

- [Overview](#)
 - [Hypothesis](#)
 - [Core Idea](#)
 - [The Fund \(Fund OS\)](#)
 - [The Agents](#)
 - Two types of agents:
 - [The Interface \(ABI Boundary\)](#)
 - [Reinforcement Learning \(High-Level\)](#)
 - [Key Principles](#)

Overview

Hypothesis

We can model a hedge fund as a **deterministic operating system** that exposes a controlled, versioned interface to independent decision agents (agentic portfolio managers).

Core Idea

The fund is the environment. Agents are digital portfolio managers that learn how to act within it.

The fund defines reality: what information exists, what actions are possible, and what constraints apply. Agents do not trade markets directly - they learn to operate the platform the fund provides.

The Fund (Fund OS)

The fund replicates the internal platform that hedge funds provide to portfolio managers. It is a deterministic, rule-based system that owns all interaction with the external world.

The fund replicates the platform hedge funds create and provide to portfolio managers. This platform includes market data, pricing models, order execution interface, etc. The

fund will: - own market connectivity, execution/internal crossing, risk, and accounting -
define what information exists via explicit interfaces - enforce all constraints
deterministically

The Agents

Agents replicate the behavior of portfolio managers at traditional hedge funds. All agents are external processes and interact with the fund exclusively through the ABI.

Two types of agents:

- **Inline Decision Agent** (Discretionary PM)

- Lives inside the trading loop
- Reacts continuously to fund state
- Produces intent every timestep

Analogous to a discretionary PM who uses human judgement to trade off dashboards, charts, and signals provided by the firm.

- **Strategy-Authoring Agent** (Quant PM)

- Operates outside the per-tick trading loop
- Designs systematic strategies
- Writes code/configs/models
- Spins up its own trading process
- Monitors the process during market hours
- Shuts it down, modifies it, or replaces it.

Analogous to a quant PM who builds a strategy, launches it as a service, and lets it trade autonomously.

Agents will be able to: - observe exposed fund state - propose trading intents - never access internal fund systems directly

The Interface (ABI Boundary)

Communication between the fund and agents occurs via a stable, versioned ABI, similar in spirit to SBE schemas or syscalls. - Binary-compatible - POD-only (plain data) - Explicit memory layout - Language agnostic This interface allows: - allows shared memory IPC - enables agents written in Python, C++, or other languages - guarantees reproducibility across research and production If a piece of information is not exposed via the ABI, it does not exist to the agent.

Example ABI (C++)

```
// fund_state_v1.h

struct FundStateV1 {
    double mid_price;
    double volatility;
    int64_t position;
    double cash;
    double unrealized_pnl;
};

struct AgentActionV1 {
    double target_position;
};
```

These structs: - can be placed directly in shared memory - are readable from Python, C++, or other languages - define a strict contract between the fund and agents

Reinforcement Learning (High-Level)

Reinforcement learning is used to train inline decision agents.

At a high level: - State: exposed fund state (prices, positions, analytics, risk metrics) - Action: trading intent (e.g., target position or risk scaling) - Reward: PnL and risk-adjusted performance metrics

Because the state and action spaces are continuous and high-dimensional, the system is designed to support policy-based neural network methods (e.g. policy-gradient approaches).

Key principles: - RL training occurs outside the fund process - The fund provides a deterministic environment for replay and simulation - Trained policies are deployed as external agents using the same ABI as in research

Key Principles

1. **Trading infrastructure and agents evolve independently**
 - Agents are part of external processes.

- The underlying model of the agent will be finetuned over time through reinforcement learning.

2. **Stable ABI Boundary** Communication between the fund and agents occurs via a versioned, binary-compatible interface (POD structs), enabling: - process isolation - language independence - reproducible simulation and replay This boundary is treated with the same care as an exchange protocol.
3. **Fund Correctness Over Agent Intelligence** The fund OS must remain safe and deterministic regardless of agent behavior.
4. **Research is external** Reinforcement learning and experimentation occur outside the fund process and consume the same interface used in production.