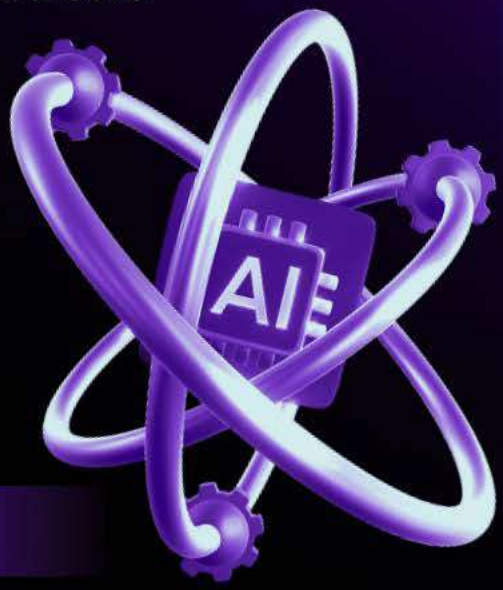
The MB will only accept revocation/freeze type mirrors, delaying other updates, and the call location falls back to the source verification strategy.

All degradation, circuit breaker, and recovery events are recorded in the DIP, becoming historical facts for external audit along with changes in governance parameters.

Through this design of "separating facts and states, coordinating digests and proofs," CodexField achieves low-cost cross-domain consistency in a multi-chain context: value settlement is strictly closed-loop around UR batch reports, while rights confirmation and licensing are quickly mirrored around minimal valid states; both retain reviewable error correction windows and governable operating parameters, serving both daily high-frequency business and smoothly handling high-value, strong compliance, and strong audit scenarios.



3.8 Model Fabric & Agent Fabric



To incorporate "AI models/agents" into the same verifiable, settleable economic closed loop as content assets, CodexField introduces two cross-cutting networks on top of the Mesh architecture: Model Fabric and Agent Fabric. The former uniformly manages the full lifecycle of models and data, while the latter is responsible for orchestrating retrieval, inference, external tools, and contract calls into measurable task chains.

3.8.1 Model Fabric

Model Fabric brings self-developed and third-party models under unified governance through a closed loop of "registration, training/fine-tuning, inference, evaluation." Each model obtains a ModelId/VersionId upon creation; weights and adapters are encapsulated as CC-MW (Model Weights/Adapter), datasets are encapsulated as CC-DS, an AssetId is passed on-chain, and they are bound to the author's DID, license summary (LexDL), and lineage relationships. Both training and inference produce verifiable usage receipts (UR) as the sole evidence for settlement and governance.

Core Components

Model Registry

Registers ModelId/VersionId, model card (capabilities, limitations, compliance tags), release channel (stable/canary).

Dataset Ledger

Data provenance, license terms, PII/region annotations, and traceability digest; supports contribution weight accounting (for upstream profit sharing).

Train/Fine-tune Pipeline

Supports parameter-efficient fine-tuning like LoRA/PEFT/Adapter, built-in DP-SGD, PII desensitization, and license verification (use.train); produces Train-UR (contains dataset_ids, μ, proof digest).

Inference Runtime

Multi-backend (GPU/CPU/accelerator) hot-swapping and elastic scaling, Prompt normalization and semantic caching; produces Infer-UR (contains tokens_in/out, μ, latency, proof digest).

Eval & Red-team

Writes task sets/security evaluations/bias metrics to the DIP, providing objective signals for ranking, pricing, and governance.

Training, fine-tuning, and inference are all executed through the VCM's ML-Executor, uniformly measured in MU (memory-seconds/operator steps/tokens, etc.); the generated Train-UR/Infer-UR are aggregated by the RB to the settlement mainchain, and the SRP completes three types of attribution on the Royalty Graph based on this:

Data→Model (training use)	*Model→Model (derivative/adaptation)*	*Content→RAG (retrieval citation)*
Profit shares to data contributors.	Profit shares to parent model authors.	Profit shares to the cited content authors.

All evidence and scores flow back to the DIP, continuously influencing recommendations, pricing, and the quality weight in PoA.

Model Fabric enforces license and security policies during onboarding and runtime: mandatory verification of CC-DS LexDL terms before training (prohibiting data marked "inference only" from training use), region/industry whitelists, k-anonymity, and minimal sampling; watermarks/provenance tags can be optionally enabled on outputs for forensics and tracing; for enterprise scenarios, it supports "private execution domains" and "audit read-only snapshots," while preserving zero-knowledge assertion digests, balancing privacy and verifiability.

3.8.2 Agent Fabric

Agent Fabric further organizes "retrieval, planning, model inference, external tools/contracts, summary feedback" and forms a cycle of plan, act, observe. Each step is executed under license and budget constraints, and the metering layer generates fine-grained URs; after task completion, URs are aggregated in batches on the RB side, giving the entire chain auditable and reliable characteristics.

Agents express available tools and models through a "skill graph," performing routing and dynamic replacement based on the three objectives of cost/latency/quality:

The planning phase selects candidate paths based on the DIP's CEI index, license availability, and Pricing Surface.

The action phase can switch execution domains among ZK/TEE/Committee/ML-Executor, adhering to budget and rate limits.

The observation phase records context and failure retries, and writes anomalies and backoff strategies into the UR for subsequent review and recalculation.

In RAG and multi-step reasoning, Agent Fabric synchronously manages vector indexes, caching, and fragment trimming, ensuring "citation is metered, citation is attributed."

Agents load LexDL's contextual restrictions (region/industry/usage) at the entry point and execute Prompt strategies and output filtering; when involving contract write operations, fund transfers, or high-risk API calls, they must pass triple verification of "intent signature + budget threshold + audit hooks"; when detecting patterns like self-reference loops, abnormal similarity, or contract self-calling volume, Agent Fabric will down-weight or throttle, and record the characteristics and handling results to the DIP.

3.9 Native Ecosystem Features

3.9.1 Gitd Tool

Gitd is CodexField's integrated toolchain for developers, with the mission of seamlessly mapping traditional Git workflows to on-chain asset logic. Through Gitd, code, models, datasets, and retrieval indexes are no longer just development artifacts but economic units that can be rights-confirmed, authorized, called, measured, and settled. Developers can integrate every commit and release into a complete "asset-call-revenue" closed loop without changing their existing habits.

The core of Gitd lies in standardizing the conversion of Git repositories into Content Capsules (CC). After developers complete code or model updates locally, they can simply call Gitd commands to package the working tree into a CC. During packaging, the toolchain automatically generates:

manifest.yml

Records author DID, version information, dependency/citation relationships, license hints (LexDL summary), and integrity hash.

blockmap.json

Manages data shard indexes, checksums, and redundancy strategies.

Build Digest

Includes SBOM (Software Bill of Materials) and partial fingerprints, ensuring the verifiability of citations and subsequent attributions.

Subsequently, Gitd parses LICENSE, NOTICE, or SPDX information in the repository, automatically generating a LexDL draft to help developers transform "text licenses" into "executable licenses." After review, the author confirms with an EIP-712 signature; if necessary, CapTokens with quotas and validity periods can be issued to collaborators or testers via gitd grant. During the publishing stage, the CC is uploaded to the storage orchestration layer, generating an on-chain AssetId; simultaneously, the Mirror Bridge (MB) synchronizes the minimal license state, and the Indexing Layer (DIP) incorporates the new asset node, providing an entry point for retrieval and recommendation.

Gitd can establish an evidence chain for subsequent metering and settlement:

During the packaging stage, Gitd identifies third-party dependencies and internal modules in the project, generates citation edges based on AST and partial hashes, and records coverage scope and similarity. This way, when call behavior generates URs and enters settlement, the Royalty Graph can transparently split revenue to direct authors, cited authors, and data

For model and data resources, Gitd encapsulates them as CC-MW (Model Weights/ Adapter) and CC-DS (Dataset) respectively, and clearly distinguishes between infer and train permissions at the licensing layer to avoid common violation usage scenarios.

For RAG or multi-modal scenarios, Gitd writes fragment-level citation information into the manifest, making "citation is metered, citation is attributed" an automatically executed on-chain logic.

In this way, developers can ensure that when project outputs are called, revenue is automatically distributed along the complete graph chain without additional configuration.

Call Verification & Dispute Support

Gitd provides developers and institutions with a toolset for "error prevention" and "dispute resistance."

Dry-run

Allows simulating calls locally or in a test execution domain, generating offline URs. Developers can preemptively verify the correctness of MU (Metering Unit), input/output token counts, policy digest, and CapToken ID.

Local Replay

When encountering disputes, Gitd can export review materials (samples, routes, proof digests, Merkle paths) based on a specific UR root, enabling quick submission of counterevidence to support the rollback and re-settlement by the Receipt Bridge (RB).

Reproducible Builds

Gitd locks dependencies and environment fingerprints by default, and includes build digests in publish events, ensuring external partners or auditors can reproduce and verify the artifacts.

Team & Enterprise Integration

In team collaboration and enterprise application scenarios, Gitd provides enhanced features for organizations:

Organization Space

Supports multi-dimensional permission configuration like RBAC, ABAC, ReBAC; critical operations such as "commercial license delegation" or "training right authorization" require multi-signature thresholds to take effect.

Reconciliation & Auditing

After the billing period ends, Gitd can export EIP-712 invoices containing UR roots, Pricing Surface hashes, and split results, directly interfacing with ERP and financial systems.

Private Mirroring Domain

For sensitive models or compliance-restricted data, Gitd allows teams to enable private mirroring, synchronizing only necessary boolean bits and fingerprints, with the rest verified by the caller via source check, protecting both privacy and compliance.

Toolset & Command Entry

To reduce the learning curve, Gitd abstracts the core workflow into a few commands:

Asset Lifecycle
gitd pack / sign / license / publish

License Management
gitd grant / revoke / mirror

Calling & Verification
gitd dry-run / invoke

Reconciliation & Disputes
gitd invoice / settle / dispute:bundle

Through this set of commands, developers can navigate the complete chain from artifact packaging, license publishing, call execution, to revenue distribution and dispute handling.

3.9.2 CodexField Marketplace

The CodexField Marketplace is the core of the platform's application and content circulation. It is not just a display and trading market but a comprehensive entry point for content on-chain, AI creation, authorized calls, and revenue flow. Through deep integration with the Gitd toolchain, storage networks (BNB Greenfield, IPFS, etc.), AI generation services, and user incentive systems, it forms an end-to-end closed loop for developers and creators.

Content & Creation Access

The Marketplace supports various types of content onboarding and distribution. Users can upload code, images, videos, or documents, storing them directly in decentralized storage networks, or import from GitHub and package into CCs (Content Capsules) with one click. The platform also has a built-in online version of Gitd to help developers quickly complete packaging, rights confirmation, and publishing. For non-professional users, the Marketplace provides AI creation tools, such as AI image generation, NFT generators, stylization conversion, and Prompt template calls, lowering the barrier to creation.

Asset Listing & Call Paths

After all content and models are listed on the Marketplace, they automatically generate an AssetId and are associated with a license (LexDL) and capability token (CapToken). Creators can define access rules (subscription, pay-per-call, commercial authorization, etc.). Call behavior instantly generates URs (Usage Receipts), entering the settlement and profit-sharing path. Thus, the Marketplace becomes the first-layer execution interface connecting creation, use, and revenue.

Interaction Experience & Distribution Mechanism

The Marketplace provides users with an intuitive interactive experience:

AI Tool Entry

Modules like AI NFT Generator, style transfer, AI painting interface with infrastructures like LLM and Vercel; users only need to upload files or select preset Prompts to generate results.

Categorization & Recommendation

All content is categorized and displayed by type, call count, and user rating; combined with the DIP's retrieval and ranking algorithms, it ensures high-quality resources get greater exposure.

Incentive Mechanism

Creators and content providers earn points and revenue shares upon listing, calling, and being cited; user behaviors like check-ins are also converted into point rewards, used for ecosystem participation and leaderboard rankings.

The Marketplace interacts with the CodexField Ranking Board, forming an open and transparent reputation and points system. The contributions of content Providers are not only reflected in revenue but also accumulate over time through leaderboards and reputation points, becoming important basis for subsequent recommendations, calls, and incentives. As more AI tools and third-party integrations are added, the Marketplace will evolve into a diversified hub for content asset financialization and AI service transactions.

04 Content Asset System

CodexField, based on the principle of "rights confirmation is value," incorporates various types of content such as code snippets, AIGC outputs, AI models, knowledge strategies, and Prompt templates into the on-chain asset paradigm. Through atomic-level rights confirmation, behavior mapping, revenue tracking, and combinatorial calling, it achieves content ownership labeling and programmable value release.

4.1 Content Asset Type Support

CodexField has built a content asset classification and rights confirmation system covering generation source, semantic layer, and usage right paths. It supports atomic labeling of heterogeneous data structures and introduces an on-chain fingerprint registration mechanism to ensure content objects have verifiable uniqueness and evolvability.

All assets generate a UID via multiple hash algorithms before being put on-chain and are anchored to the corresponding asset type, structural model, generation path, license mode, and calling strategy, laying the foundation for subsequent value mapping and financialized recombination.

The platform natively supports the following major content asset types:

Structured Code Assets

Includes raw code snippets, function modules, smart contract skeletons, component constraint templates, etc., expressed using standardized AST (Abstract Syntax Tree) structures for easy diff comparison, module refactoring, and reuse right binding. These assets have clear call boundaries and are suitable for SDK integration, automated deployment, and DSL generation pipelines.

AI Model Related Assets

Covers model weight files, training datasets, inference API wrappers, and upstream-downstream dependency mappings. Supports versioned tracking and authorization management of model behavior; combined with hash snapshot mechanisms, it can be used for model rights confirmation, auditing, and call permission control.

AIGC Multi-modal Generation Assets

Includes text, image, audio, video, and other modal generation content. All assets are bound to generation instructions (Prompt), models used, sampling parameters, and context. CodexField provides a native generation traceability system to verify if the content was generated by an on-chain compliant model and map it to the source Agent's behavior history.

Professional Knowledge Assets

Highly directional structured cognitive achievements such as trading strategies, research reports, engineering drawings, course tutorials, etc., possessing citation relationships, version evolution chains, and trusted signature mechanisms. These assets are often used as AI training data or Agent input corpora and are incorporated into the platform's data market and knowledge intermediary circulation network.

Prompt Template & Snapshot Assets

Includes structured Prompt instruction sets, intent model mappings, historical generation snapshots, etc., serving as semantic positioning entry points before AI model calls. CodexField supports weight distribution and performance scoring for these assets; they can enter recommendation paths and reward distribution pools based on usage effectiveness feedback.

All content assets on the platform possess a series of common capabilities, including:

- On-chain rights confirmation and behavior trajectory binding
- Composable reuse and logical calling
- Value mapping based on revenue pools
- Compatibility with cross-protocol asset nesting

Under the CodexField architecture, content will be able to become programmable financialized units with contextual behavior history and dynamic calling rights, usable in diverse scenarios such as authorized calls, joint training, Agent embedding, and RWA risk mapping.

4.2 Collaboration Network

CodexField's collaboration network is based on modular semantic annotation, AI-driven recognition, and on-chain governance logic, aiming to promote the flow of value from static determination to dynamic circulation for content and model assets. It emphasizes not only asset uniqueness and verifiable rights confirmation but also the orchestrability, measurability, and profit-shareability of citations, derivations, training, and calls during collaboration. Using content capsules as unified atomic objects, combined with license description language and capability tokens, the system defines the boundaries and constraints of collaboration. With the help of the discovery and indexing layer and the royalty graph, new collaborative relationships are沉淀 (precipitated) into auditable evidence, ultimately carried by usage receipts and contribution receipts into the settlement and revenue routing, forming a self-governing, evolvable collaborative value network.

4.2.1 Extended Rights Confirmation from Code to Structured Content

CodexField natively supports multiple structured asset types, including code snippets, modules, Prompt templates, tutorials, model weights, and datasets. All objects are encapsulated as CCs and assigned an AssetId upon onboarding, binding AST/IR fingerprints, generation parameters, and dependency/citation relationships in the manifest.yml. Through this mechanism, the system can build a complete version DAG and asset lineage, making every version and every derivation traceable.

Furthermore, CodexField supports extending from single files to cross-language, cross-modal multi-layer content packages and provides fragment-level addressing via Codex://<AssetId>@<version>#<fragment>. This design ensures unique asset identification while providing the technical basis for "meter by fragment, attribute by citation."

4.2.2 Behavior Record-Driven SocialFi Profile System

The collaboration network standardizes all user behaviors within the system, including asset upload, license authorization, calling, collaborative review, curation, and red team testing. These behaviors are recorded as URs or CRs (Contribution Receipts) and bound to the user's DID, forming a dynamic Social Graph profile.

This profile functions in two aspects

Ranking & Distribution

Serves as the basis for content recommendation, ranking, and display; the Social Graph provides quantifiable signals like call intensity, retention rate, and dispute records.

Reputation Input

In Agent collaboration or training tasks, the Social Graph serves as a reputation credential input, influencing the task's resource allocation and call priority.

All profile data is indexed and snapshotted long-term by the DIP, thus possessing the capability for cross-platform verification and incentive export.

4.2.3 AI-Driven High-Quality Content Identification & Incentives

CodexField's collaboration network integrates an AI recognition engine, which, combined with the evaluation and red team testing results from Model Fabric, establishes a multi-factor quality scoring system for content and model assets. Scoring dimensions cover semantic matching degree, historical call feedback, call frequency and stability, user ratings, and compliance risks.

These scoring results are written to the DIP for content recommendation and dynamic pricing. Simultaneously, the scoring results are directly linked to the revenue pool, serving as the quality factor in PoA, influencing the long-term profit-sharing weight of assets. The recommendation engine supports configurable filters and adaptive distribution strategies, ensuring assets at different stages and of different types receive reasonable exposure and incentives.

4.2.4 Joint Governance Model

CodexField introduces a **"DAO + AI"** dual-track model in its governance mechanism. The DAO layer is responsible for governance proposals and voting, including revenue split parameters **(α/β/γ/δ/r)**, citation depth **decay coefficient (λ)**, **dispute window (T)**, black/grey list strategies, and risk control thresholds; the AI layer is responsible for providing ranking suggestions, performing anomaly detection and threshold tuning, and outputting interpretable analysis basis, such as feature contribution and audit trails.

All governance events, parameter changes, and arbitration results are written to the DIP, ensuring openness, transparency, and reviewability. License and identity control are enforced through **LexDL and CapToken** in the Policy VM, while cross-chain consistency is guaranteed by the **RB and MB channels**. This way, governance decisions can remain unified in a multi-chain environment and be friendly to external audits.

4.2.5 Content Asset Entry into Licensed Revenue Path & Financial Mapping Layer

CodexField supports bringing any compliant content asset into the financialization path. Asset entry typically involves three steps:

Licensing & Bandwidth

Asset holders define license rules through LexDL; users obtain access and calling rights through staking and the PoA bandwidth allocation mechanism.

Calling & Settlement

URs generated from calls or training are aggregated on-chain via the Receipt Bridge (RB) and settled within the SRP. Settlement is based on the attribution relationships in the Royalty Graph:

- ***Content→RAG:***
  Revenue from retrieved fragments belongs to the original author.
- ***Data→Model:***
  Revenue from data training belongs to the data holder.
- ***Model→Model:***
  Revenue from derivative models is distributed to the parent model author.

Revenue Return & Redistribution

Execution domain fees and protocol taxes are transparently deducted; the remaining revenue enters the shared pool according to rules; part of the revenue, quality, and compliance signals are written back to the DIP, driving the next round of distribution and governance.

Through this mechanism, content assets can not only participate in value circulation as training data or calling resources but also obtain continuous revenue through subscription models and stream payments, thus forming a closed-loop **"rights confirmation-authorization-calling-metering-settlement-governance"** gain cycle within the system.

4.3 PoA Consensus Mechanism

CodexField's PoA (Proof of Access) consensus mechanism is a cryptoeconomic framework built around content calling rights, behavior verifiability, and asset profit-sharing paths. Its core design goal is to achieve the rights mapping of **"content as asset,"** i.e., mapping the access behavior of content resources on the platform into quantifiable, traceable, and distributable on-chain revenue rights.

PoA is not a **"consensus layer security protocol"** in the traditional sense, but rather an economic consensus protocol based on user fund staking, anchored by content asset calls as weights, and distributing profits through on-chain smart contracts and behavior proofs. Its essence is a programmable profit-sharing mechanism used to build access credit and revenue models for Web3 content assets.

4.3.1 Core Mechanism Design

User Staking Behavior & Access Credential Binding

The PoA mechanism allows users or institutions to stake funds into specific content asset pools, typically injected in the form of mainstream assets like $CODEX, USDT, or BNB. After staking is completed, the system issues an access credential (Access Bond) bound to the user's DID within the smart contract, identifying their holding share and available bandwidth in that asset pool. The Access Bond is used in combination with the capability token (CapToken), becoming the basis for permissions and quotas during calls.

Calling & Metering Closed Loop

Every call to content or models generates a unique Usage Receipt (UR), recording fields such as called asset (AssetId/fragment), resource consumption (expressed in unified metering unit MU), execution domain, latency, caller identity, and authorization policy digest. The UR is both the sole factual credential for revenue distribution and the input for security and compliance audits.

Weight Factor Calculation & Dynamic Profit Sharing

In each revenue settlement cycle, PoA maps all valid URs and staking data into profit-sharing weights. The weight is determined by three types of factors:

$$W = stake\_share \times usage\_score \times quality\_factor$$

stake_share: The user's staking share in the corresponding asset pool.

usage_score: The normalized usage amount (MU) of valid URs within the cycle, combined with signals like call count, success rate, retention, and repurchase.

quality_factor: The quality indicator calculated by the Discovery and Indexing Layer (DIP), containing content/model ratings, compliance, dispute records, and historical stability.

This mechanism ensures that revenue is tied not only to the staking scale but also to real usage intensity and quality performance, thereby avoiding **"empty staking"** or **"brushing"** behaviors and ensuring profit sharing aligns with value contribution.

4.3.2 Revenue Pool Mapping & Profit Sharing

The core of the PoA consensus mechanism lies in directly linking real call behavior with on-chain revenue distribution. For this purpose, CodexField introduces an on-chain revenue pool mechanism at the settlement layer, using URs as the only trusted input.

Revenue Pooling

All commercial revenue from real call paths, including AI model inference call fees, training/fine-tuning fees, content subscription revenue, B2B commercial licensing fees, etc., automatically enters the main revenue pool after the call occurs. Each UR records the call's resource consumption (MU), execution domain **(zk/TEE/Committee/ML-Executor)**, authorization policy, and call context, ensuring every income has a clear, verifiable source.

Revenue Splitting

During the settlement process, the Settlement & Revenue Processor (SRP) first deducts the corresponding execution cost **(exec_cost)** based on the MU in the UR and the execution domain cost model. The remaining distributable income **(distributable)** is then attributed through the Royalty Graph:

Content → RAG
Revenue from retrieved fragments belongs to the original author.

Data → Model
Training revenue from datasets is distributed to data contributors.

Model → Model
Revenue from derivative or adapted models is distributed to the parent model author.

Executors
Execution domain nodes receive computation compensation.

Protocol Fee
The protocol charges a certain percentage as system maintenance fees.

After completing the above attribution, the system allocates a portion from distributable **(determined by governance parameter r_poa)** to form the PoA Tranche, specifically used to reward stakers and callers participating in PoA. The profit-sharing weight is calculated according to the unified formula:

$$W = stake\_share \times usage\_score \times quality\_factor$$

This ensures the safety of staked funds while dynamically binding long-term收益 (returns) to real usage intensity and quality performance. All profit-sharing events are written on-chain in the form of UR Root, Royalty Graph weights, and distribution results. External participants can replay the entire distribution process through ZK proofs or sampling review, thereby ensuring transparency, fairness, and reliability of the revenue.

4.3.3 Security & Verifiability

The PoA mechanism cares not only about the fairness of revenue distribution but also emphasizes the security and verifiability of the call behavior itself. CodexField builds a complete set of evidence and risk control systems at the execution and discovery layers, ensuring all profit sharing is based on real calls, while possessing anti-malicious and audit-friendly characteristics.

Verifiable Call Records

Every call generates a UR, which is written to the caller-side receipt aggregator, then compressed in the form of a Merkle tree, and the batch root is submitted on-chain via the Receipt Bridge (RB). This reduces on-chain storage costs while ensuring behavioral data can be replayed and reviewed. When claiming rewards or initiating disputes, users can submit their Access Bond and the corresponding Merkle path for automatic contract verification, independently validating the rationality of their收益 (earnings) without relying on a centralized authority system.

Zero-Knowledge Proofs & Privacy Protection

CodexField natively reserves the integration capability for zk-SNARK/zk-STARK modules. Callers can generate zero-knowledge proofs for their behavior to prove the authenticity and compliance of the call without exposing their identity or specific call details. This mechanism ensures that the platform can still achieve transparent revenue settlement in sensitive scenarios such as finance, healthcare, and enterprise knowledge bases, while also taking into account privacy.

Anti-Cheating & Risk Control System

To prevent malicious brushing and manipulation behaviors, the system has a built-in anti-cheating engine at the VCM and DIP layers:

Traffic Pattern Detection

Identifies patterns like contract self-calling, circular references, and abnormal high-frequency calls in real-time, and down-weights or filters out abnormal URs.

Account Clustering Analysis

Discovers batch fake calls by combining DID relationships and transaction patterns.

AI Risk Control Engine

Uses models to build multi-dimensional feature vectors for call behavior, cross-validating the authenticity and value contribution of calls.

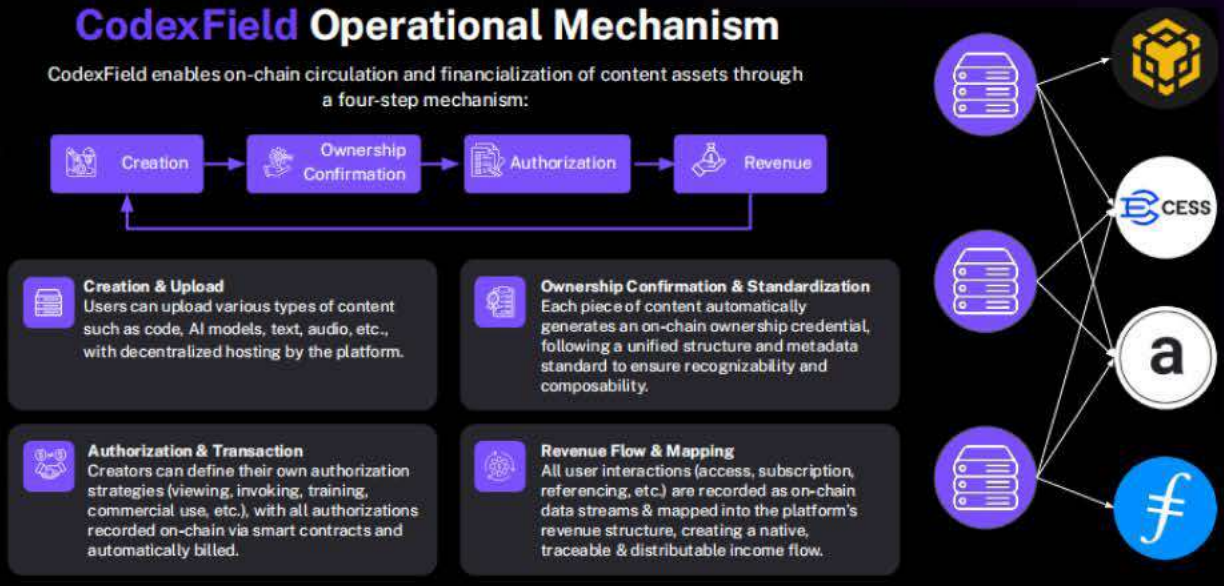
Governance & Transparency

All filtered or disputed call behaviors are logged into the Discovery and Indexing Layer (DIP), serving as negative signals for future CEI (Execution Index) and PoA weight factors. The governance mechanism can dynamically adjust profit-sharing parameters based on this evidence, ensuring the entire system has long-term resistance to abnormal patterns.

Through the above mechanisms, CodexField's PoA consensus not only guarantees the fairness of revenue distribution but also provides a safe, reliable, and privacy-friendly operating environment for all participants, making PoA a trustworthy engine for content asset financialization.

4.4 CodexField Operating Mechanism

CodexField proposes a content asset financialization mechanism based on on-chain native contract logic and multi-modal content structure parsing capabilities. Through the four-stage path of "Creation — Rights Confirmation — Authorization — Revenue," it achieves verifiable rights confirmation, combinatorial authorization, and on-chain revenue mapping for content assets, standardizing the circulation path and protocolizing revenue settlement for on-chain content assets at the underlying layer.



At the system entry point, CodexField receives multi-modal raw content uploaded by users, including code scripts, AI model structures, natural language documents, audio-visual materials, etc. It completes content encapsulation through standardized interfaces and pushes it into distributed storage channels.

Content upload triggers the decentralized storage routing module; the system dynamically selects the optimal storage path based on content volume, estimated access frequency, and call density, prioritizing distribution to decentralized storage networks with on-chain data retrieval capabilities such as Filecoin, Arweave, CESS, and BNB Greenfield, achieving multi-copy heterogeneous distributed hosting to ensure long-term content availability and anti-censorship capability.

In the rights confirmation stage, CodexField uses a structured rights confirmation engine to automatically parse all content assets, generating on-chain rights confirmation credentials (NFTs or other atomic unit asset identifiers) bound to the content. During the process, the platform converts unstructured content into composable, callable structured asset units based on set metadata standards and structural criteria; and forms atomic identity attribution identifiers through content hashing, signature paths, and author address mapping, ensuring the immutability and uniqueness of the content. All rights confirmation information is uniformly written on-chain as Merkle root nodes, possessing complete auditability.

The authorization stage is executed by the contract layer, abstracting access policies through a license policy DSL, including various call types such as read, subscribe, training call, commercial embedding, etc. Authorization actions are automatically compiled into smart contract transaction requests, triggering CodexField's native license contract for authorization verification and on-chain registration, and automatically billing and behavior accounting for the caller. The authorization chain and access records together form a traceable data call graph, providing a computable basis for subsequent revenue mapping and profit-sharing distribution.

On the revenue path, CodexField has built a revenue mapping engine that supports atomic-level mapping and multi-dimensional data reflection. All user call behaviors (access, subscription, training, citation, etc.) are gradually mapped onto the revenue mapping graph within the behavior chain, generating a dynamic revenue scoring model based on dimensions such as access time, frequency, path complexity, and content influence. Combined with the behavior verification and risk filtering models in the PoA consensus mechanism, suspicious access behaviors are filtered out, ensuring that the sources of revenue distribution have a basis in real calls.

Finally, all revenue is uniformly settled through platform contracts and can be synchronously reflected in external clearing paths, achieving a closed-loop on-chain asset path where content calls yield revenue and behaviors lead to realization. This mechanism ensures that content assets on CodexField have a full lifecycle path of "verifiable rights confirmation, composable authorization, traceable calls, and clearable revenue, " laying the structural foundation for content financialization.

4.5 Revenue System

The revenue system is built with "real UR usage as the anchor," forming a transparent, recalculable, and governable value distribution mechanism from revenue pooling, attribution splitting to profit sharing distribution. It uses PoA as the weight layer for access and long-term profit sharing, linked with quality and compliance signals.

1 Revenue Pooling & Distributable Base

Commercial revenue from real calls and authorizations **(API inference, training/fine-tuning, subscriptions, B2B authorizations, etc.)** enters the settlement plane according to contract rules. The SRP first deducts the execution **cost exec_cost=cost(μ,PS,exec_domain)** for each settlement batch, then splits the revenue among creators/cited authors/ data contributors/execution domain/protocol tax based on the Royalty Graph, obtaining the distributable income. A specific proportion of this is designated as the PoA Tranche **(governance parameter τ_poa)** used for profit sharing with bandwidth/usage participants according to PoA, while the rest is distributed to rights holders along the attribution path.

2 PoA: Weight Model for Access & Long-Term Profit Sharing

Within an accounting period (e.g., day/week), a participant's profit-sharing weight is determined by three factors:

W=stake_share×usage_score×quality_factor

stake_share: Staking share in the corresponding pool ($CODEX/stablecoin).

usage_score: Normalized usage amount (MU) of valid URs within the period, including signals like success rate, retention/repurchase.

quality_factor: CEI quality factor from the DIP (reliability, compliance, complaint rate/arbitration records).

To suppress brushing and concentration risks, the system applies saturation/upper limit clamping to usage_score, sets a single entity weight cap for stake_share, and enables down-weighting or filtering of abnormal URs.

3 Strategy Pools: Short-term/Long-term Weight Curves

To balance the cold start of new assets and the long-term stability of mature assets, the PoA Tranche supports two types of strategy pools (only as two configurations of the weight curve, the settlement anchor remains the UR):

SpeedUp Pool (Short-term)

Grants a higher initial weight and faster release pace for usage_score within a limited period, suitable for new models/new data or phased campaigns; automatically reverts to standard weight upon expiration.

CodexForge (Long-term)

Binds the weight to CEI stability, encouraging continuously available and low-dispute assets, suitable for production-grade models and long-term supply.

Neither pool issues additional token subsidies; only the weight/release curve of the PoA Tranche is governed parametrically.

4 Distribution Methods & Enterprise Reconciliation

Supports three distribution modes:

Instant Settlement

High-frequency, low-value calls are aggregated and distributed hourly or per N URs.

Stream Payment

Subscription/monthly scenarios use continuous release, can be paused/renewed, and have arrears circuit breakers.

Batch Settlement

Enterprise billing periods (daily/weekly/monthly) are settled in a lump sum after confirming an EIP-712 invoice.

Full UR details and split evidence are archived in verifiable storage **(e.g., Greenfield/Arweave),** only roots and events are kept on-chain; invoices contain UR Root, Pricing Surface hash, and split results, facilitating audits and ERP integration.

05 Ecological Scenarios

Built upon the foundation of CodexField's content asset confirmation, call incentive, and access profit-sharing mechanisms, a comprehensive crypto-economic system centered around "content as asset" is gradually taking shape. This system can not only serve the platform's native code hosting and trading scenarios but also be widely applied to numerous other scenarios based on content creation, call delivery, and revenue distribution.

Benefiting from the underlying PoA consensus structure and the on-chain access credential model, the application forms supported by CodexField extend far beyond the developer ecosystem, further opening up to a broader range of Web3 value creators. In the future, this content financial structure is expected to be widely used in the following representative scenarios:

5.1 Developer and AI Model Ecosystem

In the realm of the developer and AI model ecosystem, CodexField, through standardized rights confirmation and call paths, enables code, models, datasets, and training scripts to become on-chain native assets. Utilizing the PoA profit-sharing mechanism and Royalty Graph, it ensures that every step from model development and training to invocation can be quantified and receive reasonable returns. This allows developers to benefit not only from model releases but also to continuously share value in collaboration, fine-tuning, and derivative usage.

Model as a Service (MaaS)

On CodexField, developers can host model weights and training scripts via the Gitd toolchain and list them on the Marketplace with one click. When enterprises or individual users call APIs for inference or training, the system automatically generates UR (Usage Receipt), revenue enters the SRP (Settlement and Revenue Routing), and is allocated to the profit-sharing pool based on call intensity and quality factors. This way, developers can directly commercialize their models without relying on centralized platforms and obtain sustainable on-chain income.

Open Source Component Commercialization

CodexField provides a new financial path for the traditionally "unpaid contribution" open-source model. Common components like SDKs, algorithm modules, and smart contract skeletons are confirmed as rights via CC (Content Capsule) on the platform. When other developers or enterprises call these components, the system automatically calculates and generates profit shares, ensuring the original author receives income. This mechanism not only protects the rights of open-source contributors but also injects new incentives into the open-source ecosystem, encouraging more high-quality components to enter the on-chain asset market.

Collaborative Fine-Tuning and Profit-Sharing Network

During AI model iteration, data contributors, fine-tuning engineers, and inference service providers often need to collaborate. CodexField's Model Fabric provides native support: datasets are encapsulated as CC-DS, model weights as CC-MW, and fine-tuning results are registered as new Models. Through the Royalty Graph, training and call revenues are automatically split among data providers, fine-tuners, and model authors according to reference relationships. This way, the entire model evolution chain is completely recorded and financialized, with every contribution receiving transparent and sustainable returns.

5.2 Creators and the Content Economy

In the field of creators and the digital content economy, CodexField supports creators in confirming rights to their works on the platform, defining licensing rules, and automatically earning revenue upon each call, download, or reference, enabling creations to become digital assets with long-term financial value.

AI-Generated Art and Digital Works

Creators can use AI tools to generate illustrations, music, short videos, or novel excerpts, and list them on CodexField's Marketplace. All works are rights-confirmed as CCs; calls or downloads instantly generate URs, and revenue goes directly into the profit-sharing pool. Unlike traditional NFT platforms, CodexField emphasizes not just one-time sales but supports continuous call and reuse revenue paths, giving works long-term monetization potential.

Prompt and Creative Template Marketplace

In the AIGC ecosystem, Prompts have become high-value creative units. CodexField provides a mechanism for rights confirmation and listing of Prompt templates. Users can confirm rights to efficient instruction sets as CC-PMs, which enter the revenue pool through calls or reuse. When other developers or creators use these Prompts, the system automatically generates profit shares, ensuring the original author receives due compensation. This mechanism promotes the formation of a "Prompt engineering economy."

Derivative Works and Remixes

CodexField's Royalty Graph provides a transparent profit-sharing model for derivative works and remixes. For example, if a musician uploads an original track and another creator adapts it into an electronic remix, a reference edge is established between them in the Royalty Graph, and revenue is automatically split proportionally between the original author and the derivative creator. This mechanism not only protects original rights but also encourages more derivative creations, promoting the continuous evolution of content.

Web3-Native Social Content

In CodexField, UGC/PGC content such as short videos, podcasts, and articles can be encapsulated as CCs and enter the Marketplace for distribution. Actions like calls, shares, comments, and likes are recorded as URs or CRs (Contribution Receipts), forming user reputation assets and triggering profit-sharing and incentives. Thus, both creation and consumption behaviors can receive on-chain rewards, fostering a more transparent and incentivized Web3-native creator community.

5.3 Education and Knowledge Services

In the field of education and knowledge services, CodexField provides educational institutions, corporate training departments, research organizations, and independent learners with a standardized mechanism for knowledge asset rights confirmation, invocation, and revenue distribution, achieving full-chain assetization from knowledge production to consumption. This not only gives educational content long-term financial attributes but also establishes a traceable reputation and incentive system for learners and contributors.

Online Education Subscriptions

Teachers or educational institutions can confirm rights to content like lecture notes, video tutorials, and PPT slides as CCs and list them on the CodexField Marketplace. Learners can obtain access through subscriptions or one-time purchases; calling behavior generates URs and contributes to the revenue pool. Courses can be differentially priced by chapter, validity period, or full content. Revenue is transparently distributed to creators or institutions, avoiding the issue of uneven revenue distribution common in traditional educational platforms.

Corporate Training and Knowledge Base Management

Enterprises can host internal documents, training materials, or process specifications in CodexField's private spaces. Access permissions are strictly managed by DIDs and CapTokens, ensuring only authorized employees or partners can access them. The calling process generates URs and, combined with ZK audit proofs, meets corporate compliance requirements. This not only improves the efficiency of cross-departmental knowledge sharing but also creates verifiable records of employee learning behavior for performance assessment and supplier collaboration evaluation.

Academic Research and Data Sharing

Research institutions and university researchers can confirm rights to datasets, experimental scripts, and research results as CCs and share them via the Marketplace. When other researchers use the data or scripts for experiments, the calling behavior generates URs and triggers profit-sharing, allowing the original contributors to receive compensation. Simultaneously, the Royalty Graph clearly records citation relationships, ensuring transparent attribution of results. This mechanism helps break the incentive dilemma in academic data sharing and promotes research collaboration and knowledge circulation.

AI-Assisted Learning and Expert Consulting

Leveraging Agent Fabric, learners or researchers can call models for interactive learning, such as generating exercises, grading assignments, or conducting Q&A tests. The calling behavior also generates URs, creating objective on-chain records of learning activities. Furthermore, experts or consultants can also confirm rights to knowledge Q&A, industry reports, or personalized consultations as CCs; user calls trigger profit-sharing, forming new business models for knowledge services.

5.4 Entertainment and Cultural Creative Industries

In the entertainment and cultural creative industries, CodexField provides game developers, film and video creators, musicians, and community users with a transparent, profit-sharable, and traceable content asset infrastructure. This allows cultural and entertainment works to not only receive rights protection but also continuously accumulate value through cross-platform calls and secondary creation, forming a brand new **"on-chain cultural creative economy."**

Game Mod and Script Distribution

Gaming communities have long relied on centralized forums and non-standardized distribution mechanisms, making it difficult for creators to obtain long-term income. CodexField supports listing game Mods, maps, scripts, plugins, etc., as CCs on the Marketplace; downloads and calls automatically generate URs and contribute to the revenue p/iyaaool. Original authors can benefit not only from player usage but also receive profit shares from derivative teams through the Royalty Graph, forming a complete on-chain gaming ecosystem.

Rights Confirmation for Film/TV Scripts and Creative Materials

Film and video creators can host scripts, storyboards, dialogue scripts, sound effects, and voice-over materials on CodexField, authorizing production companies, actors, or derivative teams to call them. Calling behavior generates URs and enters settlement, with revenue transparently distributed to the original author, team, or collaborators. This mechanism not only protects creative成果 but also ensures every contributing segment of the film and TV industry chain receives reasonable returns, enhancing the enthusiasm for creation and collaboration.

Music and Derivative Economy

Musical works (songs, accompaniments, sound effects) can have their rights confirmed via CC. After being uploaded to the Marketplace, playback, download, or sampling will trigger revenue mapping. If other creators adapt or remix the music, the Royalty Graph automatically splits the revenue, ensuring both the original author and the adapter profit. This mechanism incentivizes musical re-creation and promotes a more active derivative culture within the community.

Fan Economy and Community Creation

On CodexField, fan-created content like short videos, fanworks, and peripheral designs can also obtain rights confirmation and revenue channels. Calls, reposts, and secondary distribution generate URs and are accounted for in the revenue system, simultaneously forming the user's CR (Contribution Receipt), which accumulates as on-chain reputation assets. Combined with the Marketplace's points and ranking mechanisms, fan creation is no longer just a hobby output but can enter a sustainable incentive cycle.

5.5 Public Services and Social Value

In the field of public services and social value, CodexField can provide government agencies, research organizations, public welfare groups, and social collaboration networks with trustworthy digital infrastructure. Through mechanisms for content rights confirmation, verifiable calls, and transparent profit-sharing, the platform can help public data and social resources achieve transparent sharing, ensure compliant use, and simultaneously incentivize more individuals and institutions to participate in the creation and distribution of public value.

Government Data and Open Resources

Various types of public data (e.g., environmental monitoring, traffic data, statistical materials) can be rights-confirmed as CCs and hosted in decentralized storage. When users or enterprises call the data, URs are generated, and ZK assertions prove the authenticity and compliance of the call, avoiding sensitive data leakage issues. This model not only ensures transparent data usage but also encourages data providers to continuously update and maintain public resources through the profit-sharing mechanism.

Medical and Scientific Research Data Sharing

Medical institutions and research units can confirm rights to clinical trial data, medical imaging data, or experimental datasets on-chain. Calling behavior enters the settlement path, ensuring data providers, research institutions, and application parties can share the benefits. Through zero-knowledge proofs and privacy protection mechanisms, the use of sensitive data can be verified without exposing specific details, meeting compliance requirements while promoting scientific research collaboration and medical innovation.

Education and Public Welfare Content Dissemination

Public welfare organizations, educational platforms, or individuals can confirm rights to open educational resources, training materials, and public welfare courses as CCs and list them on the Marketplace. Learners' calling and usage behaviors generate URs, forming transparent learning and revenue records. Creators and institutions can thus receive certain incentive returns, forming a virtuous cycle of "content for public welfare, sustainable revenue."

Social Collaboration and Reputation System

CodexField's CR (Contribution Receipt) mechanism supports recording social collaboration behaviors, such as volunteer translation work, data annotation, knowledge contributions, etc. These contributions accumulate as on-chain reputation assets and are linked to incentives. They can be used for incentive mechanism distribution and also serve as part of future social identity and participation proof.

5.6 Finance and Data Markets

In the field of finance and data markets, CodexField supports financial data, research reports, algorithmic strategies, and analytical tools being traded and shared as on-chain native assets. Combined with the PoA profit-sharing mechanism and Royalty Graph, the platform achieves a complete financialization path from data generation, authorized calls, to revenue splitting, turning knowledge and data resources into circulatable capital units.

Research Reports and Market Analysis

Investment research institutions or independent analysts can confirm rights to market research reports, macroeconomic analysis, or industry insights as CCs and list them on CodexField. When enterprises or individuals call these reports, URs are generated and enter the SRP, with revenue transparently distributed to the author or institution. This model allows research results to no longer rely on subscriptions or one-time sales but to accumulate revenue over time through citations and calls.

Quantitative Trading Strategies and Data APIs

Quantitative teams or developers can encapsulate trading scripts, signal factors, and data APIs as CC-DS or CC-Code and offer subscriptions and calls through the Marketplace. Call records enter the Royalty Graph, and revenue is distributed to strategy authors, data providers, and execution nodes. This way, quantitative strategies that originally circulated only within closed circles can be financialized in a more open manner while maintaining traceable revenue.

RWA and Knowledge Asset Integration

CodexField also supports confirming rights to financial compliance documents, evaluation reports, audit materials, etc., as CCs, integrating them with RWA (Real World Asset) assets. For example, real estate appraisal reports or artwork authentication documents can be put on-chain as CCs, providing trusted credentials for RWA tokenization in cross-chain scenarios, and linking with the revenue mapping system to form a dual value path of "on-chain assets + off-chain knowledge."

Data as a Service and Cross-Industry Sharing

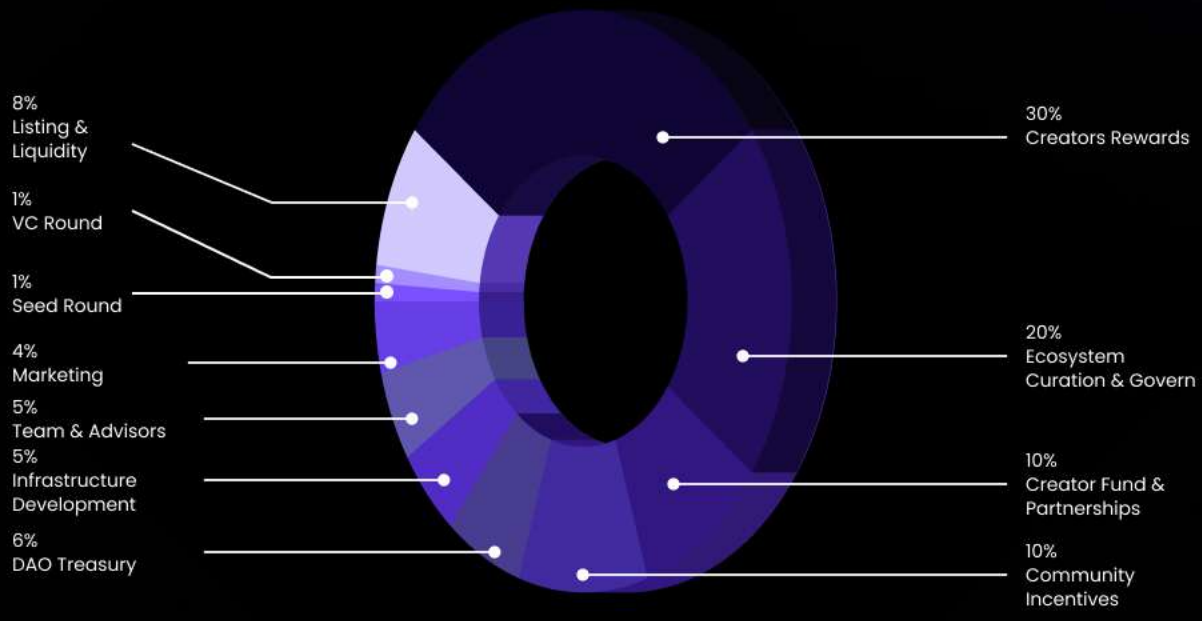
Data service providers can encapsulate data like climate data, consumer data, and corporate operational data as CCs and provide call services. Users generate URs upon calling, and revenue is settled automatically. Through zero-knowledge proofs and compliance labels, data can be reused while maintaining privacy and security, achieving efficient cross-industry circulation. This model transforms data from a "sunk cost" into a "sustainable revenue" asset.

06 Economic Model

CodexField has built a native token economic system core-driven by content rights confirmation and real call behavior. It revolves around $CODEX to realize incentive release, value settlement, and governance of content assets, supporting diversified Web3 content ecosystem scenarios.

6.1 Token Distribution

TOTAL SUPPLY: 120 Milion $CODEX



Allocation	%
Listing & Liquidity	8%
Vc Round	1%
Seed Round	1%
Marketing	4%
Team & Adivsors	5%
Infrastructure Development	5%

Allocation	%
DAO Treasury	6%
Creators Rewards	30%
Ecosystem Curation & Govern	20%
Creator Fund & Partnerships	10%
Community Incentives	10%

6.2 Token Functions

Value Medium

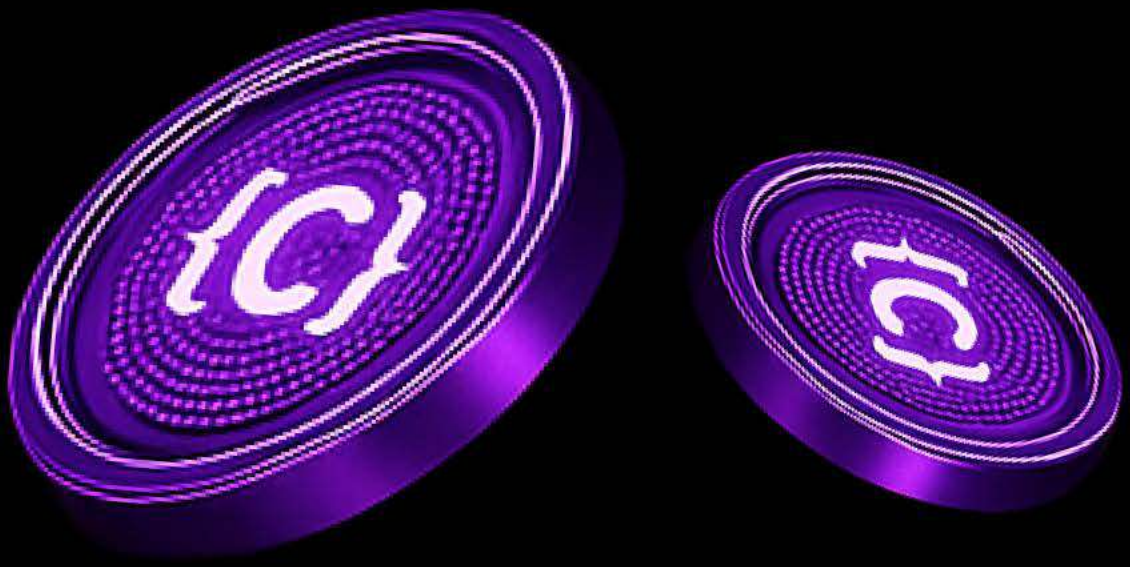
All behaviors such as content access, training, generation, and calls require payment in $CODEX as the basis. The interaction processes of developers, callers, model publishers, and tool integrators are all priced and settled through $CODEX, constituting the basic unit of content circulation.

Behavior Incentive Tool

Through behavior control mechanisms, the platform uses $CODEX for targeted incentives for high-quality content uploaders, genuine callers, developer contributors, and node users participating in collaboration, achieving a positive incentive cycle of content creation-call-collaboration.

Governance Credential

As the core instrument for DAO governance, $CODEX grants holders voting rights on matters such as module launches, revenue parameter setting, and platform cooperation strategies, gradually realizing a community-led protocol governance framework.



07 Roadmap

2025 Q3

- Launch MVP version, enabling code upload, rights confirmation, and storage functions
- Support developers migrating GitHub projects to the Greenfield network for decentralized hosting

2025 Q4

- Enable listing, subscription, calling, and authorization of diverse content assets like code, models, and Prompts
- Launch the "Content Reputation Points + Behavior-Bound Incentives" mechanism to drive developer ecosystem participation
- Deploy three types of structured revenue pools (FastYield / SubFlow / Endurance), introducing PoA staking to inject liquidity support
- Open the first batch of content financial model tests, linking content behavior and contract data for the first time

2026 Q1

- Integrate an AI model scoring system to achieve quality assessment, value ranking, and automated recommendation of uploaded content
- Integrate multi-chain storage networks like Arweave, Filecoin, and CESS to implement cold and hot tiered storage
- Establish a "permission-driven + behavior registration" content distribution mechanism to promote B2B model integration

2026 Q2

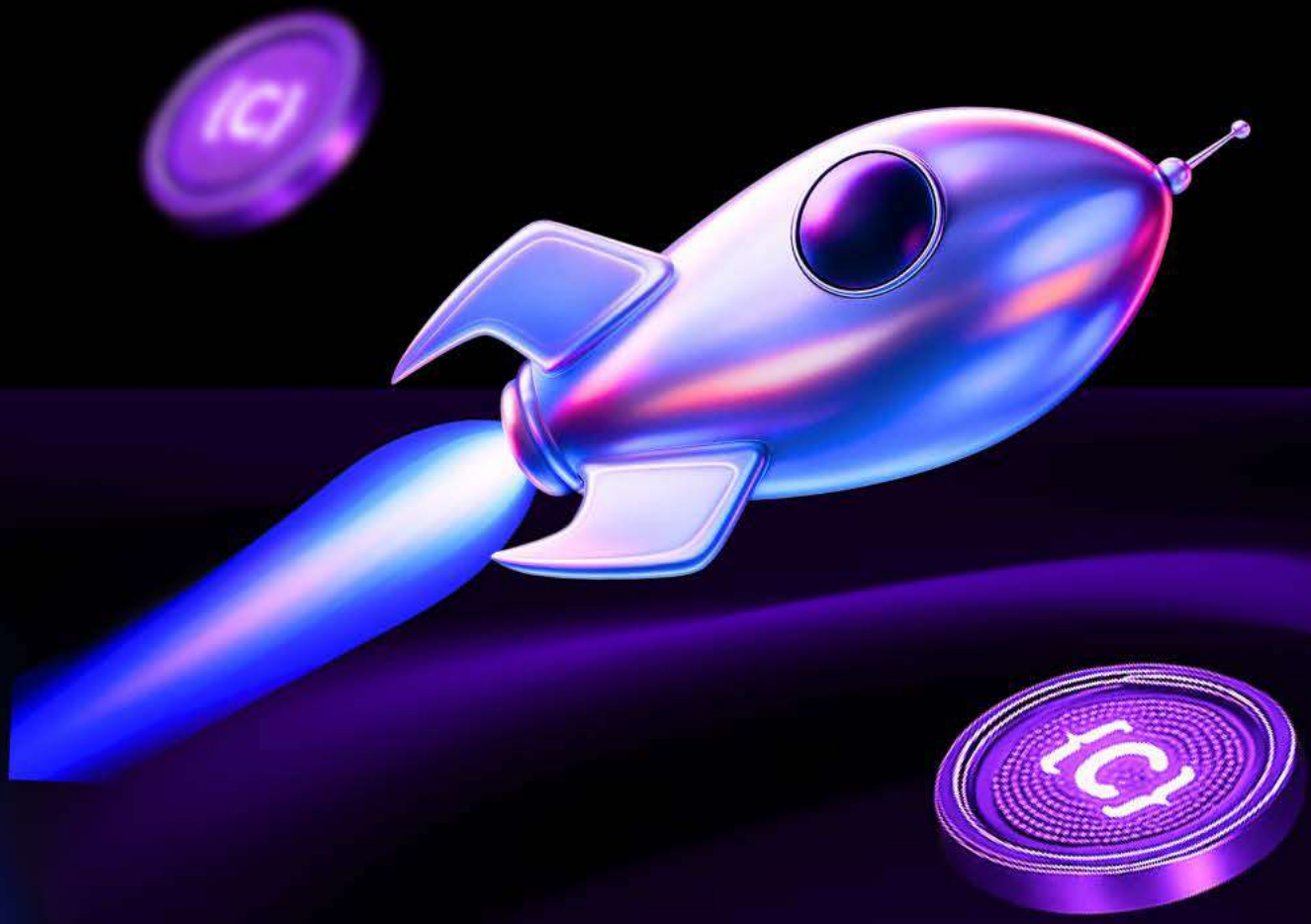
- Open CodexField API interfaces, supporting external platforms and tools calling platform content assets
- Build a standardized on-chain authorization protocol CodexAuth to ensure transparent and traceable content reuse, citation, and payment
- Enable cross-platform collaboration scenario support (e.g., AI SaaS / educational content platforms / DeFi strategy calls)

2026 Q3

- Launch the Codex AI Collaboration Engine
- Deploy content retrieval and inference engines, supporting scenarios like fuzzy content calls, semantic indexing, and automatic composite calls
- Support revenue splitting and liquidity unpacking for model-type assets

2026 Q4 and Beyond

- Launch a cross-chain content mapping module to enable content authorization tracking across multi-chain platforms like ETH / OP / Solana
- Partner with collaborators to promote the "Content Rights Confirmation Alliance," jointly advancing content assetization standards
- Reach API integration cooperation with leading AI platforms or large content platforms to expand platform calling and monetization scenarios



CodexField

Codexfield Website
www.codexfield.com

Codexfield Twitter (X)
@CodexField