Omerta: A Trust-Based Alternative to Blockchain Consensus for Decentralized Compute Markets

Technical Whitepaper

January 2026

Abstract

Decentralized compute sharing faces a fundamental challenge: how can strangers cooperate without a trusted central authority? Blockchain-based systems address this through proof-of-work or proof-of-stake consensus, achieving Byzantine fault tolerance at the cost of significant resource overhead and limited throughput. We present Omerta, a practical implementation that synthesizes decades of research on trust, reputation, and mechanism design into a working system for decentralized compute markets.

Omerta computes trust *locally* relative to each observer, rather than maintaining global scores. Trust is derived from verified transactions—actual compute sessions with measurable outcomes—rather than subjective ratings. This design is *machine-native*: unlike prior reputation systems that assumed humans would rate each other, Omerta expects trust signals to be generated automatically from transaction outcomes, enabling measurement at scales human rating could never achieve. This enables natural scaling without global consensus overhead, at the cost of accepting that different observers may have different views of the same identity's trustworthiness. Crucially, local computation does not mean isolated computation: transactions are witnessed by third parties, records propagate through gossip, and cryptographic signatures ensure authenticity. We retain the benefits of distributed verification and immutable records—we simply don't require the entire network to agree on a single global ordering before transactions can proceed.

This paper presents: (1) a trust model derived from verified transactions rather than subjective ratings; (2) local trust computation with path decay; (3) automated monetary policy that adjusts parameters in response to detected threats; (4) economic analysis demonstrating when unreliable home compute creates genuine value; and (5) double-spend resolution where currency "weight" scales with network performance, providing a practical alternative to global consensus for bilateral transactions.

We draw on the observation that human societies have always traded privacy for trust—villages had high trust precisely because everyone knew everyone's business. Omerta recreates this visibility at global scale through on-chain transparency. Unlike villages with their arbitrary social punishment, we aim to maximize freedom within the trust constraint: only verifiable anti-social behavior—failed deliveries, double-spends, verification failures—affects trust scores, where "verifiable" means the determination is reproducible from on-chain data by any observer.

We argue that implementing fair trust systems at scale was computationally intractable until machine intelligence provided the reasoning capacity to model behavior, tune parameters, and explain decisions. This paper itself was developed through human-machine collaboration, demonstrating the thesis: machine intelligence both demands the compute that systems like Omerta could provide and enables the trust mechanisms that make such systems work.

Prior trust systems like EigenTrust and FIRE were never widely deployed—they remained academic exercises, computing trust scores that connected to nothing. Omerta brings these ideas into implementation with real economic consequences: trust scores that affect payments, transfer costs, and access. The contribution is both theoretical and practical—new mechanisms where prior work was insufficient, adoption of proven approaches where they exist, and integration into a working system. The software is given away for free with no preallocation of tokens.

Unlike prior decentralized compute platforms that struggled with adoption, Omerta targets a specific opportunity: billions of home computers sit idle most of the time, representing low marginal cost compute for owners who have already paid for hardware and internet. The software will provide transparency about actual operating costs—electricity, bandwidth, wear—with user controls for participation thresholds. The software is open source with no platform fees. And machine intelligence dramatically increases the utility of distributed compute—enabling humans to orchestrate complex parallel workloads across unreliable infrastructure in ways they couldn't manage manually. This combination of low-cost supply, zero-rent platform, and machine-intelligence-amplified demand may succeed where blockchain-based alternatives with mining overhead, token economics, and human-centric design have struggled.

Contents

1 Introduction

The vision of decentralized computing—where anyone can contribute resources and anyone can consume them, without intermediaries extracting rents—has motivated decades of research. The core challenge is threefold: protecting against the rare bad actors who spoil cooperation for everyone, reducing the technical barriers that have limited participation to experts, and making the experience simple enough to be worthwhile. The practical reality is that most participants in even "trustless" systems don't write their own code—they trust wallet software, exchange interfaces, and protocol implementations written by others. The question is not whether to trust, but whom, how much, and at what cost.

1.1 A Brief History of Trust Systems

The question of computational trust saw intensive research in the early 2000s, driven by the rise of peer-to-peer file sharing networks and online marketplaces. Researchers developed algorithms to compute reputation scores, integrate multiple trust sources, reason under uncertainty, and detect manipulation. This body of work established core insights that remain valid: trust propagates through networks with decay; local computation can substitute for global consensus; time and history provide unforgeable credentials; and detection of manipulation patterns enables defensive responses. (See Section 2 for detailed treatment of specific systems.)

Yet these systems were never widely deployed. They remained academic exercises—published, cited, and largely forgotten in practice. Why?

First, they were purely reputational. EigenTrust computed trust scores, but those scores didn't connect to payments, incentives, or economic flows. Trust was a number with no consequences. Without economic integration, there was no compelling reason to deploy them.

Second, they lacked a killer application. P2P file sharing—the motivating use case—didn't have the economics to justify sophisticated trust systems. Free music downloads didn't require trustworthy cooperation; users would simply try another node. The theory outpaced the need.

Third, and perhaps most importantly, blockchain arrived. Bitcoin (2008) appeared to solve the trust problem through cryptographic consensus. Research attention shifted. Why model trust computationally when proof-of-work could enforce cooperation mathematically? The reputation systems literature quieted.

A decade later, we understand blockchain's limitations more clearly. Budish [2] demonstrated that blockchain security has inherent economic limits—the recurring costs of running a secure blockchain must be large relative to the value at stake, making it expensive for high-value applications. Proof-of-stake faces similar constraints [10]. Existing decentralized compute networks built on blockchain (Golem [24], iExec [14]) have struggled with adoption despite years of operation. The costs of global consensus may exceed what many applications require.

Meanwhile, machine intelligence has advanced dramatically. Tasks that seemed intractable—modeling complex behavior, tuning parameters across high-dimensional spaces, generating explanations for decisions—are now feasible. This creates an opportunity: **the trust systems research of 2000–2010 may be ready for practical implementation, enabled by machine intelligence capabilities that didn't exist when the theory was developed.**

Omerta represents an attempt at this synthesis. We return to the trust-based approaches developed before blockchain's dominance, informed by what we've learned since, and enabled by machine intelligence to handle the complexity that made pure implementation difficult.

1.2 Three Paths

Traditional approaches fall into two categories. **Trusted intermediary** models, exemplified by cloud computing providers, centralize authority in organizations that can enforce contracts and punish misbehavior. This works but introduces single points of failure, censorship risk, and

rent extraction. **Blockchain-based** models, exemplified by Ethereum and its descendants, replace trusted intermediaries with cryptographic consensus protocols that theoretically eliminate the need for trust. This also works but imposes significant costs: massive energy expenditure (proof-of-work), capital lockup requirements (proof-of-stake), limited transaction throughput, and delayed finality.

Between these extremes lies a spectrum of approaches trading trust for cost. Fully homomorphic encryption (FHE) and multi-party computation (MPC) enable trustless computation but with 1,000–1,000,000× overhead. Trusted execution environments (TEEs) like Intel SGX reduce overhead but trust hardware manufacturers. Layer-2 solutions (rollups, sidechains) inherit security from a base chain while improving throughput, but still pay consensus costs. Smart contracts execute deterministically but are limited to on-chain data and simple computations.

We propose a third path—or rather, we return to one that was overshadowed. Omerta is a trust-based distributed compute network that neither centralizes authority nor attempts to eliminate trust. Instead, it makes trust *subjective*, *local*, and *earned*—computed by each participant from their own position in the network, based on verifiable on-chain data accumulated over time. This approach builds directly on EigenTrust, FIRE, and related work, while making specific adaptations for compute markets and leveraging machine intelligence for implementation.

1.3 The Trust Spectrum

Trustlessness is not binary—it exists on a spectrum. Proof-of-work and proof-of-stake mechanisms genuinely increase trustlessness compared to centralized alternatives. They represent real achievements in distributed systems research. However, historical episodes reveal that a social layer always remains:

The DAO Hack (2016): An attacker exploited a smart contract vulnerability to drain $60 million from The DAO. The Ethereum community responded with a hard fork that reversed the theft—creating Ethereum (rolled back) and Ethereum Classic (preserved the "immutable" history). The community chose social consensus over mechanical execution.

Bitcoin Value Overflow (2010): A bug created 184 billion bitcoins out of thin air. Developers and node operators coordinated to deploy a fix and roll back the chain. Human judgment overrode the protocol when stakes were high enough.

These interventions were controversial—but the social layer can also work as intended:

Exchange Coordination: When exchanges collectively delist contentious tokens or coordinate responses to theft, they demonstrate genuine community action in support of shared values. This is humans exercising collective judgment when protocol alone is insufficient, and it represents the system working, not failing.

These episodes do not invalidate blockchain achievements. Rather, they reveal that we operate on a spectrum from full trust (centralized authority) to reduced trust (cryptographic consensus) to some irreducible social layer. Sometimes that social layer intervenes controversially; sometimes it acts in clear support of community values. No practical system reaches the zero-trust endpoint—nor should it.

The question becomes: given that we cannot achieve absolute trustlessness anyway, what are we paying for the trustlessness we do achieve? And could we relax our requirements slightly to capture most of the practical benefit at dramatically lower cost?

This is the same reasoning that motivates ephemeral compute over fully homomorphic encryption (FHE). FHE provides the ultimate guarantee: compute on encrypted data without ever decrypting it. No trust in the compute provider required. But FHE imposes 1,000–1,000,000× computational overhead [11], making it impractical for most workloads. Ephemeral compute—where data exists briefly on untrusted hardware with verification and economic penalties—provides weaker guarantees but serves far more use cases at practical cost.

Omerta applies this spectrum thinking to consensus itself. Blockchain consensus mechanisms genuinely reduce trust requirements, but at significant cost: energy expenditure (PoW), capital

lockup (PoS), limited throughput, and delayed finality. We ask: for compute markets specifically, can we relax the global consensus requirement while preserving the practical security properties that matter?

Our hypothesis is yes. Compute markets do not require global agreement—they require pairwise trust between specific buyers and sellers. By computing trust locally rather than achieving global consensus, Omerta aims to capture most of the practical benefit of decentralization at dramatically lower cost, making trustworthy compute sharing accessible to more people.

Compute is also uniquely suited to trust-based systems because of the nature of the goods being traded:

- **Revocable**: Providers can reclaim their machines at any time. Unlike transferring money or physical goods, access can be terminated instantly.
- **Low stakes per transaction**: Each session consumes only a bit of time, energy, and wear. No single transaction is catastrophic.
- **Already sunk costs**: Most home computers sit idle—owners have already paid for hardware, electricity, and internet. The marginal loss from a bad transaction is minimal.
- **Verifiable during execution**: Compute can be checked while running through heartbeats, random audits, and result validation. Fraud is detectable, not just punishable after the fact.

These properties make compute an ideal domain for experimenting with trust-based systems. The downside risk is bounded, the verification is tractable, and the resources were often going unused anyway.

1.4 Our Contribution

This paper's primary contribution is a **practical synthesis**—bringing established trust system research into implementation for compute markets. We distinguish between mechanisms adapted from prior work and novel contributions:

Adapted from prior work (with modifications):

1. **Local trust computation** extending EigenTrust [16], TidalTrust [12], and path-based approaches [25, 28]. Where EigenTrust computes global scores, Omerta computes trust relative to each observer—a design approach with substantial prior art [12] that we apply to compute markets.
2. **Trust propagation with decay**, a standard technique in graph-based trust systems [16, 13, 28], applied here to transaction histories rather than explicit ratings.
3. **Age-based Sybil resistance**, building on temporal defense mechanisms discussed since Douceur's original Sybil attack paper [7] and whitewashing analysis [9]. We note that this defense has known limitations (Section **??**).
4. **Cluster detection for Sybil defense**, applying standard anomaly detection techniques [27, 5] to transaction graph analysis.

Novel contributions:

5. **Economic mechanisms integrated with trust**. Prior reputation systems (EigenTrust, TidalTrust, FIRE, PowerTrust) are purely reputational—they compute scores but don't connect them to economic flows. Even TrustChain [20], which uses bandwidth tokens, doesn't tie token mechanics to trust scores. Omerta introduces:

- *Trust-based payment splits*: Provider earnings scale with trust score via an asymptotic formula
- *Transfer burns*: Coin transfers taxed based on minimum trust, preventing reputation laundering
- *Trust-proportional distribution*: Daily coin minting distributed proportionally to trust scores
- *Negative bids*: Providers can burn coins to accelerate trust building through verified work

6. **Structured accusation mechanism**. Prior systems use binary ratings or simple scores. Omerta's trust assertions include evidence hashes, derive credibility from the asserter's own trust (recursive), and apply impact/context multipliers for appropriate gray areas. This enables nuanced handling of infractions.

7. **Double-spend resolution via currency weight**. Since we detect double-spends rather than prevent them, transaction finality depends on how quickly conflicting transactions would be discovered. Omerta introduces "currency weight"—the confidence level required before a transaction is considered final—that scales with network connectivity.

8. **On-chain verification logs**. Prior systems record transactions or ratings, but not third-party verification results. Omerta stores verification outcomes on-chain, enabling trust computation to incorporate objective performance data rather than only self-reported transactions.

9. **Application to compute markets**. The specific integration of trust, payment, verification, order book, and session lifecycle mechanisms for decentralized compute rental is novel, though individual components draw on established work.

10. **Machine-native trust measurement**. Prior reputation systems (eBay, EigenTrust, FIRE) assumed humans would rate each other—clicking stars, leaving feedback, issuing certifications. Omerta assumes trust signals are generated automatically from verified transaction outcomes. Human input comes through engineering the automation and reviewing edge cases, not through direct rating. This enables trust measurement at scales and frequencies that human rating could never achieve.

2 Related Work

Omerta draws on multiple research traditions that have evolved largely independently. Reputation systems from e-commerce and P2P networks established how to aggregate trust signals. Sybil resistance research addressed identity manipulation. Blockchain and consensus work explored the trust-cost spectrum. Secure computation approaches (FHE, MPC, TEEs) pushed toward trustless execution at high cost. Volunteer and commercial distributed computing projects demonstrated both the potential and the pitfalls of shared compute. Finally, computational economics provided tools for modeling incentives and validating mechanism designs.

A key distinction runs through all comparisons: prior reputation systems assumed humans would rate each other. Omerta assumes machine-generated trust signals from verified transactions. This difference is fundamental—it changes what scales are achievable and what attack surfaces exist.

This section surveys each tradition, identifies what Omerta borrows, and clarifies where we diverge.

2.1 Reputation Systems

The challenge of establishing trust among strangers online has motivated extensive research on reputation systems [21, 22]. Resnick et al. [21] identified the core requirements: long-lived identities, captured feedback, and feedback-guided decisions. The eBay feedback mechanism demonstrated these principles at scale, though its binary ratings created opportunities for manipulation [6]. Tadelis [22] provides a comprehensive review of feedback systems in online platforms, documenting issues like grade inflation, retaliation concerns, and fraudulent reputation building.

More sophisticated approaches emerged from peer-to-peer networks in the early 2000s:

EigenTrust [16] computed global trust through iterative aggregation similar to PageRank. Its key insight—that trust can be aggregated through matrix operations—remains foundational. However, it produces global scores rather than observer-relative trust, and relies on explicit transaction ratings.

PeerTrust [25] incorporated transaction context, feedback scope, and community context factors. It recognized that trust depends on more than simple rating counts. Omerta adopts this insight but derives context from transaction records rather than explicit metadata.

FIRE [13] integrated four trust sources: interaction trust (direct experience), role-based trust (position in organization), witness reputation (third-party reports), and certified reputation (references from trustees). Omerta shares the recognition that trust has multiple components, though we deliberately exclude witness and certified reputation to eliminate subjective input vectors—relying only on verifiable transaction outcomes.

Subjective Logic [15] provided mathematical foundations for reasoning under trust uncertainty, modeling opinions as probability distributions with explicit uncertainty parameters. Jøsang's framework for trust transitivity and fusion operations could strengthen Omerta's formal foundations; we note this as future work.

PowerTrust [28] discovered power-law distributions in user feedback patterns and leveraged this for faster convergence through "power nodes." Omerta doesn't explicitly designate power nodes, but we expect trust scores to follow a similar power-law distribution—early participants, reliable providers, and high-volume traders will accumulate disproportionate trust. This is arguably a feature (natural meritocracy where proven participants gain influence) and a risk (concentration that could enable collusion or create single points of failure). We acknowledge this dynamic rather than pretending it won't occur.

Similarities with prior work: Trust propagation with decay, local computation principles, transaction context sensitivity, and the recognition that different trust components require different handling.

What Omerta changes: Trust derives primarily from verified on-chain transactions rather than subjective ratings. This dramatically reduces the fake feedback attack surface—some surface remains (colluding parties can generate fake transactions), but statistical analysis makes this harder than simply posting fake reviews. Humans can still influence trust scores directly through assertions, but at a cost: making an accusation stakes the asserter's own credibility. If you assert something others can't verify, you pay for it in trust. This mirrors how human trust networks actually work—when someone makes an accusation in a social group, their own reputation is on the line. Baseless accusations damage the accuser. The protocol codifies this natural dynamic, creating economic pressure toward honest, explainable feedback while preserving the ability to flag genuinely problematic behavior when the cost is worth it. Meanwhile, we collect objective metrics from every transaction (latency, uptime, resource delivery, verification outcomes)—more data than occasional human ratings could ever provide.

The fundamental shift: All systems above assumed humans would generate trust signals—clicking stars, writing reviews, issuing certifications. This creates inherent scale limits: humans won't rate every file download, every API call, every 30-second compute session. It also creates attack surfaces: fake reviews, rating manipulation, retaliation. Omerta assumes trust signals are generated primarily by machines observing verified transaction outcomes. Human input enters the system through multiple channels: engineering the automation that generates ratings, setting policy parameters, reviewing edge cases that automated systems flag, and—when warranted—direct assertions that stake the asserter's own credibility. This same mechanism allows machine intelligences—including future superhuman systems—to participate in the trust network: they can make assertions, stake credibility, and have their judgments weighted by their accumulated trust like any other participant. The protocol is agent-agnostic. Compute markets may be an almost ideal testing ground for such intelligences: they operate on resources society has already deemed marginally beneficial (idle compute that would otherwise go unused), security mechanisms on personal hardware bound potential damage, and users retain the ability to reclaim their machines at any time. If something goes wrong, the system can be shut down—unlike financial markets or critical infrastructure where superhuman participants could cause irreversible harm. This is not a minor implementation detail—it fundamentally changes what scales are achievable, what attacks are possible, and what kinds of intelligence can safely participate.

TrustChain [?] from the Tribler project deserves special attention as the most architecturally similar prior work. Like Omerta, TrustChain uses bilateral ledgers without global consensus, detecting double-spending rather than preventing it. Both scale by avoiding network-wide agreement. However, TrustChain focuses on bandwidth accounting for file sharing, while Omerta integrates trust with economic mechanisms for compute markets.

2.1.1 Mechanism Comparison

The following table compares Omerta's mechanisms against prior work. Checkmarks (Yes) indicate the mechanism is present; dashes (—) indicate absence.

Key observations:

1. **Economic integration is novel**: No prior trust system integrates trust scores directly with payment splits, transfer taxation, or coin distribution. EigenTrust, TidalTrust, and FIRE are purely reputational; TrustChain has bandwidth tokens but not trust-weighted economic splits.

2. **On-chain verification logs are novel**: Prior systems record transactions or ratings, but not verification results from third-party audits. This enables trust to incorporate objective performance data.

Table 1: Comparison of trust mechanisms across systems

Mechanism	**EigenTrust**	**TidalTrust**	**FIRE**	**TrustChain**	**Omer**
Trust Computation					
Global trust scores	Yes	—	Yes	—	—
Observer-relative (local) trust	—	Yes	—	—	Yes
Path-based decay	Yes	Yes	—	—	Yes
Data Source					
Subjective ratings	Yes	Yes	Yes	—	—
Verified transactions only	—	—	—	Yes	Yes
On-chain immutable records	—	—	—	Yes	Yes
Stored verification results	—	—	—	—	Yes
Economic Integration					
Trust-based payment splits	—	—	—	—	Yes
Transfer burns (reputation laundering defense)	—	—	—	—	Yes
Trust-proportional coin distribution	—	—	—	—	Yes
Negative bids for trust acceleration	—	—	—	—	Yes
Sybil Defenses					
Age-based derate	—	—	—	—	Yes
Cluster/graph analysis	—	—	—	Yes	Yes
Economic penalties	—	—	—	—	Yes
Double-Spend Handling					
Prevention (global consensus)	—	—	—	—	—
Detection + penalties	—	—	—	Yes	Yes
Currency weight scaling	—	—	—	—	Yes
Accusation Mechanism					
Signed assertions with evidence	—	—	—	—	Yes
Credibility from asserter's trust	—	—	—	—	Yes
Parameterized infraction severity	—	—	—	—	Yes

3. **Structured accusations are novel**: Prior systems use binary ratings or simple scores. Omerta's assertions include evidence hashes, asserter credibility weighting, and impact/-context multipliers that create appropriate gray areas.

4. **Currency weight for double-spend resolution is novel**: TrustChain detects double-spends but has no concept of scaling finality requirements to network connectivity. Omerta's "currency weight" allows graceful degradation across network conditions.

5. **Age as derate (not bonus) is a design choice**: While temporal defenses exist in prior work, Omerta specifically ensures age never *adds* trust—only removes a penalty. This prevents dormant identity accumulation attacks.

Why this matters: The theory developed in these prior systems is sound—the algorithms work, the math is correct, the insights are real. What was missing was economic integration and a compelling application domain. Omerta builds on this foundation by connecting trust to payments, making reputation have economic consequences, and targeting compute markets where machine-native trust measurement and verifiable delivery align naturally. The contribution is both theoretical (novel mechanisms listed above) and practical (a working system with real economic stakes).

2.2 Sybil Resistance

Douceur [7] proved that without a trusted central authority, a single adversary can present arbitrarily many identities indistinguishable from honest participants. This "Sybil attack" undermines any reputation system where influence scales with identity count.

This is a fundamental impossibility result that Omerta does not overcome. We can make Sybil attacks expensive, but not impossible—an approach shared by recent work like MeritRank [19], which explicitly "bounds" rather than prevents Sybil attacks. Honesty requires acknowledging this limitation.

Defenses fall into three categories:

Resource-based: Require each identity to demonstrate control of scarce resources—computational power [18], financial stake [17], or hardware attestation. Effective but expensive, and doesn't prevent well-resourced attackers.

Social-based: Leverage trust graph structure, noting that Sybil identities have sparse connections to honest nodes [27, 26, 5]. SybilGuard [27] and its improved successor SybilLimit [26] showed that social graph analysis can bound the number of accepted Sybils to $O(\log n)$ per attack edge. SybilInfer [5] further refined detection through statistical inference. Effective when the social graph reflects real relationships.

Temporal: Require identities to exist over time before gaining influence. This defense, explored in various systems including Freenet's Web of Trust, cannot prevent patient attackers who pre-create identities years in advance.

Omerta employs a hybrid approach: economic penalties (transfer burns), social detection (cluster analysis), temporal constraints, and computational investment. Crucially, effective aging only starts when identities begin contributing meaningfully to the network—simply creating an identity and waiting provides no benefit. An attacker cannot pre-create a pool of dormant identities; they must actually run compute sessions, which costs real resources.

Limitations we acknowledge: A well-resourced attacker who creates thousands of identities and actively matures them through real computational contribution over years will have thousands of mature identities. This attack requires substantial capital (to pay for compute during maturation) and patience. Omerta's defenses make this expensive in time and money, but do not make it impossible. We discuss residual attack surfaces in Section 8.9.

2.3 Blockchain Consensus and Its Limits

Bitcoin [18] introduced proof-of-work consensus, achieving Byzantine fault tolerance through computational cost. Subsequent systems explored alternatives: proof-of-stake [17], delegated proof-of-stake [**?**], practical Byzantine fault tolerance [3], and various hybrid approaches.

All these mechanisms solve the Byzantine Generals Problem: achieving agreement among distributed parties despite malicious actors. This requires $n \geq 3f+1$ nodes to tolerate f failures, with significant coordination overhead.

Budish [?] demonstrated fundamental economic limits of blockchain security: the recurring flow payments to miners must be large relative to the one-time stock benefits of attacking the system. This makes high-value applications expensive to secure. Gans and Gandal [10] extended this analysis to proof-of-stake, showing similar cost structures manifest as illiquid capital requirements. These analyses suggest blockchain may be over-engineered for applications that don't require global consensus.

Federated approaches occupy a middle ground. Stellar's Federated Byzantine Agreement and Ripple's trust lines allow nodes to choose which other nodes they trust for consensus, rather than trusting the entire network. However, recent analysis [**?**] reveals that FBA's "open membership" is limited in practice—"membership in the top tier is conditional on approval by current top tier nodes if maintaining safety is a core requirement." Despite this limitation, these systems influenced Omerta's design—the local trust computation is conceptually similar to choosing trusted validators, though Omerta applies this to reputation rather than consensus.

Layer-2 solutions (optimistic rollups, zk-rollups, sidechains) address blockchain scalability by moving computation off the main chain while inheriting security from L1 through various mechanisms (fraud proofs, validity proofs, or periodic checkpoints). Rollups achieve significantly higher throughput at lower cost, making them attractive for high-frequency applications. However, L2s introduce their own trust assumptions: most rely on centralized sequencers operated by single entities, creating censorship risk and single points of failure. Operators can withhold transaction data, preventing users from independently verifying state. The fragmented ecosystem of 140+ L2s requires bridges that have suffered billions in losses (Ronin $615M, Wormhole $320M). These are real trade-offs, not free scaling. For compute markets specifically, L2 solutions face an additional limitation: smart contracts cannot verify that off-chain computation was performed correctly, requiring oracles or optimistic schemes that reintroduce trust. L2s trade increased user trust for decreased transaction cost, but without managing that trust—the assumptions are implicit, scattered across sequencer operators, bridge security, and data availability guarantees. Omerta makes a similar trade-off but explicitly: by acknowledging our trust assumptions and managing them as first-class protocol concepts with reputation scores, decay mechanisms, and economic penalties, we aim to require a much smaller increase in user trust relative to the main chains.

Omerta sidesteps global consensus entirely. While individual transactions are bilateral (buyer-seller), trust has ripple effects: each interaction updates trust scores that propagate through the network. For the economy to function, the majority of interactions must be honest—which is why trust measurement exists in the first place. But this is different from requiring every node to agree on every transaction before it can proceed. By computing trust locally, Omerta eliminates the coordination overhead of global agreement while providing the security properties actually needed for compute rental. This is not superior to blockchain for applications requiring global consensus; it is a different trade-off appropriate for different applications.

Blockchain solved the Byzantine Generals Problem. Compute markets don't have Byzantine generals—they have landlords and tenants.

2.4 Secure Computation Approaches

The ultimate solution to untrusted compute would be **fully homomorphic encryption (FHE)** [11], which enables computation on encrypted data without decryption. FHE provides mathematical guarantees: the compute provider learns nothing about the data. However, current FHE implementations impose 1,000–1,000,000x overhead compared to plaintext computation [**?**], restricting practical use to narrow applications like encrypted database queries or private set intersection where the security requirement justifies the cost.

Trusted execution environments (TEEs) like Intel SGX [4] provide hardware-based isolation with much lower overhead (typically 5–30%), but require trusting the hardware manufacturer and have been vulnerable to side-channel attacks [**?**] including Spectre, Meltdown, and Foreshadow. TEEs are practical for applications where you trust Intel/AMD/ARM but not the cloud operator—a meaningful threat model, but not trustless.

Secure multi-party computation (MPC) [**?**] distributes computation across parties such that no single party learns the inputs. MPC overhead depends heavily on circuit complexity and number of parties—simple operations may run at 100–1000x slowdown, while complex operations can be millions of times slower. MPC is practical for specific high-value operations: private auctions, secure voting, threshold cryptography. It is not practical for general-purpose compute.

Where these approaches make sense: When the data is genuinely sensitive (medical records, financial data, trade secrets) and the computation is well-defined and bounded, the overhead may be justified. A hospital running private analytics on patient data, or banks computing fraud scores across institutions without sharing raw data—these are legitimate MPC/FHE use cases.

Where Omerta fits: Most compute workloads don't require cryptographic privacy guarantees. Running a build pipeline, training a model on public data, rendering video, processing batch jobs—these benefit more from cheap, available compute than from mathematical privacy proofs. Omerta targets this larger space, accepting trust requirements in exchange for practical performance.

2.5 Decentralized Computing

The altruistic distributed computing projects—BOINC [1], Folding@home [**?**], SETI@home—are among Omerta's closest ancestors. They demonstrated that volunteers would contribute compute resources for causes they believed in, without direct compensation. One purpose of Omerta is to make such projects easier: lowering barriers so that researchers without large budgets can access distributed compute for scientific work, citizen science, or public-benefit computation. We built awareness of this heritage into the protocol's design.

These systems face common challenges: verifying that claimed work was actually performed, preventing providers from delivering inferior resources, and detecting collusion. Omerta addresses these through continuous verification, trust-based payment splits, and statistical detection of manipulation patterns.

Why prior commercial systems struggled: Golem, iExec, and similar platforms have operated for years with limited adoption. We identify several contributing factors:

- *High barriers to entry*: Complex setup, specialized knowledge required, blockchain transaction fees for every operation
- *Wrong customer base*: Human developers demand reliability, low latency, and predictable pricing—exactly what unreliable home compute struggles to provide
- *Extractive economics*: VC funding requires returns; token economics require appreciation; platforms extract fees. These create friction that erodes the cost advantage of distributed compute

- *Mining overhead*: Proof-of-work and proof-of-stake consensus impose costs unrelated to compute delivery

Omerta's approach differs: the software is free and open source, requiring only a download to participate. There are no platform fees—providers keep what they earn (minus trust-based burns that fund network security). There is no preallocation of tokens, no founder stake, no investors requiring returns. Machine intelligence workloads are naturally fault-tolerant and retry-friendly, well-suited to unreliable infrastructure. And the "mining" is the useful compute itself, not a separate consensus mechanism.

What we expect to gain: We are transparent about benefits to Omerta's creators. By participating early, we gain access to cheaper compute for our own research and the ability to study trust systems on real networks with real people—something prior academic projects lacked. Early participants naturally accumulate higher trust scores through participation history, which may translate to economic advantages (lower transfer costs, priority matching). These benefits are available to all motivated early participants, not reserved for founders.

The gap in prior trust research: Decentralized compute is not just a blockchain problem—it's also a trust problem. Prior trust systems (EigenTrust, TidalTrust, FIRE) could have addressed the reputation challenges in compute markets, but they remained confined to academia. Why? They lacked economic integration and a compelling application. Omerta attempts to bridge this gap: taking the trust theory developed in the 2000s, connecting it to real economic mechanisms, and applying it to a market with genuinely unbounded demand.

2.6 Computational Economics and Mechanism Design

The design and validation of economic mechanisms increasingly relies on computational methods. Agent-based computational economics [23, 8] provides tools for studying emergent phenomena in complex markets where analytical solutions are intractable [**?**]. This approach has proven particularly valuable for mechanism design [**?**], where simulating agent behavior under proposed rules reveals edge cases and failure modes before deployment.

These computational methods have produced substantial real-world successes. Auction theory [**?**] informed the FCC spectrum auctions that raised over $200 billion while efficiently allocating scarce radio frequencies—a problem intractable without computational mechanism design. Matching market algorithms [**?**] now assign medical residents to hospitals (the NRMP match), students to schools, and kidneys to recipients, handling constraints and preferences that would overwhelm manual processes. Google's AdWords auction, designed using algorithmic game theory principles from the same literature we draw upon, processes billions of transactions daily. These are not laboratory curiosities but deployed systems handling high-stakes allocation problems.

Validation of agent-based models follows established practices [**?**]: parameter sensitivity analysis, comparison against theoretical predictions where available, and testing under adversarial conditions. These methods inform our simulation methodology in Section 7.

The trust propagation model relates to work on reputation mechanism design [**?**], particularly the challenge of eliciting honest feedback in the presence of moral hazard. The concept that trust mechanism overhead should scale with network uncertainty echoes transaction cost economics [**?**], which analyzes how institutions emerge to reduce uncertainty in exchange.

Omerta's economic simulations build on these foundations while extending them to a novel domain: trust-based compute markets where machine intelligence both creates demand and enables the trust mechanisms that make supply possible.

This document covers the theoretical foundation: Abstract, Introduction, and Related Work. For the full technical paper including System Architecture, Trust Model, Economic Mechanisms,

Attack Analysis, Simulation Results, and Discussion, see the complete Participation Verification paper.

References

[1] David P. Anderson. Boinc: A system for public-resource computing and storage. In *Proc. Grid*, 2004.

[2] Eric Budish. The economic limits of bitcoin and the blockchain. *Quarterly Journal of Economics*, 2024.

[3] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proc. OSDI*, 1999.

[4] Victor Costan and Srinivas Devadas. Intel sgx explained. In *IACR Cryptology ePrint Archive*, 2016.

[5] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Proc. NDSS*, 2009.

[6] Chrysanthos Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49(10):1407–1424, 2003.

[7] John R. Douceur. The sybil attack. In *Proc. IPTPS*, 2002.

[8] J. Doyne Farmer and Duncan Foley. The economy needs agent-based modelling. *Nature*, 460(7256):685–686, 2009.

[9] Eric Friedman and Paul Resnick. The social cost of cheap pseudonyms. *J. Economics & Management Strategy*, 10(2):173–199, 2001.

[10] Joshua S. Gans and Neil Gandal. More (or less) economic limits of the blockchain. CEPR Discussion Paper, 2019.

[11] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. STOC*, 2009.

[12] Jennifer Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, University of Maryland, 2005.

[13] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[14] iExec. iexec: Blockchain-based decentralized cloud computing, 2017.

[15] Audun Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer, 2016.

[16] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. WWW*, 2003.

[17] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, 2012.

[18] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[19] Bulat Nasrulin and Georgy Ishmaev. Meritrank: Sybil tolerant reputation for merit-based tokenomics. *arXiv:2207.09950*, 2022.

[20] Pim Otte, Martijn de Vos, and Johan Pouwelse. Trustchain: A sybil-resistant scalable blockchain. *Future Generation Computer Systems*, 107:770–780, 2020.

[21] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[22] Steven Tadelis. Reputation and feedback systems in online platform markets. *Annual Review of Economics*, 8:321–340, 2016.

[23] Leigh Tesfatsion and Kenneth L. Judd. *Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics*. North-Holland, 2006.

[24] The Golem Project. The golem whitepaper. https://golem.network/, 2016.

[25] Li Xiong and Ling Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE TKDE*, 16(7):843–857, 2004.

[26] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Proc. IEEE S&P*, 2008.

[27] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *ACM SIGCOMM*, 2006.

[28] Runfang Zhou and Kai Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel and Distributed Systems*, 18(4):460–473, 2007.