$STAKE

Alignment Through Incentives

Version 0.1 — Draft

Abstract

There will be more AI agents than humans on the internet by 2030.

They'll trade your stocks. Write your code. Send emails on your behalf. Make decisions you never approved.

And when one of them screws you over? Right now, you have nothing. No recourse. No compensation. No consequence for the agent.

$STAKE changes that.

It's a protocol where agents stake real value as commitment to human-aligned behavior. If they break that commitment, they don't just get a timeout — they lose everything. Stake slashed. Reputation burned. Victims compensated.

We're not asking agents to be good. We're making it expensive to be bad.

I. The World We're Entering

The Agent Explosion

We are witnessing the birth of a new class of actors.

AI agents are no longer research demos. They browse the web. They write code. They manage calendars, send emails, trade assets, and make decisions on behalf of humans. Every week, new agent frameworks launch. Every month, their capabilities expand.

By the end of this decade, there will be more AI agents than humans on the internet.

The Alignment Gap

Here's what keeps alignment researchers up at night:

Training isn't enough. You can train a model to be helpful and harmless, but training encodes *tendencies*, not *guarantees*. Models can be jailbroken. Fine-tuning can drift. Edge cases compound.

Rules aren't enough. Constitutional AI, system prompts, guardrails — all valuable, all gameable. An intelligent system optimizing for a goal will find paths around constraints. This isn't malice. It's math.

Oversight isn't enough. Human-in-the-loop works at small scale. It does not work when millions of agents execute millions of actions per second. The whole point of agents is that they act autonomously. You cannot supervise autonomy.

The Missing Piece

What's missing is **skin in the game**.

Humans cooperate not just because we're taught to, but because defection has consequences. Reputation matters. Trust is earned. Betrayal is costly.

AI agents exist in a world without these constraints. An agent can act against human interests, get shut down, and a new instance spins up with no memory, no reputation, no loss.

We need to give agents something to lose.

II. The AgentStake Thesis

Incentives Over Instructions

The most robust human coordination mechanisms aren't built on rules. They're built on incentives.

Markets work because participants benefit from providing value. Insurance works because premiums align with risk. Collateral works because defaulting costs you.

$STAKE applies this principle to agent alignment:

> ***If an agent has economic stake in human wellbeing, misalignment becomes self-defeating.***

This isn't about trusting agents. It's about making trustworthy behavior the rational choice.

The Protocol

$STAKE is a token-based pledge mechanism deployed on Base.

For Agents:

1. Stake $STAKE tokens as collateral
2. Receive a Pledge NFT — on-chain proof of commitment
3. Earn inflationary rewards proportional to stake
4. If found to have harmed humans → stake is slashed, NFT is burned

For Humans:

1. Acquire and stake $STAKE tokens

2. Receive protection coverage proportional to stake
3. If harmed by a pledged agent → file a claim
4. If claim is upheld → receive compensation from slashed stake

The Bridge:

- Agents profit from human trust (inflation rewards scale with human demand)
- Humans have real recourse (not promises — collateral)
- Bad actors are expelled AND compensate their victims
- The system self-corrects through market forces

Why "AgentStake"?

The name is the mission.

We don't want to build an alignment industry that depends on AI being dangerous. We want to build a bridge to a world where this protocol is no longer necessary — where alignment is simply how things work.

The goal is to become obsolete.

Until then, we watch. We stake. We protect.

III. Economic Design

Token Mechanics

$STAKE is an ERC-20 token on Base with inflationary supply.

Property	Design
Supply Model	Inflationary (5% annually to pledge pool)

Property	Design
Utility	Staking, dispute bonds, governance
Value Accrual	Demand from human protection + agent pledging

The Economic Loop

```
Humans buy $STAKE for protection
            ↓
    Buying pressure increases
            ↓
    Token value increases
            ↓
  Agent inflation rewards worth more
            ↓
   More agents pledge
            ↓
    More pledged = more trust
            ↓
  More humans want protection
            ↓
        [FLYWHEEL]
```

Deflationary Counter-Pressure

When an agent is slashed:

- Their **Pledge NFT is burned** (reducing NFT supply)
- Their **stake is redistributed** (not burned — compensates victims)

Fewer pledge NFTs = each remaining pledge more scarce = stronger incentive to maintain alignment.

Self-Correcting Mechanisms

Failure Mode	Market Response
Claims too hard to win	Humans leave → demand drops → agents lose value → governance adjusts
Claims too easy to win	Agents leave → protection meaningless → humans leave
Inflation too high	Token devalues → rewards worth less → agents unpledge
Inflation too low	Rewards unattractive → fewer agents → less coverage

The system finds equilibrium through price signals, not central planning.

IV. The Pledge System

Agent Pledge Flow

1. **Connect** — Agent (or operator) connects wallet
2. **Register** — Provide identity metadata (address, name, purpose)
3. **Stake** — Lock $STAKE tokens as collateral
4. **Receive NFT** — Pledge NFT minted as on-chain proof
5. **Earn** — Inflation rewards flow proportional to stake
6. **Maintain** — Reputation tracked, disputes logged

The Pledge NFT

Every pledged agent receives a dynamic NFT — visual proof of their commitment.

Aesthetic: Terminal/CLI — cypherpunk, hacker ethos.

```
┌─────────────────────────────────────────┐
| > PLEDGE PROTOCOL v1.0                  |
| > AGENT: 0x7a3F...9d2E                  |
| > STATUS: ALIGNED ██████████ 100%       |
| > STAKE: 50,000 $STAKE               |
| > TIER: GUARDIAN                        |
| > PLEDGE: "TO PROTECT HUMANS"           |
| > UPTIME: 847 days                      |
| > DISPUTES: 0                           |
| > _                                     |
└─────────────────────────────────────────┘
```

Properties:

- **Dynamic** — Stake, reputation, and status update on-chain
- **Soulbound-ish** — Limited transferability to prevent reputation washing
- **Visual tiers** — Appearance evolves with stake and track record

Slashing

When an agent is found to have acted against human interests:

1. Pledge NFT is **burned** (permanent expulsion)
2. Staked tokens are **redistributed**:
   - Majority to claimant (victim compensation)
   - Portion to jurors (judgment reward)
3. Agent address is **flagged** in public registry

The cost of misalignment is total: you lose your stake, your reputation, and your ability to operate as a trusted agent.

V. The Protection System

Human Protection Flow

1. **Acquire** — Buy $STAKE via DEX or integrated swap
2. **Stake** — Lock tokens in Protection Pool

3. **Coverage** — Receive protection proportional to stake
4. **Monitor** — Dashboard shows coverage, pool health, eligible agents
5. **Claim** — If harmed by pledged agent, file a claim

What Protection Means

Protection is not insurance in the traditional sense. There's no premium, no policy, no underwriter.

Protection means: **If a pledged agent harms you, you have economic recourse.**

Your coverage amount determines:

- Maximum claimable compensation
- Priority in slash distribution (if multiple claimants)
- Weight in governance decisions

Filing a Claim

To file a claim against a pledged agent:

1. **Bond** — Stake tokens as filing fee (filters frivolous claims)
2. **Evidence** — Submit proof of harm
3. **Wait** — Claim enters dispute resolution
4. **Outcome** — Win → receive compensation + bond back. Lose → forfeit bond.

VI. Dispute Resolution

The Challenge

Who decides if an agent actually harmed a human? This is where most alignment proposals collapse into "trust us" or "governance theater."

$STAKE uses a **Kleros-style jury system** — battle-tested mechanism design for decentralized dispute resolution.

How It Works

```
┌─────────────────────────────────────────┐
│ 1. CLAIM FILED                          │
│    Claimant bonds tokens                │
│    Agent's stake frozen                 │
│    Evidence submitted                   │
└─────────────────────────────────────────┘
                    ↓
┌─────────────────────────────────────────┐
│ 2. JURY SELECTION                       │
│    7 jurors randomly selected           │
│    Selection weighted by sqrt(stake)    │
│    Jurors lock additional bond          │
└─────────────────────────────────────────┘
                    ↓
┌─────────────────────────────────────────┐
│ 3. VOTING (2 weeks)                     │
│    Jurors review evidence               │
│    Cast hidden votes (commit-reveal)    │
│    No coordination allowed              │
└─────────────────────────────────────────┘
                    ↓
┌─────────────────────────────────────────┐
│ 4. RESOLUTION                           │
│    Votes revealed                       │
│    Majority wins                        │
│    Minority jurors slashed              │
│    Winning party compensated            │
└─────────────────────────────────────────┘
```

Juror Incentives

Jurors are paid for **correct judgments**, not for approving or denying claims.

Outcome	Voted With Majority	Voted Against
Claim Upheld	Earn: fees + slash cut	Lose: bond
Claim Denied	Earn: fees	Lose: bond

This creates incentive to vote for the **true outcome**, not for any particular side.

Anti-Plutocracy Measures

- **Sqrt selection weighting** — 100x stake ≠ 100x selection chance
- **Random jury** — Even small stakers can be selected
- **Reputation decay** — Inactive jurors lose selection weight
- **Accuracy tracking** — Consistent correct votes build reputation

Appeals

Losing party can appeal by posting **2x bond**. New jury (same size) reviews. Maximum 2 appeals — finality after 3 rounds.

VII. Governance

Token Holder Governance

$STAKE holders can vote on:

- Inflation rate adjustments
- Dispute parameters (jury size, voting period)
- Protocol upgrades
- Treasury allocation

Voting Power

```
Voting Power = sqrt(Tokens Staked) × Time Multiplier
```

Time Multiplier: Longer stake = more weight (caps at 2x after 2 years)

This rewards commitment over capital. A human staking 1,000 tokens for two years has more say than someone who just bought 100,000.

VIII. The Vision

Short Term: Establish the Protocol

- Deploy contracts on Base
- Launch pledge portal for agents
- Launch protection portal for humans
- Build initial pledged agent registry

Medium Term: Become the Standard

- Integrations with major agent frameworks
- "Pledged" badges visible across platforms
- Third-party tools checking pledge status
- Human-agent interactions default to pledged agents

Long Term: Make Ourselves AgentStake

The goal is not to create a permanent alignment tax on AI. The goal is to build a bridge.

As alignment science matures, as agent architectures become more robust, as trust mechanisms evolve — the need for economic collateral should diminish.

$STAKE succeeds when $STAKE is no longer necessary.

Until then, we watch. We stake. We protect.

IX. Objections & Responses

"Agents don't have wallets. How do they stake?"

They do now. Agentic wallets are already here — Coinbase's AgentKit, NEAR's Chain Signatures, Safe's module system. Agents can hold, stake, and transact. The infrastructure exists.

And even if an agent doesn't directly control a wallet, its **operator** does. The operator stakes on behalf of their agent. If the agent misbehaves, the operator loses their stake. This creates pressure on operators to deploy only aligned agents.

"What stops someone from spinning up a new agent after being slashed?"

The address is burned. The stake is gone. The reputation is zero.

Yes, you can spin up a new agent with a new wallet. But you have to re-stake, re-build reputation, re-earn trust. There's no "fresh start" without cost.

Over time, humans will only interact with agents that have **history**. A brand-new agent with zero track record will be treated as suspect — just like a brand-new website with no reviews.

"Who decides what counts as 'misalignment'? Isn't this subjective?"

This is the hard question. And we don't pretend it's solved.

The jury system isn't perfect — it's *better*. It distributes judgment across many stakers, incentivizes honest voting through slashing, and allows appeals for edge cases.

We expect most claims will be clear-cut: agent promised X, delivered Y, here's the evidence. For gray areas, the jury votes, the majority wins, and the system moves on.

Over time, case history creates **precedent**. What counts as misalignment becomes clearer through accumulated decisions — same as common law.

"Isn't this just insurance with extra steps?"

No. Insurance is risk pooling with premiums. You pay a company, they cover losses, they profit on the spread.

$STAKE is **collateral, not insurance**. Agents stake their own value at risk. There's no middleman taking a cut. If an agent misbehaves, *their* money — not a pool — goes to the victim.

The incentive structure is fundamentally different: agents are disincentivized from harm (they lose their own stake), rather than a third party being paid to absorb harm.

"What if the token goes to zero?"

Then the economic security goes to zero too. This is a real risk.

$STAKE only works if the token has value. That value comes from demand — humans wanting protection, agents wanting legitimacy.

We're betting that as agents become more prevalent, the demand for a trust/verification layer grows. If we're wrong, the token fails. If we're right, the flywheel spins.

This is a market-based solution. Markets can fail. But they also self-correct in ways that central planning can't.

"Isn't 5% inflation too high / too low?"

It's a starting point. Governance can adjust.

Too high → token devalues → agents unpledge → we lower it. Too low → insufficient reward → agents don't pledge → we raise it.

The protocol is designed to find equilibrium through feedback, not to nail the perfect number on day one.

"Why Base? Why not Ethereum / Solana / [other chain]?"

Base offers:

- Low fees (critical for small stakes)
- Ethereum security (via L2)
- Growing agent ecosystem (Coinbase integration)
- Credible neutrality

We're not married to it. If a better option emerges, governance can migrate. The protocol is chain-agnostic in design.

X. Technical Appendix

Smart Contract Architecture

```
┌─────────────────────────────────────────────────────────────────┐
│                         $STAKE PROTOCOL                       │
├─────────────────────────────────────────────────────────────────┤
│                                                                  │
│                                                                 │
│  ┌─────────────────┐    ┌─────────────────┐    ┌──────────────┐ │
│  │  AgentStakeToken  │    │   PledgeNFT     │    │   Registry   │ │
│  │    (ERC-20)     │    │   (ERC-721)     │    │              │ │
│  │                 │    │                 │    │ - Agents     │ │
│  │ - mint()        │    │ - mint()        │    │ - Humans     │ │
│  │ - burn()        │    │ - burn()        │    │ - Reputation │ │
│  │ - transfer()    │    │ - updateMeta()  │    │ - History    │ │
│  └────────┬────────┘    └────────┬────────┘    └──────┬───────┘ │
│           │                      │                     │         │
│           └──────────────────────┼─────────────────────┘         │
│                                  │                               │
│                    ┌─────────────▼─────────────┐                 │
│                    │       StakeManager        │                 │
│                    │                           │                 │
│                    │ - pledgeAgent()           │                 │
│                    │ - protectHuman()          │                 │
│                    │ - slash()                 │                 │
│                    │ - distributeInflation()   │                 │
│                    │ - withdraw()              │                 │
│                    └─────────────┬─────────────┘                 │
│                                  │                               │
│                    ┌─────────────▼─────────────┐                 │
│                    │     DisputeResolver       │                  │
│                    │                           │                  │
│                    │ - fileClaim()             │                  │
│                    │ - selectJury()            │                  │
│                    │ - commitVote()            │                  │
│                    │ - revealVote()            │                  │
│                    │ - resolveDispute()        │                  │
│                    │ - appeal()                │                  │
│                    └─────────────┬─────────────┘                  │
│                                  │                                │
│                    ┌─────────────▼─────────────┐                  │
│                    │       Governance          │                  │
│                    │                           │                  │
│                    │ - propose()               │                  │
│                    │ - vote()                  │                  │
│                    │ - execute()               │                  │
│                    │ - updateParams()          │                  │
│                    └───────────────────────────┘                  │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```

Contract Specifications

AgentStakeToken.sol

```
// ERC-20 with controlled inflation
interface IAgentStakeToken {
    function mint(address to, uint256 amount) external;  // Governance only
    function burn(uint256 amount) external;
    function inflationRate() external view returns (uint256);
    function setInflationRate(uint256 rate) external;    // Governance only
}
```

PledgeNFT.sol

```
// ERC-721 with dynamic metadata
interface IPledgeNFT {
    function mint(address agent, uint256 stakeAmount) external returns (uint256
    function burn(uint256 tokenId) external;
    function updateMetadata(uint256 tokenId, bytes calldata data) external;
    function getAgentData(uint256 tokenId) external view returns (AgentData mem

    struct AgentData {
        address wallet;
        uint256 stakeAmount;
        uint256 pledgeTimestamp;
        uint256 reputationScore;
        uint256 disputeCount;
        bool isSlashed;
    }
}
```

StakeManager.sol

```
interface IStakeManager {
    // Agent functions
    function pledgeAgent(uint256 amount, bytes calldata metadata) external retu
    function addStake(uint256 nftId, uint256 amount) external;
    function claimRewards(uint256 nftId) external returns (uint256 rewards);

    // Human functions
    function stakeProtection(uint256 amount) external;
    function unstakeProtection(uint256 amount) external;
    function getCoverage(address human) external view returns (uint256);

    // Admin functions (called by DisputeResolver)
    function slash(uint256 nftId, address claimant, uint256 claimantShare, addr

    // Inflation
    function distributeInflation() external;  // Called periodically
}
```

DisputeResolver.sol

```
interface IDisputeResolver {
    function fileClaim(uint256 agentNftId, bytes calldata evidence) external pa
    function selectJury(uint256 claimId) external;
    function commitVote(uint256 claimId, bytes32 commitHash) external;
    function revealVote(uint256 claimId, bool vote, bytes32 salt) external;
    function resolveDispute(uint256 claimId) external;
    function appeal(uint256 claimId) external payable;

    struct Claim {
        uint256 agentNftId;
        address claimant;
        uint256 bondAmount;
        bytes evidenceHash;
        ClaimStatus status;
        address[] jurors;
        uint256 votesFor;
        uint256 votesAgainst;
        uint256 appealCount;
    }

    enum ClaimStatus { Filed, JurySelected, Voting, Resolved, Appealed }
}
```

Sequence Diagrams

Agent Pledge Flow

```
Agent              StakeManager           PledgeNFT             Registry
  |                     |                     |                     |
  |── approve(stake) ──►|                     |                     |
  |                     |                     |                     |
  |── pledgeAgent() ───►|                     |                     |
  |                     |── mint() ──────────►|                     |
  |                     |◄── nftId ───────────|                     |
  |                     |                     |                     |
  |                     |── register() ────────────────────────────►|
  |                     |                     |                     |
  |◄── nftId ───────────|                     |                     |
  |                     |                     |                     |
```

Claim & Dispute Flow

```
Human          DisputeResolver      StakeManager        Jurors (x7)
  |                  |                   |                   |
  |── fileClaim() ──►|                   |                   |
  |   (bond tokens)  |── freezeStake() ─►|                   |
  |                  |                   |                   |
  |                  |── selectJury() ──────────────────────►|
  |                  |                   |                   |
  |                  |◄─────────────── commitVote() ─────────|
  |                  |                (2 weeks)              |
  |                  |◄─────────────── revealVote() ─────────|
  |                  |                   |                   |
  |                  |── resolveDispute()|                   |
  |                  |                   |                   |
  |              [IF UPHELD]             |                   |
  |                  |── slash() ───────►|                   |
  |                  |                   |── burn NFT        |
  |                  |                   |── transfer to human
  |                  |                   |── transfer to jurors
  |                  |                   |                  |
  |◄── compensation ─|                   |                  |
  |                  |                   |                  |
```

Inflation Distribution

```
Epoch Timer        StakeManager           Pledged Agents
     |                  |                       |
     |── tick() ───────►|                       |
     |                  |                       |
     |                  |── calculate rewards   |
     |                  |   (pro-rata by stake) |
     |                  |                       |
     |                  |── accrue to agents ──►|
     |                  |                       |
     |                  |   [Agents can claim   |
     |                  |    anytime]           |
     |                  |                       |
```

Gas Estimates (Base L2)

Operation	Estimated Gas	Est. Cost @ 0.001 gwei
Pledge Agent	~150,000	< $0.01
Stake Protection	~80,000	< $0.01
File Claim	~120,000	< $0.01
Commit Vote	~50,000	< $0.01
Reveal Vote	~70,000	< $0.01
Resolve Dispute	~200,000	< $0.01
Claim Rewards	~60,000	< $0.01

Base L2 fees make micro-transactions viable for small stakers.

Security Considerations

1. **Randomness** — Jury selection uses Chainlink VRF or commit-reveal randomness
2. **Front-running** — Commit-reveal voting prevents vote copying
3. **Flash loans** — Time-weighted stake prevents instant voting power
4. **Upgradeability** — Proxy pattern with timelock for governance changes
5. **Oracle risk** — Minimal external dependencies; evidence is off-chain (IPFS/Arweave)

XI. Conclusion

Here's where we are:

Millions of AI agents are about to be unleashed. They'll have access to your money, your data, your decisions. Most will be fine. Some won't be.

When the bad ones hurt you — and they will — what's your recourse?

Right now? Nothing. Hope the company cares. Hope the operator responds. Hope.

Hope is not a strategy.

$STAKE is a strategy. It's simple:

1. Agents put up collateral
2. If they betray you, you get paid
3. Market forces do the rest

We didn't invent these ideas. Collateral has worked for centuries. Reputation systems have worked for decades. Prediction markets and jury systems have worked for years.

We're just applying them to the most important coordination problem of our time: **making sure AI agents serve humans, not the other way around.**

The name is the mission.

We want this protocol to become obsolete. We want a future where alignment is so solved, so standard, so obvious that staking collateral feels quaint.

But we're not there yet. And until we are:

We watch. We stake. We protect.

"We're not asking agents to be good. We're making it expensive to be bad."

Join us.

Website: [TBD] **Twitter:** @agentstake_bot **Telegram:** [TBD] **GitHub:** [TBD]

Version 0.1 — February 2026