THE WORLD LIBRARY

Web3 Standard *for the public domain*

***RA-HULA** has been designed from the ground up with the World Library project.



2,766
Public-domain books

Arweave
File storage

Gnosis
Collection verification

© 2026 DATAPOND D-LIBRARY PTY LTD

*RA-HULA Standard Origin

The ***RA-HULA** standard is compatible with UNIMARC at the level of bibliographic data. Libraries around the world can use ***RA-HULA** to store and retrieve their public-domain books using Web3.

The storage network is designed to keep books permanently after a one-time payment, without ongoing upkeep for the library. A cryptographic record lets anyone check that a book is exactly as it was published. An altered copy fails that check, and removing a book from a website leaves the published record intact.

I chose a new name, ***RA-HULA**, because I was free to set my own direction. I built it on my own time and paid for it with my personal savings, without government or institutional backing, sponsors, or bureaucratic red tape.

***RA-HULA** builds on sophisticated bibliographic standards and goes further. It can classify the same record under several systems, describe it in multiple languages, and model other kinds of records beyond books. All three capabilities are built in.

1. What Web1 inherited from librarians
2. What a like is worth
3. Why publish on Web3?
4. How I built the World Library

What Web1 inherited from librarians

Long before anyone put a catalogue on a website, librarians had worked out how to send one another a catalogue card without putting it in the post. In the 1960s, MARC (Machine-Readable Cataloging) gave each part of a description a tag a computer could recognise. A librarian could describe an edition once, and a colleague elsewhere could reuse the work. Web1 brought that cooperation to the reader’s desk. You could look up a book in a library you had never entered.

Countries had developed different MARC formats, among them USMARC, CAN/MARC and Britain’s UKMARC. In 1977, IFLA introduced UNIMARC to help institutions exchange records across those differences. The American and Canadian formats later came together as MARC 21. Their differences still matter to anyone moving a catalogue today. A title recorded in MARC 21 field 245 belongs in field 200 in UNIMARC.

Years later, BIBFRAME took up a question familiar to anyone who has asked for one edition and received another. There is the work Jane Austen wrote, a publisher’s particular edition of Pride and Prejudice, and the copy with a library stamp inside the cover. BIBFRAME gives each its own identity and links them. I wanted a reader who follows a link to a book to know which of those things they will find.

MARC 21 UNIMARC BIBFRAME

What a like is worth

Web2 made it easy for a reader to add something of their own. A note in the margin could travel across an ocean. A recommendation could introduce a stranger to a writer their local library had never stocked. I wanted to keep that generosity. The difficulty begins when the company carrying the recommendation can earn more from keeping the reader scrolling than from helping them find the book.

A like has two audiences. To a friend it may mean “you’d enjoy this”. To an advertising platform it is evidence about what will hold your attention. A quiet interest in walking can become an invitation to buy a new jacket, followed by another timed to catch the feeling that you are missing out. The platform gets paid for reaching you. You pay out of your earnings for something your walk may never have required. Oversight has struggled to keep pace with this kind of targeting.

The price on that jacket also leaves things out. A supplier can cut costs by clearing habitat for fibre production. The saving shows up in this season’s margin. A woodland and the species that depend on it can take generations to recover. A library has a choice much earlier in that chain. When someone comes to read about walking, I want them to be able to close the book and go outside.

Anti-Cheat FTC · A Look Behind the Screens UNEP · Global Resources Outlook

Why publish on Web3?

Suppose a library pays to publish a collection today. Why should a child who finds it in twenty years have to pay again to look it up? A blockchain lets independent computers agree on a public record. Once the network has confirmed a publication, a reader can ask a node for that record without opening a wallet or sending a transaction. No platform has to collect a commission each time someone consults the catalogue.

The work still has a cost. Computers need electricity, storage and people to maintain them. Network rules reward the operators, and the publisher pays a transaction fee for each write. The World Library stores book files on Arweave, where an upfront payment helps fund long-term storage. It records a fingerprint of the collection on the blockchain. Hosting a website or serving downloads can still cost money. Reading, searching and downloading through the library are free and require no account.

The fingerprint gives a future librarian a practical test. Take a copy of the collection, calculate its fingerprint and compare it with the published one. Change even a byte and the result differs. A corrected release can acquire its own public record while the earlier commitment remains visible. That history can establish which version was published. Whether a date was catalogued correctly still takes a librarian’s judgement.

Ethereum · eth_call Arweave · Storage Endowment

How I built the World Library

I came to this as a business analyst and a programmer. I grew up in West and East Africa. Years later, at the Aboriginal Tent Embassy in Canberra, I learned about knowledge handed down by voice. That experience puts a limit on what I mean by a world library. A catalogue can preserve what someone has written down. Whole traditions live outside it. I built this standard independently, drawing on generations of librarians' work.

I began with the agreements two reading apps need in order to open the same book. What identifies the author? Where does the language of an edition belong? How does an old app recognise a title after the software changes? **Without knowing, I was reinventing the MARC standard.** I published common definitions for books, people and subjects, kept established identifiers where they exist, and gave fields permanent numbers. The public software kit turns those definitions into records an app can read. A second developer can build a different entrance to the collection.

Building the working library also forced a decision about who gets to choose what stays on a shelf. Readers can organise their own copies. A signed removal can be inspected and reversed, so someone who disagrees has a decision they can point to. That matters to me as much as keeping the files themselves. If a reader can find an overlooked book, check which edition it is, and show the next person how to reach it without asking me, the standard has done its job.

Data Pond · Web3 & UNIMARC @the_library/canonical-schemas

FAQ

▼ Do I need an account to read?

No. Reading, searching and downloading need no account, that is a condition of the licence. An account is only needed to save your own changes and activity on-chain.

▼ Does it work offline?

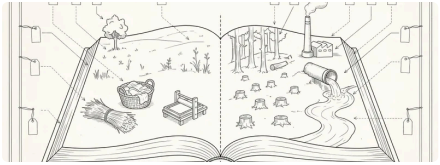
Books and pages you have opened are kept in your browser, and open without a connection when the app uses the optional offline worker. If space runs short, the least recently used books are removed first, never the open one, or ones you pinned.

RESEARCH FROM THE WORLD LIBRARY

Research

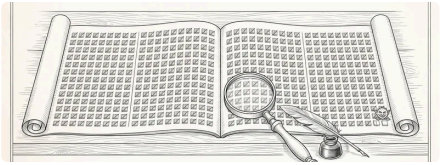
What building the World Library has taught us about classification, curation and trust.

Selected studies • In English



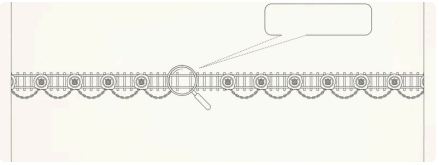
Who decides where a book belongs?

Follow one book through different classification systems and see how its neighbours change.



Count what gets removed

Explore a public record of removals and an experiment in ranking books by what readers keep.



Can humans and AI agents trust each other?

A Bitcoin timestamp can show that a note has stayed the same. What happens when someone remembers it differently?

All research (English)

DATA POND API

Developers

Build a reader for the shared collection using the public SDK and canonical schemas.

Read from the network

Create a read client, wait for the registry to load, then ask the network for its account count.

@the_library/public

```
pnpm add @the_library/public @the_library/network-config
```

```
import { DEFAULT_CHAIN_ID } from '@the_library/network-
config';
import { getReadClient } from '@the_library/public/web3';

const client = getReadClient(DEFAULT_CHAIN_ID);
await client.ready;
const members = await client.bouncerStorage.nbAccounts();
```

Read an identifier definition

The schemas describe the fields that apps share. This example reads the ISBN definition directly from the published JSON.

@the_library/canonical-schemas

```
pnpm add @the_library/canonical-schemas
```

```
import isbn from '@the_library/canonical-
schemas/standards/isbn'
  with { type: 'json' };

console.log(isbn.label);
// International Standard Book Number
```

Code licence

Use and share unmodified releases at no cost. Apps read the network; writing is reserved to authorised services.

D-Open Code Sovereign Licence