ChatGPT

Finance Data Summarization & Automation Challenge

Overview

In this hackathon challenge, participants design an AI-driven orchestration workflow using Kestra to integrate and summarize diverse financial data sources. Teams will leverage Kestra's built-in **AI Agent** to consolidate data from multiple systems (CRM, ERP, portfolio, compliance logs, news feeds, etc.) and produce concise summaries [1] [2]. Focus on *slow-moving but real-time accessible* data (e.g. end-of-day treasury positions, weekly financial reports, daily risk KPIs) rather than ultra high-frequency trading. Kestra simplifies connecting disparate data sources [3] and supports real-time updates (change-data-capture, event triggers) to keep information current [4]. Crucially, the AI Agent can "dynamically decide which actions to take and in what order," enabling it not only to report insights but to orchestrate next steps [5]. (The hackathon explicitly gives bonus credit for autonomous decision-making based on the summaries [1].)

Key Data Sources

Build a workflow that pulls from several realistic financial systems. For example:
- **CRM/Customer Data:** Client profiles, open deals, credit scores or risk ratings from Salesforce or an internal CRM.
- **ERP/Accounting:** General ledger entries, budgets, P&L and balance sheets from an ERP database or financial data warehouse.
- **Portfolio/Treasury Systems:** Asset positions, cash and debt balances, FX exposures, daily P&L and Value-at-Risk metrics from portfolio or treasury systems.
- **Compliance & Audit Logs:** KYC/KYB status, flagged or suspicious transactions, regulatory filings and audit trails.
- **Market News/Feeds:** Relevant financial news headlines, economic indicators, or research reports from public APIs or RSS feeds.

Teams should specify *exact* data fields or feeds (e.g. "weekly cash flow summary", "portfolio VaR by desk", "past-day stock market headlines"). Use Kestra tasks/plugins (JDBC, REST, CRM plugins, etc.) to fetch these. Kestra can easily **integrate various data sources** into one flow [3] and even use change-data-capture or triggers so that periodic data (e.g. monthly reports) is treated as fresh information [4].

AI Agent Tasks & Actions

The core requirement is to use the AI Agent to **aggregate and summarize** the combined data, and optionally act on it. For instance, the agent might:
- Compile a **Treasury & Risk Summary:** Merge cash forecasts, FX rates, and VaR trends into a daily report.
- Generate a **Portfolio Performance Brief:** Highlight top/bottom holdings and relate to recent news or market moves.
- Create a **Compliance Report:** List new audit alerts and regulatory issues with context and suggested actions.

Importantly, the agent should **trigger follow-up actions** based on its analysis. Example actions include:
- **Alerts & Notifications:** Send summary reports or urgent alerts via email/Slack/CRM. (Kestra even has blueprints for daily news digests to Slack/Notion [6].)
- **Automated Trades/Rebalancing:** If the agent detects that risk thresholds are exceeded, launch a sub-flow that rebalances the portfolio or places hedging orders.
- **Compliance Workflows:** Auto-create tickets or tasks in the compliance system when irregularities are found.
- **Client Communications:** Draft and send personalized portfolio updates or advisories to clients when significant events occur.

Teams should configure the AI Agent with system messages and prompts so it can autonomously decide which tasks to run. Kestra's AI Agent can "loop tasks until a condition is met" and adapt on-the-fly [5]. For example, it could first fetch data (via flow-calling tools [7]), then analyze it, and only invoke a trade-placing task if the analysis exceeds a threshold. This showcases **agentic orchestration**: the flow isn't hardcoded; the agent uses Kestra tasks (database queries, APIs, etc.) as tools to achieve its goal [7] [5].

Kestra Orchestration Role

Your solution should be modeled as a **declarative Kestra workflow** (YAML or UI) [8]. Each step – pulling from a database or API, posting a message, executing a trade – is a Kestra task. The AI Agent task ties them together. For example, one might set up: 1. A task to fetch treasury data,
2. A task to pull the latest news,
3. An AI Agent task with a prompt that ingests those inputs,
4. Conditional tasks that run if the agent's output signals an action.

Because Kestra flows are code, the entire process is versioned and auditable [8]. Teams can chain **multiple AI Agent tasks** (or even multiple agents) if needed. Kestra supports invoking sub-flows, so the agent could output a JSON flag that triggers another flow (e.g. the trading flow), achieving a multi-step pipeline. By leveraging Kestra's declarative orchestration and AI plugin, your agent can *think, remember, and decide* within the workflow [2] [5].

Example Use-Case Scenarios

Teams should illustrate the concept with concrete scenarios. Examples might include:
- **Corporate Treasury Monitor:** Fetch daily cash balances, interest rates, and debt schedules. The AI Agent summarizes liquidity position and warns if a cash shortfall is projected. If so, Kestra triggers a funding task or alerts the CFO.
- **Portfolio Risk Advisor:** Gather current holdings, market prices, and VaR metrics. The agent generates a risk report; if an asset's risk contribution is too high, it automatically executes a rebalancing flow.
- **Compliance Summary Agent:** Pull recent AML alerts, audit logs, and new regulations. Summarize the compliance health; if new issues are identified, auto-open compliance tickets or notify the legal team.
- **Executive Finance Dashboard:** Merge ERP financials (actual vs. forecast), sales pipeline, and macroeconomic news. Produce a narrative briefing; if key KPIs deviate beyond thresholds, schedule follow-up tasks or stakeholder notifications.

Each scenario should flow end-to-end: **data retrieval → AI summarization → optional automated action**. The novelty comes from creatively combining multiple data systems under one intelligent workflow and letting Kestra's AI Agent not only report, but **reason and act**. Judges will look for innovative use of data types, clear orchestration design, quality of the generated summary, and evidence of autonomous decision-making (beyond a static report) [1] [5].

Judging considerations: Use of diverse data feeds, clarity and usefulness of the AI-generated summary, creativity in the agent's actions, and effective use of Kestra's orchestration capabilities. The best solutions will showcase how Kestra's AI Agent can transform raw financial data into insights and real-time automated responses [2] [8].

Sources: Problem guidelines adapted from Kestra hackathon rules [1] and Kestra documentation on AI Agents and orchestration [2] [5] [8] [9].

---

[1] WeMakeDevs
https://www.wemakedevs.org/hackathons/assemblehack25

[2] [5] [7] AI Agents
https://kestra.io/docs/ai-tools/ai-agents

[3] [4] [6] [9] Simplify Your Data Engineering Challenges with Kestra
https://kestra.io/use-cases/data-engineers

[8] Kestra 1.0 — Declarative Orchestration with AI Agents and Copilot
https://kestra.io/blogs/release-1-0