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.

This document complements the Tangle Protocol Specification (core mechanisms and security model) and the Blueprint SDK Developer Guide (implementation details). Together they describe the complete Tangle ecosystem.

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:

Customers use products (sandbox runtime, workbench, or third-party applications), request services from operators, and pay in tokens. Payments split among developers, operators, and the protocol.

Operators provide compute by staking tokens and running infrastructure. They earn fees proportional to services provided, and can serve more services as they accumulate more stake.

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

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

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. Understanding these dynamics helps participants identify opportunities and predict growth.

1.2.1 Supply Flywheel

More operators → more compute → more products → more demand → more fees → more operators.

As operator count increases, geographic and capability diversity grows. Customers find operators meeting their specific requirements. More customer options attract more demand. More demand generates fees that attract operators with capital to stake.

The flywheel accelerates as barriers fall. Improved tooling reduces operational burden. Documentation and support lower entry costs. Reference implementations demonstrate viability.

1.2.2 Application Flywheel

More developers → more blueprints → more use cases → more customers → more fees → more developers.

Developers create blueprints when they see opportunity. Blueprint diversity attracts customers with varied needs. Customer payment generates fees and inflation rewards for developers. Developer success attracts more developers.

This flywheel benefits from composability. Blueprints can reference and extend each other. A successful oracle blueprint enables DeFi blueprints. DeFi success attracts custody blueprints. Ecosystem density increases value of each component.

1.2.3 Security Flywheel

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

Security is the product delegators produce. More stake backing operators enables higher-value services. Services managing $1M need operators with $1.5M+ stake. Higher-value services generate more fees. Fees attract more delegation.

Delegation yield compounds the effect. Delegators earning attractive yield retain and increase positions. Growing stake enables progressively higher-value use cases. Each tier of security unlocks new customer segments.

1.2.4 Flywheel Interactions

These flywheels interact in reinforcing cycles:

- More security attracts enterprise customers requiring strong guarantees
- Enterprise customers demand specialized blueprints for their industries
- Specialized blueprints require capable operators with domain expertise
- Capable operators attract delegations from participants seeking yield
- Delegations increase security, enabling more enterprise customers

The network grows as a coherent whole rather than isolated components. Early participants benefit from positions in nascent flywheels. Later participants benefit from mature infrastructure and proven economics.

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.

The sandbox runtime implements decentralized AI agent execution. Operators run agents in isolated containers with cryptographic accountability. The protocol coordinates without controlling, addressing the concentration concerns outlined in Section 1.

2.1 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.

2.2 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.

2.3 Deployment Models

2.3.1 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

2.3.2 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.

2.4 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

2.5 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 Natural Language Development Platform

Today, the workbench operates as an AI-assisted development platform for building projects against blockchain ecosystems. Users describe intent in natural language; AI agents translate intent into 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 via RFQ
3. Operator provisions sandbox container with specified isolation
4. P2P connection established (see *Blueprint SDK Guide*, P2P Networking)

5. User inputs flow to agent; outputs flow back via gossipsub
6. Protocol handles payment splits (developer, operator, delegator, protocol)

The user sees a seamless environment; the decentralized infrastructure operates through the same mechanisms described in the *Tangle Protocol Specification*, blueprints define the workbench service, operators stake to provide compute, and slashing conditions enforce isolation guarantees.

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 AI-Assisted Knowledge Work

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.

5.2.1 Pricing Examples

AI Development Session (Workbench):

- Base compute: 2 vCPU, 4GB RAM container
- AI model inference: Pass-through from LLM provider
- Session persistence: Storage for code and context
- Rate: Competitive with centralized alternatives, with cryptoeconomic accountability

AI Agent Deployment (Sandbox Runtime):

- Resource allocation: Configurable CPU/memory/storage
- Isolation premium: +10-30% for gVisor/Firecracker vs Docker
- Geographic premium: Additional cost for specific region requirements
- Security requirement: High-value workloads require operators with larger stakes

5.2.2 Pricing Models

Three pricing models accommodate different workload types:

Subscription: Predictable costs for recurring workloads. Customers fund escrow; the protocol bills at intervals.

Event-driven: Per-job pricing for variable workloads. Customers pay when they submit work.

Pay-once: Upfront payment for one-time computations, distributed when operators approve.

See the *Tangle Protocol Specification* for detailed payment flow mechanics.

5.3 TNT Token Utility

The TNT token serves multiple functions within the ecosystem:

Staking is required for operator participation. The amount staked determines service capacity and enables operators to accept higher-value workloads requiring greater economic security.

Delegation allows passive holders to earn yield by backing operators. Delegators share in operator rewards proportional to their contribution, providing a way to participate without running infrastructure.

Governance grants voting rights on protocol parameters. Token holders can propose and vote on fee structures, inflation rates, and protocol upgrades.

Payment in TNT may receive discounts on service fees, aligning customer incentives with ecosystem participation.

Fee distribution flows in TNT, creating coherent value circulation where network activity generates rewards that flow back to participants.

6 Roadmap

Development proceeds through phases targeting progressive capability and decentralization.

6.1 Current State

Core protocol contracts: Deployed and audited. Full service lifecycle operational including: blueprint registration, operator staking, service creation via RFQ, job execution, payment distribution, and slashing with dispute windows. See *Tangle Protocol Specification* for mechanism details.

Blueprint SDK: Production-ready with comprehensive features:

- Event triggers (blockchain events, cron schedules, custom conditions)
- Job handlers with extractor pattern (`TangleArg`, `TangleResult`)
- Result consumers for on-chain submission
- P2P networking (gossipsub, request-response, peer discovery)
- QoS-based slashing hooks (heartbeat, latency, availability)
- Background keepers for lifecycle automation

See *Blueprint SDK Developer Guide* for implementation details.

Workbench: Operating as AI-assisted development platform. Sessions, environments, and basic collaboration functional.

Operator network: Initial operators running sandbox infrastructure across multiple geographic regions.

6.2 Near-Term Priorities

Multiplayer collaboration: Real-time shared sessions enabling multiple users to direct agents simultaneously. Technical requirements: WebSocket state synchronization, conflict resolution for concurrent edits, role-based access control for team permissions.

Knowledge work expansion: Research, analysis, and writing tasks beyond code. Requires: new agent configurations with specialized prompts, expanded tool integrations (web search APIs, document processing libraries, data analysis frameworks), UI components for non-code outputs.

Operator network growth: Target 50+ operators across 10+ geographic regions within 6 months. Initiatives: operator onboarding documentation, one-click deployment scripts, monitoring dashboards, community support channels.

Additional isolation technologies:

- gVisor integration: User-space kernel for defense-in-depth against container escapes
- Firecracker micro-VMs: Hardware-level isolation with millisecond boot times
- Blueprint configuration: Operators declare supported isolation; customers specify requirements

Standard verification libraries: Audited, reusable implementations:

- Oracle verification (median aggregation, TWAP, threshold signatures)
- Compute verification (deterministic replay, model fingerprinting)
- Availability monitoring (heartbeat keepers, latency tracking)

6.3 Longer-Term Development

Full governance activation: Progressive decentralization transferring control to token holders. Phases:

1. Parameter governance (fee rates, inflation schedules)
2. Upgrade governance (contract upgrades via timelock)
3. Full governance (guardian multisig sunset)

See *Tangle Protocol Specification*, Governance section for mechanism details.

Cross-chain deployment:

- L2 priority: Arbitrum, Base, Optimism for lower transaction costs
- Bridge integration: Canonical bridges for TNT token portability
- Service portability: Same blueprint deployable across chains

Ecosystem grants: Fund development in key areas:

- Developer tooling (IDE plugins, debugging tools, testing frameworks)
- Verification research (ZK proofs for compute, TEE attestation)
- Operator infrastructure (monitoring, scaling, fleet management)

Advanced verification research:

- ZK proofs for compute verification without revealing inputs
- TEE integration (SGX, TDX) for hardware-attested execution
- Formal verification of protocol invariants

Protocol upgrades: Enhanced mechanisms based on operational learnings:

- Dynamic pricing algorithms responding to utilization
- Reputation systems for operator quality signaling
- Cross-service exposure limits for risk management

6.4 Dependencies

- Multiplayer requires stable single-user experience
- Knowledge work 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 measures geographic and organizational distribution. A healthy network has no single entity dominating compute provision.

Blueprint diversity tracks use case coverage. Blueprints should serve varied needs across industries and applications.

Customer growth monitors new customer acquisition and retention rates, indicating whether products deliver compelling value.

Fee volume measures protocol revenue from service payments, indicating real economic activity.

Stake growth tracks total value locked, reflecting increasing security capacity for higher-value workloads.

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

8 Risks and Limitations

Honest assessment of challenges facing Tangle Cloud:

8.1 Technical Risks

Isolation breaches: Container and VM isolation technologies have known attack vectors. While defense-in-depth (Docker + gVisor + Firecracker) reduces risk, no isolation is perfect. Slashing provides economic deterrence but cannot undo data exposure.

Verification limitations: Some computations are inherently difficult to verify (AI model inference, complex simulations). Verification mechanisms may have false negatives (undetected cheating) or false positives (incorrect slashing). Blueprint designers must understand detection probability for their specific use case.

8.2 Economic Risks

Token volatility: Slashing effectiveness depends on stake value. Rapid token price declines can reduce security margins below safe thresholds faster than services can respond. Circuit breakers help but cannot eliminate this risk.

Cold start problem: Two-sided marketplaces face bootstrapping challenges. Customers need operators; operators need customers. Early network growth requires subsidies (inflation rewards) that must eventually transition to sustainable fee-based economics.

8.3 Adoption Risks

Operator supply: Decentralized infrastructure requires sufficient operators across geographies and capabilities. Operator economics must be attractive enough to compete with traditional employment or alternative yield opportunities.

Developer tooling maturity: The SDK and development experience must match or exceed centralized alternatives. Gaps in documentation, debugging tools, or deployment automation slow adoption.

8.4 Competitive Landscape

Centralized cloud providers offer mature tooling, established trust, and economies of scale. Decentralized alternatives (Akash, Render) compete for similar users. Tangle's differentiation through cryptoeconomic accountability must prove compelling enough to overcome switching costs.

9 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.