SOI — An Experimental Investment Operating System

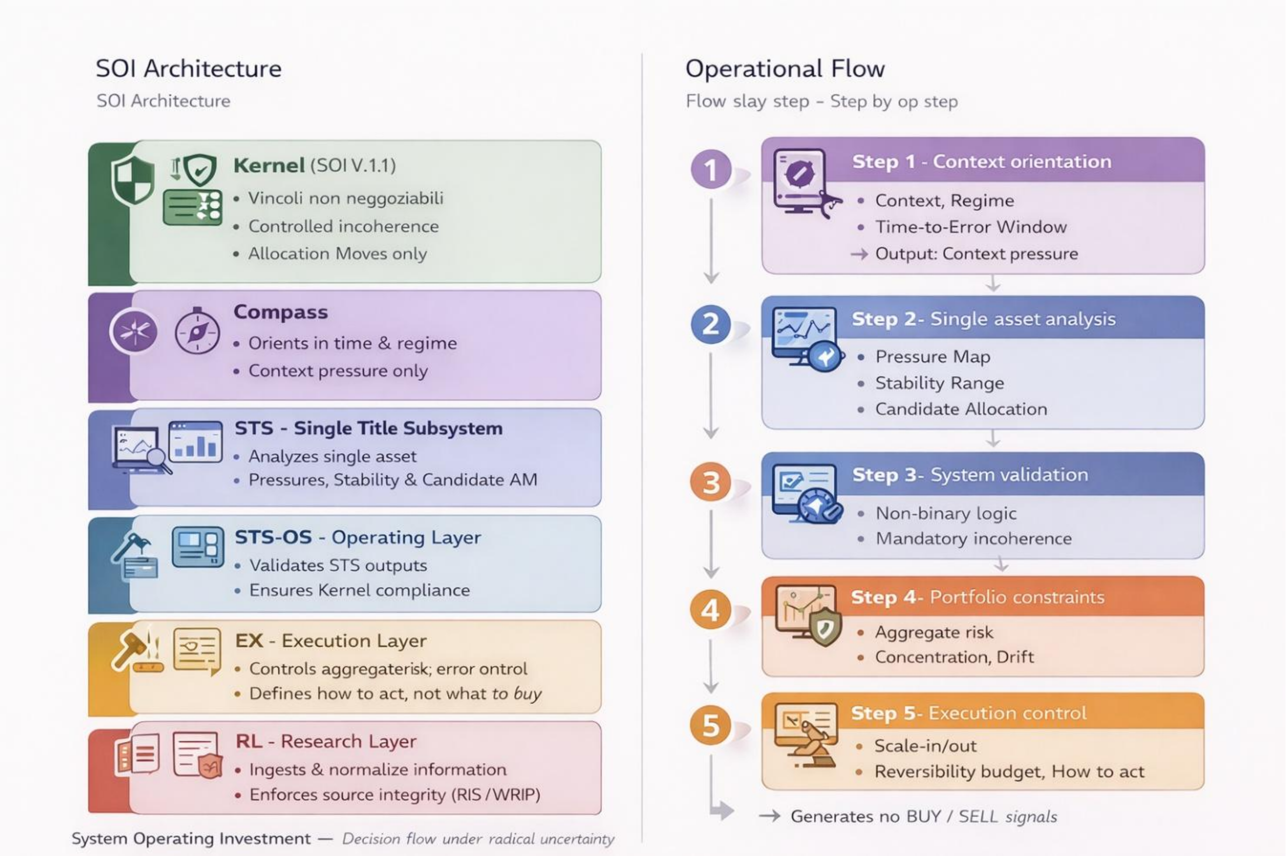
Process, Architecture, and LLM-Assisted Usage

This document describes how the SOI (System Operating Investment) framework is used in practice.

SOI is not a trading strategy and does not aim to predict markets.
It is a decision-support system designed to operate under uncertainty, preserve capital, and absorb incoherence without collapsing.

This document focuses on:

- *the operational flow of the system*
- *how each module interacts*
- *how Large Language Models (LLMs) are used as analytical tools, not decision-makers*



SOI Architecture

SOI Architecture

Kernel (SOI V.1.1)
- Vincoli non neggoziabili
- Controlled incoherence
- Allocation Moves only

Compass
- Orients in time & regime
- Context pressure only

STS - Single Title Subsystem
- Analyzes single asset
- Pressures, Stability & Candidate AM

STS-OS - Operating Layer
- Validates STS outputs
- Ensures Kernel compliance

EX - Execution Layer
- Controls aggregaterisk; error ontrol
- Defines how to act, not what *to buy*

RL - Research Layer
- Ingests & normalize information
- Enforces source integrity (RIS / WRIP)

System Operating Investment — *Decision flow under radical uncertainty*

Operational Flow

Flow slay step – Step by op step

1 **Step 1** - Context orientation
- Context, Regime
- Time-to-Error Window

→ Output: Context pressure

2 **Step 2**- Single asset analysis
- Pressure Map
- Stability Range
- Candidate Allocation

3 **Step 3**- System validation
- Non-binary logic
- Mandatory incoherence

4 **Step 4**- Portfolio constraints
- Aggregate risk
- Concentration, Drift

5 **Step 5**- Execution control
- Scale-in/out
- Reversibility budget, How to act

→ Generates no BUY / SELL *signals*

2. High-level architecture

The SOI stack (top-down)

Kernel (SOI v1.1)
– defines non-negotiable constraints
– enforces controlled incoherence
– produces Allocation Moves, not buy/sell decisions
Compass
– orients the system in time and regime
– provides context pressure, not timing signals
STS (Single Title Subsystem)
– analyzes a single asset
– produces pressures, stability, and candidate allocation
STS-OS
– validates and integrates STS outputs
– ensures Kernel compliance
EX (Execution Layer)
– manages reversibility and error control
– defines how actions are executed, not what to buy
PM (Portfolio Management)
– controls aggregate risk and survivability
RL (Research Layer)
– ingests and normalizes information
– enforces source integrity (RIS / WRIP)

3. Operational flow (step-by-step)

Step 1 — Context orientation

- Compass evaluates regime, transition, and friction
- Output: context pressure + Time-to-Error Window

Step 2 — Single asset analysis

- STS analyzes business, valuation, risks, and expectations
- Output:
    - Pressure Map
    - Stability Window
    - Allocation Move (candidate)
    - Incoherence Log

Step 3 — System validation

- STS-OS checks:
    - completeness
    - non-binary logic
    - presence of mandatory incoherence

Step 4 — Portfolio constraints

- PM evaluates aggregate risk, concentration, correlation drift

Step 5 — Execution control

- EX selects *how* to act:
  - scale-in / scale-out
  - reversibility budget
  - execution envelope

At no point does the system generate a BUY or SELL signal.

---

4. Role of Large Language Models (LLMs)

How LLMs are used

The SOI framework has been tested using multiple LLMs, including:

- GPT-based models (ChatGPT)
- Grok

LLMs are used as:

- analytical assistants
- text processors
- scenario expanders

What LLMs do NOT do

- they do not decide allocations
- they do not override Kernel constraints
- they do not produce trading signals

Why multiple models are used

- to expose narrative bias
- to generate conflicting interpretations
- to deliberately **create incoherence** that the system must absorb

Disagreement between models is treated as information, not noise.

5. Why controlled incoherence is intentional

- Markets contain irreducible contradictions
- Forcing coherence often hides risk

- SOI keeps at least one structural incoherence active at all times

Example:

- Graham-style conservative value
- Damodaran-style narrative valuation

Both coexist. Neither is eliminated.

---

6. Limitations and open questions

- SOI does not optimize returns
- It requires time and discipline
- It is not suitable for high-frequency or automated trading
- Some components are qualitative by design

Open questions:

- where does complexity exceed benefit?
- when does incoherence become unfalsifiable?

---

7. Experimental status

SOI is an experimental framework.

This document is not financial advice and not a commercial proposal.

Feedback, critique, and external testing are explicitly encouraged.