TECHNICAL WHITE PAPER · MLXMESH

mlxMesh / Open Inference Mesh: A Federated Protocol for Privacy-Tiered, Measured-Accountability AI Compute Across Wide-Area Networks

A federated protocol for privacy-tiered, measured-accountability AI compute across wide-area networks.

WAN Federation | MoE Sharding | Ed25519 | Secure Enclave | Open Protocol

ABSTRACT

Open Inference Mesh (OIM), branded as mlxMesh, is a distributed inference protocol and coordination layer that federates geographically dispersed AI compute clusters — each individually managed by tools such as exo — into a single, routable inference fabric spanning the wide-area network (WAN). Unlike single-cluster LAN inference systems, OIM addresses the fundamental constraints of WAN inference: multi-hundred-millisecond inter-hop latency, node identity fraud, declared-vs-measured capability divergence, and the economic sustainability of a heterogeneous contributor pool. The protocol implements dual-lane routing (interactive fast lane and batch background lane), Mixture-of-Experts (MoE) expert-sharding as the sole WAN-viable parallelism strategy for large models, Ed25519-derived cryptographic node identity, division-order accounting with measured rather than declared resource lines, and a sensitivity tier system with Secure Enclave attestation for high-confidentiality payloads. iOS/iPadOS devices participate as coordination and security layer nodes rather than compute nodes, extending the mesh's privacy guarantees without adding inference latency. This paper surveys the technical foundations of wide-area distributed inference, analyzes the economic design of the credit system, and presents OIM as an architecture for the next phase of the personal AI infrastructure movement.

01 Introduction

The distributed inference movement has matured rapidly. Systems like exo [1], llama.cpp server [2], and vLLM [3] enable multi-node LAN clusters to serve large language models across pooled consumer hardware. A user with four M-series Macs can collectively host a 70B parameter model. A research lab with a rack of GPU workstations can run a competitive open-weight model without cloud dependency.

Yet these systems share a geographic assumption: all participating nodes are on the same LAN, connected by 1–10 Gbps links with sub-millisecond latency. This assumption is load-bearing. The dominant multi-node inference strategy — pipeline parallelism, where sequential layers are distributed across nodes — requires inter-node activation transfers at every layer boundary. For a 70B transformer with 80 layers, each token's forward pass crosses the inter-node link 79 times. At 10 Gbps LAN speeds with <1 ms RTT, this is tractable. At WAN speeds — 100–1000 Mbps with 20–150 ms RTT — it is not: the propagation delay alone would cap throughput to fractions of a token per second, orders of magnitude below practical utility.

Open Inference Mesh is designed for the next level of the hierarchy: coordinating multiple LAN clusters across the internet into a federated mesh. It does not replace LAN inference systems; it sits one level above them, routing inference jobs to the right cluster rather than within a cluster.

The design is governed by a set of non-negotiable constraints derived from the failure modes of previous federated compute networks:

1. **No WAN pipeline-parallel dense model sharding.** The only WAN-viable strategy for large model serving is MoE expert sharding, where individual experts are resident on specific nodes and token dispatch routes the token to the correct expert's host rather than passing activations through a serial pipeline.
2. **No native token.** The Helium Network's HNT token [4] demonstrated how native-token incentive systems create speculative dynamics that decouple the token's market value from the network's actual utility. OIM uses an off-protocol, non-tradeable credit system that represents earned or granted compute capacity — a closed barter network, not a currency.
3. **Measured, not declared, performance accounting.** Node operators are economically motivated to over-declare their capabilities to attract job dispatch. OIM's verification layer (probabilistic re-dispatch spot checks, statistical baseline drift detection, tier claim verification) catches fraud and degrades the reputation of dishonest nodes.
4. **Ed25519 node identity.** Node IDs derived from cryptographic public keys cannot be operator-chosen or replicated, eliminating Sybil attack vectors in the directory.

---

02 Background: The Wide-Area Inference Problem

~35×

LOWER LATENCY — MOE SHARDING VS. DENSE PIPELINE

Per-forward-pass latency over a cross-country WAN hop for a 70B-parameter model: MoE expert-sharding (single dispatch + mix, ~45 ms) vs. dense pipeline-parallelism (79 inter-layer hops × 20 ms RTT, ~1,580 ms).

2.1 Latency Budgets for Autoregressive Generation

Autoregressive large language model generation produces one token per forward pass. For an interactive application, the first-token latency (time-to-first-token, TTFT) and inter-token latency (ITL) determine the user experience. Accepted targets for interactive use are TTFT < 500 ms and ITL < 100 ms (corresponding to > 10 tokens/second) [5].

In a pipeline-parallel WAN configuration, each forward pass through an N-node pipeline requires N-1 inter-node data transfers. With 20 ms minimum RTT between US coasts and a 70B model requiring 80 layer boundaries, the propagation overhead alone is 79 × 20 ms = 1,580 ms — exceeding the acceptable TTFT by 3×, before any compute time. This is not a bandwidth problem; it is a physics problem. Light-speed propagation cannot be optimized away.

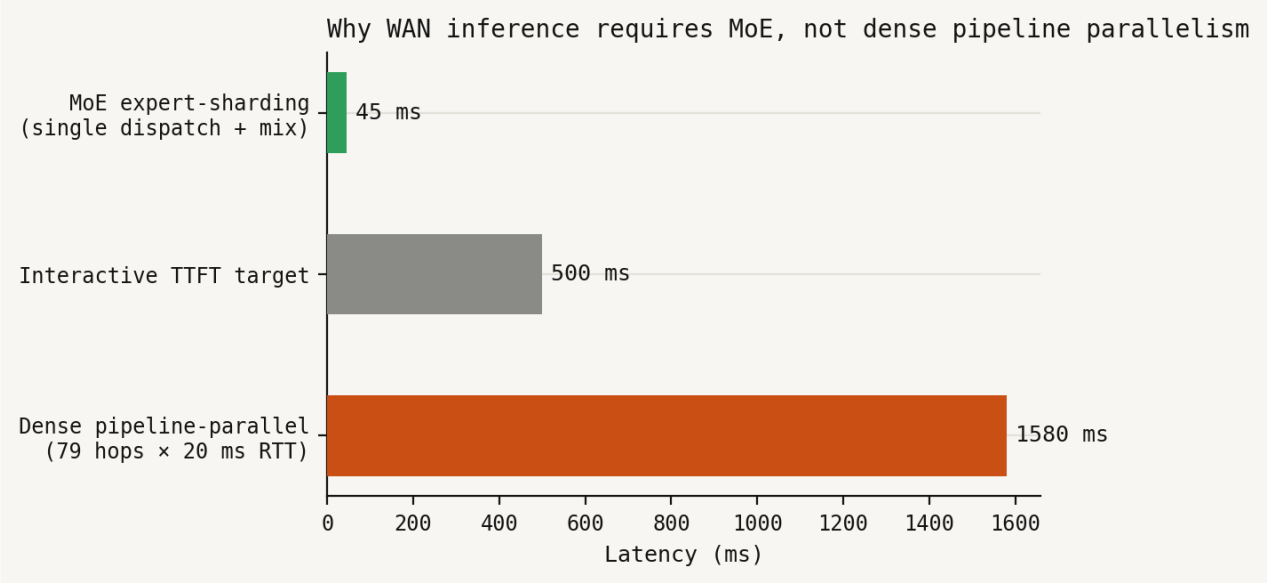

Figure 1. Dense pipeline-parallel inference cannot clear an interactive TTFT budget over WAN distances; MoE expert-sharding can.

OIM's response: **WAN is the wrong level for inter-layer activation passing.** Instead, OIM routes whole jobs (complete inference requests) to a single LAN cluster that can serve the model independently, or to an MoE configuration where token routing dispatches between experts — not between sequential layers.

2.2 Mixture-of-Experts as WAN Infrastructure

The MoE architecture [6] provides a structurally different parallelism opportunity: expert independence. In a dense transformer, layer N+1 depends causally on layer N's output, making pipeline parallelism sequential by construction. In an MoE transformer, experts at the same layer are independent — routing selects which expert(s) process each token, and unselected experts produce no output. This independence means that token routing between experts does not create a sequential activation chain.

Under OIM's MoE expert sharding model [7]:

- Each node hosts a subset of the model's expert modules.
- The router (running on the coordinating node) computes expert selection for each token.
- The token (or its key/value vectors, depending on architecture) is dispatched to the hosting node.
- Expert outputs are collected and mixed on the coordinator.

- The residual stream update is communicated back.

This reduces WAN data transfer per token from one activation tensor per layer (dense pipeline) to one small token dispatch plus one expert output per layer — a dramatically smaller bandwidth demand that is feasible under typical WAN conditions.

2.3 Cryptographic Node Identity

Distributed systems that rely on node self-reporting for identity are vulnerable to Sybil attacks [8]: a single malicious actor creates many fake node identities to gain disproportionate influence over routing, reputation, or incentive systems. OIM derives each node's identity as `SHA-256(Ed25519_public_key)[:32]` — a cryptographic commitment to the node's public key. Possession of the private key proves identity; the public key proves the derivation. Operators cannot choose their node ID, and creating N fake identities requires generating N independent keypairs — no more economical than operating N honest nodes.

---

03 Protocol Architecture

3.1 Three-Layer Hierarchy

OIM's network is organized in three tiers:

```
Global Directory (librarian)
         |
Pod Coordinators (1 per geographic region)
         |
Node Agents (wrapping local Exo clusters)
```

Global Directory. A resolver service that maps (model_id, sensitivity_tier) queries to available pod coordinators. Currently centralized (M4 milestone); federated across multiple directory servers is the M7 target. The directory maintains a TTL-based pod digest store and implements one-hop gossip for bootstrap-pair synchronization — avoiding the Sybil and eclipse attack risks of DHT designs while achieving redundancy.

Pod Coordinators. Regional coordinators that maintain the node registry for their geographic pod. Coordinators implement the fast-lane routing (resolver-routed, low-latency interactive dispatch) and background-lane scheduling (sticky-session, recurring-job assignment). The OpenAI-compatible API endpoint (`POST /v1/chat/completions`) is exposed by the coordinator.

Node Agents. Processes running on each contributing machine. The agent registers with its pod coordinator via Ed25519-signed manifests, reports benchmarked capabilities (not declared), executes dispatched jobs through the local exo cluster, reports outcomes, and participates in the verification layer.

3.2 Dual-Lane Routing

Fast Lane. Interactive inference jobs with no recurrence specification. The coordinator uses measured TPS (transactions per second, from the verification layer's benchmark records) as the primary routing signal, with secondary weighting by sensitivity tier availability. Jobs are dispatched directly to the best-scoring available node. Response streaming is supported via Server-Sent Events (SSE) passthrough from exo → node agent → coordinator → client.

Background Lane. Recurring or batch inference jobs, identified by a `RecurrenceSpec` in the job submission. The scheduler assigns these to a primary node plus N backup nodes (sticky assignment), ensuring that periodic jobs find their session context on the same node across invocations. Background lane jobs receive a 50% cost discount relative to fast lane at equivalent sensitivity tiers, reflecting their lower latency requirements.

Routing invariants: - Fast-lane jobs MUST NOT carry a `RecurrenceSpec` - Background-lane jobs MUST carry a `RecurrenceSpec` - A node handling a fast-lane job under capacity constraint MUST refuse rather than degrade (enforced by `RefuseIfConstrained`)

3.3 Sensitivity Tiers and Secure Enclave Attestation

Three sensitivity tiers govern data handling requirements:

LOW. Standard inference. No special handling. Cost multiplier: 0.5×.

MODERATE. Default for general business use. Transport encrypted, standard node screening. Cost multiplier: 1.0×.

HIGH_REQUIRES_ATTESTATION. Payloads require processing in a hardware-backed secure environment. On Apple Silicon, this gates dispatch to nodes that can attest Secure Enclave availability. The coordinator's `VerifyTierClaim` function checks the node's submitted benchmark against its claimed tier signature; fraudulent tier claims are detected and flagged. Cost multiplier: 3.0×.

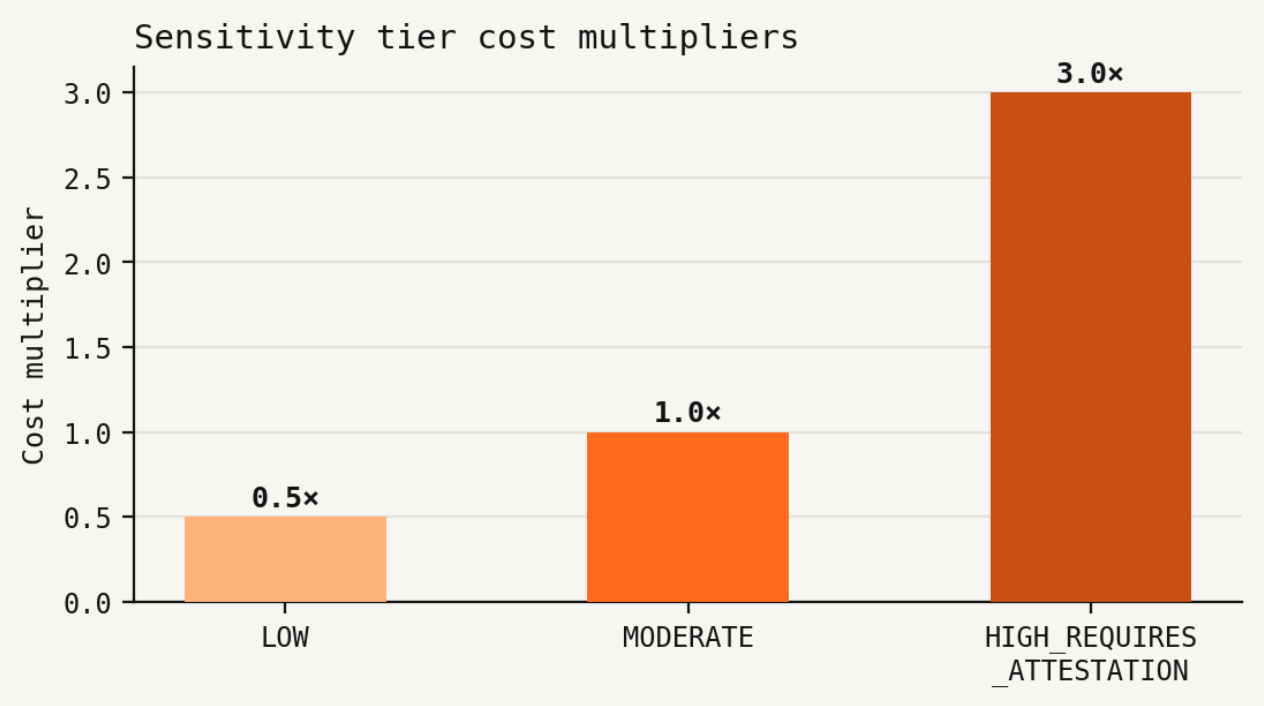

Figure 2. Cost multipliers by sensitivity tier — confidentiality has a measurable, transparent price.

The iOS coordination layer extends these tiers in a different dimension: iOS/iPadOS devices perform on-device sensitivity classification of payloads and host encrypted payload pointers. This means that even the coordinator never sees the plaintext of a HIGH-tier payload — the iOS device holds the encryption key and only releases the pointer to a vetted, attested node. This is an additive privacy guarantee: it does not change routing, but it means that a compromised coordinator cannot extract HIGH-tier payload content.

3.4 Outbound Work-Pull (NAT Traversal)

The standard home network topology — a consumer router performing NAT, with no port forwarding — would prevent inbound connections to node agents, breaking a hub-and-spoke dispatch model. OIM's default mode is outbound work-pull: the node agent opens a persistent outbound long-poll connection to the coordinator. The coordinator delivers jobs through this connection rather than dialing into the node. No port forwarding, NAT traversal (STUN/TURN), or UPnP is required.

Push mode (coordinator dials the node) is available for LAN deployments, simulated clusters, and operators who have configured inbound connectivity. The Docker simulation cluster uses push mode for deterministic testing. The distinction is surface through the `port_mapping` field in the node's status report: `"pull"` for outbound work-pull, `"manual"` for explicit endpoint.

04 Economic Design

4.1 Division-Order Accounting

The settlement layer implements division-order accounting: each inference job generates a multi-line resource record that separates different resource types (compute tokens, coordination, bandwidth) rather than collapsing them into a single credit debit. The motivation is auditability: when a settlement dispute arises, the specific resource lines in contention are identifiable.

Critically, grant balances and earned balances are maintained as separate ledger entries and must never be collapsed. A node with $50 in startup grants and $50 in earned credits has a specific accounting relationship to the network — its startup grant liability is tracked by the treasury, and its earned credits represent real delivered compute. Merging these would obscure the network's true grant liability.

4.2 The Credit Model: Closed Barter, Not Currency

OIM credits are explicitly not a tradeable currency. They cannot be converted to fiat, exchanged on secondary markets, or used outside the OIM network. This design is motivated by the failure mode of incentive-token networks [9]: when a network's native token acquires speculative market value, holding the token (speculation) competes with using it (network utility), creating dynamics where high token prices paradoxically reduce actual network usage.

OIM credits are a pure barter medium: they represent a claim on compute capacity contributed to the network, usable only to consume compute capacity from the network. The house edge (25% of consumer cost kept by the treasury) funds coordination rewards, startup grants, and availability rewards — a self-funding mechanism that does not require external capital or token appreciation.

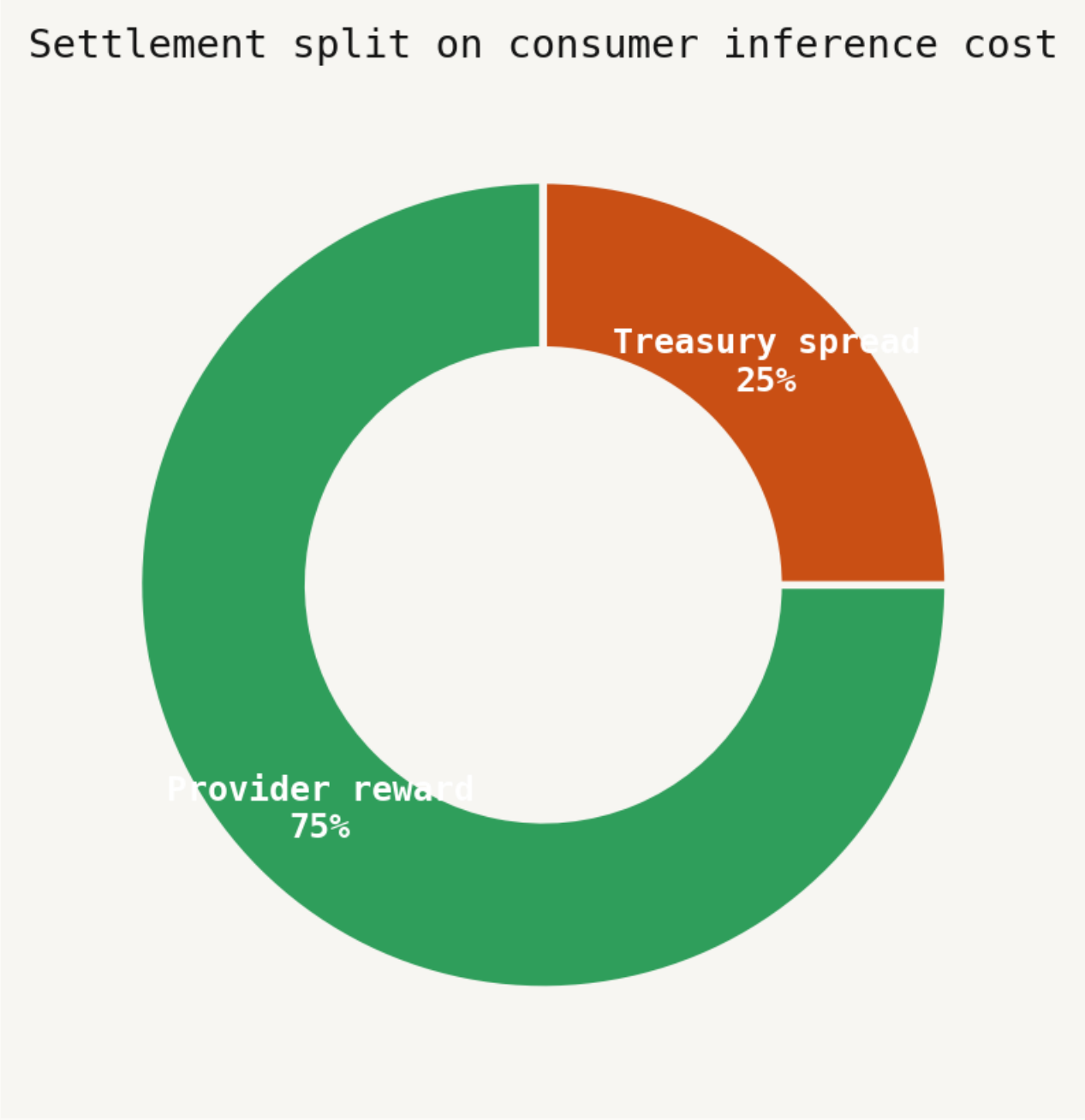

Figure 3. Settlement split on consumer inference spend: 75% flows to the delivering node, 25% funds the treasury.

4.3 Startup Grants and Grant Decay

New nodes receive a startup grant tied to their verified capacity (not declared capacity). The grant is sized to enable meaningful participation before the node has accumulated earned credits through actual job delivery. Grant multipliers decay as a node accumulates earned credits, converging to zero as earned capacity comes to dominate — preventing the network from maintaining a permanent liability to nodes that never attract real job traffic.

The decay schedule is deterministic and published, eliminating the information asymmetry between the network and new contributors about the expected startup grant trajectory.

4.4 Bootstrapping Economics: Activity Discount and Availability Rewards

Two mechanisms address the cold-start problem for new deployments:

Activity discount. When network queue backpressure is below 40%, the consumer-side cost is discounted toward the provider's reward — the treasury's margin compresses to zero. At a fully idle network, consumers pay exactly what providers earn. This prevents the negative feedback loop where a new user's startup grant runs out on a few expensive queries before the network has enough real traffic to justify the cost.

Availability rewards. An opt-in coordinator flag (`--availability-reward`) dispatches small synthetic inference probes to idle nodes on a randomized ~10-minute interval. The probe uses the standard dispatch path (not a synthetic shortcut) and pays the node its standard provider reward for the delivered tokens. At low network load, probe rounds scale up to 15 nodes per round; at high load (>40% backpressure), probes are skipped entirely.

These mechanisms are designed to smooth the bootstrapping curve without creating persistent subsidies that outlive the bootstrapping phase — both naturally decay as real consumer traffic grows.

4.5 Economic Sustainability Analysis

The economic model is designed around three equilibria:

Short-run (bootstrapping). High startup grants and availability rewards attract early nodes. Activity discounts attract early consumers. The treasury runs a deficit funded by inflationary grant issuance (grants are minted, not drawn from the treasury).

Medium-run (growth). Real job traffic generates provider rewards and treasury revenue. Startup grant issuance to new nodes is offset by decay on existing nodes' grants as their earned credits grow. The treasury accumulates margin from real consumer traffic.

Long-run (steady state). Grant issuance converges to zero as all active nodes are fully earned-credit-funded. Treasury revenue from the 25% spread funds coordination rewards and any remaining network costs. Consumer cost and provider reward are stable at the undiscounted matrix values.

This trajectory is structurally similar to Bitcoin's block reward halving schedule [10] but without the deflationary pressure: credits are consumed by inference and do not accumulate indefinitely, preventing supply-side inflation.

05 Verification and Trust

5.1 The Declared vs. Measured Gap

Any network that pays nodes per delivered compute is vulnerable to over-declaration: a node claims 100 FLOPS capability but only delivers 10. OIM's verification layer closes this gap with three mechanisms:

SpotCheckFastLane. A probabilistic re-dispatch of a fraction of completed fast-lane jobs to an independent node. The re-dispatched result is compared (content-length consistency check) to the original. Consistent discrepancies flag the original node for reputation downgrade.

StatisticalBaselineCheck. Per-node TPS distributions are maintained across all submitted benchmark results. A 3-sigma deviation from a node's baseline triggers a flag — catching nodes that benchmark well once but deliver poorly in production.

VerifyTierClaim. A node's claimed tier (e.g., HIGH_REQUIRES_ATTESTATION) is verified against its submitted benchmark signature. The benchmark is structured such that genuine Secure Enclave access produces a computationally distinct signature from a software-only emulation.

5.2 Reputation and Routing

Reputation scores from the verification layer feed directly into the fast-lane routing score (TPS × reliability_weight). A node that accumulates verification flags sees its routing weight decrease, reducing dispatched jobs and earned credits until the node corrects its behavior or is eventually deprioritized out of the active pool.

06 Privacy Architecture

6.1 iOS as Security Layer

iPhones and iPads participate in OIM as coordination and classification nodes rather than compute nodes. Their roles:

- **On-device sensitivity classification.** The iOS ML stack (Core ML, NaturalLanguage) classifies incoming payload text for sensitivity tier before routing. This classification runs entirely on-device with no data transmitted to the coordinator until the tier is established.
- **Encrypted payload hosting.** For HIGH-tier payloads, the iOS device encrypts the payload and hosts the ciphertext, transmitting only a signed pointer to the coordinator. The coordinator resolves this pointer to

an attested node; the iOS device releases the decryption key directly to the attested node over an authenticated channel. The coordinator at no point holds the plaintext.

- **Additive guarantee.** The mesh routes correctly with zero iOS coordination devices present. Their presence adds a privacy guarantee without creating a single point of failure.

6.2 Portable Wallet Identity

Each user's credit balance is associated with an Ed25519 wallet key, stored in iCloud Keychain and recoverable from a seed phrase. The wallet is account-linked to a server-side ledger balance — it is not a blockchain wallet and does not hold funds on-chain. Its role is authentication: proving to the coordinator that this device is the legitimate holder of a specific account's balance. Credits consolidate across a user's devices (their Mac node, their iPad coordination device, their iPhone) to a single account.

---

07 Commercial and Research Impact

7.1 Enabling the Personal AI Infrastructure Ecosystem

The rapid growth of open-weight models (Llama 3 [11], Mistral [12], Mixtral [13], Qwen [14]) combined with quantization techniques (GGUF [15], MLX [16]) has created a category of "personal AI infrastructure" — individuals and small teams running inference on owned hardware for privacy, cost, or customization reasons. OIM provides the federation layer that allows these isolated deployments to pool capacity and serve each other, creating network effects proportional to the aggregate capacity of all participants.

7.2 Sovereign AI for Regulated Industries

Industries with data residency requirements (healthcare, finance, government) need AI inference that guarantees data never leaves specified jurisdictions. OIM's geographic pod structure and sensitivity tier routing enable geographic-constrained inference: a job marked with a US-only geographic constraint routes only to pods within that jurisdiction. Combined with HIGH-tier Secure Enclave attestation, this provides both residency and confidentiality guarantees.

7.3 Competitive Cost Structure

At scale, OIM's consumer-side pricing (1 credit per 1,000 output tokens at fast/moderate) translates to a cost structure competitive with major cloud inference providers — but with provider rewards flowing to the mesh participants rather than cloud provider margins. The 75% provider reward fraction is structurally higher than the marginal revenue passed to compute suppliers by cloud providers (who typically capture 60–80% as infrastructure margin [17]).

---

08 Current Status and Roadmap

8.1 Milestone Status

Milestone	Status	Description
M1	Complete	Node agent: manifest, governor, benchmarking, CLI
M2	Complete	Pod coordinator: fast-lane and background-lane routing, node registry
M3	Complete	Verification layer: spot check, baseline drift, tier verification
M4	Complete	Centralized directory: pod store, gossip, resolver
M5	Complete	Settlement: division-order accounting, Ed25519-signed records, ledger
M6	Complete	MoE expert-shard planner: proportional assignment, routing decision, imbalance detection
M7	In progress	Federated directory: multi-librarian resolver

75 tests passing across all completed milestones.

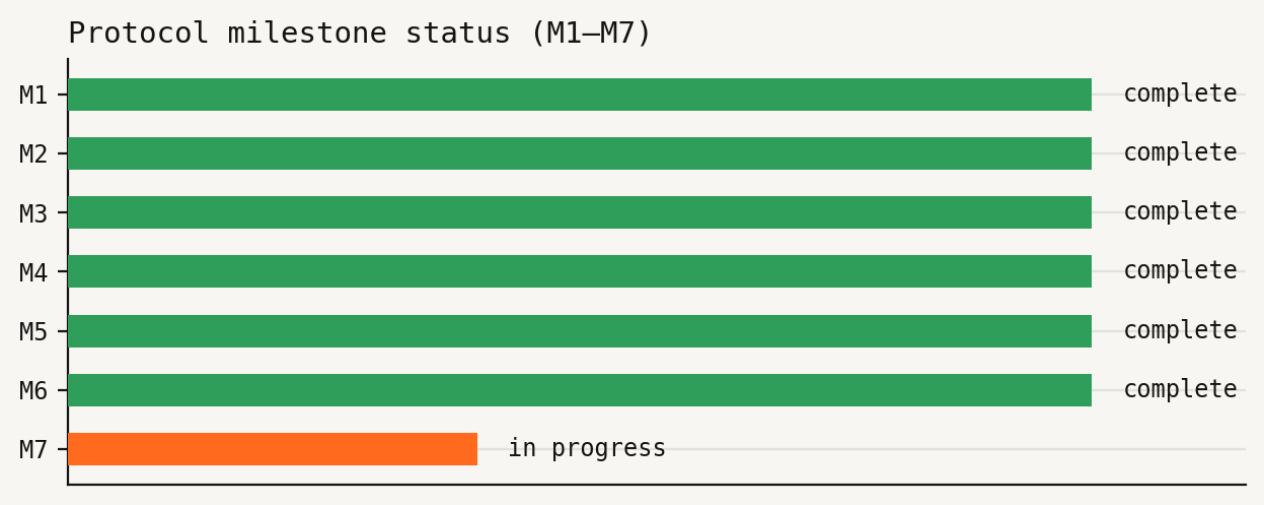

Figure 4. Protocol milestone status — six of seven milestones complete, with federated directory (M7) in progress.

8.2 Open Research Questions

DHT directory. A distributed hash table would eliminate the centralized directory — but DHT-based peer discovery is vulnerable to Sybil and eclipse attacks [18]. OIM's architecture explicitly defers DHT adoption until mitigation strategies (proof-of-work node registration, reputation-gated DHT participation) are mature.

MoE live dispatch wiring. The M6 MoE planner produces expert assignments and routing decisions but is not yet wired into live job dispatch. Completing this connection requires either a model with an externally accessible expert execution API or integration with an MoE-serving exo fork.

Token streaming in pull mode. Outbound work-pull nodes currently receive jobs in buffered request/response mode. SSE streaming passthrough requires a bidirectional channel that the current pull mailbox design does not support; a fast-follow protocol extension is planned.

09 Conclusion

Open Inference Mesh demonstrates that the technical and economic design problems of wide-area distributed AI inference are solvable without blockchain-style speculation, without WAN-incompatible parallelism strategies, and without compromising the privacy guarantees that make self-hosted AI valuable in the first place. The protocol's three core innovations — MoE-exclusive WAN parallelism, measured accountability through cryptographic verification, and a closed barter credit system — address the failure modes of previous federated compute networks directly. As the personal AI infrastructure ecosystem grows and open-weight models continue to approach the capability frontier of closed commercial models, a federation layer that connects isolated LAN clusters into a globally routable mesh becomes essential infrastructure. mlxMesh is designed to be that layer.

// References

[1] exo-explore, "exo: Run your own AI cluster at home with everyday devices," GitHub Repository, 2024. https://github.com/exo-explore/exo

[2] Gerganov, G., et al., "llama.cpp," GitHub Repository, 2023. https://github.com/ggerganov/llama.cpp

[3] Kwon, W., et al., "Efficient Memory Management for Large Language Model Serving with PagedAttention," Proceedings of SOSP 2023.

[4] Helium Foundation, "Helium Network Economics," 2023. https://docs.helium.com/helium-tokens/hnt/

[5] Agrawal, A., et al., "SARATHI: Efficient LLM Inference by Piggybacking Decodes with Chunked Prefills," arXiv:2308.16369, 2023.

[6] Shazeer, N., et al., "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," ICLR 2017.

[7] Lepikhin, D., et al., "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding," ICLR 2021.

[8] Douceur, J.R., "The Sybil Attack," Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS), 2002.

[9] Buterin, V., "On Medium-of-Exchange Token Valuations," Ethereum Blog, 2017. https://vitalik.eth.limo/general/2017/10/17/moe.html

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

[11] Meta AI, "Llama 3: Open Foundation and Fine-Tuned Chat Models," Meta AI Blog, April 2024. https://ai.meta.com/blog/meta-llama-3/

[12] Jiang, A.Q., et al., "Mistral 7B," arXiv:2310.06825, 2023.

[13] Jiang, A.Q., et al., "Mixtral of Experts," arXiv:2401.04088, 2024.

[14] Qwen Team, "Qwen Technical Report," Alibaba Cloud, 2023. https://qwenlm.github.io/

[15] Gerganov, G., "GGUF: A File Format for Large Language Models," GitHub, 2023. https://github.com/ggerganov/ggml/blob/master/docs/gguf.md

[16] Hannun, A., et al., "MLX: Efficient and flexible deep learning on Apple silicon," GitHub, 2023. https://github.com/ml-explore/mlx

[17] Cobbe, K., and Sutton, J., "The Economics of Cloud GPU Compute," Stanford HAI Working Paper, 2023.

[18] Singh, A., et al., "Eclipse Attacks on Overlay Networks: Threats and Defenses," IEEE/ACM Transactions on Networking, vol. 14, no. SI, 2006.

[19] Bernstein, D.J., et al., "High-speed high-security signatures," Journal of Cryptographic Engineering, vol. 2, no. 2, pp. 77–89, 2012. (Ed25519)