Web-based system for electronic publications cataloging using Elixir

Speaker:
Klochko Kostiantyn
MKI-251, Chernihiv Polytechnic National University
Academic advisor:
Ph. D. Rogovenko Andrii

What’re the problems?

What are the problems?

- Single Point of Failure (SPOF): central server outages block access to scientometric data for researchers worldwide.
- Vendor Lock-in & Closed Access: high API costs and access restrictions of commercial databases (Scopus, Web of Science).
- Administrative Delays: moderation and addition of new publications in centralized registries can take weeks.
- Lack of Cryptographic Guarantees: risks of metadata forgery and compromised authorship in distributed data entry.

What are the problems?

Multiple Sources & Ingestion Challenges (Metadata Sourcing) Issues:

- Data is fragmented across heterogeneous repositories (ORCID, DOI, PubMed, arXiv, ISBN catalogs).
- Significant discrepancies in metadata quality and descriptions (unmatched transliterations, incomplete author lists).
- Integrity risks: lack of data origin verification during transit (mitm, metadata poisoning)

What are the problems?

Centralized systems limitations:

- Scopus: Proprietary, paid access, centralized moderation bottlenecks.
- Google Scholar: complete absence of an official API, closed ranking algorithms, vulnerability to scraping blocks.
- Censorship & Global Accessibility Risks: political or technical downtime of central sites disables local scientific work.

Solutions

Decentralisation & Federation?

- Decentralization: an organizational and technical approach to separate system functions from central control.
- Federation: a union of independent, autonomous servers (nodes) communicating via a shared standardized protocol.
- DPI Goal: combine institutional node autonomy with cryptographic verification and sub-millisecond local search speed.

Existing implementations

Everyone uses email and the choice of email service is irrelevant, because communication happens via the SMTP protocol.

Existing implementations







Fediverse services use ActivityPub to interact with each other.

For example: Mastodon and Pleroma as alternative to Twitter, Pixelfed - Instagram, Bookwyrm - Goodreads

Existing implementations







Goodreads has not API, BookWyrm has JSON/ActivityPub API to support federation. Those platforms’re about social reading and reviewing.

OpenLibrary has Open APIs, but lack federated peer-to-peer synchronization and cryptographic protection against replay attacks.

Existing implementations



Scopus



ORCID

Google Scholar

ORCID: Open Source & Open Data and centralized.

Scopus: proprietary and centralized.

Google Scholar: proprietary and centralized.

OpenAlex: Open Source and centralized.

Existing implementations



Scopus



ORCID

Google Scholar

Only OpenAlex, ORCID and Scopus has official APIs.

but lack federated peer-to-peer synchronization and cryptographic protection against replay attacks.

DECENTRALISED ELECTRONIC PUBLICATIONS CATALOGING INFORMATION SYSTEM

The system has been developed to solve the identified problems.

- This platform transitions from a simple federation to a secure Zero-Trust environment, multi-source environment;

- JSON API and cryptographic synchronization for decentralisation;

- Designed for both academic research and universal bibliographic management.

Decentralised electronic publications cataloging information system







The system was built using:

- Elixir - a functional programming language for developing scalable, fault-tolerant, and distributed applications;

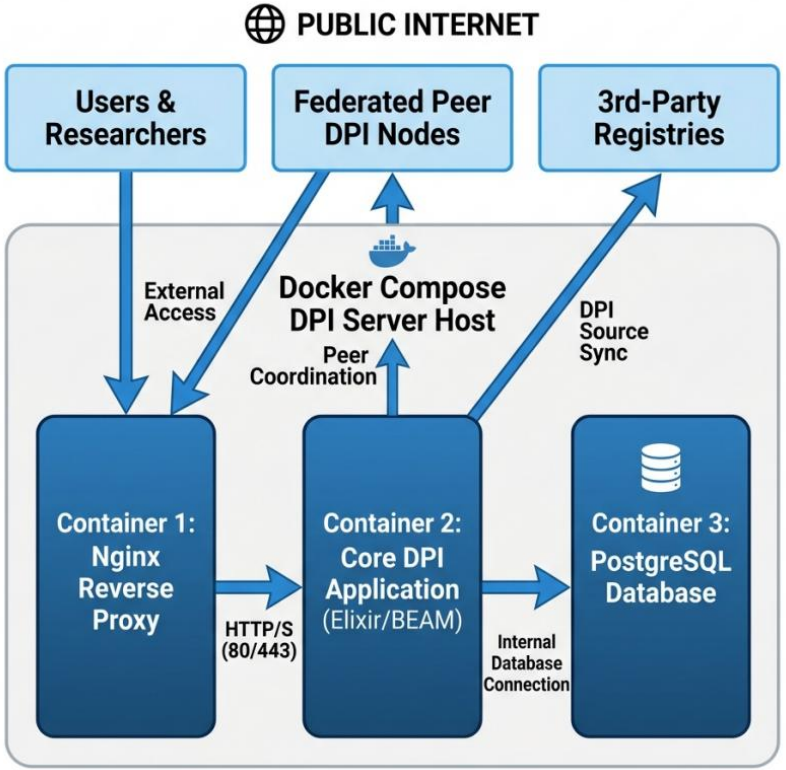
- Ash - a declarative framework that follows the philosophy of "model your domain, derive the rest.";

- Phoenix - a web framework for building interactive web applications;

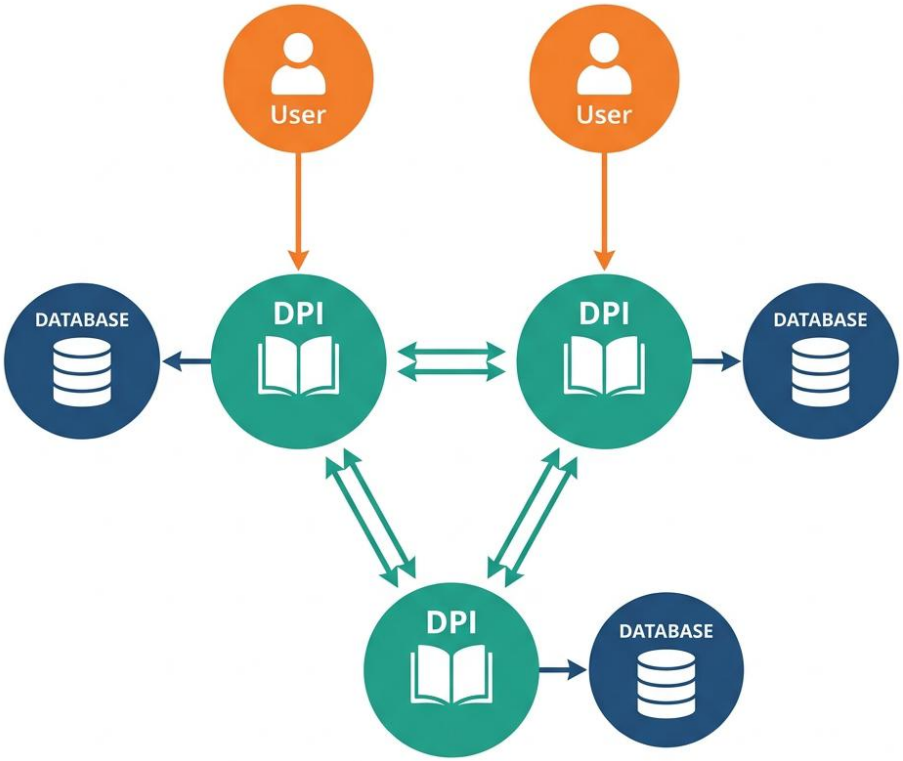
- Postgres - an object-relational database management system;

- Oban - a background job system packed with enterprise grade features, real-time monitoring with Oban Web, and complex workflow management with Oban Pro.

System architecture



Example of interaction



Asymmetric Ed25519 Signatures:

- Each server generates its own keypair;
- Public key is registered with partner nodes during initial trust setup;
- Every inter-node HTTP request is signed with the sender's private key;
- Ensures cryptographic non-repudiation and metadata integrity.

Signature Verification & Anti-Replay Protection

Signature created by sender:

- Compute SHA-512 hash over: Server ID + Request Timestamp + Payload;

- Sign the hash using sender's private key sk_node (Ed25519 algorithm);

Signature Verification & Anti-Replay Protection

Verified by receiver:

- Anti-Replay Protection (Time Window):
  - Time difference between current time and request timestamp must be less than 5 minute.
  - Requests outside the 5-minute window or with duplicate timestamps are immediately rejected.
- Cryptographic Verification:
  - Receiver validates the signature using sender's public key
  - Guarantees metadata integrity and non-repudiable origin authenticity.

Ingestion Pipeline & Metadata Harvesting

- Integration with ORCID API v3.0 (automated harvesting of researcher profiles and publication lists).
- Importers for CSV and JSON bulk file arrays.

Safe Moderation Model: Change Proposals

- Submissions from users do not mutate the main database directly.
- Edits are stored as pending JSON deltas in the pending state.
- Moderators inspect a dedicated Diff UI to approve (approved) or reject (rejected) proposals.

Safe Moderation Model: Change Proposals

- Guarantees absolute data integrity of canonical publication records.
- For compromised node, moderator can triger “unplug” to isolate external node.

Advantages

Advantages

- Simplicity: clean standardized interfaces (reactive WebUI and OpenAPI 3.0 JSON API).
- Decentralization: no Single Point of Failure (SPOF) and faster publishing and updates on local nodes without waiting for centralized moderation review.
- Data Integrity: cryptographic origin verification (Ed25519) and replay attack protection (Anti-Replay).
- Reactivity & SEO: initial static HTML for search engine indexing + instant WebSocket reactivity via Phoenix LiveView.
- Efficiency: optimized resource utilization powered by the Erlang BEAM virtual machine.

Limitations

- Initial Key Exchange: requirement for initial out-of-band public key exchange between partner node administrators (Honor-based trust).
- Local Storage Requirements: disk space requirements for full-text search indexes as publication volumes grow.

Potential improvements

Potential improvements

- Large-scale load testing in simulated multi-node clusters containing hundreds of autonomous nodes;
- Integration with existing Fediverse services;
- Improving data harvesting for sources without APIs.

Conclusions

Conclusions



Simplicity

Simple interfaces for interaction with the application: reactive WebUI and JSON API



Decentralisation

Zero-Trust architecture ensuring node autonomy and instant local data updates without central moderation bottlenecks



Data Integrity

Cryptographic Ed25519 signatures and Anti-Replay timestamp windows ensuring metadata authenticity



Integration

A universal data interface (Harvesting API and JSON API) increases the system's flexibility in acquiring information and allows applications to integrate metadata easily



Efficiency

Erlang BEAM lightweight process parallelism combined with sub-millisecond (< 1 ms) local PostgreSQL search speed



Safety & Moderation

Non-destructive ChangeProposals (JSON deltas) and an emergency Unplug Kill-Switch for rapid threat isolation.



thanks!

Any questions?

You can find me at

KKlochko@protonmail.com

Sources

1. Ash Framework. (n.d.). Ash HQ. https://ash-hq.org/

2. Phoenix Framework. (n.d.). Phoenix Framework. https://www.phoenixframework.org/

3. Tate, B., & DeBenedetto, S. (2025). Programming Phoenix LiveView: Interactive Elixir web programming without writing any JavaScript. The Pragmatic Bookshelf.

4. The Elixir programming language. (n.d.). The Elixir programming language. https://elixir-lang.org/

5. PostgreSQL. (n.d.). PostgreSQL. https://www.postgresql.org/

6. Docker: Accelerated container application development. (n.d.). Docker. https://www.docker.com/

7. Nginx. (n.d.). nginx. https://nginx.org/en/

8. Robust job processing. (n.d.). Oban Pro. https://oban.pro/

9. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. (n.d.). Tailwind CSS. https://tailwindcss.com/

10. Flowbite. (n.d.). Flowbite. https://flowbite.com/

Sources

11. GitHub - bookwyrm-social/bookwyrm: Social reading and reviewing, decentralized with ActivityPub. (n.d.). GitHub. https://github.com/bookwyrm-social/bookwyrm

12. GitHub - internetarchive/openlibrary: One webpage for every book ever published! (n.d.). GitHub. https://github.com/internetarchive/openlibrary

13. Priem, J., Piwowar, H., & Orr, R. (2022). OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. arXiv preprint arXiv:2205.01833. https://doi.org/10.48550/arXiv.2205.01833

14. Walk, P., & Van de Sompel, H. (2021). Decoupling scholarly communication functions with Linked Data Notifications. Journal of Documentation, 77(3), 641–660. https://doi.org/10.1108/JD-11-2018-0186

15. Fietkau, J., & Meyer, M. (2025). Privacy customization in a social sharing tool: Where academic publications meet social platforms. Proceedings of Mensch und Computer 2025 Workshop (MCI). https://doi.org/10.18420/muc2025-mci-ws04-301

16. Hendricks, G., Tkaczyk, D., Lin, J., & Feeney, P. (2020). Crossref: The sustainable source of community-owned scholarly metadata. Quantitative Science Studies, 1(1), 434–444. https://doi.org/10.1162/qss_a_00022

17. Fenner, M., Aryani, A., Bagnacani, A., et al. (2021). Connected research: The potential of the PID Graph. Data Science Journal. https://doi.org/10.5438/jwvf-8a66

Sources

18. Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., & Bolikowski, Ł. (2015). CERMINE: automatic extraction of structured metadata from scientific literature. International Journal on Digital Libraries, 16(3-4), 247–266. https://doi.org/10.1007/s10032-015-0249-8

19. Zignani, M., Gaito, S., & Rossi, G. P. (2024). An evidence-based and critical analysis of the Fediverse decentralization promises. arXiv preprint arXiv:2408.15383. https://doi.org/10.48550/arXiv.2408.15383

20. Josefsson, S., & Liusvaara, I. (2017). Edwards-Curve Digital Signature Algorithm (EdDSA). IETF Request for Comments: 8032 (RFC 8032). https://www.rfc-editor.org/info/rfc8032

21. Sporny, M., Cavage, M., & Richer, J. (2023). HTTP Message Signatures. IETF Request for Comments: 9421 (RFC 9421). https://www.rfc-editor.org/info/rfc9421

22. Tellvik, D. (2023). Metadata decentralized: An examination of folksonomy in book publishing [Master's thesis, Portland State University]. PDXScholar. https://pdxscholar.library.pdx.edu/eng_bookpubpaper/77/

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash

Slides Carnival

Free templates for all your presentation needs

For PowerPoint and Google Slides

100% free for personal or commercial use

Ready to use, professional and customizable

Blow your audience away with attractive visuals