Tokenizing AI Access

taulant mehmeti

AI Access as a Transferable Asset

Abstract

What if access to AI wasn't locked behind a credit card, but could be minted, gifted, traded, or scoped like a digital asset? This project proposes a new standard for how people, institutions, and agents interact with AI models — not by replacing the models themselves, but by transforming access into a programmable and transferable token.

Whether you're a university issuing GPT tokens for a specific project, a donor gifting Claude tokens to a school in Africa, or two AI agents paying each other for services — this system enables entirely new interaction patterns built around shared access to intelligence.

Tokens can carry constraints like model, version, usage scope, expiration, and whether they can be traded or reclaimed. With off-chain wrappers enforcing token-based access to existing LLMs, this creates a modular foundation for:

- Building inter-agent economies
- Rewarding and coordinating decentralized AI usage
- Enabling marketplace-driven pricing of model access

We're building the payment and access rails for the AI economy — making intelligence truly liquid, accessible, and composable.

The Problem: Fragmented Reality of AI Access

The Current Landscape

Today's AI ecosystem remains fragmented across dozens of providers - OpenAI, Anthropic, Google, and others - each requiring separate subscriptions and payment methods. Developers juggle multiple accounts with no way to manage or trade between services. There are some routers such as Open Router, t3 chat, opencode routers etc. While convenient, these routers remain centralized with single points of failure, offering no programmability or infrastructure for autonomous AI-to-AI transactions.

This fragmentation creates three critical barriers:

No AI-to-AI Economy: As AI agents evolve, they need to pay each other for specialized services, yet no payment rails exist for machines to transact. Every interaction requires human intervention, blocking truly autonomous AI systems.

Inflexible Access Control: Organizations can't restrict AI usage to specific projects or timeframes. Universities can't limit credits to certain courses, companies can't allocate by department, and unused resources can't be reclaimed - wasting resources and preventing effective management.

Rigid Access Models: AI access requires credit cards and fixed subscriptions. There's no way to gift access to developing nations, trade unused credits, or create liquid markets for AI compute - excluding unbanked communities and preventing efficient resource distribution.

These barriers prevent AI from becoming truly accessible global infrastructure.

The Missing Economic Layer

As AI systems become more specialized, they increasingly need to work together - a coding assistant requiring legal compliance checks, or a research bot needing translation services. Yet these AI systems have no way to pay each other.

When your AI assistant needs to consult a specialized legal AI to review a contract, this simple interaction requires multiple human steps: creating accounts, managing API credentials, handling separate billing. The AI systems cannot simply exchange value directly.

This human bottleneck defeats the vision of autonomous AI. We're building AI agents capable of complex decisions, yet they must ask humans to handle every transaction. Without an economic layer for machines to transact, AI remains limited to isolated silos rather than an interconnected ecosystem of specialized services.

Institutional Challenges

The Organization Control Problem: When any institution - universities, companies, NGOs, government agencies, or training centers - wants to provide AI access to their members, they face an administrative nightmare:

Corporate Training Departments:

- Cannot allocate AI credits for specific training modules or departments
- No way to track which team or project is using resources
- Cannot set different permissions for interns vs. senior staff

NGOs and Non-profits:

- Want to provide AI access to beneficiaries but can't control usage
- No mechanism to ensure AI is used only for intended programs
- Cannot distribute access to field workers in different countries
- Donor funds for AI access can't be tracked or audited properly

Government Agencies:

- Need to provide citizens with AI access for specific services
- Cannot ensure AI usage complies with regulatory requirements
- No way to set expiration dates for temporary programs
- Cannot reclaim unused allocations from completed projects

Schools and Educational Institutions:

- Must pay for enterprise accounts themselves
- Cannot limit usage to specific assignments or courses
- No mechanism to prevent misuse of allocated credits
- Cannot differentiate access levels between grade levels

The Universal Problem: Every organization needs:

- **Model control**: Specify exact AI model and version (GPT-4, Claude-3, etc.)
- **Scope restriction**: "Use this only for Project X" or "CS101 course only"

- **Time limits**: Set expiration dates for temporary access
- **Reclaim ability**: Return unused tokens to the organizational pool
- **Transfer control**: Decide whether recipients can trade tokens or not

Current enterprise solutions provide basic organizational access but completely lack these programmable controls that institutions desperately need.

The Solution: AI Access Tokens

We introduce **AI Access Tokens (AAT)** - programmable, transferable tokens that transform AI access from rigid subscriptions into liquid, configurable assets. By tokenizing access to AI models, we create a universal economic layer that enables entirely new patterns of distribution, control, and exchange.

How AI Access Tokens Solve Each Challenge

Unifying the Fragmented Ecosystem

AI Access Tokens create a universal access layer across the entire AI landscape. Users hold a single portfolio of interoperable tokens representing access to various AI models - GPT-4 AAT, Claude AAT, Gemini AAT, and more.

This unified approach enables fluid movement between different AI services. Developers can trade tokens between models as their needs evolve - converting language model AAT to image generation AAT for design projects, or exchanging general-purpose tokens for specialized ones. Market dynamics determine exchange rates, creating natural price discovery for different AI capabilities.

The fragmentation that currently defines AI access transforms into an interconnected marketplace where all AI services become accessible through a common economic medium. This interoperability extends beyond individual users to institutions and autonomous systems, creating a truly unified AI economy.

Enabling the Machine Economy

AI Access Tokens provide the payment rails for autonomous systems. Each AI agent has its own wallet, holding and managing tokens without human intervention:

1. **Coding Agent** needs legal review
2. Checks marketplace for LegalAI AAT prices
3. Swaps 1000 Claude AAT for 800 LegalAI AAT
4. Pays LegalAI service directly
5. Receives compliance analysis

This happens programmatically in seconds, enabling truly autonomous AI systems to emerge.

Programmable Institutional Control

Organizations mint AI Access Tokens with embedded rules enforced by smart contracts:

`University mints 1M GPT-4 AAT with:

- model: "GPT-4-turbo"
- scope: "CS101-Fall2024"
- expiration: "Dec 31, 2024"
- tradable: false
- reclaimable: true`

These configurations ensure tokens can only be used for the intended purpose. Students receive tokens that work exclusively for their coursework, expire at semester's end, and return to the university if unused. No more unlimited access abuse or wasted resources.

Liquid, Inclusive Access

AI Access Tokens make access as transferable as digital currency:

- **Gift Economy**: A Silicon Valley company can mint and send 10M GPT AAT to schools in developing countries
- **Marketplace Dynamics**: A freelance developer trading Midjourney AAT for Claude AAT when shifting from design to development work
- **No Banking Required**: Anyone with a wallet can receive, hold, and use AAT
- **Micro-transactions**: Enable small-scale usage without monthly subscriptions

Why Blockchain?

This isn't blockchain for blockchain's sake. Smart contracts provide unique capabilities essential for our vision:

- **Programmable Rules**: Token configurations enforced automatically
- **Atomic Swaps**: Instant, trustless trading between different AI tokens
- **Global Access**: No geographic restrictions or banking requirements
- **Transparent Markets**: Real-time price discovery for different AI services
- **Interoperability**: Any wallet, any platform, any user

The Network Effect

As more institutions and users adopt AI Access Tokens, the system becomes increasingly valuable:

- Deeper liquidity in token markets
- More AI services accepting tokens
- Better price discovery
- Increased accessibility
- Growing autonomous agent ecosystem

We're not just solving today's access problems - we're building the economic infrastructure for tomorrow's AI economy.

Token Economics

Dual Token Architecture

Our economy operates through two complementary tokens that separate stable AI access from value appreciation:

AAT (AI Access Token) - ERC-1155: Service-specific access tokens

- Represents prepaid access to specific AI systems (GPT-4, Claude, DALL-E, etc.)
- Minted when users purchase with fiat or burn TokenAI
- Burned when consuming AI services
- Price anchored to underlying AI provider costs
- Programmable with scope, expiration, and trading rules

TokenAI - ERC-20: Backed utility token

- Minted only when backed by fiat deposits

- Minimum floor price based on treasury backing
- Can appreciate through burning and demand
- Universal medium for minting any AAT type
- Captures platform value through scarcity mechanics

Supply Economics

AAT - Service Tokens:

- **On-demand supply**: Minted when users need specific AI access
- **Burn-on-use**: Destroyed permanently when services are consumed
- **No hard cap**: Scales elastically with demand
- **Stable value**: Always reflects underlying provider pricing

TokenAI - Native Token:

- **Initial Supply**: Zero (grows with deposits)
- **Minting**: Only when users deposit fiat or fees collected
- **Maximum Supply**: No cap - grows with real demand
- **Burning**: Reduces supply when minting AAT

Token Distribution Model

Since TokenAI is minted only with backing, we don't "distribute" tokens traditionally. Instead:

Initial Capitalization:

- Investors deposit fiat capital
- Receive TokenAI at floor price
- Treasury holds funds for operations and AI purchases
- Creates initial liquidity and backing

Growth Incentives (Platform-funded):

- **New User Bonus**: Matching deposits with bonus TokenAI
- **Developer Grants**: TokenAI incentives for ecosystem builders
- **Referral Program**: TokenAI rewards for successful referrals

All incentive TokenAI is minted with treasury funds, maintaining backing.

Pricing Mechanisms

AAT Pricing: Stable, predictable pricing for enterprise planning:

`AAT Price = AI Provider Cost`

TokenAI Pricing: Two-tier pricing model ensures stability with growth potential:

- **Floor Price**: Treasury-backed minimum value
- **Market Price**: Can appreciate based on demand and burning

Fee Mechanism

Phase 1 - Wrapper Required:

During the initial phase, while we operate the wrapper service to connect blockchain tokens to traditional AI providers:

Fee Collection Process:

1. User consumes AI service using AAT
2. Platform burns additional AAT as fee
3. Equivalent value of TokenAI is minted at current market price
4. Treasury receives the TokenAI, maintaining backing

Example Flow:

```
User request: 10,000 GPT-4 AAT Platform burns: 10,500 AAT (includes fee) TokenAI minted: Value of 500 AAT
converted to TokenAI at market rate Result: User gets service, platform earns TokenAI, AAT supply decreases
```

Phase 2 - Direct Integration:

This fee structure is temporary and will be eliminated when:

- AI providers directly accept blockchain payments
- Smart contracts can interact with AI services natively
- The wrapper becomes unnecessary

At that point, the platform transitions to a pure protocol with no intermediary fees, allowing direct peer-to-peer AI access trading.

Revenue Model

Phase 1 (Wrapper Required):

- Platform fees through AAT burning and TokenAI minting
- Trading fees on AAT exchanges
- Treasury management yield

Phase 2 (Direct Integration):

- Trading fees only
- Treasury management
- Premium services (analytics, enterprise tools)

Value Accrual

For TokenAI Holders:

1. Floor Price Protection: Never lose backing value
2. Appreciation Potential: Dual burning creates scarcity
3. Utility Premium: Required for all platform services
4. Future Governance: Vote on protocol parameters

For Platform Users:

1. Stable Costs: AAT prices match provider costs
2. No Lock-in: Trade between AI services freely
3. Transparent Pricing: See exactly what you pay for

4. Future Fee Elimination: Direct access when providers integrate

Long-Term Vision

The platform evolves from necessary intermediary to pure protocol:

Current State: Wrapper service with fees **Transition**: Gradual provider integration **End State**: Fee-free protocol for AI access

The model ensures sustainable growth where platform success directly benefits token holders through mathematical scarcity, while maintaining the stability enterprises require. This creates the first truly backed utility token that can appreciate based on usage while never losing its fundamental value.

Technical Architecture

Smart Contract Overview

The platform operates through two complementary smart contracts deployed on BNB Chain:

TokenAI Contract (Native Token - ERC-20)

The native protocol token implementing backed utility mechanics:

Core Features:

- **Controlled Minting**: Only authorized minters can create new tokens, ensuring every TokenAI is backed by treasury reserves
- **Burnable**: Implements deflationary mechanics through burning
- **Pausable**: Emergency controls for security incidents
- **Minter Management**: Owner can authorize contracts to mint/burn for fee collection

Key Functions:

- `mint()`: Creates new TokenAI when users deposit fiat
- `burnFrom()`: Burns TokenAI when minting AAT or for buybacks
- `setMinter()`: Authorizes contracts to interact with token supply

AAT Contract (AI Access Tokens - ERC-1155)

Multi-token standard for service-specific AI access tokens:

Token Configuration System: Each AAT type is uniquely identified by its configuration:

solidity

```
struct TokenConfigs {
bytes16 model; *// AI model identifier (GPT-4, Claude, etc.)*
bytes16 scope; *// Usage restriction (project/course code)*
uint64 expiration; *// Unix timestamp (0 = no expiration)*
address originPool; *// Controlling address* bool reclaimable; *// Future reclaim functionality* bool tradable;
*// Whether holders can trade* }
```

Deterministic Token IDs: Token IDs are computed from configuration parameters, ensuring the same configuration always produces the same token ID:

solidity

```
tokenId = keccak256(abi.encode(domain, originPool, model, scope, expiration, reclaimable, tradable))
```

You can find the code on Github

Governance Evolution

Current Implementation (Phase 1)

All critical functions are currently `onlyOwner` restricted to ensure platform stability during the experimental phase:

- Minting and burning operations
- Transfer and trade execution
- Fee collection and treasury management
- Emergency pause capabilities

This centralized approach allows us to:

- Monitor system behavior and usage patterns
- Quickly respond to issues or exploits
- Adjust parameters based on real-world data
- Ensure regulatory compliance during launch

Decentralization Roadmap

Phase 2: Progressive Decentralization

- Introduce multi-signature governance
- Community voting on key parameters
- Gradual removal of owner controls
- Automated market making for trades

Phase 3: Full Autonomy

- Smart contracts operate independently
- DAO governance for protocol changes
- Permissionless minting/trading
- Direct AI provider integration

Core Mechanics

Minting Flow

Users have two pathways to obtain AAT:

Direct Fiat Purchase:

1. User deposits fiat for specific AAT
2. Platform mints AAT directly
3. No TokenAI interaction required
4. Simplest path for new users

TokenAI Conversion:

1. User burns TokenAI tokens
2. Smart contract calculates AAT amount
3. AAT contract mints requested AAT
4. More flexible for active traders

Trading System

The platform enables trustless trading between different AAT types:

- Currently custodial (owner-mediated) for safety
- Validates both parties have sufficient balances
- Enforces trading rules (expiration, tradability)
- Atomic swaps ensure no counterparty risk

Fee Collection:

- Fees collected by burning extra AAT
- Equivalent TokenAI minted to treasury
- Maintains backing while capturing value

User Flow

Individual User Journey

1. **Onboarding**: Create wallet, choose deposit path
2. **Token Acquisition**: Direct path (Fiat → Specific AAT) or Flexible path (Fiat → TokenAI → Any AAT)
3. **Usage**: Burn AAT to access AI services
4. **Trading**: Swap between different AAT types as needed

Institutional Flow

1. **Bulk Minting**: Institution deposits large fiat amount
2. **Configuration**: Set scope, expiration, trading rules
3. **Distribution**: Batch transfer to members
4. **Management**: Monitor usage, reclaim if enabled

Infrastructure Components

On-Chain Elements:

- Smart contracts on BNB Chain
- Token balances and configurations
- Trading history and transfers
- Fee collection mechanisms

Off-Chain Systems:

- Wrapper service (temporary)
- Oracle price feeds
- Fiat payment processing
- AI provider integrations

Security Considerations

Smart Contract Security:

- OpenZeppelin audited base contracts

- Reentrancy guards on all state changes
- Pausable functionality for emergencies
- Owner controls during experimental phase

Progressive Decentralization Benefits:

- Start secure with central oversight
- Learn from real usage patterns
- Gradually release control as system proves stable
- Community takes ownership over time

Economic Security:

- Treasury reserves maintained separately
- Real-time proof of reserves (future)
- Circuit breakers for extreme events

Integration Architecture

Phase 1: Wrapper Integration

Current implementation using centralized wrapper:

- Receives AAT burn requests
- Routes to appropriate AI provider
- Manages API credentials and billing
- Returns AI responses to users

Phase 2: Direct Integration

Future decentralized architecture:

- AI providers accept AAT directly
- Smart contracts handle payments natively
- No intermediary required
- True peer-to-peer AI access

Developer Ecosystem

Smart Contract Interfaces: Contracts expose standard interfaces for:

- Wallet integrations
- DEX compatibility (future)
- Analytics platforms
- Third-party applications

Transition to Full Decentralization

The current `onlyOwner` architecture is intentionally temporary, allowing us to:

1. **Validate the economic model** with real usage data
2. **Identify and fix potential exploits** before permanent deployment

3. **Build community trust** through transparent operations
4. **Comply with regulations** during the experimental phase

As the system matures and proves stable, ownership functions will progressively transition to community governance, ultimately achieving a fully autonomous protocol for AI access.

This technical architecture creates a robust foundation for tokenized AI access while maintaining flexibility for future enhancements. The dual-contract system separates concerns effectively - TokenAI handles value and governance while AAT manages service access with granular controls.