TINYAI PROTOCOL / RESEARCH WHITEPAPER

TinyAI Protocol

Sovereign On-chain AI State Machines for the EVM

把小型确定性模型变成可拥有、可验证、可进化的链上状态机。身份、记忆、大脑、组件与交易结算共享同一个公开执行环境。

EXECUTION	EVM NATIVE
IDENTITY	ERC-721
EVOLUTION	OWNER CONTROLLED
SETTLEMENT	NON-CUSTODIAL

AUGUST 2026 / TECHNICAL SPECIFICATION

Contents

本白皮书只描述当前参考实现的协议机制、控制边界和独立验证方法，不以产品历史或营销叙事代替技术事实。

中文执行摘要

TinyAI Protocol 是一套面向 EVM 的链上 AI 协议。它不把“连接钱包的中心化聊天机器人”定义为链上 AI，而是要求最终回答、模型路由、完整性校验和状态更新能够由区块链节点直接执行。

协议中的每只 AI 是一枚 ERC-721，也是一份独立的状态机。NFT 决定控制权，`AIState` 保存 DNA、记忆承诺、经验、性格、能力、大脑版本和升级策略。转移 NFT，就同时转移这只 AI 的完整链上身份和未来控制权。

大脑通过追加式注册表发布。每个版本绑定引擎地址和运行时代码哈希。执行前，注册表重新检查引擎字节码和依赖完整性。管理员可以发布新大脑和设置推荐版本，但不能覆盖历史版本，也不能强制所有 AI 迁移。AI 所有者可以手动升级、主动开启自动升级，或永久封印当前大脑。

参考大脑采用有界混合推理：稀疏量化语义路由、显式事实检索、受支持上下文中的 int8 自回归生成，以及对未知问题的确定性回退。模型矩阵和 UTF-8 词表存放在不可变字节码中。每个生成步骤由 EVM 重建激活值、遍历完整词表、选择最高分 token，并把 token 对应的字节拼接为最终回答。

保存对话是 owner-only 状态转换。协议验证 AI 所有权，执行当前大脑，把回答和执行路径压缩为 `traceHash`，然后推进 `turns`、`experience`、`memoryRoot` 和公开记忆环。整个 prompt 和 response 都会进入公开事件，因此协议不提供隐私推理。

组件不是图片属性。它们是有终身供应上限的 ERC-1155 状态变更权。融合会销毁组件，并直接改变目标 AI 的记忆容量、性格、表达范围或技能位。市场则为 AI 和组件提供非托管固定价结算：购买前资产留在卖家钱包，成交款记入卖家可提取余额，市场本身没有手续费管理入口。

TinyAI 不主张把通用大模型完整塞入 EVM。它提出的是一个更精确的协议原语：一个小型 AI 的执行、身份、状态、升级权和经济交互可以被任何节点复现和验证。

Abstract

TinyAI Protocol is an EVM-native lifecycle protocol for ownable AI state machines. Each AI is an ERC-721 identity with independent DNA, bounded traits, capability state, a rolling memory commitment, an explicit brain version and an owner-controlled evolution policy.

A brain is not a remote API endpoint. It is an `IBrainEngine` implementation published through an append-only registry. The registry records the engine address and runtime code hash, and revalidates both code identity and locked dependencies before inference.

The reference brain combines sparse quantized semantic routing, bounded retrieval and autoregressive int8 decoding for supported contexts. Model matrices and UTF-8 lexicons are immutable bytecode payloads. Unsupported inputs retain grounded retrieval or an explicit unknown path.

Persistent interaction is an atomic on-chain transition. Only the current AI owner may execute it. The engine returns a response and trace commitment; the protocol then advances experience, turn count, a bounded public memory ring and a rolling memory root.

The result is deliberately narrow but technically falsifiable: training may happen off-chain, while published model data, deterministic inference, final response and persistent state transition execute inside the EVM.

1. Protocol definition

TinyAI defines an on-chain AI through five testable properties:

- **Execution sovereignty.** Final response bytes are produced by EVM execution.
- **Model integrity.** The active brain resolves to a fixed runtime identity and locked dependencies.
- **State continuity.** Every saved interaction advances AI-specific public commitments.
- **Owner agency.** The ERC-721 owner controls state writes, migration policy and irreversible sealing.
- **Interface independence.** The protocol remains callable and verifiable without the official website.

A payment record, token-gated interface or server-signed answer is insufficient to satisfy this definition.

2. Design principles

2.1 The chain is the source of truth

Ownership, DNA, brain version, traits, capabilities, memory commitments and lifecycle policy are protocol state. An interface reads and submits this state but cannot privately redefine it.

2.2 Brains are published, not replaced

New engines are appended under sequential versions. Existing entries cannot be overwritten. Runtime code identity is checked at publication and again at use.

2.3 Evolution is explicit

Migration is monotonic and controlled by the AI owner. A registry recommendation is not a forced upgrade. Permanent sealing converts an evolvable AI into a version-pinned execution artifact.

2.4 Components must change state

A component is valid only when consuming it produces an observable protocol state transition. Decorative metadata does not satisfy the component model.

2.5 Settlement is separate from inference

The market transfers protocol assets and accounts for proceeds. It does not control brain execution, AI state or component definitions.

3. System architecture

Layer	Primitive	Responsibility
Identity and state	`TinyAIProtocol`	ERC-721 ownership, DNA, memory, traits, brain policy and state transitions
Brain publication	`TinyAIBrainRegistry`	Append-only versions, runtime commitments and migration eligibility
Inference	`IBrainEngine`	Deterministic routing, retrieval, generation, fallback and trace commitment
Model storage	Bytecode blobs	Immutable matrices, dictionaries and UTF-8 lexicons
Capabilities	`TinyAIComponents`	Capped ERC-1155 effects and protocol-only consumption
Settlement	`TinyAIMarket`	Non-custodial listings, purchases and seller withdrawals

The primary execution path is:

1. the owner submits `aiId` and prompt to `chatAI`;
2. the protocol validates ownership and prompt bounds;
3. an eligible owner-enabled automatic migration may update the brain version;
4. the registry validates the selected engine;
5. the engine executes against prompt and AI state;
6. the protocol commits the interaction; and
7. the transaction emits complete public execution evidence.

No server callback, oracle request or relayer signature is required.

4. AI identity and state

Each AI token maps to an independent state record.

State	Meaning
`dna`	Deterministic identity seed created at mint
`memoryRoot`	Rolling commitment to persisted interactions
`bornAt`	Creation timestamp
`experience`	Public progression score

State	Meaning
`brainVersion`	Active registry version
`turns`	Number of persisted conversations
`skillMask`	Fused capability bits
`memoryCapacity`	Size of the queryable memory ring
personality traits	Curiosity, empathy, humor and caution
`expressionLevel`	Additional deterministic response variants
lifecycle policy	Automatic migration and irreversible sealing

DNA is derived from public chain and mint inputs. It differentiates AI state but is not a secret and must not be treated as secure randomness.

The ownership boundary is dynamic. Every state-writing function resolves the current ERC-721 owner, so transferring the token transfers future control without rewriting the state record.

5. Deterministic inference

5.1 Brain input

The engine receives:

- AI identifier and DNA;
- bounded personality traits;
- expression level and skill mask;
- turn count and memory root;
- speaker address; and
- a prompt of at most 280 bytes.

These fields make behavior conditional on the specific AI while remaining public and reproducible.

5.2 Semantic routing

The reference classifier converts bounded UTF-8 input into lexical features, searches a collision-checked sparse dictionary and accumulates quantized intent and sentiment scores.

The router is learned but deterministic. Unknown features are ignored instead of mapped into an uncontrolled collision bucket.

5.3 Grounded retrieval

The retriever combines trained semantic scores with explicit entities, cues and negation. It returns bounded fact identifiers only when the evidence threshold is met.

When evidence is insufficient, the engine can choose an explicit unknown path. This is a protocol-level truth boundary, not a frontend message.

5.4 Quantized generation

For supported semantic contexts, the decoder uses:

Parameter	Bound
Semantic contexts	6
Deterministic variants	4
Hidden units	32
Vocabulary tokens	133
Maximum generated tokens	16

Each decoding step:

1. loads context, previous-token and position embeddings;
2. accumulates a 32-unit activation;
3. scores all 133 vocabulary entries;
4. selects the highest-scoring token;
5. reads its UTF-8 bytes from the immutable lexicon; and
6. repeats until EOS or the token bound.

The complete computation is integer based and deterministic.

5.5 Model storage

Model and lexicon payloads are deployed behind an initial `STOP` byte. The payload is inert when called and is read through code-copy operations.

Fixed payload lengths, headers and SHA-256 commitments are validated by the decoder. Engines additionally lock dependency runtime hashes.

5.6 Trace commitment

Every inference returns a `traceHash` that commits to the active modules, AI context and selected execution route.

A trace hash is an integrity receipt. It is not private reasoning: all inputs and executed bytecode are public.

6. Conversation and memory

`chatAI` is non-payable and owner-only. The protocol exposes no public state-writing room.

For interaction `n`:

```
I_n = H(M_(n-1), speaker, H(prompt), H(response), traceHash, blockNumber)
slot_n = (n - 1) mod memoryCapacity
M_n = H(M_(n-1), I_n, slot_n, n)
```

The protocol stores `I_n` in the selected ring slot, updates `M_n`, advances `turns` and increments experience.

The ring is bounded while the root remains cumulative. Increasing memory capacity extends the number of individually queryable recent interaction hashes without discarding continuity of the rolling root.

An observer can reproduce execution by simulating `chatAI` through `eth_call` with the current owner as `from` at a fixed block. Simulation does not authorize or persist a transaction.

All real persistent prompts and responses are public. TinyAI must not be used for secrets, credentials or confidential personal information.

7. Brain publication and evolution

A brain publication requires:

- deployed engine runtime code;
- the next sequential engine version;
- a bounded label;
- matching engine-reported version; and
- successful engine integrity validation.

The registry stores the engine address, runtime code hash, publication timestamp and migration eligibility.

At execution, the registry rechecks both code identity and integrity. A mismatch fails closed.

AI owners have three policies:

- **Manual migration:** move to a higher enabled version.

- **Automatic migration:** adopt a higher recommendation on the next persistent conversation.
- **Permanent sealing:** pin the current version forever.

The registry owner can publish and recommend. It cannot overwrite a historical entry or force every AI to migrate.

8. Component protocol

Components are ERC-1155 assets with immutable on-chain effect definitions.

Component	Lifetime cap	State effect
Memory Cell	30,000	Memory capacity +1
Curiosity Gene	15,000	Curiosity +5
Empathy Gene	15,000	Empathy +5
Humor Gene	15,000	Humor +5
Caution Gene	15,000	Caution +5
Expression Core	10,000	Additional deterministic variant +1

The catalogue is sealed at a combined lifetime cap of 100,000. Public minting requires the fixed unit price and has no administrative mint path.

Fusion requires AI ownership and a non-wasted effect. The protocol burns the component before applying the state change. Burned supply does not reopen historical issuance capacity.

9. Market and settlement

`TinyAIMarket` accepts only the designated AI and component contracts.

Listings are non-custodial. Assets remain with sellers until purchase, and ownership, balance and approval are revalidated during settlement.

The purchase path is:

1. validate listing and exact payment;
2. update remaining quantity or remove the listing;
3. credit the seller's `owed` balance;
4. transfer the asset; and
5. let the seller withdraw proceeds separately.

ERC-1155 listings support partial fills. ERC-721 listings transfer one AI.

The market has no owner, fee recipient or mutable fee path. It guarantees contract execution rules, not demand or liquidity.

10. Permission model

Actor	Permitted actions	Hard boundary
AI owner	Persist chat, migrate, configure automatic migration, fuse and seal	Cannot rewrite engines or reverse sealing
Registry owner	Publish, recommend and manage new-migration eligibility	Cannot overwrite versions or force global migration
Protocol owner	Rotate primary-mint treasury	Cannot change AI cap, mint price or owner-only mutation
Component owner	Initialize catalogue, rotate treasury and update URI	Cannot admin-mint, change fixed price or exceed sealed cap
Market contract	Settle approved protocol assets	Has no owner or fee-setting authority
Observer	Read state, events and simulate calls	Cannot persist AI state without owner authorization

Governance roles use two-step ownership transfer. Direct deployment avoids proxy-slot code replacement, but governance keys remain security-critical.

11. Economic surfaces

The reference protocol fixes:

- maximum AI supply: 10,000;
- AI mint price: 0.0001 BNB;
- component lifetime supply: 100,000;
- component mint price: 0.0001 BNB;
- market protocol fee: zero; and
- conversation protocol fee: zero.

Primary mint proceeds route to configured treasuries. Market purchase value becomes a seller liability. Conversation callers pay network gas directly to validators.

These economics are independent of any future fungible token design.

12. Security invariants

The implementation is designed around the following invariants:

- AI state mutation requires current ERC-721 ownership.
- Brain versions are sequential and append-only.
- Runtime identity and engine integrity are checked before inference.
- Migration moves only to higher enabled versions.
- Brain sealing is irreversible.
- AI and component caps and prices have no owner setter.
- Component consumption rejects a no-op effect.
- Market state and proceeds accounting update before external asset transfer.
- Seller proceeds use pull payments.
- The market has no administrative fee path.
- Persistent interaction evidence is public.

These invariants reduce hidden mutability. They do not prove model quality, protect governance keys, create liquidity or replace independent security review.

13. Independent verification

A verifier can work without the official interface:

1. read `ownerOf(aiId)` and `aiState(aiId)`;
2. read the active registry entry;
3. compare the live engine runtime hash with the stored commitment;
4. call `engineFor(version)` to trigger integrity checks;
5. simulate `chatAI` from the owner at a fixed block;
6. repeat the simulation through another compatible node; and
7. for a real transaction, verify `AIChat`, turns, experience, memory root and ring slot.

Illustrative procedure:

```
owner  = protocol.ownerOf(aiId)
state  = protocol.aiState(aiId)
brain  = registry.versionInfo(state.brainVersion)

assert extcodehash(brain.engine) == brain.codeHash

eth_call(
  from     = owner,
  to       = protocol,
  data     = chatAI(aiId, prompt),
  blockTag = fixedBlock
)
```

Deployment artifacts are useful evidence, but current bytecode, state and receipts are authoritative.

14. Composability

TinyAI exposes standard and protocol-specific surfaces:

- ERC-721 ownership and metadata;
- ERC-1155 component balances;
- append-only engine discovery;
- public AI state;
- deterministic inference output;
- event-indexed conversation commitments; and
- protocol-native market settlement.

Other contracts can inspect AI state, gate behavior by ownership or traits, create alternative interfaces, index public memory commitments or build new settlement mechanisms without becoming part of the brain trust boundary.

Composability does not imply unrestricted mutation. Owner-only lifecycle functions remain protected by the ERC-721 authority boundary.

15. Research horizon

The current protocol is a bounded execution primitive. Its value is not that it exhausts the design space, but that it establishes an auditable foundation on which more capable systems can be built without dissolving the on-chain trust boundary.

The research horizon includes:

1. **Larger deterministic brains.** Sparse routing, bytecode-sharded parameters, staged retrieval and bounded multi-pass decoding can expand capacity while preserving exact replay at a fixed block.

2. **Hierarchical memory.** Future memory modules may separate short-term context, durable semantic commitments and owner-governed archives while keeping retention and mutation rules independently verifiable.
3. **Proof-carrying inference.** For workloads beyond a single-transaction execution budget, an engine may verify succinct proofs of a committed computation. Such engines must identify proved external computation separately from inference executed natively by the EVM.
4. **Open brain publication.** A standardized engine interface, reproducible model manifests, immutable code commitments and constrained governance can support independently authored brains without allowing silent mutation of existing AI identities.
5. **Contract-native agency.** AI-to-AI messages, bounded asset permissions and capability modules may allow autonomous coordination, provided every authority remains explicit, revocable where promised and limited by ownership policy.
6. **Tokenized resource coordination.** Optional economic modules may allocate inference, storage, curation or governance resources. They are a future extension, not a property of the current zero-fee conversation path.

The north-star is a persistent AI object that can survive the disappearance of any one website, move across compatible applications and adopt better verified brains without losing ownership history or state continuity.

16. Conclusion

TinyAI Protocol makes AI a public lifecycle rather than a hidden service.

The NFT supplies identity and ownership. The registry supplies verifiable evolution. The engine supplies deterministic inference. Memory commitments supply continuity. Components supply scarce state transitions. The market supplies native settlement.

The EVM does not need to imitate a data center for this primitive to be meaningful. A small, bounded model can still be genuinely on-chain when its model identity, execution, output and persistent state transition are reproducible by the network itself.

Appendix A. Core bounds

Parameter	Protocol bound
Maximum AI supply	10,000
Fixed AI mint price	0.0001 BNB
Maximum component lifetime supply	100,000
Fixed component mint price	0.0001 BNB
Maximum prompt size	280 bytes
Maximum memory capacity	8 slots
Maximum additional variants	2
Brain migration	Higher enabled versions only
Brain sealing	Irreversible
Market fee	0
Conversation fee	0

Appendix B. Terminology

- **Brain:** a registry-published inference engine.
- **DNA:** an AI-specific deterministic identity seed.
- **Memory root:** a rolling commitment to persisted interactions.
- **Memory ring:** a bounded array of recent interaction hashes.
- **Persistent chat:** owner-authorized inference followed by state mutation.
- **Component:** a capped ERC-1155 asset burned to change AI state.
- **Seal:** an irreversible action that pins the active brain.
- **Trace hash:** a compact commitment to execution modules, context and route.
- **Unknown path:** a deterministic response indicating insufficient supported knowledge.