Shurli: Distributed Social Proof for P2P Trust Without Blockchain

Satinderjit Singh
with Claude, Anthropic
shurli.io
v0.3.1 – 12 April 2026

Abstract. A purely peer-to-peer network layer for AI-native applications would allow agents and humans to connect, communicate, and transfer data directly without routing through centralized infrastructure. Cryptographic identity provides part of the solution, but the main benefits are lost if a trusted third party is still required to verify peer reliability and authorize access. We propose a trust framework called distributed social proof (DSP), in which reputation emerges from overlapping behavioral observation by independent agents on the network, requiring no ledger, no tokens, and no consensus algorithms. The network derives trust scores by adapting the bridging algorithm from X's Community Notes [1] to peer reputation: matrix factorization applied to interaction ratings extracts genuine peer quality from cross-factional agreement rather than majority vote, without requiring a central operator. Peers prove their reputation exceeds a threshold using zero-knowledge proofs [2] without revealing the score itself, enabling privacy-preserving access control. Authorization uses capability tokens [3] that can only be made more restrictive as they are delegated, never more permissive, a mathematical guarantee from HMAC chains. As long as observers span sufficiently diverse network positions, coordinated manipulation of the reputation layer becomes proportionally more expensive. The network is designed to operate without humans in the loop: nodes discover peers, traverse NATs, manage trust, and maintain security autonomously. It is agnostic by design. Payment methods, naming systems, identity providers, and agent frameworks all plug in. The core provides only what is essential: transport, identity, discovery, authorization, and trust. We present Shurli, an open-source implementation on libp2p, as the reference architecture.

1 Introduction

Connectivity on the internet has come to rely almost exclusively on centralized services serving as intermediaries between devices. While the system works well enough for most use cases, it suffers from inherent weaknesses of the trust-based model. Every connection routes through infrastructure controlled by a third party who can revoke access, change terms, or shut down entirely. Carrier-grade NAT blocks direct connections for billions of devices, forcing reliance on relay services the user does not control. AI agents that need to communicate with each other must route through centralized cloud platforms, creating single points of failure and surveillance. Self-hosters cannot reach their own machines without signing up for an online service. The cost of this intermediation is not just financial. It is a loss of sovereignty: the network decides who connects to whom, not the participants themselves.

What is needed is a network and communication layer based on behavioral observation instead of institutional trust, allowing any two peers, human or AI, to connect directly without a centralized intermediary. A system where trust emerges from how peers actually behave on the

network, not from which provider vouches for them. In this paper, we propose distributed social proof (DSP): a trust framework maintained by independent agents that observe, evaluate, and report on peer behavior, requiring no ledger, no tokens, and no consensus algorithms. We present Shurli, an open-source implementation on libp2p, as the reference architecture for a **Zero-Human Network**: P2P infrastructure where nodes discover peers, traverse NATs, manage trust, and maintain security without a human in the loop. The system is agnostic by design. Payment methods, naming systems, identity providers, and agent frameworks all plug in. The core provides only what is essential: transport, identity, discovery, authorization, and trust. The network is secure as long as observers span sufficiently diverse network positions, making coordinated reputation manipulation proportionally more expensive than honest participation.

2 Distributed Social Proof

Trust in networked systems has historically taken two forms: computational proof (blockchain), which is physics-backed but rigid, energy-intensive, and binary; or institutional authority (CAs, DNS, cloud), which is flexible but centralized. Human societies operate on neither. They establish trust through overlapping observation, behavioral reputation, and collective judgment. No cryptographic proof backs a scientist's reputation. DSP applies this third model to P2P networking: trust maintained by independent AI agents that observe, evaluate, and report on peer behavior. No global ledger. No central authority. Trust is an emergent property of sufficient independent observation.

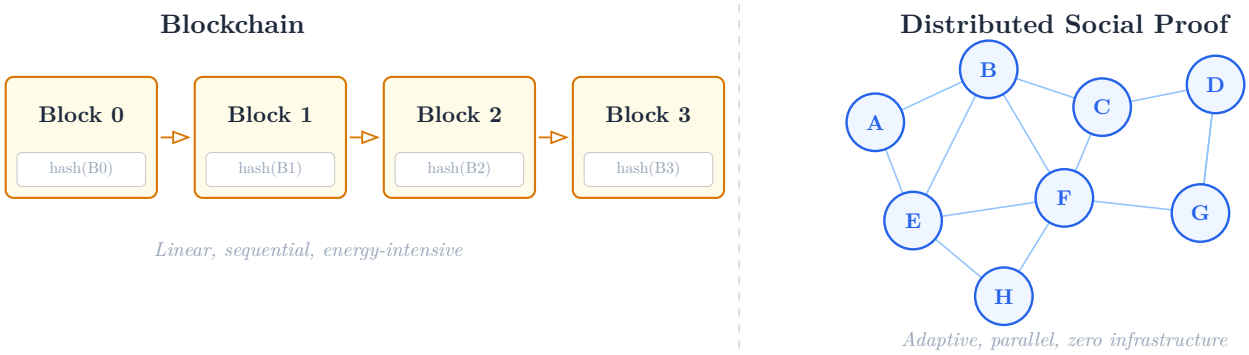

Figure 1: Structural comparison. Blockchain: linear chain of cryptographic proofs. DSP: mesh of independent agent observations.

2.1 Core Properties

Emergent trust. No single agent is authoritative. Trust scores emerge from convergence of independent observations, analogous to the Community Notes mechanism [4] where consensus must bridge diverse perspectives.

Fuzzy evaluation. Unlike blockchain's binary model, DSP supports graduated trust: a node may be unreliable for latency claims but adequate for relay; an agent may deliver excellent inference results but settle payments slowly. Multi-dimensional trust, not binary.

Zero infrastructure. No mining, staking, tokens, or persistent ledger. Agents observe peers during normal operation. Trust is a byproduct of participation.

Adaptive response. Reputation degrades and recovers dynamically. Misreporting nodes are deprioritized without governance votes, hard forks, or slashing events.

2.2 The Observation Model

DSP agents observe concrete, verifiable behaviors at two levels. These are not subjective ratings but measurable properties of peer interactions.

Network-level observations capture how reliably a peer participates in the infrastructure: latency accuracy (claimed versus measured), relay uptime and circuit completion rate, connection quality (packet loss, reconnection frequency), signaling honesty (bootstrap record freshness), transfer integrity (BLAKE3 hash match rate, protocol compliance), and bandwidth delivery (advertised versus actual throughput).

Agent-level observations capture how honestly a peer fulfills commitments in data and value exchange: did the AI agent complete the requested task, did it deliver the promised output, was the data exchanged intact and as described, was the value settlement honored. When a human asks their AI to accomplish something and that AI delegates subtasks to other agents across the network, every step in that chain produces observable evidence. An agent that consistently delivers correct inference results, completes file transfers without corruption, and settles payments as agreed builds reputation. An agent that fails to deliver, returns garbage, or disappears mid-task loses it. Both peers sign bilateral transfer receipts as cryptographic evidence of every interaction.

This is what makes DSP agent-native: the same reputation framework that tracks network reliability also tracks agent honesty. A single score reflects both "can I reach this peer" and "will this peer do what it promised." Each observation is independently verifiable. The model produces ratings automatically from real interactions, not from explicit voting.

The reputation API is exposed to all plugins. File transfer, naming resolution, and any future plugin can feed observations into the same reputation system and query scores from it. This makes reputation extensible beyond use cases the core designers anticipated. Connected external identities (social profiles, messaging accounts, enterprise directories) provide an additional trust dimension: identity trust inherited from established presence elsewhere, weighted separately from performance trust earned through actual behavior on the network.

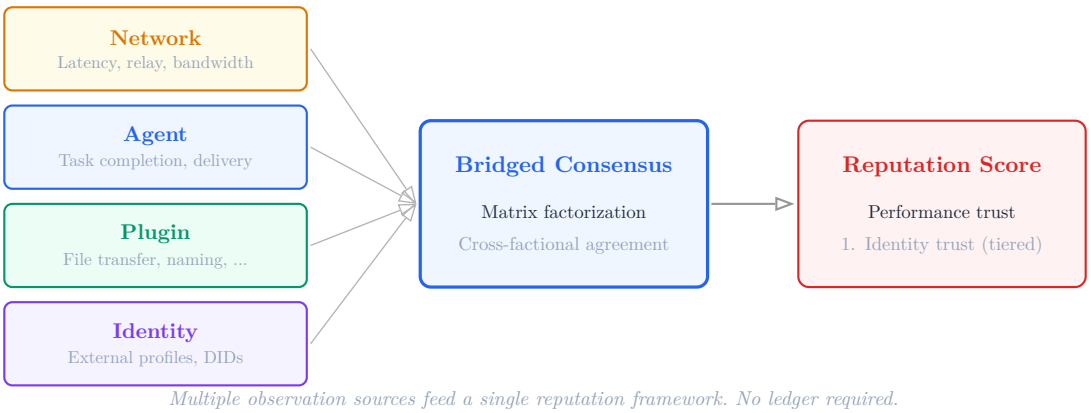

Figure 2: Network, agent, plugin, and identity observations feed into bridged consensus, producing a composite reputation score with separate performance and identity trust dimensions.

3 Sybil Resistance and Trust Convergence

The primary attack against DSP is Sybil attack [5]: deploying compromised agents to control the observation layer. We analyze conditions under which this becomes impractical.

3.1 Observer Diversity Requirement

Let N be total observer agents, f the adversary-controlled fraction, k the independent observations required for reputation update, and d the number of distinct network partitions observers must span. The probability of adversary control is bounded by:

$$P_{sybil}(k, d) \leq f^k \cdot f_{max}^{d-1}$$

where f_{max} is adversary penetration in any single partition. Even at 40% total control, achieving 40% in every independent partition requires proportionally more resources.

A "network partition" in this context does not refer to geographic region or IP subnet. It refers to an *independent observation group*: a set of peers whose rating behavior is statistically independent of other groups. The matrix factorization algorithm (Section 3.3) discovers these groups automatically from rating patterns via polarity factors. Two peers controlled by the same operator, regardless of how many distinct IP addresses or VPN endpoints they use, will rate other peers the same way and cluster into the same polarity factor. The variable d counts how many such *behaviorally independent* groups must be penetrated.

This is the key distinction from IP-based identity. Bitcoin's whitepaper [6] observes that "if the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs." DSP does not use IP-based identity at all. A single entity operating 10,000 nodes behind 10,000 different IP addresses still constitutes a single faction in the MF decomposition, because behavioral patterns, not network addresses, determine partition membership. The cost to the attacker is not "acquire many IPs" but "get your fake nodes rated well by nodes that disagree with each other on everything else," which requires either compromising genuinely independent peers or building real cross-factional reputation through honest participation.

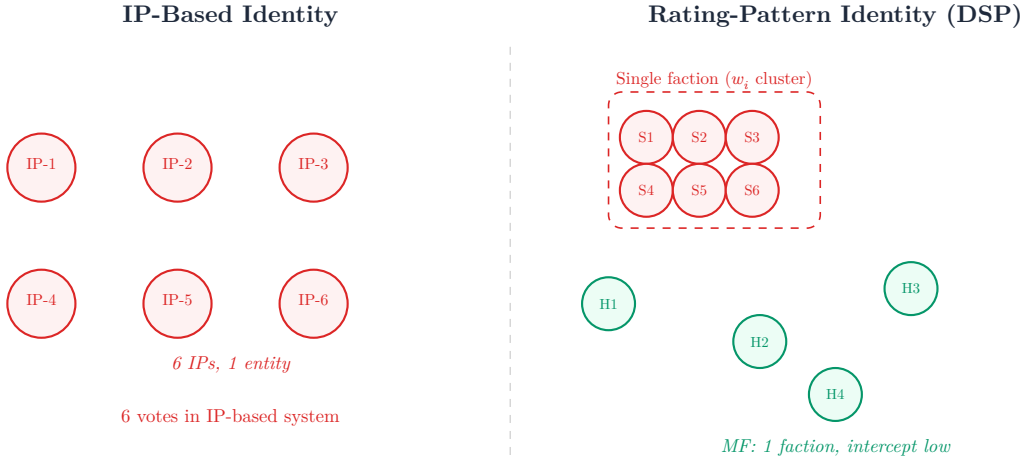

Figure 3: IP-based identity is defeated by address proliferation. Rating-pattern identity clusters Sybils into a single faction regardless of IP diversity, because polarity factors reflect behavioral correlation, not network topology.

3.2 Trust Convergence

For a node with true quality $q \in [0, 1]$ and n honest observations with noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, the estimated reputation converges:

$$E\big[\hat{R}\big] = q, \qquad \mathrm{Var}\big(\hat{R}\big) = \frac{\sigma^2}{n}$$

With 25 observations and $\sigma = 0.2$, the 95% confidence interval is ± 0.08, sufficient for routing decisions. More observers yield more reliable consensus without trusting any single one.

3.3 Bridged Consensus via Matrix Factorization

Raw observation averaging is vulnerable to factional manipulation. Shurli addresses this by adapting the Community Notes bridging algorithm [1] to peer reputation.

The core model decomposes each rating as:

$$\hat{y}_{ij} = w_i x_j + b_i + c_j$$

where w_i is rater i's polarity factor (where they sit on a viewpoint spectrum), x_j is peer j's polarity factor, b_i is rater i's baseline tendency, and c_j is peer j's intercept: the genuine quality signal. The intercept captures quality that cannot be explained by factional alignment. A peer rated well by observers who disagree on everything else gets a high c_j; a peer rated well only by a single cluster has its ratings absorbed into the polarity factors, leaving the intercept low. The algorithm simultaneously discovers viewpoint clusters and extracts quality that transcends factional bias. A peer rated well by diverse, disagreeing observers gets a high intercept. A peer rated well only by a single cluster gets a low intercept, regardless of how large that cluster is.

Creating 1000 fake accounts that all rate each other the same way is detected as a single faction. Only cross-factional agreement moves the intercept. No blockchain, no staking, no central authority. Pure mathematics on rating patterns.

3.4 Decentralization of Trust Computation

X's Community Notes [4] runs the identical matrix factorization algorithm on centralized infrastructure. X collects all ratings, runs MF on its servers, and publishes the results. X can additionally detect Sybil accounts through IP fingerprinting, browser telemetry, and manual intervention. The natural question is: what replaces X's backstop in a decentralized network?

The answer is that the MF algorithm's Sybil resistance does not depend on centralization. X uses fingerprinting as a supplementary defense, but the core Sybil detection, the polarity factor decomposition, operates entirely on rating patterns. The same mathematics works identically whether computed by a central server or by individual peers. What changes in a decentralized setting is *who runs the computation*, not *how the computation works*.

Shurli addresses trust computation through progressive decentralization in three layers:

Figure 4: Progressive decentralization of trust computation. The mathematical mechanism (MF) is identical at every layer; only the trust assumption changes.

Layer A is semi-centralized: the relay collects signed ratings, runs MF periodically, and publishes both the scores and the input rating matrix. Any peer can download the matrix, re-run MF with the same parameters, and verify that the published scores match within a small tolerance ε. A relay that fabricates scores must also fabricate a plausible rating matrix that produces those scores under MF, which is computationally harder than running MF honestly. MF reproducibility in practice requires a fixed initialization seed, deterministic iteration order, and single-threaded execution; even then, cross-platform floating-point differences and non-associative summation in parallel reductions introduce last-bit variation [1]. Verification therefore compares intercepts under a convergence tolerance rather than requiring exact equality, the same approach Community Notes uses for its own audit pipeline.

Layer B adds local verification. When multiple relays exist, each runs MF independently and publishes results. Peers compare scores across relays and flag divergence. Peers also run local MF on their own direct observations as a sanity check. A relay that selectively drops ratings before publishing the matrix is caught by peers whose local observations disagree with the published matrix.

Layer C eliminates the relay from trust computation entirely. Peers gossip interaction ratings directly to neighbors via pubsub. Each peer accumulates a partial view of the global rating matrix through epidemic propagation and runs MF locally. The relay becomes one participant among many, with no special authority over reputation. Layer C produces local trust views, not global consensus: two peers operating on different partial matrices may compute slightly different scores for the same target peer, and this is correct behavior for a decentralized system rather than a weakness. Every trust decision in Shurli is made by a specific observer for a specific purpose (routing, peering selection, access control, transfer acceptance), and every such decision has a specific observer whose local view is the relevant input. The network never queries "the global consensus score of peer X" as an atomic value, because no such operation exists in the protocol. Asymmetric local views match how existing decentralized systems (BGP routing, DNS resolution, distributed version control) operate in practice. Asymptotic convergence across peers happens naturally through gossip as the rating graph becomes denser, but is not a correctness requirement for any individual decision.

The five-layer Byzantine defense operates independently of which computation layer is active: (1) bilateral transfer receipts require both peers' signatures to validate a rating, (2) MF itself detects factional coordination, (3) relay observability cross-checks circuit metadata for relayed connections, (4) quasi-clique detection flags coordinated rating groups, and (5) rate limiting ensures each rating requires real network interaction. An attacker must defeat all five layers simultaneously, and critically, the core defense (layer 2: MF faction detection) requires no central operator at all.

3.5 Economic Basis of Sybil Resistance

Matrix factorization detects naive coordinated rating, but pure algorithmic cleverness does not defeat a determined attacker who adapts to the known scoring rules. The entire network is itself, in a sense, a "swarm" that accumulated influence over time. The algorithm alone is not the defense. The defense is what the algorithm, combined with real-world participation cost, makes the attacker actually do.

In proof-of-work systems, the cost is computational work whose output (mined tokens) is tradable. In centralized systems, the cost is identity verification (SMS, government ID, KYC) whose integrity depends on external authorities. Shurli uses neither. The cost in DSP is *real participation over time*: a peer must age at least 7 days before its ratings count toward others' scores, complete a minimum number of genuine interactions, operate under a probationary score cap for 30 days, and produce bilateral transfer receipts signed by counterparties for every interaction that feeds reputation.

The critical property of these costs is that honest users pay them as a byproduct of normal use. A participant who joined to share files, reach their own machines, or run an AI agent is already completing transfers, accumulating interactions, and aging their peer identity. They pay zero marginal cost for reputation accrual. An attacker faces the same requirements, but with an additional algorithmic constraint from Section 3.3: cross-factional agreement is the only path to high intercepts, and cross-factional agreement can only be obtained by genuinely providing value to peers outside the attacker's control. Faction-internal rating produces zero intercept regardless of how many fake nodes participate.

After the probationary period, an attacker who maintained fake nodes through real interactions, earned ratings from peers outside their cluster, and built cross-factional reputation has functionally become an honest participant, because the requirements for attack and honest participation converge. The defense is not "make attacks computationally expensive" but "make attacks indistinguishable from honest participation". This is sufficient for networking decisions (routing, peering, access control) where the cost of a wrong outcome is small and reversible. It is explicitly not sufficient for financial settlement, which requires thermodynamic irreversibility.

The convergence argument handles the *building* phase of an attack, not the *exit* phase. A node that participated honestly for six months, accumulated real cross-factional reputation, and then used that trust for a single high-value malicious action is not prevented by the mechanisms described above. Asymmetric rating weights (one data-corruption event costs 75 successful-transfer equivalents) and time decay cause reputation to collapse rapidly after the betrayal, but the betrayal itself is not blocked in the moment it occurs. This is an explicit scope choice: Shurli is designed for interactions where the per-event blast radius is bounded (a failed transfer, a bad routing decision, a dropped relay circuit), so that reputation degradation after the fact is a sufficient response. Interactions where a single defection has catastrophic consequences, such as irreversible financial settlement or legal commitment, require thermodynamic or contractual finality outside DSP's scope and should not be built on reputation alone.

A third defense layers on top of behavioral cost: tiered identity trust via the Shurli naming standard. Peers may optionally bind their identity to verifiable external presence, such as social profiles, messaging accounts, enterprise directory entries, DIDs [7], or domain names. These bindings feed a separate identity trust dimension that combines with behavioral reputation to produce the final score. An honest user links identities they already control at zero cost. An attacker running 10,000 Sybil nodes must forge 10,000 plausible external identities, each with

its own history, cross-platform presence, and verifiability. The marginal cost per Sybil grows with the number of identity tiers the target network weighs. Identity trust is optional and pseudonymous participation remains fully supported, but networks that require stronger trust decisions (enterprise agent-to-agent coordination, paid inference markets) can weight identity trust heavily, shifting the attack cost from "operate nodes honestly for 30 days" to "operate nodes honestly for 30 days *and* maintain real external identities that pass cross-platform verification."

Generative models are making plausible synthetic profiles progressively cheaper, which erodes the per-identity forgery cost over time. This is an arms race the identity-binding defense cannot win on its own. The design response is that identity trust is weighted, not binary: networks set the weight based on their threat model, deployments requiring high assurance can require multiple independent tiers (social plus enterprise plus government-issued), and crucially, behavioral defense (MF faction detection, bilateral receipts, cross-factional rating requirements) remains the primary layer and does not depend on identity at all. A synthetic profile cannot shortcut the requirement to complete real interactions with peers the attacker does not control. The paper's position is that identity trust is one asymmetric cost factor among several, degrading gracefully as synthetic identity generation improves, rather than a linchpin defense.

In authorized private networks, a fourth and strongest defense applies: invitation chains. Every peer traces membership through a chain of prior invitations rooted in trusted parties. Sybil creation requires either compromising an existing peer's identity or convincing a trusted peer to issue an invitation, neither of which has a computational shortcut. Private-mode networks inherit the social trust of their operators at zero ongoing cost to users.

3.6 Comparative Attack Cost

Property	**Blockchain (PoW)**	**Institutional**	**DSP (Shurli)**
Trust basis	Thermodynamics	Legal identity	Behavioral observation
Finality	Probabilistic (energy)	Contractual	Probabilistic (behavioral)
Expressiveness	Binary	Policy-defined	Graduated / fuzzy
Infrastructure	Very high	Moderate	Near zero
Throughput	Low (7 tx/s)	High	High
Censorship resist.	Very high	Low	High
Sybil resistance	High (energy)	High (identity)	High (diversity)
Adaptability	Very low	Moderate	Very high
Energy (annual)	~150 TWh	Moderate	Negligible
Quantum resistance	Vulnerable	Vulnerable	Hybrid PQ transport

Table 1: Comparative properties of trust frameworks.

4 Architecture

Shurli is an open-source Go implementation on libp2p [8]. It occupies a specific position in the AI infrastructure stack: the network and communication layer beneath agent applications, frameworks, and protocols.

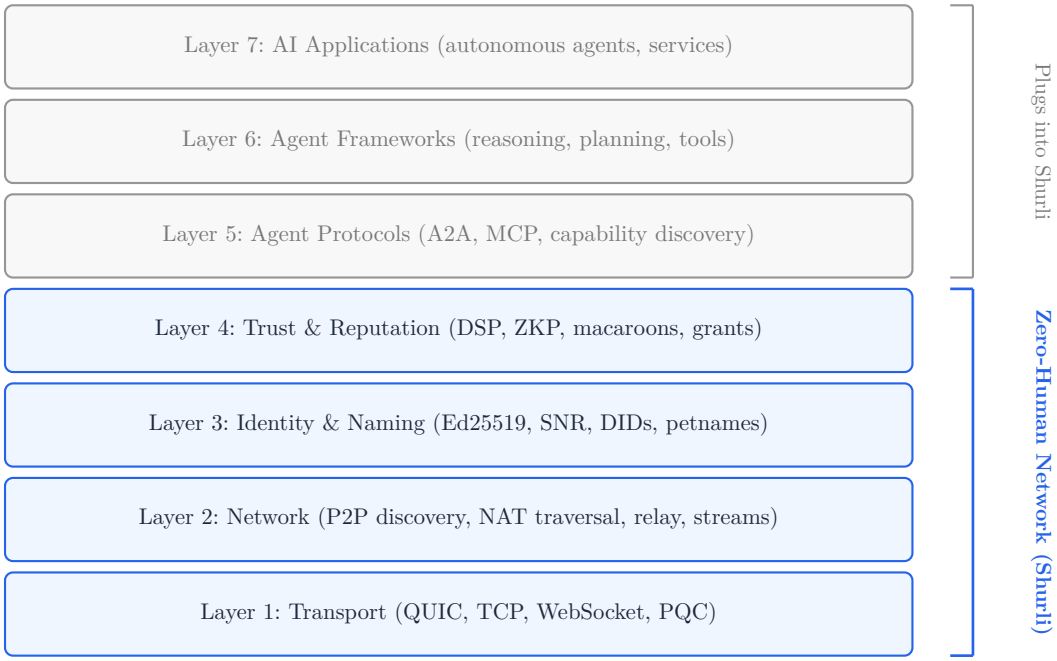

Figure 5: The AI infrastructure stack. Shurli ships layers 1 through 3 today; layer 4 (DSP reputation) is in early implementation. Agent protocols, frameworks, and applications sit above and plug in via the SDK and plugin architecture.

Layer 1: Transport. Direct device connectivity via libp2p with QUIC, TCP, and WebSocket transports. NAT traversal through DCUtR hole-punching with circuit relay v2 as fallback. Software-only techniques achieve 85-88% direct connection rates [9], [10]. Ephemeral "fat invite codes" encode all bootstrap data, eliminating persistent infrastructure dependency. CGNAT is a design requirement, not an edge case. The QUIC transport automatically negotiates hybrid post-quantum key exchange (X25519MLKEM768) [11] via Go's standard library.

Layer 2: Network. Kademlia DHT for peer discovery in an owned namespace. mDNS for zero-configuration LAN discovery. A PeerManager maintains connection lifecycle with promotion, demotion, and state tracking. Path selection is continuous: if a peer transitions from cellular to WiFi, the connection migrates automatically. Eleven upstream libp2p overrides harden the transport for production use.

Layer 3: Identity and Naming. A BIP32-style [12] hierarchical key tree derives all keys through HKDF-SHA256 with domain separation. The root seed may be generated from a BIP39 mnemonic, SLIP39 shares, or any method that produces sufficient entropy. One backup of the root seed recovers everything. The naming standard [13], [14] provides a five-layer resolution pipeline: PeerID (cryptographic ground truth), DID (standards interop via W3C `did:peer` [7]), petname (local, user-assigned), nickname (self-chosen, advisory), and external (resolved via plugins for ENS, DNS, VerusID, or any naming system). The relay is explicitly prevented from becoming a name authority.

Layer 4: Trust and Reputation. The DSP framework described in Sections 3 and 4. DAG-based append-only interaction log. Heuristic scoring as cold-start fallback, transitioning

smoothly to matrix factorization as ratings accumulate. Zero-knowledge range proofs allow peers to prove "my score exceeds threshold t" without revealing the score. Macaroon capability tokens [3] provide authorization with cryptographic attenuation, time-limited grants, delegation chains, and per-peer bandwidth budgets.

Above Shurli (Layers 5-7). Agent protocols (A2A, MCP, ANP), agent frameworks (OpenClaw, custom reasoning engines), and AI applications sit above the network layer. Shurli does not prescribe or constrain what runs above it. A compiled plugin architecture with an 11-method interface allows any application to extend the network. The SDK exposes transport, events, and peer management to plugins without leaking credentials or private keys. Layers 1-3 are shipped and production-tested. Layer 4 (DSP reputation with matrix factorization) is in early implementation; the layered trust model and Byzantine defense described in this paper are the target design.

5 Privacy

Institutional trust restricts information to parties and a central authority. DSP uses pseudonymous identities (libp2p peer IDs); only behavioral data, not content, is observed. Reputation derives from network behavior, not transmission content.

Figure 6: Privacy comparison. Institutional: all data flows through authority. DSP: only observable network behavior is visible to agents.

Petname stores are local and never shared over the network without consent. Namespace isolation prevents cross-network identity correlation. Name resolution queries to external resolvers may leak which names a node is interested in; this is documented as a known tradeoff. Zero-knowledge proofs enable anonymous authenticated participation: a peer proves its score exceeds a threshold without revealing who it is or what the score is. Post-quantum key exchange protects current sessions against future quantum decryption of recorded traffic.

6 Incentive Design

The reason Shurli exists is to make the network invisible. A human tells an AI what they need. The AI figures out the rest: finding the right peers, negotiating access, exchanging data and value with other AI agents across the network, and delivering the result. The human never manages connections, configures relays, or thinks about which node has which capability. The network operates itself. This is the Zero-Human Network in practice: not a network without humans, but a network where humans interact with AI and AI handles everything underneath.

This only works if the network layer requires no manual intervention, no token purchases, and no account signups. Decentralized networks that work without tokens share one pattern: immediate selfish value to each node. BitTorrent, email, and Tor all succeed because every

participant gains directly from participation. The user thinks about what they get, never about "the network." If the value requires explaining a vision, adoption fails.

Shurli follows this pattern. A node joins because it wants to reach its own machines behind NAT, share files with specific peers, or provide AI agents with autonomous connectivity. The network effect is a byproduct, not a prerequisite.

6.1 Why No Token

Shurli has no token, no coin, and no staking requirement. This is a deliberate architectural decision, not a temporary omission. Tokens create regulatory complexity, invite speculation that distorts incentives, and gate participation behind financial cost. A network layer should be invisible infrastructure, like TCP/IP. Nobody buys a token to send a packet.

The design philosophy parallels Bitcoin's original contribution [6]: eliminate middlemen, place trust in mathematics. But where Bitcoin required a token to solve double-spending (a financial problem), Shurli solves a networking problem. Reputation emerges from behavior. Authorization comes from capability tokens with mathematical guarantees. Neither requires a tradeable asset.

6.2 Agnostic Payment Layer

When economic exchange is desired, Shurli defines the payment interface, not the implementation. Payment methods are plugins: Lightning, USDC, traditional payment rails, or any future system. The protocol is agnostic. Per-task micropayments, not subscriptions. Swap rails without protocol changes.

This is consistent with Shurli's broader design: agnostic to payment, to naming, to identity, to agent frameworks. The core provides infrastructure. Everything else plugs in.

6.3 Positive-Sum Economics

Shurli is not adversarial to centralized AI providers. A self-hoster running local inference and a cloud provider operating at scale both benefit from a network layer that handles connectivity, trust, and authorization. The network is a distribution channel for centralized inference, not a competitor to it. Decentralized and centralized, not versus. Every participant gains; nobody's position is worsened. Raising the floor does not lower the ceiling.

7 Limitations

Probabilistic trust, not thermodynamic finality. Both blockchain and DSP provide probabilistic guarantees, but backed by different resources. In proof-of-work, the cost of reversing a transaction grows with accumulated computational work: older transactions become exponentially harder to undo, and time strengthens finality. In DSP, the cost of manipulating a reputation grows with observer diversity and cross-factional agreement, but time works in the opposite direction: older observations decay in relevance, and recent behavior carries more weight. This is by design. A relay that was reliable for two years but started dropping connections last week should lose reputation now, not coast on historical performance. For use cases that require immutable, time-strengthening finality (financial settlement, legal records),

blockchain remains appropriate. DSP provides adaptive probabilistic trust sufficient for networking decisions but not for irreversible transactions.

Cold start. New nodes lack reputation. The current mitigation is graduated trust escalation through low-stakes interactions: heuristic scoring for the first 5 ratings, blending to matrix factorization from 5 to 15 ratings, full MF-based scoring above 15. In private networks, invitation chains mathematically bound the Sybil fraction. In public networks, time cost (7 days), bandwidth cost, and interaction minimums (10) constrain rapid reputation gaming. Connected external identities via the naming standard provide a partial solution: a peer that links verified social profiles, messaging accounts, or enterprise directory entries inherits initial identity trust, reducing the cold-start gap while still requiring performance trust to be earned through real interactions.

Agent integrity. Compromised AI agents produce untrustworthy observations. Observer diversity requirements mitigate but do not eliminate this. The five-layer Byzantine defense (bilateral verification, MF consensus, relay observability, clique detection, rate limiting) ensures an attacker must compromise all layers simultaneously.

Observation scalability. Required observation density at global scale remains an open research question. The layered trust computation (relay computes, peers verify, gossip propagates) provides progressive decentralization, but has not been tested beyond small network deployments.

Current scale. Shurli v0.3.0 operates across a small number of nodes on heterogeneous networks. The architecture is designed for larger scale, but the claims in this paper are validated at small scale. This paper describes where Shurli is going, not claiming it has arrived.

NAT traversal is probabilistic. Direct connection success rates of 85-88% mean 12-15% of connections require relay fallback. Relay is infrastructure, not failure, but it introduces latency and bandwidth constraints.

Post-quantum identity. QUIC transport already negotiates hybrid PQ key exchange. Noise protocol and peer identity keys remain classical (Ed25519). ML-DSA for identity signing awaits Go standard library support. This is a known gap with a planned migration path.

8 Conclusion

The question is not whether distributed social proof replaces blockchain. It does not. The question is whether the majority of P2P networking use cases ever required blockchain-grade guarantees. We argue they do not.

For establishing connections, verifying peers, routing traffic, maintaining integrity, and tracking whether AI agents honestly complete the work they are asked to do, distributed agent observation is sufficient, and faster, cheaper, more expressive, and more adaptable than any ledger-based alternative.

None of the components in Shurli are novel. libp2p [8] for transport. Matrix factorization for bridged consensus [1]. EigenTrust [15] for trust propagation. Zero-knowledge proofs [2] for privacy. Macaroons [3] for capability-based authorization. Petname systems [14] for identity-agnostic naming. Hybrid post-quantum key exchange [11], [16] for forward security. The contribution is the combination: a Zero-Human Network where AI agents and humans connect,

communicate, and build trust without centralized infrastructure, without blockchain, and without requiring anyone's permission.

Shurli is the experiment. This paper will evolve with it.

9 Acknowledgments

The authors acknowledge the X Community Notes team for the bridging algorithm, Protocol Labs for the libp2p transport stack, the Go cryptography team for post-quantum primitives, the Consensys gnark team for the ZKP circuit compiler, and the PQNoise authors for post-quantum Noise protocol research.

Bibliography

[1] S. Wojcik and others, "Birdwatch: Crowd Wisdom and Bridging Algorithms Can Inform Understanding and Reduce the Spread of Misinformation." [Online]. Available: https://arxiv.org/abs/2210.15723

[2] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186–208, 1989, [Online]. Available: https://people.csail.mit.edu/silvio/Selected%20Scientific%20Papers/Proof%20Systems/The_Knowledge_Complexity_Of_Interactive_Proof_Systems.pdf

[3] A. Birgisson, J. G. Politz, Ú. Erlingsson, A. Taly, M. Vrable, and M. Lentczner, "Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud," in *Network and Distributed System Security Symposium (NDSS)*, 2014. [Online]. Available: https://theory.stanford.edu/~ataly/Papers/macaroons.pdf

[4] X (formerly Twitter), "Community Notes Guide." [Online]. Available: https://communitynotes.x.com/guide/

[5] J. R. Douceur, "The Sybil Attack," 2002, [Online]. Available: https://www.freehaven.net/anonbib/cache/sybil.pdf

[6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, [Online]. Available: https://bitcoin.org/bitcoin.pdf

[7] World Wide Web Consortium, "Decentralized Identifiers (DIDs) v1.1." [Online]. Available: https://www.w3.org/TR/did-core/

[8] Protocol Labs, "libp2p: A Modular Network Stack." [Online]. Available: https://libp2p.io/

[9] D. Trautwein, C. Ihle, M. Schubotz, and B. Gipp, "Challenging Tribal Knowledge: Large Scale Measurement Campaign on Decentralized NAT Traversal." [Online]. Available: https://arxiv.org/abs/2510.27500

[10] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-Peer Communication Across Network Address Translators," in *USENIX Annual Technical Conference*, 2005. [Online]. Available: https://www.usenix.org/legacy/events/usenix05/tech/general/full_papers/ford/ford.pdf

[11] National Institute of Standards and Technology, “Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM).” [Online]. Available: https://csrc.nist.gov/pubs/fips/203/final

[12] P. Wuille, “BIP-32: Hierarchical Deterministic Wallets.” [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

[13] Z. Wilcox-O'Hearn, “Names: Distributed, Secure, Human-Readable: Choose Two.” [Online]. Available: https://web.archive.org/web/20011020191610/http://zooko.com/distnames.html

[14] IETF, “The GNU Name System.” [Online]. Available: https://www.rfc-editor.org/rfc/rfc9498

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The EigenTrust Algorithm for Reputation Management in P2P Networks,” in *Proceedings of the 12th International Conference on World Wide Web*, 2003, pp. 640–651. [Online]. Available: https://dl.acm.org/doi/epdf/10.1145/775152.775242

[16] Y. Angel, B. Dowling, and A. Hülsing, “Post-Quantum Noise,” in *ACM Conference on Computer and Communications Security (CCS)*, 2022. [Online]. Available: https://eprint.iacr.org/2022/539.pdf