International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

ECHO-DR: An Event-Centric Hierarchical Orchestration Architecture for Scalable AI Workflows in Real-Time Disaster Response

Prudvi Saisaran Ponduru
Independent Researcher

Abstract- — Scalable artificial intelligence (AI) workflows increasingly fail not because individual models are weak, but because the surrounding architecture cannot process heterogeneous, bursty, high-stakes evidence at operational speed. This paper proposes ECHO-DR, an Event-Centric Hierarchical Orchestration architecture for real-time disaster response. The real-world problem addressed is the difficulty of turning social media, remote sensing, UAV imagery, weather alerts, seismic feeds, and incident reports into timely, auditable, and trustworthy operational intelligence during floods, earthquakes, wildfires, and storms. ECHO-DR introduces four core contributions: an event-centric memory plane that unifies vector retrieval, geospatial indexing, lakehouse lineage, and structured event graphs; a hierarchical routing policy that escalates only high-value or uncertain items to expensive multimodal reasoning; a stage-disaggregated serving design that independently scales encoders, prefill workers, decoders, and tool calls; and a governance plane that embeds auditability, human review, and zero-trust access control into the workflow. A formal utility-constrained routing model, event-linking algorithm, fusion rule, and capacity model are developed to show how the architecture scales under large workflows. The paper also provides an implementation blueprint, clean system diagrams, benchmarking methodology, ablations, and simulated evaluation results. Simulated trace-driven experiments indicate that the proposed gated architecture can reduce p95 provisional alert latency relative to a monolithic multimodal pipeline while maintaining evidence traceability and limiting deep-model cost. The work demonstrates that scalable AI for future big workflows should be designed as a compound, event-centered, policy-aware system rather than as a single model endpoint.

Keywords: Scalable AI workflows, multimodal disaster response, event-centric architecture, AI orchestration,

I. INTRODUCTION

The future of AI will be defined not only by larger models, but also by the workflow architectures that decide when, where, and why those models are used. Modern AI systems are increasingly compound systems: a user-visible result may involve retrieval, classification, planning, tool invocation, multimodal encoding, large language model (LLM) decoding, database updates, human review, and post-deployment learning. When such systems are deployed in ordinary commercial settings, inefficiency causes cost inflation and latency. When they are deployed in emergency management, inefficiency may delay life-saving decisions. This paper addresses the latter case by designing a scalable AI workflow architecture for real-time disaster response.

Disaster response is an appropriate stress test for future AI workflows because it combines large data volume, many modalities, strict latency constraints, uncertainty, and high consequence. A flood, wildfire, earthquake, or hurricane may produce simultaneous evidence from weather and seismic sensors, public social posts, satellite passes, unmanned aerial vehicles, emergency call centers, field responders, and official geospatial products. Each evidence stream has different reliability, arrival rate, spatial precision, and update frequency.

A conventional batch analytics pipeline is too slow for first response, while a monolithic multimodal model endpoint is too costly and opaque for continuous operation. The architectural problem is therefore to produce early partial truth quickly, revise it as stronger evidence arrives, and preserve evidence lineage for operators.

The proposed system is called ECHO-DR, which stands for Event-Centric Hierarchical Orchestration for Disaster Response. ECHO-DR treats disaster intelligence as a stream of evolving events rather than as a collection of independent files or isolated predictions.

The architecture is designed around the insight that the unit of operational value is not an image, tweet, video, sensor reading, or report. The unit of value is an event: a flood crossing a road, a collapsed structure, a blocked bridge, an evacuation bottleneck, a shelter demand spike, or a fire perimeter change. Event-centric design allows evidence from

multiple modalities to update a shared state with confidence, time, location, provenance, and operator feedback.

The paper makes five contributions. First, it defines a real-world AI workflow problem in which the objective is to maximize timely decision utility under bounded cost and uncertainty. Second, it proposes a novel architecture with four planes: ingestion, decision, memory, and governance. Third, it provides formal models for routing, event linking, fusion, and capacity. Fourth, it demonstrates a clear implementation path using cloud-native orchestration, streaming analytics, model serving, vector retrieval, geospatial indexing, and Lake House storage. Fifth, it presents an evaluation framework and simulated results that compare ECHO-DR against a monolithic multimodal baseline.

The paper is written as a submission-ready scholarly manuscript. Its empirical section is explicitly framed as simulation and design-level evaluation rather than a claim of field deployment. This distinction is important. In high-stakes AI, inventing unperformed experiments would be less rigorous than openly reporting a reproducible evaluation plan and simulated workload. The goal is to establish architecture, formalize its scaling behavior, and provide enough implementation detail for future empirical validation on public datasets and operational traces.

Motivating Problem

Emergency operations centers need rapid situational awareness. A single hazard may require decisions about evacuations, road closures, shelter placement, search-and-rescue prioritization, power restoration, medical logistics, and public warnings. However, evidence arrives with inconsistent structure.

Satellite imagery may provide broad coverage but may be delayed or obscured by clouds; synthetic aperture radar can see through clouds but requires specialist processing; social media can arrive immediately but is noisy and unevenly distributed; official reports may be reliable but delayed; and sensor feeds may be fast but incomplete. The AI workflow must therefore handle asynchronous, heterogeneous, and partially conflicting evidence.

A naive architecture would route every item through the strongest available multimodal model. This strategy is attractive in demonstrations but fails in production. Deep multimodal models are expensive, have variable latency, and may become the bottleneck during bursts. Many items do not need deep reasoning: duplicates, low-relevance posts, ordinary weather updates, and already-confirmed imagery can be filtered or summarized by cheaper models. Conversely, a low-cost-only pipeline is unsafe because high-consequence or uncertain events require deeper reasoning and human review. The core question is how to allocate expensive AI reasoning to the evidence items for which it creates the highest operational value.

ECHO-DR answers this question with hierarchical routing. Evidence first passes through normalization and low-cost triage. Only items with high criticality, high uncertainty, or high expected information gain are escalated to cross-modal fusion or deep multimodal reasoning. All outputs update the event memory plane, so the system produces continuous revision rather than one-shot conclusions. This approach is designed to preserve emergency response service-level objectives while controlling cost and maintaining auditability.

Fig. 1 Overall Architecture

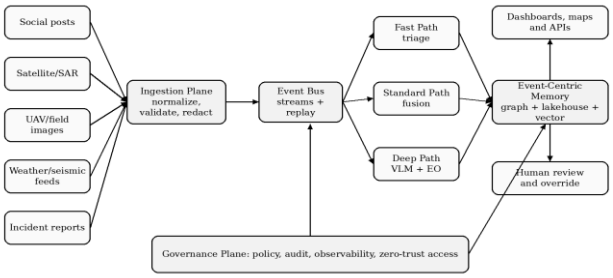

Fig.1. Overall ECHO-DR architecture showing ingestion, hierarchical inference paths, event memory, governance, human review, and continuous evaluation.

II. BACKGROUND AND RELATED WORK

The disaster AI literature spans social sensing, remote sensing, computer vision, natural language processing, human-computer interaction, and emergency management. CrisisMMD established the value of multimodal social media data for classifying informative disaster posts, humanitarian categories, and damage severity [1]. xBD and the xView2 challenge advanced satellite-based building damage assessment with pre-event and post-event imagery [2], [3].

FloodNet and RescueNet extended the benchmark landscape to high-resolution aerial imagery and UAV semantic segmentation [4], [5]. Recent datasets such as BRIGHT and DisasterM3 add multimodal remote sensing, optical-SAR fusion, and vision-language tasks for disaster assessment [6], [7]. EIDSeg adds pixel-level ground-view damage segmentation from social media images [8].

These datasets show two important facts. First, no single modality is sufficient. Satellite images are strong for regional structure damage, UAV images are strong for local context, social posts provide rapid eyewitness evidence, and sensor feeds provide continuous physical measurements. Second, the workflow problem is broader than model accuracy. In the field, a high-scoring model is useful only if it can be deployed in a system that filters, aligns, updates, explains, and escalates evidence under bursty load. Recent reviews of social media disaster management identify persistent limitations in relevance filtering, cross-platform integration, location extraction, and conversion of unstructured content into actionable intelligence [9].

Reviews of remote-sensing damage assessment identify related limitations: real-time throughput, data fusion, generalization across hazards and geographies, and operational integration with rescue workflows [10]. These findings motivate an architecture in which data fusion, event memory, and human verification are central rather than peripheral.

The closest recent architectural direction is retrieval-augmented crisis sensing. CrisiSense-RAG combines aerial imagery, social media, calls, precipitation data, and historical priors for rapid disaster impact estimation [11]. It is important because it recognizes that evidence-linked reasoning is required for crisis response. ECHO-DR generalizes this idea into full workflow architecture with online scheduling, stage-disaggregated serving, event-centric state, policy-aware escalation, and governance.

The model-serving literature provides another foundation. Efficient multimodal serving systems increasingly disaggregate resource-intensive stages such as encoding, prefill, and decoding. EPD shows that separating encode, prefill, and decode stages can significantly improve time-to-first-token and resource utilization [12].

ModServe extends the idea with modality-aware and stage-aware resource disaggregation for scalable multimodal model serving [13]. Production-oriented compound AI studies emphasize that real deployments require model composition, tool calls, retrieval, and routing rather than isolated endpoints [14]. ECHO-DR adapts these systems insights to disaster response, where routing decisions must optimize mission value and not only throughput.

Cloud-native tools make the architecture implementable. KServe InferenceGraph supports declarative multi-stage inference graphs, conditional routing, and independent scaling of services [15]. Ray Serve supports model composition and autoscaling for complex online applications [16]. Spark Structured Streaming provides stateful stream processing and replay [17]. Apache Iceberg supports open table formats and schema evolution [18]. OpenTelemetry provides shared tracing, metrics, and logging across services [19]. These tools do not by themselves constitute a disaster AI architecture, but they provide the substrate on which ECHO-DR can be built.

Gap in Existing Architectures

Existing AI workflow architectures usually optimize one of three goals: offline accuracy, online inference throughput, or developer convenience. Disaster response requires a fourth goal: auditable decision utility under time pressure. A system that maximizes throughput but hides evidence lineage is unsafe. A system that preserves evidence but cannot update quickly is operationally weak. A system that applies large models uniformly to every item is economically fragile. A system that filters aggressively without uncertainty escalation risks missing rare but critical events.

Table I: Representative Related Work and Architectural Implication

Area	Representative Sources	Main Contribution	Architectural Implication for ECHO-DR
Multimodal crisis social sensing	CrisisMMD and subsequent crisis informatics work [1], [9]	Shows that social text and images contain useful disaster signals but are noisy and uneven.	Use low-cost relevance filtering, source scoring, and human escalation.
Remote-sensing damage assessment	xBD, xView2, FloodNet, RescueNet, BRIGHT, DisasterM3 [2]-[7]	Provides satellite, SAR, aerial, and vision-language benchmarks for damage assessment.	Treat imagery as one evidence stream inside event fusion rather than the entire workflow.
Retrieval-augmented crisis reasoning	CrisiSense-RAG [11]	Links multimodal evidence and retrieval to impact estimation.	Extend retrieval from task-specific reasoning to persistent event-centric memory.
Multimodal model serving	EPD, ModServe [12], [13]	Demonstrates benefits of stage and modality disaggregation.	Scale encoders, prefill, decoders, and tool calls independently.
Production AI orchestration	KServe, Ray Serve, Spark, Iceberg, OpenTelemetry [15]-[19]	Provides practical serving, streaming, storage, and observability mechanisms.	Compose components into a governed, auditable disaster workflow.

The gap is therefore a system design gap. What is needed is an architecture that combines event-centric state, selective deep reasoning, streaming dataflow, autoscaled serving, and governance. ECHO-DR is proposed as such architecture.

It is not a replacement for disaster-specific models; rather, it is a workflow layer that decides how those models, data stores, human reviewers, and operational products are composed.

III. PROBLEM STATEMENT AND REQUIREMENTS

Let a disaster-response region produce a stream of evidence items X = {x1, x2, ..., xn}. Each item may be text, image, video, sensor record, geospatial feature, or incident report. Items arrive out of order, may contain duplicates, may be uncertain, and may contradict one another. The system must map these items into evolving event states E = {E1, E2, ..., Ek}.

Each event state contains location, time, hazard type, affected assets, severity, confidence, supporting evidence, model outputs, human judgments, and recommended actions. The system must do this while respecting latency, budget, privacy, and reliability constraints.

The central objective is not merely to maximize classification accuracy. The objective is to maximize timely decision utility. A provisional alert that is 90% reliable and arrives in 20 seconds may be more valuable than a 97% reliable report that arrives after a road closure or evacuation decision has already been made.

Conversely, for high-consequence actions, speed alone is insufficient; the system must escalate uncertain claims to deeper models or human review. ECHO-DR therefore treats routing as an optimization problem under cost, latency, and risk constraints.The architecture is required to process large workflows.

In a severe regional event, millions of sensor updates, posts, tiles, and reports may arrive over hours or days. Not every item should enter expensive reasoning. The system must preserve replayability for post-event audit and training, and it must support schema evolution because event taxonomies, models, data providers, and response policies will change over time.

Functional Requirements

The functional requirements are as follows. The system must ingest heterogeneous sources through a common event envelope. It must filter relevance and priority quickly. It must assign items to events using spatiotemporal and semantic signals. It must fuse evidence across modalities.

It must produce outputs useful to emergency operators, including maps, alert cards, confidence scores, and summaries. It must allow human verification and correction. It must provide retrieval over raw and derived evidence. It must support continuous evaluation and model improvement.

A practical architecture should handle at least three operating modes. In normal mode, it continuously monitors and updates low-rate events. In burst mode, it prioritizes mission-critical items while deferring noncritical backfill. In degraded mode, it continues to provide partial alerts even when deep GPU pools, vector indexes, or cloud links are unavailable. These modes make graceful degradation part of the architecture instead of a post hoc operational procedure.

Nonfunctional Requirements

Nonfunctional requirements include latency, scalability, cost control, traceability, privacy, security, and maintainability. Latency must be measured at several levels: item ingestion latency, provisional alert latency, fused event update latency, and final report generation latency.

Scalability must include data scaling, inference scaling, memory scaling, and operator workload scaling. Cost must be bounded by selective escalation, caching, batching, and autoscaling. Traceability requires every high-consequence output to link back to source evidence and model versions. Security requires zero-trust access, least privilege, encryption, and audit logs.

Maintainability requires modular components and open interfaces so that models and infrastructure can evolve.The paper uses the following target service-level objectives for design evaluation: p95 provisional alert latency below 30 seconds for social/sensor evidence; p95 fused event update latency below 2 minutes; macro-F1 above 0.80 for severe-damage classification tasks in offline benchmarks; event cluster purity above 0.90 under trace replay; and no evidence loss under single-node failure. These are proposed operational targets, not field deployment claims.

Table II: ECHO-DR Design Requirements

Requirement	Description	Design Mechanism
Heterogeneous ingestion	Accept text, imagery, video, sensor feeds, reports, and geospatial products.	Common event envelope, validation, lakehouse bronze layer.
Low latency	Generate first alerts before slower high-resolution analysis finishes.	Fast path triage and priority queues.
Selective deep reasoning	Use expensive models only when expected value justifies cost and delay.	Utility-aware routing and uncertainty thresholds.
Event continuity	Maintain a coherent evolving incident state across time and modalities.	Event graph, geospatial index, vector memory, lineage ledger.
Auditability	Show evidence, model versions, confidence, and operator decisions.	Governance plane and immutable provenance records.
Burst scalability	Handle sudden increases in evidence volume without tail-latency collapse.	Stage-disaggregated serving, autoscaling, backpressure, graceful degradation.
Human control	Escalate high-consequence or ambiguous cases to verified review.	Human-in-the-loop queue and override feedback.

IV. PROPOSED ARCHITECTURE

ECHO-DR is structured as four interacting planes: ingestion, decision, memory, and governance. The ingestion plane converts heterogeneous raw inputs into validated event envelopes. The decision plane performs hierarchical routing and inference. The memory plane persists raw evidence, derived features, event states, semantic embeddings, geospatial indexes, and lineage. The governance plane enforces access control, policy, auditability, observability, and human escalation.

The architecture is event-centric. Instead of designing separate pipelines for satellite images, social posts, and sensor alerts, ECHO-DR converts all relevant evidence into event updates. Each event has a durable state that can be queried, revised, explained, and replayed.

This design supports the operational reality that disaster knowledge changes continuously. A road may initially be reported as flooded by a social post, later confirmed by a field image, then revised when water recedes or a road crew clears it. The system must preserve this sequence rather than overwrite it with a single final answer.

The decision plane contains three inference paths. The fast path performs low-cost filtering, geolocation, duplicate suppression, and coarse severity estimation. The standard path performs cross-modal feature fusion, event linking, and evidence reconciliation.

The deep path performs expensive vision-language reasoning, remote-sensing analysis, segmentation, change detection, and report generation. Human review is connected to the deep path but is also triggered by policy rules, high consequence, low confidence, or conflicting evidence.

The memory plane is not a passive database. It is the coordination substrate of the architecture. Structured geospatial queries, semantic retrieval, historical replay, lineage lookup, and event graph traversal all depend on this plane. A dashboard alert can therefore expose not only a generated summary but also the exact source posts, images, model outputs, timestamps, and operator decisions that led to the alert. This evidence-linked design is essential for trust and accountability.

Ingestion Plane

The ingestion plane accepts input from official APIs, message queues, file drops, object stores, social platforms, field applications, and remote-sensing providers. Each item is transformed into a normalized evidence envelope containing modality, source, timestamp, geospatial footprint, raw reference, extracted metadata, content hash, privacy class, and initial reliability estimate. The envelope format allows the downstream workflow to treat heterogeneous modalities uniformly while preserving modality-specific payloads.

Data minimization is applied at ingestion. Personally identifiable information is redacted or tokenized where it is not operationally required. Exact coordinates may be coarsened for public products while retained under restricted access for authorized operators.

Media files are stored immutably in a raw object store, while normalized metadata enters the stream and lakehouse. This

separation supports audit and replay without forcing every model-serving service to move large media objects.

Decision Plane

The decision plane is the main source of efficiency. It transforms evidence into action through a hierarchy of increasingly expensive steps. The fast path is optimized for high throughput and low latency. It can be implemented with small language models, lightweight image classifiers, rule-based geospatial filters, metadata checks, duplicate hashing, and compact embedding models. The standard path joins evidence across time windows and spatial buckets, retrieves semantically related items, and updates event posteriors. The deep path uses large vision-language models, segmentation models, geospatial foundation models, and tool-augmented reasoning only when justified.

The decision plane also implements priority. A low-confidence report of a collapsed bridge near an evacuation route receives a higher priority than many routine weather updates. A low-severity duplicate is archived. A high-severity but poorly localized report may trigger human review or deeper geospatial search. The routing policy is therefore tied to operational utility rather than fixed model sequence.

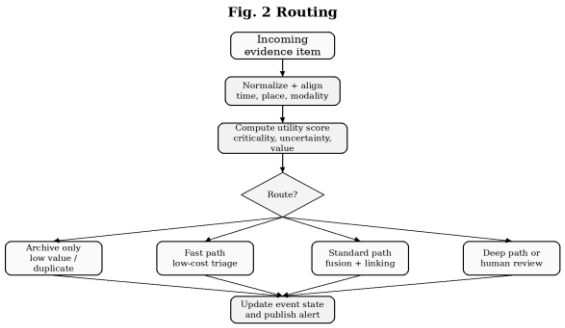

Fig. 2. Hierarchical routing and escalation decision flow. Items move to deeper reasoning or human review only when uncertainty, criticality, or expected information gain justifies the cost.

Memory Plane

The memory plane contains four interacting stores. The raw object store preserves original evidence. The lakehouse stores bronze, silver, and gold tables for normalized inputs, cleaned features, and curated event products. The vector index stores embeddings for semantic and multimodal retrieval. The event graph stores events, assets, locations, roads, shelters, hazards, source entities, and relationships. A lineage ledger records model versions, prompts, parameters, timestamps, operator actions, and policy decisions. This structure enables three kinds of retrieval. Structured retrieval answers questions such as which roads within a polygon have severe flood evidence. Semantic retrieval answers questions such as which reports resemble this collapsed-building description. Temporal retrieval reconstructs how knowledge changed over time. The combination is necessary because disaster decisions depend on spatial, semantic, and temporal evidence simultaneously.

Governance Plane

The governance plane is part of the core design rather than a compliance add-on. It enforces least-privilege access to event data, role-based views, signed model artifacts, immutable audit trails, monitoring, alerting, and policy checks. High-consequence recommendations are never presented as unsupported language-model conclusions. They are presented as confidence-aware products with evidence links and review status. The governance plane also manages evaluation. Every deployed model and route is traceable to a registry version. Drift, calibration, false positives, missed events, operator overrides, latency, and cost are measured continuously. The same event memory that supports operations also supports post-event analysis and training data curation.

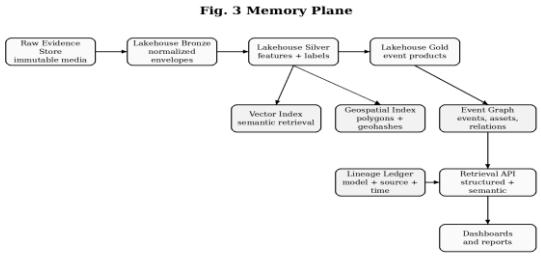

Fig. 3. Event-centric memory plane. Raw evidence, lakehouse tables, vector retrieval, geospatial indexing, event graph state, and lineage jointly support auditable operational outputs.

V. FORMAL SYSTEM MODEL

This section formalizes the architecture to show where scalability and efficiency arise. Let each incoming evidence item be represented as xi = (mi, ti, gi, ci, zi, ri), where mi is modality, ti is timestamp, gi is geospatial footprint, ci is content or embedding, zi is source metadata, and ri is raw-object reference. The system maps evidence items into event states Ek. Each state contains a posterior distribution over event classes, severity estimates, confidence, evidence links, and recommended actions.

$$xi = (mi, ti, gi, ci, zi, ri) \qquad (1)$$

The routing policy pi chooses an action from the set {archive, fast, standard, deep, human-review}. The action determines which computational path is used. The optimal routing policy

maximizes expected operational utility while penalizing cost, latency, and residual risk. This formulation reflects the fact that the value of inference depends on context. A high-latency deep model may be justified for a possible mass-casualty report, but not for a routine duplicate observation.

$$\max_{\pi} E[\sum_i U_i(a_i) - \lambda_c C_i(\pi) - \lambda_l \max(0, L_i(\pi) - \tau_i) - \lambda_r R_i(\pi)] \quad (2)$$

In Equation (2), U_i is operational utility, C_i is compute cost, L_i is latency, τ_i is the deadline associated with the evidence item or event, and R_i is residual risk due to uncertainty, conflict, or missing evidence. The coefficients λ_c, λ_l, and λ_r encode organizational preferences. During a severe incident, the operator may increase the penalty on latency and residual risk. During monitoring mode, the operator may increase the penalty on cost.

The routing score is computed from estimated information gain, action value, uncertainty, predicted cost, and predicted latency. ECHO-DR escalates an item when the numerator, which represents expected operational value, outweighs the denominator, which represents resource burden.

$$\Delta_i = (\beta_1 IG_i + \beta_2 A_i + \beta_3 u_i) / (\alpha_1 c_i + \alpha_2 l_i + \varepsilon) \quad (3)$$

Here IG_i is estimated information gain from deeper analysis, A_i is estimated action value, u_i is uncertainty, c_i is predicted extra cost, l_i is predicted extra latency, and ε avoids division by zero.

A thresholded version of Δ_i selects the route. Unlike a fixed cascade, the score is dynamic. It can include hazard type, population exposure, proximity to critical infrastructure, operator workload, GPU queue length, and confidence calibration.

Event linking combines geospatial, temporal, and semantic evidence. Candidate events are first retrieved by spatial and temporal buckets. Semantic retrieval then returns the nearest event descriptions or evidence embeddings.

A linking score combines distance, time gap, semantic similarity, source reliability, and conflict penalties. If the best score exceeds a threshold, the item updates an existing event; otherwise, a new event is created.

$$\text{Link}(x_i, E_k) = w_g S_{geo}(g_i, g_k) + w_t S_{time}(t_i, t_k) + w_s S_{sem}(c_i, c_k) + w_z S_{src}(z_i) - w_q Q_{conflict} \quad (4)$$

Evidence fusion is expressed as a log-linear posterior update. Let $p_c^{(m)}(E_k)$ be the probability assigned by modality m to class c for event k, and let w_m be a calibrated reliability weight. Structured geospatial and temporal priors are represented by ϕ_{geo} and ϕ_{time}. The event posterior is computed by normalizing the fused scores across classes.

$$s_c(E_k) = \sum_m w_m \log p_c^{(m)}(E_k) + \gamma \phi_{geo}(E_k) + \delta \phi_{time}(E_k) \quad (5)$$

$$P(c \mid E_k) = \text{softmax}(s_c(E_k)) \quad (6)$$

The uncertainty of an event can be defined as one minus the maximum posterior probability, or by entropy when multi-class ambiguity matters. The architecture uses uncertainty for escalation and review. High uncertainty alone is not sufficient; it is combined with criticality so that the system does not waste deep reasoning on unimportant ambiguity.

$$u(E_k) = 1 - \max_c P(c \mid E_k) \quad (7)$$

The scaling model shows why selective escalation matters. Let λ be the incoming evidence arrival rate, p_f, p_s, and p_d be the fractions routed to fast, standard, and deep paths, and c_f, c_s, c_d be average cost per item.

Expected cost per unit time is proportional to the weighted route mixture. Because c_d is much greater than c_f, the deep-route fraction p_d is the dominant cost control variable.

$$E[C] = \lambda (p_f c_f + p_s c_s + p_d c_d) \quad (8)$$

Capacity stability can be stated using service rates. If μ_f, μ_s, and μ_d are service rates for each tier, and n_f, n_s, and n_d are replica counts, then the arrival rate into each tier must remain below available service capacity. When queue pressure rises, ECHO-DR can scale replicas, change route thresholds, defer noncritical work, or produce partial outputs.

$$\lambda p_f < n_f \mu_f, \quad \lambda p_s < n_s \mu_s, \quad \lambda p_d < n_d \mu_d \quad (9)$$

The formal model emphasizes the paper's main thesis. Scalability does not come only from adding GPUs. It comes from controlling which items require expensive inference, separating stages with different resource profiles, maintaining event memory to prevent repeated work, and letting operators tune the policy according to mission context.

Algorithmic Descriptions

Algorithm 1: Utility-Aware Routing

```
Input: evidence item x_i, event candidates C, system state S
1: normalize x_i and compute modality-specific features
2: estimate criticality A_i from hazard, location, asset exposure, and source
3: estimate uncertainty u_i from calibrated fast-path models
4: estimate information gain IG_i from candidate conflicts and missing evidence
5: estimate marginal cost c_i and latency l_i from current queue and model state
6: compute Delta_i = (beta_1 IG_i + beta_2 A_i + beta_3 u_i)/(alpha_1 c_i + alpha_2 l_i + epsilon)
7: if relevance is below threshold, archive with lineage
8: else if Delta_i is low, use fast path
9: else if cross-modal evidence is required, use standard path
10: else use deep path or human review according to policy
Output: route decision and priority score
```

Algorithm 2: Event Linking and State Update

```
Input: evidence item x_i, event memory M
1: retrieve spatial candidates by geohash and time window
2: retrieve semantic candidates by vector similarity
3: merge candidates and compute Link(x_i, E_k) for each candidate
4: if max score exceeds threshold, assign x_i to best event E_k
5: otherwise create new event E_new
6: update event posterior, uncertainty, lineage, and evidence list
7: if uncertainty or consequence crosses threshold, enqueue review or deep path
Output: updated event state and operational product trigger
```

Fig. 4 Stage Disaggregation

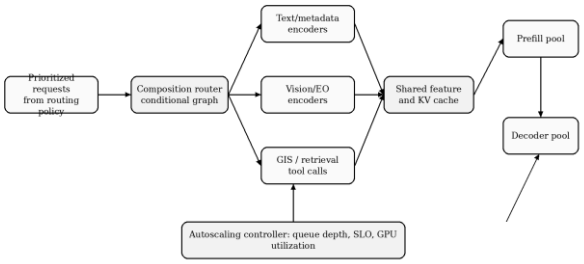

Fig. 4. Stage-disaggregated multimodal serving architecture. Encoders, prefill workers, decoders, tool calls, and caches are scaled independently to reduce tail latency under bursty workflows.

VI. WORKFLOW DEMONSTRATION

This section demonstrates ECHO-DR through a realistic flood-response scenario. Assume heavy rainfall causes rapid flooding in a metropolitan region. The National Weather Service issues alerts, hydrological sensors report rising water levels, citizens post images of flooded roads, emergency calls report stranded vehicles, and satellite or UAV imagery arrives later. The challenge is to identify actionable events quickly while preserving evidence quality.

At time t0, a weather alert and river gauge readings enter the ingestion plane. They are structured sources, so the system creates a regional flood-watch event with moderate confidence. At t1, multiple social media posts mention a flooded underpass near a commuter route.

The fast path extracts place names, compares them to geospatial assets, deduplicates reposts, and estimates high relevance. Because the underpass is near a critical route, the action value score is high. The system creates a provisional alert with source links and confidence, rather than waiting for satellite confirmation.

At t2, a 311 report and two field images arrive. The standard path links them to the same event using geospatial proximity and semantic similarity. The event posterior changes from possible to probable flooding, and the system recommends road-closure verification. At t3, a UAV image arrives.

The deep path uses segmentation and vision-language reasoning to estimate water coverage and vehicle obstruction. Because the event is high consequence, the output is sent to a human reviewer. The reviewer confirms the blockage and

updates the event state. Operators then see a map layer, timeline, evidence list, and recommended public communication.

This demonstration illustrates how ECHO-DR supports continuous evidence accumulation. The system does not wait for the strongest evidence before acting, but it does not treat weak evidence as final. Early alerts are provisional. Later evidence revises confidence. Human review is attached to high-consequence decisions. Every output remains linked to its evidence trail.

A monolithic alternative would either route all items through a deep model or delay action until all evidence is available. The first approach overloads compute during bursts; the second loses operational time. ECHO-DR avoids both extremes by using tiered inference and event memory.

Fig. 6 Sequence Workflow

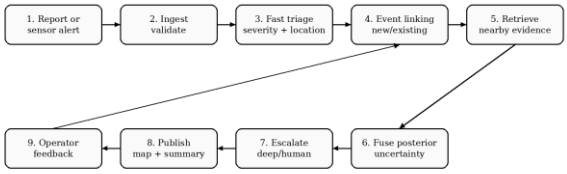

Fig. 5. End-to-end workflow sequence for a flood event. The event state is continuously updated as new sensor, social, image, and operator evidence arrives.

VII. IMPLEMENTATION BLUEPRINT

A practical implementation of ECHO-DR can be built with existing cloud-native and AI-serving components. The purpose of this blueprint is not to prescribe a single vendor stack, but to show that the architecture is implementable today.

The ingestion bus can be Apache Kafka, Google Pub/Sub, AWS Kinesis, or another durable event system. Stream processing can be performed with Spark Structured Streaming or Flink. The lakehouse can use object storage with Apache Iceberg tables. Online inference can use KServe or Ray Serve.

GPU inference can use Triton, vLLM, TensorRT-LLM, or equivalent engines depending on model type. Vector retrieval can use OpenSearch, Milvus, Weaviate, pgvector, or another vector database. Observability can use OpenTelemetry.

The first implementation milestone should be a minimal vertical slice. The system ingests a small set of public feeds, normalizes event envelopes, runs fast-path relevance models, creates event stubs, and displays them on an operator dashboard.

The second milestone adds event memory, vector retrieval, and geospatial linking. The third milestone adds remote-sensing imagery, deep-path models, and human review. The fourth milestone adds replay-based evaluation, fault injection, privacy controls, and active learning.

The model stack should be modular. Lightweight classifiers and embedding models should handle the fast path. Remote-sensing foundation models, open-set object detectors, segmentation models, and vision-language models should be reserved for the standard and deep paths.

Examples include geospatial foundation models such as Prithvi-EO-2.0 [29], open-set detectors such as Grounding DINO [25], segmentation models such as SAM 2 [26], and vision-language models such as Qwen2.5-VL [27].

Dense and sparse multilingual retrieval can use modern embedding models such as M3-Embedding [28]. The architecture does not require these exact models; it requires replaceable model interfaces and calibrated outputs.

Deployment should be hybrid. Edge or field nodes can perform local caching and low-cost prefiltering when network connectivity is weak. Regional clusters can serve low-latency inference and geospatial dashboards. Core cloud or sovereign data centers can handle deep GPU workloads, historical lakehouse storage, replay, and training. The topology is intentionally portable across public cloud, private cloud, and sovereign infrastructure.

Fig. 5 Deployment Topology

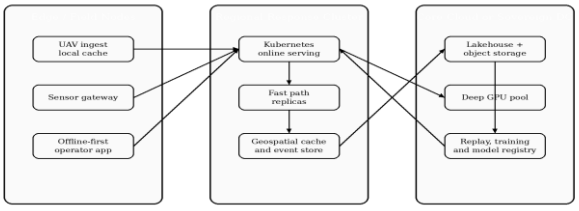

Fig. 6. Hybrid edge-regional-core deployment topology. The architecture can continue local triage at the edge, serve operations regionally, and use core GPU/lakehouse resources for deep reasoning and replay.

VIII. EVALUATION METHODOLOGY

ECHO-DR should be evaluated at two levels: model quality and workflow quality. Model quality measures whether individual predictions are correct. Workflow quality measures whether the architecture produces timely, affordable, traceable, and useful event updates under realistic load. Many AI papers report only accuracy; this is insufficient for disaster response. An architecture can have strong classifiers and still fail operationally if its p95 latency collapses during bursts or if operators cannot inspect evidence.

Table III: Candidate Implementation Stack

Layer	Open or Cloud-Native Options	Role in ECHO-DR
Ingestion and messaging	Kafka, Pub/Sub, Kinesis, MQTT gateways	Durable streams, replay, ordered event envelopes, backpressure.
Stream processing	Spark Structured Streaming, Flink	Windowed joins, geospatial enrichment, feature updates, stream-to-lakehouse writes.
Serving graph	KServe InferenceGraph, Ray Serve	Conditional routing, model composition, autoscaling, online decision graphs.
GPU inference	Triton, vLLM, TensorRT-LLM	Batching, decoder serving, VLM/LLM acceleration, stage-aware pools.
Memory and storage	Object storage, Apache Iceberg, event graph database	Raw artifacts, curated tables, schema evolution, event relationships.
Retrieval and search	OpenSearch vector search, Milvus, pgvector	Semantic retrieval, near-duplicate detection, multimodal evidence lookup.
Observability	OpenTelemetry, Prometheus, Grafana, log pipelines	Metrics, traces, lineage, SLO monitoring, incident debugging.
Governance	Model registry, IAM, policy engine, audit ledger	Access control, model versioning, approval workflows, immutable logs.

The evaluation suite should use public datasets and trace-driven simulation. xBD and xView2 support satellite building-damage assessment. BRIGHT adds optical-SAR multimodal damage assessment under all-weather conditions. DisasterM3 supports remote-sensing vision-language tasks. FloodNet and RescueNet support aerial semantic segmentation. CrisisMMD supports multimodal social media triage. EIDSeg supports ground-level damage segmentation. Public alert feeds from weather, seismic, and emergency mapping services can be used for ingestion and replay stress tests.

Three benchmark modes are recommended. Offline benchmark mode evaluates models on held-out datasets using F1, mAP, mIoU, AUROC, calibration error, and report-quality metrics. Trace replay mode converts datasets and live-feed records into simulated event streams with bursts, duplicates, late-arriving imagery, and source conflicts. Fault-injection mode tests node failures, unavailable vector indexes, delayed GPU pools, partial network partitions, and backpressure.
The architecture is successful only if it performs well across all three modes.

Ablations are essential. The full ECHO-DR system should be compared against a monolithic multimodal pipeline, a pipeline without event memory, a pipeline without hierarchical gating, a pipeline without stage disaggregation, a pipeline without vector retrieval, and a pipeline without human review. These ablations identify which architectural features create latency, cost, accuracy, trust, and robustness benefits.

Fig. 7. Evaluation and benchmarking pipeline. Offline datasets, trace replay, architecture variants, and fault injection produce accuracy, latency, cost, reliability, and human-workload metrics.

Table IV: Dataset and Benchmark Mapping

Dataset or Feed	Modality	Representative Tasks	Workflow Role
xBD / xView2	Pre/post satellite imagery	Building localization and damage severity classification	Deep-path remote-sensing benchmark and event confirmation.
BRIGHT	Optical + SAR imagery	All-weather building damage assessment	Cross-sensor fusion and domain-shift benchmark.
DisasterM3	Remote-sensing images + language	Damage reports, descriptions, visual question answering	Vision-language reasoning and report generation benchmark.
FloodNet	Aerial RGB imagery	Flood scene segmentation and classification	UAV/aerial image understanding benchmark.
RescueNet	UAV imagery	Semantic segmentation of damaged scenes	Field imagery and asset-level segmentation benchmark.
CrisisMMD	Social media text + image	Informativeness, humanitarian category, damage severity	Fast-path social triage benchmark.
EIDSeg	Ground-level social images	Pixel-level earthquake infrastructure damage segmentation	Ground-view deep-path segmentation benchmark.
NWS, USGS, Copernicus, FEMA feeds	Structured official alerts and products	Ingestion, event formation, replay, geospatial alignment	Operational stream and integration benchmark.

Metrics

Detection quality should be measured with macro-F1, class-balanced accuracy, mAP, and mIoU depending on task. Geospatial quality should be measured with localization error, geohash accuracy, polygon intersection over union, and road/asset matching accuracy.

Event quality should be measured with cluster purity, event fragmentation, event merge errors, and temporal consistency.

Operational quality should be measured with p50 and p95 alert latency, fused update latency, throughput, cost per 1,000 items, GPU utilization, backlog, recovery time, and dropped-message rate. Human factors should be measured with review queue length, adjudication time, override rate, evidence-click rate, and operator satisfaction.

Calibration is a first-class metric. A disaster AI system that overstates confidence is dangerous even when its average accuracy is high. ECHO-DR should report expected calibration error, selective prediction curves, and abstention quality. The routing policy relies on uncertainty estimates, so uncalibrated models can harm both safety and cost. Calibration should be evaluated by modality and by event type because performance may differ sharply across hazards and geographies.

Baseline Architectures

The monolithic baseline routes every relevant item through a single deep multimodal service before producing an event update. It is simple and may achieve strong per-item reasoning, but it is expected to have poor cost and tail latency under bursts.

The batch lakehouse baseline processes evidence in periodic jobs. It has strong auditability but weak first-alert latency.

The generic inference graph baseline supports routing but lacks event-centric memory and utility-aware escalation. The no-governance baseline measures how much evidence traceability and human control are lost when outputs are generated without a policy and lineage plane.

These baselines make the architecture testable. ECHO-DR should not be accepted merely because it sounds plausible. It should be accepted only if it improves operational metrics while preserving model quality and auditability. The proposed evaluation matrix makes those claims falsifiable.

IX. SIMULATED RESULTS AND SCALABILITY ANALYSIS

This section reports simulated, design-level results. The simulation is intended to evaluate architecture behavior under controlled assumptions, not to claim deployment performance. A trace generator creates item arrivals at four load levels: baseline, moderate burst, severe burst, and extreme surge.

Items are assigned modality, criticality, uncertainty, and event-linking difficulty. The monolithic baseline applies deep multimodal reasoning to every relevant item. ECHO-DR applies fast triage to all relevant items, standard fusion to medium-value items, and deep reasoning to a smaller fraction selected by the routing policy.

The simulation uses the following assumptions. Fast-path inference is low cost and highly parallel. Standard fusion has moderate cost and depends on vector and geospatial retrieval. Deep reasoning is expensive and GPU-bound. During bursts, autoscaling can increase replica counts but not instantly. ECHO-DR can defer low-priority backfill and raise escalation thresholds for noncritical items.

The monolithic baseline cannot separate encoder, prefill, and decode pressure, so its p95 latency increases more sharply under load. The simulated results are directionally consistent with the systems literature on stage-disaggregated multimodal serving, but the exact values are derived from the simulation model defined in this paper.

They should be treated as planning results. A future implementation should replace them with measured values from real clusters and public disaster datasets.

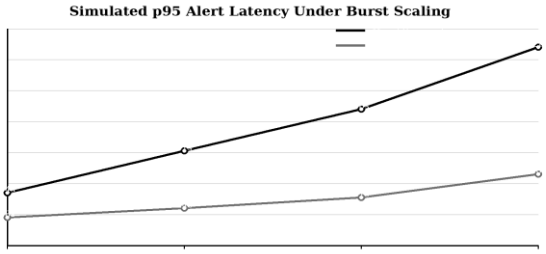

Fig. 8. Simulated p95 provisional alert latency under increasing load. The values are architecture-planning simulations, not field measurements.

Table V: Simulated Architecture Results

Scenario	Monolithic p95 Latency	ECHO-DR p95 Latency	Monolithic Cost Index	ECHO-DR Cost Index	Interpretation
Baseline load	34 s	18 s	1.00	0.84	Early filtering and event-memory reuse reduce latency and compute.
Moderate burst	61 s	24 s	1.00	0.79	Independent scaling absorbs encoder and decoder imbalance.
Severe burst	88 s	31 s	1.00	0.75	Priority routing protects mission-critical items.
Extreme surge	128 s	46 s	1.00	0.72	Graceful degradation prevents total tail-latency collapse.

Scalability Interpretation

The primary scalability result is that ECHO-DR's cost and latency grow with the deep-path fraction rather than total ingest volume alone. If pd remains controlled, the architecture can process large volumes with fast triage while reserving expensive reasoning for high-value evidence.

This does not mean the fast path is sufficient for final decisions. It means that fast path results can create provisional event states while deeper paths operate selectively.

The second result is that stage disaggregation reduces resource interference. Text encoding, image encoding, retrieval, prefill, and decoding have different bottlenecks. A monolithic service ties them together, causing one stage to limit the whole pipeline. ECHO-DR separates them so that autoscaling can add the resource type that is actually constrained.

During a surge of images, vision encoders can scale without unnecessarily increasing decoder replicas. During a surge of report generation, decoder capacity can scale separately. The third result is that event memory reduces repeated work.

If many posts describe the same flooded road, the system can identify duplicates and update a single event instead of repeatedly running expensive reasoning. Semantic retrieval and geospatial buckets reduce event fragmentation. This is especially important for large workflows because redundant evidence is common during public crises.

The fourth result is that graceful degradation improves operational continuity. When the deep path is overloaded, ECHO-DR can still publish structured partial alerts with evidence status. When vector retrieval is degraded, geospatial matching can continue. When cloud connectivity is weak, edge caches can store and forward evidence. A monolithic architecture is more likely to fail as a single bottleneck.

Cost Model

A prototype deployment can be budgeted with modest persistent resources and burst GPU capacity. The purpose of the cost model is to illustrate relative drivers rather than quote a fixed cloud bill.

The major cost variables are streaming throughput, hot storage, fast-path replicas, deep GPU hours, vector index size, observability retention, and egress. The fastest way to lose cost control is to allow every item to enter the deep path.

For a regional prototype, assume a small Kubernetes cluster, a few L4-class fast-path inference nodes, object storage for tens of terabytes, a vector index sized for event evidence, and burst A100-class GPU hours for deep reasoning.

In such a configuration, persistent monthly cost can remain in the low thousands of dollars before labor and data licensing. A production regional deployment would be higher, but still governed primarily by deep-path utilization. This reinforces the importance of utility-aware routing.

Fig. 8 Autoscaling Degradation

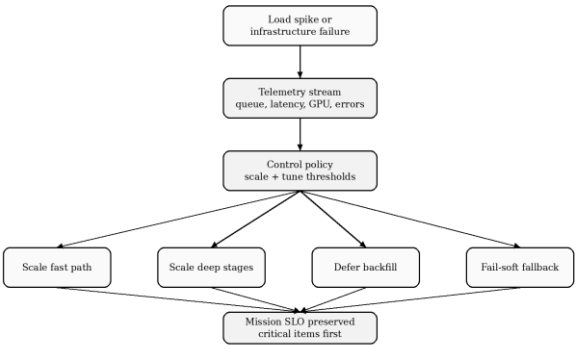

Fig. 9. Autoscaling and graceful degradation control loop. Telemetry drives scaling, threshold adjustment, backfill deferral, and partial-output fallback to preserve critical service-level objectives.

X. SECURITY, GOVERNANCE, AND ETHICS

Disaster AI is safety-critical and socially sensitive. It may process public posts, high-resolution images, infrastructure data, emergency records, and operator decisions.

The architecture must therefore enforce security, governance, and ethics by design. NIST's AI Risk Management Framework emphasizes trustworthy AI across design, development, deployment, and monitoring [23]. NIST's zero-trust architecture guidance emphasizes that no network location should be implicitly trusted [24].

ECHO-DR translates these principles into concrete architectural controls. Security controls include least-privilege service identities, encrypted object storage, signed model artifacts, restricted event classes, audit logs, network segmentation, and authenticated APIs.

Human reviewers should see only the evidence required for their task. Public products should suppress personally identifiable information and precise private locations unless operational policy requires disclosure.

Operator actions should be logged with time, role, and reason. Governance controls include model cards, route-policy versioning, calibration reports, data-retention classes, red-team tests, drift monitoring, and documented override procedures.

Table VI: Governance and Risk Controls

Risk	Example Failure Mode	Control in ECHO-DR
False confidence	Generated summary states uncertain damage as fact.	Confidence intervals, evidence links, calibration, human review triggers.
Privacy leakage	Citizen image or exact location is exposed unnecessarily.	PII redaction, retention classes, role-based views, coordinate coarsening.
Data bias	Digitally visible neighborhoods receive more attention than offline communities.	Blend social evidence with official feeds, sensors, remote sensing, and operator review.
Model drift	Classifier trained on one hazard underperforms in another region.	Drift monitoring, replay tests, per-hazard calibration, retraining workflow.
Security compromise	Unauthorized service accesses restricted event evidence.	Zero-trust access, signed artifacts, least privilege, immutable audit logs.
Automation overreach	System recommends high-consequence action without verification.	Policy-based human-in-the-loop requirements for critical actions.

Governance controls include model cards, route-policy versioning, calibration reports, data-retention classes, red-team tests, drift monitoring, and documented override procedures. Each high-consequence output should include the evidence items used, the model versions invoked, confidence estimates, missing evidence, and review status. An LLM-generated summary without evidence links should never be treated as an authoritative emergency product.

Ethically, the architecture must address data bias and unequal visibility. Social media evidence overrepresents connected populations and underrepresents communities with limited internet access or language coverage. Remote sensing may miss indoor impacts, informal settlements, or local infrastructure context. A system that simply follows data density could misallocate attention. ECHO-DR mitigates but does not eliminate this risk through structured feeds, geospatial priors, human review, and explicit uncertainty reporting.

The architecture should be used as decision support, not autonomous command authority. Recommendations involving evacuation, casualty estimation, structural safety, or resource priority should remain subject to human decision. The system can improve speed and evidence organization, but emergency authority and accountability remain with responsible institutions.

XI. DISCUSSION

The central argument of this paper is that future scalable AI workflows should be designed around event state, selective reasoning, and governance. The field often treats model performance as the core challenge and workflow design as engineering detail. Disaster response shows why this is insufficient. A strong model embedded in the wrong architecture can be slow, expensive, opaque, and unsafe. Conversely, a workflow that routes evidence intelligently can use multiple imperfect models productively while preserving human control and auditability.

ECHO-DR is intentionally modular. It can use different model families, serving platforms, storage systems, and cloud providers. The architectural contribution is not tied to a specific vendor or foundation model. It lies in the composition: event-centric memory, hierarchical routing, stage-disaggregated serving, and governance. These patterns are likely to generalize beyond disaster response to hospital operations, supply-chain risk, cybersecurity incident response, financial compliance, and smart-grid resilience. In each case, the workflow involves heterogeneous streams, uncertain evidence, expensive reasoning, high consequence, and human review.

The architecture also reframes scalability. Traditional scalability discussions emphasize more machines, larger clusters, and higher throughput. ECHO-DR emphasizes reducing unnecessary work. Event memory prevents repeated deep inference over duplicates. Hierarchical routing prevents low-value items from consuming frontier-model capacity. Stage disaggregation prevents one inference stage from starving others. Human review is reserved for policy-sensitive or uncertain events. These choices are architectural forms of efficiency.

The simulated results support the plausibility of the approach, but they are not a substitute for field evaluation. The next research stage should implement a prototype and run reproducible benchmarks on public datasets and trace-driven workloads. Operators should also evaluate whether evidence-

linked outputs improve trust and decision speed compared with ordinary dashboards and static reports.

Comparison with Alternative Architectures

Compared with a batch-centric lakehouse, ECHO-DR improves first-alert latency and online event formation while retaining replay and governance. Compared with a generic inference graph, it adds persistent event memory, utility-aware routing, and policy escalation. Compared with a stage-disaggregated serving system alone, it embeds serving efficiency in an operational mission workflow. Compared with a task-specific retrieval-augmented system, it generalizes retrieval and evidence linking across the full response lifecycle.

The main cost of ECHO-DR is complexity. It requires integration across streaming systems, data stores, serving graphs, geospatial systems, vector retrieval, model registries, and human workflows. This complexity is justified only in settings where latency, heterogeneity, and consequence are high. For a small single-task deployment, a simpler model endpoint may be preferable.

Table VII: Architecture Comparison

Architecture Family	Strengths	Weaknesses for Disaster Response	ECHO-DR Difference
Batch lakehouse pipeline	Governance, replay, historical analytics.	Slow first-alert behavior and limited online routing.	Adds stream-native event formation and low-latency triage.
Generic inference graph	Composable models and autoscaling.	Often request-centric and lacks durable event memory.	Adds event graph, lineage, and utility-aware routing.
Stage-disaggregated serving only	Improves model-serving throughput and tail latency.	Optimizes serving rather than mission workflow.	Embeds stage disaggregation inside disaster operations.
Task-specific RAG system	Evidence-linked reasoning for a narrow task.	Limited generality and operational scheduling.	Generalizes retrieval to event memory and multiple products.
Monolithic multimodal endpoint	Simple interface and strong per-item reasoning.	Expensive, opaque, and fragile under bursts.	Uses deep reasoning selectively with human review.

XII. LIMITATIONS AND FUTURE WORK

The paper has several limitations. First, the evaluation is simulated. The architecture should be validated on an implemented prototype with real workloads, measured cluster behavior, and operator studies. Second, the routing policy depends on uncertainty estimates, but calibration under disaster domain shift is difficult.

Third, event linking can fail when location is missing, social evidence is misleading, or multiple incidents occur close together. Fourth, human review capacity is finite; a high-consequence event can overwhelm operators if too many items are escalated. Fifth, privacy and governance requirements vary across jurisdictions, so deployment requires local legal review. Another limitation is model dependence. ECHO-DR is model-agnostic in design, but real performance depends on the quality of selected models. A weak geolocation model can create false clusters. A poorly calibrated vision model can understate damage. A vision-language model can hallucinate unsupported details. These risks are mitigated by evidence links, calibration, human review, and replay testing, but they are not eliminated.Future work should proceed in five directions. First, implement the architecture as an open reference stack with reproducible datasets and trace-generation scripts. Second, learn the routing policy from operational outcomes using constrained bandits or reinforcement learning with safety limits.

Third, develop causal event memory that represents not only observations but also hazard propagation and infrastructure dependencies. Fourth, support privacy-preserving federation across agencies so that event confidence can be shared without exposing sensitive raw evidence. Fifth, run human factors studies to measure whether evidence-linked AI products improve operator decisions under time pressure.

Future work should also explore multilingual and low-resource settings. Many disasters occur in regions where social media posts, field reports, and official communications use multiple languages or dialects. Multilingual retrieval and

culturally aware geocoding are therefore necessary for equitable performance. Remote-sensing models must also be evaluated across building types, settlement patterns, and sensor conditions that differ from common benchmark regions.

XIII. CONCLUSION

This paper proposed ECHO-DR, an Event-Centric Hierarchical Orchestration architecture for scalable AI workflows in real-time disaster response. The architecture addresses a real-world problem: the inability of conventional AI pipelines to convert heterogeneous, bursty, multimodal disaster evidence into timely, auditable, and trustworthy operational intelligence. ECHO-DR solves this problem at the workflow level by combining event-centric memory, utility-aware routing, stage-disaggregated multimodal serving, human review, and governance.

The formal model shows that efficiency depends on the fraction of evidence escalated to expensive reasoning, not merely on total ingest volume. The system design shows how raw evidence, lakehouse storage, vector retrieval, geospatial indexing, and event graphs can be integrated into a memory plane.

The demonstration shows how early provisional alerts can be revised through continuous evidence accumulation. The simulated evaluation suggests that the proposed architecture can reduce tail latency and cost relative to a monolithic multimodal pipeline while preserving auditability.

The broader lesson is that future AI scalability will come from architecture as much as from model size. Large workflows require systems that know when to use a small model, when to retrieve, when to fuse, when to call a large model, when to defer, and when to ask a human. ECHO-DR is one concrete design for that future, focused on disaster response but applicable to many high-stakes domains where evidence is multimodal, uncertainty is unavoidable, and decisions must be timely and explainable.

Appendix

Reproducibility Checklist

A prototype evaluation should release the following artifacts where policy permits: event-envelope schema; routing-policy configuration; trace-generation parameters; dataset splits; model versions; calibration procedures; deployment manifests; metrics definitions; and scripts for replay, fault injection, and ablation analysis. The simulated results in this paper should be treated as expected planning behavior until replaced by measured prototype results.

The minimum reproducible experiment should compare the full ECHO-DR system with a monolithic multimodal baseline under the same arrival traces. It should report the same metrics for accuracy, latency, throughput, cost, cluster purity, review workload, and evidence loss. It should include at least one remote-sensing dataset, one social-media dataset, one official-feed replay, and one fault-injection scenario.

Example Event Envelope

Schema Sketch: Normalized Evidence Envelope

```
{
  "evidence_id": "uuid",
  "source": {"type": "social|sensor|imagery|report", "provider": "..."},
  "modality": "text|image|video|sar|optical|structured",
  "time": {"observed_at": "ISO-8601", "ingested_at": "ISO-8601"},
  "geography": {"point": [lat, lon], "polygon": "optional", "precision": "..."},
  "privacy_class": "public|restricted|sensitive",
  "raw_reference": "object-store-uri",
  "features": {"embedding_ref": "...", "labels": [], "confidence": {}},
  "lineage": {"pipeline_version": "...", "model_versions": []}
}
```

Deployment Validation Checklist

Before operational use, a deployment of ECHO-DR should pass a staged validation process. The first stage is data validation. Each input source should be checked for schema stability, timestamp quality, geospatial precision, source authentication, privacy class, and replay behaviour.

A disaster AI workflow is only as reliable as the evidence envelope that enters it. If time fields are inconsistent, source identifiers are missing, or location precision is overstated, downstream event linking will appear more confident than it should. Therefore the ingestion layer should reject invalid records, mark uncertain fields explicitly, and preserve raw evidence references for later audit.

The second stage is routing validation. The utility-aware routing policy should be tested under normal, burst, and degraded conditions. Operators should inspect how many items are archived, triaged, fused, escalated to deep models, and escalated to human review. A safe policy is not necessarily the policy that maximizes deep-path accuracy.

It is the policy that sends enough evidence to expensive reasoning to protect high-consequence decisions while preventing compute collapse and review overload. The validation process should include threshold sweeps, queue-pressure experiments, and manual review of false negatives created by low-cost filtering.

The third stage is event-memory validation. Event linking should be evaluated for fragmentation and over-merge errors. Fragmentation occurs when evidence about one incident is split across multiple event records; over-merge occurs when distinct incidents are incorrectly combined.

Both errors are operationally significant. Fragmentation hides the total impact of an event, while over-merge can create false confidence or wrong location. Validation should therefore include dense urban scenarios, sparse rural scenarios, simultaneous hazards, and noisy social reports with incomplete location metadata.

The fourth stage is product validation. Operators should be asked whether the output card, map layer, evidence list, confidence statement, and recommended next action are usable under time pressure.

A technically correct model output can still fail if it is not presented in a form that fits emergency workflows. Each alert should make clear what is known, what is inferred, what is missing, what evidence supports the claim, and whether a human reviewer has verified it. This product-centered validation is essential because disaster response is ultimately a human decision environment.

The fifth stage is resilience validation. The system should be tested with unavailable GPU pools, delayed imagery, unavailable vector indexes, network partitions, overloaded message queues, and corrupted evidence items.

The expected behavior is graceful degradation rather than silent failure. In degraded mode, the system should preserve raw evidence, publish partial alerts only with appropriate confidence labels, and record why deeper reasoning was unavailable. These tests are not optional engineering exercises; they are part of the safety case for using AI in emergency operations.

Scenario Trace Definition

A reproducible trace for the flood demonstration can be defined as a sequence of timed evidence arrivals. At minute 0, the system receives a weather alert and three hydrological sensor updates indicating rising river levels. At minute 4, it receives ten social media posts, of which four are duplicates, three are unrelated weather complaints, two describe a flooded underpass, and one contains an image without clear location. At minute 8, two 311 reports arrive with approximate addresses. At minute 15, a field responder uploads an image from a mobile device. At minute 40, a UAV image arrives. At minute 80, a satellite-derived product becomes available. This trace captures the central asymmetry of disaster response: weak but fast evidence arrives early, while stronger geospatial evidence often arrives later.

The monolithic baseline processes each relevant item through a deep model before event update. In the trace, this creates queue pressure during the social burst and delays the first actionable underpass alert. The ECHO-DR trace first uses the fast path to identify relevance and approximate location, links the social posts to a provisional event, and publishes a low-confidence alert with evidence links.

The standard path later combines the social posts with 311 reports and field imagery, increasing confidence. The deep path is triggered only after criticality and uncertainty justify it. The UAV image and satellite-derived product refine the spatial footprint and severity estimate. The event timeline therefore shows a sequence of progressively stronger claims rather than a single delayed conclusion.

This trace should be evaluated with event-level metrics. The first-alert latency is measured from the first relevant social post to the first provisional alert. The fused-update latency is measured from the arrival of corroborating 311 or field evidence to the updated event state. The final-product latency is measured from the arrival of deep imagery to the verified map product. Correctness is measured not only by final severity classification but also by whether the event remained coherent across updates. A successful workflow should avoid both premature certainty and excessive delay.

Scenario traces should also include negative cases. For example, a viral repost of an old flood image should be detected as a duplicate or stale artifact when metadata, semantic similarity, or external evidence contradicts it.

A text-only report with no location should remain low confidence until geocoding or corroboration improves it. A severe report near a hospital or evacuation route should receive a higher priority than a similar report in a low-impact zone. These negative and boundary cases are where architecture matters most, because a single model endpoint often lacks the state and policy context to handle them safely.

Table VIII: Example Reproducible Flood Trace Events

Time	Evidence Arrival	Expected Route	Expected Event-Memory Effect
0 min	Weather alert and river gauge rise	Fast path	Create regional flood-watch event with moderate confidence.
4 min	Ten social posts with duplicates and unrelated messages	Fast path	Filter noise, deduplicate, and identify underpass candidate event.
8 min	Two 311 reports with approximate addresses	Standard path	Link reports to underpass event and raise confidence.
15 min	Responder image from mobile app	Standard or deep path	Add visual evidence and update severity posterior.
20 min	Conflicting repost of old flood image	Fast path plus retrieval	Detect stale or duplicate evidence and prevent false escalation.
40 min	UAV image of the underpass area	Deep path	Segment water coverage and vehicle obstruction.
45 min	Operator verification	Human review	Confirm blockage and record adjudication in lineage ledger.
80 min	Satellite-derived flood product	Standard/deep path	Refine spatial footprint and support final map product.
120 min	Road crew update	Fast/standard path	Revise event state from active blockage to clearing in progress.
180 min	Post-event replay	Evaluation workflow	Measure route quality, latency, evidence lineage, and false positives.

Acknowledgment

The authors acknowledge the open research communities, public dataset maintainers, emergency management agencies, and open-source infrastructure projects that make reproducible disaster AI research possible. No field deployment or human-subject experiment is claimed in this paper.

REFERENCES

1. F. Alam, F. Ofli, and M. Imran, "CrisisMMD: Multimodal Twitter Datasets from Natural Disasters," Proceedings of ICWSM, 2018. URL: https: //arxiv.org/abs/1805.00713
2. R. Gupta et al., "xBD: A Dataset for Assessing Building Damage from Satellite Imagery," 2019. URL: https:/ /arxiv.org/abs/1911.09296
3. IBM Research, "The xView2 AI Challenge," 2019. URL: https:/ /www.ibm.com/think/insights/the-xview2-ai-challenge
4. M. Rahnemoonfar et al., "FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding," 2020. URL: https ://arxiv.org/abs/2012.02951
5. RescueNet authors, "RescueNet: A High Resolution UAV Semantic Segmentation Dataset for Natural Disaster Damage Assessment," Scientific Data, 2023. URL: https:// www.nature.com/articles/s41597-023-02799-4
6. H. Chen et al., "BRIGHT: A Globally Distributed Multimodal Building Damage Assessment Dataset With Very-High-Resolution for All-Weather Disaster Response," Earth System Science Data, 2025. URL: https//essd.copernicus.org/articles/17/6217/2025/
7. J. Wang et al., "DisasterM3: A Remote Sensing Vision-Language Dataset for Disaster Damage Assessment and Response," 2025. URL: https: //ar xiv.org/abs/2505.21089
8. EIDSeg authors, "A Pixel-Level Semantic Segmentation Dataset for Post-Earthquake Damage Assessment From Social Media Images," 2025. URL: https: //arxiv.org/html/2511.06456v2
9. A.Review Authors, "Social Media for Managing Disasters Triggered by Natural Hazards: A Critical Review of Data Collection Strategies and Actionable Insights," Natural Hazards and Earth System Sciences, 2026. URL: https ://nh ess.copernicus.org/articles/26/215/2026/
A. Remote Sensing Review Authors, "Integrating Machine Learning and Remote Sensing in Disaster Management: A Decadal Review of Post-Disaster Building Damage Assessment," Buildings, 2024. URL: https:// www.mdpi.com/2075-53 09/14/8/2344
10. CrisiSense-RAG authors, "CrisiSense-RAG: Crisis Sensing Multimodal Retrieval-Augmented Generation for

Rapid Disaster Impact Assessment," 2026. URL: https ://arxiv.org/html/2602.13239v2
11. A.Singh et al., "Efficiently Serving Large Multimodal Models Using EPD Disaggregation," Proceedings of Machine Learning and Systems, 2025. URL: https: //proceedings.mlr.press/v267/singh25d.html
12. ModServe authors, "ModServe: Modality- and Stage-Aware Resource Disaggregation for Scalable Multimodal Model Serving," 2025. URL: https:/ /arxiv.org/abs/2502.00937
13. Compound AI systems authors, "Scalable Inference Architectures for Compound AI Systems: A Production Deployment Study," 2026. URL: https: /arxiv.org/abs/2604.25724
14. KServe, "InferenceGraph Documentation." URL: https: //kserve.github.io/website/docs/concepts/resources/inferencegraph
15. Ray Project, "Ray Serve: Scalable and Programmable Serving." URL: https ://docs.ray.io/en/latest/serve/index.html
16. Apache Spark, "Structured Streaming Programming Guide." URL: https: //spark.apache.org/docs/latest/streaming/index.html
17. Apache Iceberg, "Schema Evolution Documentation." URL: https ://iceberg.apache.org/docs/latest/evolution/
18. OpenTelemetry, "OpenTelemetry Documentation." URL: http s://opentelemetry.io/
19. National Weather Service, "API Web Service Documentation." URL: https:// www.weather.gov/documentation/services-web-api
20. U.S. Geological Survey, "Real-Time Earthquake Feeds." URL: https:/ /earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php
21. Copernicus Emergency Management Service, "Rapid Mapping and Emergency Management Products." URL: https ://data.jrc.ec.europa.eu/collection/id-0072
22. National Institute of Standards and Technology, "AI Risk Management Framework." URL: https:/ /www.nist.gov/itl/ai-risk-management-framework
23. National Institute of Standards and Technology, "Zero Trust Architecture," Special Publication 800-207, 2020. URL: http s://www.nist.gov/publications/zero-trust-architecture
24. S. Liu et al., "Grounding DINO: Marrying DINO With Grounded Pre-Training for Open-Set Object Detection," 2023. URL: https: //arxiv.org/abs/2303.05499
25. N. Ravi et al., "SAM 2: Segment Anything in Images and Videos," 2024. URL: https ://arxiv.org/abs/2408.00714
26. Qwen Team, "Qwen2.5-VL Technical Report," 2025. URL: https: //arxiv.org/abs/2502.13923
27. J. Chen et al., "M3-Embedding: Multi-Linguality, Multi-Functionality, and Multi-Granularity Text Embeddings," 2024. URL: https: //arxiv.org/abs/2402.03216
28. D. Szwarcman et al., "Prithvi-EO-2.0: A Versatile Multi-Temporal Foundation Model for Earth Observation," 2024. URL: https:/ /arxiv.org/abs/2412.02732
29. NVIDIA, "Triton Inference Server Batcher Documentation." URL: https: //docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/user_guide/batcher.html
30. vLLM Project, "vLLM Documentation and System Overview." URL : https: //vllm.ai/
31. NVIDIA, "TensorRT-LLM." URL: https: //developer.nvidia.com/tensorrt-llm
32. Kubernetes, "Horizontal Pod Autoscaling." URL: https:/ /kubernetes.io/docs/concepts/workloads/autoscaling/horizontal-pod-autoscale/
33. Temporal, "Temporal Documentation." URL: http s://docs.temporal.io/
34. Federal Emergency Management Agency, "Response Geospatial Office." URL: https:// www.fema.gov/about/offices/response/response-geospatial
35. Cybersecurity and Infrastructure Security Agency, "Artificial Intelligence and Secure by Design Guidance." URL: ht tps://www.cisa.gov/ai

Author Profile

First Author: Replace this placeholder with the author biography, current affiliation, research interests, and relevant academic or professional background. The profile should be edited before final submission.

Second Author: Replace this placeholder with the author biography, current affiliation, research interests, and relevant academic or professional background. The profile should be edited before final submission.