NFB

Whitepaper v1.0

NFB ‘Non-Fungible Books’
Revolutionizing the publishing industry
December,2025

www.nfbchain.com
nfbchain@nfbchain.com
Read. Own. Earn.

Executive Summary

NFBChain is a Web3-based publishing platform and marketplace that aims to establish a new publishing standard built on real digital ownership, programmable royalties, and transparent value distribution moving the publishing industry away from closed, centralized, and inefficient structures.

Although the global publishing industry has reached a scale of hundreds of billions of dollars, content creators continue to suffer value loss due to high commission rates, long settlement periods, and dependency on centralized platforms and distribution channels. Readers, on the other hand, do not truly purchase digital content but instead acquire temporary usage licenses, lacking genuine ownership rights in the digital world.

NFBChain transforms books and publications into verifiable, transferable digital assets called **Non-Fungible Books**, which generate continuous revenue from secondary sales. Through smart contracts, royalties are automatically distributed; authors and publishers continue to receive a share from every transaction not only from the initial sale, but from all secondary transactions as well.

By creating a shared value space for authors, publishers, readers, and collectors, the NFBChain ecosystem aims to transform publishing from a consumption-driven model into a digital asset economy focused on ownership and long-term value creation, while also positioning NFBChain as the financial technology company of the publishing industry.

Introduction

Publishing is one of the most deeply rooted industries carrying humanity's cultural heritage. However, while digitalization has made books more accessible, it has also weakened the concept of ownership, rendering content platform-dependent and temporary.Today, digital books have become closed content for readers—non-transferable, non-resellable, non-inheritable, and unable to exist outside the platform. Digitalization has eliminated the secondary markets, collectible value, and cultural continuity that physical books have provided for centuries in the digital realm.On the other hand, Web3 technologies have made possible concepts such as verifiable ownership of digital assets, programmable economic rules, and decentralized value distribution. By applying this technological transformation to the publishing industry, NFBChain aims not merely to transform publishing, but to redefine it entirely.

Market Landscape & Problems

High Commission Rates
In the publishing ecosystem, authors and publishers have become dependent on centralized platforms for digital distribution and on distributors for traditional distribution. These platforms and distribution channels charge commission rates ranging between 50% and 80% in return for hosting, distribution, user access, and end-user sales.

This structure:

- Significantly reduces authors' revenues
- Limits publishers' profit margins
- Makes content production unsustainable in the long term

While centralized platforms and distributors take a disproportionate share of the value chain, authors and publishers the true creators of the content remain under economic pressure.

Long Settlement Periods
In both traditional and digital publishing models, payments are typically made with maturities ranging from 120 to 240 days. Consignment sales structures, deferred payment plans, and manual accounting processes negatively impact the cash flow of content creators.

Although this affects all publishers, it creates serious financial vulnerability especially for:

- Independent authors
- Small and medium-sized publishing houses

Delayed revenue collection directly puts new content production and operational sustainability at risk.

Centralized and Closed Ecosystems
Web2 publishing platforms have full and unilateral control over content management, user data, pricing, and visibility. Algorithms, revenue-sharing models, and access rules are often non-transparent and can be changed unilaterally by the platforms.
In traditional publishing, on the other hand, publishers and authors have no access to end-user data whatsoever; if such data is recorded, it remains solely with distribution and sales channels.

This centralized structure:

- Makes authors and publishers dependent on platforms and distributors
- Eliminates content owners' control over data and pricing
- Prevents fair competition within the ecosystem

As a result, publishing exists not as an open, competitive, and transparent market, but within a closed and monopolized system.

Financial Sustainability Issues
When high commission rates, long settlement periods, high paper and printing costs, and centralized platform/distribution structures come together, a financially unsustainable economic model emerges for the publishing ecosystem.

In this model:

- Authors generate revenue only from the initial sale.
- Digital content has no secondary economic value.
- Readers remain completely outside the economic cycle.
- Digital publications are consumed rapidly, and their economic life cycle ends in a short period of time.

This situation both reduces content quality and limits the long-term growth potential of the sector.

NFB Vision

NFBChain's vision is to move publishing away from a centralized, temporary, and one-directional "consumption" model and transform it into a new "digital asset economy" built on ownership, programmable intellectual property, and sustainable revenue sharing.

NFBChain positions books and publications not as static files that are merely read and consumed, but as living digital assets on the blockchain—verifiable, transferable, and capable of generating economic value over time. This paradigm shift moves the publishing industry from a "product sales" business to a living "intellectual property ecosystem."

True Digital Ownership

NFBChain aims to eliminate the erosion of ownership created by Web2 publishing models.
Through the Non-Fungible Books model, digital books are:

- Owned in a verifiable manner on-chain
- Transferable and tradable in secondary markets
- Recorded with platform-independent ownership

As a result, readers become not merely users who "access" digital content, but true owners in the real sense of the word.

Programmable Royalties and Revenue Sharing

One of the core elements of NFBChain's vision is to make intellectual property revenues programmable and automated.

Through smart contracts:

- Primary sale revenues
- Secondary sale royalties
- Multi-right-holder revenue splits are predefined and distributed without the need for manual intervention.

With this model, authors and publishers continue to earn income from their works not only at the initial sale, but throughout the entire lifecycle of the publication.

Decentralized and Transparent Publishing Infrastructure

NFBChain moves the publishing ecosystem away from the closed structures of centralized platforms and onto a transparent, auditable, and open infrastructure.

Within this vision:

- Sales and royalty flows are traceable on-chain
- Commission rates and rules are predefined
- Content owners retain control over data and pricing

As a result, publishing gains a fair and predictable economic model free from algorithmic uncertainty.

Sustainable Digital Publishing Economy

NFBChain places long-term financial sustainability at the core of publishing.

To achieve this:

- Optimized platform fees replace high commission rates
- Instant or short-settlement payment mechanisms are enabled
- Continuous value creation is generated through secondary markets

While readers become part of the economic cycle of the content they purchase, authors and publishers gain recurring and predictable revenue models.

Metadata Standards, ONIX Compatibility & Programmable Rights

The publishing industry has relied for decades on ONIX (Online Information Exchange) standards to exchange bibliographic, commercial, and licensing metadata across the book supply chain. Developed by EDItEUR, the ONIX family—ONIX for Books, ONIX for Serials, and ONIX for Publications Licenses (ONIX-PL)—provides XML-based international standards that enable machine-to-machine communication between publishers, distributors, retailers, and libraries.

In the Web2 publishing stack, ONIX plays a critical role in standardizing metadata such as ISBN, title, author, publisher, pricing, distribution status, and licensing terms. However, while ONIX efficiently communicates metadata, it does not enforce or automate rights. Royalties, licensing conditions, and secondary value flows remain off-chain, dependent on manual contracts, delayed accounting, and opaque platform logic.

As digital publishing expanded, the industry attempted to address ownership and rights issues through DRM systems, digital banderol solutions, and centralized license managers. These Web2 approaches, however, failed to give digital content native uniqueness. Digital books remained technically copyable files, while ownership and resale were restricted by platform policies rather than enforceable economic rules.

NFBChain Approach: Extending ONIX with Web3 Programmability

NFBChain does not replace ONIX—it extends it into Web3.

In the NFBChain architecture:

- Bibliographic and commercial metadata defined by ONIX is stored off-chain using decentralized storage
- A cryptographic commitment of this metadata is anchored on-chain via smart contracts on Ethereum.

This design preserves compatibility with existing publishing standards while adding verifiable ownership, traceability, and programmability at the protocol level.

Programmable Royalties & Native Uniqueness

The core innovation of NFBChain is the introduction of native digital uniqueness through smart contracts.

Every published asset on NFBChain whether:

- Digital NFB ,
- Rare NFB ,
- Physical books linked to on-chain ownership

receives a unique, verifiable on-chain identity.

This identity does not only represent ownership. It also encodes:

- Primary and secondary sale rules
- Automated royalty distribution
- Derivative and ancestry relationships
- Licensing permissions and economic constraints

directly at the smart-contract layer.

Where ONIX-PL defines licensing terms as structured messages, NFBChain converts licensing into executable logic. Rights are not only declared , they are enforced by code.

A Web3 Publishing Standard

Through this approach, NFBChain elevates the role ONIX played in Web2 publishing from a metadata communication standard to a programmable publishing standard for Web3.

- Every asset is fully traceable across its lifecycle,
- Every sale is settled transparently on-chain,
- Every royalty is distributed automatically and instantly,
- Every derivative use follows pre-defined, enforceable rules.

NFBChain transforms publishing from a file-and-license-based model into a native digital asset economy, where ownership, rights, and value flow transparently and continuously on the blockchain.

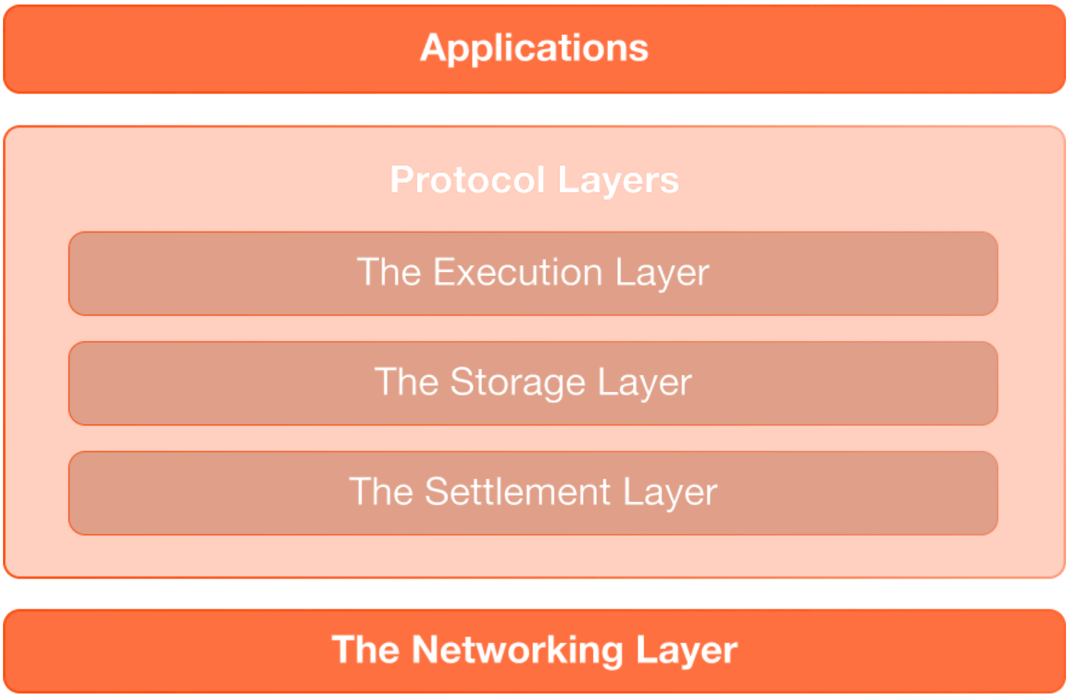
NFB Ecosystem Overview

The NFBChain ecosystem is designed as a modular and scalable Web3 publishing architecture that unifies all stakeholders in the publishing industry under a single decentralized infrastructure. The ecosystem provides an end-to-end structure covering the entire process from content creation and distribution to ownership and secondary market transactions.

NFB is built around the Marketplace, Author / Publisher / Reader roles, the Non-Fungible Books model, and on-chain revenue-sharing mechanisms.

$$P_{total} = P_{publisher} + P_{reader} + P_{author}$$

NFB Architecture Design



Applications

Protocol Layers

The Execution Layer

The Storage Layer

The Settlement Layer

The Networking Layer

Ecosystem Components

In V1, the NFBChain ecosystem consists of four main system components:

Marketplace App (iOS / Android)

The Marketplace App is the access and transaction hub for end users. The application is designed with two core modules:

- **Digital NFB**
- **Rare NFB**

Application functions:

- SSO-based registration + 2FA
- Non-custodial wallet generation
- USDT / USDC / NFB token deposit & withdrawal
- NFB buying, selling, and transferring
- Secondary market listing + price setting

- Reading with DRM-based verification
- Read-to-Earn and loyalty mechanisms (V2)

The Marketplace App is the primary end-user interface of the ecosystem and transforms the trio of “reading + ownership + trading” into a single, unified product experience. From the user’s perspective, the app appears as two main modules: Digital NFB and Rare NFB. However, behind the scenes, these modules connect to the same protocol layers with different access rules.

User Onboarding & Wallet Infrastructure

The registration process begins with SSO + 2FA and progresses without exposing blockchain complexity to the user. Once registration is completed, the system generates a non-custodial wallet in the background and grants the user full control over this wallet.

At this stage, the application connects to the blockchain via the Networking Layer using RPC connections and indexing services, while secure key storage and signing processes are handled locally on the user’s device.

Transactions & Marketplace Execution

Although token deposit and withdrawal functions appear at the application layer with the simplicity of a “banking app,” they actually reflect transfers executed on the Settlement Layer.

Flows such as NFB purchasing, selling, transferring, and secondary market listing trigger marketplace smart contracts running on the Execution Layer. As a result, when a user purchases a work, they do not merely gain access to a file—they acquire a verifiable ownership record on-chain.

Reading, DRM & Ownership Verification

On the reading side, the Marketplace App links on-chain ownership verification with the DRM/entitlement flow. This connection represents one of the system’s most critical security boundaries:

- The content file remains encrypted in the Storage Layer
- The application retrieves the decryption key only if on-chain ownership is successfully verified

This ensures that even if the file is leaked, it cannot be read by anyone who does not own it. When Read-to-Earn and loyalty mechanisms are activated in V2, the application layer will again connect reading actions to a verifiable economic flow through the same Execution + Settlement Layers.

Publisher Admin (Web)

The Publisher Admin is the web-based panel used by publishing houses for content and sales management. It serves as the entry point of the content pipeline.

Publisher panel functions:

- Account activation after application approval
- Content upload + metadata entry
- ISBN verification / edition management
- Digital NFB mint request / Rare NFB mint request
- Primary sale initiation
- Sales reports, revenue tracking, royalty breakdown
- Wallet address definition

In V1, the Publisher role is primarily designed for publishing houses. Individual authors are listed under publishers in V1; self-publishing will be enabled in the V2+ phase. The Publisher Admin is a web panel built specifically for publishers and acts as the gateway to the content pipeline within the ecosystem. It is not merely a "CMS for file uploads," but an operator interface that transfers the metadata, verification, and sales configurations required for the correct on-chain creation of an NFB asset into the protocol.

Once publisher onboarding is completed, the publisher panel becomes active and the publishing house defines its own wallet address. The critical distinction here is that wallet ownership and key management are entirely the publisher's responsibility; the system only enforces protocol rules to automatically distribute revenues at the moment of sale.

During the content upload and metadata entry phase, the Publisher Admin generates the identity layer of the book that will be linked on-chain. ISBN verification, edition management, and contributor fields are consolidated at this layer. This data does not necessarily need to be written directly on-chain; however, the hash of the metadata package stored in the Storage Layer is linked to the NFB asset via the Execution Layer. As a result, while content management appears like a typical "Web2 panel" from the publisher's perspective, the underlying process technically creates a programmable IP record referenced on-chain.

On the minting side, the Publisher Admin prepares mint requests for Digital NFB or Rare NFB. These requests include economic rules such as supply, sale type, royalty distribution, and if applicable access or subscription parameters. The approval mechanism is connected to the General Admin; once approved, the mint operation is executed by the Execution Layer and finalized on the Settlement Layer.

Publish-to-Earn Model

NFBChain adopts a Publish-to-Earn model to enable sustainable growth of the publishing ecosystem. This model does not only reward content consumption, but also content creation, catalog onboarding, and institutional participation in the royalty infrastructure.

The Publish-to-Earn approach incentivizes publishers, copyright societies and institutional content providers to actively integrate with the NFBChain protocol. As a result, token distribution is tied to measurable and verifiable publishing contributions, not speculative behavior.

Participate-to-Earn Multiplier

The Participate-to-Earn Multiplier is the coefficient that determines the amount of $NFB tokens a publisher or copyright society receives under the Publish-to-Earn model.

This multiplier represents the quality and scope of the contribution made to the ecosystem.

Events That Trigger the Multiplier

The Participate-to-Earn Multiplier is activated in cases such as:

1. A publisher joining NFBChain

- Completing onboarding and verification
- Publishing its first on-chain catalog
- Accepting the programmable royalty model

2. A national or international copyright society joining

- Integrating NFBChain for rights recognition / verification
- Providing multi-country and multi-language coverage
- Supporting on-chain royalty flows

Each new participation increases the earn multiplier of the relevant publisher or society, which in turn boosts the amount of tokens they are eligible to receive.

Token Allocation Logic

$NFB tokens distributed under Publish-to-Earn are **not fixed grants**. The allocation amount is determined programmatically based on:

- Participant type (publisher / copyright society / collective entity)

- Geographic coverage (local, regional, multinational)
- Catalog size and activity level
- Long-term commitment to the protocol

Vesting and Distribution Rules

All $NFB tokens allocated under the Publish-to-Earn program follow this vesting structure:

- **6-month cliff**
- **36-month linear vesting**

General Admin (Web)

The General Admin is the platform's operational control panel. Access and permissions are separated using RBAC .

General Admin functions:

- Publisher onboarding review / approval / rejection
- User management
- Content management
- Commission configuration
- Auditing & reporting
- Support workflows
- System metrics

The General Admin serves as the operational control panel of the platform, with permissions clearly separated through RBAC. Although this panel may appear to function as a form of "centralized management," in line with NFBChain's vision, the role of General Admin is not to arbitrarily control the protocol's economic rules, but to execute the operational security and quality control processes required in V1.

Publisher onboarding review/approval/rejection, content validation, and system policies are handled here. However, critical parameters especially commissions and platform fees are ideally managed within Execution Layer smart contracts in a time-locked and auditable manner. In this context, the General Admin panel acts merely as a parameter input interface; the true enforcement and execution power resides within the protocol itself.

On the auditing and reporting side, the General Admin's most critical dependency is the Networking Layer. Event indexers, on-chain transaction mapping, audit logs, support tickets, and system metrics are all fed through this layer. This enables the admin team to produce reports with on-chain verifiability, such as:

- Which work was minted through which transaction
- How much royalty was distributed from each sale

- Which user executed which transfer

Support processes are also a key component of this architecture. A user complaint may be opened like a typical Web2 support ticket; however, the resolution process is often verified through on-chain event logs. This approach technically resolves one of the most critical issues in publishing: revenue transparency.

Blockchain Layer

The blockchain layer is the layer where immutable ownership records and royalty mechanisms operate.

Core functions maintained on-chain:

- NFB mint / burn
- Ownership
- Transfers and marketplace sales
- Royalty distribution
- Payout records
- On-chain event logs

The Blockchain Layer is not a single layer. In NFBChain's design, the on-chain side is handled across three layers: Execution + Storage + Settlement.

In the Execution Layer, NFB mint/burn, ownership, transfers, marketplace sales, royalty distribution, and payout records are executed via smart contracts. This enables a rule-based publishing economy: royalty structures are fixed at the time of minting, operate automatically on secondary sales, and significantly reduce the need for manual accounting.

In the Storage Layer, content files and book metadata are stored off-chain in a verifiable manner. Since the content hash is anchored on-chain, any modification to the file can be detected. This gives digital works much like physical books a verifiable identity for the first time. This layer is especially critical for Digital NFBs, providing both security and cost optimization.

The Settlement Layer ensures the final correctness and economic finality of all these operations.
Since NFBChain does not operate its own chain in V1, the selected EVM network serves as the provider of finality and security. NFBC and stablecoin transfers are finalized here; marketplace sales are settled here; royalty payout records become immutable here. For this reason, the term "Settlement" more accurately reflects NFBChain's real need than "Consensus." NFBChain's priority is not to design network consensus, but to ensure the finality of the publishing economy and the indisputable recording of ownership.

Non-Fungible Books Model

An NFB is the on-chain digital asset representation of a book or periodical publication. The NFB model is structured across two layers:

- Asset Layer: Ownership, transfers, sales, royalties
  Content Layer: File storage, encryption, access control, reading experience

NFB Asset Classes

A) Digital NFB

- Edition-based supply (e.g. 999 / 9,999 / unlimited)
- Secondary sales enabled with perpetual royalties
- Access limited exclusively to the NFB Marketplace reader

B) Rare NFB

- Collection-focused, single or extremely limited supply
- Signed editions, special prints, unpublished content, manuscripts, etc.
- If physical delivery is involved: off-chain logistics procedures are required
- Secondary sales are supported; royalty rules can be customized based on asset type

Book Metadata

A robust metadata standard is critical to ensure the reliability of NFBs and the correctness of royalty distribution.

On-chain Metadata

Fields written on-chain are intentionally kept to a minimum for immutable verification purposes:

- Book title
- Original title
- Publishing house
- Author
- Publication date
- Mint date
- ISBN
- Print year
- Print / edition information
- Content hash

- Metadata URI

Objective:
To anchor the proof of “this work is this work” on-chain, while keeping detailed and updatable information in an off-chain JSON structure.

Off-chain Metadata

To support reader experience and publishing standards, extended metadata fields are stored in an off-chain JSON structure:

- Editor-in-Chief
- Editor
- Graphic Design
- Translator / translation language
- Category / genre / tags
- Cover images, preview pages
- Content description, chapters
- License type

These fields remain flexible and updatable off-chain while being cryptographically linked to the on-chain record through the metadata hash.

Access & DRM

The goal is to ensure that “the reader who owns the token” and “the reader who can access the file” are the same person. To achieve this, token ownership verification is combined with encrypted content distribution.

Core DRM Flow

- The user opens a book in the application
- The app sends an Access Request to the backend
- The backend verifies token ownership on-chain
- If verification succeeds, the backend generates a device/session-based read session
- An encrypted content segment + a derived key are delivered specifically for that device/session
- The reader decrypts and displays the content only within the application

Content Storage

- Files can be stored on distributed storage layers such as IPFS / Arweave / Filecoin
- Content is stored in encrypted form at the storage layer
- Only the CID / hash is stored on-chain
- The decryption key is never stored in plain text in public storage

This model does not claim to reduce piracy to zero; the objective is to significantly reduce piracy compared to Web2 models and to bind access directly to ownership.

Commission & Royalties

The economic core of NFBChain is the automatic calculation and distribution of shares for both primary and secondary sales.

Primary Sale

When a publisher lists a work for sale, the revenue split is defined as:

- Publisher share
- Author share
- Translator / editor / designer, etc. shares
- Platform fee

Distribution:
At the moment of purchase, automatic multi-address distribution is executed in a single transaction.

Secondary Sale

When a user lists a work and a sale occurs:

- Publisher royalty: **10%**
- Platform fee: **10%**
  Seller net: **80%**

These ratios can be:

- Configured per asset type
- Modular based on campaigns or partner agreements

End-to-End Workflows

User Registration & Wallet Provisioning

- Account creation via SSO
- 2FA activation
- A non-custodial wallet is generated in the background without exposing complexity to the user
- Private keys are stored on the device; export/import options are provided
- Cloud backup is optional

Publisher Onboarding

- Publisher applies via a static website
- Document submission + agreements
- Publisher dashboard is activated after admin approval
- Payout wallet address is collected

Content Upload & Mint

- PDF / EPUB upload
- Metadata entry
- System performs ISBN and duplicate checks
- Admin approval
- Content hash + CID generation
- Mint call is executed; token is minted
- On-chain events are indexed

Primary Sale

- Publisher defines price and supply
- Sale contract / listing is created
- User purchases the asset
- Commission and royalty split is executed
- User gains access

Secondary Sale

- User creates a listing
- Buyer completes the purchase
- System executes the split (publisher royalty + platform fee + seller)
- Ownership transfer + access transfer are completed

Reading / Access

- Reader selects Open
- Ownership is verified
- DRM session is initiated
- Encrypted content is decrypted and displayed within the reader

Revenue Model

Platform Revenue

- Transaction fees
- Premium features
- Enterprise publisher solutions
- API access

Publisher Revenue

- Primary sales
- Perpetual secondary royalties
- Publish-to-Earn incentives

User Revenue

- Secondary sale profits
- Read-to-Earn rewards
- Token utility

Security & Responsibility

NFBChain is built on a non-custodial approach:

- Private keys are stored on the user's device
- The platform does not custody user funds
- Key management responsibility lies with the user
- The platform only automates verification and distribution within marketplace flows

TOKEN

What Is the NFB Token?

NFB is the native utility token of the NFB Marketplace ecosystem, operating on Ethereum and compliant with the ERC-20 standard. It represents the protocol and application layer through which publishing ownership, licensing, royalties, and secondary market revenues can be programmed. NFB is not the gas token of an independent network; instead, it leverages Ethereum's secure finality to standardize publishing-specific ownership and revenue flows through smart contracts.

The core design objective of NFB is to unite two historically disconnected economies within the publishing industry under a single framework:

1. **The content economy**
2. **The user economy**

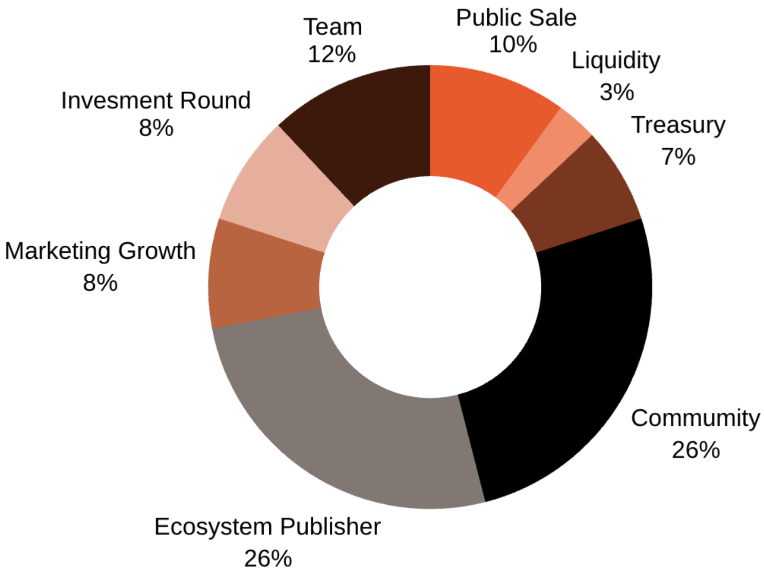
This connection is established through a contract layer that transforms the digital book from a mere "access license" into a transferable, resellable digital asset with automatically flowing royalties. The NFB token functions as the economic instrument that accelerates the growth of this asset economy, coordinates incentives, and enables transparent management of platform revenues.

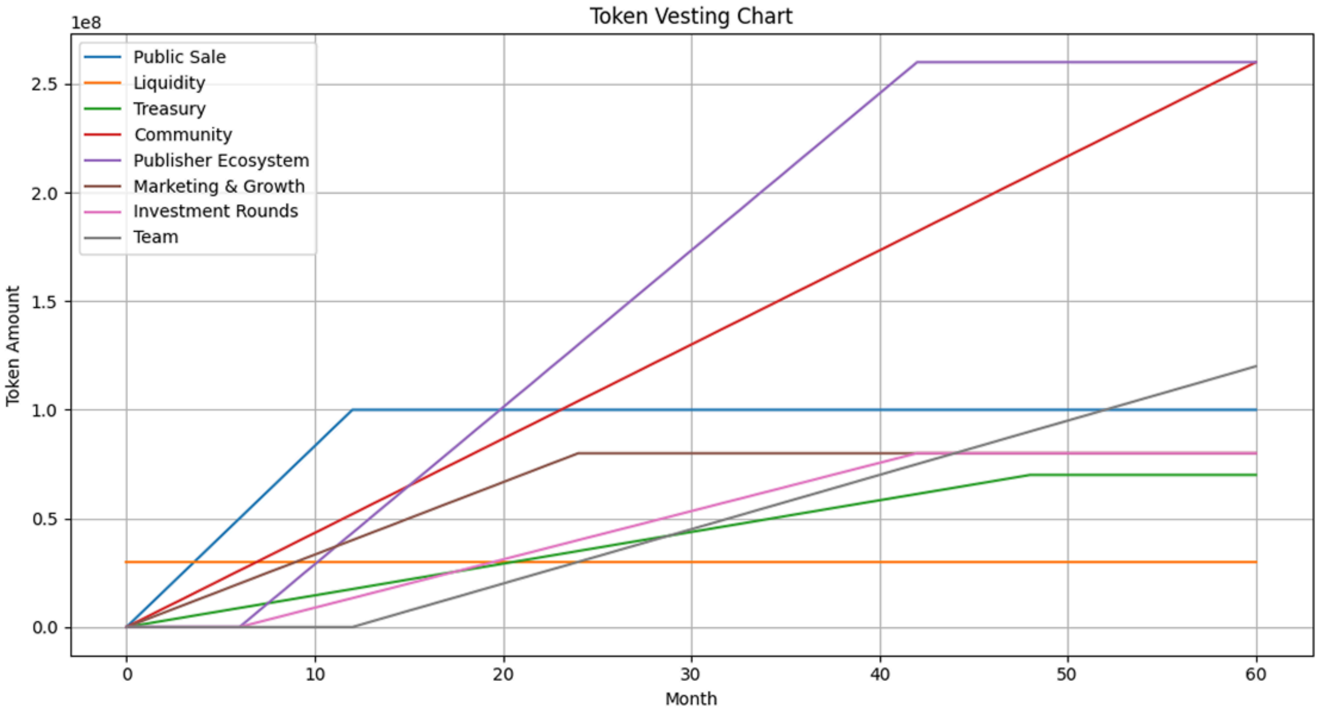
Tokenomics

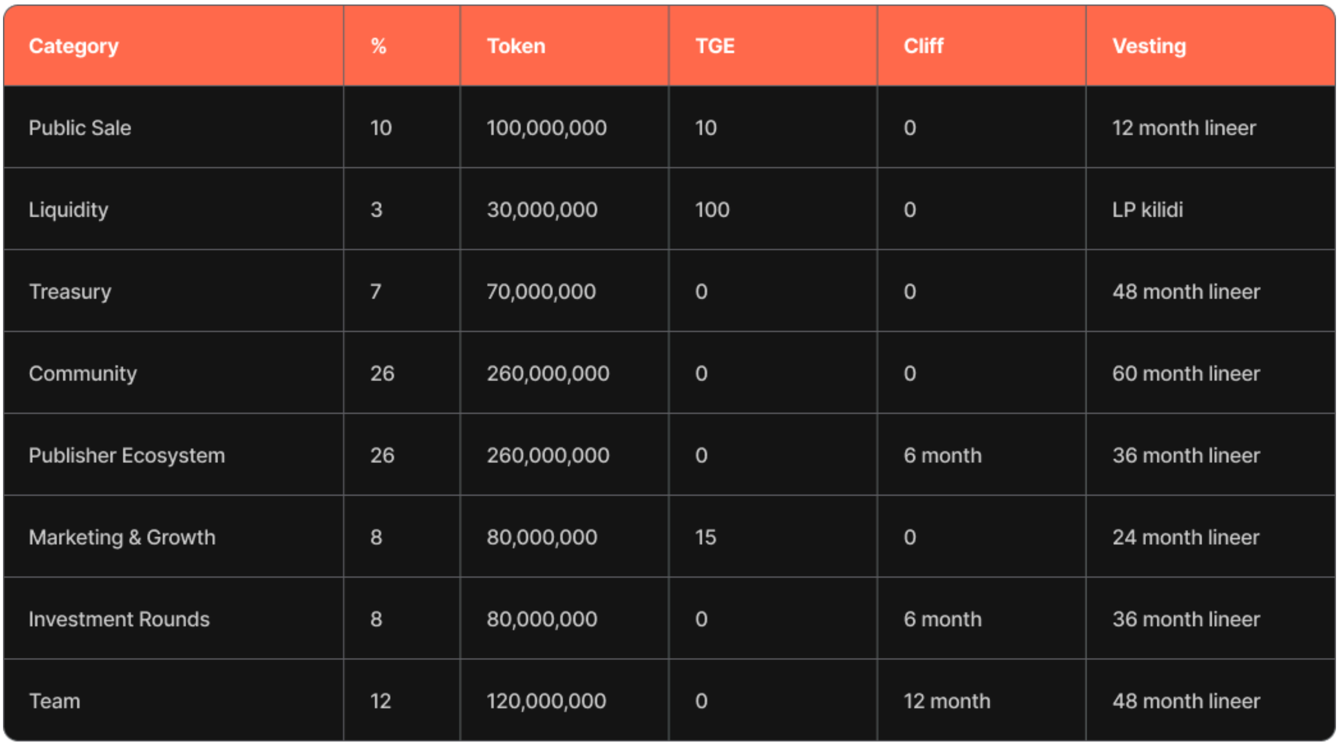
NFB tokenomics are designed to balance two fundamental tensions at the scale of a publishing protocol: on one side, incentives that accelerate catalog and user growth; on the other, a predictable supply and lock-up structure that preserves market confidence. For this reason, NFB's total supply is defined as fixed, and circulation is tightly controlled through clear vesting rules.

- **Total Supply:** 1,000,000,000 NFB
- **Standard:** ERC-20, minted on Ethereum

The distribution model is structured to simultaneously account for liquidity health, ecosystem growth, security budgeting, and long-term team commitment. On the public sale side, the objective is price discovery and broad distribution without releasing large amounts of supply into circulation at once. On the POL side, the goal is not short-term liquidity spectacle, but long-term market stability. The treasury is not treated as a "sellable reserve," but as a controlled resource dedicated to funding operations, audits, integrations, and growth programs.


Team
12%
Public Sale
10%
Liquidity
3%
Treasury
7%
Invesment Round
8%
Marketing Growth
8%
Community
26%
Ecosystem Publisher
26%


Token Vesting Chart
1e8
Public Sale
Liquidity
Treasury
Community
Publisher Ecosystem
Marketing & Growth
Investment Rounds
Team
Token Amount
0.0
0.5
1.0
1.5
2.0
2.5
0
10
20
30
40
50
60
Month



Category	%	Token	TGE	Cliff	Vesting
Public Sale	10	100,000,000	10	0	12 month lineer
Liquidity	3	30,000,000	100	0	LP kilidi
Treasury	7	70,000,000	0	0	48 month lineer
Community	26	260,000,000	0	0	60 month lineer
Publisher Ecosystem	26	260,000,000	0	6 month	36 month lineer
Marketing & Growth	8	80,000,000	15	0	24 month lineer
Investment Rounds	8	80,000,000	0	6 month	36 month lineer
Team	12	120,000,000	0	12 month	48 month lineer

TGE Circulating Supply

- **Public Sale TGE:** 10,000,000
- **Liquidity TGE:** 30,000,000
- **Marketing & Growth TGE:** 12,000,000

Total TGE unlock: 52,000,000 NFB (5.2%)

Smart Contracts & Automation

NFB's on-chain architecture is designed in a modular way to securely, audibly, and deterministically fulfill four core requirements critical to publishing:

1. work ownership,
2. sales flows,
3. royalty routing,
4. payment and audit records.

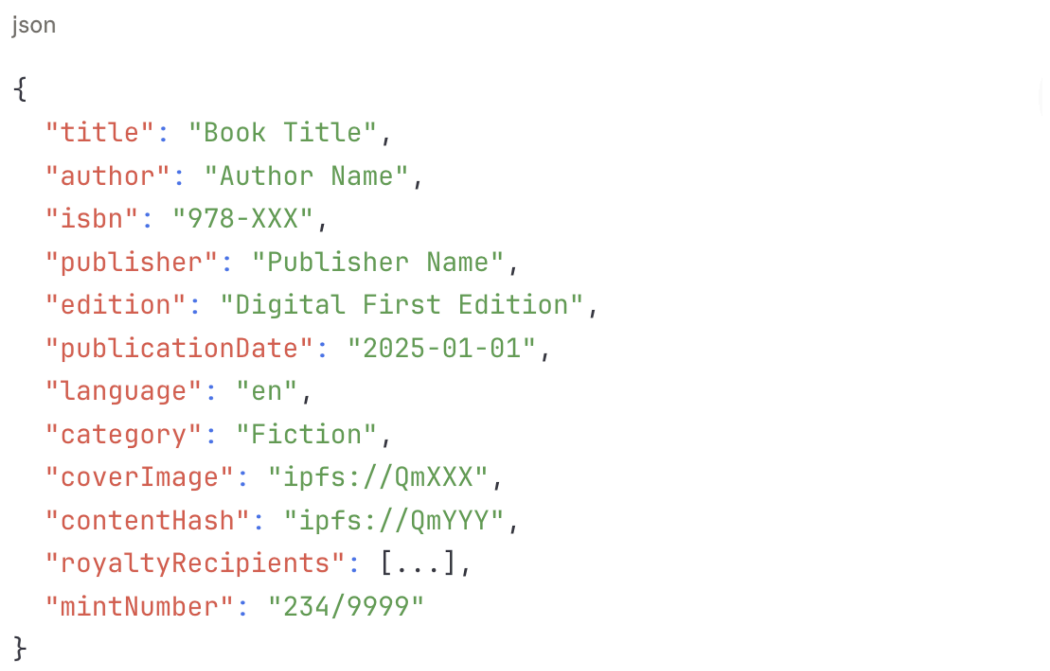
The smart contracts deployed on Ethereum connect the operations requested by the Marketplace application, Publisher Admin, and General Admin into an on-chain–verifiable workflow.

On the asset side, there are two primary NFB asset classes: Digital NFB and Rare NFB. Technically, this distinction determines the standard used based on the production logic of the work.

- Digital NFBs, produced in edition-based quantities (e.g., 999–9,999) under the same ISBN/edition, are more efficiently implemented using the ERC-1155 standard in terms of cost and scalability.
- Rare NFBs, being single or truly unique assets, align naturally with the ERC-721 standard.

In both models, token metadata carries the digital equivalent of the publishing world's bibliographic record, with critical fields recorded either directly on-chain or in a verifiable manner via on-chain hashes.

This bibliographic record is not merely marketing information; it is the identity layer that makes ownership and royalty flows legally and operationally meaningful. In the application and contract design, bibliographic fields are handled through an extended schema, as outlined below:



json

```json
{
  "title": "Book Title",
  "author": "Author Name",
  "isbn": "978-XXX",
  "publisher": "Publisher Name",
  "edition": "Digital First Edition",
  "publicationDate": "2025-01-01",
  "language": "en",
  "category": "Fiction",
  "coverImage": "ipfs://QmXXX",
  "contentHash": "ipfs://QmYYY",
  "royaltyRecipients": [...],
  "mintNumber": "234/9999"
}
```

The content file itself is not written directly to the blockchain. Instead, content is stored in decentralized storage, while the blockchain holds the content hash + content URI + versioning record.
Files uploaded via the Publisher Admin are hashed within the content pipeline, and only the cryptographic fingerprint required for verification is written on-chain. This ensures that the exact content used at the time of minting can always be proven, and that any subsequent

modification to the content can be cryptographically detected while simultaneously preventing blockchain bloat and optimizing cost and performance.

The sales flow is not implemented with the simplistic “transfer NFT – send payment” logic, because publishing sales inherently involve multi-party royalty and commission distribution. For this reason, marketplace transactions are executed atomically on-chain using a Sale Router / Escrow pattern.
When a user purchases a work, the payment is first escrowed in the contract; the correct revenue split is applied based on the sale type, and all transfers are finalized in a single transaction. Atomicity is a critical security feature here: either the payment is correctly distributed and the asset is transferred, or the entire transaction is reverted. This technically prevents trust-breaking scenarios common in publishing, such as “the asset was transferred but royalties were not paid.”

On the royalty side, the goal is not to treat distribution as a one-time transfer list, but to design a revenue vault model where income is accumulated and rights holders can securely claim their shares.
In NFBChain, a Royalty Vault is created for each work/edition, and the stakeholder breakdown defined in the Publisher Admin is fixed on-chain as a royalty policy. Publisher revenue from sales flows into this vault. Distribution from the vault can operate in two ways:

- **Push (automatic payout):** for scenarios with a small number of stakeholders
- **Pull (claim-based):** for multi-stakeholder or high-volume scenarios, optimizing gas costs

The claim model avoids inflating gas usage during peak periods and allows rights holders to withdraw income incrementally and securely. Batch-claim functions can further improve user experience.

For security and operational control, contracts are designed with role-based authorization, emergency pause mechanisms, transparent event logging for critical parameters, and a controlled upgradeability strategy.
Upgradeability does not imply arbitrary change: privileges are bound to a time-locked multisig, and any change to critical parameters—such as commission rates, whitelisted payment tokens, or publisher approval rules—leaves an on-chain trace. This ensures auditable technical behavior in a trust-critical industry like publishing.

Access and reading are inherently hybrid processes:

- The blockchain is the immutable source of truth for “who owns the asset”
- The application layer securely implements “how the content is accessed”

When a user opens a work, the application verifies ownership via the wallet and then securely delivers the content key. DRM validation is not an on-chain check, but an application protocol triggered by on-chain ownership, reinforced by content security and anti-abuse controls.

NFB Royalty Module: Contract Architecture & Flow Design

The primary objective of the NFB royalty system is to automatically route revenues generated during sales (primary/secondary) both according to marketplace economics and to content stakeholders including, where applicable, ancestral rights in derivative chains.
For this reason, the architecture clearly separates the Sale Router from the Royalty Policy. The Sale Router enforces marketplace rules; the Royalty Policy enforces intellectual property rules.

Core Asset Model: Work, Edition & Token Representation

An NFB asset is modeled at two levels:

- **Work:** represents the identity of an intellectual property
- **Edition:** represents a commercial variant of that work

In V1, practical usage is generally edition-centric, since ISBN, print year, and print quantity are directly associated with the edition. As a result, royalties and sales logic typically operate at the Edition level.

Edition metadata is not merely visual or product-page information—it is the foundation of royalty distribution. It is recommended in the whitepaper to split bibliographic data ("Book Title, Original Title, Publisher, Author, Publication Date, Mint Date, ISBN, Print Year, Print Run, Editor-in-Chief, Editor, Graphic Design, Translator, etc.") into two parts:

- an on-chain hash/commitment

- detailed metadata stored off-chain (IPFS/Arweave)

The core principle is to maintain an immutable metadata root on-chain, while heavy files (PDF/EPUB, cover images, additional media) reside off-chain. If the content changes, the hash no longer matches cryptographically proving tampering.

NFBAssetRegistry: Work / Edition Registration & Identity Proof

The NFBAssetRegistry contract records whether a Work or Edition is recognized by the protocol. After a publisher uploads content and admin approval is granted, the registry records:

- `registerWork(...)` → creates a Work identity (publisherId, canonical title hash, creator references)

- `registerEdition(workId, editionMetadataHash, isbnHash, supply, formatType, royaltyPolicyId, vaultAddress)` → creates an Edition

ISBN verification is largely handled off-chain in V1 (admin panel + verification services). On-chain, the ISBN hash is stored instead of the raw ISBN, reducing regulatory and operational risk while preserving provability.

The registry's role is not merely record-keeping—it is the single source of truth referenced by all other contracts. Marketplace, Vault, Policy, and DRM access modules all operate via `editionId → registry lookup`.

NFBNFT: Ownership & Transfer Standard

NFB assets are represented using two token standards:

- **Digital NFBs:** ERC-1155
- **Rare NFBs:** ERC-721

The NFBNFT contract is the sole verifiable source of mint and transfer events. Minting authority does not belong directly to publishers; publishers submit mint requests via the admin panel, and minting occurs after approval aligning with V1 onboarding realities.

MarketplaceSaleRouter: Sales & Settlement Layer

Marketplace economic rules are not embedded in royalty policies; they are enforced by the **MarketplaceSaleRouter**, ensuring independence from derivative graphs.

The router handles two sale types:

Primary Sale

- Listed by publisher
- Platform fee applied
- Remaining revenue routed to the Edition Royalty Vault

Secondary Sale

- User-to-user sale
- Platform fee → treasury
- Publisher royalty → Edition Royalty Vault
- Seller proceeds → seller
- All settled in a single transaction

Only whitelisted tokens (USDT / USDC / NFB) are accepted. The router acts as both escrow and atomic delivery mechanism no payment without transfer, no transfer without payment.

Royalty Vault: Revenue Accumulation & Claims

Each Edition has a dedicated **Royalty Vault** (or a minimal proxy clone). Its role is to hold revenue and enable claims.

Vault revenue sources:

- Publisher share from primary sales
- Publisher royalty from secondary sales
- Licensing fees, tips, donations
- Ancestral shares from derivative chains

Balances are tracked per token.

RoyaltySplitter: Internal Stakeholder Distribution

Publishing royalties rarely belong to a single address. Each Edition defines a stakeholder table (publisher, author, translator, editor, etc.) with percentage allocations.

Distribution can be implemented via:

- an internal ledger inside the Vault
- a separate RoyaltySplitter contract

This enables on-chain automation instead of off-chain manual payouts, with batch and claim optimizations for large stakeholder sets.

RoyaltyPolicyLAP / RoyaltyPolicyLRP

These policy contracts do not enforce marketplace fees. Their sole responsibility is to compute and route ancestral royalties in derivative graphs.

- **LAP (Linear Ancestral Policy):** fixed percentages routed up the stack
- **LRP (Linear Relay Policy):** royalties decay as they propagate upward

Policies are funded exclusively from the publisher royalty portion. Seller proceeds (80%) and platform fees (10%) are never affected.

DRMAccessModule

Reading access is ownership-based but enforced at the application level.

The DRMAccessModule:

- checks `hasAccess(wallet, editionId)`
- verifies ownership via NFBNFT
  enforces access rules (session limits, revocation, watermarking)

Ownership remains immutable on-chain; access logic can evolve independently.

Event Logging & Auditability: On-chain Audit Map

All critical actions emit structured events:
EditionRegistered, MintRequested, MintApproved, SaleExecuted, SecondaryListed, RoyaltyRouted, VaultClaimed, DerivativeRegistered, LicenseTermsSet

The General Admin panel indexes these events to answer:
"Who earned what, from which sale, verified on-chain?"

This is one of the strongest trust layers for publishers and investors.

Upgradeability & Security

In V1, core components (Registry, Router) are upgradeable via proxy.
Edition-level royalty policies and splits are immutable once minted. Changes require minting a new edition aligned with legal and publishing standards.

Marketplace contracts are hardened against:

- reentrancy
- approval abuse
- fee rounding errors
- front-running
- price manipulation

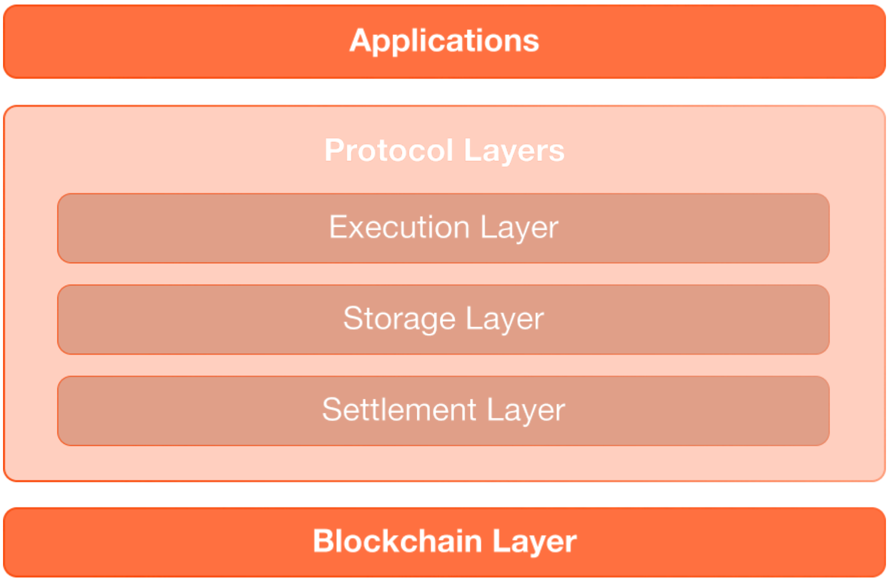
Claim flows are permissionless, with batch and rate-limit protections against gas griefing.

TECHNICAL ARCHITECTURE

System Overview

NFB is designed as a hybrid Web3 architecture operating within the Ethereum ecosystem. The system provides a balanced structure between on-chain verifiability, off-chain performance, and royalty security.

The architecture consists of four main layers:



Blockchain Infrastructure

Network Selection

Primary Network: Ethereum
Deployment Model:

- Mainnet: Governance, Treasury, Rare NFB registry ,Marketplace, minting, transfers

Selection Criteria:

$NFB Token Contract

Purpose

$NFB is the native utility + governance token of the NFB Marketplace:

- **Payments:** Platform fees can be paid using $NFB
- **Governance:** Voting power in DAO governance processes
  Staking: Read-to-Earn boosts, token-gating, publisher bonds & slashing mechanisms

Core Token Model

- **Hard Cap:** Fixed **MAX_SUPPLY**
- **Burn Mechanisms:**
  - Fee burn
  - Penalty burn
- **DAO Parameters:**
  - `feeDiscount`
  - `burnRate`
  - staking tiers
  - treasury allocations

Roles / Controls

- **DEFAULT_ADMIN_ROLE:** Restricted to timelock / multisig only
- **MINTER_ROLE:** Optional; disabled outside of genesis
- **PAUSER_ROLE:** Emergency pause authority

Minting authority is disabled after the initial distribution or strictly limited to the VestingVault contract.

```
solidity

contract NFBToken is ERC20, AccessControl, Pausable {
    uint256 public constant MAX_SUPPLY = 1_000_000_000e18;
    bytes32 public constant PAUSER_ROLE = keccak256("PAUSER_ROLE");

    uint256 public totalBurned;

    constructor(address treasury, address timelock) ERC20("Non-Fungible Books", "NFB") {
        _mint(treasury, MAX_SUPPLY);
        _grantRole(DEFAULT_ADMIN_ROLE, timelock); // DAO timelock admin
    }

    function burn(uint256 amount) external whenNotPaused {
        _burn(msg.sender, amount);
        totalBurned += amount;
    }

    function pause() external onlyRole(PAUSER_ROLE) { _pause(); }
    function unpause() external onlyRole(PAUSER_ROLE) { _unpause(); }
}
```

Book NFT Contract (ERC-721)

Edition-Based Minting

Each “book” is defined on-chain as an **Edition**, and every minted copy corresponds to a **tokenId.**

Edition attributes include:

- **ISBN** (unique)
- **maxSupply** (edition size)
- **royaltyBps** (EIP-2981 compliant)
- **rare flag** (collector edition indicator)
- **publisher / author splits** (defined in the distribution contract)

```
solidity
```

```
struct Edition {
    string isbn;
    uint256 maxSupply;
    uint256 minted;
    uint96 royaltyBps;     // 0–10000
    bool isRare;
    address publisher;
    bytes32 contentCIDHash; // IPFS CID hash / pointer
}
```

Key Rules

- **Immutable edition rules:** `maxSupply` and `ISBN` are immutable
- **Mint limit:** `minted < maxSupply`
- **EIP-2981 compliance:** `royaltyInfo` calculates royalties based on the sale price

Marketplace Contract

Core Responsibilities

When a sale is executed, the Marketplace contract performs the following actions atomically within a single transaction:

- Validates the listing (active status, correct price, seller ownership)
- Calculates platform fees and royalties
- Distributes payments atomically
  - fee → treasury
  - royalty → distributor
  - remainder → seller
- Transfers the NFB
- Emits events

Payment Modes

- **USDC / USDT** (ERC-20)
- **$NFB** (ERC-20) → eligible for fee discount

- **Native ETH**

Fee & Discount Model

- **platformFeeBps**
- **nfbFeeDiscountBps**

Effective fee calculation:

- Paid with USDC / USDT → `feeBps = platformFeeBps`
- Paid with $NFB → `feeBps = platformFeeBps × (1 - discount)`

Royalty Distribution

- `royaltyInfo(tokenId, salePrice)` (EIP-2981) returns only the receiver and amount
- In NFB, the receiver is typically the RoyaltyDistributor, which then applies the author / publisher revenue split

Marketplace Data Structures

```
solidity

struct Listing {
   address seller;
   uint256 price;
   address payToken; // USDC / USDT / NFB
   bool active;
}
mapping(uint256 => Listing) public listings;

uint256 public platformFeeBps;       // e.g., 1000 (10%)
uint256 public nfbDiscountBps;       // e.g., 2500 (25% discount on fee)
address public feeTreasury;          // DAO treasury
address public royaltyDistributor;   // splits royalties
```

Buy Flow

```
solidity

function buy(uint256 tokenId) external nonReentrant {
    Listing memory l = listings[tokenId];
    require(l.active, "NOT_LISTED");

    address seller = l.seller;
    uint256 price = l.price;
    address payToken = l.payToken;

    // 1) Ownership check
    require(nft.ownerOf(tokenId) == seller, "SELLER_NOT_OWNER");

    // 2) Fee calculation (NFB discount if payToken == NFB)
    uint256 feeBps = platformFeeBps;
    if (payToken == address(nfbToken)) {
        // feeBps = feeBps * (1 - discount)
        feeBps = (feeBps * (10_000 - nfbDiscountBps)) / 10_000;
    }
    uint256 platformFee = (price * feeBps) / 10_000;

    // 3) Royalty calculation (EIP-2981)
    (address royaltyReceiver, uint256 royaltyAmount) =
        IERC2981(address(nft)).royaltyInfo(tokenId, price);

    // 4) Seller proceeds
    uint256 sellerProceeds = price - platformFee - royaltyAmount;

    // 5) Pull payment from buyer & distribute atomically
    IERC20(payToken).transferFrom(msg.sender, feeTreasury, platformFee);
    IERC20(payToken).transferFrom(msg.sender, royaltyReceiver, royaltyAmount);
    IERC20(payToken).transferFrom(msg.sender, seller, sellerProceeds);

    // 6) Transfer NFT
    nft.safeTransferFrom(seller, msg.sender, tokenId);

    // 7) Close listing
    listings[tokenId].active = false;

    emit ItemSold(tokenId, seller, msg.sender, price, payToken, platformFee, royaltyAmount);
}
```

Security & Edge Cases

- **ReentrancyGuard:** Payment and transfer flows are protected against reentrancy attacks
- **Checks–Effects–Interactions:**
  All validations are performed first, followed by state updates, and finally external transfers
  (The execution order can be optimized accordingly if needed.)
- **Approval safety:**
  `safeTransferFrom` is enforced, and explicit marketplace approval is required
- **Price manipulation protection:**
  Listings use a **price snapshot** mechanism; if the price changes, the transaction reverts
- **Royalty bypass prevention:**
  All secondary sales are enforced exclusively through the marketplace contract

Events

solidity

```
event ItemListed(uint256 indexed tokenId, address indexed seller, uint256 price, address
payToken);
event ItemSold(uint256 indexed tokenId, address indexed seller, address indexed buyer,
            uint256 price, address payToken, uint256 fee, uint256 royalty);
event ListingCancelled(uint256 indexed tokenId);
```

Storage Architecture

Content Storage

Content Type	Storage
Digital NFB	IPFS
Rare NFB	IPFS + Arweave
Metadata	IPFS JSON

Flow

- The file is encrypted using **AES-256**
- The encrypted file is uploaded to **IPFS**
- The **CID** is recorded on-chain

DRM & Access Control

Model: Ownership-based DRM

mathematica

```
NFBOwnership
  ↓
```

```
Wallet Signature
  ↓
Backend Verification
  ↓
Time-limited DRM Token
  ↓
Decryption Key
```

Features

- Device limit
- Offline reading
- Invisible watermark

Backend Architecture

Stack

- Node.js
- PostgreSQL
- Redis
- IPFS SDK

```
css

Backend Services
├── Auth Service
├── Publisher Service
├── Marketplace Service
├── Blockchain Adapter
├── Content & DRM Service
└── Analytics Service
```

Frontend Architecture

Web

- Next.js
- ethers.js
- wagmi
- WalletConnect

Mobile

- React Native
- Secure Enclave / Keystore
- Biometric auth

Security Architecture

Smart Contracts

- OpenZeppelin base
- 3rd party audit
- Timelock upgrades
- Emergency pause

Infrastructure

- HTTPS + HSTS
- WAF + rate limit
- Secrets Manager
- RBAC

Utility & Use Cases

The $NFB Token is the utility and governance token of the NFB Marketplace ecosystem. Its design supports in-platform economic activity, incentivizes long-term participation, and enables the sustainable scaling of a content-centric Web3 publishing economy.

The core role of $NFB is to unify value transfer, incentive mechanisms, access authorization, and governance functions under a single coordination layer. While the supply side follows a controlled distribution and vesting model, the demand side is driven by direct, in-product utility.

$NFB Token Utility

Platform Fee Discounts

On primary and secondary sales executed on the NFB Marketplace, users and publishers receive fee discounts when platform commissions are paid in $NFB instead of USDT/USDC.

- **Maximum discount rate:** 10%
- **Applicable to:** Marketplace transaction fees, premium services, licensing fees

This mechanism creates continuous, product-driven demand for the $NFB token.

Read-to-Earn Multiplier

NFB's Read-to-Earn system is based on verified reading behavior, rather than passive click-based rewards. The amount of rewards a user earns increases proportionally with the amount of $NFB staked.

This model:

- Reduces circulating supply
- Increases long-term user retention
- Enables more balanced reward pool distribution

Governance & Protocol Voting

$NFB holders participate in decision-making through the NFB DAO, influencing critical platform parameters. Governance is designed to balance operational efficiency with community participation.

Governance topics include:

- Platform commission rates
- Burn schedules and burn rates
- Read-to-Earn reward weights
- Featured categories and themes
- Allocation of ecosystem incentive budgets

Voting is executed using a snapshot + time-lock execution model.

Token Gating & Access Authorization

$NFB functions as an authorization key for content and community access.

Example scenarios:

- Minimum $NFB balance required for pre-sale participation
- Access to exclusive author content or live events
- Entry to private Discord / community channels

This structure transforms the token into not only a financial asset, but also a social and cultural value instrument.

Publisher Bond Mechanism

To protect publishing quality and royalty integrity, NFB implements a $NFB bonding mechanism for publishers.

- A specified amount of $NFB is locked before content submission
- In cases of copyright infringement, spam, or malicious behavior, part or all of the bond may be burned

This mechanism:

- Deters spam and low-quality content
- Increases platform trustworthiness

- Incentivizes publishers to maintain long-term quality standards

Licensing & AI Payments

When enterprises or AI developers acquire commercial licenses through the NFB Marketplace, payments are executed in **$NFB**.

Use cases include:

- Dataset access
- Training licenses
- Derivative content usage rights

Payments are automatically distributed to the relevant rights holders via smart contracts.

Renting & Lending

NFB owners can rent out their digital books for predefined periods.

- Rental fees and collateral are paid in $NFB
- Access automatically expires at the end of the rental period
- Ownership is never transferred

This enables time-based monetization while preserving permanent ownership rights

Real-World Use Cases

Scenario 1: Reader & Secondary Market

User: Elif – University student

- Elif purchases the book “Economics 101” for 50 USDT
- At the end of the semester, she lists the book for sale on the secondary market for 40 USDT

Revenue Distribution:

- **Elif:** 32 USDT (80%)
- **Publisher:** 4 USDT (10%)
- **Platform:** 4 USDT (10%)

➡ Elif effectively uses the book at a net cost of 18 USDT

➡ The publisher continues to receive royalties even from second-hand sales

Scenario 2: Author & Programmable Licensing

User: Ahmet – Independent author

- While uploading his book, Ahmet activates a commercial derivative license
- Revenue share: 15%
- An artist wants to adapt the book into a graphic novel and purchases the license in a single transaction

➡ 15% of revenue from every graphic novel sale is automatically routed to Ahmet
➡ No legal process, manual contracts, or intermediaries are required

Scenario 3: Artificial Intelligence & Ethical Data Usage

User: AI-Tech Corp

- Requires copyright-cleared and permission-based Turkish novels
- Filters opt-in content through the Marketplace API
- Pays licensing fees in bulk using **$NFB**

➡ Payments are distributed instantly and transparently to the authors whose works are used

➡ AI training becomes ethical, consent-based, and rights-respecting

Scenario 4: Collector & Rare Works

User: Can – Digital collector

- Can purchases a Rare NFB limited to only 100 copies by a renowned author
- The work includes an exclusive author video stored within a Token Bound Account

After 3 years:

- He sells the work at auction for 10× the original price
- The author and publisher automatically receive royalties from the sale

Governance Model

The NFB Marketplace adopts a Progressive Decentralization approach to balance long-term sustainability, censorship resistance, and the trust requirements unique to the publishing ecosystem. This model enables rapid product development and a strong security focus in the early stages, while gradually transferring governance authority to $NFB token holders as the ecosystem matures.

NFB governance deliberately avoids the classic "DAO from day one" approach. In areas such as publishing, intellectual property, and royalties, arbitrary rule changes pose a direct threat to platform trust. Therefore, the governance architecture is built on the principles of controlled transition, transparency, and immutability.

Progressive Decentralization Strategy

Governance authority is transferred across three main phases:

Phase 1: Incubation & Development

Governance:
NFB Core Team

Focus:

- Achieving product–market fit
- Stabilizing marketplace and royalty modules
- Rapidly resolving security vulnerabilities
- Standardizing publisher onboarding processes

Community Role:

- Advisory only
- Feedback via Discord, forums, and closed beta programs
- No binding votes

The goal of this phase is not decentralization, but to prevent irreversible damage caused by premature or misaligned governance.

Phase 2: Participatory Governance

Governance:
NFB DAO (Beta) + Core Team (limited veto rights)

Mechanism:

- The community begins voting on selected parameters
- Snapshot-based, gasless voting is introduced

Votable Topics (examples):

- “Featured Category of the Month” (which book genres receive higher Read-to-Earn rewards)
- Allocation of a defined portion of the marketing budget
- Prioritization of ecosystem incentive programs

This phase serves as a transition period to test community governance reflexes and measure DAO operational capacity.

Phase 3: Full DAO & Community Control

Governance:
NFB DAO

Authority:

- Allocation of protocol revenues
- Platform commission rates

- Token burn policies
- Full treasury management
- Smart contract upgrades

At this stage:

- Core team admin keys are burned, or
- Transferred to time-locked contracts

The protocol becomes technically and operationally community-controlled.

NFB DAO Structure & Voting Mechanism

The NFB DAO is designed to be fair, manipulation-resistant, and to reward long-term participation.

Proposal Submission

- Any wallet holding at least 100,000 $NFB
- May submit an NFB Improvement Proposal (NIP) via the governance forum

Each proposal must include:

- Technical impact analysis
- Economic implications
- Implementation timeline

Voting Power

NFB does not use a “1 token = 1 vote” model.
Instead, it applies Time-Weighted Voting:

- Tokens are staked and locked
- Voting power increases with lock duration

Example:

- Unlocked tokens: 1× voting power

- 6-month lock: 2×
- 12-month lock: 4×

This mechanism:

- Limits short-term speculation
- Empowers long-term participants

Quorum Requirements

For a proposal to be valid:

- At least 10% of total voting power must participate

For critical protocol changes, a higher quorum may be required.

Governance Scope

The NFB DAO functions as the platform’s legislative body, with authority over:

Economic Parameters

- Platform transaction fees
- Base royalty rates for secondary sales
- Token burn frequency and amounts

Treasury Management

- Ecosystem fund allocation
- Grants and incentive programs
- Strategic partnerships

Content & Ecosystem Policies

- Prohibited or incentivized content categories
- Ecosystem growth strategies

Protocol Upgrades

- Integration of new modules
- Major contract upgrades

Immutability of Royalty & Content Rules

Work-level royalty rules are explicitly protected in NFB governance.

Once an **Edition** is minted:

- Royalty percentages
- Stakeholder distributions
- License terms

cannot be changed retroactively.

If changes are required:

- A new Edition is created
- Rights of the original Edition remain fully preserved

This approach provides legal and commercial certainty for publishers and authors.

Decentralized Justice & Dispute Resolution

NFB implements a decentralized dispute resolution mechanism for critical issues such as copyright infringement and stolen content.

Digital Court Process

1. **Report:** A user submits a claim with evidence and deposits $NFB as collateral to prevent abuse.

2. **Jury Selection:** Anonymous expert jurors are randomly selected from the DAO.

3. **Ruling:** If infringement is confirmed:

    - The work is frozen
    - Publisher bond is slashed
    - Whistleblower is rewarded

4. **Appeal:** The publisher may appeal by staking a higher bond.

This system removes the need for centralized moderation and creates a self-policing ecosystem.

Economic Model & Revenue Structure

The NFB Marketplace economic model is built on:

- Sustained secondary market activity
- Automated royalty flows
- Persistent token demand

The model aligns platform revenue with ecosystem participant incentives.

Revenue Streams

1. Marketplace Fees

Every primary and secondary sale on the NFB Marketplace incurs a fixed platform fee.

- **Standard Platform Fee:** 10%
- **$NFB Payment Discount:** 25% discount on fees

Payment Method	Platform Fee
USDC / USDT	10%
$NFB	7.5%

All fees are transferred to the DAO Treasury.

2. Secondary Market Royalty Flow

Revenue from secondary sales is split as follows:

- Seller: 80%
- Publisher / Author Royalty: 10%
- Platform: 10%

Royalties are automatically distributed via the RoyaltyDistributor contract.

3. Read-to-Earn: Non-Inflationary Yield

Read-to-Earn rewards:

- Distributed from a pre-allocated Ecosystem Allocation pool
- Staked $NFB increases reward multipliers
- Circulating supply decreases through staking locks

This model converts verified reading activity into economic value.

4. Publisher Bond & Slashing

Publishers must lock a specified amount of $NFB before publishing content.

- Copyright infringement / spam → bond slashed
- Verified publishers → bond withdrawable

This mechanism provides:

- Quality filtering
- Deflationary pressure
- Platform security

5. Licensing & AI Revenues

- Enterprise licenses
- AI training data access
- Derivative content licenses

Payments:

- Made in $NFB
- Distributed directly to rights holders via Royalty Vaults

Token Value Capture Mechanisms

Core mechanisms supporting $NFB value:

- Fee payment discounts
- Staking & reward boosts
- Governance requirements
- Publisher bonding
- Token burn

- Token-gated access

$NFB is not merely a speculative asset; it is the operational fuel of the ecosystem.

Security & Data Ownership

NFB cryptographically and contractually addresses the publishing industry's core challenges: ownership, access, and royalty security.

Digital Ownership

- Books are held as ERC-721 NFTs in user wallets
- Even if the platform shuts down:
  - NFT ownership remains intact
  - IPFS CIDs remain verifiable on-chain

Ownership = control of the wallet's private key

Content Integrity & DRM

- Content is encrypted using **AES-256-GCM**
- Decryption keys are provided only to users who:
  - Prove NFT ownership
  - Receive a time-limited DRM token

- File hashes are anchored on-chain → immutability guaranteed

Royalty & Revenue Security

- Royalties are smart-contract enforced
- No manual intervention
- Sellers cannot bypass royalties
- Secondary market royalties cannot be avoided

Smart Contract Security

- Reentrancy guards
- Checks–effects–interactions pattern
- Immutable edition rules
- DAO-controlled upgrades with timelocks
- Emergency pause mechanisms

All critical contracts:

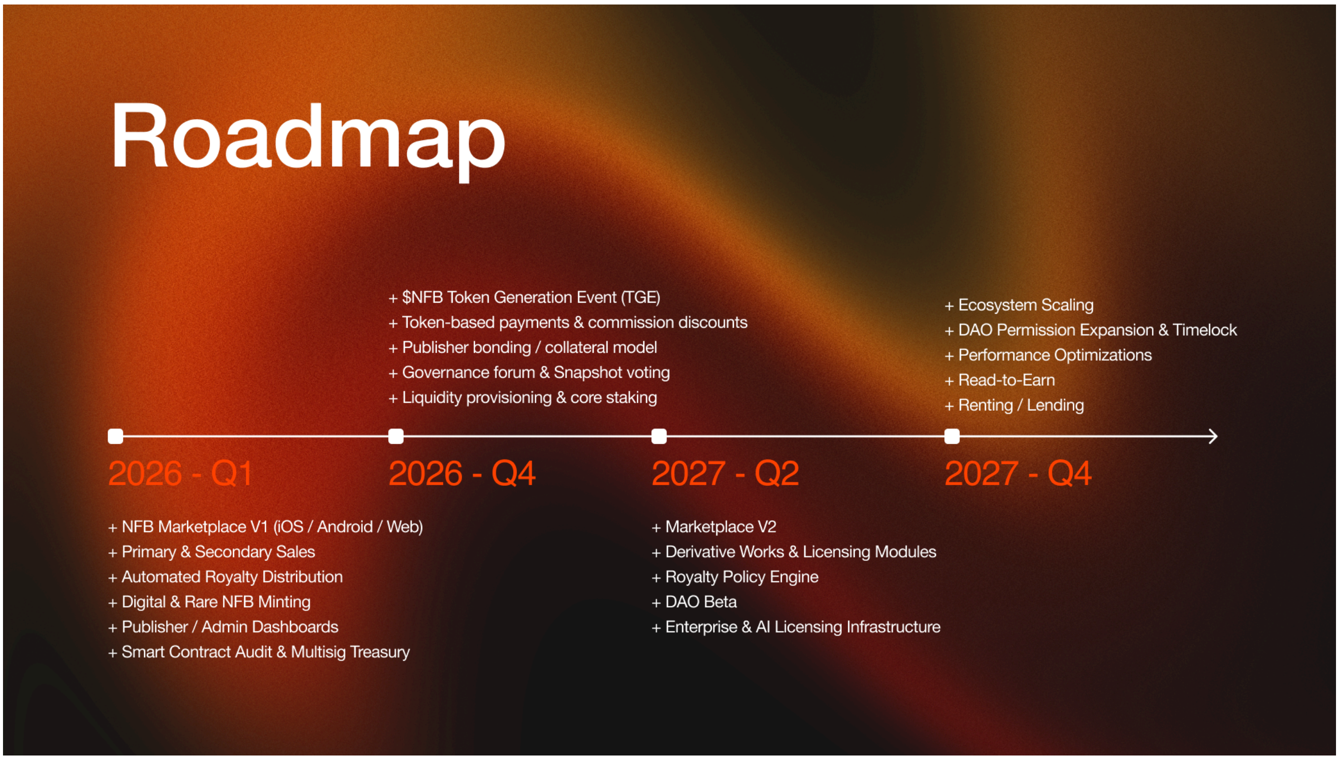
- Are externally audited
- Emit public event logs
- Operate with deterministic distribution rules

Data Ownership & Privacy

- Personal user data is stored off-chain
- On-chain data includes only:
  - Pointers
  - Hashes
  - Ownership records

Architecture is GDPR / KVKK compliant.

ROADMAP



Roadmap

2026 - Q1

+ NFB Marketplace V1 (iOS / Android / Web)
+ Primary & Secondary Sales
+ Automated Royalty Distribution
+ Digital & Rare NFB Minting
+ Publisher / Admin Dashboards
+ Smart Contract Audit & Multisig Treasury

2026 - Q4

+ $NFB Token Generation Event (TGE)
+ Token-based payments & commission discounts
+ Publisher bonding / collateral model
+ Governance forum & Snapshot voting
+ Liquidity provisioning & core staking

2027 - Q2

+ Marketplace V2
+ Derivative Works & Licensing Modules
+ Royalty Policy Engine
+ DAO Beta
+ Enterprise & AI Licensing Infrastructure

2027 - Q4

+ Ecosystem Scaling
+ DAO Permission Expansion & Timelock
+ Performance Optimizations
+ Read-to-Earn
+ Renting / Lending

Team & Advisors

NFB is founded and developed by a disciplined, multidisciplinary, and experienced team with strong expertise across publishing, blockchain technologies, and digital asset management. The team structure is designed to cover product development, protocol architecture, publishing operations, legal, finance, marketing, and visual communication, ensuring both technical depth and operational scalability.

Mehmet Arslantunalı – Founder
Founder and vision lead of NFB. Responsible for overall strategy, business model, publishing philosophy, and long-term growth planning. Architect of NFB's vision that bridges traditional publishing with Web3 infrastructure.

Büşra Eldener – Co-Founder
Plays an active role in operations, publisher relations, and product processes. Oversees daily platform operations, content policies, and partner coordination.

Tunç Erkan – Co-Founder
Focuses on strategic planning, business development, and ecosystem relationships. Actively involved in corporate partnerships and long-term expansion strategies.

Nevzat Çalışkan – Co-Founder
Responsible for financial structure, operational processes, and organizational coordination. Contributes to the sustainability of the protocol's business model.

Kazım Selman Poyraz – Co-Founder
Works on core software development. Focuses on backend services, system integrations, and performance-oriented infrastructure components.

Sevgi Berk – Co-Founder
Responsible for application interfaces, user experience, and frontend architecture. Leads product design processes by aligning technical requirements with user-centric design.

Ahmet Sütbaş – Co-Founder
Provides strategic guidance at the founder level. Supports long-term vision, product positioning, and organizational decision-making processes.

Murat Yıldırım – CTO & Partner
Technical lead of NFB. Responsible for smart contracts, marketplace architecture, DRM integration, and scalable infrastructure. Protocol security and system architecture are developed directly under CTO leadership.

Dr. Onur Baran Çağlar – IT Consultant & Partner
Provides consultancy on software architecture, data security, and enterprise-grade technology systems. Contributes to the platform's long-term technical roadmap.

Cemil Şinasi Türün – Blockchain & Crypto Asset Advisor
Advises on digital asset economics, token models, and market dynamics.

İsmail Hakkı Polat – Blockchain & Crypto Asset Advisor
Provides strategic input on Web3 ecosystems, crypto asset regulations, and the digital economy.

Doğu Taşkıran – Strategic Technology Advisor
Advises on long-term technology strategy, innovation, and emerging product domains.

Hüseyin Karslıoğlu – Sistem Global / SGlobe & Partner
Supports corporate structuring, financial consultancy, and international business development processes.

Burak Özturan - Blockchain Developer

Büşra Sütbaş - Blockchain Developer

Gamze Güler – Marketing Manager
Responsible for marketing strategy, community growth, campaign design, and Web2–Web3 communication. Manages brand positioning and user acquisition strategies.

Atakan Kelleci – Content Manager
Oversees platform content, publisher communication, and editorial process coordination.

Zeynep Kaynak – Content Creator
Produces social media content, storytelling assets, and community-focused creative materials.

Gülşen Sarıgöz – Community Manager
Manages community engagement and ecosystem communication across Discord, X, and other social channels.

Faruk Kaya – Graphic Designer
Responsible for brand visual identity, UI/UX assets, and marketing materials.

Zülal Bakacak – Graphic Designer
Works on digital design, publisher storefronts, and in-platform visual assets.

Cansu Şimşek – Graphic Designer
Contributes to campaign visuals, social media designs, and creative production workflows.

Gülşen Akay – Accounting
Manages accounting, financial reporting, and operational financial processes.

Barbaros Aydoğdu – Supply Chain Manager
Responsible for publishing supply processes, digital content flow, and operational

coordination. Ensures seamless operation across the publisher–content–user value chain.