Tangle Cloud

The Decentralized Compute Layer for AI

Product Vision & Infrastructure

Tangle Foundation

January 2025

Abstract

Tangle Cloud provides decentralized compute infrastructure for AI workloads. This document describes the product vision, technical infrastructure, and development roadmap. The core products—the sandbox runtime and the agentic workbench—demonstrate how decentralized infrastructure can match centralized alternatives in capability while exceeding them in accountability and value distribution. We detail the isolation technologies, deployment models, pricing mechanisms, and the path toward a fully decentralized compute economy.

Contents

1 Vision: Decentralized AI Infrastructure

The deployment of increasingly capable AI systems demands infrastructure that can scale while maintaining accountability. Current options impose constraints that limit innovation and concentrate value.

Centralized cloud providers work well but concentrate power. They set prices unilaterally. They define terms of service. They choose which customers to serve. They capture the economic value their platforms generate. For AI infrastructure underpinning significant economic activity, this concentration creates systemic risk.

Tangle Cloud offers an alternative. Independent operators compete to provide compute. Prices emerge from markets rather than corporate decisions. Economic value distributes to participants rather than concentrating in platform owners. No single entity can deny service or change terms unilaterally.

1.1 The Product-Protocol Connection

Every interaction with Tangle Cloud products generates economic activity that accrues to network participants.

A customer uses a product (sandbox runtime, workbench, or third-party application). The product requests services from operators. The customer pays in tokens. Payments split among developers, operators, and the protocol.

Operators provide compute. They stake tokens, run infrastructure, and earn fees. The more services they provide reliably, the more they earn. The more stake they accumulate, the more services they can serve.

Developers create blueprints defining service behavior. They earn from fee splits and inflation rewards proportional to adoption.

Delegators provide additional stake to operators. They share in operator rewards proportional to delegation.

The protocol captures a governance-controlled fee (currently 10%) funding development, security audits, and ecosystem growth.

1.2 Network Effects

Tangle exhibits positive network effects that compound over time.

Supply flywheel: More operators → more compute → more products → more demand → more fees → more operators.

Application flywheel: More developers → more blueprints → more use cases → more customers → more fees → more developers.

Security flywheel: More delegators → more stake → more security → higher-value use cases → more fees → more stake.

These flywheels interact. More security attracts enterprise customers. Enterprise customers demand specialized blueprints. Specialized blueprints require capable operators. Capable operators attract delegations.

2 The Decentralized Sandbox Runtime

The sandbox runtime is where autonomous work actually executes. It provides secure, accountable compute for AI agents and automated workflows.

2.1 Motivation: Trust and Distribution

Centralized providers make decisions unilaterally. They set prices. They define terms. They choose customers. They capture value. For infrastructure underpinning significant economic activity, concentration in few hands raises concerns.

Decentralized infrastructure offers an alternative. Independent operators compete to provide compute. Prices emerge from markets. Economic value distributes to participants. No single entity controls access.

The sandbox runtime implements this alternative for AI agent execution. Operators run agents in isolated containers. Customers pay with cryptographic accountability. The protocol coordinates without controlling.

2.2 Isolation Technologies

Operators host sandbox environments using industry-standard isolation:

Docker provides container-based isolation with process, filesystem, and network separation. Containers are lightweight, well-understood, and widely deployed. Suitable for many workloads where standard isolation suffices.

gVisor adds an additional isolation layer. gVisor intercepts system calls, providing a user-space kernel that limits attack surface. Suitable for higher-security requirements where container escapes are a concern.

Firecracker and micro VMs provide hardware-level isolation. Micro VMs boot in milliseconds while providing VM-strength isolation. Suitable for workloads requiring the strongest guarantees, where even kernel-level exploits should not compromise the host.

Operators choose technologies based on their security requirements, performance needs, and the blueprints they serve. The protocol provides the economic framework; operators make technical decisions.

2.3 Isolation Guarantees

Regardless of technology, certain guarantees must hold:

- **Process isolation**: No access to host resources or other sessions
- **Filesystem isolation**: Private storage with quotas
- **Network isolation**: Restricted external access per policy
- **Resource limits**: Bounded CPU, memory, and other consumption

Operators who fail to maintain guarantees face slashing.

2.4 Deployment Models

2.4.1 Decentralized Operators

Independent operators stake assets, run infrastructure, and compete for customers. They earn fees proportional to services provided. They choose their technology stack, geographic location, and pricing strategy.

Benefits:

- No single point of failure
- Competitive pricing from market dynamics
- Geographic and organizational diversity
- Economic accountability through stake

2.4.2 Managed Cloud

For customers preferring a managed experience, Tangle offers hosted infrastructure with the same protocol guarantees. The managed option provides:

- Simplified onboarding
- Consistent SLAs
- Integrated billing
- Technical support

Both models use the same protocol. Customers can mix operators—some decentralized, some managed—based on their requirements.

2.5 Sandbox as Sidecar

The sandbox serves as both primary execution environment and as a sidecar alongside other systems:

- DeFi protocols for AI-assisted monitoring
- Oracle networks for data processing
- Keeper networks for decision logic
- Any system needing secure, accountable AI execution

2.6 Pricing Mechanisms

Service pricing uses request-for-quote (RFQ):

1. Customer broadcasts quote request specifying requirements
2. Operators return signed quotes with pricing
3. Customer selects and submits quotes
4. Service activates with committed prices

This accommodates operator heterogeneity—different hardware, locations, and capabilities command different prices. Customers evaluate holistically, not just on price.

For recurring workloads, subscription pricing provides predictable costs. Customers fund escrow; the protocol bills at intervals.

For variable workloads, event-driven pricing charges per job. Customers pay when they submit work.

3 The Agentic Workbench

The workbench is where autonomous work is authored and refined. It provides the creative environment for designing, testing, and deploying AI-powered workflows.

3.1 Vibe Coding Platform

Today, the workbench operates as a vibe coding platform for building projects against blockchain ecosystems. Users describe what they want; AI agents create the implementation.

The platform handles development environment complexity. Pre-configured containers include SDKs, tools, and dependencies:

- **Ethereum**: Foundry, Hardhat, OpenZeppelin
- **Solana**: Anchor, Solana CLI, program scaffolding
- **Other ecosystems**: Configured on demand

Users focus on what they want to build, not environment setup.

3.2 Session Persistence

Sessions persist across interactions. Each maintains:

- The codebase
- Conversation history
- Agent's accumulated project context

Users return to ongoing projects, review agent work, provide feedback, and iterate. Development becomes human-AI collaboration where the agent remembers previous decisions.

3.3 Technical Architecture

The workbench connects to the sandbox runtime through the protocol:

1. User initiates coding session
2. Workbench requests service from operator
3. Operator provisions sandbox container
4. Bidirectional connection established
5. User inputs flow to agent; outputs flow back
6. Protocol handles payment and accountability

The user sees a seamless environment; decentralized infrastructure operates invisibly.

3.4 Collaborative Features

The workbench supports multiplayer collaboration:

- Teams work on shared projects with synchronized state
- Multiple people observe agent progress
- Coordinated input on complex tasks

Unlike traditional IDE collaboration (editing same files), workbench collaboration means directing the same agent infrastructure: multiple humans guiding multiple agents toward shared goals.

Parallel development: One team member defines architecture while another refines implementation. A third focuses on testing. Agents work in parallel, each supervised by the relevant expert.

Collective oversight: AI agents make mistakes. Teams collectively review outputs, catch errors, and guide refinement. Shared oversight improves quality beyond what individuals achieve.

Knowledge transfer: Junior developers work alongside seniors, observing how experienced engineers direct agents. The workbench becomes a training environment.

3.5 Vibe Working: Beyond Code

The workbench expands beyond code to general knowledge work:

Research agents gather, synthesize, and present information. Deploy an agent to analyze competitors, summarize reports, identify regulations, and present findings.

Analysis agents process data and generate insights. Upload a dataset; the agent performs statistics, identifies patterns, generates visualizations, and suggests interpretations.

Writing agents produce reports, documentation, and communications. Transform analysis into polished output: summaries, documentation, presentations.

The vision is a unified environment for all AI-assisted work. Users engage with agents across task types without switching tools.

3.6 The Parallel Agents Paradigm

Traditional AI interfaces present single-threaded conversation. Real projects involve exploration, comparison, and parallel investigation.

Spawning sub-agents: A primary agent working on a smart contract spawns sub-agents to research gas optimization, investigate similar implementations, and draft tests. Each runs independently, reporting results back.

Forking for exploration: Facing an architectural decision, fork the context and direct each down a different path. Both develop in parallel. Observe progress, compare results, pick the winner.

Spatial canvas: The workbench presents parallel activity visually, showing all active agents, their status, recent outputs, and relationships. Users manage parallel work without overwhelm.

3.7 The Evaluation Loop

Every execution generates traces: what agents did, inputs received, outputs produced, operation timing. These traces feed evaluation systems:

- Which prompt structures produce better results?
- Which model configurations work for which tasks?
- Which operators deliver lower latency?

This creates a flywheel: more usage → more traces → better system → more usage. Early users benefit from infrastructure; later users benefit from accumulated learning.

4 Ecosystem and Extensions

The sandbox and workbench are first-party products, but they represent just the beginning.

4.1 Product Interactions

Users can design workflows in the workbench and deploy to sandbox runtime, or interact with sandbox directly through APIs. Enterprise systems, automated pipelines, and third-party applications access compute through standard interfaces.

The workbench itself consumes sandbox compute, making it simultaneously a product and a protocol customer.

4.2 Third-Party Extensions

The protocol's openness enables:

- Specialized workbenches for domains (legal, medical, financial)
- Infrastructure tools for operators (monitoring, scaling, fleet management)
- Aggregator services helping customers find operators

Each third-party product creates and captures value while the protocol captures its fee regardless of which products generate activity.

4.3 Use Cases Enabled

Confidential AI for enterprises: A pharmaceutical company needs AI analysis of proprietary clinical data. They select operators with verified security credentials, require specific isolation, and maintain cryptographic evidence of proper handling.

Verifiable AI for high-stakes decisions: A trading firm needs confidence that analysis used the specified model and data. Verification mechanisms detect model substitution or data manipulation.

Collaborative AI for distributed teams: A research consortium spans institutions with different governance. Each runs operator infrastructure within its boundaries while participating in collaborative workflows.

Autonomous agents with accountability: Deploy AI agents that take real-world actions. Agents run in sandboxed environments with logged actions and economic guarantees.

Developer monetization at scale: Create a blueprint; earn from every service instantiation across all operators. Inflation provides additional rewards proportional to adoption.

5 Incentives and Pricing

5.1 Operator Economics

Operators earn from multiple sources:

Service fees: Direct payment for services provided, distributed according to exposure commitment.

Inflation rewards: Protocol inflation distributed based on job execution success rate and stake weight.

Tips and premiums: Customers may offer premiums for priority, specific hardware, or geographic proximity.

Operator profitability depends on:

- Hardware costs (compute, storage, network)
- Operational costs (maintenance, monitoring)
- Stake opportunity cost
- Market pricing dynamics

5.2 Customer Pricing

Customers pay based on:

- Compute time and resources consumed
- Isolation technology required
- Operator quality and location preferences
- Service duration and commitment

The RFQ system enables price discovery. Customers can specify budgets; operators quote within those constraints.

5.3 TNT Token Utility

The TNT token serves multiple functions:

Staking: Required for operator participation. Stake determines service capacity.

Delegation: Passive holders earn yield by backing operators.

Governance: Token holders vote on protocol parameters.

Payment: Services priced in TNT may receive discounts.

Fee distribution: Rewards flow in TNT, creating coherent value circulation.

6 Roadmap

Development proceeds through phases targeting progressive capability and decentralization.

6.1 Current State

Core protocol contracts: Deployed and audited. Full lifecycle operational.

Blueprint SDK: Complete with triggers, handlers, consumers, P2P networking, and QoS.

Workbench: Operating as vibe coding platform. Sessions, environments, and basic collaboration functional.

Operator network: Initial operators running sandbox infrastructure.

6.2 Near-Term Priorities

Multiplayer collaboration: Real-time shared sessions. Multiple users directing agents simultaneously. Synchronized state and coordinated outputs.

Vibe working expansion: Research, analysis, and writing tasks. New agent configurations, expanded tools (web search, document processing, data analysis).

Operator network growth: Geographic and organizational diversity. Improved tooling. Lower entry barriers.

Additional isolation: gVisor and Firecracker integration. SDK updates for configuration. Verification mechanisms for compliance.

Standard verification libraries: Audited implementations of common patterns (oracle verification, compute verification, availability monitoring).

6.3 Longer-Term Development

Full governance activation: Transfer control to token holders. Relax guardian oversight as governance proves capability.

Cross-chain deployment: L2s (Arbitrum, Base, Optimism) for lower transaction costs. Additional L1s if demand justifies.

Ecosystem grants: Fund developer tooling, verification research, operator infrastructure.

Advanced verification research: ZK proofs for compute verification, TEE integration, formal verification of protocol properties.

Protocol upgrades: Sophisticated pricing mechanisms, additional delegation modes, enhanced slashing parameters.

6.4 Dependencies

- Multiplayer requires stable single-user experience
- Vibe working expansion requires proven agent reliability
- Governance activation requires sufficient token distribution
- Cross-chain requires mainnet stability

Near-term items are largely independent; longer-term items build on earlier foundations.

6.5 Adaptability

The roadmap adapts as conditions change. Governance adjusts priorities. Success is measured by metrics: operator count, service volume, developer adoption, stake growth.

7 Measuring Success

Key metrics for ecosystem health:

Operator diversity: Geographic and organizational distribution. No single entity should dominate.

Blueprint diversity: Use case coverage. Are blueprints serving varied needs?

Customer growth: New customers and retention. Is the product compelling?

Fee volume: Protocol revenue. Is value being generated?

Stake growth: Total value locked. Is security increasing?

These metrics guide parameter adjustments (inflation rates, fee structures, minimum stakes) to ensure sustainable growth.

8 Conclusion

Tangle Cloud demonstrates that decentralized AI infrastructure is not merely theoretical but operational. The sandbox runtime provides secure execution on distributed operator infrastructure. The workbench provides collaborative authoring for AI-assisted work. Together they show a path beyond centralized cloud dependence.

The products connect to the protocol's economic layer. Every session, every job, every agent interaction generates value that flows to operators, developers, delegators, and the protocol. The compute economy becomes a shared enterprise rather than a corporate extraction.

The question is not whether AI infrastructure works—centralized providers have proven that. The question is who controls it, who benefits, and whether alternatives should exist. Tangle Cloud provides that alternative: infrastructure that is resilient, competitive, and owned by its participants.

The future of AI compute will be built by those who build it now.