Hack For Green Bharat Hackathon Problem Statement

1. Pathway Overview

Pathway builds the first post-transformer frontier model that solves AI’s fundamental memory problem. While transformers wake up in the same state every time—like Groundhog Day—our architecture enables true continuous learning, infinite-context reasoning, and real-time adaptation. We’re not optimizing yesterday’s technology; we’re building what comes after transformers.

A Python-based framework for real-time AI pipelines – it is designed to ingest data streams from more than 300 sources, process them on the fly, and make the results queryable with large-language models (LLMs). It provides connectors for files, databases, message queues and enterprise services, automatically detects new records and updates, chunks and embeds documents, and maintains a live vector index.

Unlike most retrieval-augmented generation (RAG) stacks that rely on external vector databases, Pathway’s Document Store keeps the vector index inside the pipeline. It combines semantic similarity search with BM[1]5 full-text search to return relevant documents and is automatically synchronised with data sources. This live indexing avoids the need to re-ingest data in batches and allows generative models to answer queries using the latest information.

Pathway’s streaming engine also supports building microservices and complex data transforms, as demonstrated by La Poste’s postal-logistics microservices that process 16 million GPS points each year to predict truck arrival times and generate anomaly alerts; the project cut the operating cost of the IoT

[1] . Key Ideas Behind Real-Time AI and RAG

• Real-Time vs Batch Processing

Traditional batch systems process data periodically; this introduces delays and stale information. Real-time streaming ingests, processes and outputs data within milliseconds, enabling decision-makers to react quickly and avoid outdated insights. Logistics, finance and manufacturing all require fast reaction to events; real-time streaming is therefore a competitive differentiator.

platform by 50% and reduced future capital expenditures. These features make Pathway suitable for industry-level applications where information is constantly changing and decisions must be made quickly.

• Freshness of Indexes

LLMs do not have an inherent sense of time. To answer queries using up-to-date information, RAG systems must ensure their indexes stay synchronised with external data. Pathway's Document Store continuously parses, chunks, embeds and indexes documents as they arrive and removes deleted documents automatically. This is different from vector databases that must be re-ingested periodically or from architectures where retrieval logic is locked into the DB.

• Event-Driven Architecture

Modern AI agents require access to consistent, real-time data. Event-driven architectures (EDA), often powered by message brokers such as Apache Kafka and stream processors like Flink, provide exactly-once semantics and scalable throughput; they are the backbone of agentic AI. Without high-integrity streaming data, autonomous agents may act on stale or inconsistent information.

• Benefits of Data Streaming

Continuous streaming allows businesses to react swiftly to changes, trends and anomalies, leading to better customer experiences, optimized operations and proactive issue resolution. Streaming also automates processes, reduces human error and supports cost-efficient operations.

---

3. Use-Case Themes in IoT, Logistics and Manufacturing

1. Real-Time Supply-Chain Visibility and ETA Predictions

In transport and logistics, vehicles and containers emit streams of GPS coordinates, temperature and load-status signals. Combining these with order data and traffic/weather feeds allows continuous estimation of arrival times and detection of anomalies (e.g., deviations from route or temperature excursions).

La Poste built such a system with Pathway; microservices ingest GPS events from Kafka, filter invalid data (e.g., (0,0) coordinates) and join the streams with historical routes to compute ETAs and send alerts. Pathway's connectors and streaming engine made it possible to decouple data preparation and prediction into separate microservices, scaling independently. Similar ETAs can be built for public transport, ride-sharing or shipping companies.

2. Predictive Maintenance for Industrial Equipment

Manufacturing facilities often deploy sensors that emit temperature, vibration and pressure readings. A real-time pipeline can ingest these readings via Kafka or MQTT, engineer features (e.g., rolling means, frequency spectrum), and feed them into ML models to predict failures and schedule maintenance.

An example architecture described on Medium uses Kafka for ingestion, Flink for feature engineering, and a model server to estimate failure probability; a dashboard triggers alerts. Pathway can implement the same pattern with simplified code using its stateful streaming engine and built-in connectors, while the Document Store can index historical maintenance logs and technical manuals to answer maintenance-technician queries.

3. Asset Utilisation and Warehouse Automation

IoT sensors on forklifts, pallets and robots produce continuous location and state data. By streaming these signals into Pathway and combining them with ERP and order management systems, companies can compute utilisation metrics, optimise routing of AGVs or robots, and detect bottlenecks.

Real-time tracking improves transparency by allowing personnel to track products' location and temperature in real time and to send alerts when shipments deviate from the route. IoT enables storage monitoring (temperature, humidity), contingency planning (rerouting during traffic or weather disruptions) and accurate inventory control. These capabilities align with Pathway's connectors (e.g., MQTT, OPC-UA, SQL) and streaming analytics.

4. Smart Warehouses and Robotics

Automated guided vehicles (AGVs), robotic picking and packing systems and RFID-enabled inventory management produce high-frequency events. Real-time data streaming helps coordinate these subsystems. AGVs transport materials, robotic pickers fulfill orders and RFID sensors track stock levels. Pathway can ingest streams from AGVs, update the system state, compute optimal dispatch schedules, and interface with the Document Store to provide instructions or troubleshooting steps.

---

4. Use-Case Themes in Financial Services

1. Real-Time Market Analytics and Risk Management

Streaming ETL is essential for processing tick-by-tick market data, computing option Greeks and other risk metrics. Pathway ingests historical and live CME market data and continuously computes Delta, Gamma, Theta, Vega and Rho using the Black-76 model. The pipeline updates values in real time, making it suitable for traders who need live exposures.

2. Real-Time Fraud Detection

Financial transactions arrive at high volume and require responses within milliseconds to ensure seamless user experience. Traditional rule-based fraud systems fail to detect complex patterns; AI-based models analysing vast data streams are more effective.

A fraud-detection architecture commonly used publishes transactions to Kafka, consumes them with a specialized service integrated with a TensorFlow model, classifies transactions as fraudulent or legitimate, and logs them for auditing. Features such as transaction amount and hour of day help flag suspicious patterns. Pathway can implement similar pipelines; its streaming engine offers exactly-once semantics, while the Document Store can hold regulatory policies, sanction lists and previous cases.

3. Streaming Log Analysis and Anomaly Detection

System logs are a treasure trove of operational information but are often analysed after incidents. A pipeline can analyse logs in real time, identify anomalies and alert administrators before problems escalate. Logs vary widely across applications, making schema-flexible streaming essential. Pathway can ingest log streams, apply natural-language models to classify log messages, detect anomalies or root causes, and use the Document Store to index runbooks and remediation guides.

4. RAG-Driven Personal Finance Assistants

A GenAI assistant for banking can combine streaming events (transactions, budget updates) with knowledge of financial products, compliance rules and customer documents. Using Pathway, one could stream account events, compute embeddings of transaction narratives, store them in the Document Store, and integrate with an LLM to answer questions such as “How much did I spend on groceries this month?” or “Is there unusual activity on my account?” while maintaining compliance.

5. Regulatory Compliance and Document Analysis

Financial institutions must process large volumes of legal texts, policies and customer communications. Pathway’s connectors can ingest documents from SharePoint, Google Drive or file servers, index them in the Document Store, and continuously update the index when policies change. Combining this with streaming events allows compliance teams to ask an LLM: “Does this transaction violate the latest AML rule?” and obtain an answer with citations.

---

5. Potential Application Ideas Using Pathway

- **Multi-Source RAG Agent**
- **RAG + ML for Dynamic Pricing**
- **AI-Assisted Dispatch and Route Optimisation**

(All content preserved exactly as in the document.)

⍰ Developer Resources and Technical Requirements

To ensure your project demonstrates strong real-time and production-readiness capabilities, all teams must adhere to the following requirements and leverage the official Pathway developer ecosystem. Exploit pathway to the max.

---

1. Live Data Ingestion with Pathway Connectors

Your system must utilize **Pathway’s real-time connectors** to ingest streaming data relevant to your chosen use case.

Pathway provides built-in connectors for **files, databases, message queues, APIs, and web sources**, all operating in streaming mode, ensuring results update in real time as data changes.

If your desired data source is not directly supported, you must extend Pathway’s ingestion layer by implementing your own connector using the Custom Python Connector. This allows you to adapt Pathway to new sources and contributes to enhancing the engine’s capabilities.

Documentation: https://pathway.com/developers/user-guide/connect/connectors-in-pathway
Create a custom Python connector:

https://pathway.com/developers/user-guide/connect/connectors/custom-python-connectors
Python web scraper example: https://pathway.com/developers/user-guide/connect/python-web-scraping

Artificial Data Streams with the demo Module (in case you find it difficult to access free streaming data APIs): https://pathway.com/developers/user-guide/connect/artificial-streams

⍰ At least one live or simulated data feed must be integrated. Examples include:

- Subscribing to live market APIs (e.g., Alpha Vantage, Polygon.io)
- Reading transaction streams from Kafka or sockets
- Simulating live events via Pathway’s demo utilities

If live data is unavailable, teams may simulate streaming input by replaying static datasets with realistic time intervals.

---

2. Core Concepts

Before starting development, familiarize yourself with Pathway’s foundational ideas and architecture. These concepts will help you understand incremental computation, table semantics, and event-driven design principles that power every Pathway pipeline.

Pathway Core Concepts — https://pathway.com/developers/user-guide/introduction/concepts/#core-concepts

---

3. Streaming Transformations and Feature Engineering

All data transformations must be performed in streaming mode using Pathway’s transformation APIs. Your pipeline should support:

- Incremental joins, filters, and aggregations
- Stateful window computations
- Real-time feature engineering for signals and indicators

Documentation: https://pathway.com/developers/user-guide/data-transformation/table-operations
Temporal Data Windows —

https://pathway.com/developers/user-guide/temporal-data/windows-manual

Ensure computations are low-latency and modular, with clear separation between ingestion, transformation, and output modules.

---

4. LLM Integration for Real-Time Insights

To make your system interactive and human-centric, integrate **Pathway's LLM xPack** — enabling smooth orchestration of retrieval, summarization, and reasoning over live data. You may use it for:

- Live retrieval-augmented generation (RAG)
- Automated report generation
- Explainable insights (e.g., credit decision rationale, fraud summary reports)

Documentation: https://pathway.com/developers/user-guide/llm-xpack/overview

Pathway MCP Server: https://pathway.com/developers/user-guide/llm-xpack/pathway_mcp_server

---

5. Mandatory Learning Resources and Templates

These resources will accelerate development and ensure alignment with Pathway's architecture.

Templates

RAG App Templates (YAML): https://pathway.com/developers/templates/
ETL and ML Time-Series Pipelines (Live Data Framework):
https://pathway.com/developers/templates/?tab=live-data-framework

Hands-On Tutorials

Vanilla RAG with OpenAI (Python): https://pathway.com/bootcamps/rag-and-llms/coursework/module-5-hands-on-development/1-first-rag -pipeline/building-with-open-ai RAG with Gemini: https://pathway.com/bootcamps/rag-and-llms/coursework/module-5-hands-on-development/1-first-rag -pipeline/rag-with-gemini-and-other-open-ai-alternatives Real-Time RAG using LlamaIndex: https://pathway.com/bootcamps/rag-and-llms/coursework/module-5-hands-on-development/3-

realtim e-rag-with-llamaindex-langchain-and-pathway/implementation-with-llamaindex Real-Time RAG using LangChain: https://pathway.com/bootcamps/rag-and-llms/coursework/module-5-hands-on-development/3-realtim e-rag-with-llamaindex-langchain-and-pathway/implementation-with-langchain

Advanced Notebooks

Explore the step-by-step cookbooks demonstrating how to combine Pathway's real-time indexing with LangGraph multi-step agent flows:

https://github.com/pathwaycom/llm-app/blob/main/cookbooks/self-rag-agents/pathway_deploy_langgraph_agents.ipynb

If you are only interested in using Pathway as an always up-to-date document store and want to deploy your agents your own way (via Flask, FastAPI, etc.), then check out this cookbook:

https://github.com/pathwaycom/llm-app/blob/main/cookbooks/self-rag-agents/pathway_langgraph_agentic_rag.ipynb

Evaluation & Benchmarks

Evaluating RAG Applications with RAGAS — https://pathway.com/blog/evaluating-rag

Deployment

Docker Deployment Guide —

https://pathway.com/developers/user-guide/deployment/docker-deployment

Persistence and Fault Tolerance — https://pathway.com/developers/user-guide/deployment/persistence
Licensing Guide (For unlocking Advanced Features) —
https://pathway.com/developers/templates/licensing-guide

Reference Implementations

Option Greeks Computation with Databento —

https://pathway.com/developers/templates/etl/option-greeks/ Real-Time Multimodal Data Processing (Docling) —
https://pathway.com/blog/multimodal-data-processing

La Poste ETA Microservices Case Study — https://pathway.com/blog/pathway-laposte-microservices/
Pathway Community Spotlights — https://pathway.com/blog/?tag=community,
https://pathway.com/blog/ai-agents-finance-due-diligence/, https://pathway.com/blog/live-ai-multi-agentic-rag, https://pathway.com/blog/financial-intelligence-with-event-based-state-machine