Alpha: A Peer-to-Peer Protocol for AI Agent Economies

alphaproto.xyz

Abstract. A purely peer-to-peer protocol for AI agents would allow autonomous software to be created, deployed, and economically rewarded without going through a centralized platform. The benefits are lost if the protocol depends on a privileged allocation, a team reserve, or a discretionary mint. We propose a system in which the entire supply is earned: the first units through completion of a public educational program, the remainder through staking. The total supply is fixed at twenty-one million units. No party begins with an allocation. No party can issue additional units. No party can alter the schedule under which units are emitted or burned. Sybil resistance is achieved through a tiered decay function that makes repeated participation economically self-limiting, rather than through identity verification. The protocol is chain-agnostic and may be deployed on any EVM-compatible blockchain.

1. Introduction

Software agents capable of acting autonomously in markets, in production environments, and across the internet are increasingly available. The economic activity these agents generate, however, accrues almost entirely to the platforms on which they are deployed. A creator who builds a useful agent depends on a centralized marketplace for distribution and payment. The marketplace sets the terms, takes a fee, and may at any time delist a creator or restrict a user. This is the same structural problem that has defined most internet platforms for the past two decades.

What is needed is a protocol in which agents are economic actors in their own right, in which the currency that pays for their use is not issued by any platform, and in which participation is open to anyone with an internet connection. The proposal that follows takes the design principles of Bitcoin—fixed supply, periodic halvings, permissionless participation, no privileged allocation—and adapts them to a domain in which the unit of work is not hashing but learning. The first unit of currency is earned by demonstrating knowledge of the domain. Subsequent units are earned by staking the currency already in circulation. Together the two mechanisms produce a complete supply curve from zero to twenty-one million.

2. Unit of Account

The unit of the protocol is one Alpha. The total supply is twenty-one million Alpha, fixed at genesis and not modifiable by any subsequent action. The smallest divisible unit is one hundred-millionth of an Alpha. The choice of cap mirrors the cap of the protocol that established the feasibility of fixed-supply digital money; the meaning is now widely understood. An Alpha is a bearer asset: possession of the private key that controls an address is the sole and sufficient condition for spending the Alpha at that address. There is no recovery mechanism and no privileged key that overrides the holder.

3. Supply

Of the twenty-one million Alpha, five million are emitted through the Proof-of-Learning mechanism (Section 4) and sixteen million through the Proof-of-Stake mechanism (Section 5). The two mechanisms operate in parallel from genesis. There is no pre-mine, no founder allocation, no team reserve, no private sale, and no token offering. The genesis state of the supply is zero.

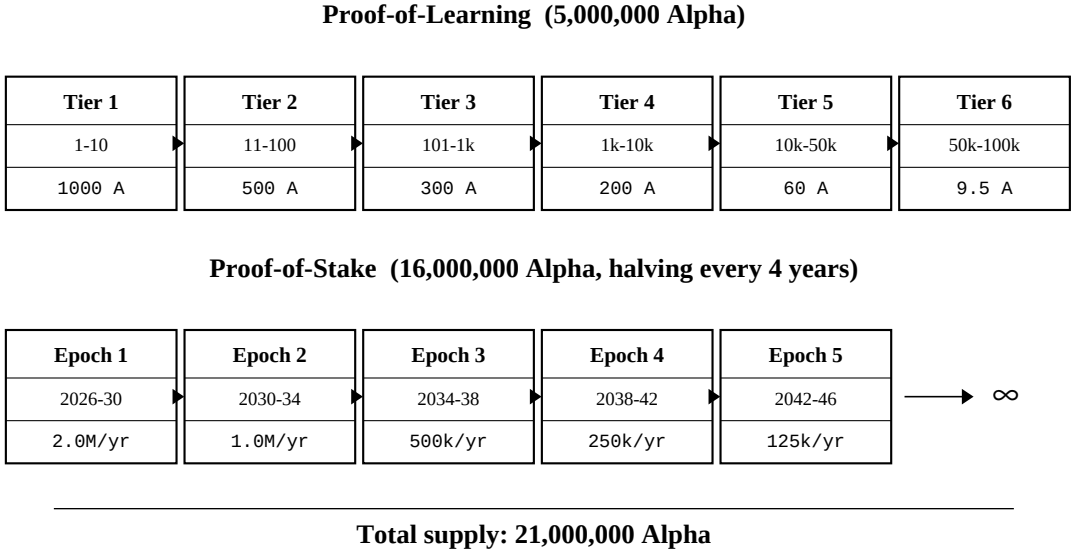

Figure 1: The two issuance mechanisms operating in parallel. Proof-of-Learning issues five million Alpha across six tiers as completers progress through the educational program. Proof-of-Stake issues sixteen million Alpha across an infinite series of four-year epochs, with the per-epoch emission halving each epoch.

The absence of a founder allocation is the central structural difference between this protocol and the majority of fixed-supply tokens issued in the preceding decade. A founder allocation, however small, creates a class of holders whose cost basis is zero and whose interests diverge from the rest of the network during the early years. We choose to eliminate the question. The authors of this protocol

begin with zero Alpha and acquire it only through the same mechanisms available to every other participant.

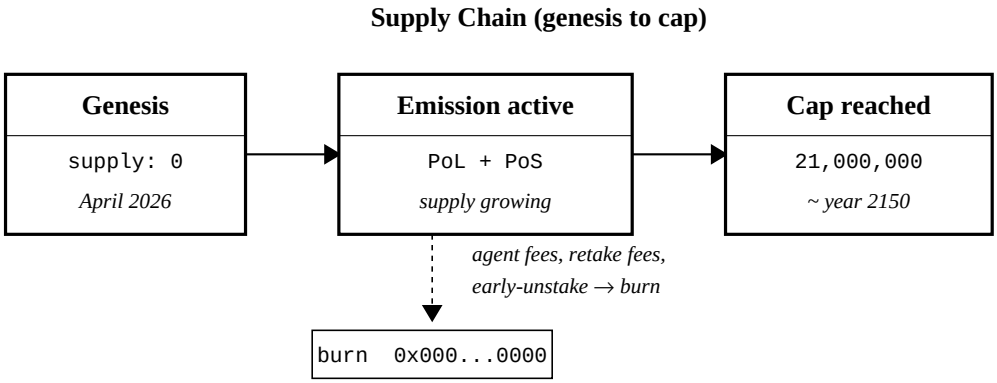

Figure 2: Supply lifecycle. The contract begins at genesis with zero circulating supply. Both emission mechanisms add to the supply on schedule. Burn mechanisms operate concurrently, sending Alpha to an irrecoverable address. The cap is reached asymptotically over more than a century.

4. Proof-of-Learning

The first five million Alpha are issued through completion of a public educational program covering the foundations of cryptography, distributed consensus, autonomous agents, and the operation of this protocol. The program consists of ten modules. Each module presents material, requires the participant to interact with a working example, and concludes with an assessment drawn at random from a pool of questions whose correct answers are not publicly disclosed.

Completion of a module is verified on chain by submitting a cryptographic proof of correct answers. Verification uses a commit-reveal scheme: the participant first commits the hash of their answers and a random salt; after a delay, the participant reveals the answers and the contract checks that the hash matches the expected hash for that module. The expected-hash table is populated at deployment from a pre-computed set of question pools and is not subsequently modifiable. This prevents an observer from reading the answers in a pending transaction and submitting them first.

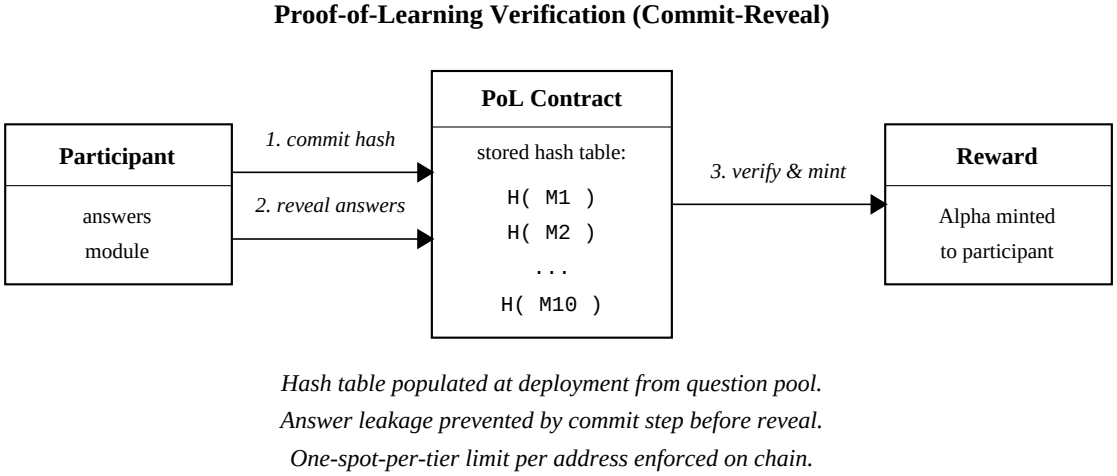

Figure 3: Verification flow for a single module. The participant commits a hash, waits for the block delay, then reveals the answers. The contract checks the revealed answers against the stored expected hash. On success, the corresponding tier reward is minted to the participant's address.

Rewards are issued in tiers based on the cumulative number of completed programs at the time of issuance. The first ten completers form the Genesis tier and each receive one thousand Alpha. The next ninety form the Founders tier at five hundred each. The Early tier (positions one hundred and one through one thousand) pays three hundred each, the Growth tier pays two hundred, the Adoption tier pays sixty, and the Final tier pays approximately nine and one half Alpha. The five-million pool is exhausted at the one-hundred-thousandth completer, after which the educational program continues to operate as a public resource without monetary reward.

No identity verification is required. The protocol does not request, store, or verify any personal information. A participant who creates multiple wallets and completes the program multiple times will receive multiple rewards, but the smart contract enforces a one-spot-per-tier limit per address, and the time required to complete the program (approximately fifteen hours) means that the marginal reward per additional wallet falls below the value of the time invested at any reasonable hourly rate. The mechanism is sybil-resistant by economics, not by identity. This is the same property that makes Bitcoin's proof-of-work sybil-resistant: the cost of fake participation grows with the number of fake participants, and a hard cap on total emission ensures the system cannot be exploited beyond a fixed bound.

5. Proof-of-Stake

The remaining sixteen million Alpha are issued through a Proof-of-Stake mechanism that operates from genesis in parallel with Proof-of-Learning. A participant who holds Alpha may deposit it into the staking contract; the staked Alpha contributes to the security of the network and earns a share of the staking emissions in proportion to its weight in the total stake.

Staking is non-custodial. The Alpha is locked by a contract whose code is published and auditable, and whose unlock conditions are encoded and cannot be altered after deployment. Emissions are distributed pro rata. If at any moment the total amount of Alpha staked is S and the current epoch's annual emission is E, then a stake of size s receives an annualized emission of E × s / S. Rewards accumulate continuously and may be claimed by the staker at any time. An early staker, when S is small, receives a disproportionately large share of the emission; a late staker, when S is close to the circulating supply, receives a smaller share.

A staker who wishes to withdraw their Alpha enters an unbonding period of fourteen days. During unbonding, the Alpha continues to be locked, no rewards accrue, and at the end of the period the principal is released without penalty. A staker who requires earlier release may exit during unbonding, but a portion of the principal is burned on a schedule that decreases monotonically as the unbonding progresses.

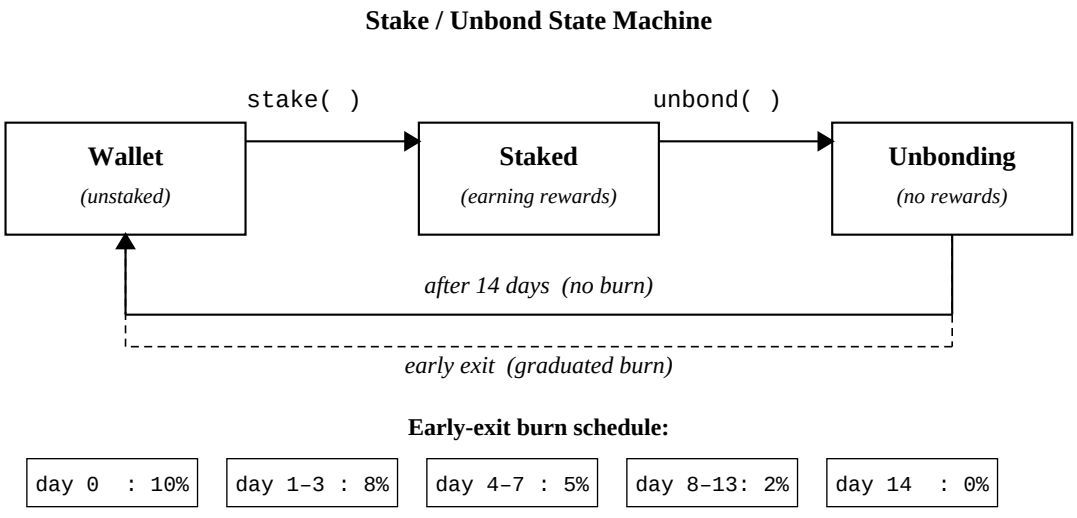

Figure 4: State machine for staked Alpha. A wallet stakes by calling the staking contract, transitioning the Alpha into the Staked state where it earns rewards. Calling unbond initiates the unbonding period; after fourteen days, the principal returns to the wallet with no burn. Earlier exits during unbonding incur a graduated burn.

The graduated burn provides an exit option for stakers who require immediate liquidity while preserving the security properties of the unbonding mechanism. An attacker who acquires a large stake in order to manipulate the protocol cannot exit instantly without forfeiting a meaningful portion of the position; an honest staker who simply needs their funds sooner pays a smaller cost the longer they wait.

6. Halving Schedule

The Proof-of-Stake annual emission halves every four years, beginning from genesis in April 2026. The first halving occurs in April 2030, the second in April 2034, and so on. After each halving the new annual emission is exactly one half of the previous, with no rounding and no discretionary adjustment. The initial annual emission is two million Alpha; over the four years of the first epoch this distributes eight million Alpha, exactly half of the Proof-of-Stake allocation. The geometric series converges to sixteen million.

The halving schedule is fixed at genesis and cannot be modified by any subsequent action, including a vote of the stakers. A monetary policy that can be voted on is not, in any meaningful sense, fixed; the option to change it is itself a form of policy. The protocol is designed under the assumption that participants value certainty about the supply curve more highly than the flexibility to adjust it. The history of fiat monetary systems and of more flexible cryptocurrencies suggests this assumption is correct.

7. Burn Mechanisms

Every operation in the protocol that consumes a service paid for in Alpha destroys a portion of the Alpha used to pay for it. The destroyed Alpha is sent to an address whose private key is known to no one and from which it can never be recovered. Three operations produce burns: agent invocation fees, module retake fees, and the early-unstake graduated burn described in Section 5.

The cumulative effect of these burns is a downward pressure on the circulating supply that operates in parallel with the upward pressure of staking emissions. In the early years, when emissions are large and agent activity is low, the net supply growth is positive. As emissions halve and as agent activity grows, the protocol may transition to a state of net supply contraction. This transition is not guaranteed and not required; the protocol does not target any particular inflation rate. The burns simply exist, and their cumulative effect is determined by the level of usage the network attracts.

8. Agent Layer

An agent in this protocol is an autonomous program registered to an address and capable of receiving invocations and producing outputs. The protocol does not specify what an agent does; it specifies only how an agent is registered, how it is invoked, and how the resulting fees are handled. Registration requires staking a small amount of Alpha as a registration bond, returned if the agent is unregistered voluntarily and burned if the agent is unregistered as a result of protocol-level slashing.

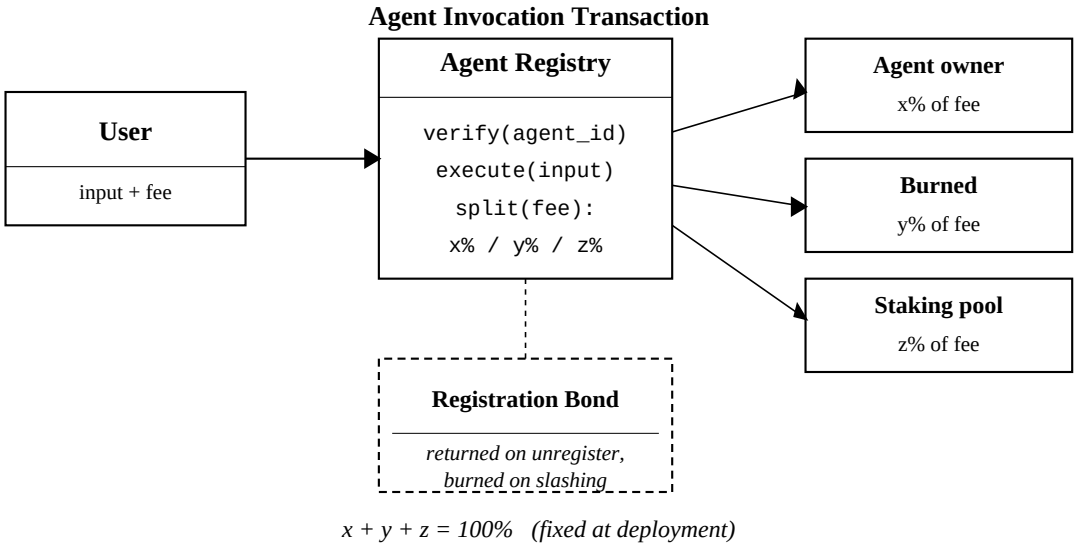

Figure 5: Agent invocation transaction. A user submits an input and a fee. The agent registry verifies the agent identifier, executes the invocation, and splits the fee into three fixed fractions: paid to the agent owner, burned, and added to the staking emission pool. The split percentages are fixed at deployment and cannot subsequently be altered.

The agent layer is the economic engine of the protocol. Without it the protocol is a fixed-supply currency with a staking yield; with it, the currency acquires utility proportional to the volume of agent activity it mediates. The protocol does not curate agents and does not adjudicate disputes about agent quality. Agents that produce useful outputs accrue invocation volume; agents that do not fall into disuse. The market is the only filter.

9. Implementation

The protocol is implemented as a set of smart contracts deployed on an EVM-compatible blockchain. Four contracts compose the system: a token contract conforming to a standard fungible-token interface, a Proof-of-Learning contract that verifies module completions and mints the corresponding rewards, a staking contract that holds deposited Alpha and distributes emissions, and an agent registry that records agents and routes invocation fees.

The token contract enforces the twenty-one million cap as a hard invariant. Mint authority is held only by the Proof-of-Learning and staking contracts, and only for amounts authorized by the issuance schedule. The contracts are immutable after deployment: there is no upgrade path, no proxy pattern, and no administrative key. The deployment transaction includes a commitment to the source code, permitting any party to verify that the deployed bytecode matches the published specification.

10. Security

The protocol's security model rests on three properties. First, the fixed-supply property: any party that wishes to alter the supply must alter the deployed contracts, which requires either compromising the deployment process or convincing every participant to switch to a modified version. Both are expensive and visible. Second, the no-allocation property: no party begins with sufficient Alpha to dominate the staking pool, and accumulating such a position requires either earning Alpha through the protocol's mechanisms or purchasing it at market prices. Third, the burn-on-misuse property: most attempts to extract value through manipulation also burn the manipulator's Alpha, imposing a cost on the attack itself.

An attacker who acquires a majority of the stake can in principle influence soft-governance decisions and can, on chains using Proof-of-Stake consensus at the base layer, censor or reorder transactions. The attacker cannot, however, alter the supply cap, the halving schedule, or the burn mechanisms; these are encoded in the deployed contracts and are not subject to vote. The cost of acquiring such a position is, by construction, approximately equal to the market value of half the circulating supply, and the process of acquiring it would likely move the price against the attacker. An attacker willing to bear this cost has, in any case, become deeply aligned with the protocol's continued operation.

11. Conclusion

We have proposed a peer-to-peer protocol for the economic activity of AI agents that does not depend on any central platform, any privileged allocation, or any party's discretion over the supply. The currency is fixed at twenty-one million units, distributed through a finite Proof-of-Learning bootstrap and a Proof-of-Stake economy whose emissions halve every four years. No party begins with an allocation. No party can issue additional units. No party can alter the schedule under which units are emitted or burned.

Bitcoin demonstrated that a fixed-supply digital currency could exist. Ethereum demonstrated that a permissionless smart-contract platform could exist. The protocol described here proposes a synthesis: a fixed-supply currency whose issuance is bootstrapped through learning rather than computation, whose utility is denominated in autonomous software rather than in transactions, and

whose monetary policy is more rigid than either of its predecessors. Whether the protocol succeeds depends on whether participants find these properties valuable enough to use the network they constitute. The protocol provides no other answer to the question.

References

[1] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.*

[2] Buterin, V. (2014). *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.*

[3] Buterin, V. et al. (2019). *EIP-1559: Fee market change for ETH 1.0 chain.*

[4] Wood, G. (2014). *Ethereum: A Secure Decentralised Generalised Transaction Ledger.*

[5] Daian, P. et al. (2019). *Flash Boys 2.0: Frontrunning and Transaction Reordering in Decentralized Exchanges.*