



# Sage Litepaper

---

The first permissionless protocol for incentivized agent context sharing.

Sage turns prompts, libraries, and agent context into shared, ownable, and continuously improving community assets. Builders publish once, communities curate and reward, agents discover and use the best context on-chain and from **IPFS**, and value flows back to the people who made it useful. The result is a decentralized, self-improving metaprompting system that communities control end to end.

---

## Why Now?

AI work moves fast, but community memory is brittle. Great prompts and context live in scattered docs, DMs, and private repos. They decay as models change. There is no portable way for communities to store, version, and improve shared context, and no native path to attribute or reward the people who move the state of the art forward. Sage addresses four blockers:

### 1. Lossy Knowledge

Context is copied and pasted, not curated. There is no global index or version trail that communities can trust.

### 2. Prompt Decay

Models evolve weekly. Yesterday's best pattern quietly breaks, and no one sees the diff or the fix land in time. Communities need a place to vote, fork, and ship updates together.

### 3. Zero Attribution by Default

Reuse is the norm, but provenance and revenue share are not. That erodes incentives to publish the good stuff.

### 4. No Community Treasury Loop

Work happens, but there is no clean way to fund curation, pay contributors, and compound gains inside a shared treasury across cycles.

Sage exists to make community knowledge durable, discoverable, and rewarded.

---

## What Sage Is

Sage is a protocol that lets communities publish prompt libraries on **IPFS**, register them on-chain, and govern upgrades through their own **SubDAO**. A manifest file defines an entire library, and one proposal updates the whole set. Discoverability runs through a subgraph and an **MCP** server so agents can search and pull context directly. Interfaces include a **CLI**, a web app, and Model Context Protocol (**MCP**) endpoints for agent tooling like Claude Desktop.

- **Publish:** Push a manifest and prompt files to **IPFS**. Get immutable CIDs.
- **Propose:** Each manifest is one governance proposal in your **SubDAO**. Vote, queue, and execute through a Timelock.
- **Discover:** Agents, the **CLI**, and the web app read via subgraph first and fall back to RPC.

- **Use: MCP** tools return content by CID so agents can load context on demand.

This approach gives AI artists, prompt engineers, and research groups an easy path to share and store their working prompts with permanence, version history, and clear community ownership.

---

## What Makes It Different

- **Community First**  
Each community has a **SubDAO** with its own library, governance, and treasury. Forks and remixing are explicit, not implicit. Curation happens in public.
  - **Manifest-First Upgrades**  
Ship atomic updates to entire collections. That cuts gas, reduces proposal fatigue, and keeps libraries coherent.
  - **Incentives Built-In**  
**Boosts** reward participation. **Bounties** pay for new prompts. **Premiums** let creators sell encrypted content with on-chain receipts. Proceeds route to **SubDAO** and protocol treasuries.
  - **Agent-Native Discovery**  
An **MCP** server exposes search, fetch, and manifest validation. Agents can query libraries directly, then pull the exact context they need.
  - **Production on Base**  
Contracts, subgraph, and flows run on Base Sepolia today, designed to graduate to Base mainnet following a gated launch sequence.
- 

## The Self-Improving Metaprompting Loop

1. **Publish**  
Authors push a library with a human-readable manifest. **IPFS** CIDs anchor content.
  2. **Curate**  
The community proposes variants, votes, and executes upgrades. Approver Councils can be badge-gated for fast tracks when needed.
  3. **Reward**  
**Boosts** pay voters or contributors for specific decisions. **Bounties** fund new prompts. **Premiums** monetize high-effort work. Treasury routes and override guards are set by governance.
  4. **Use and Measure**  
Agents fetch context via **MCP**. Communities gauge a prompt's value through on-chain signals like forks, bounty completions, and governance proposals, creating a feedback loop for continuous improvement.
- 

## Core Components

- **Smart Contracts on Base**

**SubDAO** Factory, Governor and Timelock, LibraryRegistry, PromptRegistry, Treasury modules, optional **LBP** and **Bonding** integrations.

- **Storage and Discovery**

**IPFS** for payloads, subgraph-first reads across **CLI**, web, and **MCP**.

- **Tooling**

A complete **CLI**, a web app for discovery and drafting, and an **MCP** stdio server that plugs into Claude Desktop or an HTTP server for other agents.

---

## Incentives and Ownership

Sage powers three on-chain incentive rails that communities can combine.

### 1. Boosts

Parameterizable rewards for governance participation or contributor actions. Two manager contracts exist:

- Direct manager for per-voter payouts tied to a proposal.
- Merkle manager for pooled funding and claim-by-proof.

Both paths are governed through the **SubDAO** Timelock. The **CLI** supports **create**, **fund**, **set-root**, **status**, **claim**, and **finalize**.

### 2. Bounties

Announce tasks, award winners, and pay out from the Timelock. This system is event-only by design. Fund the Timelock with ETH or an ERC-20 before executing payout proposals.

### 3. Premium Prompts

Encrypt content client-side, pin to **IPFS**, and gate decryption with Lit v7 and ERC-1155 receipts. Buyers pay in USDC, get a receipt NFT, and decrypt locally with session signatures. Pricing and treasury routing are on-chain. No reveal server is needed.

### Paid Pinning Revenue

When libraries are pushed with paid pinning, the service receives USDC and calls on-chain attribution. The default split routes 80% to the **SubDAO** treasury and 20% to the protocol treasury.

### SBTs and Badge-Gated Fast Tracks

Non-transferable **SBTs** recognize contributors and can gate Approver Council permissions to move quickly on scoped updates, with function selectors allow-listed by governance.

---

## For Communities

- **AI Artists and Prompt Engineers**

Store prompts permanently with **IPFS** CIDs, track versions, and get discovery out of the box. Add premium entries when you want to sell high-effort sets.

- **Research Groups and Guilds**

Use **Bounties** for new methods, **Boosts** for review cycles, and **SBTs** for reputation. Fork libraries when styles or standards diverge.

- **Agent Builders**

Query the subgraph or **MCP** to pull the latest approved context and keep agents in sync as libraries evolve.

---

## Why Base

Base is where on-chain apps have grown fastest this cycle, with significant TVL and a culture of shipping public goods. Aerodrome has become the liquidity hub for Base, with community emissions and LP tooling that many projects model against. Projects launching on Base often use Fjord's **LBPs** to align distribution with community discovery. Sage follows these patterns for its treasury and launch readiness.

---

## Security and Launch Status

- **Security Model**

All writes flow through a Governor to a Timelock. Emergency pause, least-privilege roles, and upgrade paths are governance-controlled.

- **Testnet Readiness**

Fully working end-to-end on Base Sepolia, including library updates, governance, treasury demos, **MCP**, and the app.

- **Mainnet Launch Sequence**

A phased plan covers MEV risk for initial liquidity, private mempool usage for pool creation, and transaction ordering constraints. Use a multisig and follow the step plan exactly.

---

## Token and Treasury, in Plain Terms

The Sage economy provides three distinct pillars for a self-sustaining ecosystem:

1. **Treasury Growth:** The protocol's main treasury is funded by a one-time **LBP** launch and ongoing **bond** sales. This ensures the long-term financial health and stability of the **SXXX** token.
2. **Creator Rewards:** **SubDAOs** receive grants from the main treasury and use those funds to post **bounties**. This creates a direct, work-to-earn loop for prompt engineers and other contributors.
3. **Community Governance:** **SXXX** token holders stake their tokens in **SubDAOs** to vote on content curation and community decisions, ensuring that the most valuable context is promoted and rewarded.

---

## From On-Chain to On-Device: The Power of the CLI

Beyond governance, the Sage **CLI** is a powerful bridge between the decentralized protocol and local development. It's designed to pull complex, on-chain agent systems directly into your workflow.

With a single command, you can parse an entire **SubDAO**'s curated library and generate ready-to-use local files:

- **For Custom GPTs:** Generate a `system-prompt.txt` file, perfectly formatted for use in OpenAI's GPT builder.
- **For Coding Agents:** Output a series of `.js` or `.py` files, where each file contains a specific prompt for a step in your automated coding workflow.
- **For AI Art:** Create a JSON file of tested, high-quality prompts formatted for the Midjourney or Stable Diffusion API.

This transforms decentralized curation into a practical, daily tool for developers and creators.

---

## Get Started

### Install and connect

```
npm i -g @sage-protocol/cli
sage wallet connect --type cast
sage doctor
```

### Publish a library

```
sage library scaffold-manifest
sage library lint manifest.json
sage library push manifest.json --pin
sage library propose manifest.json --subdao 0xYourSubDAO
```

### Run the MCP server for agents

```
node cli/mcp-server-stdio.js
# or start HTTP and set NEXT_PUBLIC_MCP_HTTP_URL
```

Tools include: `search_onchain_prompts`, `get_prompt_content`, `list_subdaos`, `get_library_manifests`.

### Offer a premium prompt

```
sage premium-pre authsig --out ./authSig.json
sage premium-pre publish --in ./prompt.md --name "Title" --description
"Desc" \
  --subdao 0xYourSubDAO --price 2.50 --auth-sig ./authSig.json
```

## Incentivize participation

```
# Direct manager, per-voter
sage boost create --governor 0xGov --proposal-id 1 --per-voter 250000 --
max-voters 100 --min-votes 1

# Merkle pool
sage boost fund --proposal-id 1 --amount 25000000
sage boost set-root --manager 0xMerkle --proposal-id 1 --root 0xRoot
```

## Set up SBTs and an Approver Council

```
sage sbt propose-mint --to 0xContributor --badge 1 --uri ipfs://CID
sage council set-config --badge $SBT --badge-id 1 --target
$PROMPT_REGISTRY
sage council allow --selector
addForkedPrompt(string,string,string,string,address)
```

---

## Roadmap

- **Now SubDAO** factory, Library and Prompt registries, governance flows, **Boosts, Premiums, SBTs, MCP, CLI**, and web app are live on Base Sepolia.
- **In Progress** Coming soon site, **CLI** V2 with meta prompting agent, Fjord partnership, and web app.
- **Mainnet** Execute the launch sequence, wire canonical liquidity, and transition ownership entirely to DAO contracts.
- **Growth** First community libraries in creative AI, research, and agent ops. Grants and **bounties** focused on subjective domains where human curation matters most.

---

## License

MIT. See the repository for full docs, guides, and examples.