

Comprehensive Technical Architecture and Feasibility Analysis for the NodiWatch Satellite Surveillance Platform

The conceptualization of the NodiWatch platform presents a sophisticated, multi-modal approach to environmental monitoring, attempting to synthesize optical and radar satellite imagery with advanced machine learning paradigms. Designed for the Eco-Tech Hackathon 2026, the project addresses three profound crises within the river ecosystems of Bangladesh: industrial pollution, illegal riverbank encroachment, and climate-accelerated riverbank erosion.¹ While the underlying premise of leveraging cloud-based remote sensing for environmental justice aligns perfectly with the hackathon's mandate to integrate modern technologies with traditional environmental science¹, a rigorous technical evaluation reveals distinct discrepancies between the proposed capabilities of open-access satellite sensors and the physical realities of optical physics, geospatial resolution, and legal evidentiary standards.¹

This analysis provides a deep, exhaustive blueprint for realizing the NodiWatch architecture using open-source frameworks and student-accessible infrastructure. It delineates the required technical pipeline from end to end, critically evaluates the project's feasibility by isolating physically impossible claims, reviews existing literature to ground the project in established methodologies, and provides a strategic realignment of the Phase 2 video pitch to ensure compliance with the strict evaluation criteria set forth by the hackathon organizers.¹

Evaluation of the Phase 2 Video Pitch Presentation

The transition from a conceptual slide deck to a highly constrained video pitch requires a significant structural and narrative reorganization. The Eco-Tech Hackathon Phase 2 evaluation involves an online video submission that must adhere to stringent formatting and content guidelines, which differ fundamentally from the structure of a traditional slide deck.¹ The evaluation criteria heavily weigh "Environment Relevance & Impact" (20%), "Use of AI/ Modern Technology" (20%), and "Presentation & Clarity" (15%).¹

Adherence to Strict Formatting Constraints

The hackathon organizers have established non-negotiable physical constraints for the Phase 2 video submission that dictate the visual and temporal rhythm of the presentation. The video must not exceed five minutes in length and must have a file size of under 100MB.² A critical constraint that fundamentally alters the presentation dynamic is that only one person—the lead author—may be featured in the video, and this individual must appear on-screen in person for

at least 50% of the total video duration.² Furthermore, while PowerPoint slides are permitted, they are restricted to occupying a maximum of half the screen.²

The current presentation document relies heavily on full-screen, visually dense infographics, such as the "Tri-Layer Heatmap Dashboard" and the "AI Architecture" flowcharts.¹ Utilizing these slides in a split-screen format on a standard display will render the text unreadable, particularly the finer details regarding Bayesian probability scores, specific spectral indices, and the map legends. The visual assets must be entirely redesigned to feature large, high-contrast typography and highly simplified architectural diagrams that remain legible when compressed into a fractional viewing window alongside the speaker. The visual strategy must pivot from comprehensive data display to symbolic visual anchoring, where the slide simply reinforces the verbal narrative delivered by the lead author.

Content Alignment and Strategic Narrative Gaps

The content of the pitch is mandated to cover specific thematic areas: the presenter's background or story, identification of the target market, a clear description of the problem, a concise description of the product/solution, a summary of the business model, progress to date, and next steps with a long-term vision.²

| Required Pitch Component | Current Coverage in Presentation Document | Strategic Alignment and Necessary Revisions |
|---------------------------|--|---|
| Background / Story | Completely absent. The current deck begins immediately with the problem statement. ¹ | A narrative introduction of the lead author must be drafted, establishing their technical or environmental expertise to build immediate credibility and fulfill the mandatory "story" requirement. ² |
| The Problem | Highly detailed. Accurately frames the "Triple Threat" (pollution, encroachment, erosion) and quantifies the economic and human displacement costs. ¹ | This section is strong but must be condensed for time. The 60% industrial pollution and \$500M annual erosion loss statistics provide excellent narrative hooks that should be delivered verbally while on camera. ¹ |

| | | |
|-------------------------|---|--|
| Target Market | Broadly categorized into B2G, B2B, and B2I segments. ¹ | The video pitch must specifically identify the immediate target market/customer to satisfy the rubric. ³ Focusing initially on the Department of Environment (DoE) and the Bangladesh Water Development Board (BWDB) as primary beachhead markets is recommended. |
| The Solution | Extensively detailed through AI layers (CNN, Random Forest, Bayesian, SAR) and dashboard features. ¹ | The technical explanation is robust but risks overwhelming the judges in a verbal pitch. The explanation must clearly articulate how the technology generates the environmental value without becoming bogged down in hyperparameter specifics. ³ |
| Business Model | Outlined as a freemium model with B2G SaaS licenses and Green Banking API pay-per-query streams. ¹ | This satisfies the "Sustainability and Market Accusation Strength" (10%) criterion. ¹ The pitch must concisely summarize this revenue generation plan to prove long-term viability. ³ |
| Progress to Date | Briefly mentioned on the Roadmap slide as "Current Stage: Early Prototype - GEE Pipeline Validated". ¹ | This requires significant expansion to meet the "Prototype Quality" expectations. ¹ A live, screen-recorded demonstration of the Google Earth Engine pipeline successfully classifying water or |

| | | |
|------------------------------|--|--|
| | | detecting a historical erosion event is mandatory to prove the team's technical capabilities. |
| Next Steps and Vision | Touched upon via the scalability path to expand to 1,400+ rivers. ¹ | Must be explicitly framed as the closing argument, detailing the immediate next steps (e.g., field validation) and the long-term vision of providing a digital twin for river resilience. ² |

The current presentation claims that the system will provide "court-ready evidence" and "admissible time-series satellite evidence for prosecution".¹ As will be explored in the feasibility analysis, these claims are legally and technically hazardous given the spatial resolution of the data. The pitch should pivot from claiming to provide definitive legal proof to providing "high-probability enforcement intelligence" that enables government agencies to perform targeted, highly efficient physical inspections.

Critical Feasibility Analysis: Identifying Impractical Claims

A profound understanding of satellite sensor limitations is paramount to presenting a credible technical architecture. The NodiWatch proposal contains several claims that transcend the physical capabilities of the specified satellite constellations, specifically Sentinel-1, Sentinel-2, and Landsat 8/9. Identifying, mitigating, and rephrasing these impossibilities is critical to surviving the "Technical Depth" scrutiny of the Phase 2 evaluation.¹

The Impossibility of Thermal Detection via Sentinel-2

The NodiWatch architecture explicitly claims to utilize Sentinel-2 imagery for "pollution + encroachment" monitoring, specifically stating that the AI uses "multispectral indices (NDTI, NDWI, thermal bands)" to detect "thermal discharge (temp spikes)" from industrial power plants.¹

This represents a fundamental physical impossibility based on the instrument payloads. The Sentinel-2 Multispectral Imager (MSI) is an optical sensor equipped with 13 spectral bands spanning the visible, near-infrared (NIR), and short-wave infrared (SWIR) spectrums, ranging from 442.3 nm to 2202.4 nm.⁴ It is entirely devoid of thermal infrared (TIR) sensors.¹ Therefore, Sentinel-2 cannot measure land or water surface temperature under any algorithmic

configuration.

While the proposal mentions Landsat 8 and 9 as secondary data sources, which do carry the Thermal Infrared Sensor (TIRS), the spatial resolution of these thermal bands is natively 100 meters, which is then resampled to 30 meters for distribution.¹ In the context of narrow, highly dynamic urban river systems like the Buriganga or Turag, a 100-meter thermal pixel will suffer from extreme spectral mixing. The pixel will aggregate the temperature of the water, the adjacent riverbank, nearby tin-roofed factories, and local vegetation. Detecting a localized thermal plume from a single factory discharge pipe using a 100-meter pixel is highly impractical and algorithmically noisy. To correct this critical error, the architecture must abandon claims of pinpointing factory-level thermal spikes using Sentinel-2 and instead focus on broader, segment-level thermal pollution monitoring using Landsat, or rely entirely on optical proxies such as turbidity and water color via the Sentinel-2 MSI.

Spatial Resolution Limits in Factory Attribution

NodiWatch proposes a Bayesian probability model that queries OpenStreetMap to identify factories within a 500-meter radius of a pollution hotspot, generating precise probability scores, such as "Textile Mill A: 78%", to pinpoint the exact polluter among thousands of dense factories.¹

This claim ignores the strict spatial limitations of the sensor. Sentinel-2's highest resolution optical bands operate at 10 meters per pixel.¹ A single pixel represents 100 square meters of surface area. Industrial effluent is typically discharged through pipes or narrow drainage canals that are fractions of a meter in diameter. By the time the effluent plume expands enough to dominate a 10-meter pixel—or a 20-meter pixel, which is required for critical red-edge and SWIR bands used in advanced index calculations—the effluent has already mixed with the ambient river water and, crucially, with the effluent of neighboring factories.¹

In densely packed industrial zones like Dhaka, multiple textile mills, tanneries, and chemical plants often share common, subterranean drainage networks that empty into the river at a single, consolidated outfall. Satellite imagery can only detect the plume at the outfall on the surface, not the subterranean network leading to it. Attempting to assign a 78% legal probability to a specific textile mill when another textile mill operates 50 meters away and utilizes the same drainage network is scientifically unsupportable using 10-meter optical data.¹ The Bayesian model should be logically reframed as an "Industrial Cluster Profiling" tool. Instead of identifying a single building, the model should assign probabilities to the type of effluent present in a river segment, thereby allowing the Department of Environment to narrow their physical search to a specific class of factories within a cluster, vastly improving the realism of the proposal.

The Fallacy of "Court-Ready" Encroachment Evidence

The proposal claims that Sentinel-2 data will automatically generate "court-ready evidence" and

"legal-grade proof" of riverbank encroachment.¹ Encroachment, particularly the illegal filling of riverbanks by land grabbers ("nodi dokhol"), often occurs incrementally, advancing merely a few meters a year to avoid immediate detection.¹

In environmental law and cadastral surveying, establishing property boundaries and proving illegal land reclamation requires sub-meter accuracy, typically achieved via RTK GPS or high-resolution drone photogrammetry. A 10-meter spatial resolution introduces a minimum inherent margin of error of ±10 meters, which is often exacerbated by sub-pixel water fractional cover, mixed boundary pixels, and atmospheric distortions.¹ If a land grabber fills 5 meters of a riverbank, it may not register as a full pixel change, or it may be entirely indistinguishable from natural seasonal sedimentation or tidal variation. Presenting automated evidence with a 100-square-meter pixel uncertainty in an Environmental Court against well-funded land developers is highly impractical and unlikely to meet evidentiary standards. The platform must be positioned as a wide-scale "early-warning triage system" designed to alert the National River Conservation Commission (NRCC) to deploy drone or physical surveying teams to specific coordinates, rather than serving as the final definitive legal evidence itself.

Geotechnical Complexity of Erosion Prediction

The platform claims the ability to "predict where banks will collapse next" using Sentinel-1 Synthetic Aperture Radar (SAR) combined with machine learning.¹ Sentinel-1 SAR is exceptionally well-suited for tracking historical shoreline regression because its C-band radar penetrates monsoon clouds, providing continuous 12-day observations regardless of weather conditions.⁸

However, predicting the exact spatial and temporal occurrence of future bank collapse based solely on historical two-dimensional surface regression is highly flawed. Riverbank erosion ("nodi vangon") is a complex three-dimensional geotechnical and hydrodynamic phenomenon. It is driven by sub-surface soil saturation, underwater toe-scouring by high-velocity currents, shear stress along the riverbed, and highly localized bathymetry. SAR backscatter from an orbiting satellite cannot measure underwater current velocity, analyze sub-surface soil composition, or determine the critical angle of the underwater bank slope. While machine learning can extrapolate historical retreat rates to forecast general high-risk corridors, claiming it can deterministically predict the exact location of a bank collapse oversimplifies river morphology.¹ The rhetoric should be carefully adjusted to "probabilistic erosion risk forecasting" based on historical morphological velocity and surface indicators, rather than absolute deterministic prediction.

Review of Existing Similar Works and Methodological Validation

A robust technical plan requires situating the proposed architecture within the context of existing academic and open-source methodologies. A comprehensive review of current

literature validates several of NodiWatch's core concepts while highlighting established, peer-reviewed algorithms that must be integrated into the proposed pipeline to ensure functional viability.

Riverbank Erosion Detection via Sentinel-1 SAR

The utilization of Sentinel-1 SAR for monitoring riverbank erosion in Bangladesh is well-documented, highly effective, and directly validates the proposed Layer 3 erosion engine. Groundbreaking research conducted by Freihardt and Frey on the Jamuna River established a definitive methodology for utilizing Google Earth Engine (GEE) to track erosion using Ground Range Detected (GRD) SAR data.⁸ Because the monsoon season brings heavy cloud cover precisely when hydro-geomorphological changes are most severe, optical satellites are rendered useless. The established algorithm relies on radar backscatter classification and temporal change detection across seasons.⁸

The methodology involves calculating the backscatter coefficient and applying rigorous speckle reduction techniques, such as the Refined Lee filter or spatial boxcar filters, to manage the inherent noise in SAR imagery.⁸ Pixels are classified into specific categories—such as settlement, trees, agricultural fields, sand, and water—based on empirical thresholds of mean backscatter. For instance, water surfaces generally act as specular reflectors, scattering radar energy away from the sensor, resulting in very low backscatter, while land and vegetation cause diffuse scattering, returning higher values.⁸ Erosion is quantified by detecting pixels that were classified as vegetated or structured land before the monsoon, which subsequently exhibit the low backscatter signature of water or wet sand immediately after the monsoon season concludes.⁸ The source code for this Jamuna River implementation is publicly available and provides a peer-reviewed foundation that the NodiWatch team must adapt and implement within their GEE pipeline to ensure functional accuracy.⁸

Optical Water Quality and Industrial Effluent Monitoring

Monitoring water quality and profiling industrial effluent via Sentinel-2 optical indices is a rapidly maturing field of remote sensing. Research specifically targeting the Buriganga River—a primary target for the NodiWatch pilot—demonstrates the high efficacy of this approach.¹⁰ Researchers successfully utilized Sentinel-2 Level-2A surface reflectance imagery, processed through the Sen2Cor atmospheric correction algorithm within GEE, to extract critical optical indices indicative of pollution.¹⁰

Key indices proven effective for the highly polluted waters of Dhaka include the Chromophoric Dissolved Organic Matter (CDOM) Index, calculated using the ratio of the Green (B3) to Blue (B2) bands. High CDOM values are strongly indicative of heavy organic pollution, which is highly correlated with raw sewage and specific industrial discharges, such as untreated tannery waste.¹⁰ Additionally, the Normalized Difference Chlorophyll Index (NDCI), utilizing the Red Edge (B5) and Red (B4) bands, measures chlorophyll-a concentrations, serving as a proxy for

eutrophication caused by nutrient-rich runoff.¹⁰

The literature highlights a critical preprocessing step that is often overlooked in hackathon projects: spatial resolution harmonization. Because the critical Red Edge bands (B5, B6) and SWIR bands (B11, B12) required for advanced index calculations operate at a 20-meter spatial resolution, they must be mathematically resampled—typically using bilinear interpolation—to match the 10-meter resolution of the visible bands before index computation can occur without generating artifact arrays.¹⁰ NodiWatch must explicitly incorporate this resampling step into its optical pipeline. Furthermore, studies on detecting synthetic textile dyes using Sentinel-2 highlight the utility of short-wave infrared (SWIR) bands, validating the proposition to classify effluent types based on unique spectral signatures.¹²

Machine Learning for River Encroachment Detection

The application of machine learning for automated river encroachment detection has seen practical implementation in Bangladesh through open-source initiatives. The Omdena Bangladesh chapter successfully executed a challenge focused entirely on automatic river encroachment detection using satellite imagery and machine learning.¹³ Their approach utilized open-source satellite data to perform exploratory data analysis of illegally occupied riverbanks, specifically targeting the Buriganga and Meghna rivers, aiming to deliver specific geolocations of illegal structures and filling.¹³ This precedent establishes that identifying morphological narrowing and infrastructure encroachment via spatial analysis is highly feasible within the scope of a hackathon and validates NodiWatch's proposed Layer 1 temporal analysis engine.¹⁴ The methodology typically involves extracting water masks using indices like the Modified Normalized Difference Water Index (MNDWI) and applying temporal overlay analysis to highlight zones of land gain at the expense of river width.¹¹

Bayesian Source Attribution in Environmental Modeling

The application of Bayesian probabilistic inference for pollution source attribution is an established practice, but its mathematical implementation requires careful adaptation for satellite data.¹⁶ In environmental modeling, the "backward problem"—identifying an unknown emission source from observed ambient concentrations—is notoriously ill-posed due to nonuniqueness and atmospheric smoothing.¹⁶

The standard mathematical formulation utilizes Bayes' theorem, where the posterior probability of the source parameters is proportional to the product of the prior probability and the likelihood function.¹⁶ In traditional models, complex algorithms such as Markov chain Monte Carlo (MCMC) are used to sample the posterior distribution, relying heavily on a forward atmospheric or hydrodynamic dispersion model to define the source-receptor relationship.¹⁶

The NodiWatch proposition of using a Bayesian model without an underlying hydrodynamic dispersion model—which requires in-situ flow velocity, depth, and bathymetry data completely

unavailable from optical satellites—is a significant deviation from traditional scientific literature. To remain technically viable and computationally feasible within a Next.js environment, the NodiWatch Bayesian implementation must be engineered as a simplified, spatial probabilistic ranking heuristic. It must utilize categorical prior probabilities based on industry type tags from OpenStreetMap and spatial likelihood functions based on distance decay from the outfall, rather than attempting a full, computationally prohibitive inversion of the advection-diffusion equation.

Comprehensive End-to-End Technical Pipeline and Setup Architecture

To deliver a functional, high-performance prototype for the Eco-Tech Hackathon 2026, the team must architect a cloud-native, serverless pipeline that effectively bridges the heavy geospatial processing requirements of Google Earth Engine with the responsive, user-facing capabilities of a modern web framework. The following section outlines the exact, step-by-step technical setup from A to Z, using exclusively open-source tools and free student tiers, specifically addressing the known architectural bottlenecks of integrating GEE with serverless environments.

Phase 1: Infrastructure Initialization and Authentication

The core architectural challenge of integrating Google Earth Engine (GEE) into a Next.js application hosted on a serverless platform like Vercel revolves around authentication and strict file system constraints. Vercel utilizes ephemeral serverless functions that operate on a read-only filesystem, with the sole exception of the /tmp directory. The standard GEE Node.js client library utilizes the xmlhttprequest module for synchronous API calls, which intrinsically attempts to write temporary synchronization files (e.g., .node xmlhttprequest-sync-*) to the root directory during the initialization process.¹⁷ This behavior inherently triggers an EROFS: read-only file system error, causing the application to crash upon deployment.¹⁷

To bypass this critical limitation and establish a secure connection:

1. **Google Cloud Platform (GCP) Configuration:**
 - Initiate a GCP project utilizing a student account to access free tier credits.
 - Navigate to the API Library and explicitly enable the "Earth Engine API".
 - Access IAM & Admin > Service Accounts and create a new, dedicated service account intended solely for the NodiWatch backend operations.¹⁸
 - Generate a new key for this service account and download the resulting JSON private key file.¹⁸
 - Register this specific service account email address within the Google Earth Engine platform to grant it the necessary computation and data access permissions.
2. **Bypassing the Vercel EROFS Limitation:**
 - Under no circumstances should the raw JSON key file be committed to the GitHub

repository. Instead, convert the entire contents of the JSON file into a Base64 encoded string.¹⁹

- Within the Vercel project dashboard, navigate to the environment variables settings and create a new variable named GEE_PRIVATE_KEY_BASE64, pasting the encoded string as its value.¹⁹
- To circumvent the Node.js xmlhttprequest file system write issue entirely, the team should adopt one of two architectural patterns: either utilize the Earth Engine REST API directly via standard asynchronous fetch calls by generating a JWT signed by the service account key, or offload the GEE processing to an external Python-based backend (such as FastAPI) deployed on a platform like Render, which provides a standard file system environment. For a Next.js monorepo, dynamically decoding the Base64 string in memory and initializing the client is required, but mitigating the synchronous write requires patching the library or utilizing the REST API.

Phase 2: Geospatial Database Architecture (Supabase PostGIS)

To persistently store historical river boundaries, factory geolocation data extracted from OpenStreetMap, and the generated pollution alerts, the system requires a robust spatial database. Supabase provides a fully managed PostgreSQL database with the PostGIS extension pre-installed, available on a generous free tier suitable for student projects.²⁰

1. **Supabase Initialization and Extension:** Create a new project within the Supabase dashboard. Navigate to the integrated SQL Editor and execute the following commands to establish the spatial architecture within a dedicated schema:

SQL

```
CREATE SCHEMA IF NOT EXISTS "gis";  
CREATE EXTENSION IF NOT EXISTS "postgis" WITH SCHEMA "gis";
```

2. **Table Architecture:** Construct tables optimized for complex spatial querying.
 - The factories table will contain columns for id, name, industry_type, and geom. The geom column must be defined as a PostGIS Point geometry type to store the exact latitude and longitude coordinates ingested from the OpenStreetMap API.²¹
 - The pollution_events table will contain id, timestamp, severity, pollution_type, and a geom column defined as a PostGIS Polygon representing the segmented spatial footprint of the satellite-detected hotspot.
3. **Spatial Indexing:** Spatial indexing is absolutely crucial for performance when calculating distances between dynamically detected hotspots and thousands of potential factory locations. Implement a GiST (Generalized Search Tree) index:

SQL

```
CREATE INDEX idx_factories_geom ON factories USING GIST (geom);
```

4. **Backend Integration:** Utilize the supabase-js client library within the backend routes to perform remote procedure calls (RPCs). These RPCs will execute custom PostGIS

functions written in SQL, such as ST_Distance or ST_DWithin, to rapidly query and return factories located near a detected hotspot.²⁰

Phase 3: The Google Earth Engine Processing Pipeline

The computationally heavy lifting of satellite data extraction, calibration, and algorithmic processing occurs entirely within the Google Earth Engine cloud infrastructure, effectively offloading the computational burden from the frontend application layer.

Layer 1: Water Segmentation & Encroachment Detection Pipeline

1. **Data Ingestion:** The pipeline filters the COPERNICUS/S2_SR_HARMONIZED (Sentinel-2 Level-2A surface reflectance) image collection by the specific bounding box of the target river segment and the desired temporal windows (e.g., aggregating imagery from the dry season of 2016 versus the dry season of 2026 to avoid monsoon flooding anomalies).
2. **Cloud Masking:** Apply the QA60 quality assessment band to mask out atmospheric interference. This involves executing bitwise operations to exclude pixels where bit 10 (opaque clouds) or bit 11 (cirrus clouds) are active, retaining only images with a total cloudy pixel percentage below 20% to ensure pristine optical data for boundary mapping.¹⁰
3. **Water Extraction:** While the proposal mentions Convolutional Neural Networks (CNNs), training a highly accurate deep learning model from scratch within the temporal constraints of a hackathon is resource-intensive. A highly practical and scientifically robust alternative is to utilize the Modified Normalized Difference Water Index (MNDWI): (Green - SWIR) / (Green + SWIR). This index suppresses built-up land noise effectively.¹¹ Applying an empirical threshold (e.g., > 0.1) classifies the pixels into a binary water mask.
4. **Temporal Differencing:** The core encroachment detection involves subtracting the 2026 binary water mask from the historical 2016 water mask. The resulting positive pixels represent areas that were classified as water in 2016 but are classified as land or built-up infrastructure in 2026, providing a quantitative spatial footprint of river encroachment.

Layer 2: Spectral Fingerprinting and Classification

1. **Spatial Resampling:** To ensure algorithmic integrity, the pipeline must ensure that all required 20-meter bands (such as the Red Edge and SWIR bands) are computationally resampled to 10 meters using bilinear interpolation via the `image.resample('bilinear')` function in GEE.¹⁰
2. **Index Calculation:** Compute a stack of specific optical arrays to feed the classification model. This includes turbidity proxies using the Normalized Difference Turbidity Index (NDTI) and organic loading proxies using the CDOM Index (B3/B2 ratio).¹⁰
3. **Machine Learning Classification:** Feed these calculated index bands, along with raw optical bands, into an `ee.Classifier.smileRandomForest` algorithm. This model must be trained on a curated FeatureCollection of polygon geometries representing ground-truthed locations of known tannery outfalls, textile outfalls, and clean river water.

Layer 3: SAR-Based Erosion Tracking Pipeline

1. **Radar Ingestion:** Access the COPERNICUS/S1_GRD image collection. Filter for 'VV' (vertical transmit, vertical receive) polarization and 'IW' (Interferometric Wide swath) instrument mode, which are optimal for water-land boundary delineation.
2. **Speckle Filtering:** SAR data is inherently noisy due to the constructive and destructive interference of the coherent radar waves. The pipeline must apply a spatial filter, such as a 3x3 Refined Lee filter or a focal median filter, to smooth the backscatter values and reduce speckle noise.⁸
3. **Thresholding and Detection:** Analyze the backscatter histogram to establish a classification threshold. Based on literature for Bangladeshi rivers, a threshold around -13.2 dB effectively separates water and wet sand from vegetated land.⁸ The pipeline identifies erosion by isolating pixels that transition from high backscatter (land) before the monsoon season to low backscatter (water) immediately following the monsoon, calculating the spatial geometry of the eroded landmass.

Phase 4: Modified Bayesian Factory Attribution Logic

Because implementing a fully coupled computational fluid dynamics (CFD) dispersion model is impossible within the computational and data constraints of a hackathon project, the Bayesian attribution layer must be cleverly engineered as a spatial probability heuristic executed within the PostGIS database environment.

When the GEE pipeline detects a pollution polygon, the geographic centroid coordinates and the predicted effluent_class (derived from the Random Forest model) are transmitted to the Supabase database.

1. **Prior Probability Assessment:** The system queries the local database of industrial nodes (previously ingested from OpenStreetMap). A prior probability is assigned based on the categorical match of the industry type. If the Random Forest detects a spectral signature consistent with "Textile Dye," all factories tagged with industrial=textile receive a high prior probability weight, while a facility tagged as building=warehouse receives a negligible prior weight.
2. **Spatial Likelihood Calculation:** The system utilizes the PostGIS ST_Distance function to calculate the exact Euclidean distance from the centroid of the detected pollution hotspot to each factory located within a 500-meter search radius, established using ST_DWithin. The likelihood function is designed to decay exponentially as distance increases, mathematically modeling the physical reality that a concentrated pollution plume is most likely to originate from the nearest outfall.
3. **Posterior Calculation and Normalization:** The backend algorithm multiplies the categorical prior probability by the spatial distance decay factor for each candidate factory. It then normalizes the results across all candidate factories within the search radius so that the total probability equals 100%. The system outputs the resulting ranked list, presenting the intelligence as probabilistic guidance (e.g., "Factory X: 78% likelihood of

association") rather than absolute deterministic proof.

Phase 5: The Tri-Layer Dashboard Frontend Integration

The frontend serves as the critical visualization layer, translating the complex mathematical intelligence generated by GEE and Supabase into the "actionable enforcement intelligence" promised in the project goals.

1. **Framework:** The application is built using Next.js utilizing the modern App Router architecture for optimized server-side rendering and efficient API route handling.
2. **Mapping Library:** Implement a highly performant WebGL mapping library such as MapLibre GL JS or react-leaflet to handle complex spatial geometries without browser lag.
3. **Data Fetching and Rendering Strategy:**
 - For discrete vector geometries—such as the point locations of factories and the specific polygon boundaries of encroachment zones—the frontend fetches GeoJSON data directly from the Supabase REST API or via GraphQL.
 - For the computationally massive continuous raster data—specifically the pollution severity heatmaps and the raw SAR imagery—the architecture relies on GEE's tile generation capabilities. The backend utilizes the ee.Image.getMapId() function to generate dynamic, map-ready tiles directly from the Earth Engine servers. The resulting Tile URL is passed back to the Next.js frontend, where it is rendered as a lightweight raster tile layer superimposed over the base map. This architectural decision is vital; it ensures that the client's browser is not burdened with downloading or rendering gigabytes of raw satellite imagery, guaranteeing a fluid and responsive user experience.

Strategic Realignment and Conclusions

The NodiWatch architecture represents a highly ambitious and conceptually brilliant integration of planetary-scale cloud computing and local environmental enforcement mechanisms. To secure victory in Phase 2 of the Eco-Tech Hackathon 2026, Team AlphaVerse must strategically align their presentation with the unyielding physical realities of satellite remote sensing and the strict structural constraints of the video pitch format.

The presentation narrative must proactively discard claims of identifying specific factory pipes using thermal bands on Sentinel-2, or generating sub-meter court-ready property boundaries from 10-meter optical data. These claims, while well-intentioned, constitute technical impossibilities that will be swiftly dismantled by any evaluating judge possessing a background in Geographic Information Systems or remote sensing physics.

Instead, the pitch narrative must pivot aggressively to highlight NodiWatch's true, defensible technological strength: it operates as an unprecedented, automated, macro-scale environmental triage system. It brilliantly utilizes the cloud-penetrating capabilities of Sentinel-1

SAR to monitor insidious riverbank erosion relentlessly through the monsoon season, and it leverages complex multispectral indices to profile the nature of industrial effluent clusters with high accuracy. By transforming raw, petabyte-scale satellite arrays into an easily interpretable spatial probability matrix, NodiWatch possesses the power to direct incredibly limited government resources—such as Department of Environment physical inspection teams and Bangladesh Water Development Board drone surveyors—to the highest-risk geographical coordinates with unparalleled efficiency.

By executing the detailed, open-source GEE-to-Supabase pipeline outlined in this report, securely managing service account authentication, and focusing the tightly constrained 5-minute video pitch heavily on a live, visual demonstration of the water segmentation algorithms and SAR speckle-filtering process, the team will demonstrate profound technical depth. This approach will satisfy all evaluation criteria, presenting NodiWatch not merely as an idea, but as a highly viable, scalable, and meticulously engineered environmental solution.

Works cited

1. presentation.pdf
2. accessed February 22, 2026,
uploaded:photo_2026-02-16_17-38-02.jpg-b13de523-a1c9-4073-a292-496fa21872fa
3. accessed February 22, 2026,
uploaded:photo_2026-02-16_17-38-26.jpg-e5f986df-7563-43bc-ae36-aa5a8f9b321c
4. Sentinel 2 Bands and Combinations - GIS Geography, accessed February 22, 2026, <https://gisgeography.com/sentinel-2-bands-combinations/>
5. Sentinel-2 Multispectral Imager - NASA Earthdata, accessed February 22, 2026, <https://www.earthdata.nasa.gov/data/instruments/sentinel-2-msi>
6. Sentinel-2 Data for Land Cover/Use Mapping: A Review - MDPI, accessed February 22, 2026, <https://www.mdpi.com/2072-4292/12/14/2291>
7. Sentinel-2 Bands, accessed February 22, 2026, <https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/bands/>
8. Assessing riverbank erosion in Bangladesh using time series of Sentinel-1 radar imagery in the Google Earth Engine - NHESS, accessed February 22, 2026, <https://nhesst.copernicus.org/articles/23/751/2023/>
9. Code of tool to assess riverbank erosion along Jamuna River in the Google Earth Engine, accessed February 22, 2026, <https://zenodo.org/record/7252970>
10. Water Quality Monitoring of Buriganga River, Dhaka: Incorporating Remote Sensing and Machine Learning - ResearchGate, accessed February 22, 2026, https://www.researchgate.net/publication/399119220_Water_Quality_Monitoring_of_Buriganga_River_Dhaka_Incorporating_Remote_Sensing_and_Machine_Learning
11. Water Bodies' Mapping from Sentinel-2 Imagery with Modified Normalized Difference Water Index at 10-m Spatial Resolution Produced by Sharpening the SWIR Band - MDPI, accessed February 22, 2026,

<https://www.mdpi.com/2072-4292/8/4/354>

12. A Novel Approach to Using Spectral Imaging to Classify Dyes in Colored Fibers - MDPI, accessed February 22, 2026, <https://www.mdpi.com/1424-8220/20/16/4379>
13. prathimacode-hub/Automatic-River-Encroachment-Detection - GitHub, accessed February 22, 2026, <https://github.com/prathimacode-hub/Automatic-River-Encroachment-Detection>
14. Automatic River Encroachment Detection in Bangladesh with Machine Learning and Remote Sensing | Local Chapters | Omdena, accessed February 22, 2026, <https://www.omdena.com/chapter-challenges/automatic-river-encroachment-detection-in-bangladesh-with-machine-learning-and-remote-sensing>
15. River classification and change detection from landsat images by using a river classification toolbox - Semantic Scholar, accessed February 22, 2026, <https://pdfs.semanticscholar.org/350b/5f43593aa9a95e67504dc0a6e8d7b0ce6041.pdf>
16. Bayesian Inference for Source Reconstruction: A Real-World Application - PMC, accessed February 22, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC4897316/>
17. google earth engine - GEE in a NextJs monorepo on Vercel? - GIS StackExchange, accessed February 22, 2026, <https://gis.stackexchange.com/questions/462901/gee-in-a-nextjs-monorepo-on-vercel>
18. Service Accounts - Earth Engine - Google for Developers, accessed February 22, 2026, https://developers.google.com/earth-engine/guides/service_account
19. Linking Google credentials to vercel - next.js - Stack Overflow, accessed February 22, 2026, <https://stackoverflow.com/questions/74299271/linking-google-credentials-to-vercel>
20. Geo Queries with PostGIS in Ionic Angular - Supabase, accessed February 22, 2026, <https://supabase.com/blog/geo-queries-with-postgis-in-ionic-angular>
21. PostGIS: Geo queries | Supabase Docs, accessed February 22, 2026, <https://supabase.com/docs/guides/database/extensions/postgis>