Pathway presents HackWithNewYork @Microsoft NYC

Empower AI with Real-Time Data – A Hackathon Challenge for Builders

CONTENTS OF THIS DOCUMENT

We recommend reviewing all sections thoroughly. Use this table to quickly ensure you haven't missed any critical components. Let's go! 🚀

Background: The Power of Real-Time RAG

Large Language Models (LLMs) are incredibly powerful, but they have a weakness—their knowledge can become stale.

Enter **Retrieval-Augmented Generation (RAG)**: a technique where an AI retrieves up-to-date information from external sources before generating answers. This means an AI assistant could fetch the latest data on the internet or even your private document store, and then summarize or answer questions based on current information, bypassing the static limits of its training data.

Real-Time RAG takes this a step further by ensuring that as soon as data **changes**, the AI's knowledge and responses change with it. The result? Answers that are not only contextually accurate but also **instantly updated** with live information, greatly reducing outdated responses and AI hallucinations.

About Pathway

Meet **Pathway**—the open-source engine (Pathway Engine GitHub | Pathway LLM App GitHub) powering this hackathon's realtime magic. Pathway is a Python-based framework for high-performance streaming data processing and AI integration. In simple terms, Pathway lets you build pipelines that ingest data continuously (from APIs, files, databases, or sensors) and transform or index it on the fly. Under the hood, it uses a powerful Rust engine and incremental computation to handle streams and batches in one unified platform. It's trusted by the likes of F1 team(s), NATO, Intel, etc.

The Pathway team (a crack squad of researchers and advisors from places like Google Brain and OpenAI) built this tool to make real-time ETL and live indexing easy for developers—no separate Kafka or Flink stack needed. Pathway's advisor panel includes Lukasz Kaiser (co-creator of TensorFlow and co-author of foundational models behind ChatGPT), while the full-time research team features global programming champions and leaders such as Jan Chorowski (ex-Google Brain researcher and co-author with Nobel Prize winner Geoffrey Hinton).

What does this mean for you? It means you can focus on your application logic while Pathway handles the heavy lifting of keeping data flows synchronized. Pathway can, for example, ingest updates from 300+ sources with **built-in connectors**, and **automatically maintain up-to-date** data tables or vector indexes in memory. It's already trusted in domains from logistics to finance, but it's also perfect for hackathon projects—you get an always-on data backbone for your idea. Crucially for RAG, Pathway supports real-time **vector/hybrid search** and indexing: as new text or documents arrive, it can embed and index them immediately, so your LLM queries always hit the latest info. In short, Pathway is the real-time brain behind dynamic RAG apps.

You can check out the two repos below for a bit more context.

- Pathway LLM App Templates (GitHub)
- Pathway Core Repo (GitHub)

Problem Statement Track

The Problem: Stale AI in a Real-Time Financial World

Financial markets move in the blink of an eye, but too often our AI tools are a step behind. Traders, analysts, and compliance officers face an onslaught of information – from rapid-fire news feeds and price ticks to regulatory filings – yet **many AI systems respond with static, outdated insights**. Traditional large language models (LLMs) **lack real-time awareness**, relying on stale training data and missing the latest context. This leads to critical pain points in finance and investing:

- **Information Overload & Delays:** Markets generate more data than any human or static system can handle. Key signals from APIs, streams, or message queues arrive continuously, but legacy tools struggle to ingest and act on them instantly. Opportunities are lost in the lag.
- **Outdated AI Responses:** Whether in trading algorithms, compliance tools, or robo-advisors, AI that doesn't update with new data can give yesterday's answers to today's problems. A static trading bot might ignore this morning's earnings surprise, and a compliance assistant might miss a new regulation – with costly consequences.
- **Crowded Insights & Groupthink:** In hyper-competitive markets, everyone has access to similar data. Without real-time, unique signal integration, analysts fall victim to groupthink, lacking the edge to uncover novel opportunities. The winners are those who can synthesize fresh, alternative data faster than the crowd.
- **Keeping Up with Change:** From surprise economic news to fast-evolving regulations, the financial domain is in constant flux. Humans can't reliably keep up with hundreds of pages of filings, news articles, and social media buzz each day. Delayed awareness means compliance risks and missed market moves.
- **Missed Signals, Missed Opportunities:** When systems can't integrate new information on the fly, critical signals (a sudden spike in credit card fraud, an anomaly in portfolio performance, a geopolitical headline) slip through the cracks. The result? Missed opportunities, or worse – unseen risks.

The Challenge: Build a Live Fintech AI Solution

Your mission in this 5-day hackathon is to turn these pain points into opportunities. You will design and prototype a **Live AI-powered application for finance** that stays up to date with the market movements. The challenge is deliberately broad – you choose the use-case that inspires you – but the goal is the same: **create an AI solution that continuously learns from streaming data and updates its insights in real-time.** Leverage the power of Pathway, a Python-based streaming AI platform, to ingest and process live data, so your application's intelligence never goes stale.

Think about where real-time "live AI" could revolutionize finance or investing. Your project should address one or more of the core industry challenges above by dynamically integrating new data into AI-driven decisions or advice. It's up to you to define the specific problem and user scenario. For example, your team could build a:

- **Real-Time Trading Co-Pilot:** An assistant that tracks live market data, news, and social sentiment to surface "what's moving my portfolio?" insights via RAG/agents. It could use RAG/agents to answer a trader's question with fresh, verifiable context.
- **Sanctions Insight Bot**: Match the latest OFAC and EU lists with the day's high-value wires—say the next hundred transfers. The moment a newly listed name sneaks into a payment, Pathway freezes the item and shows "Payment paused, entity sanctioned three minutes ago." One short LLM call writes the plain-English blurb; the other ninety-nine wires sail through untouched. Furthermore, A click opens an LLM-powered Q&A panel about that entity.
- **KYC Behaviour-Drift Watchdog:** Keep each customer profile in Pathway and stream their live activity. When behaviour truly veers—for example a small local shop suddenly wires fifty thousand dollars overseas, Pathway fires an alert and the LLM explains, "Amount is 25 times the usual daily spend, unfamiliar destination country." Analysts see the why at a glance.
-
- **Market Anomaly Detector:** An AI service that continuously scans for out-of-pattern events – e.g. unusual stock order flows, sudden price moves, or macro indicators – and generates explanations or risk assessments. It might combine streaming data analytics with an LLM that continuously updates its answers as data evolves.
- **News-to-Portfolio Impact Estimator:** A tool that ingests breaking news, earnings reports, and social media sentiment, and dynamically updates an investor's portfolio outlook. For instance, if a tweet or news article breaks that affects a company, the AI could immediately highlight which of your holdings might be impacted and by how much.

…Or any other live Fintech AI idea you can dream up! The key is that your application should not be a static demo – it must showcase **real-time data flow and AI that reacts to new information**. We encourage creative approaches, whether it's a live chatbot advisor that

“learns” new facts on the fly, an automated analyst that writes reports with ever-fresh data, or an alert system that orchestrates multiple AI agents to investigate and report on an event.

Pathway’s Real-Time AI Superpowers

To build your solution, you’ll have the Pathway platform at your disposal – specifically designed for real-time, live-data applications. Here are some core Pathway features to supercharge your hackathon project:

- **Real-Time Data Ingestion:** Effortlessly stream data from APIs, feeds, databases, or message queues into your pipeline using Pathway’s built-in connectors. Pathway can easily ingest data from **hundreds of sources with automatic synchronization** – whether it’s Kafka topics of market ticks, WebSocket feeds of crypto prices, or JSON APIs of news. This means your app can consume live updates continuously, instead of waiting for batch updates.
- **Live Vector Search & Hybrid Indexing (RAG):** Make your AI smarter with Retrieval-Augmented Generation. Pathway lets you index documents and data in real-time, enabling **live vector searches** over embeddings and traditional keyword searches combined. This hybrid search approach ensures you retrieve not just relevant information, but the *right* information when answering questions. For example, you might index SEC filings or research reports as they arrive; when a user asks a question, your app can do a semantic vector lookup *and* a keyword match to find the most precise, up-to-date source.
- **Continuous Updates (Real-time RAG):** Say goodbye to stale answers. With Pathway’s streaming architecture, your AI’s knowledge base and outputs update on the fly as new data comes in. **Indexing documents in real-time ensures your RAG answers are always based on the latest data, not yesterday’s news**. In practice, this could mean an LLM agent that initially answers a query will refine or correct its answer minutes later if new information relevant to that query appears. Pathway’s real-time ingestion keeps AI responses up-to-date and context-aware, adapting as data evolves.
- **Agentic Workflows & Multi-Agent Orchestration:** Push the frontier by designing your solution as a team of AI agents working together. Pathway supports complex workflows where different agents can fetch data, perform analyses, and generate results in a coordinated pipeline. In an agentic setup, your AI system can autonomously recognize when it needs more information or a different expertise, and then trigger the appropriate agent. In other words, your AI can go from a passive responder to an active problem-solver – autonomously retrieving real-time updates, reasoning over them, and taking action. For example, a “market watcher” agent could monitor data streams and alert a “report writer” agent to generate a summary, which then calls on a “fact-checker” agent, and so on. Pathway helps orchestrate this chain in real-time.

- **Scalable, Low-Latency Pipelines:** Finance waits for no one, and Pathway is built to handle the speed and scale of modern markets. Its distributed, Python-based engine can **process massive datasets and high-frequency streams without sacrificing performance**. You can ingest millions of events and update AI models or indexes on the fly, all with millisecond-level latency. In short, Pathway provides a **production-grade, high-throughput pipeline** so your hackathon prototype can scale to real-world data volumes. Plus, it's all done in Python, so you can write your streaming transformations in a familiar language and trust Pathway to optimize and parallelize under the hood. And critically, **low-latency updates will keep your models and features current** as new data comes in – a must for real-time trading and risk applications.

Getting Started and What to Aim For

This challenge is beginner-friendly – you don't need prior experience with streaming data or Pathway to participate. Pathway's high-level API, extensive connectors, and example templates will help you hit the ground running. Over the next 5 days, focus on delivering an MVP (minimum viable product) that demonstrates real-time AI capabilities. Judges will be looking for:

- **Real-Time Functionality:** Does your application truly update and respond live as new data arrives? Show off that dynamic behavior – e.g., an evolving dashboard, continuous alerts, or AI responses that change over time.
- **Effective Use of Pathway:** Are you leveraging Pathway's unique strengths (stream ingestion, live indexing, etc.) to solve the problem? We want to see you unlock Pathway's "live AI" superpowers in your design.
- **Innovation & Impact:** How creatively does your solution address the chosen finance problem? Will it make a meaningful difference for users (traders, analysts, consumers, etc.) by alleviating information overload or providing a timely edge?
- **Clarity & Polish:** Since this is a hackathon, your prototype doesn't need to be perfect, but a clear presentation of how it works and what it achieves is essential. Bonus for intuitive UIs or visualizations that help showcase the live aspect of your project.

Imagine a future where trading desks, investment apps, or compliance departments run on AI co-pilots that are never out of date, always learning from the latest data. That future starts now – with you. This hackathon is your chance to explore how Pathway's real-time AI platform can transform finance, turning data deluge into actionable insight in the moments that matter. We can't wait to see the live Fintech innovations you create. Good luck, and happy hacking!

Solution Requirements

No matter which track you choose, every project must hit a few key requirements (think of these as the judging criteria basics):

- **Pathway-Powered Streaming ETL:** Use the Pathway framework to handle your data ingestion and processing. Your pipeline should continuously ingest and process data in real-time (e.g., reading a file directory, listening to an API or webhook, etc.) – this forms the backbone of your solution.
- **Dynamic Indexing (No Rebuilds):** New or updated data should be indexed or integrated on-the-fly, with **no manual reloading** of your AI's knowledge base. In other words, show off Pathway's real-time indexing engine – data changes flow through to answers immediately.
- **Live Retrieval/Generation Interface:** Provide an interface for users (or an API) to query or get outputs from your system. This could be a chatbot, a search bar, a Q&A endpoint, or a generative text/insight output. The crucial part: the responses **reflect the latest data**. If the underlying data updates at T+0, a query at T+1 should already include that update.
- **Demo Video:** Prepare a short video (e.g. 2-5 minutes) demonstrating your project in action. This should showcase live updates – for instance, you might screen-record your app, first showing it answer a query, then introduce a data change (add a file, trigger an update), and finally show the app responding to the same query with the new info. Prove to us in the video that your solution truly works in real time!
- **Optional:** Want multi-step logic or escalations? Orchestrate agents with LangGraph, Crew, etc. and use Pathway for connectors/vector stores, ETL for RAG, etc.—all of this, depending on your bandwidth. If you're short on time, you can label this as *future scope* if time runs short.

Note: You can use any additional libraries or models you like (LangChain, LlamaIndex, OpenAI API, etc.), but Pathway must be the core engine for streaming data and incremental processing. Also, as mentioned, **agentic RAG is optional** – if you want to incorporate an autonomous agent that decides how to route queries or when to fetch new data, go for it, but it's not required for a successful submission.

Note:

Non-Negotiable: Participants integrating an AI agent framework must deploy the custom agentic workflow by exposing their agent logic via a REST API endpoint using, ensuring seamless interaction with the real-time RAG pipeline powered by Pathway

Deliverables

Each team should submit the following:

- **Working Prototype** – A link to your running application on a **GitHub repository** with a runnable project. It should include clear instructions (in a README) on how to set up and run the system along with a working demo of how your solution works. We will be looking to run your Pathway pipeline and interface to test the real-time behavior and the presented solution.
- **Code Repository** – Your source code, preferably on GitHub itself. Make sure to document how Pathway is used in your solution (e.g., which module handles data ingestion, how the indexing is done, how the query interface works). If you used any pre-built components or templates, note that as well.
- **Demo Video** – As described in requirements, a short screen-recorded video demonstrating the live update flow. This is **crucial** for us to experience your hack without needing to run it from scratch immediately. Ensure the video highlights the before-and-after of a data update clearly.
- **Brief Presentation or Write-up** – *(Optional but encouraged)*. You might include a few slides or a markdown document summarizing your project's architecture and the problem it solves. Emphasize how data moves through Pathway, and how the LLM/RAG component produces results. This helps judges appreciate your design and any creative choices you made.

Evaluation Criteria

We're keeping the judging light and fun, but with an eye on key aspects of your hack. Here's what we'll be looking for:

- **Real-Time Functionality:** Does your project truly achieve real-time updates? We'll check that data changes propagate to the user-facing results with minimal latency. Using Pathway effectively here is a big plus.

- **Technical Implementation:** How well did you integrate Pathway and build your pipeline? Clever use of Pathway's features (connectors, streaming joins, vector indexing, etc.) will be noted. Also, overall code quality and project completeness matter (but remember, this is a hackathon – scrappy is okay as long as it works!).

- **Creativity & Innovation:** We reward originality. Did you tackle a unique problem or combine tools in an interesting way? Is your solution something that could be extended into a real product or open-source project?

- **Impact & Usefulness:** Think about the "so what" factor – does your hack demonstrate a compelling use-case for real-time RAG? Would a user or business find it genuinely useful or cool? We'll favor hacks that showcase why live data integration makes a difference.

- **User Experience & Demo Quality:** We don't expect polished UIs in a hackathon, but a clear presentation helps. If there's a user interface, is it intuitive to follow? Does your demo (and any write-up) explain the project well? Make it easy for us to understand what's happening and why it's awesome.

All criteria will be weighed collectively – this isn't a strict point system, but rather a holistic assessment. A simple hack that **nails the real-time aspect** and is well-presented can beat a complex hack that only half works. Aim for a **working proof-of-concept** that highlights the core idea effectively.

Starter Resources

To help you get going with Pathway and dynamic RAG, we've compiled some useful resources:

- **Pathway Documentation** – The official docs for the Pathway framework (installation, API, guides). Start here to learn how to set up data sources, create processing pipelines, and use Pathway's features (like its vector store, streaming joins, etc.).

- **Pathway GitHub Repo** – Explore Pathway's source code, examples, and README for insights into usage. The GitHub examples folder contains ready-to-run notebooks and templates for common scenarios.

- **RAG Beginner's Guide** – New to Retrieval-Augmented Generation? Check out *"Retrieval Augmented Generation: Beginner's Guide to RAG Apps"* on the Pathway blog. It covers why RAG is useful and how real-time data integration changes the game (great background reading).

- **LangChain Integration Guide** – If you plan to use LangChain or LlamaIndex with Pathway, see *"LangChain and Pathway: RAG Apps with always-up-to-date knowledge"*. This guide shows how Pathway can serve as a live data backend for LangChain pipelines, enabling up-to-date document search from LLMs.

- **Pathway Discord & Forums** – Got questions or need help debugging? The Pathway community is there for you (check the Discord link in the docs). While we can't give away solutions, folks can often point you in the right direction if you're stuck.

And of course, **don't hesitate to use your own ingenuity** and other tools. Stack Overflow, AI forums, and Pathway's examples are your friends. We've given you the building blocks – now it's up to you to build something amazing!

Domain Focus – Financial Applications (Examples):

For instance, in finance, potential applications include:

- **Compliance:** Automate the interpretation of new regulations (e.g., AML, MIFID) and flag changes via alerts.
- **Due Diligence:** Extract key metrics from pitch decks or risk reports.
- **Analyst Reports:** Generate dynamic investment analyses that reference real-time data.
- **Asset Management:** Merge diverse ESG data into up-to-date compliance summaries.

Bonus Ideas

- **Automated Alerts:** Trigger Slack/email alerts if a newly ingested document changes the answer to a compliance query.
- **Enhanced Summaries:** Summarize key financial metrics (e.g., EBITDA, ROI) in a structured table.
- **Extending Pathway:** Extend an existing class in Pathway to unlock new capabilities.
- **Handling Complex Data:** Show how to process non-textual formats (tables, charts) using vLMs where pure text-based LLMs might struggle.
- **Cutting-Edge Implementations:** Explore state-of-the-art models (e.g., Gemini 2.0) or design a "Faraday cage" approach (hosting LLMs locally) with zero external API dependency.

All the resources you need to get started:

- RAG Introductory Blog
- Building your first Realtime RAG pipeline with Pathway:
    - Building with OpenAI
    - Building with Gemini/Other LLMs
    - Implementation of Pathway with LlamaIndex
    - Implementation of Pathway with LangChain
- Building a Realtime Agentic RAG pipeline using LangGraph and Pathway
- LLM Tooling (Pathway's core software development kit for building a custom RAG pipeline, integrating Pathway into your existing codebase, or doing deep customizations)
- API Documentation for Pathway LLM xpack
- Pathway Developer Documentation: Link to Pathway Developer Docs
- How to deploy agents with Pathway?
    - Here you will see how you can build custom endpoints using Pathway RAG classes. There are two ways to serve agents: using the serve_callable API (which is easier to manage and recommended) or with an external web server like

FastAPI. If you prefer, you can start with an external web server and move the endpoint to Pathway later.

- Tips for resolving doubts?
    - Leverage Gen AI wisely. If you see difficult-to-comprehend error messages, the least you should do is ask the query on Gemini/ChatGPT, etc.
    - Utilize the #get-help channel on Pathway's Discord, if needed. However, given the nature of the competition, we wouldn't be able to share direct answers.