Vibe-a-thon (Geek Room X Pathway)

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Answers that are not only contextually accurate but also **instantly updated** with live information, greatly reducing outdated responses and AI "hallucinations."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 powering this hackathon's real-time 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 Tracks

Track: Open Innovation – Real-time RAG Playground

Unleash your creativity. This track is for open-ended innovation around Pathway and dynamic RAG. Your mission is to build *anything you can imagine* that showcases an AI solution continually learning from live data. The only hard requirement: use Pathway to ingest data in real-time (from a file feed, API stream, etc.) and build a query or generation interface that reflects incoming updates instantly.

Think of it as a playground for real-time AI ideas:

- You could create a **breaking news chatbot** that ingests news articles or tweets as they stream in and answers questions with up-to-the-minute facts.
- You might build a **collaborative document Q&A assistant** that lets a team ask questions about a knowledge base which teammates are updating in real time.
- Or perhaps an **IoT control panel** where an AI monitors sensor data and alerts, providing explanations or recommendations as new events happen.

The sky's the limit – **any domain, any use-case**, as long as your solution demonstrates an LLM or generative model actively augmented by fresh data. We encourage out-of-the-box ideas here. Surprise us with an application we didn't even know was possible! Just remember to leverage Pathway's real-time processing to handle your data streams and updates.

Bonus Twist: You can reimagine it in another domain if that excites you. The pattern is the same: live data changes feeding instant AI insights. Maybe you'll build a **Finance Portfolio Copilot** – when new stock prices or news headlines drop, your app updates a financial summary or Q&A assistant for an investor. Or an **IoT/Manufacturing Copilot** – when sensor fault logs stream in from machines, your app offers real-time diagnostic suggestions or updates a maintenance dashboard. These alternative takes are optional, but could earn creativity points (and show how versatile real-time RAG can be!). Just ensure the core requirement holds: **live data in, immediate insights out**.

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 or a GitHub repository with a runnable project. It should include clear instructions (in a README) on how to set up and run the system. We will be looking to run your Pathway pipeline and interface to test the real-time behavior.
- **Code Repository** – All your source code, preferably on GitHub. 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:** Especially for Track 1 (and bonus use-cases in Track 2), 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/ Bing AI search, 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.