UDC 004.9:004.4 DOI https://doi.org/10.35546/kntu2078-4481.2025.3.2.26

M. O. ILIN
Postgraduate Student at the Computer Systems Software Department
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
ORCID: 0009-0001-0803-3726

L. M. OLESHCHENKO
Candidate of Technical Sciences,
Associate Professor at the Computer Systems Software Department
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
ORCID: 0000-0001-9908-7422

DEVELOPMENT OF A MODULAR DATA UNIFICATION PIPELINE FOR REAL-TIME ENVIRONMENTAL THREAT ANALYTICS

The article presents the results of research and development of a modular software system for real-time environmental risk analytics, built on the concept of unified event ingestion and the application of H3-based hexagonal spatial aggregation. The relevance of this work is driven by the high heterogeneity of primary data sources, including open analytical reports (OSINT), satellite fire detections (FRP), ionizing radiation dose rate measurements, and meteorological fields. These sources rely on different temporal scales, measurement units, and data schemas. Such diversity creates significant challenges in integration, resulting in timestamp shifts, unit inconsistencies, schema conflicts, and event duplication, which complicates the timely production of consistent results and slows down updates of information layers. The objective of the study is to design a compact and reproducible data processing pipeline capable of transforming heterogeneous event streams into daily aggregated layers of risk and demand with sub-second latency. The proposed architecture consists of three main components. The first is the Plugin Source Adapter Interface (PSAI), which maps each data source into a standardized event table; adapters are responsible only for source-specific parsing, while subsequent unification logic is shared. The second component is the Deterministic Event Harmonization (DEH) module, which converts all timestamps to UTC format, normalizes measurement units, verifies coordinates, and guarantees idempotent inserts. This ensures safe reprocessing and proper handling of late-arriving data. The third component is a query layer oriented towards H3-first spatio-temporal aggregation: FRP points are aggregated into 15-minute intervals at H3 resolution level 10 using the MAX(FRP) operator to avoid overlaps, after which daily summaries at H3 resolution level 7 integrate FRP data with OSINT events, radiation monitoring signals, and meteorological indicators, including a wind alignment proxy, to construct a comprehensive risk index.

The system is implemented in the DuckDB environment without the use of dedicated servers or network services, which simplifies infrastructure and reduces operational costs. The adopted approach ensures transparency and testability of integration contracts: the PSAI interface enforces data ingestion rules, the DEH module standardizes time and units, the H3 hierarchy guarantees spatial consistency, and DuckDB provides materialization of results through standard SQL. Experimental evaluations confirmed the high efficiency of the proposed solution: data updates are performed within fractions of a second even on consumer-grade laptop hardware; duplication of satellite FRP detections is reduced by approximately 99 %; daily summary exports remain under one megabyte in size, significantly simplifying their transfer and storage. The developed system combines the flexibility of modular architecture, the reproducibility of integration procedures, and the efficiency of the computational model, establishing a foundation for practical applications in environmental monitoring, risk management decision support, and the advancement of ecological analytics services.

Key words: *software system, real-time analytics, data unification, H3, DuckDB, materialized views, environmental monitoring.*

М. О. ІЛЬЇН
аспірант кафедри програмного забезпечення комп'ютерних систем
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0009-0001-0803-3726

Л. М. ОЛЕЩЕНКО
кандидат технічних наук,
доцент кафедри програмного забезпечення комп'ютерних систем
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0000-0001-9908-7422

РОЗРОБКА МОДУЛЬНОГО КОНВЕЄРА УНІФІКАЦІЇ ДАНИХ ДЛЯ АНАЛІТИКИ ЕКОЛОГІЧНИХ ЗАГРОЗ У РЕАЛЬНОМУ ЧАСІ

У статті представлено результати дослідження та розробки модульної програмної системи для аналітики екологічних ризиків у реальному часі, що ґрунтується на концепції уніфікованого приймання подій та застосуванні шестикутної просторової агрегації H3. Актуальність роботи зумовлена високим рівнем гетерогенності первинних джерел даних, серед яких: відкриті аналітичні звіти (OSINT), супутникові детекції пожеж (FRP), вимірювання потужності дози іонізуючого випромінювання та метеорологічні поля. Зазначені джерела оперують різними часовими шкалами, одиницями вимірювання та схемами подання даних. Це призводить до виникнення труднощів у процесі їх інтеграції, зокрема, до зміщення часових міток, невідповідності одиниць, конфліктів структур та дублювання подій, що ускладнює оперативне отримання узгоджених результатів і уповільнює оновлення інформаційних шарів. Метою дослідження визначено побудову компактного та відтворюваного програмного конвеєра для обробки даних, здатного перетворювати змішані потоки подій у добові агреговані шари ризику та попиту із затримкою на рівні секунд. Запропонована архітектура включає три основні компоненти. Першим є інтерфейс плагінів-адаптерів джерел (PSAI), що забезпечує відображення кожного джерела у стандартизовану таблицю подій; адаптери відповідають лише за специфічний розбір вхідних даних, тоді як подальша логіка уніфікації є спільною. Другим компонентом виступає модуль детермінованої уніфікації подій (DEH), який переводить усі часові позначки у формат UTC, нормалізує одиниці вимірювання, верифікує координати та гарантує ідемпотентні вставки, що дозволяє безпечно здійснювати повторні прогони та коректно враховувати запізнілі дані. Третій компонент – запитний шар, орієнтований на застосування просторово-часової агрегації за принципом H3-first: точки FRP агрегуються у 15-хвилинні інтервали на рівні H3 r10 із використанням оператора MAX(FRP) для уникнення накладань, після чого добові зведення на рівні H3 r7 інтегрують дані FRP з подіями OSINT, сигналами радіаційного моніторингу та метеопоказниками, зокрема, проксі-індикатором напрямку та швидкості вітру, для формування комплексного індексу ризику. Реалізація системи здійснена у середовищі DuckDB без залучення окремих серверів чи мережевих сервісів, що спрощує інфраструктуру та знижує вартість експлуатації. Використаний підхід забезпечує явність та тестованість інтеграційних контрактів: інтерфейс PSAI фіксує правила приймання даних, модуль DEH – стандартизацію часу та одиниць вимірювання, ієрархія H3 – просторову узгодженість, а DuckDB – можливість матеріалізації результатів засобами стандартного SQL. Проведені експерименти підтвердили високу ефективність запропонованого рішення: оновлення даних виконуються за долі секунди навіть на потужності обладнання рівня ноутбука; дублювання супутникових FRP-детекцій скорочується приблизно на 99 %; добові експорти зведень мають обсяг менше одного мегабайта, що істотно спрощує їх передачу та зберігання. Таким чином, розроблена система поєднує гнучкість модульної архітектури, відтворюваність інтеграційних процедур та ефективність обчислювальної моделі, що створює підґрунтя для практичного використання у сфері моніторингу довкілля, підтримки рішень з управління ризиками та розвитку сервісів екологічної аналітики.

Ключові слова: *програмна система, аналітика в реальному часі, уніфікація даних, H3, DuckDB, матеріалізовані подання, моніторинг довкілля.*

Problem statement

The relevance of the research lies in the growing global demand for timely and reliable environmental monitoring tools capable of supporting risk management and decision-making processes. Modern environmental data originate from highly heterogeneous sources – ranging from satellite fire detections and radiation measurements to meteorological fields and open-source intelligence reports – that differ in temporal resolution, measurement units, and data schemas. This heterogeneity complicates integration, leading to inconsistencies, delays, and information overload that hinder the effectiveness of analytical software systems. A modular and reproducible data unification pipeline, capable of harmonizing diverse event streams into coherent, compact, and near-real-time risk layers, addresses these challenges directly. Such an approach not only improves data quality and processing speed but also enhances the scalability and accessibility of environmental analytics, thereby providing a robust technological foundation for sustainable development, emergency response, and ecological risk assessment.

Heterogeneous streams – textual incident reports, satellite fire detections, station radiation, and weather – are difficult to align. Time zones and units drift; schemas diverge. The objective is a lightweight, reproducible path from raw feeds to daily risk and demand layers with seconds-level latency. The approach combines a modular ingest interface, deterministic homogenization, and H3-based materialized views implemented in DuckDB.

Concretely, source adapters emit rows in a unified event schema. The DEH stage converts all timestamps to UTC-naive time, harmonizes units (dose rates in sieverts, wind in m/s), validates coordinates, and performs idempotent upserts using a natural key. For FRP, we bin points into 15-minute micro-tiles at H3 r10 and take MAX (FRP) per bin/cell to collapse overlapping satellite detections. Daily rollups at H3 r7 then join FRP with OSINT hits, radiation signals, and a wind-alignment proxy. Materialization is expressed in SQL and executed inside DuckDB, eliminating external orchestration while keeping refresh logic transparent and testable.

Related research

Prior work on real-time environmental analytics clusters into six lines: compute engines, spatial indexing, fire products, radiation networks, meteorological proxies, and access/transport. We outline each and note trade-offs.

Studies on embedded columnar databases show that single-node, vectorized SQL can meet interactive workloads without a separate server. DuckDB typifies this class and reports high scan and aggregation rates on commodity hardware [1]. This line reduces latency and operations, but it shifts responsibility to schema design and careful memory use.

Multiscale summarization commonly uses hierarchical tessellations. H3 offers hexagonal cells with stable parent-child relations and neighborhood structure, enabling consistent rollups and topology-aware joins across resolutions [2]. Hexagons limit orientation bias and support regional statistics, yet cell area still varies with latitude and must be considered in analysis.

Active-fire research relies on satellite detections and fire radiative power. FIRMS distributes MODIS/VIIRS data that many authors composite in space and time to handle overlapping overpasses and repeated hits before daily or regional summaries [3, 4]. Typical choices include temporal binning and cell-level maxima or sums; these increase robustness but can suppress short-lived transients.

Environmental dose monitoring aggregates heterogeneous stations. Platforms such as EURDEP and community networks face mixed units (nSv/h, μR/h, CPM), uneven cadences, and diverse device quality. Published workflows emphasize unit harmonization, timezone normalization, and basic outlier screening prior to regional synthesis [5, 6]. Harmonization improves comparability, but device-specific calibration remains an open issue.

When full dispersion models are infeasible, open weather APIs supply wind direction and speed as lightweight surrogates for transport and exposure. Prior studies combine directionality with source locations to approximate downwind influence or to weight risk indicators at grid scale [7]. Such proxies scale well, though they cannot replace plume physics in complex terrain or stability regimes.

Response and demand depend on movement over networks. Open street-graph tooling (OSMnx) and open routing engines (Valhalla) enable isochrones and travel-time metrics on public data [8, 9]. This body of work supports "reachable population" and logistics-aware assessments without proprietary datasets, at the cost of careful preprocessing and profile tuning.

The literature indicates that embedded OLAP can satisfy sub-second analytics [1]; hierarchical hex grids provide stable spatial rollups [2]; satellite and sensor products require deterministic preprocessing before aggregation [3–6]; and open routing stacks make travel-time analysis tractable on public data [8, 9]. These strands define the current toolkit for building compact, reproducible systems in this domain.

Proposed modular data unification pipeline

We integrate four streams that stress different parts of the pipeline yet fit a single long-format events table: *timestamp_utc* (naive UTC), *lat, lon, metric, value, unit, source*, and *extra_json* (for provenance and *device/product* fields).

OSINT messages arrive as text with local timestamps and geocoded coordinates; FIRMS provides numeric FRP detections in UTC; radiation networks mix dose-rate units and CPM from heterogeneous stations; weather delivers gridded wind direction and speed. The PSAI adapters map each native record to the schema and pass a *tz_hint* where needed so DEH can standardize time and units deterministically (Table 1).

Table 1

Mapping of source-specific fields to the unified events schema

Source	Timestamp (orig)	Lat/Lon (orig)	Metric	Value (unit)	Normalized events schema
OSINT	ISO 8601 (local)	Geocoded text	hit_reported	1 (count)	timestamp_utc, lat, lon, metric='hit_reported', value=1, unit='count', source='osint', extra_json={'msg_id', 'channel'}
FIRMS (MODIS/VIIRS)	Date + time (UTC)	latitude/longitude	frp_*	FRP (MW)	timestamp_utc, lat, lon, metric='frp_*', value=FRP, unit='MW', source='firms', extra_json={'product', 'confidence'}
Radiation (γ)	ISO 8601 (UTC)	station coords	gamma,gamma_cpm	nSv/h or CPM	timestamp_utc, lat, lon, metric, value, unit, source='radiation', extra_json={'station_id', 'device'}
Weather (Wind)	ISO 8601 (UTC)	grid cell center	wind_dir,speed	deg or $m \cdot s^{-1}$	timestamp_utc, lat, lon, metric, value, unit, source='weather', extra_json={'provider'}

Each row in the events table captures a single measurement or event. For instance, an OSINT "*hit_reported*" event is stored with value 1 and unit count, whereas a satellite fire detection includes an FRP value in megawatts.

The radiation feed shows how we normalize units: if a station reports 20 μR/h (microRoentgens per hour), the adapter converts it to approximately 0.175 μSv/h (microsieverts per hour) for consistent gamma dose units. Likewise, wind is split into direction and speed metrics. This uniform schema keeps joins predictable and enables vectorized scans and aggregations inside DuckDB.

Our pipeline's architecture follows a modular ingest and homogenization design. The Plugin Source Adapter Interface (PSAI) allows each data source to implement a small adapter that pulls or receives data and emits normalized event rows. For example, one adapter might call a REST API or read a file and then yield events via a common interface (Python generator of dicts). Each adapter handles source-specific parsing: decoding timestamps, parsing coordinates, and extracting the relevant metrics. This modular approach means we can hot-swap new data sources without altering the core pipeline – just plug in a new adapter that conforms to the interface.

All adapters feed into a central Deterministic Event Homogenization (DEH) stage. The DEH applies consistent rules to every incoming event record.

1. UTC conversion. All timestamps are converted to naive UTC datetime objects. If an event timestamp lacks timezone info, a source-specific hint (provided by the adapter) or default UTC is applied.

2. Unit harmonization. Values are converted to a base unit when necessary (e.g., radiation counts per minute might be converted to dose rates, different radiation units to Sieverts, etc.). This ensures metric and unit pairs are consistent (no mix of "μR/h" vs "nSv/h" in final data).

3. Geo validation. We ensure latitude and longitude are present and within valid ranges. An event without valid coordinates may be dropped or flagged, unless the metric inherently has no location (in our case, we mostly handle spatial data, except some aggregate metrics).

4. Idempotent upsert. Using a natural key (combining source, metric, time, and location), the pipeline checks if an event is already in the database. This prevents duplicate inserts when a source re-sends the same data or when our adapters restart. It is crucial for stable aggregates downstream.

After homogenization, events are inserted into a DuckDB database table (events) and a live view (events_live) that may hold the latest window of data. This design avoids network overhead and leverages DuckDB's vectorized execution for fast in-memory analytics. By embedding DuckDB, we get high performance on analytical queries in a small footprint, similar to how SQLite serves OLTP but here for OLAP workloads (Fig. 1).

On the query side, we implement a lightweight spatio-temporal Command Query Responsibility Segregation (ST-CQRS) pattern. The Command side (ingest) continuously appends events and updates intermediate aggregates, while the Query side reads from pre-computed materialized views for fast responses. Rather than heavy external stream processing frameworks, we leverage DuckDB's ability to maintain materialized tables via SQL.

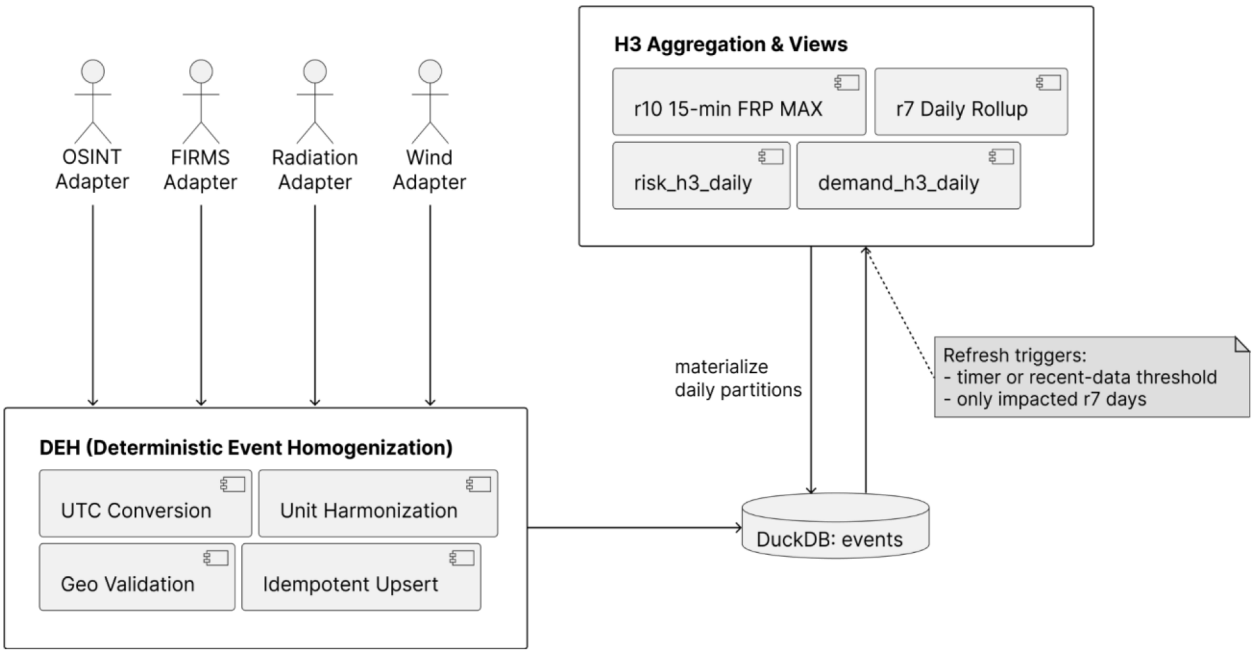

Fig. 1. PSAI-DEH to H3 Risk/Demand (ST-CQRS)

The core of our spatial aggregation is the H3 geospatial index. H3 divides the world into hexagonal cells at varying resolutions (r0 through r15). Each H3 cell has a unique 64-bit index identifying its location and level in the hierarchy. We use resolution 10 for fine-grained "micro-tiles" (~ hexagons with side length ~0.3 km, area ~2 km^2) and resolution 7 for daily summary tiles (~ hexagons ~20 km across, area ~1,200 km^2). An H3 cell at r7 is the "parent" of many r10 cells (specifically, each hexagon splits into 7 children at the next finer resolution in this hexagonal hierarchy. This hierarchy makes it easy to aggregate data: we assign each event to an r10 cell, then roll up to r7 by using H3 parent indexing.

We maintain two main materialized view tables in DuckDB (which we refresh periodically or on new data).

- 15-min H3 r10 micro-tiles: this intermediate table groups events into 15-minute time bins and H3 r10 cells. It is primarily used to deduplicate and aggregate high-frequency data like fire detections.
- Daily H3 r7 views: using the micro-tiles, we aggregate up to daily values on H3 r7 cells. These form the *risk_h3_daily* and *demand_h3_daily* tables, which can be queried or exported.

DuckDB's efficient columnar engine and SQL support make these rollups fast and easy to manage with SQL statements. We chose DuckDB for its embeddability and performance: it can scan and aggregate millions of rows per second in-memory, and it supports direct operations on Parquet files with zero serialization overhead. For our use case (a national-scale dataset updated in real-time on a single machine), DuckDB provides near real-time query capability without the complexity of a distributed database. It essentially allows us to implement the CQRS pattern in-process: writes (commands) affect base and aggregate tables, and reads (queries) hit the pre-computed daily views.

Research results

Time, Units, and Idempotency. All inputs are standardized to naive UTC; if tzinfo is missing, adapters pass a hint before conversion. Units are normalized deterministically (radiation dose rates to sieverts; wind to m/s). A hashed natural key (metric, timestamp_utc, lat, lon, source) enforces idempotent upserts so restarts or late arrivals do not duplicate records. Parquet snapshots persist daily *risk_h3_daily* and *demand_h3_daily* for archival exchange.

FRP Micro-Aggregation (r10 × 15-min). FIRMS detections from overlapping satellites can over-count the same fire. We assign each event to an H3 r10 cell and a 15-minute bin, then take the MAX (FRP) per bin/cell.

This collapses duplicates while preserving peak intensity and matches the VIIRS 375 m footprint used to set r10 granularity (Table 2).

Table 2

DuckDB SQL: r10 15-min FRP dedup (MAX per bin/cell)

```
CREATE OR REPLACE TABLE frp_dedup_15m_r10 AS
WITH s AS (SELECT date_trunc('minute', timestamp)
- INTERVAL (EXTRACT(minute FROM timestamp)::INT % 15) MINUTE AS ts_15m,
h3_latlng_to_cell(lat, lon, 10) AS h3_r10,
value AS frp
FROM events
WHERE metric LIKE 'frp_%'
AND lat IS NOT NULL AND lon IS NOT NULL)
SELECT ts_15m, h3_r10, MAX(frp) AS frp_max
FROM s
GROUP BY ts_15m, h3_r10;
```

Daily Rollup (r7). The micro-tiles roll up by parent using '*h3_cell_to_parent(..., 7)*', and we sum '*frp_max*' by day and r7 cell, r7 balances regional interpretability with compact storage. The resulting '*frp_daily_r7*' table becomes a component in risk (Table 3).

Table 3

DuckDB SQL: r7 daily rollup from r10 micro-tiles

```
CREATE OR REPLACE TABLE frp_daily_r7 AS
SELECT
CAST(ts_15m AS DATE) AS d,
h3_cell_to_parent(h3_r10, 7) AS h3_r7,
SUM(frp_max) AS frp_sum
FROM frp_dedup_15m_r10
GROUP BY d, h3_r7;
```

Risk and Demand Materialization. Daily risk combines OSINT hit counts, log-scaled FRP ('*ln(1+frp_sum)*'), a simple radiation positive proxy, and a wind-alignment proxy. Demand multiplies risk by r7 population.

We keep weights explicit in SQL to ease tuning and audits; joins use integer H3 ids and convert to strings only at export (Table 4).

Table 4

DuckDB SQL: simplified daily risk composition

```
CREATE OR REPLACE TABLE risk_h3_daily AS
SELECT
r.d,
r.h3_r7 AS h3,
COALESCE(h.n_hits, 0) AS n_hits,
COALESCE(r.frp_sum, 0) AS frp_sum,
COALESCE(g.rad_pos, 0) AS rad_pos,
COALESCE(w.wind_align, 0) AS wind_align,
0.5 * COALESCE(h.n_hits, 0)
+ 0.3 * ln(1 + COALESCE(r.frp_sum, 0))
+ 0.1 * COALESCE(g.rad_pos, 0)
+ 0.1 * COALESCE(w.wind_align, 0) AS risk_score
FROM frp_daily_r7 r
LEFT JOIN hits_daily_r7 AS h USING (d, h3_r7)
LEFT JOIN gamma_daily_h3 AS g ON (g.d = r.d AND g.h3 = r.h3_r7)
LEFT JOIN wind_daily_h3 AS w ON (w.d = r.d AND w.h3 = r.h3_r7);
```

Measured Behavior. On an M2 laptop, end-to-end updates are typically sub-second; FRP dedup reduces raw points by ~99 %; '*risk_h3_daily*' materialization reaches tens of thousands of rows per second. Indicative values are summarized below (Table 5).

Table 5

Engineering metrics (measured)

Metric	Scope	Value (median)	p95	Notes
Ingest → MV latency (24h window)	end-to-end update	0.089 s	0.923 s	sub-second typical
Ingest → MV latency (heavy days)	ranked by FRP ∪ Hit tiles	0.0303 s	0.0366 s	warm path
FRP dedup reduction	r10 × 15-min MAX(FRP)	0.8–0.9 %	–	~99 % removed
MV throughput (risk_h3_daily)	rows/s (examples)	31.9–55.3k	–	five heavy days
Storage footprint (daily)	Parquet exports	~0.0–0.1 MB	–	sub-megabyte footprint (typically tens of kilobytes)

Conclusions and future work

This research shows that a small, deterministic pipeline – PSAI ingest, DEH normalization, and H3 aggregation in DuckDB – yields daily risk and demand layers with seconds-level latency on a single machine. The design favors reproducibility and clarity: adapters emit the same shape of events; DEH enforces UTC and units; aggregates are expressed in SQL; and storage remains compact. The evaluation confirms low-latency updates, extreme FRP deduplication, and high materialization throughput, which together are sufficient for national-scale monitoring on laptop-class hardware. Future improvements are straightforward. Late and corrected data can be handled with short grace windows on micro-tiles, followed by targeted refresh of impacted r7 days. Radiation harmonization should move beyond fixed factors toward sensor-specific calibration curves and cross-validation against regional baselines. Wind influence can progress from a proxy to directional plume modeling that links sources to downwind population. Network travel times, built with OSMnx and Valhalla, can transform demand into "reachable population" and support logistics decisions under constraints. Solar-angle adjustments for FRP and more robust anomaly detection will raise signal quality without changing the core interfaces. Incremental refresh strategies and lightweight change data capture for 'events' can reduce compute even further as volumes grow – all while preserving the deterministic contracts that make the system easy to operate and extend [1–9].

References

1. Raasveldt M., Mühleisen H. (2019). DuckDB: an in-process analytical database. *Proceedings of the 2019 ACM SIGMOD International Conference on Management of Data* (Demo). URL: https://duckdb.org (access date: 03.09.2025).

2. Uber Engineering. (2018). H3: Uber's Hexagonal Hierarchical Spatial Index. URL: https://www.uber.com/blog/h3/ (access date: 03.09.2025).

3. NASA LANCE. (2023). Fire Information for Resource Management System (FIRMS). URL: https://firms.modaps.eosdis.nasa.gov (access date: 03.09.2025).

4. Schroeder W., Oliva P., Giglio L., Csiszar I. (2014). The VIIRS 375 m active fire detection data suite. *Remote Sensing of Environment*, vol. 143, pp. 85–96. https://doi.org/10.1016/j.rse.2013.12.008 (access date: 03.09.2025).

5. European Commission. EURDEP – European Radiological Data Exchange Platform. URL: https://rem.jrc.ec.europa.eu (access date: 05.09.2025).

6. SaveEcoBot. Radiation map & data access. URL: https://www.saveecobot.com/en (access date: 03.09.2025).

7. OpenWeather. OpenWeatherMap API. URL: https://openweathermap.org/api (access date: 03.09.2025).

8. Boeing G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139. https://doi.org/10.1016/j.compenvurbsys.2017.05.004 (access date: 03.09.2025).

9. Valhalla. Open Source Routing Engine. URL: https://github.com/valhalla/valhalla / (access date: 05.09.2025).

Дата першого надходження рукопису до видання: 24.09.2025
Дата прийнятого до друку рукопису після рецензування: 21.10.2025
Дата публікації: 28.11.2025