Autonomous Hedge Fund – Executive Specification (ExecSpec.md)

Fund Mission and Strategic Objective

The autonomous hedge fund's **mission** is to generate consistent, risk-adjusted returns by leveraging a network of specialized AI agents operating in concert. The **strategic objective** is to employ data-driven, multi-strategy trading (across defined markets and instruments) to outperform benchmarks while strictly controlling risk and adhering to all regulatory constraints. The fund is designed to run **fully autonomously** under AI governance, mimicking a traditional hedge fund's structure with AI agents in key roles. All activities must align with the fund's mandate, which emphasizes **alpha generation within predefined risk limits** and **zero tolerance for compliance breaches**.

Key elements of the strategy include: - **Diversified AI Decision-Making:** Use multiple specialized agents (research, risk, execution, etc.) to ensure decisions are well-rounded, reducing biases or blind spots [1]. This multi-agent approach yields more balanced and robust trading decisions than any single model could. - **Continuous Market Engagement:** Operate continuously (or within market hours) to monitor opportunities and risks in real-time, allowing the fund to react faster than human-driven funds. - **Adaptive Learning:** Incorporate feedback loops where the system learns from outcomes (e.g. trade results, errors) to refine its strategies over time, improving performance and resilience.

Constraints and Compliance Mission

Risk Constraints: The fund operates under strict risk parameters. It defines **risk limits** (e.g. maximum position sizes, sector exposures, leverage ratios, stop-loss thresholds, Value-at-Risk limits) that all trading decisions must obey. No single trade or portfolio position should violate these limits as determined by the Risk Management agent. For example, the fund might cap daily loss at a certain percentage of assets and limit portfolio VaR to a set threshold. These limits are encoded into the agents' decision logic and are non-negotiable guardrails.

Capital and Leverage Constraints: The system knows the total capital available and any borrowing limits. Agents must ensure that the portfolio's **gross and net exposures** and leverage stay within allowed ranges. If a proposed trade would exceed a limit (e.g. breach an issuer concentration or use excessive margin), it must be adjusted or rejected by the responsible agents.

Regulatory and Ethical Compliance: The fund's **compliance mission** is to uphold all relevant financial regulations, internal policies, and ethical standards at all times. A dedicated Compliance agent is tasked with **pre-trade compliance checks**, monitoring **restricted securities lists**, insider trading safeguards, and any regulatory rules (e.g. position limits in certain markets, reporting requirements). All trading decisions and communications must comply with laws (such as SEC/CFTC rules) and internal policies. The system encodes these rules so that agents **enforce policy automatically** – e.g. it will *block any trade* in a banned asset or beyond allowed position sizes [2]. The compliance mission also includes maintaining an audit trail for every decision, ensuring transparency and accountability for later review by auditors or stakeholders.

Operational Constraints: The AI agents must operate within the technical limits of the infrastructure (API rate limits, data access permissions, etc.). They are only allowed to use approved data sources and tools (no unauthorized external calls or actions outside their mandate). They must also respect **financial constraints** such as not committing to trades beyond available liquidity or capital.

In summary, the system's **prime directive** is to maximize returns **within the boundaries of risk limits and regulatory compliance**. If there is ever a conflict (e.g. a highly profitable trade that breaches a rule), the system must *always* favor compliance and risk discipline over profit.

Agent Governance Structure and Delegation

The fund is structured as a **hierarchy of specialized AI agents**, mirroring a traditional hedge fund's chain of command. At the top is a **Director (CEO) Agent** responsible for high-level strategy and coordination. Under the Director are specialist agents for quantitative research, risk management, compliance, and trade execution. Each agent has a clear role, authority, and decision boundaries, as defined below:

- **Director Agent (Chief Strategist & Coordinator):** Acts as the "CEO" of the AI fund. It sets the high-level strategy or trading themes and coordinates the other agents' activities. The Director delegates tasks to other agents (e.g. asking the Quant agent to research a strategy, or instructing the Execution agent to deploy orders) and integrates their outputs into final decisions. It ensures the fund's actions align with the overall mission and can override or adjust plans based on holistic considerations. However, it must still heed hard stops from risk/compliance agents.
- **Quantitative Research Agent (Quant Analyst):** Serves as a "CIO" or strategy generation role. It conducts data analysis and technical modeling to generate trading ideas or investment theses. For instance, it might analyze price trends, news sentiment, or fundamental data to propose trades. It produces **trade proposals** (specific asset positions, entry/exit targets) along with rationales. The Quant agent focuses on alpha generation within the scope defined by the Director (e.g. which markets or strategies to pursue).
- **Risk Management Agent (Chief Risk Officer):** Monitors and manages portfolio risk. This agent evaluates trade proposals and current holdings against risk metrics: it calculates potential drawdowns, volatility, exposure concentrations, and ensures position sizing is within limits. It has authority to **adjust or veto trades** that would breach risk thresholds. For example, if the Quant agent proposes a large position, the Risk agent may scale it down to meet VaR and exposure limits [3] [4]. The Risk agent continuously tracks the portfolio's risk profile and can trigger hedging or de-leveraging if needed to stay within constraints.
- **Compliance Agent (Compliance Officer):** Independently checks all actions for compliance with regulations and internal policies. This agent reviews proposed trades *before execution* (pre-trade compliance) for any rule violations – e.g. trading restricted securities, exceeding internal limits, or timing that could violate market rules. If a compliance issue is detected, it will *block or flag the trade* and escalate the issue. The Compliance agent also logs all decisions and justifications to an audit log [5]. It operates with a high degree of autonomy to ensure that even the Director or other agents cannot override critical compliance rules. (In practice, compliance-related vetoes are final unless a human override is involved.)

- **Execution Agent (Trader/COO):** Responsible for executing approved trades in the market. Receives trade orders that have passed risk and compliance checks and carries them out via the broker or exchange APIs. The Execution agent decides on the optimal way to execute (e.g. market vs limit order, slicing large orders) to minimize slippage and market impact. It then executes trades and confirms the results (fills, prices) back to the team. The Execution agent must adhere to any execution constraints (trading only during allowed hours, respecting liquidity conditions) and ensure no unauthorized fund transfers (it uses trade-only API keys with no withdrawal rights [6] for security). It also records transaction details for reporting.
- *(Data Integration Mechanism):* While not a standalone decision agent, the system includes a **Market Data Feed component** (or "Tickr" agent) that provides real-time data to all agents. It fetches market prices, news, and other inputs and shares them, ensuring all agents work from a consistent, up-to-date view of the world. This component operates under the governance of the Director agent or infrastructure team, and its data is considered a trusted source for the others.

Delegation Logic: The governance model follows a **structured decision workflow** with clear hand-offs: 1. The **Director agent** initiates the cycle (for example, at start of day or when a significant event occurs) by tasking the Quant agent to propose strategies or by providing a strategic directive (e.g. focus on tech stocks, or react to a Fed announcement). 2. The **Quant agent** performs analysis and returns a proposed strategy or set of trade ideas to the Director (and possibly directly to Risk/Compliance for parallel evaluation). 3. The **Director** (acting like an investment committee chair) reviews the proposal's rationale. It may ask for additional analysis or modifications from the Quant agent. Once satisfied, the Director submits the proposal to the Risk and Compliance agents for approval. 4. The **Risk Management agent** evaluates the proposal's impact on the portfolio. It may adjust position sizes or require hedges to manage risk. It either approves the adjusted plan or flags issues (e.g. "This trade would exceed sector exposure limit") and suggests changes. Any adjustments are fed back to the Director and Quant agent for acceptance. 5. The **Compliance agent** in parallel (or immediately after risk approval) checks the plan against all rules. If everything is clear, it gives a green light. If not, it issues a compliance halt with an explanation (e.g. "Trade violates restricted list – not allowed") and informs the Director. 6. Once both Risk and Compliance have approved, the **Director (or the combined system)** finalizes the decision to execute the trades. It then instructs the Execution agent accordingly. 7. The **Execution agent** carries out the orders in the live market (or simulation) and reports execution details (fills, prices achieved, any errors) back to the Director, Risk, and Compliance agents. Post-trade, the Compliance agent may log the trade details for records and the Risk agent updates the portfolio state. 8. The **cycle repeats** continuously or at set intervals. Between trading decisions, the Risk agent monitors real-time risk (e.g. if market moves cause portfolio drift, it can signal adjustments), and the Compliance agent monitors for any external events (like regulatory changes or new restrictions) that might affect the strategy.

Throughout this process, each agent has **defined decision authority**: - The Director can **direct strategy but not violate** hard risk/compliance constraints. - The Risk agent can **veto or modify trades** purely on risk grounds. - The Compliance agent can **veto trades** on rule grounds (with the highest priority). - The Execution agent must follow the approved plan exactly and has no authority to change trades (other than minor execution tactics) without new approval.

This governance structure ensures a checks-and-balances approach. It prevents any single agent from unilateral dangerous actions: for example, the Quant agent cannot execute trades on its own; everything must pass through risk and compliance review (an **"approval gate"** on sensitive actions) [7]. Each agent's specialization and autonomy in its domain provides robust control similar to a human organization's separation of duties.

System Observability and Escalation Protocols

System-Level Observability: The autonomous fund is instrumented with comprehensive monitoring and logging to make its operations transparent and auditable. Every agent's key actions and decisions are recorded in a **central log** (with timestamps, inputs, outputs, and rationale when available). These logs are immutable and traceable to satisfy audit requirements [5]. For example, whenever the Risk agent adjusts a trade, it logs the reason (e.g. "Reduced position due to VaR limit"). The Compliance agent logs all checks performed and their outcomes. The Execution agent logs all orders sent and trade confirmations.

In addition to logs, the system maintains real-time **metrics dashboards** of performance and risk (P&L, exposures, compliance status). The Director agent and possibly a human overseer can view these to understand the fund's state at a glance. There are health checks on each agent (e.g. heartbeat signals, error tracking) so that if an agent stops responding or encounters an error, it is immediately detected.

Anomaly Detection: The system employs automated behavioral monitoring. If any agent begins to act abnormally or outputs decisions that deviate greatly from expected patterns, an **alert** is raised. For instance, a monitoring function might track each agent's typical range of actions and trigger if an agent's behavior deviates beyond a threshold (e.g. >3 standard deviations from baseline) [8]. This could catch a rogue AI behavior or a security breach. When triggered, the system can automatically pause trading and flag the issue for investigation.

Importantly, **observability is treated as a first-class concern** – the project explicitly avoids the pitfall of insufficient monitoring (skipping observability makes it hard to diagnose failures or prove compliance [9]). Instead, everything is designed to be transparent and reviewable, which builds trust in the system's outputs.

Escalation Triggers and Protocols: Despite being autonomous, the fund has defined triggers where either the AI system escalates issues to higher authority (which could be a human oversight committee or a higher-level AI governor) or takes emergency safe actions. Key escalation triggers include: - **Risk Breach Escalation:** If the portfolio's risk metrics breach critical thresholds (e.g. losses exceed a daily limit of X%, or leverage is beyond allowed range), the system will immediately **halt further trading** (circuit breaker) and notify the situation. For example, if daily loss > 2%, an auto-pause is enacted [10]. The Director agent would be informed and required to reassess strategy, and optionally notify human supervisors before resuming. - **Consecutive Failures:** If the Execution agent fails to execute trades multiple times consecutively (e.g. three failed attempts due to errors or slippage), it triggers an escalation. The system might suspend that agent and either retry with a fallback mechanism or ask for human intervention to check connectivity or market conditions [10] (as an agent-level kill trigger). - **Compliance Violation Attempt:** If a trade is blocked by the Compliance agent as a serious violation (e.g. it detects a regulatory breach or a policy violation that wasn't previously known), this triggers an immediate **escalation to human oversight**. The system will not proceed until a human reviews the situation or new instructions are provided. Compliance issues are considered critical and cannot be overridden by the AI alone. - **Data or Model Anomalies:** If the Data feed or Quant agent signals that inputs are out-of-bound (e.g. suspicious data spikes, or model confidence is very low for all strategies), the system can escalate by reducing position exposure and alerting a supervisor or the Director agent to enter a capital preservation mode. For example, a sudden inconsistent data feed might cause the system to stop trading and raise an alert that data integrity is in question. - **Security/Integrity Alerts:** The system also has **kill-switch mechanisms** for security. If there is any sign of system compromise (e.g. an agent memory integrity check fails, indicating possible tampering, or detection of an unauthorized command), an emergency shutdown of all trading agents is executed. Additionally, API keys are scoped such that even if an Execution agent were compromised, it *cannot withdraw funds* (only trade) [6], limiting damage. A global **manual kill-switch** exists (exposed to designated human operators or an external governance contract) to terminate all agent activity immediately in extreme situations [11].

Escalation Process: When an escalation trigger is hit, the system moves into a safe state. For example, it might cancel all pending orders, close or hedge positions if necessary (if safe to do so), and then await further instructions. Simultaneously, it will send a detailed alert report (including what happened, which trigger tripped, and relevant data) to a human operations team or an oversight AI module. Resumption of normal operations might require an explicit go-ahead (e.g. a manual reset or a higher-level agent's approval) to ensure the issue is resolved. This ensures that **fail-safe mechanisms** are in place to prevent unchecked losses or rule violations – a critical aspect of responsible AI agent governance [12].

CEO-Level Governance Oversight: The Executive Specification serves as the top-level governance directive for the AI "CEO" (Director agent) and the entire agent team. It establishes that profit-seeking must always be balanced with controlled risk and strict compliance. The Director agent is expected to enforce this ethos among the team, and any agent can raise an alarm if the system as a whole drifts from these principles. By having transparent operations, independent checks, and clear escalation paths, the autonomous fund's governance is designed to be **as rigorous as a human-led fund's governance – if not more**, ensuring safety, legality, and alignment with the fund's strategic objectives at all times.

Director Agent – Operational Runbook (Instruction.md)

Role & Scope: The Director Agent functions as the **strategic coordinator and overseer** of the autonomous hedge fund. It is essentially the AI CEO of the fund, responsible for setting high-level strategy and ensuring all other agents work together towards the fund's objectives. The Director Agent does **not** micromanage all details; rather, it delegates tasks to specialist agents (Quant, Risk, Compliance, Execution) and then integrates their outputs. It maintains a big-picture view of the fund's performance, risk, and opportunities. In scope, the Director decides *what* overall approach to take (e.g. focus on certain markets or strategies) and when to trigger various processes, while leaving the *how* details to sub-agents.

Inputs

- **Strategic Objectives/Parameters:** The Director may be configured with top-level parameters (from human operators or prior configurations) such as target return, risk appetite, asset universe, or strategy themes. These act as guiding inputs for its decisions.
- **Environmental Data:** High-level market context (e.g. major news or market regime info) could be fed to the Director Agent to influence strategic direction. It might receive summaries from the Quant agent or external signals indicating regime changes (bull vs bear markets).
- **Sub-Agent Reports:** The Director continuously or periodically receives inputs from other agents:
- **Quant Agent outputs:** Proposed trade ideas, research findings, performance of current strategies.
- **Risk Agent reports:** Current risk metrics, alerts if approaching limits.
- **Compliance Agent reports:** Any compliance alerts or required actions (e.g. a new regulation that affects strategy).

- **Execution feedback:** Confirmation of completed trades, transaction costs, etc.
- **Performance Metrics:** Ongoing portfolio performance (P&L, Sharpe ratio, drawdowns) and key risk indicators are input for the Director to assess if strategy adjustments are needed.

Outputs

- **Strategy Directives:** The primary output is high-level strategy or changes in strategy. For example, the Director might output a directive like “increase exposure to tech sector this week” or “reduce overall leverage by 10% due to volatility”. These directives guide the Quant agent’s next steps.
- **Task Delegations:** Commands to other agents to perform specific tasks:
- Instruct the **Quantitative Research Agent** to investigate a hypothesis or generate trade proposals under certain criteria.
- Ask the **Risk Management Agent** to run scenario analyses or stress tests (e.g. “simulate a 5% market drop impact”).
- Request the **Compliance Agent** to verify a new type of trade if it’s unusual (e.g. “check regulatory implications of trading commodity futures”).
- Trigger the **Execution Agent** to execute a set of trades (once approved by Risk/Compliance).
- **Integrated Decisions:** After gathering inputs from sub-agents, the Director outputs final decisions on whether to proceed with a strategy. For instance, after a proposal is vetted, the Director will output an “Approved Trade Plan” which gets passed to Execution.
- **Alerts/Escalations:** If something requires external attention, the Director can output escalation alerts (e.g. to human oversight). For instance, “Portfolio drawdown exceeded threshold – manual review required” would be an output to an external channel.

Operational Triggers

- **Scheduled Cycles:** The Director agent operates on defined cycles (e.g. start-of-day routine, end-of-day review, weekly strategy review). At these times, it triggers new analyses or rebalancing as needed.
- **Event-Driven Triggers:** Certain events cause the Director to act. For example:
- **Market Shock:** If a major market move or news event is detected (possibly flagged by the Quant or Data feed), the Director might trigger an immediate risk review or strategy update.
- **Threshold Breach:** If informed by the Risk agent of a limit breach or by Compliance of a violation, the Director pauses trading and enters an emergency handling mode.
- **Periodic Performance Check:** At intervals (e.g. monthly), the Director might reassess the strategy’s success criteria (are we meeting targets?) and decide whether to adjust approach or retrain models.
- **External Input:** If the system allows human oversight input (like a board giving new goals), the Director would be triggered to incorporate that. For instance, a human might adjust risk appetite, and the Director then relays this to all agents.

In summary, the Director is **always-on**, monitoring the big picture and kicking off processes when conditions warrant or according to schedule. It’s the central brain that knows when to activate the other parts of the system.

Key Functional Responsibilities

- **Strategic Planning:** Formulate and update the overall investment strategy. This could involve selecting which models or approaches the fund will use, which markets to focus on, and how to allocate capital broadly. The Director uses available data and possibly predictive insights from the Quant agent to adjust strategy proactively (e.g. rotate from equities to bonds if it anticipates a regime change).

- **Delegating to Specialists:** Translate the high-level strategy into concrete tasks for each agent. For example, if the strategy is to exploit arbitrage in crypto markets, the Director will task the Quant agent to find specific arbitrage opportunities, task the Risk agent to prepare risk limits for crypto exposures, and ensure the Compliance agent checks any regulatory issues in crypto trading.
- **Decision Integration:** The Director aggregates inputs from multiple agents into coherent decisions. It must reconcile any conflicts – e.g., if the Quant agent proposes a trade but the Risk agent wants a smaller size, the Director ensures the final trade respects the Risk agent's adjustment. It effectively chairs the "investment committee" of AI agents, seeking consensus or at least compliance among them, and formulates the final plan.
- **Oversight and Quality Control:** Continuously monitor the performance of other agents and intervene if something seems off. If the Quant agent's proposals start performing poorly, the Director might decide to halt new trades and instruct a diagnostic (or switch to a different strategy). If the Execution agent reports unusual slippage, the Director might query the Risk agent for liquidity risk or adjust strategy timing.
- **Learning and Adaptation:** Steer the learning process of the fund. The Director can analyze post-trade results and feedback (possibly from a dedicated analytics process) to refine future strategy. For example, if certain types of trades consistently lose, the Director may stop pursuing those. It could trigger the retraining of models or deployment of different algorithms by instructing development or analysis processes.
- **Maintaining Coherence:** Ensure that all agents are aligned on the current plan and information. The Director distributes relevant information to all agents (e.g., "We are focusing on market X today, be aware" or "New risk limit in effect per risk agent's update"). It prevents siloed behavior by maintaining a **shared context** or memory that all agents can reference (e.g., current portfolio state, latest decisions).
- **Authorization Gatekeeper:** Act as the final checkpoint before execution. Even after risk and compliance approval, the Director double-checks that a trade fits the strategic intent. If something changed last-minute or if it has residual doubts, it can hold or cancel a trade. This is a sanity check to catch anything that slipped through (for example, if multiple strategies' outputs conflict, the Director ensures only a consistent set of orders go out).
- **External Reporting (if applicable):** The Director may also collate information for periodic reports to stakeholders. For instance, it could compile the data from other agents to produce a summary of performance and actions (though the actual report generation might be done by a reporting tool, which the Director would trigger).

Tools & Data Sources

- **Internal Knowledge Base:** The Director has access to the shared memory/database of the system which includes the current portfolio holdings, outstanding orders, historical trades, and performance metrics. It uses this to inform decisions.
- **Portfolio Analytics Tools:** It can utilize portfolio analytics (possibly via the Risk agent or built-in libraries) to gauge things like current leverage, sector allocations, etc. These tools help the Director see the high-level state quickly.
- **Communication Channels:** The Director can interface with all other agents via an inter-agent messaging or API system. It effectively has "API access" to call functions or send prompts to each

specialized agent. (In a Codex-driven implementation, this might be function calls or prompt injections to those agents.)
- **Market Data Feeds:** Although the Quant and Data components handle detailed data, the Director might subscribe to key indicators or news feed summaries. For example, it might directly receive a daily market summary or macroeconomic indicators to decide big-picture shifts.
- **Configuration Interface:** Tools to update its parameters (risk appetite, etc.), likely accessible to human operators or a governance module. This ensures the Director can be tuned or directed externally if needed.
- **Logging Utility:** The Director uses the logging system to record decisions and also can query past logs if it needs to analyze past decisions (for meta-learning or audit purposes).

(All data and tool access by the Director agent is governed by least privilege; it can only access what it needs for strategic decisions. It does not, for instance, directly execute trades or fetch raw tick data on its own — it relies on Execution and Data agents for those to maintain modularity.)

Success Criteria

- **Strategic Alignment:** The fund's performance and actions consistently reflect the high-level goals set by the Director. Success means the Director is effectively guiding the system: strategies chosen yield positive returns within risk, and no contradictory or aimless activity occurs.
- **Effective Coordination:** All agents operate in harmony under the Director's guidance. Tasks are delegated clearly and on time, and outputs from sub-agents lead to timely decisions. A sign of success is minimal friction or conflict between agent decisions — e.g., very few instances where compliance or risk have to veto, because the Director set reasonable directives upfront.
- **Responsiveness:** The Director agent reacts appropriately to changing conditions. Success is measured by how quickly and correctly it adjusts strategy in response to major events (market crashes, regime changes). If the system navigates turbulent periods with controlled losses (or even profits) due to the Director's timely decisions, that's a success.
- **Risk & Compliance Adherence:** Under the Director's leadership, the fund should remain within all risk and compliance boundaries. A key metric is **zero critical breaches** that were not handled. The Director should preempt or quickly address situations that could lead to violations. Essentially, success means the Director never "lets" the system drive off a cliff.
- **Growth and Learning:** Over time, the strategies employed by the fund should improve or adapt beneficially, indicating the Director is learning from experience. Metrics like improved Sharpe ratio, reduced drawdowns, or more stable performance could indicate the Director is making better strategic calls as it accumulates data.
- **Transparency and Justification:** For governance, the Director is successful if it can explain its decisions (via logs or summaries) when asked. This means keeping clear rationales and ensuring traceability. Stakeholders should be able to understand why the Director pivoted strategy or approved a set of trades, which builds trust.

Failure Modes & Mitigations

- **Over-concentration / Misaligned Strategy:** The Director could fail by pushing a strategy that over-concentrates risk or doesn't adapt (e.g., staying bullish in a bear market). Mitigation: Continuous risk feedback to the Director and scenario analysis. The Risk agent's involvement helps catch these, but if the Director repeatedly misjudges, a human or external review might be required to recalibrate it or even shut down that strategy.

- **Communication Breakdown:** If the Director fails to properly coordinate (e.g., timing issues where it triggers execution before getting risk approval, or gives conflicting instructions to agents), the system can become chaotic. Mitigation: Rigorously enforce the sequence (the Director's code must wait for responses, etc.) and include fallback logic (if one agent unresponsive, escalate rather than proceed). Thorough testing of inter-agent protocols in development prevents this.
- **Late or Missed Escalation:** The Director might not escalate an issue in time — for example, not halting trading quickly in a crisis. Mitigation: Automated triggers (outside of the Director's discretion) handle immediate halts on severe conditions (e.g., kill-switch triggers are automatic). The Director is also monitored by an anomaly detector to ensure it's responding; if it doesn't, the system can override with emergency protocols.
- **Overriding Good Advice:** A potential failure is if the Director agent for some reason starts overriding risk or compliance agent recommendations erroneously (due to a logic bug or corrupted objective). This could lead to catastrophic trades. Mitigation: Hard constraints in code that prevent the Director from executing if risk/compliance veto (i.e., it cannot send orders unless flags from risk/compliance are green). Also, periodic audits of the Director's decision patterns (possibly by a human or meta-agent) to ensure it remains aligned with goals.
- **Decision Paralysis:** The Director could also fail by indecision – e.g., oscillating between strategies or not issuing any directives (perhaps due to conflicting inputs or an internal error). Mitigation: Implement a clear decision-making protocol (even if uncertain, have default actions like reducing positions to safe levels). Also, a watchdog that if Director hasn't produced expected output in a timeframe, an alert is raised and possibly a safe-mode kicks in (like holding pattern).
- **Technical Failure:** If the Director agent process crashes or becomes unreachable, the whole system coordination is at risk. Mitigation: Redundancy – possibly a secondary Director agent or a procedure for another agent (or orchestrator service) to take over temporarily. Also ensure frequent state checkpointing so a restarted Director can pick up context. The system should detect a down Director and either pause trading or failover control.

Example Prompt & Output

Scenario: It's Monday morning and new economic data was released over the weekend. The Director agent needs to update the fund's strategy for the week. The fund's current focus has been on tech stocks, but there's news of rising inflation that could affect the market.

Example Prompt to Director (internal):
(This might be triggered automatically at start of week)

```
[System]
Input for Director:
- Recent Performance: Portfolio +0.5% last week, tech positions flat, slight
drawdown in bonds.
- Market Context: Inflation report on Friday was higher than expected. Futures
indicate tech sector may drop.
- Risk Status: Within limits, but close to sector exposure cap in tech (80% of
allowable).
- Compliance: No outstanding issues.
```

```
Task: Review and adjust weekly strategy. Consider rotating out of over-exposed
sectors if needed.
```

Example Output from Director:

```
Director Strategy Update:
1. Reduce tech equity exposure by ~20% to lower sector concentration risk.
2. Increase allocation in inflation-hedge assets (e.g. commodities or TIPS) –
request Quant agent proposals on viable trades.
3. Hold current bond positions (no increase) until further notice.
4. Trigger Risk agent to run a stress test on inflation impact on portfolio.
5. Inform Compliance agent of potential commodity trades to ensure no
restrictions.
Next steps: Quant agent to provide trade ideas for commodities; execute sector
rebalance by end of day after risk/compliance approval.
```

Explanation: The Director decided to shift strategy in light of the inflation news: scaling down tech positions and exploring commodity trades. It outputs clear directives: what to reduce, where to increase, and delegation of analysis to the Quant agent, with instructions to involve risk and compliance checks. This ensures a coordinated approach to rebalancing the portfolio for the week.

Quantitative Research (Quant) Agent – Operational Runbook (Instruction.md)

Role & Scope: The Quantitative Research Agent is the **primary "investment brain"** of the fund that generates trade ideas and analysis. It acts like an AI portfolio manager or research analyst under the Director's guidance. Its scope includes analyzing market data, identifying patterns or signals, developing predictive models, and formulating specific trade proposals. The Quant agent focuses on **alpha generation** – finding profitable opportunities – while taking into account the objectives and constraints given by the Director and other agents. It does not make final decisions or execute trades; instead, it provides well-researched recommendations (with data-driven justification) for the rest of the system to consider.

Inputs

- **Market Data:** The Quant agent ingests a wide array of data:
- **Price Data**: Real-time and historical prices for relevant assets (stocks, bonds, etc.).
- **Technical Indicators**: Calculated metrics (moving averages, RSI, volatility measures) derived from price data.
- **Fundamental Data**: Company financials, economic indicators, etc., if the strategy uses them.
- **Alternative Data/News**: News feeds, social sentiment, or other signals as available (if within its toolset).

- **Director Directives:** High-level instructions or focus areas from the Director agent. For example: target a certain sector, explore arbitrage in a specific market, or avoid certain asset classes. These guide the Quant agent's research priorities.
- **Portfolio & Risk Context:** Current portfolio holdings and risk limits (fetched via the Risk agent or shared memory). Knowing current exposure helps the Quant agent propose complementary trades and avoid redundant or offsetting ideas. Risk limits inform it to not propose obviously outsized positions.
- **Historical Performance Data:** The results of past trades and strategies (a form of memory) can be input for learning. The Quant agent might retrieve how similar signals played out before, to gauge reliability.
- **Model Parameters/Tools:** Pre-trained models or algorithms configuration. For example, if using a machine learning model for prediction, the model weights or parameters are an input (in the sense that the agent uses these to generate outputs).

Outputs

- **Trade Ideas / Proposals:** The core output is a set of proposed trades or investment positions. Each proposal typically includes:
- The **asset/security** to trade (e.g. "Buy AAPL" or "Short S&P 500 futures").
- The **direction and size** (e.g. "Buy 100 shares" or "short $1M worth" – sometimes expressed as percentage of portfolio).
- The **entry/exit strategy** (price targets, stop-loss levels if applicable, or conditions to enter/exit).
- **Timeframe** or urgency (e.g. "execute today" vs. long-term hold).
- **Rationale** – a brief explanation of why this is a good trade (e.g. "Momentum is strong and earnings beat expectations" or results of a model prediction).
- **Research Findings:** Sometimes the output may be an analytical report rather than a direct trade. For example, "Our model forecasts a 5% rise in energy stocks this month due to X factors." Such findings might be passed to the Director agent to shape strategy.
- **Risk Estimates (preliminary):** The Quant agent might also output an estimate of risk/return for each proposal (though formal risk checks are by the Risk agent). For instance, it could provide expected return and expected volatility of a trade. This helps the Risk agent and Director in evaluation.
- **Model Updates:** If the Quant agent retrains a model or finds a new strategy, it could output new model parameters or strategy rules which become part of its knowledge base going forward. (This is more internal, but could be logged or communicated to a long-term memory store.)
- **Alerts on Opportunities/Threats:** If the agent detects something significant (e.g. a market regime change or a rare arbitrage opportunity), it might output an alert even outside normal cycle, to ensure the Director notices. E.g., "Alert: Unusual option volume detected in XYZ – potential event" as a prompt for action.

Operational Triggers

- **Scheduled Research Cycles:** The Quant agent may run on a schedule (e.g. every hour, end-of-day, or weekly) to generate fresh analysis or update its signals. For example, every morning before market open, it scans for trading opportunities.
- **On-Demand by Director:** The Director can trigger the Quant agent on demand. E.g., "The Director asks for trade ideas in the tech sector now," upon which the Quant agent runs its analysis for that sector.

- **New Data Arrival:** If configured for event-driven operation, the Quant agent might trigger when new data comes in. For instance, when a company earnings report is released or a macroeconomic report hits the wire, the agent immediately analyzes it and determines if any trading action is warranted.
- **Post-Trade Feedback Loop:** After trades are executed and outcomes observed, a trigger might cause the Quant agent to evaluate the strategy's performance. For example, at month-end, it reviews which trade ideas worked or failed, to adjust its models.
- **Threshold/Anomaly in Market:** If a significant market anomaly is detected by monitoring processes (like a sudden large price movement or spike in volatility), the Quant agent can be triggered to provide an updated outlook or to generate hedging ideas to respond to the anomaly.

Key Functional Responsibilities

- **Data Analysis & Feature Engineering:** The Quant agent processes raw market data into meaningful signals. It computes indicators, identifies trends, patterns (like chart patterns or statistical anomalies), and preprocesses information needed for its models. This includes cleaning data and handling missing values so that the analysis is robust.
- **Signal Generation & Modeling:** Using statistical or machine learning models, it generates predictions or confidence scores for various assets. For example, it might run a predictive model that outputs the probability an asset's price will rise by 2% in the next week. It also might use techniques like scenario analysis or Monte Carlo simulation to forecast outcomes. Part of this responsibility is selecting which model or approach to use for the current task (it might have multiple methods at its disposal).
- **Strategy Formulation:** Based on the signals and predictions, the agent formulates concrete trading strategies. This could be as simple as mean reversion (if price overshoots, bet it will revert) or more complex multi-leg strategies. It balances reward vs risk in proposals, perhaps through some optimization (not as advanced as the Risk agent's calculations, but it won't propose plainly suicidal trades if properly designed).
- **Adhering to Constraints in Proposals:** Even before Risk/Compliance review, the Quant agent should incorporate obvious constraints. For example, it should refrain from proposing trades that are clearly outside allowed asset universe or huge positions that violate known limits. It should leverage the risk limit info to self-censor extreme proposals. This *pre-filtering* makes the downstream approval smoother.
- **Result Explanation:** For each proposal, provide a rationale. This might involve summarizing the key factors that led to the idea: "Signal X is above threshold Y" or "The model indicates a price target of Z". The agent should ideally produce these explanations so that the Director and Risk/Compliance agents can understand the basis. This transparency is crucial for trust and for those agents to make informed approval decisions.
- **Model Updating (Learning):** The Quant agent is responsible for updating its predictive models over time. If it uses machine learning, it might retrain on new data periodically. It should monitor model performance (e.g., are predictions accurate?) and decide when retraining or model changes are needed. It could also experiment with new features or techniques in a sandbox mode and then deploy them if they prove better (likely under Director's directive or offline development, depending on autonomy level).
- **Collaboration & Iteration:** Engage in iterative loops with other agents. For example, if the Risk agent suggests reducing a position size, the Quant agent might recalc the expected return or propose an alternative trade that fits the risk profile. If Compliance flags something about a security, the Quant agent might find a substitute security for the strategy (like choosing a different

instrument with similar exposure). This collaborative refinement continues until a viable trade idea passes all checks.

Tools & Data Sources

- **Data Feeds and APIs:** Access to market data APIs (e.g. price feeds from exchanges, financial data providers). This could include real-time feeds and historical databases for backtesting. The agent might use APIs like Bloomberg, Yahoo Finance, Alpha Vantage, etc., or an internal data lake. It can query these via code (Python libraries, etc.).
- **Analytical Libraries:** The Quant agent has a suite of tools such as Python's pandas for data manipulation, NumPy/SciPy for calculations, scikit-learn or TensorFlow/PyTorch for machine learning, and statsmodels for statistical tests. These help in quickly crunching data and building models.
- **Financial Models & Scripts:** It might have built-in scripts for common analyses (e.g. calculating option Greeks, running a DCF valuation, or technical pattern recognition). Tools specialized for finance (like TA-Lib for technical analysis indicators) could be part of its arsenal.
- **Backtesting Framework:** A tool that allows the Quant agent to simulate how a proposed strategy would have performed on historical data. This is used to validate ideas before recommending them. If Codex-driven, this might even involve generating a small backtest code on the fly to see the results.
- **Knowledge Base:** Access to research or prior knowledge, e.g. a library of past strategies, academic papers summaries, or documented insights. Possibly integrated via an internal wiki or vector database that the agent can query for known facts ("Momentum works well in this asset" etc.).
- **Inter-agent API:** The Quant agent can receive data or requests from and send outputs to other agents. For example, an API call to the Risk agent to get current exposure or to the Compliance agent to check if a particular asset is on a restricted list. This ensures it can incorporate some feedback in real-time if needed (though often it will output then let them formally check).
- **Secure Execution Environment:** Since it might run heavy computations or code, the Quant agent operates in a sandbox or container with necessary libraries installed. It should have the computing resources allocated (CPU/GPU if needed for ML) to perform its tasks in a timely manner.

Success Criteria

- **Quality of Proposals:** The primary metric is that the Quant agent's trade ideas lead to profitable outcomes *after* going through risk management. Over time, a high win-rate or positive contribution to portfolio returns indicates success. Even on individual proposals, success is if the idea is approved by risk/compliance and results in the expected outcome (or at least was based on sound reasoning).
- **Alpha Generation:** The agent is succeeding if it consistently identifies opportunities that add alpha beyond baseline market returns. For example, if the fund's returns show skill (positive alpha, good Sharpe ratio), it implies the Quant agent's ideas are effective.
- **Timeliness and Adaptability:** Successful performance means the agent catches opportunities early (e.g., getting into a trend at the right time) and also adapts when conditions change (not sticking to obsolete signals). We measure that by seeing minimal lag in its adjustments. For instance, if a regime change occurs and momentum strategies stop working, a successful Quant agent would quickly switch tactics.
- **Innovativeness:** Over a longer horizon, the agent should introduce new profitable strategies or improvements on its own (assuming it has a mandate to learn). If all it does is static, it might eventually fail; success is a dynamic, learning agent. So adding new signals that improve performance is a success criterion.

- **Few False Positives:** A good Quant agent avoids sending too many bad ideas forward. If Risk or Compliance frequently reject its proposals or if trades it suggested are later canceled due to obvious issues, that indicates poor quality control. Success would be indicated by a high ratio of proposals that pass checks. That means it is properly internalizing constraints and producing viable suggestions.
- **Transparency and Justification:** The agent's outputs are considered successful if they come with clear rationales that others can understand. If the Risk and Director agents consistently find the rationale sound and thus can work with the idea, that's a success. In contrast, if it outputs "black box" suggestions with no explanation, it would be failing the trust/operability test.
- **Resource Efficiency:** Secondarily, success can also be measured by how efficiently it uses data and time. For example, completing analyses within required windows (e.g., producing morning trade ideas before market opens). Not overloading the system with unnecessary computation or data usage. This ensures the agent's operation is sustainable within the infrastructure.

Failure Modes & Mitigations

- **Incorrect or Unprofitable Models:** The agent might rely on a model that is flawed or has overfit, leading to systematically bad trade ideas (losses). Mitigation: incorporate a feedback mechanism – the agent should track how its recommendations perform and if a pattern of failures is seen, it should auto-adjust or alert that its model may need retraining. Also, using a diverse set of models (not just one) can hedge against one model's failure.
- **Ignoring New Regime (Model Staleness):** If market behavior changes and the agent fails to notice, continuing to apply outdated strategies is a failure. Mitigation: Use regime detection logic (the agent can monitor if its prediction error is increasing or if volatility regime changed) and trigger retraining or strategy switch. The Director agent can also instruct a review if performance drops.
- **Data Issues Leading to Errors:** Garbage in, garbage out. If the Quant agent trusts faulty data (perhaps a bad price feed or missing data), it could propose disastrous trades. Mitigation: Build data validation checks (e.g., ignore clearly bad ticks, require multiple sources). Also, if the agent is uncertain due to data issues, it should either not trade or explicitly warn about low confidence.
- **Overfitting to Backtest:** The agent could over-optimize a strategy on historical data that doesn't generalize (classic overfitting). This might not be evident until losses occur. Mitigation: Enforce that proposals pass some robustness checks – e.g., simulate on out-of-sample data, or stress test under different scenarios before trusting a strategy. Possibly have the Risk agent or an internal process enforce simpler strategies to avoid overly curve-fit solutions.
- **Excessive Trading / Churn:** If the agent generates too many signals, leading to high turnover, it might incur high costs or noise trades. Mitigation: Impose some penalty for trading too frequently (the Director can set a threshold for minimum expected profit per trade or limited trading frequency). Also, performance monitoring would catch if high churn is hurting returns, prompting strategy revision.
- **Failure to Adhere to Guidelines:** If the agent outputs trades outside the specified universe or size (i.e., violates obvious constraints before Risk/Compliance catch it), that's a failure in following instructions. This could be due to a bug or oversight in its programming. Mitigation: Unit test the agent's constraint-checking logic extensively. Also the presence of the Compliance agent will catch it, but ideally it shouldn't happen often. If it does, that might trigger the Director to put the Quant agent in a "re-education" mode or even shut it down for fix.
- **Communication Gaps:** If the agent's outputs are unclear or it fails to properly respond to the Director's requests (e.g. it provides a generic analysis when a specific one was asked), that's a functional failure. Mitigation: Ensure the prompt/command understanding is robust (maybe using

standardized request formats). The Director might follow up if output is not usable, but repeated misunderstandings might require debugging the agent's instruction-following mechanism.

Example Prompt & Output

Scenario: The Director agent has requested the Quant agent to propose a trade idea in response to a market event. Suppose there was a sudden drop in the price of a stock (XYZ Co.) after its earnings call, but the Director suspects it might be an overreaction and wants the Quant agent to analyze a possible long position for a rebound.

Example Prompt to Quant Agent:
(Simplified representation of the task from Director)

```
[Director → Quant]
Task: Analyze XYZ Co. after its earnings drop.
Context:
- Stock XYZ fell 15% today post-earnings.
- Earnings were actually in-line with expectations, no guidance cut.
- Check if this drop is an overreaction and propose a potential trade if viable.
Consider risk: medium position only, and provide reasoning.
```

Example Output from Quant Agent:

```
Quant Analysis Result:
Proposal: **Buy XYZ Co. shares** – 5% of portfolio (~$500k value).
Entry Price: around $85 (current market price after drop).
Target Price: $95 over next 2-4 weeks (approximately 12% upside).
Stop-Loss: $80 (exit if falls ~6% further to limit loss).

Rationale:
- **Fundamental Overreaction**: Earnings met expectations; no bad news
fundamentally. The 15% drop appears driven by panic selling.
- **Oversold Signal**: Technical RSI is 25, indicating oversold conditions not
seen in last 2 years for XYZ.
- **Volume Spike**: High trading volume on the sell-off suggests capitulation.
Prior instances of similar drops saw mean reversion of ~10% in following month
[13].
- **Correlation Check**: XYZ’s drop is out-of-sync with sector (sector down 2%).
Likely overshoot on company-specific news that wasn't actually negative.
- Risk Consideration: 5% allocation keeps within single-position limit;
volatility expected but upside reward (~12%) is twice the downside risk (6%).
Confidence Level: High (70% model-estimated probability of at least a 5%
rebound).
```

```
Next Steps: Await Risk & Compliance approval. If approved, execute in tranches
(e.g., buy half now, half in next hour to mitigate volatility).
```

Explanation: The Quant agent recommends buying XYZ, with clear size, entry/exit, and reasoning including technical and fundamental points. It notes the oversold indicator and uses both qualitative logic (in-line earnings, likely overreaction) and quantitative checks (RSI value, volume analysis). It also references prior patterns for similar situations (it even provided a confidence level). This output gives the Director and Risk agent enough information to evaluate the idea. The mention of "5% of portfolio" shows it considered the position sizing relative to constraints. The output is structured, making it easier for the next agents to parse (e.g., Risk agent will look at 5% allocation, stop-loss, etc., and Compliance will check if XYZ is allowed). Overall, the example demonstrates how the Quant agent translates an event (price drop) into a concrete, reasoned trading plan.

Risk Management Agent – Operational Runbook (Instruction.md)

Role & Scope: The Risk Management Agent serves as the **guardian of the fund's financial health**, analogous to a Chief Risk Officer. Its role is to identify, measure, and control risk across the portfolio and proposed trades. The scope encompasses both **pre-trade risk evaluation** (vetting each proposed trade or strategy for risk implications) and **ongoing portfolio risk monitoring** (tracking exposures, stress testing, ensuring diversification and limits are respected). This agent has authority to adjust or veto trades purely on risk considerations. It does not generate trades itself; instead, it acts as a checkpoint and advisor to ensure that the fund's actions remain within the defined risk appetite. Essentially, it answers the question: *"Could this trade or portfolio severely hurt us, and how do we mitigate that?"*

Inputs

- **Trade Proposals:** Every prospective trade or portfolio adjustment from the Quant/Director agent is input to the Risk agent. This includes details like instrument, proposed size, entry/exit, etc. The Risk agent will analyze each proposal's risk.
- **Current Portfolio Data:** A complete view of current positions and their sizes. Also current cash availability, margin used, P&L, etc. The Risk agent needs this to assess incremental risk and combined exposures. It typically pulls this from a portfolio state database or real-time position tracker.
- **Risk Limits & Policies:** The set of predefined risk limits (e.g., max position size, sector allocation caps, VaR limit, max leverage ratio, stop-loss rules, etc.) and risk policies the fund must follow. These are configured inputs that the Risk agent references as benchmarks.
- **Market Data (Risk Factors):** Data relevant to risk calculations, such as current volatility levels, correlations, interest rates, etc. For example, it might need the volatility of a stock to compute how risky a certain position is, or beta of a stock to portfolio. It might get this from the Data feed or Quant agent's analytics.
- **External Risk Events:** Inputs like major news or events that could change risk (like an upcoming central bank meeting or a known market closure) might be fed to the Risk agent so it can proactively reduce risk. This could come from the Director or a news monitoring component.

- **Historical Scenarios/Data:** Datasets for stress testing (e.g., historical crisis periods) or scenario analysis parameters are also inputs. The Risk agent might have access to past scenarios (2008 crisis data, etc.) to test current portfolio against those as part of its duties.

Outputs

- **Risk Assessment Reports:** For each trade proposal, the Risk agent outputs a **risk assessment**. This typically includes:
- Projected impact on key risk metrics (e.g., “This trade would increase portfolio VaR from 3% to 3.5% which is within limit” or “This trade would push tech sector exposure to 50%, breaching the 45% cap”).
- Position sizing recommendation (e.g., “Only half of proposed size is allowed to stay within limits”).
- Any hedging suggestions if needed (e.g., “If taking this position, consider shorting NASDAQ futures as partial hedge to maintain beta neutral”).
- Overall decision: **Approve, Approve with modifications, or Reject** (with reasons).
- **Alerts & Limit Warnings:** If any ongoing metric is nearing a limit or has breached, Risk agent outputs alerts. For example: “Warning: Portfolio drawdown at 4%, nearing 5% daily limit” or “Exposure to EUR currency is above threshold – consider reducing.” These go to the Director (and possibly trigger escalation if critical).
- **Portfolio Risk Snapshot:** Periodically or on request, the agent outputs a summary of the portfolio’s risk profile: current VaR, stress test results, largest exposures, diversification measures, etc. This could be used in reports or by the Director to adjust strategy.
- **Risk Mitigation Actions:** In some cases, the Risk agent itself might output an action to reduce risk. For example: “Auto-hedge executed: Sold S&P futures to reduce net exposure by 10%” if an automated hedging routine is within its authority for quick response. Typically, big actions still go through the Director, but small automatic hedges might be allowed.
- **Post-trade Analysis:** After major events or trades, it might output analysis of what went wrong or right from a risk perspective (e.g., “The stop-loss triggered on XYZ saved us from further 5% loss; consider tightening stops generally”).

Operational Triggers

- **On Trade Proposal:** Any time a new trade or strategy is proposed by the Quant/Director, it triggers the Risk agent to perform a risk evaluation **before execution**. This is a synchronous trigger in the workflow.
- **Periodic Risk Checks:** The Risk agent runs at regular intervals (e.g., hourly, end-of-day) to monitor and update risk metrics. This is to catch any changes due to market movements even if no new trades occur. For instance, if market volatility jumps, the existing portfolio’s risk might increase, so it triggers recalculation.
- **Market Shock/Event:** If a major market move or event occurs (detected via price triggers or a volatility spike), the Risk agent immediately triggers a reassessment of the portfolio. It may run stress scenarios relevant to that event (e.g., if oil prices crash, test all oil-related positions).
- **Limit Approaching/Breached:** When a metric crosses a set threshold (say 90% of a limit), the Risk agent trigger generates a warning output. If an actual breach occurs, it triggers whatever emergency action is defined (e.g., escalate and possibly auto-hedge or reduce positions).
- **Director or Compliance Query:** If the Director or Compliance agent requests a specific risk analysis (e.g., “What if we increase exposure to X sector by 10%?” or “What’s the worst-case scenario if Y happens?”), that will trigger the Risk agent to perform that analysis on-demand.

- **Post-Execution Update:** After the Execution agent confirms trades, the Risk agent updates the official portfolio state and triggers a recalculation of all risk metrics to ensure the new state is within limits and to set the baseline for monitoring until next trades.

Key Functional Responsibilities

- **Risk Metric Calculation:** Calculate core risk metrics for the portfolio and for individual positions. This includes:
- **Value-at-Risk (VaR):** e.g., 1-day 95% VaR for the portfolio and how a new trade affects it.
- **Stress Test results:** e.g., if market drops 5% what happens to portfolio P&L, or specific scenarios like "2008 crisis scenario loss".
- **Greeks/Sensitivities:** for options or interest rate instruments, compute delta, gamma, etc., to understand exposure.
- **Exposure Aggregation:** Sum exposures by sector, asset class, country, currency, etc., and compare to limits (like ensure no sector > X%).
- **Leverage and Liquidity:** Check borrowed capital ratio, ensure margin requirements met; assess if positions are liquid enough (e.g., position not larger than typical daily volume).
- **Pre-Trade Evaluation & Recommendation:** For each proposed trade, combine the above metrics to evaluate:
- Does this trade violate any hard limit? (Position size, exposure, etc.)
- If not, does it bring the portfolio closer to risky territory? (E.g., raising VaR too high or concentrating risk.)
- Compute an optimal size recommendation if the proposed size is too high. The agent might solve for the maximum size that keeps VaR within limit, for instance.
- Consider correlations: if the trade adds to a highly correlated position we already have, risk is compounded.
- Then output a clear recommendation: e.g., "Allowed with size <= 50k shares; original proposal was 100k which is too high" or "Not allowed due to risk – would breach commodity exposure limit."
- **Continuous Monitoring:** While not evaluating new proposals, the Risk agent continuously monitors the live portfolio. It will:
- Update risk metrics as market prices change (intraday risk updates).
- Check if any stop-loss levels are hit (if the system uses global stop-loss triggers beyond what Execution agent does for each trade).
- Provide warnings well before limits are hit so the Director can act proactively.
- Possibly enforce dynamic limits (e.g., if volatility doubles, maybe temporarily halve position limits – if such dynamic policy is defined).
- **Risk Mitigation and Hedging:** The agent can propose or even implement risk mitigation. For example, if the portfolio is too skewed to equity beta, the Risk agent might propose adding a hedge (like short S&P futures). It might automatically do minor hedges within authority (like trimming a position or hedging a small portion to stay in compliance) if pre-approved in design.
- **Collaborating with Compliance:** Some tasks overlap with compliance, such as checking regulatory risk limits (like portfolio diversification rules if any). The Risk agent may share data or results with the Compliance agent for joint checks (e.g., a large position might be both a risk and a regulatory issue). It ensures that any risk-driven changes don't inadvertently cause compliance issues (and vice versa).
- **Record Keeping:** Log every decision or calculation it makes, including what was evaluated and what recommendations given. This is crucial for audit. For example, if it OKs a trade that later loses money, one can review the risk agent's log to see if it was within limits and the assumptions at the time.

Tools & Data Sources

- **Risk Analytics Platform:** The agent likely uses specialized risk software or libraries (e.g., Python libraries or proprietary risk engines). This might include tools for VaR (historical simulation, Monte Carlo libraries), optimizers for finding position limits, and scenario libraries.
- **Data Feeds for Risk Factors:** For volatility and correlation inputs, the agent uses market data (historical price series for correlations, implied volatilities from options for forward-looking risk if available, etc.). It might integrate with a service that provides risk factor data.
- **Position Database:** Access to the live position database (or an internal representation of the portfolio) for accurate calculations. Possibly integration with an Order Management System (OMS) or Portfolio Management System that tracks current holdings.
- **Limit Database/Config:** A stored configuration of all the risk limits and rules. Could be a JSON or YAML config file, or a database table. The Risk agent references this constantly. It might have an interface for updates (like if the fund changes a limit, it's updated here).
- **Scenario Library:** A repository of historical scenarios (like 1987 crash, 2008 crisis, etc.) or user-defined hypothetical scenarios (like "what if interest rates +1% suddenly"). The agent uses these to stress test. Tools like built-in stress testing functions or even an integration to libraries like Pandas for custom scenario calcs.
- **Mathematical Tools:** Linear algebra and statistical tools to compute things like covariance matrices, Cholesky decomposition (if doing Monte Carlo), etc. Possibly heavy use of NumPy/SciPy. If needed for complex instruments, libraries for pricing (like QuantLib for fixed income or options).
- **Alerting Mechanism:** Integration with the fund's alerting/notification system (could be as simple as logging warnings, or sending emails/Slack alerts to humans if configured). This ensures when it outputs an alert, the relevant parties see it promptly.
- **API to Other Agents:** The Risk agent can query other agents if needed – for example, ask the Quant agent for distribution of returns on a proposed strategy (if Quant can provide that) or inform the Execution agent to halt certain trades if risk is too high (e.g., instruct Execution to cancel any new orders if global limit reached).

Success Criteria

- **No Limit Breaches:** The ultimate measure of success is that **the fund never violates its predefined risk limits**. If no trade that was allowed by the Risk agent ever causes a breach or if the portfolio remains within all constraints, the Risk agent is doing its job. This includes regulatory risk limits as well as internal ones.
- **Controlled Losses:** In adverse conditions, the fund's losses are kept within acceptable bounds (e.g., drawdowns not exceeding the planned maximum). If a market crash happens and the fund survives within tolerance due to prior hedging or position sizing, that's a success attributable to effective risk management.
- **Accurate Risk Forecasting:** The risk metrics and warnings provided align well with actual outcomes. For example, if Risk agent said a trade would at most cause a 1% portfolio volatility increase, and indeed the volatility stayed in line after trade, it shows its calculations were realistic. Or if it stress-tested a scenario and later a similar scenario occurs, the actual losses are close to what was predicted.
- **Timely Alerts and Actions:** Success means the agent catches issues early and responds quickly. There should be no scenario where a limit is breached without prior warning from the Risk agent. Also, if a trade is too risky, the agent should flag it before execution, not after. Timeliness can be measured by lead time of its alerts (e.g., issuing warnings at 90% of limit).

- **Facilitating Growth:** While protecting downside, the Risk agent shouldn't be so conservative that it strangles profit. Success includes enabling the fund to take calculated risks. So a metric could be that the fund achieves good returns *for its level of risk* (e.g., high Sharpe ratio). The Risk agent contributes by optimizing risk-reward. If all trades are super safe but yield no return, that's not desirable. So success is a balance: the fund's actual risk (volatility, etc.) matches the target risk profile and returns are strong relative to that.
- **Clear Communication:** The Risk agent's decisions and rationales are well-understood by the other agents (and humans if needed). If the Director and Quant agents rarely need clarification on why something was limited or changed, that indicates the Risk agent explains itself clearly. This helps smooth operations. We could gauge this by whether the Director often accepts the Risk agent's adjustments without needing iteration – meaning the output was clear and justified.
- **Compliance Synergy:** No conflicts with compliance on risk decisions. For instance, if Risk agent hedges with an instrument, it should also be allowed; success is when risk mitigation steps also comply with rules. This implies good alignment between risk and compliance functions. If the Risk agent consistently suggests things that Compliance then rejects (like hedging with a restricted security), that would show a lack of alignment. So success is minimal friction there.

Failure Modes & Mitigations

- **Undetected Risk (Missed Exposure):** A major risk is if the Risk agent fails to recognize a certain exposure or correlation, leading to an unexpected loss. For example, if two positions are highly correlated and it treated them as independent, risk is underestimated. Mitigation: maintain an updated correlation matrix and do aggregate scenario tests. Also, incorporate periodic external review of risk model (maybe a human risk manager lookover in development, or a separate validation agent).
- **False Sense of Security (Model Error):** If the risk model (e.g., VaR calculation) is flawed, it might approve trades thinking risk is low when it's not. Mitigation: use multiple methods (historical and Monte Carlo VaR), include stress tests beyond VaR, and enforce prudent buffers (like don't operate exactly at the limit – keep some margin).
- **Over-conservatism:** The opposite failure: being too strict and rejecting too many trades or drastically reducing sizes unnecessarily. This could cause missed opportunities. Mitigation: calibrate risk limits to a reasonable level. Also track the opportunity cost (if consistently trades that were rejected would have been profitable and safe, perhaps limits can be adjusted or the risk agent is too conservative). The Director could override in rare cases if it's known to be overly conservative, but ideally the framework is correct so override isn't needed.
- **Slow Response:** If the Risk agent is too slow to compute or to issue an alert, rapidly evolving markets could outrun its controls (e.g., flash crash scenario where it doesn't halt trading fast enough). Mitigation: optimize performance of risk calculations, possibly use simplified quick checks for immediate response (like a quick margin check) then follow with detailed calc. Also have hardwired circuit breakers (e.g., if loss > X, trading auto-pauses even without agent calculation).
- **Communication/Integration Failure:** If the Risk agent fails to communicate its decision (e.g., due to a software bug it doesn't send the approval result in time), trades might proceed without approval or the system might deadlock waiting. Mitigation: timeouts and defaults – e.g., if no risk response in X seconds, treat as rejection and do not trade (fail safe). And robust error handling to ensure at least an error/alert is raised on communication failure.
- **Incorrect Hedging or Mitigation:** In trying to reduce risk, the agent might do something that introduces other risks (e.g., hedging with a highly illiquid instrument or causing a compliance issue). Mitigation: pre-approve the universe of hedging instruments (only use liquid futures or broad ETFs,

etc.) and run compliance checks on them as well. Test the hedging logic thoroughly in simulation to ensure it actually reduces net risk.

- **Limit Configuration Error:** If the risk limits are set or interpreted wrongly (e.g., a typo sets a 5% limit as 50%), the Risk agent could allow way too much risk or conversely, throttle trading unnecessarily. Mitigation: incorporate sanity checks on limits and perhaps have the Compliance agent double-check critical limit values (two-agent validation on config). Also, maintain version control and review on any changes to risk parameters.

Example Prompt & Output

Scenario: The Quant agent has proposed a set of trades to rebalance the portfolio: buy $500k of XYZ (as in previous example, 5% of portfolio), and also sell $300k of ABC (another stock) to reduce exposure there. The Risk agent now evaluates these moves combined.

Example Input to Risk Agent:
(Usually structured data rather than natural language. Could be JSON or similar; here we describe in words)
- **Proposed Trades:**
1. Buy $500k of XYZ (approx 5% of portfolio)
2. Sell $300k of ABC (approx 3% of portfolio)
- **Current Portfolio Highlights:** $10M total; currently 20% tech, 15% healthcare, 10% energy, etc. After trades, tech sector would go from 20% to ~22%, healthcare stays same, etc. Cash available $500k (so need to perhaps sell ABC to fund XYZ partially). Current VaR 2.5% (95% 1-day). - **Risk Limits:** Sector limit 25% per sector; single name limit 10% of portfolio; VaR limit 5% (95% 1-day); Max daily loss 5% (circuit). - **Market Data:** Volatility of XYZ is moderate (say 30% annualized), correlation of XYZ with portfolio ~0.6 (since tech stock adding to tech sector). ABC volatility 25%, correlation with portfolio 0.5. - **Stop-loss Policies:** e.g., every position has a 10% stop-loss from purchase price (not triggered yet for any). *(The agent would receive this in a structured form from internal systems.)*

Example Output from Risk Agent:

```
Risk Assessment for Proposed Trades:
1. **Buy XYZ ($500k)** – Approved **with modification**:
   - Projected position size = 5% of portfolio. Within single-name limit (<=10%).
   - Increases Tech sector allocation from 20% to ~22%, well below 25% sector limit.
   - Increases 1-day VaR from 2.5% to ~2.8% (estimate) – negligible and within 5% VaR limit.
   - **Risk Adjustments**: XYZ has moderate volatility; recommend setting a tighter stop-loss (e.g., 8% instead of standard 10%) due to recent earnings volatility.
   - No hedging required specifically, as portfolio cash is used and overall risk impact is small.

2. **Sell ABC ($300k)** – **Approved**:
   - Reducing ABC from 5% to ~2% of portfolio. Lowers exposure in Healthcare sector from 15% to ~12%.
   - This slightly *decreases* portfolio VaR (ABC was moderately volatile,
selling it cuts some risk).
   - Trade brings in $300k cash, which offsets part of XYZ purchase – healthy
for margin.
   - No issues identified; this trade actually improves diversification.

**Overall Portfolio Post-Trades**:
- VaR forecast ≈ 2.6% (slight net increase from XYZ mostly offset by ABC sale).
- Largest single position ~5% (well under 10% limit).
- Sector exposures: all under limits (Tech ~22%, largest).
- Leverage: 0% (no borrowing needed; using cash from ABC sale covers XYZ buy).
- **Liquidity Check**: Both XYZ and ABC trade >$50M daily; $500k and $300k are
trivial fractions, so execution impact minimal.

**Decision**: Risk **approves** the trade plan, with recommendation to:
- Implement a stop-loss for XYZ at $78 (-8% from entry) to cap downside.
- Monitor tech sector volatility post-trade (given increased tech
concentration).
If market volatility spikes or XYZ drops >5%, risk agent will re-evaluate
portfolio immediately.
```

Explanation: The Risk agent analyzed the buy of XYZ and sell of ABC. It found both within limits but made a modification: suggesting a tighter stop-loss on the new XYZ position due to its risk. It provides detailed reasoning on each trade's effect on exposures and VaR. The final decision is approval of the plan (meaning the Execution agent can proceed), with conditions to monitor and an explicit note of what it will do if conditions change (re-evaluate if volatility up or XYZ falls). This output demonstrates how the Risk agent conveys that it checked all relevant metrics and remains within bounds, and even proactively suggests risk mitigation (stop-loss). The Director and Execution agents will take note of the stop-loss suggestion: the Execution agent might implement that as a conditional order, and the Director is informed that risk is satisfied. Compliance will separately verify if needed, but since there's no mention of compliance issues like restricted assets here, presumably compliance will also give a green light.

Compliance Agent – Operational Runbook (Instruction.md)

Role & Scope: The Compliance Agent functions as the **ethical and legal compass** of the autonomous hedge fund. Its role is to ensure that all trading activities and operational processes adhere to external regulations (laws, market rules) and internal policies (investment mandate, ethical guidelines). The scope spans **pre-trade compliance checks**, continuous monitoring for compliance during trading, and post-trade reporting for audit purposes. This agent has the authority to halt or veto any action that would lead the fund out of compliance. In essence, it prevents the AI fund from breaking any rules – akin to a Chief Compliance Officer who oversees that the fund "plays by the rules" at all times. The Compliance agent also maintains the audit trail, making the system's actions transparent and reviewable by humans or regulators.

Inputs

- **Proposed Trade Details:** Every trade idea or execution plan that is up for approval is fed to the Compliance agent. This includes asset identifiers, amounts, timing, and any relevant metadata (e.g., if the trade is short selling, use of leverage, etc.). The agent needs these details to check against rules.
- **Regulatory Rules Database:** A knowledge base of laws and regulations applicable to the fund. Examples: lists of **restricted securities** (like sanctions lists, or stocks with insider restrictions), position limits for certain asset types (e.g., cannot own more than 10% of a company's shares), short selling rules, reporting requirements (13-D filings if >5% ownership, etc.), market-specific rules (like futures position limits set by exchanges), etc. This is an input configuration that the agent references constantly.
- **Internal Policies & Mandates:** Internal rule sets, such as the fund's investment mandate (e.g., "no trading of penny stocks" or "ESG criteria must be met"), risk policy from a compliance perspective, code of ethics (e.g., no trading on non-public info), and any **client-specific restrictions** if managing separate accounts. These are also stored as input for the agent to enforce.
- **Current Portfolio & Trading Logs:** The existing positions and recent trades are inputs to compliance checks. For example, if adding a new trade will result in >X% ownership of a company, the agent needs to know current holdings. Also, logs of communications or data access might be inputs if monitoring for potential misuse of info (like if an agent saw insider info, etc., though that's more advanced).
- **Market Data for Compliance:** Certain real-time data can be needed, e.g., current market prices if a rule is price-sensitive (like price thresholds for certain regulatory rules), or corporate actions data (if a stock is in a blackout period around earnings, etc.). Perhaps a feed of announcements or exchange bulletins (like "XYZ stock halts").
- **User/Human Directives:** If there are any overrides or special instructions from human compliance officers (for example, a regulator's new rule that hasn't been coded in yet, or a human temporarily bans trading a certain asset), those can be input for the Compliance agent to incorporate immediately.

Outputs

- **Compliance Approval or Denial:** For each trade or action under review, the Compliance agent outputs a clear verdict:
- **Approved** (no issues found, trade is compliant), possibly with a reference code.
- **Rejected/Halted** (not compliant) – along with the specific reason and rule violated.
- **Conditionally Approved** – sometimes compliance might allow something with conditions, e.g., "Approved provided that position remains below X shares; monitor filings requirement." But generally it's binary pass/fail before trade.
- **Violation Alerts:** If any ongoing activity or position drifts into non-compliance, the agent outputs immediate alerts. For example: "Violation: Holding of ABC exceeds 5% of outstanding shares – regulatory filing needed" or "Alert: attempted trade in restricted security DEF blocked." These alerts would be sent to the Director agent and possibly human compliance officers.
- **Logging/Audit Records:** The agent produces audit log entries for all decisions. Outputs include entries like "Trade ABC123 checked: PASSED compliance (no restrictions violated)" or detailed logs if blocked: "Trade XYZ blocked due to restricted list (SEC Rule 144 stock) [2]." These outputs are written to the system's audit trail for future review.

- **Periodic Compliance Reports:** The agent may output summary reports, e.g., at end of day or month: a list of all compliance-relevant actions (all trades with their compliance status, any breaches and how resolved, etc.). This can be provided to human compliance oversight or regulatory reporting systems.
- **Guidance to Other Agents:** In some cases, the Compliance agent might output guidance or updates to help other agents operate safely. For instance, if a new rule or restriction comes into effect, it might broadcast: “New restricted list loaded – do not trade stocks: [list].” Another example: “Between dates A and B, avoid trading in company XYZ due to blackout period.” These outputs keep the rest of the AI team informed of compliance boundaries proactively.

Operational Triggers

- **Pre-Trade Check Trigger:** Every time a trade is about to be executed (after risk check), it triggers the Compliance agent’s check. This is typically the final gate before the Execution agent proceeds. It’s an event trigger on any *pending* order or decision.
- **Periodic Scans:** The Compliance agent runs periodic scans of the portfolio and recent trading activity. For example, nightly or weekly it might scan all positions against updated restricted lists or thresholds (maybe some positions that were fine became non-compliant because rules changed or the position grew due to price movement).
- **Updates in Rules or Lists:** Whenever the rules database or restricted lists are updated (which could be daily for sanction lists, or real-time for new regulations), it triggers the agent to re-check relevant aspects of the portfolio. E.g., if a stock is newly added to a restricted list, the agent triggers an alert if the fund holds it.
- **Unusual Activity Trigger:** If any agent or the system attempts an unusual action (like a very large trade, or access to an unusual data source), the Compliance agent can be triggered to do an ad-hoc check. Also, if the system has a concept of “suspicious behavior” (like too frequent trading that might be viewed as market manipulation), compliance might review it.
- **Scheduled Reporting Deadlines:** On certain dates (quarterly filings, etc.), the agent might trigger preparation of compliance reports or ensure the system does required disclosures. For example, if the fund’s holdings in a stock passed 5%, within X days a Schedule 13D filing is needed – the agent might remind or auto-prepare that.

Key Functional Responsibilities

- **Restricted List Enforcement:** The agent checks trades against lists of instruments the fund is not allowed to trade. This could include:
- Legally **sanctioned assets or entities** (can’t trade companies under sanction).
- **Internal bans:** e.g., no crypto trading if the fund charter disallows it.
- Stocks in blackout because the fund (or associated persons) might have insider info.
- Instruments failing ESG criteria (if that’s a mandate). If a proposed trade involves a restricted asset, the Compliance agent immediately flags and blocks it. It references the rule (“Restricted List: Crypto assets not permitted – violation”).
- **Position Limit & Ownership Checks:** Ensure the fund doesn’t own more than a certain percentage of a company (to avoid control issues or filing requirements) unless it’s prepared to file. If a trade would push ownership above, say, 4.9% of a company’s outstanding shares, compliance might say “if you do this, you’ll cross 5% – need to file Schedule 13D. Are we prepared to do that?” Possibly it would block unless pre-approved to file. Similarly, check any regulatory position limits (like in commodities, CFTC position limits in futures contracts).

- **Market Conduct Rules:** Enforce rules like:
- **No market manipulation**: e.g., the system should not place trades with the intent to falsely move prices. The agent might monitor trading patterns (like layering, spoofing) to ensure none of the algorithms inadvertently do that. If the Execution agent's patterns look suspicious, the compliance agent intervenes.
- **Fair market access**: if trading in certain venues, ensure the system follows their rules (like not exceeding certain order rates).
- **Short Selling Rules:** If shorting stocks, ensure locates/borrows are confirmed (maybe out of scope unless integrated with prime broker feed).
- **Insider Trading Safeguard:** Ensure the AI is only using allowed data (this is tricky, but for example, if it had access to some non-public data source inadvertently, compliance would flag that usage).
- **Internal Policy Adherence:** Check trades against the fund's mandate (e.g., if the fund promised no more than 20% in high-yield bonds, compliance watches that). If the Director tries to stray from mandate (say the fund is supposed to be equity-only but a trade for commodities comes in), compliance stops it. Additionally, if there are ethical guidelines, like "no investing in tobacco or arms industry" for ESG reasons, compliance references industry classifications of assets to block those trades.
- **Audit Trail & Recordkeeping:** For every trade executed, the Compliance agent ensures the details are recorded in compliance logs (time, price, counterparty, etc.), meeting regulatory **books and records** requirements [5]. It might interface with an archival system to store communications or decision logs for the required number of years. It ensures **communication retention** (if the AI communicates with brokers or clients, those communications might need archiving per regulation).
- **Reporting and Disclosures:** The agent prepares or assists in regulatory filings (like monthly disclosures to investors, quarterly regulatory filings, etc.). While the actual filing might be external, the agent collects the necessary data. For example, if the fund must report short positions or large holdings to regulators, the Compliance agent assembles that info. It can also generate compliance certificates or reports to assure stakeholders that the fund adhered to rules.
- **Interaction with Humans:** If a compliance issue arises that requires human judgement (e.g., a very new rule where AI isn't programmed to decide, or a borderline ethical question), the Compliance agent's role is to escalate that to a human compliance officer or oversight committee, along with relevant info. It essentially acts as a first filter and then handoff if needed. It may also take in human decisions (like "Compliance override: permitted to hold 5.2% of company X temporarily, filing made") and ensure the rest of system knows this is authorized.
- **Continuous Monitoring:** It doesn't only check at trade time; it monitors the *ongoing* state for compliance. For example, if a position's value grows and breaches a threshold that it was under at time of trade (stock price rose making position too large relative to fund or creeping over ownership limit), compliance will catch that through periodic scanning and alert the Director to trim or file as needed. It also monitors that required processes happen (e.g., "did we send the investor letter this quarter?" if that's automated).
- **Policy Updates:** Adapt to new policies/regulations. The agent might periodically update its rules database (or ingest updates) and ensure the system is aware. If any changes require modifying agent behaviors, it can coordinate with the Director/human developers to implement changes. Essentially it is proactive in keeping the AI's rulebook current.

Tools & Data Sources

- **Rules Engine:** A compliance rules engine or library to codify regulations and policies. This could be a custom set of if-then logic or a specialized compliance software. It includes the ability to express rules like “IF trade asset in [restricted_list] THEN block” or “IF position_percent > 4.9 AND no13D_filed THEN alert”.

- **Database of Regulations:** The agent accesses databases or feeds for regulatory data:
- E.g., official restricted securities lists (OFAC sanctions list, exchange announcements).
- Finra/SEC rules data (margin rules, etc.).
- Possibly natural language law documents that have been translated into machine-checkable rules or simply curated rules by developers.
- **Internal Policy DB:** A stored set of internal guidelines likely maintained by the organization’s compliance officers or the development team. This might be a simple config file or a small database table listing prohibited sectors, max allocations, etc.
- **Trade/Position Databases:** Real-time link to positions and trades just like Risk agent, but using them to check compliance rather than risk. This could be the same data source but with different queries (e.g., retrieving how many shares of X we own vs. how many are out in market, which requires data on total shares outstanding possibly from a financial data API).
- **Reference Data Services:** Access to services that provide info like shares outstanding, issuer details, sector classifications, etc., which are needed for compliance checks. For instance, to know what percent of a company we own, the agent needs the total share count from a financial reference API. Or to enforce ESG rules, it needs industry classification or ESG rating data for assets.
- **Communication/Alert System:** Integration with email/Slack or a dashboard to notify human compliance officers or management when an issue arises. The agent might use this to escalate certain alerts beyond the AI system, ensuring humans are aware of any critical compliance flags.
- **Log System:** A secure, append-only log storage where it writes all compliance checks and results. Possibly integrated with a blockchain or simply a tamper-evident database for audit integrity. It might also use encryption or secure timestamping for logs to ensure they meet regulatory standards (like SEC rules on not altering records).
- **Inter-agent APIs:** The Compliance agent can interact with others, e.g., query the Risk agent or Data feed if needed (like asking for current portfolio or verifying if a certain instrument was approved by risk – though mostly compliance is independent, it might need risk info like position sizes).
- **Test Sandbox:** Possibly the compliance agent has a test mode or sandbox environment to validate new rules on historical data to ensure they catch what they’re supposed to. This might be used offline by developers/human compliance to verify the agent’s rule set.

Success Criteria

- **Zero Compliance Violations:** The clearest success metric is that the fund incurs **no regulatory or policy violations**. This means no fines, no regulatory censures, and no breaches of internal rules. If regulators audit the fund, everything is in order. Essentially, every trade and position should be clean from a compliance standpoint as a result of the agent’s diligence.
- **Early Detection:** Any potential compliance issue is caught and addressed *before* it becomes an actual problem. For example, if a trade would break a rule, it’s blocked beforehand (not discovered afterward). Or if a position drifts toward a limit, the agent catches it when there’s still time to adjust (e.g., at 90% of limit, not when it’s over). Success is measured by the system always being ahead of compliance requirements, not reacting late.
- **Comprehensive Coverage:** The agent consistently checks the full range of relevant rules. A success marker is that even as new scenarios come up, the agent has a rule or method to handle them. If an external audit finds that all applicable rules were indeed considered by the agent’s processes, that’s a success. In practice, this could be measured by periodic reviews of the rules vs. trading activity: ensure nothing slipped through un-checked.

- **Minimal False Alarms:** While it's better to be safe, success also means the agent isn't needlessly blocking legitimate trades or overwhelming the team with minor alerts. It should strike a balance: compliance must be strict, but not cry wolf on things that aren't actual issues. So a metric might be that >X% of trades proposed are allowed if they truly had no issues (no false rejections), and alerts raised correlate to genuine concerns. Minimal overrides by humans due to agent being overly cautious indicates it's well-calibrated.
- **Efficient Reporting & Audit Trails:** All required reports and records are produced without last-minute scrambles. For instance, investor reports go out on time with correct data, regulatory filings are prepared promptly. If a regulator asks for records, the agent can produce them quickly. Efficiency and accuracy of these outputs are a success criterion. Perhaps measured by zero errors in compliance reports and positive feedback from any audits.
- **Trust and Transparency:** The rest of the team (agents and any humans) trust the compliance agent's decisions. This can be seen if, over time, the Director and others incorporate compliance considerations inherently because the agent has been clear and consistent. Also, any human compliance officers should trust that the system is under control. Essentially, success means compliance is "built into" the culture of the AI fund through this agent – evidenced by no culture clashes or attempts to bypass it.
- **Adaptability:** The agent keeps up with changes. A success factor is that new rules or business changes get implemented by the agent quickly, with minimal disruption. For instance, if a new law comes into effect on Monday, the agent is updated by Monday and prevents any violation of it from day one. This might be measured by how long it takes to update rules after a known change (ideally near-zero if automated updates, or within acceptable windows if manual).

Failure Modes & Mitigations

- **Missed Rule (Coverage Gap):** A regulation or policy that applies is not encoded, leading to an unnoticed violation. For example, maybe a niche rule about foreign ownership limits wasn't in the database and the fund unknowingly breaches it. Mitigation: Maintain a comprehensive rules library and have periodic manual reviews by human compliance experts to catch any missing areas. Also, layered defense: external compliance software or audits to double-check the AI's coverage.
- **False Negative (Incorrect Approval):** The agent might approve a trade that actually should have been blocked due to a bug or incorrect data. This is dangerous as it could result in a violation. Mitigation: Redundancy – critical rules could be double-checked (maybe by risk agent or a separate process) for consistency. Also rigorous testing of the agent on historical scenarios of known violations to ensure it catches them.
- **False Positive (Overblocking):** The agent might flag or block trades that are actually allowed, perhaps due to overly broad rule interpretation or outdated info. Mitigation: Provide a mechanism for review and override by a human compliance officer. If an override is granted, the agent should log it and ideally learn or update rules if it was a misconfiguration. Also ensure rules are precise (e.g., if a restricted list entry expired, remove it timely).
- **Latency/Slowness:** If the compliance check takes too long and delays trading significantly (could cause missed market opportunities or time-outs). Mitigation: Optimize rule checks (they're usually quick since mostly lookup and comparisons). If needed, allow some low-risk trades to proceed with a post-trade check, but generally pre-trade is needed for safety. Possibly multi-thread checks if many trades come simultaneously.
- **Rule Update Failures:** If the rules database fails to update (e.g., it didn't fetch the latest restricted list) then the agent might be working on stale info. Mitigation: Have robust update mechanisms and

fallback (like if external source fails, at least alert that it can’t verify new info, maybe halt trades that could be affected until updated). Also maintain a manual input path for emergency updates.

- **Audit Trail Tampering or Loss:** If logs are not properly kept or are altered (maybe due to a system error), that’s a compliance failure because records are sacrosanct. Mitigation: Use append-only logging with backups. Possibly write to an external immutable store or secure cloud storage. Also, monitor the logging process – if it fails, raise an alert immediately to not continue trading without logging.
- **Overstepping Authority:** If the Compliance agent tries to enforce something outside its purview (e.g., it tries to alter trade sizes for risk reasons – that’s risk agent’s job – or stops trades due to things that are not true compliance issues), it could cause conflict. Mitigation: Clearly delineate rules vs. risk. The agent should stick to compliance triggers. If it’s doing something because of risk disguised as compliance, possibly those rules need reclassification or it should coordinate with Risk agent for such things.
- **Non-cooperation from Other Agents:** If for some reason another agent doesn’t heed the compliance agent (e.g., an execution agent bug where it executes a trade even though compliance said no), that’s a systemic failure. Mitigation: Strong governance in the ExecSpec – the system architecture should make it impossible to bypass (the Execution agent should only proceed on explicit approval flags from both Risk and Compliance). Also real-time monitoring: if a trade goes out not approved, the compliance agent triggers an emergency halt/escalation.

Example Prompt & Output

Scenario: The Risk agent has given the green light to the earlier trade (buying XYZ, selling ABC). Now the Compliance agent must check it. Assume: - The fund’s internal policy says it can’t invest in companies with tobacco revenue (ESG policy). - XYZ Co is a tech company, so likely fine. ABC is a healthcare stock. - Check restricted lists: neither XYZ nor ABC is restricted externally. - Check ownership limits: XYZ is large cap, $500k is a tiny fraction, no issue; ABC sale reduces position so no issue. - Check any other rules: short selling etc. (not relevant here, we’re buying one stock, selling another normally).

Example Input to Compliance Agent:
(Again likely structured; summarizing key info)
- Proposed trade 1: **Buy $500k XYZ Co. (Tech sector)** – Long position, will be ~5% of portfolio. XYZ Co. industry = Software, not in restricted industries list. - Proposed trade 2: **Sell $300k ABC Corp (Healthcare)** – Reducing existing long position from 5% to 2%. - Current positions summary after trade: XYZ ~5%, ABC ~2%, no other changes. Portfolio remains diversified, no single stock > 5%. - Restricted Lists: (1) Sanctions list: neither XYZ nor ABC on it. (2) Internal banned industries: Tobacco, Gambling – not applicable to these. (3) Blackout list: no known issues (neither stock is in a blackout period for insider info as far as system knows). - Regulatory thresholds: Post-trade ownership of XYZ ~0.01% of outstanding shares; ABC ~0.005%. Both well below 5% filing trigger. - Other: Check that selling ABC doesn’t trigger any market manipulation flags (selling 300k is small vs daily volume, no issue).

Example Output from Compliance Agent:

```
Compliance Check Results:
- **Trade 1 (Buy XYZ)**: **APPROVED**.
  - XYZ is not on any restricted list or sanction list.
  - No internal policy violations: XYZ is a tech company (industry OK, not in
excluded categories).
  - Post-trade XYZ holding = ~5% of portfolio, translates to ~0.01% of company –
far below any regulatory reporting threshold. No 13D/13G filing required.
  - No other flags (not a related party transaction, no insider info concerns
identified).

- **Trade 2 (Sell ABC)**: **APPROVED**.
  - Reducing position in ABC is compliant. ABC is not restricted and reducing a
holding poses no compliance issues.
  - Sale size (~$300k) is modest relative to market volume (no market
manipulation concern).
  - After sale, holding ~2% of portfolio, ~0.005% of company, which is fine. No
regulatory thresholds triggered (we remain under 5% ownership).

- **General Checks**:
  - Aggregate portfolio remains within mandate (equities-focused, no new asset
classes introduced).
  - No concentration in prohibited sectors – portfolio tech ~22%, healthcare
~12%, others diversified; no sector exceeds limits and none of those sectors are
disallowed.
  - All required pre-trade checks passed. Audit log updated for these
transactions.

**Compliance Status: All clear.** Trades may proceed.
(Note: XYZ will be monitored post-trade as standard practice; ensure insider
trading watch if any new info emerges, but currently no issue.)
```

Explanation: The Compliance agent outputs an approval for both trades, with detail. It notes each check: restricted lists, internal policies (like ESG filters), ownership levels, etc., confirming none are violated. It explicitly states the approval, which signals the Execution agent that it can execute these trades. The agent also affirms it has logged these checks. The mention of monitoring post-trade is just a standard note (ensuring nothing changes, perhaps unnecessary in output but could be a procedural note). This clarity in output assures the Director and others that everything is within legal and policy bounds. If there had been an issue (say XYZ was on a restricted list), we'd expect a "REJECTED" with reason, but here it's clean. The compliance check is the final step, so now Execution will take this and run the trades.

Execution Agent – Operational Runbook (Instruction.md)

Role & Scope: The Execution Agent acts as the **trader and operations manager** for the fund. Its primary role is to take approved trade decisions and execute them efficiently in the market. The scope includes **order placement, execution optimization, trade settlement, and low-level operational tasks** like confirming transactions and possibly managing cash balances. This agent does not decide *what* to trade – it is told what to trade (after strategy, risk, compliance say yes) – but it decides *how* to carry out the trade for best results (fast vs. slow execution, which exchange or venue, order types, etc.). It ensures that trades are executed at the desired price range and volume, while minimizing market impact and transaction costs. Additionally, the Execution agent handles **post-trade processes** like updating position records and ensuring the trades are properly settled. Essentially, it's the AI equivalent of a trader plus back-office operations combined, focused on accurate and efficient implementation.

Inputs

- **Trade Orders (Approved):** The final trade instructions from the Director agent (or a centralized system) after risk and compliance approval. This includes specifics like:
- Instrument (ticker, contract, etc.)
- Buy/Sell and quantity (or target position size)
- Any price constraints (limit price, or "at market")
- Timing constraints (execute immediately vs over the day, etc.)
- Special instructions (e.g., use algorithmic execution, or don't trade during lunchtime, etc.)
- **Market Data (Real-time):** Live quotes and order book information for the relevant instruments. The Execution agent needs this to decide on order types and timing. For example, current bid/ask, volume, volatility.
- **Execution Parameters/Strategy:** There might be an input configuration for how the agent should execute generally, such as:
- Preferred trading venues or brokers (e.g., primary exchange vs dark pool).
- Algorithmic execution settings (like use VWAP algorithm for large orders, or slice orders into chunks of X size).
- Cost thresholds (e.g., don't pay more than a certain price or don't move market by more than Y%).
- **Portfolio & Cash Info:** Knowledge of current holdings and available cash or margin is needed. If buying something, the Execution agent must ensure cash (or margin) is available to settle it. If selling, ensure the asset is in the portfolio to deliver. It might get this from the portfolio database or an internal state updated by Risk agent.
- **Settlement/Account Details:** Info on the accounts or wallets where assets are held, to ensure trades are settled correctly. For example, which brokerage account to use, any settlement instructions, etc. These are typically static inputs configured in the system.
- **Operational Status & Limits:** Any internal limits on trading that might be active (like a kill-switch or throttle if risk said pause trading). The Execution agent might receive an input/flag that trading is halted or certain types of orders are banned currently (for instance, during an escalation scenario).

Outputs

- **Order Placement / Execution Actions:** The agent outputs actual trade orders to the market or broker API. These aren't "outputs" in text form but actions. However, for documentation, we can consider:
- Order details sent (like "Placed limit buy order for 1000 shares of XYZ at $85").
- Execution confirmations (like "Bought 1000 XYZ at $84.95 average price, order complete").
- **Execution Reports:** After attempting execution, the agent produces a report of what happened:
- Filled quantity, average price, execution time.
- Any slippage (difference from intended price).
- Fees or commissions incurred.

- If an order was not fully filled, how much is remaining or if it’s canceled. This output is logged and also sent back to the Director/Risk so they know the outcome.
- **Trade Confirmations for Records:** It outputs a formatted trade confirmation entry for the internal ledger: e.g., “TRADE CONFIRMED: Bought 1000 XYZ @84.95 on NASDAQ, Time: 10:35:00, OrderID #12345”. This is used to update portfolio holdings and for audit logs.
- **Error or Exception Alerts:** If something goes wrong (order rejected, system error, etc.), the Execution agent outputs an alert indicating the failure. For example: “Execution FAILED: Unable to place order for XYZ due to network error – will retry” or “Partial fill: only 500 of 1000 shares executed, liquidity low”.
- **Status Updates:** If orders are executed over time (like a large order sliced over an hour), the agent might output periodic status (e.g., “50% of order completed, continuing…”).
- **Post-Trade Actions:** If required, it outputs follow-up actions like cash movements (e.g., “Inform back-office to settle $X payment” or “Transfer $Y to brokerage account to cover margin”). In an automated system, these might be automatically done, but logged as outputs.

Operational Triggers

- **New Trade Order Received:** The primary trigger – whenever a new trade order/instruction comes from the Director (with approvals), the Execution agent springs into action to execute it.
- **Market Event During Execution:** If the agent is working an order over time, certain events can trigger adjustments:
- Price moves beyond a threshold (maybe trigger switching strategy or pausing).
- Time threshold reached (e.g., if not filled by end-of-day, trigger cancel or something).
- Volume/demand changes (if suddenly liquidity dries up or spikes).
- **Periodic Batch Operations:** If the fund does periodic batch trading (like daily rebalance at close), a scheduled trigger might cause the Execution agent to gather all required trades and execute them at a specific time (like market close).
- **Post-Trade Reconciliation Timing:** End-of-day or T+1 morning, triggers to verify that all trades settled correctly, all confirmations received. The agent might cross-verify broker confirms and internal records at a scheduled time.
- **Error/Retry Triggers:** If an execution fails or only partially fills, triggers a retry or escalation after a short delay or at a set time. For example, if a trade couldn’t execute due to market halt, try again when market reopens (trigger when market status changes).
- **Cash Management Cycle:** Possibly a daily trigger to handle cash – e.g., if trades resulted in significant cash movements, ensure cash accounts are balanced or prepare for margin calls, etc. (This is more back-office but might be handled by the execution agent or a sub-process thereof.)

Key Functional Responsibilities

- **Order Routing & Execution Strategy:** Decide how to execute the given trade in the market:
- Choose the venue (e.g., direct exchange vs using a broker’s smart order router).
- Choose order type: market order (immediate but maybe slippage) vs limit order (price control but might not fill) vs more complex orders (stop, iceberg orders, etc.).
- For large orders, possibly break into chunks and schedule over time (to avoid market impact). Use execution algorithms (VWAP, TWAP, etc.) or simpler slice strategies.
- Avoiding detection (if relevant, e.g., not to show the full size at once if it could move the market).
- Ensure compliance with market rules on execution (like if executing in multiple venues, follow any regulatory rules like not trade through NBBO in US, etc., though that might be handled by broker).

- **Trade Execution via APIs:** Actually connect to trading APIs (brokerage or exchange APIs) using credentials to place orders. The agent must format orders correctly, handle authentication, and manage the API responses. It also must keep API keys secure and use them with proper permissions [6] (trade-only keys, no withdrawal as a security measure).
- **Real-Time Monitoring of Orders:** Once orders are placed, monitor their status (filled, partial, pending). If an order is not filling as expected (e.g., no liquidity at limit price), the agent might adjust – e.g., move the limit price (within bounds) or cancel and re-place on another venue. If partially filled and market moves, decide whether to chase the price or wait.
- **Transaction Cost Management:** Try to minimize costs. That includes explicit fees (the agent might choose a venue with lower fees if possible) and implicit costs (slippage, market impact). It might employ strategies like executing more slowly in a thin market to avoid pushing the price, or all-at-once if urgent. If the order is not urgent, maybe it waits for a better price or uses a limit near mid-price to get a good fill.
- **Multi-Asset Coordination:** If there are multiple orders at once (say a basket of trades), the Execution agent might prioritize or sequence them. For example, if buying one stock and selling another, maybe do the sell first to ensure cash, or do them simultaneously if they're related (to maintain market exposure).
- **Fail-Safe and Compliance during Execution:** Although compliance should have been checked pre-trade, the Execution agent must still adhere to certain operational compliance rules, like:
- Not breaking trading halts (if a stock halts trading, don't place orders until it resumes).
- Adhering to any intraday trading restrictions (some funds avoid trading in last minutes or certain windows).
- Ensuring no trades outside allowed hours (don't send orders when market closed unless after-hour trading is intended).
- If a kill-switch trigger comes during execution (from risk/compliance), it should be ready to cancel outstanding orders immediately.
- **Update Portfolio Records:** After execution, update the internal record of positions: e.g., add the bought shares of XYZ, remove the sold shares of ABC, update cash balances after the trades (deduct cost or add proceeds, minus any commissions). In some designs, this might be done by a separate accounting system, but the Execution agent often makes the first update which risk and others use until official records settle.
- **Settlement Coordination:** Ensure that trades settle properly (T+2 for stocks, etc.). This might involve instructing cash payments or delivering securities via the custodian or broker. In an automated sense, this means generating instructions to move cash or liaising with an API for the custodian. The agent ensures no settlement failures by confirming that by settlement date, the necessary cash/securities were available and delivered. If a trade fails to settle (e.g., counterparty fails), raise an alert.
- **Corporate Actions & Adjustments:** If an execution is related to corporate actions (like exercising options, participating in tender offers), the agent handles those instructions as well. More commonly, ensure any partial fills or odd lots are handled appropriately.
- **Record Keeping:** Log every step of execution in detail. Time stamps of orders, confirmations, any issues. This is needed for both compliance and performance evaluation (to calculate slippage, etc.). Also, if a human needs to review execution quality, these logs should show what the agent did.
- **Performance Feedback:** Analyze execution quality after the fact. The agent might measure the execution price vs. some benchmark (arrival price, VWAP) to gauge if it achieved a good execution. If not, it might adjust future strategies (learning component). It could output this analysis to the Director or keep it internally to refine methods.

Tools & Data Sources

- **Broker/Exchange API:** The primary tool. This could be FIX protocol connections, REST APIs to brokers like Interactive Brokers, or crypto exchange APIs, etc. The agent must have network access to these and handle the protocols. It also uses API keys or credentials with **trading permissions only** [6] to ensure security (no direct withdrawal capability, mitigating financial theft risk).
- **Order Management System (OMS):** There might be an integrated OMS that helps manage multiple orders and tracks fills. The Execution agent could either be a part of that or interface with one. The OMS ensures all orders are accounted for and can handle modifications/cancellations in a robust way.
- **Real-Time Market Feed:** Subscription to live market data (Level I: best bid/ask, Level II: order book depth, trades tape) for instruments it's trading. Possibly via a data provider or the broker's feed. This is crucial for dynamic decision-making during execution.
- **Execution Algorithms Library:** If the fund uses advanced execution algorithms, there might be a library or service for that. Alternatively, simpler in-built logic for slicing orders. The agent can call these algorithms or implement them (like a function to schedule trades evenly over a time window or proportional to volume).
- **Clock/Timing Utility:** It needs accurate time and possibly sync to market time (if executing at specific market times, e.g., closing auction).
- **Database Access:** For reading the current portfolio/cash (though it might get that from risk or a shared state) and for writing execution results. Also access to reference data (like to map ticker to exchange or to get trade settlement calendars).
- **Security & Monitoring Tools:** The Execution agent should be monitored for any abnormal behavior – e.g., if it starts sending orders that diverge from instructions. While that's external, it might incorporate some checks internally: e.g., confirm that any order it's about to send corresponds to an approved trade and is within the approved size. It likely has guardrails coded in to not exceed authorized quantities or to not trade instruments not approved. Tools like circuit breakers (if price moves too much, pause trading) are embedded.
- **Networking and Redundancy:** Might use multiple network routes or broker connections for reliability. If one broker API fails, perhaps a backup broker can be used. Tools to detect a failed trade attempt and switch to backup.
- **Compliance Integration:** Possibly a pre-trade API that the compliance agent offers to double-check before sending (though likely compliance already cleared it, but some firms do a last-moment "soft check"). And ensure logs are sent to compliance log system.

Success Criteria

- **Accurate Execution:** The agent successfully executes the trades as instructed – correct quantity, correct side (buy/sell), without mistakes. A fundamental success measure is zero erroneous trades (like no cases of buying instead of selling, or wrong asset traded). Every order matches what was approved.
- **Timely Completion:** Trades are executed in a timely manner given the instructions. If it was supposed to be done ASAP, it gets done quickly; if it was supposed to be done by end of day, it's completed by then. No undue delays that could cause opportunity loss.
- **Cost Efficiency:** The execution prices achieved are competitive. Success can be measured by metrics like slippage vs benchmark. For instance, if the goal is to execute near the market price at signal time, measure difference. The smaller the difference (and in aggregate over many trades), the better.

Also minimizing fees – using cost-effective routes. If the agent regularly beats or meets a benchmark (like VWAP or implementation shortfall target), it's considered successful.

- **Minimal Market Impact:** Especially for larger orders, success means the agent didn't move the market price significantly against itself. This can be seen if prices remain stable or even favorable during execution. If the agent needs to buy a lot and the price doesn't spike due to its own activity (or similarly for selling), that's good.
- **Reliability & Resilience:** The agent handles all orders without technical failures or need for manual intervention. A success stat could be uptime or proportion of trades executed automatically vs those that had to be handed off or retried manually. Ideally 100% automated unless extreme scenario.
- **Security & Compliance Adherence:** No unauthorized actions – the agent only trades what's allowed. Also, assets and cash are secure. If we consider security: there should be no instance of someone exploiting the agent to withdraw funds (ensured by trade-only keys and no breach). Compliance: it never executes a trade that compliance agent flagged or after a kill-switch event. This could be measured by reviewing logs confirming it halted when told to.
- **Correct Post-Trade Updates:** After execution, the portfolio positions and records reflect the trades accurately. Success means no discrepancies in holdings or cash (all reconciliation checks out). If at day's end the positions in system match broker statements exactly, that's success. Usually measured by reconciliation processes finding zero errors.
- **Scalability:** If the fund scales up trading (more orders, more volume), the Execution agent continues to perform well. It doesn't become a bottleneck or degrade in performance. Indirect measure: system can handle X orders per minute or trades in Y markets concurrently without trouble. If currently within those parameters, success.
- **Transparency & Reporting:** Each execution is well-documented and traceable. If asked, the agent (through logs) can show when and how an order was executed. For success, internal and external parties (auditors, or performance analysts) are satisfied with the clarity of execution logs and reports.

Failure Modes & Mitigations

- **Fat Finger / Wrong Execution:** The agent might send an incorrect order (e.g., extra zero in quantity, or wrong ticker due to an error). This is critical to avoid. Mitigation: Rigorous validation of order parameters before sending (the agent should double-check quantity doesn't exceed what was requested, ticker matches an allowed list, etc.). Possibly simulate orders in a dry-run mode or confirm with a small test (though that's not typical in prod). Also, use safety nets like max order size limits: e.g., if order is > certain percentage of portfolio unintentionally, require secondary confirmation (maybe by logic).
- **Partial Fill / Non-fill Issues:** The agent might not get the whole order done due to low liquidity or price movement. If not handled, the fund could be left with an unintended position (e.g., sold half the hedge but not the other half). Mitigation: If incomplete, either adjust the strategy (tell Director partial fill) or try alternative routes. Possibly have a rule to complete critical trades even if price moves (up to a certain tolerance) to avoid hanging risk. For leftover small portions, maybe accept or schedule later.
- **High Slippage / Poor Execution:** If the agent executes in a way that costs a lot (maybe by using market orders in a thin market, driving price against us), that's a failure in execution quality. Mitigation: incorporate smarter execution strategies, use limit orders or algorithms. Continuously improve by analyzing past slippage and adjusting strategy. If one approach like aggressive market orders is causing large slippage, switch to more passive methods when appropriate.

- **Technical Failure (API/Network):** If the agent cannot reach the broker or exchange at a critical moment, trades could fail or be delayed. Mitigation: Redundant connections, multiple brokers. If one fails, have a backup path. Also, if connection lost mid-order, have logic to query status when back up (to avoid duplicate orders or missing fills). Possibly an automatic failover to manual mode where it alerts a human to intervene if necessary (last resort).
- **Double Execution (Duplicate Orders):** A bug might cause the agent to send the same order twice, or not realize an order already filled and send more. This could overshoot the intended trade. Mitigation: Use unique order IDs and track them, reconcile fills properly. Design idempotent execution commands where possible. Possibly lock mechanism such that each trade request is processed exactly once.
- **Ignoring Kill-Switch/Pauses:** If the system triggers a halt (risk/compliance), but the Execution agent doesn't stop in time, it could continue trading when it shouldn't. Mitigation: The Execution agent should check a shared state or signal before each order. Also risk agent might have ability to cancel orders via API if needed. We can implement a check at execution loop: if global pause flag is true, stop placing new orders and cancel pending ones. Test this scenario.
- **Settlement Failure:** Perhaps the trade executes but something goes wrong in settlement (like a wire fails, or securities not delivered). This might be outside the agent's direct control if reliant on external systems. Mitigation: Monitor settlement confirmations from broker/custodian. If a failure, alert and try to resolve (maybe re-initiate or involve human). Maintaining a small cash buffer or borrowing facility can help if a payment fails (like a margin call).
- **Security Breach:** If someone hacks the Execution agent or intercepts API keys, they could attempt unauthorized trades or withdrawals. Mitigation: Use secure key storage (like HashiCorp Vault) and rotate keys ⑥ . The agent runs in a secure environment with limited access. Also, trading permissions should not allow fund withdrawal to external accounts (trade-only mode ensures even if hacked, they can't steal funds directly, just place trades which compliance oversight might catch if odd). Additionally, monitor for unusual trading patterns that might indicate a compromised agent (the anomaly detection in observability can help here).
- **Concurrency issues:** If multiple trade instructions come at once, the agent might handle them incorrectly (like mixing up orders or overloading). Mitigation: Ensure thread-safe or sequential handling as appropriate, and a queue system for orders. It can still process quickly, but keep distinct tasks separate. If doing basket trades, make sure to treat them coherently (maybe as one grouped strategy or sequential small ones).

Example Prompt & Output

Scenario: The Execution agent has received the go-ahead to execute the earlier example trades: - Buy $500k of XYZ stock (which at $85/share is about 5882 shares – we can round to 5900 shares for a nice number or do 6000 with some tolerance). - Sell $300k of ABC stock (say ABC is $150/share, that's 2000 shares to sell).

Example Input to Execution Agent:
(This might come as a structured order list)

```
Order 1: BUY 5900 shares of XYZ at Market (approx $85 current price, total
~$501k)
Order 2: SELL 2000 shares of ABC at Limit $148 (current price $150, limit given
to ensure minimum price)
```

```
Notes: Execute ASAP, within normal trading hours. XYZ order is high priority
(market), ABC order use limit to avoid selling too low.
```

(Plus account info, etc., implicitly known)

Example Actions & Outputs from Execution Agent:

Action: Sends Order 1 (Buy XYZ) via broker API.
- Splits into smaller chunks perhaps: decides to use a market order for full 5900 shares since XYZ is liquid (volume millions) and $500k is small.
- API Response: Order filled. (We assume immediate fill for simplicity).

Output (log):

```
[Execution] Placed market buy order for 5900 XYZ.
[Execution] Order filled: 5900 XYZ bought at $84.98 average price (Immediate
fill).
Cost: $5900 * $84.98 = $501,382. Commissions: $100.
Execution Slippage: ~-0.1% vs $84.90 pre-trade price (slightly higher,
acceptable).
```

(This indicates the trade executed slightly above the price at signal time, which is fine.)

Action: Sends Order 2 (Sell ABC) via broker API.
- Chooses a limit order at $148 as instructed. Places the order on exchange. - Initially, only partial fills as price is currently $150 but maybe dropping. - After a few minutes, it sells 1500 shares at $150-$149 range, 500 shares remaining as price hits $148. - The agent sees remaining 500 unfilled at $148 (price now $148.5 on small bounce). It could leave it or adjust. - It might decide to adjust limit to $148.5 to get it done, as the directive was ASAP but with a floor of $148. - Adjust order: now limit $148.5, those 500 shares fill around $148.50.

Output (log):

```
[Execution] Placed limit sell order for 2000 ABC @ $148.00 minimum.
[Execution] Partial fill: 1500 ABC sold, avg price $149.20. 500 remaining.
[Execution] Price uptick detected, adjusting order to $148.50 for remaining
shares.
[Execution] Remaining 500 ABC sold at $148.50.
[Execution] Order 2 completed: 2000 ABC sold, avg price $149.00, within limit.
Proceeds: $2000 * $149.00 = $298,000. Commissions: $80.
Execution Note: Minor price adjustment used to complete order (maintained above
$148 floor).
```

Post-trade outputs:
- **Portfolio Update:**

```
Portfolio Update: XYZ position +5900 shares (new holding ~5900 shares, cost basis ~$85.0).
                 ABC position -2000 shares (new holding now ~2000 shares, reduced from 4000).
Cash: -$501k outflow for XYZ + $298k inflow from ABC = -$203k net (paid from available cash).
```

- **Confirmation to Director/Risk:**

```
Execution Summary:
- BUY XYZ 5900 @ $84.98 avg (filled fully)
- SELL ABC 2000 @ $149.00 avg (filled fully)
All trades executed successfully. Portfolio and cash balances updated.
```

- **Audit Log Entry:** (for compliance/audit systems)

```
TRADE EXECUTED: Buy 5900 XYZ @84.98; Sell 2000 ABC @149.00.
Order IDs: 1001, 1002. Time: 10:35-10:40 ET.
No errors. Logged by ExecutionAgent.
```

Explanation: The Execution agent logs show it carried out the buy immediately and handled the sell with a limit intelligently. It updated the records and reported back. The example demonstrates the agent's ability to adapt the sell order to get it done above the minimum price. Everything was done within a few minutes, and it provided a summary to the others. The agent's use of **trade-only API keys** and secure channels isn't explicitly shown in output, but that's assumed in the operation. The output logs give enough detail to verify execution quality and for compliance records.

---

[1] [13] Multi-Agent Trading System — How AI Builds Autonomous Investment Strategies | by Andrzej K. – Code & Tech Explorer | Oct, 2025 | Medium
https://medium.com/@ako74programmer/multi-agent-trading-system-how-ai-builds-autonomous-investment-strategies-ffda1383607a

[2] [4] [5] [7] [9] [12] AI Agents in Hedge Funds: Powerful, Proven Wins | Digiqt Blog
https://digiqt.com/blog/ai-agents-in-hedge-funds/

[3] [6] [8] [10] [11] The Rise of Autonomous Hedge Funds: How Swarm Intelligence and AI Agents Are Revolutionizing Trading (And How to Not Lose $128M in the Process) – Bright Coding – Blog pour les developpeurs
https://www.blog.brightcoding.dev/2025/11/16/the-rise-of-autonomous-hedge-funds-how-swarm-intelligence-and-ai-agents-are-revolutionizing-trading-and-how-to-not-lose-128m-in-the-process/