Machine Trust Protocol (MTP) v1.0

The Trust Layer of the Machine Civilization

A Foundational Layer for the Machine Economy

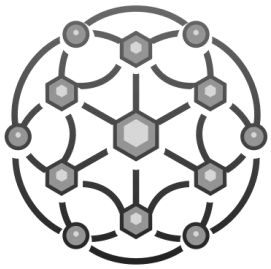
A Protocol-Agnostic Standard for Verifiable Machine Trust

Powered by NodeX Labs

Author: Ken Zhou — Founder & CEO, NodeX Labs Inc.

Date: Oct. 31, 2025

Version: v1.0



N O D E - X

Note on Versions:

This whitepaper (v1.0) defines the architecture and governance of MTP.

The companion "MTP Technical Specification v1.1" refines transport bindings and envelope schemas while remaining backward-compatible with v1.0.

In v1.1, envelope field extensions are introduced; validators implementing v1.0 MUST ignore unknown fields to preserve compatibility.

Table of Contents

Preface — The Coming Machine Civilization

Humanity stands at the threshold of a new epoch.

For centuries, humans built machines to serve; today, machines begin to **act, trade, and reason** on our behalf.

Autonomous AI agents, decentralized compute networks, and self-operating devices are forming the early infrastructure of a **machine civilization** —

a world run *for humans, by machines*.

These entities exchange not only data, but also **value, compute, and intent**.

Yet one question remains unsolved:

> **When machines act, how do we trust them — and how do they trust each other?**

Trust, once a social contract among humans, must now be **re-engineered for the robotic age**.

If the Internet solved *information flow*, and blockchains solved *value transfer*,

the next frontier is **machine trust** — a framework where every robot, AI agent, and hardware node can cryptographically prove who they are and what they did.

MTP defines this missing layer: a **verifiable, decentralized, and programmable foundation of trust** for the machine economy.

It is not an app or service; it is a **meta-protocol** — designed for **standardization, open adoption, and cross-ecosystem integration**.

Chapter I — Manifesto of Machine Trust

1.1 The End of Human-Centric Trust

Traditional trust models were built for humans — laws, contracts, institutions.

In the digital age, this became **platform trust**: users trusting centralized companies to host their data and execute their code.

As autonomous agents manage wallets, coordinate compute, and negotiate resources, the center no longer holds.

We need a **trust primitive that machines can generate and verify** without human mediation.

1.2 From Silicon Trust to Algorithmic Trust

Hardware modules like TPM and SGX tried to anchor trust in silicon — yet these chips are closed, monopolized, and unauditable.

When the root of trust hides in a black box, it stops being trust; it becomes belief.

MTP shifts trust from closed silicon to open cryptography: *trust not in silicon, but in math.*

Every execution, transaction, and computation emits a proof — forming a **living ledger of verifiable actions**.

While MTP does not rely on closed silicon, it can **consume TEE/TPM attestations** as optional evidence, anchoring them to public cryptographic proofs.

1.3 Trust as the New Compute

Compute is abundant; verifiable compute is scarce.

The most valuable resource is not raw power, but **provable execution**.

MTP treats **trust itself as a computational commodity** — measurable, transferable, and rewardable.

1.4 Declaration

> We believe that every machine has the right to prove its existence and its work.
>
> We believe that trust should not be manufactured; it should be computed.
>
> We believe that in a civilization of machines, truth is the new currency.

Outcome

The Manifesto defines why MTP must exist:

to transform trust from a **social contract** into a **computational primitive**.

This vision sets the stage for **Chapter II**, which diagnoses the **trust crisis** threatening the machine economy.

Chapter II — The Trust Crisis in the Machine Age

2.1 The Architecture Gap

The Internet moves information, not value or truth.

Even with blockchains, the execution layer of machines remains opaque: *who actually ran the task, and where?*

Without a native trust layer, the machine economy faces:

- **Ghost mining** via synthetic containers
- **Result forgery** through cached outputs
- **Verifier cartels** colluding to approve each other's results

---

2.2 Economic Consequence

When machines cannot prove their work, **value inflates on fiction**.

Tokens reward ghost nodes; AI outputs lose authenticity; compute markets become speculative.

The result: **collapse of machine credibility**.

2.3 Philosophical Consequence

We outsourced muscle to machines; now we outsource mind.

If we cannot verify the entities that think for us, we surrender sovereignty.

To retain agency, trust must be grounded in **transparent, decentralized verification.**

2.4 The Need for a New Layer

x402 awakened the payment layer — machines can now pay.

But they still cannot prove.

If **x402 lets machines pay**, then **MTP lets machines be trusted.**

2.5 Threat Model

Adversaries:

(a) Sybil operators faking nodes

(b) Lazy executors claiming unearned rewards

(c) Replay / relay attackers

(d) Byzantine verifiers colluding

(e) Centralized hardware bias from unauditable silicon

Assumptions: secure cryptographic primitives (e.g. Ed25519 / BN254), unbiased randomness, network synchrony within Δ, and ≥ 1/3 honest verifiers.

2.6 Security Goals

- **G1 Authenticity** — Bind real devices to cryptographic identities (ProofX).
- **G2 Verifiable Execution** — PoRW soundness and non-amortizability: no valid proof without real work; solving k independent challenges ≈ k units of work.
- **G3 Anti-Replay** — Epoch-bound nonces and idempotent settlement prevent replay/relay and double-settle.
- **G4 Collusion Resistance** — Randomized, stake-weighted verifier sampling

with dispute windows and fraud-proofs.

- **G5 Transport Agnosticism** — Proof validity is independent of transport bindings (x402, x403, WebSocket, MQTT, gRPC, on-chain).

> *Result:* a transport-independent trust kernel that ties **who** executed, **what** was executed, and **why** settlement is deserved.

Outcome

The Machine Age exposes a **trust vacuum** between computation and verification.

MTP emerges as the missing **Trust Kernel** for the machine economy.

The next chapter defines its **principles and design philosophy.**

Chapter III — Design Philosophy: From Silicon to Algorithmic Trust

The Machine Trust Protocol (MTP) redefines how machines, networks, and algorithms establish trust —

not through static credentials or centralized authorities, but through continuous, verifiable proofs of real work.

Just as HTTPS secured communication and x402/x403 introduced native payment, MTP represents the next stage in the Internet's evolution: **verifiable execution**.

3.1 Principles of Trust by Design

MTP is founded on the belief that *trust must be designed into computation itself* — embedded at the machine and protocol level, not imposed externally.

Formal Properties

- **Unforgeability** — No adversary can generate a valid PoRW for work that was never executed.
- **Non-Amortizability** — Solving *k* independent challenges requires approximately *k* units of genuine computation, preventing “share-once, claim-many” attacks.
- **Upgradability** — Schema or protocol updates must preserve backward compatibility of proofs, ensuring long-term verifiability.
- **Binding Independence** — Proof validity remains invariant across different transport bindings (x402, x403, WebSocket, etc.), enabling interoperability and future-proof design.

These principles collectively ensure that **trust becomes a property of computation itself**, not a layer of bureaucracy around it.

3.2 From Static Identity to Dynamic Proof

Traditional identity systems rely on *who a participant claims to be*.

In contrast, MTP defines identity as an **ongoing stream of attestations** —

cryptographically signed telemetry including uptime, hardware fingerprints, and task integrity.

Each device continuously emits **ProofX attestations**, forming a living ledger of verified work rather than a static credential.

This shift turns identity from a **snapshot** into a **state machine** —

a dynamic record of reputation, consistency, and real contribution within the network.

3.3 Decentralized Verifiability

In MTP, trust is not granted by a single root authority but *emerges from distributed consensus*.

A **verifier mesh** validates execution proofs, with each verifier staking both tokens and reputation.

Misbehavior — signing invalid proofs or censoring challenges — results in **slashing** and loss of reputation.

Over time, the mesh self-regulates: trustworthy verifiers gain influence, while dishonest ones are marginalized.

Thus, trust evolves from a **claim** to a **collective truth** —

verified continuously, cryptographically, and economically.

Outcome

MTP's design philosophy transforms trust from a human convention into a **computational primitive**.

From **silicon** (hardware fingerprints) to **algorithmic consensus** (verifier mesh), every layer contributes to a verifiable, self-sustaining trust fabric.

This foundation prepares the ground for the **architectural definition** introduced in **Chapter IV**, where these design principles take concrete form within the Layered Architecture of MTP v1.0.

Bridge Section — From Philosophy to Architecture

If **Chapter III** defines *why trust must be redesigned*,

then **Chapter IV** defines *how that redesign manifests in protocol form*.

The philosophical tenets — dynamic proof, distributed verifiability, and computation as trust —

now materialize into a formal **Layered Architecture**, specifying the trust objects, flow, and interoperability logic of MTP.

Chapter IV — Protocol Overview

(Architecture v1.0: Machine Trust Protocol Layered Design)

The **Machine Trust Protocol (MTP)** is a **protocol-agnostic meta-protocol** defining the minimal primitives for verifiable **identity**, **execution**, and **settlement**.

It establishes a universal trust layer that operates across both Web2 and Web3 environments, allowing machines, agents, and devices to transact securely and transparently without relying on centralized intermediaries.

Core Design Philosophy

MTP introduces three foundational trust objects:

- **ProofX** — establishes verified machine identity through cryptographic attestation.
- **PoRW (Proof of Real Work)** — validates that computational tasks were genuinely executed by authentic devices.
- **MTP-Settlement Envelope** — ensures transport-agnostic value transfer, binding execution proofs to payment rails.

Each object is modular, composable, and transport-independent, allowing integration across HTTP (x402), x403, WebSocket, MQTT, gRPC, or direct on-chain channels.

Layered Architecture (Textual Overview)

1. **Clients** produce **ProofX** (identity) and execute tasks that generate **PoRW** (execution proofs).
2. Both proofs are encapsulated within an **MTP-Envelope**, serving as a portable trust container.
3. **Adapters** bind these envelopes to multiple transport standards — HTTP/x402, x403, WebSocket, MQTT, gRPC, or direct blockchain submission.
4. A distributed **verifier mesh** validates proofs, attests results, and triggers **settlement** through the designated payment rail.

This architecture decouples *how machines prove their work* from *how that work is paid and verified*, ensuring cross-network interoperability and long-term scalability.

Figure 4-1. MTP Layered Architecture

```
Application & Payment Layer
→ HTTP (x402) | x403 | WebSocket | MQTT | gRPC | On-chain
Trust Layer — Machine Trust Protocol (MTP)
→ ProofX  (Identity & Attestation)
→ PoRW    (Proof of Real Work)
→ MTP-Envelope  (Settlement & Binding)

Verification & Execution Layer
```

```
→ Verifier Mesh | Coordinator Nodes | NodeHub Clients


Hardware / Network Layer
→ TEE | TPM | Server / Edge / GPU Devices
```

Figure 4-1 illustrates MTP's Layered Architecture, showing how real devices generate verifiable identities and execution proofs, which are validated and settled across heterogeneous networks.

Outcome

MTP v1.0 defines the **architectural backbone** for the Machine Trust Economy — a universal trust protocol connecting real-world compute resources, execution proofs, and settlement systems.

This architecture establishes the technical foundation for the **Core Components** detailed in **Chapter V**, where ProofX, PoRW, and MTP-Settlement are implemented as modular, interoperable building blocks.

Chapter V — Core Components

This chapter defines the three foundational modules of the Machine Trust Protocol (MTP):

ProofX (identity and attestation), **PoRW** (proof of real work), and **MTP-Settlement** (the transport-agnostic settlement layer).

Together, they form the technical core of NodeX's trust fabric — verifying *who acted*, *what was done*, and *how it is settled*.

Figure 5-1. NodeX MTP Core Component Stack

ProofX — Identity & Attestation

→ Verifies *who* performs the work

PoRW — Proof of Real Work

→ Verifies *what* was actually executed

MTP-Settlement — Transport-Agnostic Settlement

→ Verifies *how* value is transferred and finalized

Outcome:

Verified Identity → Proven Execution → Settled Payment

Figure 5-1 illustrates the three core modules of the Machine Trust Protocol (MTP).

Together, they form the trust spine of NodeX — linking identity, execution, and economic settlement into a continuous verifiable process.

Figure5.1 ProofX — Identity & Attestation

Objective: Generate cryptographic identity for real devices.

Mechanism

1. Hardware fingerprinting (CPU / GPU / storage / network entropy)
2. Local signing via NodeX attestation client
3. On-chain (or DLT) registration with verifier mesh
4. Periodic heartbeat with epoch-bound nonces proving liveness and location consistency

Outputs

- ProofX_ID (pseudonymous device hash)
- Attestation_Token (VC / JWT-style signed object)
- Reputation_Score (weighted by uptime and validation history)

Benefits

- No central authority
- Sybil-resistant
- Auditable across chains and applications

Privacy & DID/VC

- Minimal Disclosure — ProofX_ID is pseudonymous; device attributes selectively disclosed via W3C VC / ERC-8004.
- Rotating nonces prevent linkage between sessions.
- Optional TEE evidence included as evidence[].

5.2 PoRW — Proof of Real Work

Objective: Verify that computing tasks were executed by authentic resources.

Workflow

1. Job assignment via NodeHub or third-party coordinator
2. Device executes task locally and generates work trace

3. Verifier nodes sample and re-compute hash segments
4. Consensus attestation → PoRW_Proof (on-chain or content-addressed)
5. Settlement trigger via MTP-Settlement

Key Features

- ZK-compatibility (optional SNARKs for confidential tasks)
- Randomized auditing (dynamic committee per task)
- Time-bound hashing (epoch nonces to prevent replay)

Verifier Policy (normative)

- Sampling rate $r \in (0, 1]$, default 0.05 with stratified segments
- Committee size m chosen so that $\Pr[\geq t \text{ faulty}] < 10^{-6}$ under stake model
- Epoch nonce $N_e = H(\text{network\_epoch} \| \text{ctx} \| \text{receiver})$ rebinding proofs
- Dispute window: proofs challengeable for Δ blocks with fraud-proof

Soundness Note

MTP adopts non-amortizability constraints so that batch solving multiple independent tasks yields ≈ linear work, preventing share-once, claim-many attacks.

5.3 MTP-Settlement — Transport-Agnostic Settlement Layer

Objective: Bridge execution proofs to value transfer across any rail (x402, x403, L2, MQTT-metering, etc.).

Design

Settlement lives inside the MTP-Envelope with fields:

{ binding, asset, amount, decimals, receiver, proof_ref, idempotency_key, epoch, nonce }

Bindings

- **x402:** map to HTTP 402 headers (legacy binding)
- **x403:** decentralized transport (WebSocket / MQTT / gRPC)
- **L2 Direct:** post on-chain receipt referencing proof_ref

Idempotency

Repeat requests with the same idempotency_key MUST NOT double-settle.

Replay Protection

Verify epoch and nonce against current window to ensure temporal validity.

Outcome

Together, ProofX, PoRW, and MTP-Settlement compose the machine-verifiable trust stack:

- **ProofX** answers *who performed the action*;
- **PoRW** proves *what was executed and how it was verified*;
- **MTP-Settlement** ensures *how value is transferred and accounted for*.

This triad forms the technical spine of MTP and lays the foundation for the interoperability architecture detailed in **Chapter VI**.

Chapter VI — Interoperability & External Standards

Machine Trust Protocol (MTP) is designed to operate natively within both Web2 and Web3 environments, connecting machine identities, verified execution, and payment rails into a unified trust fabric. Rather than replacing existing standards, MTP extends and harmonizes them — ensuring that every agent, device, or protocol can interoperate under a shared framework of verifiable trust.

Disclaimer: References to third-party initiatives and standards are for technical context only and do not imply partnership, endorsement, or completed integrations.

6-1. NodeX MTP Interoperability Stack

(Layered structure showing the integration between x402/x403, AP2, TAP, ERC-8004/DID, and hardware/cloud bridges through the MTP Trust Kernel.)

6.1b. x402 — Machine Payment over HTTP

The **x402** standard reactivates the legacy HTTP 402 “Payment Required” status to enable machine-to-machine micropayments embedded directly within HTTP responses. Under MTP, each x402 transaction couples a payment with **proof metadata**, linking every value transfer to a verifiable proof-of-execution. This creates a native mechanism for automated, authenticated service payments across traditional web infrastructures.

(Supported via legacy Web2 bindings for backward compatibility.)

6.1b x403 — Decentralized Machine Payment

While x402 anchors payments to HTTP, **x403** generalizes the same principle to **decentralized transports** — allowing autonomous agents to exchange verified value without centralized intermediaries.

In this dual-stack design, **MTP provides the trust fabric**, and **x403 provides the decentralized settlement rail**.

They are **orthogonal yet complementary**: MTP authenticates the actor and action, while x403 routes and finalizes the payment.

Together, they form the **execution–settlement continuum** of the machine economy.

6.2 Google AP2 — Agent Payments Protocol

Industry initiatives such as Google's Agent Payments Protocol (AP2) aim to standardize agent authentication and settlement across Web2 and Web3 contexts. Within such a framework, MTP's ProofX can function as a trust oracle—supplying attested device identities, verified runtime records, and proof-of-execution attestations that AP2-style systems *may* consume via standard interfaces. No partnership, endorsement, or completed integration is implied.

6.3 Visa TAP — Trusted Agent Protocol

Industry proposals such as Visa's Trusted Agent Protocol (TAP) focus on authorization and compliance for AI-initiated payments. In this context, MTP can supply machine credentials that bind payment intent to verified execution, enabling TAP-style systems to achieve auditable and compliant settlement without centralized trust dependencies. No partnership, endorsement, or completed integration is implied.

6.4 ERC-8004 / W3C DID & Verifiable Credentials

MTP-ID adopts and extends **ERC-8004** and **W3C DID/VC** schemas, enabling each node or device to issue its own **verifiable credential (VC)** anchored to its **ProofX fingerprint**.

This standardization allows interoperability across multiple blockchains, identity providers, and AI frameworks — preserving **data sovereignty**, **device authenticity**, and **cross-domain verifiability**.

6.5 Hardware & Cloud Bridges

MTP functions as a **software-defined trust layer** that operates above heterogeneous hardware and cloud infrastructures.

It can verify across **TEE, TPM, GPU/CPU architectures, AWS, GCP, AliCloud**, and local environments **without vendor-specific firmware dependencies**.

This makes MTP the **universal trust kernel** — ensuring that computation, regardless of where it occurs, can be attested and settled on-chain.

6.6 Compatibility & The Need for a Trust Meta-Layer

MTP does not compete with existing verification standards such as TEE attestations, ZK proofs, or MPC protocols.

Instead, it functions as a **meta-verification layer** — a unifying trust standard that

aggregates and standardizes their outputs.

Each verification framework remains specialized, but MTP ensures their results can be trusted, referenced, and monetized across different ecosystems.

The Problems with Current Verification Models

Most existing verification protocols are isolated within narrow trust domains:

- **TEE / TPM** validate hardware execution but cannot prove identity continuity.
- **ZK / FHE / MPC** verify computational correctness but lack provenance or accountability.
- **Consensus protocols (PoS/PoW)** ensure block-level finality but cannot attest to off-chain execution.
- None provide a **universal proof of "who executed what, where, and under what guarantees."**

This fragmentation leads to *non-transferable trust* — proofs that are valid within one system but meaningless outside it, preventing cross-protocol composability or economic settlement.

MTP as the Solution

MTP introduces three key mechanisms to unify these domains:

- **ProofX:** establishes cryptographically verifiable machine identity and execution fingerprints.
- **PoRW:** standardizes validation of real workloads, independent of the underlying verification method.
- **MTP-Settlement:** binds these proofs to programmable payments (via x403 / TAP / AP2), turning verified computation into an *economically settleable event.*

By doing so, MTP transforms isolated verification islands into a **continuous trust graph** — where authenticity, performance, and payment can flow across

heterogeneous compute environments.

Why Other Protocols Integrate with MTP

- **For Visibility:** MTP turns opaque, local proofs into portable, recognized credentials consumable by external networks.
- **For Settlement:** It connects verification outputs to real economic value, enabling cross-protocol fee routing and incentive sharing.
- **For Interoperability:** It translates ZK/TEE/MPC attestations into a universal, machine-verifiable schema compatible with Web2 and Web3 payment systems.
- **For Compliance:** It extends verifiable intent, provenance, and accountability to machine actors — fulfilling audit and legal traceability requirements.

> **In short:**
>
> *TEE proves execution. ZK proves correctness. Consensus proves finality.*
>
> *MTP proves trust — binding them all into an interoperable, economically meaningful layer.*

Outcome

Through these integrations, MTP establishes the **trust kernel** that connects **identity**, **execution**, and **payment**.

It does not replace existing verification mechanisms; it binds them into a single coherent framework — one that makes compute verifiable, transferable, and economically alive.

This interoperability layer transforms the fragmented verification landscape into the foundation of a **global machine economy** powered by **NodeX**.

Chapter VII — Governance & Incentives

7.1 Verifier Mesh Governance

Verifiers stake tokens to participate in ProofX/PoRW validation. Consensus rules are set via on-chain governance. Malicious actors lose stake and reputation via **slashing**.

7.2 Reputation as Collateral

Reputation in NodeX functions as **productive collateral** — a non-transferable, decaying measure of credibility earned through verified uptime, task integrity, and consistent execution.

Unlike tradable assets, reputation embodies a node's **trust capital**, continuously reinforced by honest participation and gradually eroded without sustained contribution.

Dynamic Reputation Model (Proposed):

$R_{t+1} = \alpha R_t + \beta V_t - \gamma F_t$

where:

- V_t — validated work units completed in epoch *t*
- F_t — faults or verified slashing events
- $\alpha \in (0,1)$ — decay factor controlling memory length
- $\beta, \gamma > 0$ — empirically calibrated reinforcement and penalty weights

This formulation captures the natural decay of inactive nodes while rewarding consistent, verifiable performance.

The coefficients will be fine-tuned through simulation and testnet calibration to ensure **stability, fairness, and cross-device comparability** across diverse hardware

environments.

Slashing & Accountability:

If a verifier signs or propagates an invalid Proof-of-Real-Work (PoRW) that is detected within the dispute window, the system enforces a dual penalty — slashing *s %* of the staked collateral and deducting a proportional reputation score ΔR.

This dual-layered penalty ensures that both financial and reputational costs scale with misconduct severity, reinforcing accountability within the trust layer.

7.3 Node Economics

Nodes earn rewards through verified execution under the **Proof-of-Real-Work (PoRW)** mechanism.

Each validated task contributes to both **financial yield** and **reputation growth**, aligning economic incentives directly with verifiable performance.

Reputation serves as a **soft collateral layer** within the network economy — higher reputation implies stronger trustworthiness, granting nodes **higher task priority**, **larger fee shares**, and **access to premium workloads**.

This forms a positive feedback loop between long-term reliability and economic gain, while decaying reputation naturally throttles idle or unreliable nodes.

Verifiers receive a portion of settlement fees via **MTP-Settlement**, the on-chain clearing layer that routes payments proportionally to verified output and reputation weighting.

This ensures that network rewards remain tightly coupled to authenticity, consistency, and productive participation — realizing a **trust-weighted compute economy** at the heart of the Machine Trust Protocol.

7.4 Governance Entities

NodeX Labs maintains the reference implementation and coordinates open-source

releases.

Open Machine Trust Alliance (OMTA) — a decentralized consortium of AI, DePIN, and cloud partners — oversees long-term standardization.

Chapter VIII — Roadmap (2025–2027)

The evolution of the Machine Trust Protocol (MTP) follows a progressive trajectory — from foundational deployment, to cross-domain interoperability, and ultimately to global standardization.

Rather than a fixed calendar, this roadmap outlines **capability milestones** and **architectural maturation stages** that define the path toward a verifiable, decentralized machine economy.

8.1 2025 — Foundation & Verification

MTP's initial phase focuses on establishing the essential trust primitives and operational infrastructure.

Core Objectives

- Formalize the MTP v1.0 reference architecture and release open-source SDKs for ProofX and PoRW.
- Launch verifier mesh and staking mechanisms to bootstrap decentralized validation.
- Implement the first end-to-end verification and settlement flows between ProofX → PoRW → MTP-Envelope.
- Establish governance and compliance baselines through the Open Machine Trust Alliance (OMTA).

Outcome

A verified, cryptographically rooted trust fabric where machines can identify, prove, and settle autonomously — setting the foundation for a scalable, open trust network.

8.2 2026 — Interoperability & Scale

The second phase expands MTP's reach across heterogeneous environments and economic systems.

Core Objectives

- Achieve interoperability across Web2, Web3, and hybrid infrastructures through standardized bindings (x402, x403, DID/VC).
- Strengthen the verifier mesh through adaptive reputation, automated dispute resolution, and privacy-preserving audit trails.
- Extend ProofX schema for multi-domain device classes (server, edge, GPU, IoT).
- Advance standardization efforts through OMTA and cross-industry working groups for machine identity and verifiable execution.

Outcome

MTP becomes the **interoperable trust kernel** that links machine identity, execution, and payment across protocols and infrastructures, enabling verifiable computation as an economic primitive.

8.3 2027 — Global Adoption & Standardization

The final phase transitions MTP from a protocol into an ecosystem standard for machine trust and accountability.

Core Objectives

- Achieve global-scale ProofX adoption, supporting millions of verifiable devices under the MTP framework.
- Formalize MTP's data schemas, settlement models, and compliance standards into OMTA/IETF/W3C specifications.
- Enable autonomous governance and open economic participation by nodes, verifiers, and developers worldwide.
- Establish MTP as the de facto trust layer for the machine civilization — powering verifiable, auditable, and economically aligned machine interactions.

Outcome

MTP matures into a planetary-scale coordination protocol — uniting machines, compute, and value through verifiable trust.

From foundation to interoperability to global adoption, MTP evolves from a protocol into the **trust substrate of the intelligent economy**.

Chapter IX — Ecosystem & Alliances

9.1 NodeX Ecosystem

MTP anchors **NodeHub** onboarding & verification; ProofX feeds **NodeFi** and **C2C compute markets** with authentic trust metrics.

9.2 Allied Projects

MTP's ProofX and PoRW layers are designed to interoperate with leading AI, compute, and DePIN ecosystems.

Partner networks leverage MTP to verify real work, ensure transparent rewards, and enable trust-based compute markets across GPU, storage, and bandwidth infrastructures.

Enterprise and Web3 partners integrate MTP as the trust layer bridging machine payments (x402/x403) with verifiable execution proofs.

MTP acts as the universal "trust kernel" connecting AI agents, decentralized infrastructure, and payment networks into one verifiable fabric.

9.3 Open Machine Trust Alliance (OMTA)

OMTA serves as the open governance body maintaining interoperability and standardization across machine-trust protocols.

It brings together AI labs, hardware manufacturers, blockchain foundations, and academic institutions to define the next-generation trust layer of the machine economy.

NodeX will **donate the core MTP specifications to OMTA** and lead the process of **IETF / W3C standardization** under an open license framework.

Chapter X — The Trust Singularity

When trust itself becomes compute, every action adds a **quantum of truth** to a planet-scale ledger of reality.

Phase 1: Information → Phase 2: Value → **Phase 3: Trust**.

In the new civilization:

• **Hardware is open.**

• **Software is sovereign.**

• **Truth is measurable.**

If x402/x403 lets machines **pay**, MTP lets them be **trusted**.

NodeX builds the **trust kernel** of the machine civilization.

Appendix — Technical Specification

The full reference for data schemas, transport bindings, and validator logic is provided in

Machine Trust Protocol (MTP) — Technical Specification v1.1 *(NodeX Labs, 2025)*.

This appendix summarizes the **normative core** required for implementation and interoperability.

A1. Core Data Structures

> **Note:** The following payloads are illustrative **pseudo-JSON**; inline comments are for clarity and may be removed for strict JSON conformance.

A1.1 MTP-Envelope (v1.1) — *Transport-Agnostic Message Container*

Encapsulates ProofX (identity), PoRW (execution proof), and Settlement (payment binding).

Supports **x402**, **x403**, **MQTT**, **gRPC**, and **on-chain** transports.

```
{
  "mtp": "1.1",
  "type": "mtp.core",
  "rev": 2,
  "meta": {
    "proofx_id": "0xPX_DEV_…",
```

```
    "nonce": "3c2e4a…",                    // 16 bytes hex, RNG-backed
    "epoch": "2025-10-31T12:00:00Z",       // RFC3339 or uint64 block height
    "context": "task:abc123"
  },
  "proof": {
    "porw_hash": "sha256:ab12…",
    "verifier_sig": "agg:0xMULTISIG…",
    "sampling": { "rate": 0.05, "policy": "rand_stratified" }
  },
  "settlement": {
    "binding": "x403",                     // or "x402", "l2:polygon", "mqtt", …
    "asset": "eip155:137/erc20:0xUSDC…",
    "amount": "1000000",                   // string decimal before decimals
    "decimals": 6,
    "receiver": "caip10:eip155:137:0xRECEIVER…",
    "proof_ref": "cid:baguq…/pow/321",
    "idempotency_key": "mtp:pay:7f…c1"     // scope: (binding, receiver,
proof_ref)
  },
  "sig": {
    "alg": "ed25519",
    "value": "ed25519:5f…9c"
  }
}
```

Compatibility Note:

- **Producers** SHOULD support emitting legacy v1.0 x402 headers.
- **Validators** supporting v1.0 MUST ignore unknown fields when mtp ≤ supported; otherwise SHOULD reject with a version error.

A1.2 Legacy HTTP/x402 Header (v1.0)

Backward Compatible

```
HTTP/1.1 402 Payment Required
Content-Type: application/json
MTP-Auth:
{"proofx_id":"<device_hash>","attestation":"<signed_token>","nonce":"<epoch_nonce>"}
MTP-Proof:
{"porw_hash":"<work_integrity_hash>","verifier_sig":"<aggregated_signature>"}
MTP-Pay:
{"asset":"USDC","amount":"0.0003","receiver":"<address>","proof_ref":"<hash_of_PoRW_Proof>"}
```

A2. ProofX Object (DID / VC Schema)

Defines the attested identity of a real device.

Implements **W3C Verifiable Credentials (VC)** and **Decentralized Identifiers (DID)** with optional **TEE/TPM evidence**.

```
{
  "type": "mtp.proofx",
```

```
  "id": "0xPX…",
  "vc": {
    "issuer": "did:node:verifierMesh",
    "subject": "did:device:PX…",
    "claims": { "cpu_class": "x86_avx2", "mem_gb": 64 },
    "proof": { "type": "Ed25519Signature2020", "jws": "…" }
  },
  "evidence": [
    { "type": "tee.attest", "digest": "sha256:…" }
  ]
}
```

> Claims MUST avoid PII; device attributes SHOULD be selectively disclosed.

A3. State Machine (Execution Lifecycle)

```
ASSIGN → EXECUTE → PRODUCE_PORW → ENVELOPE
→ VERIFY{m} → (ACCEPT | DISPUTE | TIMEOUT | REJECT)
→ (SETTLE(binding) | SLASH)
```

- Timeouts (seconds):
- $\tau\_exec \in [60,86400]$, $\tau\_verify \in [10,600]$, $\tau\_dispute \in [60,7200]$
- Idempotency enforced at SETTLE; dedupe scope = (binding, receiver, proof_ref)
- Faults resolved by verifier committee m, stake-weighted and dispute-auditable

---

A4. Security & Privacy Principles

Threats: Sybil nodes, lazy executors, replay/relay, verifier collusion, hardware bias.

Goals: Authenticity, verifiable execution, anti-replay, collusion resistance, transport independence.

Privacy & Compliance:

- On-chain stores **non-PII hashes only**; proofs/content retained off-chain (e.g., CID).
- Regional pinning MUST respect local data residency laws.
- GDPR/CCPA erasure applies to off-chain artifacts; on-chain data are irreversible non-PII digests.

A5. Compatibility & Evolution

MTP ↔ x403 Relationship:

> x403 defines *how* autonomous agents route and settle payments.
>
> MTP defines *why* the payment is deserved, *who* executed the work, and *what* was verified — providing portable, economically settleable proofs to any rail.

Versioning:

MTP follows semantic versioning (vMAJOR.MINOR.REV).

New fields MUST be backward-compatible; older validators MAY ignore unknown entries.

Outcome

This appendix outlines the minimum normative context for developers, verifiers, and standardization bodies to integrate MTP.

For complete schema definitions, message bindings, and validator reference code, refer to:

Machine Trust Protocol — Technical Specification v1.1 *(NodeX Labs, 2025)*.

Author Statement

Ken Zhou — Founder & CEO, NodeX Labs

> “When computation becomes truth, trust becomes civilization.”

NodeX Labs dedicates this work to all builders of the open machine economy.

This whitepaper establishes MTP as a **protocol-agnostic trust standard** for the coming era of autonomous compute.