Application 2026-02-336 Mycelica: Edge-Based Semantic Graph Infrastructure received

The following submission was recorded by NLnet. Thanks for your application, we look forward to learning more about your proposed project.

Contact

name ▶	Ekke Kaha
phone ▶	+37255581699
email ▶	ekkekaha@gmail.com
organisation name ▶	
country ▶	Estonia
consent ▶	You may keep my data on record

Project

code ▶	2026-02-336
project name ▶	Mycelica: Edge-Based Semantic Graph Infrastructure
fund ▶	Commons_Fund
requested amount ▶	€
website ▶	• https://github.com/Ekats/Mycelica/tree/master/docs

synopsis

Mycelica builds navigable hierarchies from document corpora without manual categorization.

Import documents (research papers via OpenAIRE, conversations, markdown, code). Generate local embeddings. Compute semantic similarity edges. Build hierarchy from edge topology using dendrogram extraction with adaptive threshold cuts. AI used only for naming categories.

Current state: Working prototype tested on 53,000 research papers. GUI (Tauri desktop), CLI, TUI interfaces. Linux and Windows builds. AGPL-3.0 licensed.

Proposed work:
- Edge type classification (supports, contradicts, extends, replicates)
- Citation graph overlay from OpenAIRE metadata
- Full-text PDF extraction for richer embeddings
- Incremental hierarchy updates without full rebuild
- Cross-database federated queries

experience

No formal involvement with similar organizations. Independent developer with shipped projects: Tartubus (public transit PWA,

React), Breax (Unity game, Google Play).
Mycelica is my first open source infrastructure project.

usage

Solo developer, Estonia-based. No other funding sources.

Development: €20,500 at €30/hour (Estonian cost-recovery rate)

Work areas (selection based on development progress):
- Browser extension (Holerabbit: web visit tracking, session detection, graph integration)
- Code parser improvements (language coverage, call graph accuracy, cross-file references)
- Edge type classification (supports, contradicts, extends, replicates)
- Citation graph integration (OpenAIRE citation metadata as edges)
- Incremental processing improvements (partial hierarchy updates)
- Cross-database federated queries
- Reference manager import (Zotero, Mendeley libraries)

Hardware: €2,500
- Local LLM workstation (GPU with 24GB+ VRAM, 64GB RAM, NVMe storage)
- Reduces API dependency, enables offline

development

API/infrastructure: €1,500
- LLM APIs for comparison testing and fallback
- CI/CD and demo hosting

Travel: €500
- FOSDEM or NGI community event

Total: €25,000
Timeline: 9-12 months part-time

comparison

Existing knowledge graph tools require manual structure:
Obsidian, Roam, Logseq, and Notion all depend on users creating folders, tags, and links by hand. Their graph views visualize connections you already made. They don't discover structure. If you import 50,000 documents, you get 50,000 documents in a flat list until you manually organize them.

Research discovery tools stay flat:
Connected Papers and Semantic Scholar visualize citation relationships but don't build navigable hierarchies. You search, get results, browse laterally. There's no "zoom out to see the field, zoom in to see subclusters" navigation. Inciteful and Litmaps have similar limitations: they show connections between

papers but don't organize them into browsable levels.

NLnet-funded semantic projects require explicit schema definition:

NextGraph uses RDF triples where users manually define subject-predicate-object relationships.

Atomic Data requires defining property schemas before data makes sense. Selva provides a graph database where developers explicitly set parent-child relationships via JavaScript API.

These are powerful tools for structured data, but structure must be imposed, not discovered.

What Mycelica does differently:

The design principle: organize external information the way associative memory organizes internal information. Not files in folders, but clusters of related things with weighted connections between them, where structure emerges from what actually relates to what.

Import a corpus. Mycelica generates embeddings locally, computes semantic similarity edges between documents, then builds a navigable hierarchy from edge topology. Categories form because their contents connect through edges. Documents without edges between them get separated into different branches regardless of superficial keyword overlap.

The algorithm: edge weights already encode hierarchical structure. Sort edges by

similarity descending, run Union-Find tracking merge events, extract hierarchy through adaptive threshold cuts validated for balance and cohesion. AI is used only for naming categories after structure exists.

Deterministic and auditable:
Same edges produce the same tree. Every structural decision traces back to measurable edge weights, not opaque AI interpretation. Researchers can inspect why documents clustered together (they share edges above threshold) and why clusters split (cohesion dropped below validation threshold). This reproducibility distinguishes Mycelica from tools where AI makes organizational decisions that can't be explained or replicated.

Practical result:
Import 53,000 research papers. Instead of a flat search interface or 53,000 files to manually sort, you get an explorable hierarchy: start at root, drill into "Neuroscience," then "Motor Control," then "Cortical Motor Systems," down to individual papers. Structure formed automatically from connectivity.

challenges

Edge type classification for research content is the hardest problem. Code edges are structural: function A calls function B,

detectable from syntax. Research relationships are semantic: paper A supports paper B's conclusions, contradicts them, extends the methodology, or replicates findings. Current semantic edges only capture similarity, not relationship type. Detecting support vs contradiction requires understanding content beyond embedding distance. Possible approaches: classifier on paper pairs with known relationships, LLM classification with local inference, linguistic markers in abstracts ("consistent with," "in contrast to"), or citation context when available. Probably some combination.

Citation graph integration has subtlety. OpenAIRE provides citation metadata for "cites" edges alongside similarity edges. But these mean different things. High similarity without citation might indicate cross-disciplinary connection or missed literature. Citation without similarity might mean methodological borrowing. How to weight and combine these for hierarchy building isn't obvious.

Incremental hierarchy updates challenge the current architecture. Adding documents means rebuilding the entire hierarchy. Inserting documents might shift optimal cut thresholds, reorganizing unrelated branches. Approaches include local re-optimization, lazy updates, or append-only staging. Each has tradeoffs I

haven't fully explored.
Scale is uncertain. Works at 53,000 papers. What happens at 500,000? Cohesion validation checks pairs of groups, which gets expensive.

Code parser expansion. Rust, TypeScript, Python, and C work. Adding more languages is straightforward but each has idioms. Bigger challenge: richer doc↔code linking. Currently docs link to code via backtick references. Could detect when documentation discusses a function conceptually without naming it, or link code comments back to architectural docs they implement.

ecosystem

Primary users:
Researchers working across disciplines who need to see connections between papers that don't cite each other. Developers navigating unfamiliar codebases via call graphs and doc-to-code links. People whose knowledge is scattered across conversations, notes, and bookmarks with no way to see how it connects.
Deployment:
Local-first desktop application. Download, run, own your data. No server, no account. Cross-platform builds exist (Linux AppImage, Windows installer).
Engagement:
Code is AGPL-3.0 on GitHub with documentation

```
of algorithms and architecture. Rust and Tauri
communities are natural first audiences since
the project is built with those tools.
Integration with existing tools: Zotero/
Mendeley import for existing collections,
OpenAIRE for research papers, RDF export for
semantic web tooling. Fits into workflows
people already have.
Will attend FOSDEM or NGI events if funded.
```

pgp

attachments

- mycelica.mp4
- mycelica.png

Check

Please check that the above contact details are correct and that any attachments you have included have been uploaded. If you are in doubt, and near a deadline, don't hesitate to resubmit - better safe than sorry. If you want to make changes to the proposal, do the same.

If you experience any technical problems, please contact the webmaster.

I checked the box but did not receive an email

Besides the obvious candidate for undelivered email (check your spam folder if you have it), some people run into their

own outdated email configuration. Do you use a legacy forwarding mechanism for your mail, from me@example.com to theactualmailbox@another.example.org? In that case, the final mailserver may toss these out due the use of modern anti-spoofing techniques (notably DMARC, DKIM and SPF) at our side. Essentially, forwarding the original email as was done historically means that you can't satisfy the origin and integrity conditions - and thus our email to you will be discarded...

The structural solution is to do the forwarding with a mechanism like *Sender Rewriting Scheme*. Ask your service provider, or consult the documentation of your software how to do that.