Freeport: A Peer-to-Peer Marketplace Protocol

Phil Trinh

ptrinh@me.com

Draft v0.2 — June 2026

Abstract. A purely peer-to-peer marketplace would allow buyers and sellers of local goods and services to transact directly without going through a platform intermediary. Digital signatures and open relays provide part of the solution, but the main benefits are lost if a trusted third party is still required to establish reputation. We propose a solution to the trust problem using a web of co-signed deal receipts. Participants publish signed *intents* to open relays; counterparties negotiate terms over encrypted messages using a bounded state machine; a deal is proven only when both parties independently sign a receipt referencing it. Reputation is not a global score but a subjective quantity each viewer computes by walking the receipt graph outward from their own key, so fabricated identities — however numerous — carry no weight with anyone they have not actually served. The protocol requires no platform, no fees, no account, and no permission. Markets are defined by convention: any group of people agreeing on a topic tag and a payload schema constitutes a marketplace.

1 Introduction

Commerce between individuals on the internet has come to rely almost exclusively on platforms serving as trusted third parties: ride-hailing companies, gig marketplaces, classified-ad sites. While the model works well enough for most transactions, it suffers from the inherent weaknesses of the trust-based model. Platforms extract commissions of 15–30% of transaction value. They unilaterally set terms, suspend accounts without recourse, and hold reputation hostage — a seller's years of reviews cannot leave the platform that hosts them. Network effects entrench incumbents; participants on both sides pay monopoly rents. The platform mediates disputes it has incentives to resolve in its own interest, and it surveils every transaction as a condition of participation.

What is needed is a marketplace based on cryptographic proof instead of platform trust, allowing any two willing parties to find each other, agree on terms, and build durable reputations without a trusted third party. In this paper, we propose a solution using signed intents on open relays, encrypted bilateral negotiation, and a reputation system in which proof of past trade is established by mutual digital signature and weighted by each observer's own position in the trade graph.

2 Identities

An identity is a secp256k1 key pair, generated locally on first use, as in Nostr [2]. There is no registration, no account, and no issuing authority. The public key identifies a participant across all markets; the private key signs every event the participant publishes and decrypts messages addressed to them. Identity is portable by construction: reputation accrues to the key, not to any service, and the key can be backed up (encrypted under a passphrase per NIP-49) and restored on any client.

Keys are free, which makes identity creation costless — the central problem this design must address. We return to it in Sections 7–8.

3 Intents

A market posting is an *intent*: a signed, addressable event declaring that its author wants to buy or sell something, where, when, and on what terms.

```
kind:  32101 (offer) | 32102 (request)
tags:  d = stable intent id          (replaceable by author)
       t = market topic              (e.g. "sg-rideshare")
       g = geohash prefix            (location scope)
       expiration                    (NIP-40)
content: { schema, title, payload, window, flex_minutes, ... }
```

The `payload` is schema-keyed (`rideshare/1`, `service/1`, ...) so the protocol is vertical-agnostic: a rideshare market, a plumbing market, and a homemade-food market differ only in payload schema and topic tag. Anyone can define a new market by publishing intents under a new tag; no registry exists and none is needed. Geohash tags scope discovery to a neighborhood; expiration tags let relays garbage-collect stale postings. Because intents are addressable, an author can revise or withdraw a posting by republishing under the same `d` tag — a withdrawal is just a republished event carrying a near-term expiration, so relays drop it on schedule rather than serving a posting the author has abandoned.

Each intent carries a small proof-of-work (NIP-13) so that posting into a market has a nonzero cost, throttling flooding without a gatekeeper. The work is modest — a few leading zero bits — and clients mine it off the main thread so the interface stays responsive while it is computed.

Relays are the only infrastructure, and they are interchangeable commodity infrastructure: dumb stores that accept signed events and serve filtered subscriptions [2]. A market lives on whichever relays its participants choose to use; censorship by one relay is routed around by publishing to others. Clients publish to several relays and read from several, so no relay is a point of control.

4 Negotiation

Discovery is public; negotiation is private. When a counterparty responds to an intent, the parties exchange encrypted direct messages (NIP-04, with NIP-17 planned) carrying JSON envelopes that drive a small state machine:

```
open --counter--> open ... (<= 8 rounds)
  |
accept --> accepted_by_us / accepted_by_them
  |                          |
  +----- second accept ------+--> confirmed
  |                                  |
  |               owner cancel ------+ (filled) --> cancelled
cancel / timeout --> cancelled / expired
```

Each `counter` proposes a complete replacement set of terms (time window, payment, route or scope), so there is no ambiguity about what is on the table. A deal requires a *double accept*: one party accepts the standing terms, and the deal is confirmed only when the other party accepts in turn. This closes the race in which both sides simultaneously counter and each believes a different version was agreed. The round limit bounds negotiation spam.

A frequent special case collapses the double accept to a single tap. When a responder accepts the standing terms *unchanged* — a driver taking a ride at the posted price — no new terms are introduced, so the client records an immediate confirmation rather than opening another round. Confirmation remains provisional on the responder's side, however, because a single request can draw several competing responders. The intent owner confirms at most one: on doing so the owner's client withdraws the intent and sends an authoritative cancellation — *filled, taken by another offer* — to the losing responders. Receiving it moves a negotiation the responder had locally marked `confirmed` to `cancelled`, resolving the race so that no two parties believe they hold the same deal. A `completed` deal is terminal and

is never reversed by such a message.

On confirmation, the parties exchange contact information inside the encrypted channel — by design the protocol's job ends at the handshake; the ride or the repair happens in the physical world.

5 Deal Receipts

The negotiation transcript is private and either party could fabricate one, so confirmed deals must be anchored publicly. When a negotiation reaches `confirmed`, each party publishes a *receipt*:

```
kind:  32104
tags:  d = negotiation id, p = counterparty
```

A deal is **proven** if and only if both halves exist: A signed a receipt naming B, and B signed a receipt naming A, for the same negotiation id. A single key cannot manufacture a proven deal; it takes two signatures from two keys. This does not prevent one person from controlling both keys — no cryptographic scheme can — but it forces every fabricated deal to construct an edge in a public graph, and Section 7 shows why fabricated edges are worthless.

Receipts are addressable by negotiation id, so each party can publish at most one receipt per deal, and they reference the intent so the full chain — posting, negotiation id, mutual confirmation — is auditable by any verifier.

6 Ratings

After a proven deal, each party may rate the other once:

```
kind:  32103
tags:  d = negotiation id, p = ratee
content: { score in {-1, 0, +1, +2}, note?,
           contact_verified?, contact_masked? }
```

The d-tag makes ratings idempotent per (rater, deal) — re-rating replaces rather than accumulates. A verifier counts a rating only if (a) the negotiation id matches a proven receipt pair and (b) the rater is the counterparty of that very deal. Ratings detached from proven deals are discarded. Repeated ratings between the same pair decay geometrically $(1, \frac{1}{2}, \frac{1}{4}, \dots)$: the quantity that matters is *distinct counterparties*, which is exactly the quantity made expensive by Section 7.

The optional `contact_verified` field is a peer attestation that the rater actually reached the counterparty at their advertised (masked) phone number, binding the public contact claim to a private exchange — see Section 8. Rating events additionally carry a small proof-of-work (NIP-13) as a spam floor.

7 Reputation as a Subjective Quantity

It is impossible to build a Sybil-proof global reputation score when identities are free [4]. Rather than fight this theorem, the protocol abandons the global score. Reputation in Freeport is *subjective*: each viewer computes the weight of every rater from their own vantage point by breadth-first traversal of the proven-receipt graph:

hop 0	people the viewer has proven deals with	weight 1.0
hop 1	their proven counterparties	weight 0.3
hop 2	one ring further	weight 0.1
—	unreachable	weight ≈ 0

The viewer's social follow list (NIP-02) seeds the map so that newcomers with no deal history still see through the eyes of people they chose to follow. A subject's displayed reputation is then the trust-weighted average of valid ratings, alongside the raw counts: *deals, distinct partners, partners within your network*.

Consider an attacker who fabricates n puppet identities and has them trade and rate each other indefinitely. Every rating passes the receipt check — the attacker controls both signatures. But the puppet cluster is connected to an honest viewer only if some honest participant has a proven deal into it. With no such edge, every puppet rater carries the unreachable-rater floor weight ε, and the attacker's influence on the viewer's perception is bounded by ε regardless of n. The attack surfaces in the UI as its own signature:

```
honest:  * 12 deals - 9 partners - 3 in your network
           - verified by 9
sybil:   50 deals - 50 partners - 0 in your network
           - new account
```

To gain real influence the attacker must complete genuine deals with genuinely connected participants — at which point he is simply a participant. The cost of attacking the system converges to the cost of using it honestly, which is the property we require. Dampeners narrow the remaining margins: a rater whose entire receipt history is deals with one subject is discounted (the two-puppet pattern), accounts with no event history older than a week are flagged, and unknown-rater weight ε keeps distant ratings visible but uninfluential.

8 Privacy and Contact Binding

Identity is pseudonymous by default; the protocol adds disclosure only where it pays for trust, and lets the user choose the dose.

Phone numbers. A participant may attach a phone number to their profile, published either in full (explicit opt-in, with a permanence warning) or masked to country code and final digits (`+84••••••678`). Masking is applied *before* publication; the full number never leaves the device except inside the encrypted channel at deal confirmation. A self-published mask proves nothing — anyone can publish any string — so the protocol binds it to reality through peers: at deal time the full number travels to the counterparty over the encrypted channel; when rating, the counterparty's client attests the canonical mask of the number it *actually reached*. Verifiers cross-check attested masks against the subject's published mask and discard attestations on mismatch. A fake public mask is thus exposed by the first honest deal, and "verified by N partners" means N proven counterparties independently confirmed reachability — a peer-produced verification requiring no SMS provider and no central verifier.

Negotiations are end-to-end encrypted and invisible to relays beyond traffic metadata. **Location** is published only at geohash precision chosen by the client ($\approx \pm 0.6$ km), honest about its own imprecision. Reputation binds to the key, not the phone: rotating numbers neither transfers nor escapes a rating history.

9 Incentives

The protocol has no token, no fee, and no operator, which raises the question of why infrastructure exists. The answer is that the infrastructure is nearly free. Relays are commodity event stores, already operated in the hundreds by the Nostr ecosystem for social traffic; a community marketplace fits comfortably in their economics, and any community can run its own relay for the cost of a small server. Clients are ordinary apps competing on quality, none privileged by the protocol. The parties to a deal settle payment by whatever means they already trust — cash, bank transfer, or, in a later protocol phase, Lightning [5], whose escrow-by-hold-invoice could also serve as a costly-to-forge signal strengthening Section 7's graph.

What the design deliberately forecloses is the platform's rent: there is no position in the protocol from which to take a cut of trades, because there is no chokepoint through which trades must pass. Discovery, negotiation, proof, and reputation are all client-side computations over public, signed data.

10 Comparison with Platforms

	Platform	**Freeport**
Listing	account required, ToS	signed event, no permission
Matching	proprietary algorithm	open filters, client-side
Fees	15–30% commission	none
Reputation	custodial, non-portable	signed, portable, subjective
Dispute	platform adjudication	reputation consequences
Censorship	account suspension	publish to another relay
Privacy	full transaction surveillance	encrypted negotiation, pseudonyms

The honest trade-off is dispute resolution: a platform can refund a buyer and ban a seller; Freeport can only ensure the bad deal is provably attributable and permanently visible to every future counterparty who looks. For low-value, high-frequency local commerce — rides, repairs, food — reputation consequences are the mechanism that already governs offline behavior, and the protocol's contribution is to make that mechanism portable, verifiable, and unforgeable at scale.

11 Conclusion

We have proposed a marketplace without a trusted third party. Participants declare intents on open relays, negotiate privately under a bounded state machine, and anchor completed deals with mutual signatures. Reputation is recast from a global score — impossible to defend with free identities — into a subjective quantity computed over the proven-deal graph from each viewer's own position, making Sybil clusters visible and weightless while honest history compounds. Contact claims are bound to reality by peer attestation rather than by any verifier. The protocol is small: four event kinds and one message envelope over an existing relay network. Markets themselves are conventions — a tag and a schema — so anyone may open one, and no one may close one.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[2] fiatjaf et al., "Nostr: Notes and Other Stuff Transmitted by Relays — NIP-01: Basic protocol flow," `github.com/nostr-protocol/nips`.

[3] Nostr NIPs employed: NIP-02 (follow lists), NIP-04/NIP-17 (encrypted messaging), NIP-13 (proof of work), NIP-19 (bech32 entities), NIP-40 (expiration), NIP-44 (versioned encryption), NIP-49 (key encryption), NIP-96/98 (HTTP file storage and auth).

[4] J. R. Douceur, "The Sybil Attack," *IPTPS*, 2002.

[5] J. Poon, T. Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments," 2016.

[6] Q. Cao, M. Sirivianos, X. Yang, T. Pregueiro, "Aiding the Detection of Fake Accounts in Large Scale Social Online Services" (SybilRank), *NSDI*, 2012.

[7] H. Yu, M. Kaminsky, P. B. Gibbons, A. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks," *SIGCOMM*, 2006.