A distributed inference network

Silicon Network runs large open models on a fleet of Apple Silicon, behind one OpenAI-compatible endpoint. A short overview of what it is, how it works, and what is built today. This reflects the current design, not final benchmarks.

1. The problem

Frontier inference runs in a handful of data centers owned by a handful of companies. That concentration sets the price, decides which models you may call, and puts your data on someone else's hardware. Meanwhile a large amount of capable compute sits idle: workstations, servers, and Apple Silicon machines with more memory and bandwidth than most workloads use. The gap is coordination, not silicon.

2. How it works

Three parts make up the system. The **engine** runs on each node: it loads models, shards them across peers on the local network, and serves an inference endpoint. The **gateway** is the public entry point: it authenticates API keys, routes each request to a node that holds the requested model, meters usage, and streams the response back. The **control plane** holds accounts, keys, deployments, node registrations, and metering.

The API speaks the OpenAI, Claude, and Ollama wire formats (streaming, tools, and vision included), so existing clients point at Silicon Network by changing a base URL. Large models are split across nodes with tensor and pipeline parallelism, so a set of ordinary machines can serve a model none of them could hold alone.

3. What is built today

The engine, the web dashboard, the control plane, and the gateway are built and run. Payment works by stablecoin on an EVM chain or by card. Still ahead: global node federation across regions, the scheduler and marketplace that match a request to the best node, operator credits, and verifiable inference. We describe those as design, and will not claim them before they are real.

4. Economics

Pricing is usage-based: you pay per token for the model you actually call, with no seats and no idle clusters to rent. On the other side, operators who contribute compute earn credit for the work their nodes perform. The meter is one ledger for both sides: real requests, real tokens, recorded once.

5. Trust

A distributed network raises a fair question: how do you know a node ran the model you asked for, and ran it honestly? Today the answer is operational: node identity, health, and reputation. The direction is cryptographic: a proof that a specific model produced a specific output, so trust does not depend on knowing the operator. It is the hardest part of the roadmap.