

CRISISWATCH AI

Humanitarian Crisis Detection & Funding Gap Intelligence Platform

Databricks x United Nations Geo-Insight Challenge — Full Technical Specification

Field	Details
Competition	Databricks x United Nations Geo-Insight Challenge
Track	Which Crises Are Most Overlooked?
Stack	Databricks, Agent Bricks, Genie Spaces, Three.js, Raspberry Pi Cluster
Hardware	4x Raspberry Pi 4B, 1x Jetson Nano, 1x RPi Camera Module
Document Version	1.0 — Full Technical & Implementation Spec

February 21, 2026

1. Executive Summary

CrisisWatch AI is an end-to-end humanitarian intelligence platform that identifies overlooked global crises by correlating humanitarian needs severity with Pooled Fund coverage gaps. The system fuses data from OCHA's Humanitarian Needs Overview (HNO), Humanitarian Response Plans (HRP), CBPF Pooled Funds, and real-time edge sensor streams into a unified, interactive 3D globe powered by Three.js. A distributed Raspberry Pi compute cluster serves as both a live IoT edge data source and a physical data-art installation that illuminates when funding gaps are detected in the region each node represents.

At the AI layer, Databricks Agent Bricks orchestrate multi-step reasoning pipelines that flag anomalous beneficiary-to-budget ratios, surface comparable benchmark projects, and generate plain-language crisis briefings. Genie Spaces provides natural-language SQL querying of the unified Delta Lake, enabling UN analysts to ask questions like 'Which clusters in the Horn of Africa are underfunded relative to peer crises in the last 18 months?' and receive instant, citation-backed answers.

2. System Architecture Overview

The platform is structured across four primary tiers: Edge (Raspberry Pi Cluster), Data Lakehouse (Databricks + Delta Lake), AI Reasoning Layer (Agent Bricks + Genie Spaces), and Presentation (Three.js Globe Interface). These tiers communicate via a streaming pipeline built on Databricks Structured Streaming with Kafka as the message broker.

2.1 Architecture Diagram (Logical)

TIER 1: EDGE	4x Raspberry Pi 4B (Regional Nodes: Africa, MENA, Asia, Americas) + 1x Jetson Nano (Inference Gateway) + 1x RPi Camera (Gesture Input)	<i>MQTT / HTTP</i>
TIER 2: INGESTION	Databricks Auto Loader + Kafka Structured Streaming. Batch ingestion of HNO, HRP, CBPF, population datasets. Stream ingestion of edge telemetry and OCHA API feeds.	<i>Delta Lake</i>
TIER 3: AI LAYER	Databricks Agent Bricks: Multi-step agentic pipelines for anomaly detection, benchmark matching, and briefing generation. Genie Spaces: Natural language interface over Delta Lake for UN analyst queries.	<i>REST + WebSocket</i>
TIER 4: PRESENTATION	Three.js WebGL Globe, interactive region click/point detection, real-time funding gap heatmap overlays, Genie Spaces embedded NL query panel, live edge-node status indicators.	<i>Browser / WebGL</i>

3. Data Pipeline & Delta Lake Architecture

3.1 Source Datasets

The platform consumes five primary public datasets alongside real-time edge streams:

Dataset	Source	Key Fields Used
HNO Data	data.humdata.org	Crisis severity, PIN (People in Need), cluster needs by country
HRP Data	data.humdata.org	Project-level: cluster, beneficiaries targeted, budget requested/received
CBPF Pooled Funds	cbpf.data.unocha.org	Allocation amounts per project, fund, country, cluster
Population Data (COD)	data.humdata.org	National/subnational population denominators
Requirements & Funding	data.humdata.org	Global requirements vs. funded amounts per crisis

Edge Telemetry	RPi Cluster (live)	Regional health proxies, anomaly signals, timestamp streams
----------------	--------------------	---

3.2 Delta Lake Schema

All data lands in a Bronze-Silver-Gold medallion architecture on Databricks:

Bronze Layer — Raw Ingestion

bronze.hno_raw -- Raw HNO CSV/JSON as-landed
bronze.hrp_projects_raw -- Raw HRP project-level records
bronze.cbpf_allocations_raw
bronze.population_raw
bronze.edge_telemetry_raw -- Streaming: RPi MQTT messages

Silver Layer — Cleaned & Conformed

silver.crisis_severity -- Normalized PIN, IPC phase, sector scores
silver.hrp_projects -- Deduplicated, validated projects with
geo-joins
silver.cbpf_allocations -- Standardized fund + project linkage
silver.population_cod -- Admin0/Admin1 population denominators
silver.edge_signals -- Parsed, validated RPi telemetry

Gold Layer — Analytics-Ready

gold.funding_gap_index -- Per-country, per-cluster gap score (needs
vs. funding)
gold.project_efficiency -- Beneficiary-to-budget ratios, z-score flag-
ging
gold.comparable_projects -- Similarity-matched benchmark project
pairs
gold.globe_viz_payload -- Pre-aggregated GeoJSON for Three.js
API
gold.genie_query_view -- Materialized view optimized for Genie
Spaces

3.3 Key Computed Metrics

Metric	Definition & Formula
--------	----------------------

Funding Gap Index (FGI)	$FGI = (\text{Requirements} - \text{Received_Funding}) / \text{Requirements} \times 100$. Ranges 0–100. Scores above 70 flagged as critically overlooked.
Beneficiary-to-Budget Ratio (BBR)	$BBR = \text{Targeted_Beneficiaries} / \text{Project_Budget_USD}$. Computed per project and per cluster aggregate.
BBR Z-Score	$Z = (\text{BBR_project} - \text{BBR_cluster_mean}) / \text{BBR_cluster_std}$. Flags projects ± 2 as anomalous.
Crisis Attention Score (CAS)	$CAS = \text{Media_Coverage_Index} \times \text{Funding_Ratio}$. Inversely related to overlooked status. (Lower CAS = more overlooked.)
Coverage Mismatch Index	$CMI = FGI \times (1 - \text{CBPF_Share})$. High CMI indicates high need, low pooled fund attention.

4. Databricks Agent Bricks — AI Reasoning Pipelines

Agent Bricks are the core intelligence engine of CrisisWatch AI. Four specialized agents run as Databricks workflows, each with access to the Gold layer Delta tables via Unity Catalog and external tool integrations.

4.1 Agent 1: Crisis Anomaly Detector

This agent runs on a scheduled trigger (every 6 hours) and on-demand when new HNO or HRP data lands. Its role is to identify crises with statistically unusual funding coverage relative to needs severity.

Component	Specification
Tools Available	Delta Lake query tool (Unity Catalog), Python UDF for statistical scoring, OCHA FTS API connector
Reasoning Steps	1. Pull latest gold.funding_gap_index. 2. Compute rolling 12-month baseline per region. 3. Flag countries where FGI delta > 15 points in 90 days. 4. Cross-reference with CBPF allocation recency. 5. Output structured JSON anomaly report.
Output Schema	{ crisis_id, country, cluster, fgi_score, fgi_delta_90d, cbpf_last_allocation_date, anomaly_severity: [LOW MED HIGH CRITICAL], recommended_action }

Integration	Writes to gold.anomaly_log. Triggers Globe WebSocket push for real-time UI update. Optionally pages OCHA Slack webhook.
--------------------	---

4.2 Agent 2: Project Benchmark Matcher

When an analyst clicks a project on the globe or submits a Genie query about a specific project, this agent computes semantic and statistical similarity to surface comparable projects for benchmarking high or low BBR ratios.

Component	Specification
Embedding Strategy	Project descriptions embedded using Databricks Mosaic AI embedding endpoint (BGE-Large). Stored in Delta table with Databricks Vector Search index.
Similarity Dimensions	Cluster type (WASH, Health, Food Security, etc.), Geographic region, Crisis phase (IPC 1–5), Project scale (beneficiary count bucket), Temporal overlap
Retrieval	ANN search on Vector Search index, filtered by cluster + crisis phase. Top-5 similar projects returned with cosine similarity score.
Output	Comparison table: current project vs. 5 benchmarks showing BBR, budget, beneficiaries, funding source, gap index. Rendered in globe side panel.

4.3 Agent 3: Genie Spaces NL Query Bridge

Genie Spaces is configured over the gold.genie_query_view materialized view, enabling UN analysts to query the entire humanitarian data warehouse in plain English. The bridge agent handles query routing, result formatting, and globe synchronization.

Example queries Genie Spaces handles natively:

- Which countries have a Funding Gap Index above 75 and received less than \$10M in CBPF allocations this year?
- Show me all WASH cluster projects in Sub-Saharan Africa where beneficiary-to-budget ratio is more than 2 standard deviations below the cluster average.
- Compare food security funding in Yemen vs. South Sudan over the last 3 years.

- What projects are most similar to HRP-2024-ETH-00423 and how does their efficiency compare?
- Rank the top 10 most overlooked crises by Coverage Mismatch Index.

When a Genie query returns geographic results, the bridge agent automatically sends a highlight command to the Three.js globe via WebSocket, causing matching regions to pulse on-screen.

4.4 Agent 4: Crisis Briefing Generator

For any flagged crisis region, this agent synthesizes a structured 2-page executive briefing using data from across the Gold layer. Briefings are generated in Markdown and rendered as downloadable PDFs directly from the globe interface.

Component	Specification
Briefing Sections	Situation Overview, People in Need (PIN) Breakdown by Cluster, Funding Gap Analysis, CBPF Allocation History, Comparable Crises (from Agent 2), Recommended Actions
Model	Databricks Model Serving endpoint (Claude or Llama-3 70B via DBRX) with retrieval-augmented generation from Delta context
Grounding	All statistics cited with source table, row ID, and timestamp to ensure auditability for UN use
Output Format	Structured JSON → Markdown → PDF via Databricks Workflows. Download link served via REST API to globe UI.

5. Genie Spaces Configuration

Genie Spaces is set up as a persistent, shareable data room for humanitarian analysts. It connects directly to Unity Catalog and surfaces a curated subset of gold tables with semantic column descriptions to improve query accuracy.

5.1 Space Configuration

Parameter	Value
Space Name	CrisisWatch Humanitarian Intelligence
Connected Catalog	crisiswatch.gold (Unity Catalog)
Primary Tables	gold.funding_gap_index, gold.project_efficiency, gold.comparable_projects, gold.genie_query_view

Access Control	Row-level security by region (UN cluster access permissions)
Semantic Descriptions	All columns annotated with plain-English descriptions for improved NL query accuracy (e.g., 'fgi_score: Funding Gap Index from 0–100, higher = more underfunded')

5.2 Trusted Assets

The following Databricks SQL queries are pre-registered as Trusted Assets in Genie Spaces, enabling reliable answers to the most common analyst questions without model hallucination risk:

- `top_overlooked_crises`: Returns countries ranked by CMI with FGI and CBPF data
- `cluster_bbr_distribution`: BBR z-score histogram per cluster per region
- `funding_gap_trend`: 24-month rolling FGI time series per country
- `cbpf_recent_allocations`: Last 90 days of CBPF allocation by country + cluster
- `peer_project_comparison`: Given a project ID, returns benchmark projects + metrics

6. Three.js Globe — Interactive Frontend

The primary interface is a WebGL-rendered 3D globe built with Three.js. Every country and subnational region is represented as a GeoJSON mesh layer on the globe surface. Funding Gap Index values drive a continuous color heatmap (green = funded, red = critical gap). Users interact via mouse hover, click, and — uniquely — a camera-based point gesture system powered by the Raspberry Pi Camera Module.

6.1 Globe Tech Stack

Component	Technology
3D Engine	Three.js r158 + WebGL2
Globe Geometry	Custom sphere with GeoJSON TopoJSON country/region meshes projected using D3-geo Orthographic → Three.js BufferGeometry
Heatmap Layer	Per-face vertex color shader driven by FGI score → GLSL gradient (green #00C57A to red #E8530A)
Data API	FastAPI (Python) on Databricks serving REST endpoint → <code>gold.globe_viz_payload</code> Delta table

Real-Time Updates	WebSocket connection to Databricks Serving endpoint. New anomaly signals push country highlight commands to globe.
Gesture Input	MediaPipe Hands (WASM) + RPi Camera stream via WebRTC. Index finger tip coordinates mapped to globe ray-cast for region selection.
UI Framework	Vanilla JS + CSS Grid. Side panel for project drill-down. Embedded Genie Spaces iframe for NL queries.

6.2 Point-to-Highlight System

The Raspberry Pi Camera Module streams video to a WebRTC endpoint served locally. The browser receives this stream and runs MediaPipe Hands inference in a Web Worker. When an index finger is extended and the hand confidence exceeds 0.85, the fingertip's normalized (x, y) screen coordinates are mapped to a Three.js Raycaster. The ray intersects the globe sphere and identifies the underlying country mesh face, which is then highlighted with a pulsing emissive shader animation. This creates a hands-free pointing interface for live demonstrations.

```
// Three.js raycasting from fingertip coordinates
const raycaster = new THREE.Raycaster();

const pointer = new THREE.Vector2(normX * 2 - 1, -(normY * 2 - 1));

raycaster.setFromCamera(pointer, camera);

const hits = raycaster.intersectObjects(countryMeshes);

if (hits.length > 0) highlightRegion(hits[0].object.userData.iso3);
```

6.3 Globe Interaction Modes

Mode	Behavior
Orbit Controls	Click + drag to rotate globe. Scroll to zoom. Double-click to snap to country.
Hover Tooltip	Country ISO3, FGI score, total people in need, last CBPF allocation date.
Click Drill-Down	Opens side panel: cluster breakdown, HRP project list with BBR z-scores, Agent 2 benchmark suggestions, briefing download button.

Finger Point	RPi Camera gesture: point at screen region → ray-cast → auto-select country and open drill-down panel.
Genie Query Sync	Genie results with country-level data trigger automatic globe highlight of matching regions (WebSocket command).
Anomaly Flash	When Agent 1 detects a new crisis spike, the affected country mesh flashes red with a pulsing animation for 30 seconds.

7. Raspberry Pi Cluster — Edge Architecture

The four Raspberry Pi 4B nodes are each assigned to one of four global humanitarian regions. Each node runs a lightweight Python agent that polls region-specific OCHA RSS/API feeds, simulates field data collection (configurable), and streams telemetry to the Databricks ingestion layer. Physically, each Pi is mounted with an LED array that illuminates in proportion to the region's current Funding Gap Index — turning the hardware cluster into a live data art installation.

7.1 Regional Node Assignment

Device	Region	Key Crises Monitored
Pi Node 1	Sub-Saharan Africa	Somalia, DRC, Ethiopia, Sudan, CAR, Chad
Pi Node 2	MENA + Horn of Africa	Yemen, Syria, Libya, Iraq, Afghanistan
Pi Node 3	Asia-Pacific + South Asia	Myanmar, Bangladesh (Rohingya), Pakistan, Philippines
Pi Node 4	Americas + Rest of World	Haiti, Venezuela, Colombia, Ukraine spillover
Jetson Nano	Inference Gateway	Runs MediaPipe, edge ML models, MQTT broker

7.2 Edge Node Software Stack

Component	Implementation
OS	Raspberry Pi OS Lite 64-bit (Bookworm)
Runtime	Python 3.11 + asyncio

OCHA Feed Poller	Polls OCHA ReliefWeb API every 15 min. Extracts crisis updates, new HRP projects, funding announcements.
Telemetry Publisher	MQTT client (paho-mqtt) → Jetson Nano broker → Databricks Kafka connector
LED Controller	RPi GPIO → WS2812B LED strip via rpi_ws281x library. Brightness = FGI score. Color: green→yellow→red gradient.
Health Beacon	Heartbeat ping every 60s with node ID, region, last poll timestamp, and LED state. Used for globe node status indicator.

7.3 Jetson Nano — Inference Gateway

Component	Implementation
Role	Central MQTT broker + real-time ML inference for camera-based gesture recognition
Camera Processing	Receives RPi Camera Module stream via GStreamer. Runs MediaPipe Hands model (TFLite) at 20 FPS on CUDA.
Output	Fingertip coordinates streamed to browser via local WebRTC server (aiortc). Also publishes gesture events to MQTT for globe WebSocket relay.
Failover	If Jetson is offline, browser falls back to pure client-side MediaPipe WASM (no camera stream required — uses laptop webcam).

7.4 LED Physical Installation Design

Each Raspberry Pi is housed in a clear acrylic enclosure with a WS2812B addressable LED ring (16 LEDs per node) mounted on the front face. The LED brightness and color are controlled by the latest FGI score for the node's assigned region, fetched every 5 minutes from the Databricks REST API. A score of 0–30 renders green, 31–60 yellow, 61–85 orange, and 86–100 pulsing red (critical). When the globe in the browser highlights a region matching a Pi node's assignment, the corresponding physical Pi's LEDs flash white three times — creating a physical-digital feedback loop that makes the live demo viscerally engaging for judges.

8. Bonus Feature Ideas & Extension Roadmap

The following features are suggested as stretch goals for the hackathon demo and future development. They are organized by implementation difficulty and impact:

8.1 High-Impact, Feasible in Hackathon Timeframe

Feature	Description
Temporal Replay Mode	Globe playback slider showing how funding gaps evolved month-by-month over 2019–2024. Three.js animation interpolates FGI heatmap values across time. Powered by pre-computed gold table partitioned by month.
Cluster Breakdown Ring	When a country is selected, an arc diagram renders around it on the globe showing the funding gap split by humanitarian cluster (WASH, Food, Health, Shelter, Education, Protection) using Three.js tube geometries.
RPi LED Sync on Genie Query	When a Genie Spaces query returns regional results, the matching Pi node(s) flash their LEDs in sync with the globe highlight — physical confirmation of the query result.
Mobile Gesture Mode	Fallback for the camera system: use device accelerometer (DeviceMotion API) to rotate the globe by tilting the phone. Point mode uses front-facing camera with MediaPipe WASM.

8.2 Medium-Effort, High-Value Extensions

Feature	Description
News Correlation Layer	Ingest GDELT or ReliefWeb news index. Score each country by media attention volume. Overlay on globe as second heatmap layer. Crises with high FGI and low media attention = most overlooked. Toggle between layers.
Predictive FGI Forecasting	Train a gradient boosted model (Databricks AutoML) on historical FGI + conflict events + weather anomalies to forecast funding gaps 90 days out. Display forecast confidence interval on globe.
Multi-Fund Comparison Panel	Side-by-side comparison of CBPF, CERF, bilateral donor flows for the selected country. Treemap visualization rendered in Three.js showing fund composition.

Voice Query Interface	Web Speech API → Genie Spaces query. Speak 'show me underfunded crises in East Africa' → globe animates to region and highlights matches. Enables truly hands-free operation with gesture + voice.
Crisis Similarity Network	Force-directed graph (Three.js + D3 force simulation) showing which crises are structurally similar based on Agent 2 embeddings. Clusters of crises that share characteristics visible as connected nodes floating above the globe.

8.3 Longer-Term Roadmap Ideas

- Integration with INFORM Risk Index for structural vulnerability weighting in FGI calculation
- Automated OCHA report generation: Agent Bricks auto-drafts funding appeals using Gold layer data + GPT-4o vision for satellite imagery analysis
- Conflict event overlay from ACLED (Armed Conflict Location Event Data) correlated with funding gap spikes
- UN agency dashboard: Separate Genie Space per UN agency (WFP, UN-HCR, UNICEF) with agency-specific project portfolios
- Donor recommendation engine: Given a donor's historical allocation patterns, suggest crises they are most likely to fund and estimate impact per dollar
- Field data integration: QR-code based mobile form (ODK/KoBoToolbox) where field workers scan to submit rapid needs assessments, auto-ingested into Bronze layer

9. Implementation Plan

9.1 Phase Breakdown

Phase	Timeline	Deliverables
Phase 1	Day 1–2: Data Infrastructure	Ingest all 5 datasets into Databricks. Build Bronze → Gold pipeline. Validate computed metrics (FGI, BBR, CMI).

Phase 2	Day 2–3: AI Agents	Configure Agent Bricks agents (Anomaly Detector, Benchmark Matcher, Briefing Generator). Set up Genie Spaces with trusted assets.
Phase 3	Day 3–4: Globe Frontend	Build Three.js globe with heatmap shader. Wire REST API. Implement hover/click interactions. Add Genie Spaces iframe embed.
Phase 4	Day 4: Hardware Integration	Flash RPi nodes. Configure MQTT + telemetry. Implement LED FGI controller. Set up Jetson Nano camera pipeline.
Phase 5	Day 4–5: Gesture System	Integrate MediaPipe with RPi Camera stream. Implement raycaster region selection. Test physical-digital LED sync.
Phase 6	Day 5: Polish & Demo Prep	Performance optimization. Demo script. Rehearse globe + gesture + Genie live demonstration.

9.2 Key Technical Risks & Mitigations

Risk	Mitigation Strategy
Risk: Network latency RPi → Databricks	Mitigation: Pre-aggregate Gold table payloads. Globe loads full dataset on init. Subsequent updates are delta patches only via WebSocket.
Risk: MediaPipe latency on gesture recognition	Mitigation: Run inference on Jetson Nano GPU. Throttle globe raycasts to 10 FPS. Add confidence threshold (0.85) to avoid jitter.

Risk: HNO/HRP data quality issues	Mitigation: Silver layer includes data quality scoring. Missing values imputed with regional medians. Anomalies flagged but not suppressed.
Risk: Genie Spaces cold-start on complex queries	Mitigation: Pre-warm 10 most common queries as Trusted Assets. Cache results in Delta table with 6-hour TTL.
Risk: Three.js performance with all countries rendered	Mitigation: LOD (Level of Detail) — simplify geometry at zoom-out. Only render cluster breakdown ring on zoom-in.

10. Appendix: API Reference & Data Dictionary

10.1 Globe REST API Endpoints

Endpoint	Description	Key Response Fields
GET /api/globe/heatmap	Returns GeoJSON with FGI scores per country for globe color layer	country_iso3, fgi_score, cmi_score, cbpf_total_usd
GET /api/country/{iso3}	Country drill-down: cluster breakdown, top projects, FGI trend	cluster_name, bbr, bbr_z_score, hrp_project_list
GET /api/project/{project_id}	Single project detail + Agent 2 benchmark suggestions	project metrics, comparable_projects[]
POST /api/genie/query	Proxy to Genie Spaces NL query. Returns results + globe highlight list	nl_query, results, highlight_iso3[]
WS /ws/globe	WebSocket for real-time anomaly push events from Agent 1	{ type: 'anomaly', iso3, severity, message }

10.2 MQTT Topic Schema (Edge → Databricks)

Topic: crisiswatch/{region_id}/telemetry

Payload: {

"node_id": "pi-africa-01",

"region": "SUB_SAHARAN_AFRICA",

"timestamp": "2025-02-20T14:32:00Z",

"fgi_local": 82.4,

```

    "led_state": "CRITICAL_RED",
    "ocha_feed_items_new": 3,
    "health": "OK"
  }

```

10.3 Glossary

Term	Definition
FGI	Funding Gap Index: percentage of humanitarian requirements not covered by funding
BBR	Beneficiary-to-Budget Ratio: people served per USD of project budget
CMi	Coverage Mismatch Index: composite of FGI and inverse CBPF pooled fund share
PIN	People in Need: OCHA estimate of affected population requiring humanitarian assistance
CBPF	Country-Based Pooled Funds: UN-managed emergency funds allocated to NGO/UN projects
HNO	Humanitarian Needs Overview: annual country-level needs assessment document
HRP	Humanitarian Response Plan: annual fundraising and project plan per crisis country
Agent Bricks	Databricks agentic AI framework for multi-step reasoning pipelines with tool use
Genie Spaces	Databricks natural language interface for Delta Lake / SQL analytics
IPC Phase	Integrated Food Security Phase Classification: 1 (minimal) to 5 (catastrophe)

CrisisWatch AI — Technical Specification v1.0

Built for the Databricks x United Nations Geo-Insight Challenge. All projects shared with UN teams.