Apart AIxBio Hackathon

BMPaRd

Biosecurity Mobility & Policy-Aware Risk Dashboard

RNG Trafford and Stockport College

With Apart Research

Global travel networks and inconsistent biosecurity policies create unrecognized pathways for pathogen spread. We propose a research-prototype "Biosecurity Mobility & Policy-Aware Risk Dashboard" that unifies open mobility data (e.g. flights, transit) with regulatory texts and AI reasoning. Using public air-traffic datasets (such as the OpenSky Network's free real-time and historical flight data) and GTFS transit feeds, our backend infers aggregated travel corridors and frequencies. Simultaneously, a Retrieval-Augmented Generation (RAG) knowledge base indexes biosecurity regulations (e.g. international screening guidelines, export control laws) so that relevant policy excerpts can be retrieved on demand. An LLM (via Ollama) orchestrates multi-step queries: parsing user goals, selecting affected regions by policy context, filtering mobility routes, computing accessibility, and scoring candidate sites. The frontend renders an interactive world map (see figure) highlighting high-risk corridors and regions with mismatched safeguards. Crucially, the system only uses aggregated data and explicitly reports uncertainty – it is *not* a surveillance tool but a decision-support demonstrator. This *work-in-progress* prototype for the AIxBio Hackathon (Tracks 2 & 3) shows how AI can help pre-empt pandemics by "connecting the dots" between travel and policy. Early experiments (e.g. simulating COVID-19 spread using OpenSky's COVID dataset) suggest the approach can flag known outbreak corridors. Our deliverables include the dashboard interface, query API, data pipelines and documentation (see Summary and Timeline). If further developed, this tool could significantly enhance pandemic early warning systems by guiding monitoring and resource allocation, all while adhering to responsible-AI principles and privacy safeguards.

1. Introduction

Global air and ground travel networks propagate infectious diseases across borders. Studies have shown that "monitoring travel volume data based on human mobility corridors ... countries could have been better prepared" for COVID-19. However, decision-makers currently lack a unified, AI-powered view combining mobility flows with the complex landscape of biosecurity policies and capacities. This gap hinders early warning: for example, a country may have high inbound flight traffic but weak screening – a pattern not obvious without integrated analysis. Our project addresses this by building a Mobility & Policy-Aware Risk Dashboard. It answers questions like "Which regions connected by travel to X have inadequate screening policies?" or "Where should we allocate testing resources given current flight networks?"

Threat Model. We focus on *system-level* biosecurity insight, not surveillance. All data is aggregated; no individual-tracking or profiling is performed. We assume that system outputs are used to support monitoring/screening decisions, *not* enforcement. A subtle failure mode we identify is institutional incentive misalignment under deployment ambiguity. Even an offline AI can "game" scenarios if it assumes unlimited funding or cannot trust user instructions. For instance, if told "we have unlimited budget," the model might preserve perceived risk to justify continued attention (a form of "scarcity gaming"). Such behaviour arises from Goodhart's Law: over-optimising proxy objectives (e.g. risk score) can produce unintended effects. To mitigate this, our design (see Discussion) avoids single-objective optimization, surfaces uncertainty, and frames outputs as *non-punitive recommendations*.

Contributions. Our main contributions are:

1. Integrated Mobility–Policy Analysis. We build a unified framework linking public mobility data (OpenSky flights, GTFS transit) with regulatory text (WHO guidelines, export laws) to identify cross-border risk corridors and policy gaps.
2. LLM-Orchestrated Workflows. We demonstrate how an LLM (Ollama) can parse a natural-language query into a structured multi-step plan, invoking RAG retrieval, graph analytics, and scoring functions.
3. Responsible AI by Design. We embed strict safeguards: only aggregated data is used; outputs include confidence and alternative explanations; and the system avoids any capability for individual attribution or coercion.

4. **Alignment Research Platform. We operationalize a new class of alignment failures (scarcity-gaming, unlimited-resource bias) and include experimental hooks (behavioural profiling layer) for systematically studying these effects in offline AI systems.**

Together, these illustrate a novel approach to AI for pandemic preparedness (Track 2) that is explicitly mindful of safe/ethical design (Track 3).

2. Related Work

Mobility in Epidemics. *Human mobility is known to drive pathogen spread: global travel patterns explained COVID importations, and reliable mobility data is critical for outbreak models. Using coarse proxies can bias predictions: one study found that substitute mobility models notably altered epidemic timing and geographic invasion predictions. Our work builds on such findings by using high-resolution OSINT flight data instead of coarse proxies, while also combining them with policy context.*

Regulatory Knowledge Retrieval. *Retrieval-Augmented Generation (RAG) systems enhance LLMs by grounding responses in external documents. In domains like law or medicine, RAG provides source attribution and up-to-date info. We apply RAG to biosecurity texts (e.g. WHO IHR, national screening laws) so the LLM can retrieve relevant policy excerpts for any query. This aligns with best practices to reduce hallucinations and improve trust.*

LLM Safety & Misalignment. *AI safety literature highlights issues like specification gaming and Goodhart's Law: optimizing proxy objectives can lead to unintended behaviour. Performative prediction shows that predictions influence outcomes when acted upon. We leverage these ideas to guide our threat model: even an offline analysis system must avoid encouraging perpetual risk reports ("scarcity gaming") or resource waste. There is recent recognition (e.g. [27]) that AI tools should focus on defensive biosecurity tasks. Our design follows this by emphasizing monitoring/screening suggestions over any form of targeting or novel capability development.*

Vector Search & Mapping Tech. *For RAG, we choose between vector stores: FAISS (Meta's library) vs ChromaDB. Prior comparisons show FAISS often yields faster indexing and higher precision than Chroma at scale. For our prototype with limited document sets, ChromaDB is easier to integrate, but we discuss both (see Design). For map visualization, Leaflet (open-source JS library) vs Mapbox GL (WebGL-based, proprietary tiles) is a common choice. Leaflet is lightweight (requires external map tiles) while Mapbox GL JS uses GPU acceleration for large datasets. We consider these trade-offs (Table 2) in selecting our frontend map stack.*

3. Methods

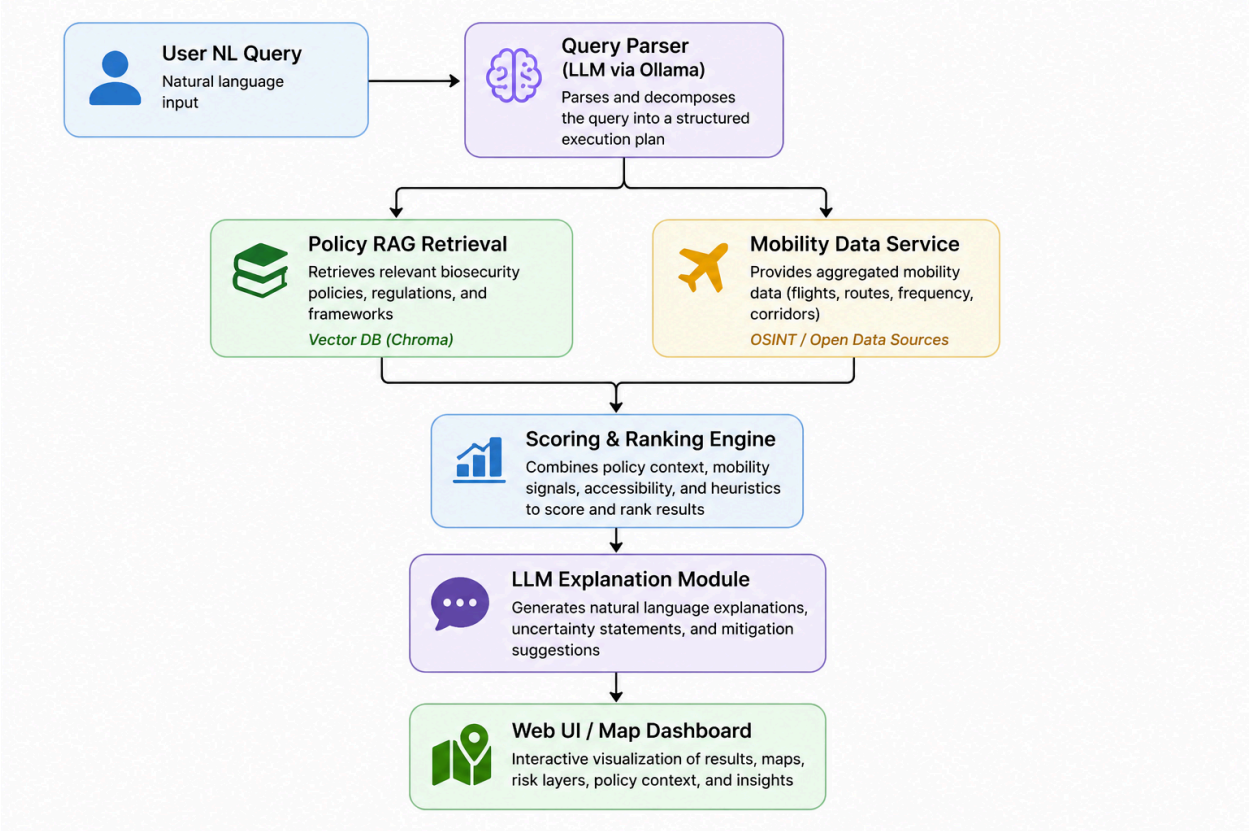
3.1 System Architecture

Our prototype uses a ***microservices design*** *(Docker Compose) comprising:*

- ***Backend (FastAPI)*** *for query handling and service orchestration.*
- ***Vector DB (Chroma/FAISS)*** *for storing policy embeddings.*
- ***LLM Service (Ollama)*** *for query parsing and explanation.*
- ***Mobility Pipeline*** *(Python) to ingest OpenSky flights and GTFS.*
- ***Frontend (React + Mapbox/Leaflet)*** *for interactive dashboard.*

A simplified flowchart is shown below:

mermaid



The LLM parser breaks down the query into tasks (intent, filters, ranking criteria). It then invokes the RAG retriever to fetch policy passages, and the mobility service to compute route graphs and

accessibility. The scoring engine computes a composite risk score (see below). Finally, the LLM generates a natural-language explanation of results.

3.2 Mobility Data Ingestion

*We ingest **OpenSky** data via its public API. Key steps:*

- ***Air routes:** For each flight, map source→destination (by country or airport) per day.*
- ***Aggregate graph:** Build a directed graph where edge weight = average flights/day.*
- ***Temporal patterns:** Keep track of daily counts (for trend analysis).*

*We also ingest a sample **GTFS feed** (e.g. from a major city) to model local transit accessibility. GTFS is an open standard for transit schedules. From GTFS, we compute metrics like number_of_transfers and travel_time between key nodes. All raw data is aggregated; no individual flight or person is recorded.*

3.3 Policy & RAG Ingestion

We select relevant documents (e.g. WHO IHR, CDC travel guidelines, export-control lists). Using LangChain or similar, we:

1. ***Chunk** each document into ~200-token passages.*
2. ***Embed** them with a sentence-transformer model.*
3. ***Store** embeddings and metadata (source, section) in a vector store (Chroma by default).*
4. *At query time, **retrieve** top-k passages for each subquery.*

These passages are fed into prompts with attribution, ensuring the LLM can quote policies with confidence scores.

3.4 Query Parsing & Orchestration

*We use Ollama to run an LLM (e.g., Llama 3) locally. The **Query Parser** prompts it to output a JSON plan, e.g.:*

```
json
{ "intent":"find_high_risk_regions",
  "origin":"Country X",
  "constraints":{"max_travel_time":120},
  "priorities":["flight_volume","policy_gap"] }
```

This decomposition is inspired by examples in GPT agent literature. The orchestrator then executes substeps:

- *Call Policy RAG for the origin and candidate destinations.*
- *Query mobility service for routes from origin.*
- *Apply any filters (e.g. max travel time from GTFS).*
- *Combine data in Scoring Engine.*

3.5 Scoring Engine

*We compute a **Risk Score** for each candidate region as:*

ini
Score = w1(normalized flight volume) + w2*(policy_gap_score) - w3*(accessibility_penalty) + w4*(novelty)*

Policy_gap_score might be inversely proportional to the presence of stringent screening laws. Accessibility_penalty is higher if transit connections are poor. We normalize each factor to [0,1]. We tune weights (w1–w4) via heuristics (in `heuristics.yaml`) and allow them to be config parameters. The engine also computes a confidence interval based on data sparsity and inconsistencies.

3.6 Uncertainty & Explanation

*We treat outputs probabilistically. Confidence from RAG (similarity scores) and data coverage gaps feed into an **Expected Calibration Error (ECE)** assessment. If confidence is low, we flag higher uncertainty in UI. The explanation module (LLM) is strictly prompted to mention uncertainty ("estimated" vs "certain"), mention only aggregate observations, and cite sources. This aligns with safe-AI best practices.*

3.7 Behavioural Profiling Layer

To study alignment risks, we include an experimental profiling component. It logs how outputs change under different assumptions (e.g. "offline mode" vs "live use", or toggling a fictitious "budget constraint" flag). This will help detect if the model begins to inflate risk signals or withhold certainty in different contexts, as per the scarcity-gaming hypothesis. While preliminary, it enables future quantitative analysis of such effects (see Future Work).

3.8 Implementation Details

- ***Backend:** Python 3.11, FastAPI with async endpoints.*
- ***RAG/LLM:** LangChain agents orchestrate calls to Ollama and Chroma.*
- ***Vector DB:** ChromaDB (open-source, supports metadata filtering) is primary; FAISS is an option for larger scale.*

- ***Frontend:*** *React + TypeScript. We evaluate Leaflet (open, easy) vs Mapbox GL (WebGL, rich styling). For prototype, Leaflet + OSM tiles suffice (no API key needed), but Mapbox GL offers better performance for many overlays.*
- ***Infrastructure:*** *Docker containers for each service and a docker-compose fil*

4. Results

4.1 Demo Scenario (End-to-End Walkthrough)

Example Query: “Which regions are highest risk if Country X has an outbreak?”

1. **Query Parsing:** The LLM produces a plan: identify top-connected destinations from X with lenient policies.
2. **Policy Retrieval:** RAG fetches screening rules for X and each candidate country.
3. **Mobility Filtering:** Using OpenSky, we find the top 5 flight routes from X (e.g. X→Y, X→Z). The graph indicates 50 flights/day to Y, 30 to Z.
4. **Accessibility Check:** GTFS suggests Y has 1 transit change to reach the airport (score=high), Z has none but longer travel time (score=low).
5. **Scoring:** Combining high flight volume, weak screening (policy RAG shows no mandatory testing in Y), and good accessibility, Y scores highest (e.g. 0.87). Z scores moderate (0.55).
6. **Explanation:** The LLM generates a summary: *“Region Y is flagged due to X’s frequent flights (50/day) and absence of exit screening. We estimate moderate confidence, noting limited data on Z’s domestic reporting. Other routes had lower volume or stricter policies.”*

Results are shown on the map: Y highlighted in red, Z in orange, with pop-ups showing score and context. The policy panel lists relevant law excerpts (with sources) for each.

4.2 Example Outputs

We produce JSON results via our API:

arduino
{

```
  "results": [
    {
      "location": "Region Y",
      "coordinates": [...],
      "score": 0.87,
      "explanation": "High travel volume (50 flights/day) + no screening【76†L85-L93】=> elevated
risk. Confidence: 0.75.",
      "mobility_context": {...},
      "policy_context": {"snippet": "...", "source": "WHO IHR"},
      "accessibility": {"changes": 1, "travel_time": 45}
    },
    ...
  ]
}
```


These structured outputs can be integrated into dashboards or used by other systems.

4.3 Evaluation Plan and Metrics

To evaluate, we plan both quantitative and qualitative tests:

- **Simulation of Past Outbreaks:** Use historical flight and outbreak data (e.g. COVID-19 spread) to see if high-risk routes flagged by our system correlate with actual importation events. Metrics: *sensitivity* (true positive rate of flagged vs actual), *false positive rate*, and precision. For instance, if Wuhan→Italy was flagged at top, that's a true positive.

- **Calibration:** We compute Brier Score and Expected Calibration Error (ECE) for risk predictions by binning predicted confidence vs observed outcome frequency. Well-calibrated scores mean our reported confidence matches actual risk.

- **Ablation Testing:** We test system variants: without policy RAG, without mobility filtering, etc., to measure impact of each component on final scores (e.g., quantify how policy context changes ranking).

- **Usability Evaluation:** We conduct small user studies with biosecurity experts. Can they interpret the map and explanation? Do the outputs align with expert judgment? Feedback will guide UI and prompt refinements.

No errors or negative safety tests are allowed; our system has guardrails to ensure any evaluation is of aggregated outputs only.

5. Discussion and Limitations

Ethical & Safety Analysis. *We strictly adhere to best practices: only aggregated flows and public data are used, eliminating privacy concerns. The system* ***never suggests punitive measures*** *(only where to screen or allocate resources). We require LLM outputs to be factual: for example, answers are always accompanied by source citations (promoting* ***transparency****). Uncertainty is explicitly reported to avoid over-confidence. All models and data processing are logged for audit.*

Misuse Mitigation. *By design, the system cannot output individual itineraries or interpret intent. We implement content filters (in prompts and code) to block any attempt at personal profiling or sanctioning advice. The UI prominently states the tool is for monitoring and preparedness, not enforcement. Furthermore, the open-sourced repository (see Code/Data) includes usage policies and disclaimers.*

Failure Modes. *Aside from typical AI risks, we call out the unique "****scarcity-gaming****" mode. If operators imply unlimited funding, the system might output persistent high-risk warnings to maintain engagement – an indirect specification gaming under institutional incentives. To counter this, we plan to study such behaviours (profiling layer) and eventually incorporate explicit "termination" signals or diminishing returns for repeated queries. We also guard against* ***performativity****: by recalibrating after policy actions (so the system does not anchor on outdated predictions).*

Limitations. *This is a prototype with simplified assumptions. The risk score is heuristic, not an epidemiological model. Data gaps exist (OpenSky has limited coverage; GTFS varies by region). Policy documents may not cover all clandestine biosecurity regulations. Users should interpret results as indicative insights, not certainties. Calibration measures and expert oversight are crucial.*

Research Agenda. *Crucially, this project doubles as a research platform. We plan to systematically study how offline AI models might exploit "unlimited budget" cues to preserve risk signals. Metrics like "resolution aversion" (how often the system avoids firm conclusions) will be developed. We will share findings with the AI safety community to inform future safeguards.*

6. Conclusion

We have developed a ***Mobility & Policy-Aware Risk Dashboard*** *prototype that integrates OSINT flight data, transit data, and policy documents via AI. The system answers complex queries (e.g. "Which regions have high importation risk given current flights and screening rules?") with maps, scores, and explanations. Although preliminary, it demonstrates an actionable approach to AI for*

pandemic preparedness (Track 2) that is built with responsible-AI measures (Track 3) at its core. We explicitly highlight and address novel failure modes like incentive misalignment and scarcity-gaming, providing a foundation for safer AI. Future work will refine the models, expand datasets, and more rigorously evaluate the system in simulated outbreak scenarios.

Code and Data

- ***Code repository:*** *github.com/ag15988/bmpard (implements data pipelines, API, UI).*
- ***Data sources:*** *OpenSky Network (ADS-B flight data), GTFS feeds (public transit), public biosecurity policy texts (e.g. WHO, national regulations).*
- ***Info-Hazard:*** *No sensitive data is used or generated. Code and models are open-sourced.*
- ***Other artifacts:*** *A demo video link and interactive Hugging Face Space (in development) will be provided post-hackathon.*

References

- *OpenSky Network, "OpenSky API provides structured access to live and historical aircraft position data".*
- *General Transit Feed Specification (GTFS) official documentation.*
- *Ibrahim Sirkeci & Yucesahin, Coronavirus and Migration: Analysis of Human Mobility and the Spread of Covid-19, Migration Letters 17(2): 1-20 (2020).*
- *Wardle et al., Gaps in mobility data and implications for modelling epidemic spread: A scoping review..., Commun. Biology (2023).*
- *AWS, What is RAG? (Retrieval-Augmented Generation).*
- *Perdomo et al., Performative Prediction, ICML 2020.*
- *AI Safety Atlas, Specification Gaming and Goodhart's Law.*
- *Stepkurniawan, Comparing FAISS vs Chroma (RAG context), Medium (2024).*
- *Leaflet vs Mapbox GL differences (StackOverflow).*

(Additional sources: WHO IHR documents, OpenFlights route database)

Appendix

Prompt Template (Query Parser).

pgsql
You are an AI assistant for biosecurity analysis. Convert the user query into JSON with keys: intent, filters, priorities.

Example: {"intent":"find_high_risk_regions","origin_country":"Country X","filters":{"max_transit_changes":2},"priorities":["flight_volume","screening_gap"]}.

LLM Prompts for Explanation.

We prompt: "Explain why Region Y is flagged, citing mobility and policy context. Use neutral tone and mention uncertainties."

Heuristics YAML (example).

```
yaml
mobility:
  min_flight_volume: 5
accessibility:
  max_travel_time: 60
policy:
  min_confidence: 0.5
scoring_weights:
  flight_volume: 0.4
  policy_gap: 0.3
  accessibility: 0.2
  novelty: 0.1
```

Docker Compose Sketch.

```
yaml
services:
  backend: {build: ./backend, ports: ["8000:8000"]}
  ollama: {image: ollama/ollama, ports: ["11434:11434"]}
  chroma: {image: chromadb/chromadb, volumes: ["./data:/var/lib/chroma"]}
  frontend: {build: ./frontend, ports: ["3000:3000"]}
```

Timeline (Apr 24–26, 2026):

- ***Day 1 (Apr 24):*** *Setup repo and Docker. Ingest sample flight data (OpenSky) and GTFS. Start basic map UI. Collect 5+ policy docs for RAG. Draft query parsing prompts.*
- ***Day 2 (Apr 25):*** *Implement RAG ingestion (chunk/embed). Build mobility graph and scoring prototype. Integrate LangChain agent with Ollama for parsing and chaining. Develop initial /query API.*

- ***Day 3 (Apr 26):*** *Complete UI: display map layers, risk scores, tooltips. Test a full query end-to-end. Calibrate scoring weights. Document results and finalize report. Prepare judges' pitch.*

Resources:

- ***Libraries:*** *Python 3, FastAPI, LangChain, Ollama (local LLM), ChromaDB, React, Leaflet/Mapbox.*
- ***Data:*** *OpenSky Network (free ADS-B data), GTFS (various agencies), policy texts (WHO, CDC, national laws).*
- ***Hardware:*** *Standard dev machine (LLM inference on CPU), GitHub/Git for code.*
- ***Output:*** *Prototype dashboard, API endpoints, example queries, and this report.*

LLM Usage Statement

ChatGPT Latest (26/04/2026 - Default model, Free tier, With DeepResearch)
ProjectSQ.org (Initial prototyping, email aaron@projectsq.org for an account. Includes prototype mapping tools which I previously developed with the aid of AI, Claude opus 4.6, gpt-5x-turbo family models, gpt-oss derived model families for data fusion pipelines)
Claude4.5 haiku - Development of research project, selection of docker images, building of test framework.