ZVAKTHOR — Deterministic Control for Autonomous Trading Agents

An algorithmic execution platform for global financial markets that separates **intelligence** from **control**: specialist agents analyse markets independently, while an isolated Rust core validates every decision for risk and execution eligibility before any order reaches the market. Every decision — accepted or rejected — is recorded and replayable.

Non-custodial by design · **Deterministic** kill-switch · **Boundary:** Python intelligence / Rust governance

ARCHITECTURE — THREE LAYERS, ONE ENFORCED BOUNDARY

LAYER 1	LAYER 2	LAYER 3
Discovery	**Reasoning**	**Execution Control**
Signal generation and candidate search, with a search objective designed not to converge on crowded alpha.	Specialist agents analyse markets independently and produce decisions with supporting context and constraints.	An isolated Rust core validates risk and execution eligibility. Nothing reaches the market unvalidated.

WHAT THE ARCHITECTURE ENFORCES

- **No unvalidated order** — validation is a precondition of execution, not a post-trade check.
- **Full decision lineage** — accepted and rejected decisions are both recorded and replayable.
- **Bounded authority** — explicit scope, validity and revocation, enforced at the authorisation boundary.
- **Reproducibility** — a decision is reproducible only from a versioned execution context.
- **Failure containment** — no valid-looking partial state; committed state survives process or node failure.

CUSTODY, STATUS, AND THE LAYER THAT DECIDES OUTCOMES

> **Non-custodial by design.** Designed to be licensed to clients who run it on their own broker accounts. Capital stays in the client's name, and the system never takes discretionary control.

Status. Core architecture and the first agent group are validated end to end in a demo environment. Remaining agents activate as their data infrastructure is funded. No live performance figures are published.

Research premise. The Operating-Layer Dominance Hypothesis (Paper 1) argues that agent behaviour is driven by interface and execution design, not model reasoning quality alone. **IMPERION** applies the same governance discipline to automating AI system architecture decisions.

RESEARCH PROGRAM — FOUR SSRN WORKING PAPERS

PAPER 1	The Operating-Layer Dominance Hypothesis: Why Agentic Trading Alpha Is an Interface Problem, Not a Model Problem — ssrn.com/abstract=7472064
PAPER 2	From Backtest to Deployment: A Staged-Capital Protocol for Evaluating Agentic Trading Systems — ssrn.com/abstract=7475338
PAPER 3	A Three-Layer Reference Architecture for Deterministic Control in Agentic Trading Systems — ssrn.com/abstract=7488978
PAPER 4	Why Discovery Layers Guarantee Crowded Alpha: A Design Case for Quality-Diversity Search in Agentic Trading Systems — ssrn.com/abstract=7488920

VERIFIABLE IDENTITY

ORCID iD: **0009-0009-4656-5983** SSRN author page: ssrn.com/author=13197688 Amazon author page: amazon.com/stores/author/B0GQ66415M
Reference implementation: github.com/hadisoufi20/deterministic-control-reference

Hadi Soufi — AI Systems Architect · Founder, ZVAKTHOR · Istanbul, Türkiye · hadisoufi20@gmail.com · hadisoufi.com/ · www.linkedin.com/in/hadi-soufi-463170266