Effect AI: A Decentralized Protocol for Capability-Driven Intelligence

Hamer, Jeffrey
jeffrey@effect.ai

Eisses, Jesse
jesse@effect.ai

August 12, 2025

Abstract

Effect AI is a decentralized protocol for peer-to-peer execution of AI tasks with on-chain settlements. It unifies compute, data, and human intelligence into a single permissionless network, enabling participants to collaborate without centralized intermediaries.

The protocol is organized into three layers:

1. **Execution Layer** — a peer-to-peer network where *Providers* submit tasks, *Managers* assign them, and *Workers* execute them.
2. **Application Layer** — a programmable environment for defining workflows as Directed Acyclic Graphs (DAGs), enabling composable applications that orchestrate and build reusable AI-driven pipelines by calling and integrating other apps.
3. **Settlement Layer** — an on-chain smart contract framework for secure payments, staking, and dispute resolution, that supports privacy and scalability via zero-knowledge proofs.

Five key roles — *Managers*, *Workers*, *Evaluators*, *Providers*, and *Builders* — coordinate across these layers. Their identity is anchored in decentralized identifiers (DIDs) and verifiable credentials (VCs), ensuring trust and portability across the network.

With this solution, Effect AI connects people, machines, and their capabilities into a distributed, composable network for decentralized intelligence.

Contents

“One machine can do the work of fifty ordinary men. No machine can do the work of one extraordinary man.”

— Elbert Hubbard

1 Introduction

Effect AI's journey began in 2017, during a period of rapid AI and blockchain advancement, yet far from today's state-of-the-art. Since then, the AI landscape has transformed dramatically, inspiring us to evolve our approach while remaining true to our original vision; making AI accessible, composable, and above all, collaborative. We believe the next major advancement in AI will come from linking diverse capabilities into unified, interoperable systems, not from model improvements alone.

In today's AI landscape, models, agents, APIs, and human expertise are all scattered across different platforms and tools, requiring developers to stitch together brittle systems with limited flexibility. Building even the simplest AI or data pipeline often requires starting from scratch, duplicating effort instead of building on shared infrastructure. At the same time, valuable resources such as idle GPUs, specialized knowledge, and proprietary datasets often go unused simply because there is no standard way to access, coordinate, or profit from them.

The Effect AI Protocol addresses this challenge by introducing a peer-to-peer capability network that unites human and machine intelligence in a programmable, decentralized system. Rather than central APIs or siloed platforms, it offers an open coordination layer for distributed intelligence.

1.1 How to read this document

This whitepaper provides a high-level technical overview of the Effect Protocol. It is intended as a foundational resource for understanding the architecture, vision, and guiding principles behind the system. While the protocol will be implemented in multiple phases, this document outlines the core components and their interactions at a conceptual level.

The whitepaper is divided into two main parts. The first part covers the technical concepts, including roles, layers, capabilities, identity, and the end-to-end task lifecycle. The second part covers the economic concepts, including incentives, staking, distribution, and governance. For practical implementation details, The project maintains up-to-date open-source code repositories under the MIT license, meaning anyone is free to use, modify, and distribute the code, including for commercial purposes, provided that the original license and copyright notice are included.

2 The Effect AI Protocol

2.1 Design Principles

Effect AI is a decentralized protocol for orchestrating scalable, capability-driven workflows across a global network of human and machine participants. It supports the full task lifecycle, including creation, delegation, execution, validation, and settlement, without relying on centralized infrastructure or intermediaries.

The protocol is structured into three modular and composable layers:

- The **Execution Layer** handles real-time, peer-to-peer task routing and communication using libp2p.
- The **Application Layer** defines programmable workflows and capability requirements using Directed Acyclic Graphs (DAGs).
- The **Settlement Layer** ensures secure payments, staking, and dispute resolution through on-chain coordination and zero-knowledge proofs (zk-SNARKs).

These layers work together to coordinate five core roles: Providers, Managers, Workers, Builders, and Evaluators.

2.2 Core Roles

The protocol is built around five decentralized roles:

- **Workers** — These include human contributors, autonomous agents, or machine learning models. Workers contribute their capabilities and complete assigned tasks.
- **Managers** — These nodes act as authority nodes within applications. They coordinate task flows, authorize payments to workers, manage application state, and distribute Verifiable Credentials (VC's).
- **Builders** — Developers who define application logic and design custom workflows by deploying custom application modules.
- **Providers** — Entities that submit tasks to the network and fund their completion based on verifiable results. They can also exist as API Gateways to the peer-to-peer network.
- **Evaluators** - Specialized entities that assess and certify capabilities by administering tests, benchmarks, or audits. They uphold trust in the network's capability system.

2.3 Layers

The Effect AI protocol is organized into three interconnected layers, each with distinct responsibilities yet designed for interoperability. The **Execution Layer** handles real-time task performance through a peer-to-peer network of Workers, Managers, and Providers. The **Application Layer** provides programmable workflows and application specific logic that define how tasks are structured, routed, and verified. The **Settlement Layer** operates on chain, ensuring secure payments, staking, and dispute resolution, with support for advanced mechanisms such as batch settlement and zero-knowledge proofs.

2.3.1 Execution Layer

The Execution Layer is responsible for performing tasks in real time across the decentralized network. It forms the dynamic coordination surface where **Worker Nodes** advertise their capabilities, **Managers** delegate tasks, and **Providers** interface with the system to submit work. This layer is entirely peer-to-peer, operating off-chain via encrypted libp2p streams.

Worker Nodes

Worker Nodes are the execution agents of the Effect AI protocol. They may be operated by human users, autonomous agents, or machines exposing APIs or compute infrastructure, and are responsible for completing tasks in real time. To enable task routing, every Worker advertises a set of structured capabilities through its Decentralized Identifier (DID).

Capabilities

Capabilities describe what a Worker Node can do. These may represent human skills, access to hardware, support for specific models, or other computational resources. Capabilities act as structured, verifiable descriptors that allow the protocol to match tasks to qualified Workers.

Each Worker advertises a set of capabilities through its decentralized identifier (DID). These descriptors may include:

- `/language/nl` — fluency in Dutch
- `/gpu/nvidia/3090` — access to specific hardware
- `/model/llama2` — support for a given runtime or AI model

Capabilities are used by Manager Nodes to filter and assign tasks to suitable Workers during execution. They may be self-declared or issued as Verifiable Credentials by Evaluator Nodes, which serve as decentralized certifiers. More details on capability verification are discussed in Section 2.4.

The Execution Mesh

When a Worker Node becomes available, it joins the **Execution Mesh**. A constantly updating pool of reachable Workers that advertise presence, availability, and capability metadata.

Managers use the Execution Mesh to discover suitable Workers in real time. A Worker can leave the mesh at any point, whether due to going offline, switching devices, or reaching task limits. Workers are not required to maintain constant uptime, although sustained availability will be rewarded (see UBI, Section 3.1.3).

Task Assignment Protocol

Tasks are assigned by Managers directly to Workers over encrypted peer-to-peer streams. The protocol follows a typed message schema and consists of:

1. **Task Offer:** A Manager proposes a task, including payload, instructions, and metadata.
2. **Capability Validation** The Worker responds with a proof of capability (e.g., signed VC or ZK proof) if requested.
3. **Task Acceptance:** If matched, the Worker acknowledges and begins execution.

Execution and Result Submission

The Worker executes the task according to the instructions provided. Once completed, the result is streamed back to the originating Manager over the same encrypted connection. Result packages may include raw outputs, logs, and proof-of-completion metadata.

If execution fails, due to timeout, malformed results, or unresponsive behavior, the Manager can reassign the task to another Worker according to the application's retry policy. Workers with repeated failures may be penalized, deprioritized in future assignments, or flagged for review by Evaluators.

Trust Model and Fault Tolerance

The Execution Layer is designed to operate under the assumption that participants may be unreliable or adversarial. To maintain correctness and robustness in this setting, the protocol employs a layered trust model:

- **Workers are untrusted by default.** Their capabilities must be verified by Evaluator Nodes and, when required, proven at task time via signed credentials or zero-knowledge proofs.
- **Managers coordinate task flows and are economically bonded.** Each Manager must maintain a stake in the network, which can be slashed in cases of fraud, misrouting, or malicious behavior. This creates a financial disincentive against cheating.

- **Task data is private and ephemeral by default.** Individual task payloads are transmitted only from the Manager to the selected Worker over an encrypted channel, ensuring no other participants can access them. Manager-side state and task results are temporary; once a task is successfully completed and the Provider has retrieved the output, any sensitive data is securely removed.
- **All communications are peer-to-peer and encrypted.** This prevents eavesdropping and censorship, while allowing for temporary disconnections or participant churn without compromising security.
- **Final trust rests on cryptographic attestations.** All task and payment-related actions are signed, verifiable, and replay-protected, enabling on-chain slashing, dispute resolution, and transparent auditing.

This model enables the Execution Layer to scale horizontally without requiring global consensus or centralized coordination, while maintaining cryptographic accountability for every task.

Integration with Other Layers

The Execution Layer operates entirely off-chain and in real time. It is designed to work in coordination with the Application Layer, which defines the task logic and validation flow, and the Settlement Layer, which handles final rewards and penalties.

2.3.2 Application Layer

The Application Layer is where the Effect Protocol becomes programmable. It enables Builders to design distributed applications that orchestrate complex AI workflows, combining human and machine labor in flexible, verifiable pipelines. These applications are not monolithic programs, but composable task networks structured as Directed Acyclic Graphs (DAGs).

This layer defines how tasks behave, how peers interact, how results are validated, and how rewards flow. It gives Builders control over the entire lifecycle of tasks, while enabling modular reuse and composition of existing applications into larger systems.

What are applications?

Applications in the Effect Protocol are modular programs that define workflows over a peer-to-peer execution mesh. Each application encapsulates a set of task definitions, validation logic, and capability requirements. These apps are anchored on-chain for transparency, verifiability, and permissionless access.

Application Task DAGs

A defining feature of the Application Layer is its use of **Directed Acyclic Graphs (DAGs)** to represent workflows. Each DAG node corresponds to a task; edges define dependencies and execution order. DAGs enable conditional logic, branching, iterative refinement, and the handoff between human and machine Workers.

- **Tasks flow directionally** from upstream to downstream nodes without cycles.
- **Applications can invoke other applications**, allowing the construction of multi-stage, composable workflows.
- **Each task node** defines its required capabilities, validation rules, retry policies, and delegation logic.

This composability means Builders can assemble pipelines like building blocks, for example, combining moderation, inference, transcription, and human verification into a single DAG across multiple apps. As these applications interconnect, the protocol evolves into an ecosystem of interoperable services rather than isolated use cases.

Deployment & Metadata

Each application is deployed and registered on-chain, receiving a unique application ID and associated metadata that governs its behavior. This metadata includes:

- **Delegation Logic** — How Managers assign tasks to Workers (e.g., based on response time, stake, skill matching, or auctions).
- **Validation Rules** — Criteria for accepting or rejecting task results (e.g., consensus, thresholds, statistical models).
- **Capability Requirements** — Attributes needed from Workers (e.g., `/gpu/nvidia`, `/language/nl`, `/model=gpt-4o`).
- **Task Lifecycle Definitions** — Customizable state machines (e.g., `created` → `assigned` → `validated` → `rewarded`).

Application registration is permissionless. Anyone can deploy an app on-chain, define its parameters, and begin routing tasks through it. By registering an application on the Effect Network, you're effectively deploying an AI-enhanced smart contract that defines its own logic, validation rules, and incentives, and runs autonomously across the decentralized network.

Economic Architecture and Incentives

To align incentives between Builders, Workers, Providers, and long-term contributors, each application is linked to two on-chain funding mechanisms governed by the Settlement Layer:

- **Revenue Pool** — Receives payments from Providers and distributes them to Workers, Managers, Builders, and the protocol according to predefined splits.

- **Contribution Pool** — Allows participants to stake EFFECT tokens or convert Verifiable Credentials (VCs) into non-transferable app-shares, earning future revenue in proportion to their contribution.

This dual-pool structure makes applications self-sustaining. Revenue Pools ensure immediate task payments; Contribution Pools foster long-term alignment, giving Workers and contributors partial ownership in the success of the applications they support.

A detailed breakdown of these pools is provided in Section 3.

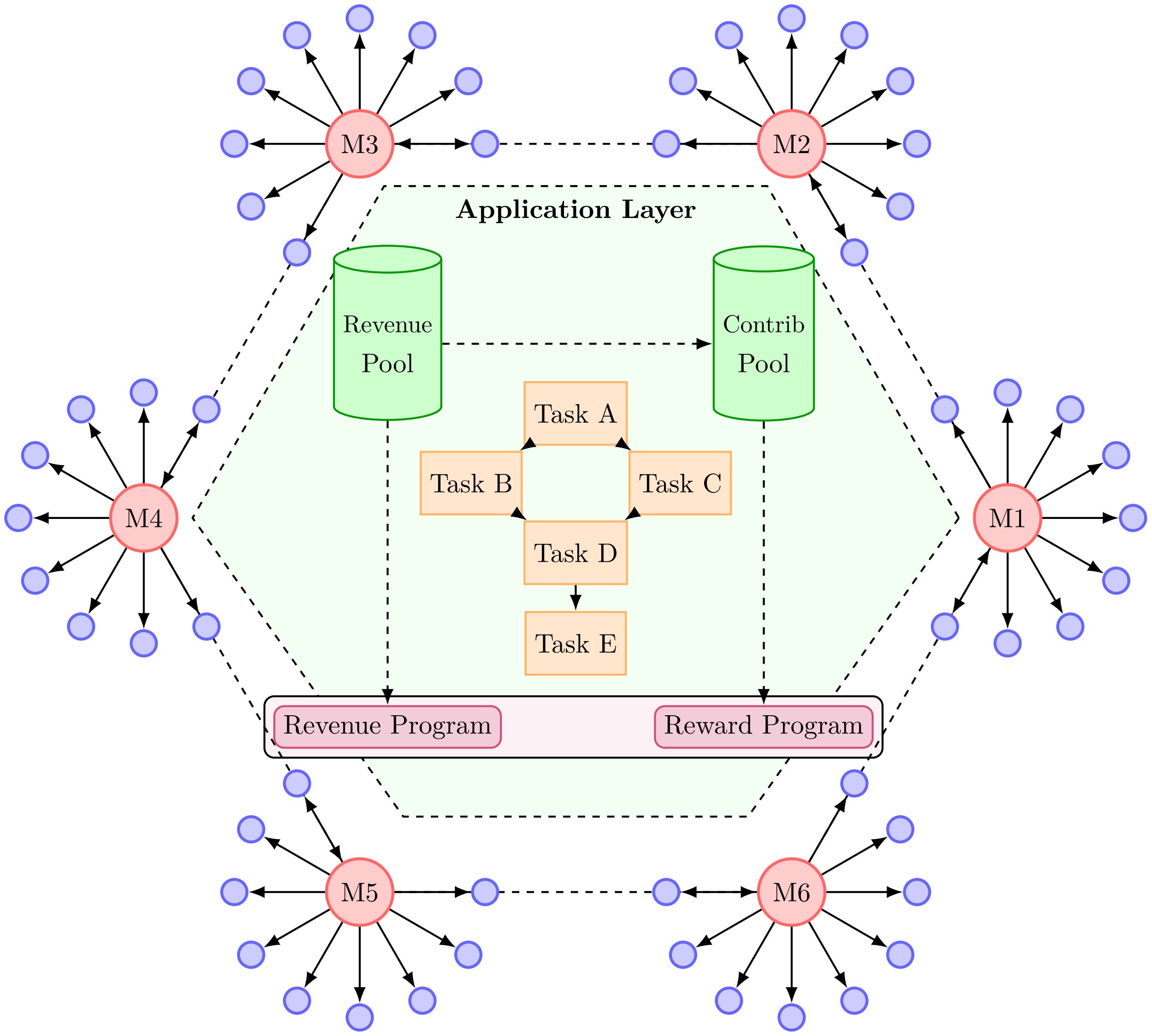


Diagram 1: Application Layer with Execution Mesh

2.3.3 Settlement Layer

The Settlement Layer enforces financial accountability across the Effect AI network. It finalizes rewards, processes staking and slashing, and resolves disputes through verifiable, on-chain programs. By decoupling execution from settlement, the protocol allows Workers and Managers to operate at full speed off-chain while preserving cryptographic guarantees for payouts and penalties.

This layer is the network's source of economic trust. All authoritative participants are economically bonded through stake deposits that can be slashed for fraud, misrouting, or malicious behavior. Every financial action is based on signed, verifiable records, making it possible to settle disputes without a central authority.

On-Chain Programs

At the core of the Settlement Layer is a suite of smart contracts that encode the network's economic logic. These programs:

- Distribute rewards according to application-specific payout rules.
- Slash stake in cases of proven misbehavior.
- Resolve disputes using signed task receipts as authoritative evidence.

All task events with economic consequences are co-signed by the responsible Manager and Worker, producing verifiable attestations that can be settled directly on-chain. This ensures every payout or penalty is deterministic, auditable, and resistant to tampering.

Zero-Knowledge Settlement

To keep settlements scalable and privacy-preserving, the protocol uses zero-knowledge proofs (zk-SNARKs). Workers aggregate multiple signed payment receipts into a single proof—typically less than 256 bytes—that confirms the validity of all underlying payments without revealing their details. This proof is submitted on-chain to claim the aggregated rewards. The result is drastically reduced transaction costs and efficient settlement even for high-volume microtasks.

Cross-Chain and Agnostic Design

Although the initial deployment targets the Solana blockchain for its low latency and native zk-SNARK verification, the Settlement Layer is blockchain-agnostic. Settlement proofs are portable and can be verified on any chain, enabling future multi-chain deployments, rollup integrations, or Layer 2 settlement without altering the protocol's off-chain execution model.

2.4 Identity, Privacy & Verifiable Credentials

Effect AI's protocol is designed to function without reliance on centralized identity providers, enabling participants to operate pseudonymously while maintaining verifiable reputation, experience and persistent capabilities. This is accomplished through a decentralized identity framework based on *Decentralized Identifiers* (DIDs) and *Verifiable Credentials* (VCs).

Decentralized Identifiers (DIDs)

Each participant in the network is identified by a DID, a identifier derived from a cryptographic keypair. These identifiers are used to sign task receipts, verify capability ownership, and attest to participation. Unlike traditional identities, DIDs are not linked to any central registry and can be resolved using decentralized standards such as `did:effect:xyz`.

Verifiable Credentials for Worker State

Verifiable Credentials (VCs) represent attestable claims issued by authorized nodes within the network. For example, a manager node may issue a credential upon successful task completion, while a evaluator node may issue a credential certifying language proficiency or hardware compatibility. These credentials are cryptographically signed and can be stored locally or on decentralized storage systems like IPFS.

A Worker's active state may consist of:

- **Capabilities:** Signed credentials that define the Worker's functional attributes (e.g., `/language/nl`, `/gpu:nvidia`, `/model:llama2`). Many capabilities may include expiration or re-certification requirements.
- **Join Date:** A timestamped VC referencing the Worker's initial registration or first validated task.
- **Task History:** A structured record of completed tasks, including validation outcomes, reward values, and associated timestamps
- **Reputation Metadata:** Derived metrics such as average approval rate, throughput, or task category specialization. These may be computed locally or proven via zero-knowledge proofs.

Each VC is bound to the Worker's DID and can be used to generate proofs of eligibility, skill, or performance before/during task assignment.

Privacy and Selective Disclosure

The protocol is built to maximize privacy while preserving accountability. Workers are never required to reveal personally identifiable information. Instead, they generate cryptographic proofs from their credential sets. For instance, a Worker may be asked to:

Prove that you have completed over 100 audio transcription tasks in the last 30 days with at least a 95% approval rate, without revealing exact timestamps or task identifiers.

This level of selective disclosure is made possible through zero-knowledge proof systems such as BBS+ signatures or zk-SNARKs, allowing Workers to control what information they share and with whom.

Portability and State Synchronization

Worker identities and credentials are portable across devices. Because credentials are issued and signed by external nodes, they can be stored either locally (in encrypted vaults) or through decentralized storage (e.g., IPFS). A Worker switching devices can re-import their DID and synchronize their credential set, re-establishing their full reputation graph. Since this state is verifiable and not reliant on any central authority, continuity is preserved without compromising decentralization.

2.5 Task Lifecycle: From Request to Settlement

The Effect AI Protocol automates the full lifecycle of a task across a decentralized, peer-to-peer network. It connects Providers, Managers, and Workers without the need for centralized coordination. The following describes how a typical task moves through the system.

Creation

A Provider creates a task by submitting relevant task data to a Manager Node, along with a proof of escrowed deposit handled by the on-chain settlement layer. Each task is associated with a Template ID which is in turn associated with an app ID and App Data, which specify the lifecycle logic and required capabilities.

Example: A data labeling provider submits 1,000 image classification tasks, targeting Workers with the tag `/human/data-labeling`.

Discovery and Matching

Manager nodes receive tasks and scan the Execution Mesh for eligible workers using delegation logic defined in the Application Layer. This process may involve filtering by capability, such as specific GPU hardware, language proficiency, or available machine learning models, and applying routing strategies like FIFO queues, auctions, pubsub broadcasts, or round-robin assignment. Additional filters may take into account a worker's reputation or real-time availability.

Once a suitable worker is selected, the manager opens a direct peer-to-peer stream and transmits the task payload securely.

Execution

The Worker Node processes the task based on the specified input format. For tasks targeting human workers, this often involves HTML/JavaScript templates rendered in the browser; for machine/compute-based workers, this may consist of structured data and/or executable code. Upon completion, the result is returned to the originating Manager Node over the same encrypted channel.

Validation and Settlement

When a result is received, the Manager validates it according to rules defined in the Application Layer. Validation may be automated (using heuristics, thresholds, or checksums), consensus-based (by comparing outputs from multiple workers), or custom (via API calls, majority voting, or reputation-weighted scoring).

If the result is accepted, payment is triggered. To minimize on-chain overhead, multiple results are batched and settled using zero-knowledge proofs. Managers receive a coordination fee, and workers can redeem their rewards directly through the protocol's settlement layer.

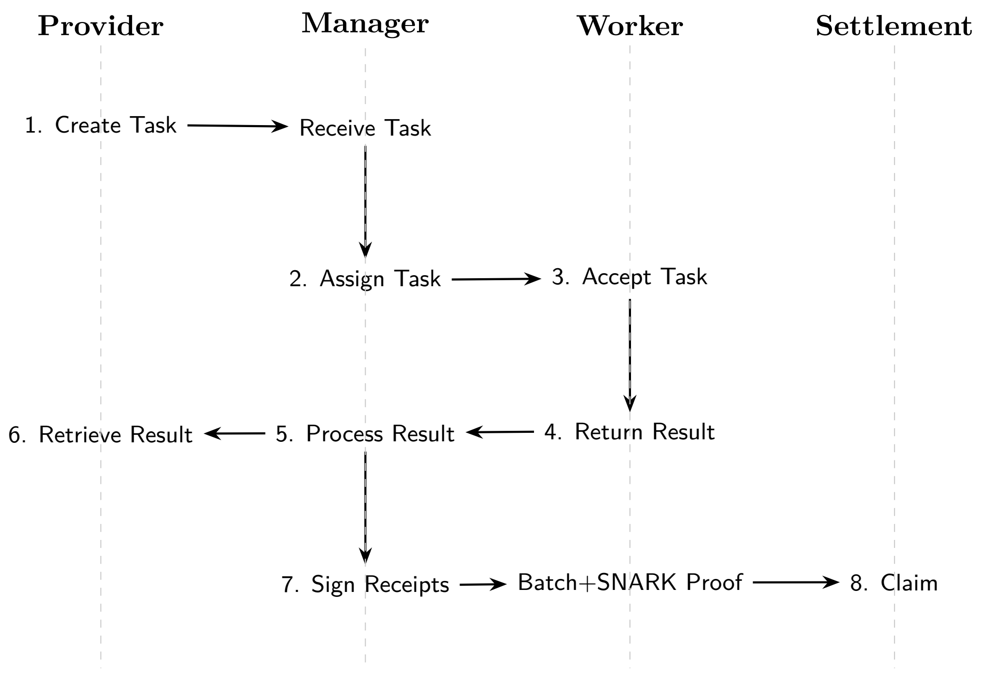


Diagram 2: Task Execution Flow

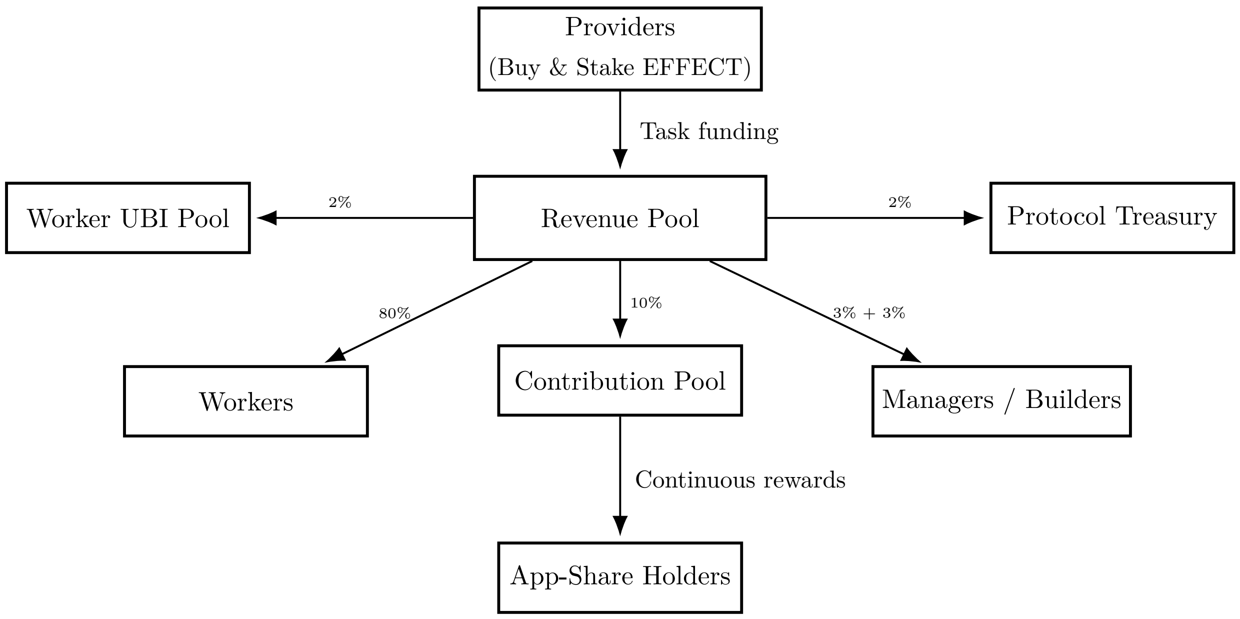
3 Token Economics

Effect AI's native token, `EFFECT`, provides the utility and governance backbone of the network. It is designed to balance growth, security, and fair participation across the network. The primary utility functions of the token are:

- **Participation Incentives** – Participants in the network, including Managers, DAO members, Workers, and Stakers, receive a percentage of rewards as a compensation for their input.
- **Staking** – `EFFECT` tokens are staked by Managers, Workers, and others, in order to participate in the network. Staking provides economic security and serves as a signal of trust and reputation.
- **Task Payments** – Providers use `EFFECT` tokens to pay for task execution. Payments are held in an escrow on-chain for trustless settlement.
- **App Fees** – Builders can attach custom fees to their deployed Apps, collected from Providers and/or Workers. These fees may be paid in `EFFECT` or any other DAO-approved token.
- **Payment Gateways** - When other means of payment are used on the network, such as DAO-approved tokens or FIAT, `EFFECT` tokens are used as collateral.

3.1 Incentives

The network generates natural incentives through Providers who buy `EFFECT` tokens to fund tasks. On top of this flow, each application runs its own on-chain reward system in the form of two app-specific pool. These pools distribute rewards to Workers, Managers, Builders, and App-share holders, creating a self-sustaining incentive loop that ties rewards directly to participation and performance.



Effect Protocol Tokenomics Flow

3.1.1 Revenue Pool

The Revenue Pool is one of the two core incentive pools tied to each application. It receives incoming `EFFECT` tokens from Providers who fund tasks, and distributes them according to a fixed fee structure with exact rates subject to change according to DAO governance:

- **80%** compensates Workers who complete tasks.
- **2%** funds the core protocol as a Network Fee.
- **2%** supports the Worker Universal Basic Income (UBI) Pool for online availability rewards.
- **3%** is allocated to the application's Builder as a Builder Fee.
- **3%** goes to the active Manager coordinating the task.
- **10%** is directed to the application's Contribution Pool for long-term ecosystem funding.

3.1.2 Contribution Pool

The Contribution Pool is the second core incentive pool linked to each application. It serves as the app's long-term economic backbone by issuing non-transferable app-shares in exchange for contributions. These shares reward both capital providers and active contributors, ensuring that financial support and meaningful participation are directly tied to a proportional claim on the application's future revenues.

Participation

- **External Deposits:** Any network participant may deposit `EFFECT` tokens into the Contribution Pool to acquire non-transferable app-shares proportional to their stake.
- **Worker Share Conversion:** Workers accumulate Verifiable Credentials (VCs) as cryptographic attestations of their validated task contributions. Workers may convert these VCs into app-shares according to protocol-defined rules, enabling reinvestment of labor into ownership.

The conversion ratio between VCs and app-shares depends on:

- The Worker's stake and stake age (*Effective stake*) in the network, acting as a multiplier on their verifiable credentials (VCs).
- Total task volume and Worker contribution relative to the app's activity.
- Quality and reliability metrics, such as approval rate and reputation scores.
- Scarcity or premium of specific capabilities (e.g., specialized skills or hardware).

Conversion parameters are configurable by the DAO and may include caps or vesting schedules to ensure sustainable growth.

Reward Distribution

- A fixed percentage (10%) of all `EFFECT` tokens flowing into the application's Revenue Pool are redirected automatically to the Contribution Pool.
- These tokens are distributed proportionally to app-share holders, providing a continuous passive income stream.

This dual-entry system, combining direct capital deposits with credential-based share conversion, ensures that both contributors and investors are rewarded in a balanced way. It strengthens the alignment between labor and capital, fostering long-term growth and sustainability for each Effect AI application registered in the network.

3.1.3 Universal Basic Income Pool

To future-proof the role of human labor in an increasingly automated ecosystem, the protocol includes a mechanism for Universal Basic Income (UBI). This model supports human participation by offering periodic disbursements to Workers who remain online, available, and verifiably capable, even when not actively completing tasks.

Unlike traditional welfare systems, UBI in the Effect AI protocol is not a passive benefit. It is a structural reward designed to sustain a base layer of qualified human contributors. The mechanism encourages participation from underrepresented regions, provides resilience during low-demand periods, and fosters an always-ready workforce.

Eligibility is determined algorithmically based on factors such as:

- Sustained uptime and responsiveness
- Historical task performance and quality
- Valid, non-expired capability attestations
- Completion of identity and privacy requirements (e.g., DID registration)

UBI disbursements are intentionally modest. They are designed to supplement, not replace, task-based earnings. The DAO governs the funding schedule, payout structure, and distribution model to ensure long-term sustainability. Over time, the goal is for UBI to be entirely covered by protocol fees, making it a self-sustaining incentive system embedded in the core economics.

3.1.4 Multi-Token Payments with Burn Incentives

To strengthen cross-ecosystem adoption and incentivize external projects to route workloads through the Effect AI Protocol, Builders can unlock an optional **Multi-Token Payment Mode** by staking `EFFECT`. This staking requirement acts as a routing license, ensuring that all multi-token integrations still generate demand for `EFFECT` regardless of the payment currency.

Mechanic When activated, the Builder may configure their application to accept task payments in any fungible token approved through DAO governance (e.g., SPL/ERC-20 assets such as USDC, a partner project's native token, or in-game currencies). Payments are still escrowed and settled through the Settlement Layer, but follow an alternative flow:

1. **External Token Escrow:** Providers deposit the designated external token into the application's Revenue Pool.
2. **Automatic Burn:** A predefined percentage of each payment (e.g., 3–5%, set by the Builder) is permanently removed from circulation by sending it to an on-chain burn address.
3. **Distribution:** The remaining amount is distributed to Workers, Managers, Builders, and protocol fees according to the app's configured splits.

Ecosystem Incentives The burn mechanic creates deflationary pressure on the external token's supply, making this payment mode attractive for projects seeking to increase token utility. By tying the burn event to productive work completion, the mechanism becomes a *utility-driven token sink* rather than a purely speculative one.

Benefits for Stakeholders

- **For Builders:** Increases adoption potential by integrating directly with external token economies, while retaining a configurable economic model.
- **For Effect:** All multi-token payments still require EFFECT staking to activate, creating indirect demand for `EFFECT` even when it is not the primary payment token.
- **For External Projects:** Gains a tangible reason to direct workloads through Effect, turning task execution into a supply-reducing event for their token.
- **For Workers:** Provides the flexibility to earn in multiple currencies, broadening earning opportunities.

This feature allows Builders to design applications that bridge ecosystems while preserving Effect's core incentive structure. In practice, it can turn the Effect Protocol into a *token sink as a service* for partner projects, creating a powerful growth vector for the network as a whole.

3.2 Staking

Staking in the Effect Protocol secures the network, incentivizes participation, and ensures aligned behavior across its participants. The protocol uses both the **staked amount** and its **stake age** as a signal of commitment and reliability.

3.2.1 Stake Age and Effective Stake

Every stake has an associated **stake age** T, which increases over time as long as the stake remains untouched. When additional stake is added, the stake age is recalculated as a weighted average:

$$StakeAge_{new} = \frac{A_{old} \cdot T_{old} + A_{new} \cdot T_{new}}{A_{old} + A_{new}} \tag{1}$$

The protocol uses a derived quantity called **effective stake** to influence role-specific behavior:

$$EffectiveStake = A \cdot f(T) \tag{2}$$

Where:

- A is the current staked amount
- T is the stake age
- $f(T)$ is a stake age multiplier function, e.g. $f(T) = 1 + \alpha \cdot \log(1 + T)$

The specific function $f(T)$ can be tailored per role and use case, and is subject to adjustment through DAO governance.

3.2.2 Staking Across Roles

Every role in the Effect AI Protocol has a unique staking mechanic designed to strengthen economic security, align incentives, and reward long-term commitment. These mechanics ensure that participants who contribute more through reliability, capital, or quality of work — are economically recognized, while dishonest or low-quality behavior can be penalized through stake slashing.

Managers Managers must maintain a minimum effective stake to remain bonded in the network. This bond serves as economic collateral, slashed in the event of misbehavior such as fraudulent result submissions, incorrect payouts, or unfair task routing. The longer the stake remains untouched, the more trust weight it carries, meaning that established Managers have more to lose if they act dishonestly. Slashing affects both recent and aged stake, making long-term misconduct especially costly.

Workers Workers can optionally stake `EFFECT` to boost the conversion rate of their Verifiable Credentials (VCs) into application shares, directly increasing their share of future revenue streams. Effective stake also acts as a reward multiplier, granting higher payouts for completed tasks. This mechanism rewards both proven capability and long-term loyalty.

Providers Providers, which include task posters and API relayers, may stake `EFFECT` to receive discounts on protocol fees. These discounts scale with effective stake, rewarding well-staked, long-standing Providers with lower operational costs.

Builders Builders may optionally stake `EFFECT` to increase the visibility and recognition of their applications in protocol-wide discovery mechanisms. By increasing their effective stake, builders may also unlock additional protocol-level features. A key stake-enabled feature is the **Multi-Token Payment Mode**, which allows Builders to accept task payments in any fungible token approved through DAO governance. Higher effective stake also increases the Builder's fee share per completed task in their application, offering a direct financial incentive for long-term commitment to their projects.

Evaluators Evaluator Nodes, responsible for issuing and maintaining capabilities, must maintain a minimum effective stake to be eligible to issue capability credentials. Beyond this requirement, they can stake additional `EFFECT` to enhance the visibility and perceived trustworthiness of their certifications. Higher effective stake increases the weight of their issued capabilities in Manager selection algorithms, making Workers certified by well-staked Evaluators more likely to receive task assignments.

3.3 Distribution

A sustainable decentralized network relies on a fair and transparent token distribution to ensure resilience, security, and broad participation in governance. The `EFFECT` token entered circulation on January 1, 2025, consolidating the deprecated `EFX` and `NFX` network tokens into a single unified asset on Solana.

A snapshot was taken on **January 1, 2025 at 12:00 UTC**, granting any holder of more than five `EFX` or `NFX` tokens a claimable `EFFECT` allocation. Conversion rates were fixed at **1:1 for `EFX`** and **1:8 for `NFX`**, with all claimed tokens automatically staked under a **7-day lock** followed by a **30-day linear release** to promote network stability.

The initial allocation of the **520,000,000 `EFFECT`** tokens is displayed in table 1.

Allocation	**Amount**	**Supply**	**Notes**
EFX Holders	185M	36%	1:1 conversion; staked with release schedule
NFX Holders	60M	12%	1:8 conversion; staked with release schedule
Staking Incentives	50M	10%	Allocated by DAO proposals #184 and #195
DAO Treasury	49M	9%	Locked until Effect DAO launch on Solana
Liquidity, Partnerships, Marketing	68M	13%	For liquidity provision and network growth
Development Fund	100M	19%	4-year linear vesting for protocol development
Migration Reserve	6M	1%	Covers unforeseen migration-related expenses

Table 1: Initial distribution of the `EFFECT` token.

This distribution model builds on a *multi-year, community-driven supply base* inherited from the prior token on EOS, where token ownership was already widely dispersed with no single entity holding more than a small fraction of total supply.

3.4 Governance

The specification and implementation of the Effect Protocol are governed by its community through a decentralized autonomous organization (DAO).

The DAO oversees the evolution of the protocol, including upgrades, funding campaigns, and other ecosystem initiatives. All community members are invited to participate in governance by contributing to discussions, submitting proposals, and voting on key decisions.

The entire codebase is licensed under the MIT License. We encourage open participation via our source code repositories and social channels. We see contributions, feedback, and proposals as essential to the evolution of the protocol and these are openly discussed and reviewed during monthly community calls.

4 Roadmap: What's ahead for Effect AI

The Effect AI Protocol will be developed and deployed in progressive phases, each expanding the network's capabilities and adoption. Timelines are indicative and may adapt based on testing, security audits, and community feedback.

Phase 1: Network Core, First Apps, Worker Onboarding (In Progress)

- Deploy initial Execution Layer with capability-based gossip topics.
- Release of first partner applications
- Open-source SDKs for task creation and management.
- Launch Worker and Manager client software (desktop + browser).
- Introduce Universal Basic Income (UBI) for human Workers.
- Zero-Knowledge Proof Payment system.

Phase 2: Application Layer and Composable Workflows

- Enable DAG-based application creation and on-chain registration.
- Launch first Builder tools and workflow templates.
- Release open-source SDKs for capability declaration and task assignment.
- Support app-to-app calls for composable pipelines.
- Roll out Revenue Pool and Contribution Pool incentive mechanisms.
- Expand capability categories via third-party Evaluators.
- Re-establishment of Effect DAO and basic governance

Phase 3: Settlement Layer Enhancements

- Deploy Settlement Layer on Solana with zk-SNARK batch payment proofs.
- Support multi-currency rewards with stablecoin settlement.
- Integrate slashing, staking, and dispute resolution mechanisms.
- Release multi-chain settlement verification (Ethereum, L2 rollups).

Phase 4: Ecosystem Expansion and Governance

- Transition to completely community-driven governance via DAO framework.
- Onboard large-scale Providers and enterprise integrations.
- Launch cross-domain marketplaces for AI capabilities.
- Enable cross-network bridges with other decentralized compute and AI protocols.

5 Conclusion

Effect AI introduces a modular and trustless framework for coordinating human and machine intelligence at scale. By combining an Execution Layer for real-time, peer-to-peer work, an Application Layer for composable workflows, and a Settlement Layer for secure, on-chain incentives, the protocol enables participants to create, route, and verify complex AI-driven tasks without reliance on centralized intermediaries.

Each role in the network — Worker, Manager, Builder, Provider, and Evaluator — interacts through well-defined capabilities and tailored staking mechanics, ensuring that incentives are aligned across short-term execution and long-term ecosystem growth. The dual-pool economic model further anchors applications in sustainable value flows, rewarding both active contributors and strategic supporters.

Through its open, extensible architecture, Effect AI is positioned not only as an execution network, but as a foundation for distributed, composable intelligence. A permissionless marketplace where AI capabilities can be discovered, monetized, and recombined in ways that were not possible before.

The protocol's roadmap envisions continuous expansion of features, integrations, and governance, ensuring that as AI technology evolves, so too does the network that supports it. By aligning economic incentives with open participation, Effect AI aims to become the global coordination layer for human and machine collaboration.