The Book

Pay-per-post publishing with verifiable records, permanent storage, and auditable community actions.

Website	dbook.dev
Contact	hello@thebook.xyz
Version	Draft v0.1
Date	January 07, 2026

Summary

The Book is a minimalist publishing platform that makes digital publishing actions provable. Creators publish articles, run polls, and launch petitions. Content is stored permanently, while key actions are anchored to a public ledger so anyone can verify authorship, version history, votes, and signatures.

At a glance

What it is	Pay-per-post publishing plus auditable polls and petitions.
Core user	Authors and organizers who want durable publishing and verifiable public actions.
Storage	Content on Arweave (or similar). Records on Solana.
Monetization	Per-post unlocks, optional tokenization after publish, storage sold as a utility.

This is an early draft intended to align product, architecture, and economics. It is not a legal contract, and it does not offer securities or investment advice.

1. Executive summary

The Book focuses on a simple idea: publishing and community actions should be verifiable without relying on one company's database. It combines three workflows that often live in separate products:

- **Articles**: publish free or paid posts with a verifiable version trail.
- **Polls**: run on-chain votes with auditable tallies.
- **Petitions**: collect signatures that can be independently verified.

The platform stores content on permanent storage (e.g., Arweave) and anchors metadata and critical actions on Solana. This split keeps costs practical while making the key claims provable: what was published, by whom, when, and what the community did.

Design principles

- Reader and creator value first. Avoid jargon in the product surface.
- Proof where it matters. Anchor authorship, versions, votes, and signatures.
- Low friction. Browsing is free. Wallet actions are only required when something must be recorded.
- Utility pricing. Storage is treated as a consumable resource (MB/GB), not a confusing credit system.

Near-term goal: make it easy for creators and organizers to publish durable, shareable, verifiable artifacts. Long-term goal: become an aggregator for rights-bearing digital goods starting with writing, then expanding to other formats where provenance and durable access matter (guides, research, templates, public records).

2. Problem

Today’s publishing platforms optimize for feeds, ads, and centralized control. For many categories of content, this creates three issues:

- **Weak provenance**: it is hard to prove what was said at a specific time, or what changed later.
- **Low trust in outcomes**: polls and petitions are easy to manipulate if the counting is opaque.
- **Fragile economics**: creators often rely on subscriptions or platform distribution; both are brittle.

What users want instead

- A stable link that keeps working years later.
- Proof of authorship and a verifiable change log.
- Auditable community actions with transparent rules.
- Simple, per-item monetization where it makes sense (pay to unlock a specific post).

Non-goals

- We are not building a general social network feed.
- We are not making every post a speculative asset by default.
- We do not claim to prevent all misinformation. We aim to make records and actions verifiable.

3. Product overview

The Book supports three first-class content types. Each has a consistent lifecycle: draft, publish, verify, and share.

Content type	Purpose	What is verifiable
Article	Publish writing (free or paid).	Authorship, timestamp, version history, optional unlock receip
Poll	Capture community preferences.	Question/options, vote records, tally rules, final results.
Petition	Collect support for a demand.	Petition text, signature records, total count, optional targets.

Paid posts

Paid posts are unlocked via a one-time wallet payment. This avoids subscription fatigue and makes pricing explicit per item. Unlock receipts can be used to prove access rights and enable future features like revenue-sharing, bundling, or transferable access.

Optional minting after publishing

Posts can be minted after publication. This keeps the authoring flow simple and avoids forcing tokenization on every creator. Minting can be used to attach explicit rights to a piece, such as revenue entitlement on future unlocks.

> **UX rule:** browsing is free. Wallet prompts only occur when an action must be recorded or funds must move (publish, vote, sign, unlock, mint).

4. Architecture and verification

The Book separates content storage from verification records. This keeps publishing affordable while preserving integrity.

Layer	What it stores	Why it exists
Permanent storage (Arweave)	Posts (HTML/JSON), attachments, media references	Stable access and long-term availability.
Solana records	Metadata pointers (URIs), authorship, versions, links, polls, petitions, signatures	Fast, auditable verification and ownership.
Client + indexer	Caching, search, previews, notifications	Usability, discovery, and product-grade UX.

Wallet actions and signatures

Wallet prompts can feel confusing. The product should make the meaning explicit:

- **Message signature**: proves wallet ownership for authentication. It does not move funds.
- **Transaction signature**: changes on-chain state and may require network fees. It can also move funds for unlocks or storage.

A recommended UX pattern is a one-time preflight dialog the first time a user publishes or buys: it lists the expected signatures and what they do, with a “don’t show again” option.

Integrity model

- Every published artifact includes a content hash and a pointer to permanent storage.
- Edits are handled as new versions linked to prior versions. Readers default to the latest.
- Polls and petitions publish their rules (revoting, eligibility, targets) alongside their records.

5. Economics

The Book’s economics are designed to be understandable to mainstream users. There are three primary levers:

- **Paid unlocks**: users pay to unlock a specific post. Authors earn immediately.
- **Utility storage**: authors purchase storage (MB/GB) that is consumed during publishing.
- **Optional ownership rights**: minted posts can carry rights, such as a share of future unlock revenue (design space).

Storage as a utility

Instead of “publishing credits,” the product should surface a storage balance and show MB/GB remaining at publish time. If storage is insufficient, the publish action should be blocked with a clear call to purchase more storage before opening the wallet.

Revenue rights and transfers (optional)

A rights-bearing NFT can represent an entitlement tied to a post (e.g., future unlock revenue). This is optional by design and should not be required to read or publish. If implemented, rights should be explicit in the UI and enforced by contracts, not by promises.

What we will measure

- Paid unlock conversion rate (views to unlock).
- Repeat unlock rate per reader (reader retention).
- Creator earnings per post (and distribution across creators).
- Cost to publish (storage + fees) and failure rate at publish time.

6. Roadmap and open questions

Near-term roadmap (0-3 months)

- Reduce wallet friction (single signing where possible, clear preflight explanations).
- Default cover image from first in-editor image, with “Set as cover” override.
- In-app notifications (bell + unread badge), generated off-chain by indexing actions.
- Targets for petitions: role/office-first, sources required, community verification workflow.

Mid-term roadmap (3-9 months)

- Improved discovery: search, tags, reputation signals, bookmarks, and saved lists.
- Revision viewer: diff or version browser for posts and petition changes.
- Creator tooling: analytics, unlock funnels, audience export, and delivery packs for petitions.
- Expanded asset formats: templates, research reports, and other rights-bearing digital goods.

Risks

- **UX friction**: wallet prompts and storage flows can add drop-off. Must be minimized and explained.
- **Moderation**: comments and targets can be abused. Start with low-risk features (bookmarks, in-app notifications) and add controls.
- **Cost structure**: storage and indexing costs must be predictable; pricing must be simple for creators.
- **Regulatory ambiguity**: rights-bearing assets must be designed carefully to avoid misleading users.

Contact

Email: **hello@thebook.xyz**
Website: **dbook.dev**