Infernet: A Peer-to-Peer Distributed GPU Inference Protocol

Abstract

Infernet is an open-source, peer-to-peer protocol enabling individuals to contribute spare GPU cycles for inference workloads and earn compensation in return. Clients can rent distributed GPU compute power from the network, scaling inference jobs across multiple provider nodes with security, verification, and trustlessness.

Introduction

The demand for AI inference compute continues to rise, with centralized providers often expensive, inflexible, or limited by region. Infernet aims to democratize access to GPU inference by allowing anyone with spare GPU cycles to contribute to a global, distributed network.

Providers earn tokens for completed jobs, while clients gain scalable, on-demand inference power. Inspired by SETI@home and BOINC but designed for commercial viability, Infernet focuses on trustless verification, efficient job distribution, and seamless multi-node inference aggregation.

Core Architecture

1. Node Roles

- **Provider Nodes**: Contribute GPU compute, execute inference tasks in secure containers.
- **Client Nodes**: Submit inference jobs and pay for compute time.
- **Aggregator Nodes**: Coordinate multi-node jobs, manage task distribution, verification, and result assembly.

2. Discovery Layer

- Decentralized discovery using libp2p or Kademlia DHT.
- Nodes announce availability, GPU specs (VRAM, CUDA cores, bandwidth), and reputation score.

3. Job Submission Layer

- Clients submit containerized workloads with metadata:
  - Inference model (container or WASM)
  - Input payload (split into shards if needed)

    - Payment offer and deadline
- Jobs are assigned either directly (small tasks) or via aggregators (multi-node coordination).

4. Multi-Node Aggregation

- Aggregator splits large input datasets into shards.
- Assigns shards to multiple provider nodes.
- Collects partial results, verifies correctness, merges them.
- Returns final output to client.

5. Verification & Proof of Work

- Redundant computation for result verification (minimum two nodes receive the same shard).
- Validator sampling: small "test jobs" with known outcomes.
- Future zkML proof integration for trustless computation verification.

6. Payment Layer

- Micro-payments facilitated via:
    - Lightning Network (BTC)
    - Polygon or Solana stablecoins
    - Custom Infernet Token (optional)
- Escrowed by aggregators, released upon result verification.

7. Reputation System

- Track each node's:
    - Uptime
    - Task completion rate
    - Latency benchmarks
- Slashing and bans for dishonest or slow nodes.

8. Security Considerations

- Containerized execution with sandboxing.
- Encrypted input payload support.
- Support for secure enclaves (Intel SGX / AMD SEV).

Economic Model

- **Client pricing**: Pay per job or per inference request, determined by compute complexity.
- **Provider earnings**: Earn tokens proportional to work done, speed, and reputation.

- **Aggregator fees**: Aggregators earn a percentage (0.5–2%) for coordination and verification.

Potential Use Cases

- AI inference for startups needing affordable compute.
- Decentralized inference APIs.
- On-demand compute for research institutions.
- Private inference jobs for enterprise use.
- Distributed training of custom AI models.

Distributed Model Training

In addition to inference, Infernet Protocol enables distributed training of custom AI models across the network. This capability has the potential to democratize AI model development and challenge the dominance of large AI providers.

Training Architecture

- **Federated Training**: Coordinate model training across multiple provider nodes.
- **Gradient Aggregation**: Efficiently combine model updates from distributed training runs.
- **Checkpoint Management**: Secure storage and versioning of model checkpoints.
- **Hyperparameter Optimization**: Distributed search for optimal model configurations.

Economic Incentives

- **Higher Compensation**: Training jobs typically offer higher rewards than inference due to increased resource utilization.
- **Long-running Relationships**: Providers can commit to extended training campaigns for stability.
- **Dataset Contribution**: Providers can optionally contribute data for improved training, with additional compensation.

Security and Privacy

- **Differential Privacy**: Built-in mechanisms to protect training data privacy.
- **Secure Aggregation**: Cryptographic protocols to combine model updates without revealing individual contributions.

- **Verifiable Computation**: Ensure training steps are performed correctly without revealing the model architecture.

Democratizing AI Development

- **Accessible Computing**: Startups and researchers can train large models without massive capital investment.
- **Specialized Models**: Train domain-specific models that may be overlooked by major AI providers.
- **Community Ownership**: Enable collaborative training of open-source models owned by the community.

Roadmap

- **Phase 1**: MVP for single-node inference jobs (Q2 2025)
- **Phase 2**: Aggregator and multi-node inference support (Q3 2025)
- **Phase 3**: Reputation, verification, and slashing system (Q4 2025)
- **Phase 4**: Payment integration (BTC Lightning, Polygon, Solana) (Q4 2025)
- **Phase 5**: zkML Proof research & implementation (2026)

Conclusion

Infernet offers a decentralized solution to the growing demand for inference compute power. By pooling spare GPU cycles globally and ensuring verification, scalability, and trustlessness, Infernet will create a new marketplace for distributed inference services.

Open-Source Repository

GitHub Repository: https://github.com/profullstack/infernet-protocol

Domain

Visit us at: https://infernet.tech and https://infernetprotocol.com - these domains will so

Contact

For collaboration and contributions: protocol@infernet.tech