Seeking Digital Sovereignty

Integrating Nostr and Decentralized Mesh Networks for Sovereign, Censorship-Resistant Communication

Matthew Remmel

Executive Summary

This project proposes to evaluate the viability of integrating the Nostr protocol with mesh networking technologies, focusing primarily on Reticulum, to advance their shared mission of enabling sovereign, censorship-resistant, and decentralized communication—even in fully off-grid scenarios. Nostr's open, relay-based design resists centralized control and fosters an ecosystem of diverse clients and relays, while Reticulum delivers resilient, infrastructure-free networking across varied physical transports.

The research will prototype and demonstrate proof-of-concept integrations and bridges enabling Nostr relays and clients to operate over mesh links. It will systematically evaluate usability by quantifying the bandwidth and latency constraints required to deliver an acceptable user experience in real-world conditions. Additionally, it will analyze the roles and responsibilities of different user types—including mobile users, desktop clients, and relay operators—to identify optional and necessary configuration steps or operational trade-offs.

Objectives will include empirically validated minimum network requirements, clear design guidelines for developers, and open-source reference implementations. These deliverables will provide concrete, actionable tools for enabling the Nostr ecosystem to remain robust and accessible even in environments affected by poor infrastructure, connectivity restrictions, infrastructure-level censorship, or deliberate shutdowns.

Keywords Nostr, Mesh, Sovereign, Decentralized, Censorship Resistant, Networking

Grant Proposal

Grant Call
OpenSats Nostr Fund, July 2025 (opensats.org/funds/nostr)

Funding Requested
$150,000 USD

Acknowledgments
- @markqvist (unsigned.io)
- Reticulum Project (reticulum.network)
- Nostrastic Project (github/nostrastic)

Contact
email: matt@remmel.dev
nostr: primal.net/p/npub[..]

1. Problem Statement

Access to uncensored, resilient communication is a foundational requirement for digital sovereignty, civil liberties, and community resilience—particularly in contexts where accessing centralized or government-operated infrastructure is unsafe, unreliable, or subject to deliberate disruption. While the internet has enabled unprecedented levels of global communication, its dominant modes of connectivity remain vulnerable to centralized control, surveillance, and censorship. Natural disasters, infrastructure failures, and authoritarian interventions can sever communities from vital information flows exactly when they need them most.

The Nostr protocol seeks to address these problems at the application layer. By relying on a decentralized, relay-based design, Nostr allows anyone to operate relays and clients, avoiding single points of failure and censorship. However, even the most robust

application-layer design cannot fully mitigate scenarios where physical connectivity to the wider internet is unavailable, unreliable, or adversarial.

Mesh networking technologies, such as Reticulum, directly address this challenge by enabling resilient, infrastructure-free networking across diverse physical mediums—including radio, Wi-Fi, and serial links. By creating ad-hoc, peer-to-peer links that do not depend on existing infrastructure, mesh networks can maintain connectivity within and between communities even in off-grid, or censored environments.

Furthermore, Reticulum is designed from the ground up with strong privacy and cryptographic guarantees, making it especially well-suited for the problems at hand. Unlike many ad-hoc networking solutions that only offer raw connectivity, Reticulum includes private cryptographic identities, message authentication, and end-to-end encryption as first-class features. This ensures that even over untrusted physical links and network nodes, participants can verify the authenticity of messages and maintain anonymity and confidentiality where required.

This proposal seeks to bridge these two complementary technologies to create a truly sovereign communication platform that remains functional and resistant to censorship, even in fully disconnected or adversarial conditions. Through prototyping, systematic evaluation, and open-source development, this project will deliver the empirical data, technical integrations, and clear design guidelines necessary to make Nostr mesh integration a practical and accessible option for developers, relay operators, and end users alike.

2. Goals and Objectives

This project has five primary objectives designed to demonstrate, evaluate, and enable the practical integration of Nostr with decentralized mesh networks. Together, these goals aim to deliver robust, actionable, open-source resources to strengthen sovereign communication.

2.a. Proof of Concept Integrations and Bridges

We will design and implement working proof-of-concept bridges that enable Nostr relays and clients to communicate over Reticulum-based mesh networks. These integrations will demonstrate the viability of operating standard Nostr clients and relays in fully off-grid contexts by routing messages through Reticulum links. The prototypes will focus on minimal viable bridging, with clear separation of concerns to facilitate maintenance and extension by other developers. Deliverables will include well-documented open-source code and example deployments to validate end-to-end messaging over mixed internet and mesh topologies.

2.b. Systematic Evaluation of Network Requirements in Lab Setting

We will conduct controlled laboratory testing to quantify the technical requirements for acceptable Nostr performance over mesh links. This includes benchmarking bandwidth consumption, message latency, and relay-client handshake behavior under constrained conditions. The goal is to identify empirically supported minimum network parameters and configuration options, helping developers and operators plan deployments with realistic expectations. The lab environment will allow repeatable experiments with

varying topologies, link qualities, and load profiles to provide a robust foundation for design recommendations.

2.c. Systematic Evaluation of Network Requirements in Real World Setting

Building on lab findings, the project will test the prototype bridges and Nostr clients in realistic field conditions, such as rural or urban mesh deployments, radio-based links, and mixed-infrastructure scenarios. This evaluation will uncover practical challenges not evident in lab testing, such as signal degradation, node churn, and unpredictable link availability. By systematically recording these factors and their impact on user experience, the project will provide actionable guidance for planning and risk assessment in practical, real-world settings.

2.d. Design and Development Guidelines

We will produce findings from prototyping and evaluation into clear, accessible guidelines for developers and operators. These guidelines will cover recommended configurations, minimum network requirements, and integration patterns for combining Nostr and Reticulum in constrained environments. The goal is to lower the barrier to adoption by providing practical, well-tested advice that enables other projects to build on this work with confidence. All documentation will be open and freely licensed to maximize impact across the ecosystem.

2.e. Reference Quality Open Source Implementation

All prototypes, bridges, tools, and any other miscellaneous code developed under this grant will be released as open-source software under permissive licensing. The project will prioritize maintainable, modular design to encourage reuse and contribution by the broader community. In addition to code, the project will deliver example deployments and comprehensive documentation to serve as reference implementations for future integration efforts. By delivering high-quality, usable tooling and information, the project aims to accelerate adoption of sovereign, robust communication solutions worldwide.

3. Project Outline

3.a. Project Planning and Setup

- Define detailed scope, objectives, and deliverables with timelines
- Develop project schedule with milestones and review points
- Set up Github repositories, and establish project structure and documentation (README, architecture, testing plans, guidelines)
- Define communication channels and plan for interested parties, collaborators and stakeholders
- Coordinate with collaborators and subject-matter experts
- Set up issue tracking and project management tools, along with initial ticket items

3.b. Bridge/Integration Prototyping

- Design three phase minimal-viable integration architecture and bridging approach with clear interfaces and separation of concerns for maintainability
- Phase One: Implement client and server Reticulum proxies to allow existing Nostr clients and relays to communicate over a mesh network (Figure 1)

- Phase Two: Extend an existing Nostr relay to support Reticulum as a first-class interface (Figure 2)
- Phase Three: Modify an existing Nostr client to support Reticulum as a first-class interface (Figure 3)
- Ensure compatibility with standard Nostr client/relay behavior
- Write developer-level documentation for installation and usage
- Create example configurations for different deployment scenarios

3.b.i. *Phase One:*

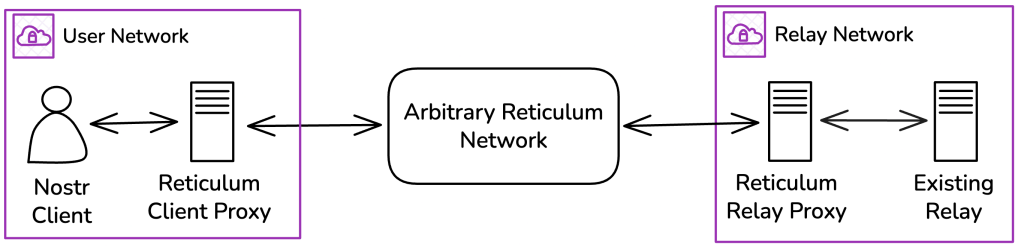

Figure 1: Communication using client and relay Reticulum proxies

In Phase One, we will implement client and server Reticulum proxies to bridge existing Nostr clients and relays over a mesh network. This approach preserves full compatibility with the current Nostr ecosystem while enabling communication across the Reticulum network via intermediary proxies. Figure 1 illustrates how the client and relay proxies interact to route messages seamlessly between traditional Nostr infrastructure and the Reticulum mesh.

3.b.ii. *Phase Two:*

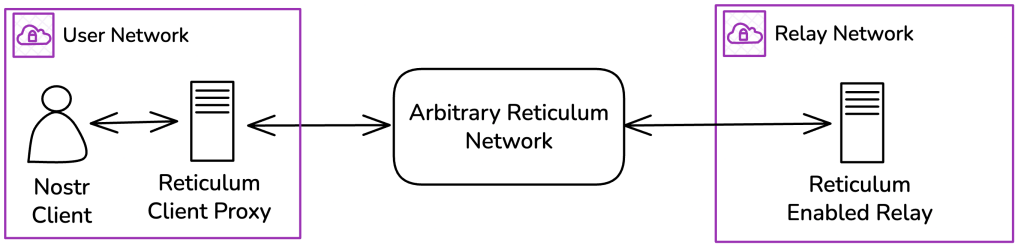

Figure 2: Communication using client proxy to Reticulum enabled relay

Phase Two extends this integration by modifying an existing Nostr relay to support Reticulum as a first-class interface. Instead of relying on a separate relay proxy, the relay itself will natively communicate over Reticulum, reducing complexity and latency. As shown in Figure 2, the client proxy can now connect directly to a Reticulum-enabled relay over the mesh network. A first proof-of-concept implementation can be built on github/monstr, since it is written in Python, like the Reticulum reference SDK, and contains a basic relay implementation that is designed to be extensible. We would like the final reference implementation to be built on an SDK that supports multiple languages, to enable more widespread integration with relays built in non-python languages.

3.b.iii. *Phase Three:*

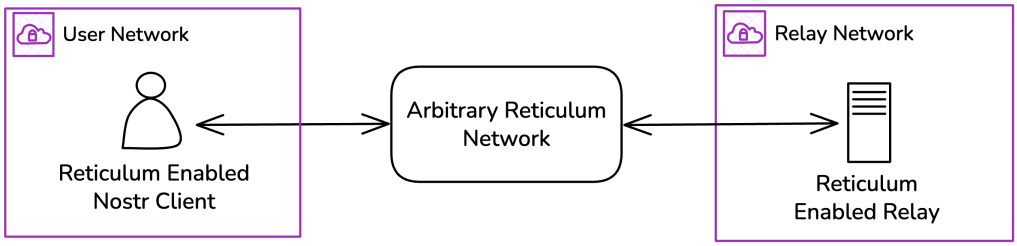

Figure 3: Communication using Reticulum enabled client and relay

In Phase Three, we will modify an existing Nostr client to support Reticulum as a first-class interface, removing the need for a separate client proxy. This completes the integration by enabling end-to-end Reticulum-native communication between clients and relays. Figure 3 demonstrates the simplified architecture, with both client and relay fully Reticulum-enabled and directly leveraging the mesh network. As with phase two, a first proof-of-concept implementation can be built on github/monstr, since it also has pre-made Nostr client classes. We would like the final reference implementation fo this to also be built on an SDK that supports multiple languages.

3.c. Laboratory Evaluation

- Define lab test plan, goals, and success criteria
- Set up controlled lab environment with simulated topologies and constraints
- Emulate varying link qualities (e.g., latency, bandwidth limits)
- Benchmark bandwidth consumption under typical Nostr usage patterns
- Benchmark bandwidth consumption under constrained Nostr usage patterns, such as direct-message-only and publish-only scenarios
- Measure message latency and delivery reliability across mesh links
- Evaluate relay handshake and client-subscription behavior under constrained conditions
- Record all experimental configurations and results systematically
- Analyze and document minimum network requirements under lab conditions for acceptable user experience

3.d. Field Testing and Real-World Evaluation

- Identify suitable test locations and use-cases (urban mesh, rural deployments, radio links)
- Recruit and coordinate with local operators and volunteers for field trials
- Deploy prototype bridges in real-world environments
- Measure real-world performance metrics: latency, bandwidth, node churn, availability
- Observe operational challenges such as interference, power constraints, and node mobility
- Document real-world deployment procedures and best practices
- Analyze results to identify practical limitations and mitigation strategies

3.e. Design Guidelines and Documentation

- Summarize key findings from lab and field evaluations

- Define recommended configurations for mesh-integrated Nostr relays and clients
- Provide clear integration patterns for combining Nostr and Reticulum
- Describe deployment planning considerations (e.g., node placement, power needs)
- Offer troubleshooting guidance for common failure modes
- Prepare accessible, well-organized documentation suitable for developers and operators
- Ensure all documentation is open and freely licensed

3.f. Reference Implementation Release

- Refactor prototype code for clarity, modularity, and maintainability
- Conduct code reviews and testing to ensure stability
- Package and publish all software under permissive open-source license
- Provide example deployments, including configuration files and instructions
- Write comprehensive user and developer documentation
- Create contribution guidelines to encourage community involvement
- Set up ongoing issue tracking and versioning

3.g. Dissemination and Community Engagement

- Announce results and releases in relevant communities (Nostr, Reticulum, mesh networking)
- Share findings on project website, mailing lists, social media, and forums
- Provide support channels for Q&A and troubleshooting
- Organize or participate in discussions, presentations, or workshops, if desired
- Solicit feedback for future improvements and refinements
- Foster ongoing collaboration and contributions from the broader community

4. Evaluation

The success of this project will be evaluated based on whether it delivers practical, validated tools and guidance for operating Nostr over Reticulum mesh networks in constrained environments. Evaluation will focus on:

- Verification that the prototype bridges enable functional Nostr messaging over Reticulum links in both lab and field settings
- Measurement of key network performance metrics (e.g., bandwidth usage, latency) against thresholds for acceptable user experience
- Identification and documentation of minimum viable network requirements for reliable operation
- Assessment of real-world deployability through field testing, capturing operational challenges and mitigation strategies
- Review of the clarity, completeness, and usability of all published design guidelines and documentation
- Validation of the codebase for quality, modularity, and ease of integration by third-party developers
- Openness and accessibility of deliverables, ensuring that all code, data, and documentation are published under permissive licenses and are genuinely usable by the broader community

Results will be shared through regular reports and documentation, with all experimental data, analysis, and software made freely available to support transparency, reproducibility, and adoption.

5. Project Milestones

Below is a rough outline of project milestones, designed to be conservative enough to remain achievable even in the face of unexpected challenges.

5.a. Month 1

- Define detailed scope, deliverables, and schedule
- Set up repository, documentation structure, and issue tracking
- Set up communication channels and guidelines
- Coordinate with any collaborators and interested parties
- Design and begin work on minimal-viable integration architecture for bridges

5.b. Months 2–3

- Finish development of minimally viable bridge with basic message routing
- Build necessary tooling to emulate simulated link qualities
- Develop tooling to measure bandwidth consumption, latency metrics, and delivery reliability
- Build deterministic testing tools for systematic and repeatable communication tests
- Define lab testing plan and success criteria

5.c. Months 3–6

- Complete proof-of-concept bridge with full support for standard Nostr relay/client behavior
- Write developer-level documentation and example configurations
- Set up controlled lab environment with simulated constraints
- Conduct lab testing to measure bandwidth, latency, and reliability
- Analyze results to identify minimum viable network requirements

5.d. Months 6–9

- Prepare for field testing: identify locations and coordinate participants
- Acquire and configure necessary hardware to run various network topologies that include multiple physical Reticulum transports (radio, Wi-Fi, Ethernet, etc.)
- Deploy nodes and prototype bridges in real-world environments (urban, rural, radio links)
- Measure real-world performance metrics and operational challenges
- Document deployment procedures, best practices, and mitigation strategies
- Refine the prototype based on field findings, and repeat tests as necessary

5.e. Months 9–12

- Summarize lab and field results into clear design guidelines and recommendations
- Refactor and finalize code for open-source release
- Publish all code, example deployments, and comprehensive documentation
- Announce results to Nostr, Reticulum, and open-source communities

- Provide support channels, solicit feedback, and encourage ongoing contributions

6. Qualifications

This project will be led by myself, Matthew Remmel. I have over a decade of experience delivering robust, secure, and scalable software systems. My background combines deep professional expertise in distributed systems with a strong interest in robust, decentralized software.

I hold a Master of Science in Computer Science with a specialization in Information Security, and I have extensive experience architecting and implementing high-throughput, event-driven systems in production environments. I have led engineering efforts in domains requiring stringent security and compliance (SOC 2, PCI DSS, ISO 27001), demonstrating the ability to deliver auditable, maintainable, and reliable software under demanding requirements.

Beyond my professional work, I have a long-standing personal interest in hardware and embedded programming. This hands-on familiarity with low-level systems, radios, and constrained devices complements my software background, making me well-suited to bridge the gap between application-layer protocols (like Nostr) and physical mesh networking technologies (like Reticulum).

Key relevant qualifications include:

- Expertise in secure, event-driven, distributed systems and protocols
- Professional experience building and operating highly available infrastructure and applications
- Strong information security background, with experience translating risk assessments into actionable design and policy
- Fluency in a wide range of programming languages and paradigms, supporting the design of modular, maintainable integrations
- Practical familiarity with mesh networking principles and the challenges of constrained or offline environments

By combining formal security expertise, large-scale distributed systems experience, and hands-on enthusiasm for hardware and embedded development, I am prepared to deliver the technically sound, usable, and open-source tools this project proposes.