

Persona-as-Configuration: Generative Stakeholder Reporting for Agricultural Floods

Oliver Aleksander Larsen, Tiziano Santilli, Francesco Daghero, and Mahyar T. Moghaddam

University of Southern Denmark, Maersk Mc-Kinney Moller Institute,
Odense, Denmark
{olar,tisa,fdag,mtmo}@mmmi.sdu.dk

Abstract. Cyber-physical systems built on deterministic edge inference, such as on-vehicle flood detection for agricultural fields, produce structured decision logs that must be interpreted differently by heterogeneous stakeholders. Pairing such systems with large language models (LLMs) to generate stakeholder-specific reports introduces a tension: the generative layer is non-deterministic, while the edge plane must remain replayable and auditable. We propose an architectural pattern resting on two invariants: *unidirectional consumption*, in which the generative layer is a strict read-only consumer of the deterministic plane and never writes back, and *persona-as-configuration*, in which stakeholder adaptation is a versioned prompt-template artifact rather than runtime improvisation. We instantiate the pattern as a context-aware dashboard layer over the JSON decision logs of a previously published edge-based standing-water detection system, and analyse how the integration boundary admits standard generative-reliability mitigations as configuration- or middleware-level extension points. A structured expert review rated the pattern favourably across five ISO/IEC 25010-aligned quality dimensions, with strongest agreement on separation of concerns. End-user evaluation with agricultural stakeholders is planned for future work.

Keywords: software architecture · generative AI · persona pattern · trust boundary · audit · cyber-physical systems · edge decision logs

1 Introduction

Standing water in agricultural fields threatens vehicle mobility, crop viability, and soil structure, especially on narrow off-road tracks where drivers must make rapid go/no-go decisions under changing environmental conditions [32]. Detecting such hazards in real time requires inference on commodity edge hardware

This version of the contribution has been accepted for publication after peer review, but is not the Version of Record and does not reflect post-acceptance improvements or corrections. Accepted at CASA 2026 (9th Workshop on Context-Aware, Autonomous and Smart Architectures), co-located with ECSA 2026. The Version of Record will appear in the Springer LNCS post-proceedings of ECSA 2026.

operating under intermittent connectivity and constrained power budgets [8]. Yet accurate detection alone is insufficient: the value of a flood-monitoring system depends on whether the right people can act on its outputs.

Six stakeholders need distinct outputs: farmers (field accessibility), agronomists (soil saturation), farm managers (route safety), government agencies (audit-ready documentation), insurers (georeferenced damage evidence), and decision-makers (cross-role executive summary). Translating a single set of sensor readings into these varied narratives requires natural-language generation that is both data-grounded and role-aware. Large language models (LLMs) offer this capability, yet integrating a non-deterministic generative component into a deterministic edge system raises architectural questions that have received little attention.

The LLM in this architecture is not a report generator bolted onto a monitoring system: it is the *mechanism* by which a single deterministic event stream is multiplexed into N stakeholder-appropriate views. Without it, persona-based reporting collapses into N hand-written templates that are infeasible at the cardinality of sensor-pattern $\times$ anomaly-type $\times$ persona and cannot synthesise spatial clusters or correlate anomaly magnitudes. The persona-as-configuration architectural element ceases to exist. The architectural function of the generative component is therefore multiplexing-via-personas, not summarisation, positioning this work within the family of context-aware, smart cyber-physical architectures.

We propose a two-layer architecture for agricultural flood monitoring that addresses both detection and stakeholder communication. The *edge detection layer*, contributed in our prior work [13], classifies standing-water hazards on resource-constrained on-vehicle hardware and persists structured decision logs. This paper treats it as a black-box producer. The *human-interface layer* consumes those logs through a web-based dashboard and generates stakeholder-tailored reports using a cloud-hosted LLM, and is the locus of the two contributions that follow.

C1 (Unidirectional consumption invariant). A strict read-only-consumer relationship from the generative layer to the deterministic edge plane admits LLM non-determinism into a safety-relevant cyber-physical system without compromising replayability or audit.

C2 (Persona-as-configuration pattern). Stakeholder adaptation is elevated to a first-class configuration element by mapping each stakeholder type to a versioned prompt template, applying the *Persona* pattern [27] at the architectural rather than the prompt-engineering level.

Both contributions are assessed through a structured expert review tied to the ISO/IEC 25010 product quality model [10]. Empirical end-user evaluation is planned for future work.

2 Background and Related Work

2.1 Standing-Water Detection and Edge Perception

Detecting standing water for off-road vehicles spans satellite mapping, fixed camera networks, and on-vehicle edge perception [25]. Our prior work [13] contributes a deployed edge-based detection system that uses finite-state-machine-guided model tiering and multi-model consensus on Raspberry-Pi-class hardware. The present paper treats that system as a black-box producer of JSON decision logs and focuses entirely on the human-interface layer that consumes them.

2.2 LLMs for Operational Decision Support

Generating stakeholder-appropriate text from structured data has been studied as data-to-text natural language generation for over two decades [18]. Large language models extend this work to less templated, narrative outputs and are increasingly applied to structured-data interpretation for non-technical stakeholders. An et al. proposed IoT-LLM, a framework that preprocesses Internet of Things (IoT) signals into LLM-amenable formats with retrieval-augmented generation (RAG [6]) and chain-of-thought reasoning [1]. A systematic review of LLMs in IoT identifies data interpretation and contextual reasoning as the principal gains over rule-based analytics [5]. In agriculture, IoT and AI integration have reshaped precision farming [14,31], yet the focus remains on sensing rather than generating stakeholder-specific narratives from sensor outputs. A review of LLM applications in disaster management identifies insufficient integration across stakeholder groups and inadequate transformation of situational data into actionable insights as critical gaps [29]. Our architecture addresses both by generating stakeholder-differentiated reports from edge decision logs.

Beyond model selection, prompt engineering plays a key role in tailoring LLM outputs to specific audiences. White et al. introduced a prompt pattern catalog including the *Persona* pattern [27]. Schreiber et al. later extended this catalog into a pattern language with multi-persona and dynamic-switching variants at the prompt-engineering level [21]. Schulhoff et al. presented a taxonomy of 58 prompting techniques [22]. A survey on persona in LLMs distinguishes *role-playing* (assigning a persona to the model) from *personalization* (adapting to a user’s persona) [26], and recent empirical work shows that role assignment alone in system prompts does not consistently improve LLM task performance [33]. Our claim is correspondingly architectural: we apply the *Persona* pattern as a first-class architectural element, mapping each stakeholder type to a versioned prompt configuration that treats role-based prompting as a unit of governance, audit, and substitution, not as an accuracy lever.

2.3 Dashboard Architectures for IoT and Flood Monitoring

IoT application architectures route sensor networks through layered topologies [4] to human-facing dashboards providing visualization, alerting, and trend

analysis. FAIS combines IoT APIs, USGS camera feeds, and convolutional neural network classifiers into a Python web dashboard with map-based flood analytics [20]. Recent systems integrate ultrasonic and pressure sensors with web dashboards and Firebase alerting for real-time flood-risk classification [15]. These architectures are typically cloud-centric (sensors stream data to a central server that renders dashboards) and focus on visualization and alerting rather than natural-language interpretation.

Read-only consumption downstream of the sensing tier is itself standard in this IoT application layer [4]; what changes the problem here is a non-deterministic consumer. Our architecture consumes offline JSON artifacts from a resource-constrained edge system through a generative module, so the Layer 1 to Layer 2 boundary becomes a deliberately stated invariant that protects the deterministic edge plane, not an incidental flow direction, and is paired with persona-as-configuration and the reliability-seam analysis of §6.2.

2.4 Architectural Patterns for AI-Integrated Systems

Software architecture research has recently begun to address LLM integration challenges, extending the ML-systems architectural agenda [23] with pattern-oriented techniques [2]. Bucaioni et al. [3] proposed a functional reference architecture for LLM-integrated systems, noting the absence of commonly accepted architectural blueprints. Shamsujjoha et al. [24] adapted the Swiss Cheese Model into multi-layered runtime guardrails for agents based on foundation models, with each layer independently protecting specific quality attributes. Xu et al. [30] surveyed LLM-enabled cyber-physical systems (CPS) and categorised the roles LLMs play, highlighting the tension between probabilistic LLM reasoning and deterministic CPS requirements.

Our architecture resolves this tension through strict unidirectional data flow, not runtime guardrails. Where Bucaioni et al. [3] catalogue functional blocks and Shamsujjoha et al. [24] layer runtime guardrails, we make the boundary itself a load-bearing architectural invariant (§3.2). Closest in spirit, González-Potes et al. [7] pair deterministic rule-based supervision with an LLM advisory channel for industrial batch control. The safety partition is shared, but not the stakeholder-identity-as-configuration treatment of §4. The cited guardrails would attach at the seams identified in §6.2.

3 Architecture

This section presents the system architecture in three parts: the architectural drivers that shape the design, the two-layer organisation and its key invariants, and the dashboard component decomposition with the LLM integration point characterised as an architectural boundary. Figure 1 provides an overview.

3.1 Architectural Drivers

Six drivers shape the architecture and are annotated downstream where each is satisfied. **D-Det** (Determinism of the safety-critical path): the edge plane must be replayable bit-for-bit independent of the generative layer, driving the unidirectional invariant of §3.2. **D-Rep** (Replayability and audit): every report must be reconstructable from the underlying decision log, driving the schema-stable JSON contract of Table 1. **D-Stk** (Stakeholder heterogeneity): one deterministic stream must be rendered as N role-appropriate views, driving persona-as-configuration in §4. **D-Cost** (Bounded LLM cost): a cloud LLM call is orders of magnitude more expensive than a per-frame edge inference and cannot be invoked per frame, driving on-demand invocation in §3.4. **D-Conn** (Intermittent agricultural connectivity): the edge plane runs offline, so cloud-bound interpretation must be deferrable through batch ingest of JSON at the base station. **D-Sub** (Independent provider evolution): the chosen LLM may need to be swapped (e.g. cloud-to-local) without touching surrounding code, driving the substitutable `llm_report` interface of §3.4.

The relevant reference frame for these drivers is not application architecture (hexagonal, clean, MVC) but cyber-physical control-plane separation [30,24]: the dominant concern is the safety boundary between deterministic and non-deterministic compute, not the inversion of business-logic dependencies. A hexagonal decomposition could be applied within Layer 2 (data-loader, presenter, LLM-adapter as ports) but would not address the cross-layer invariant that defines the architecture and that D-Det and D-Rep require.

3.2 Two-Layer Organisation and the Unidirectional Invariant

Layer 1 (Edge Detection). The edge layer, described in detail in our prior work [13], runs on-board a vehicle traversing agricultural fields. It combines a Raspberry-Pi-class capture node, a processing node that owns a finite-state machine for adaptive model tiering, and an optional NVIDIA Jetson worker for GPU-accelerated inference. Environmental sensor anomalies are fused with multi-model vision consensus to classify each frame as 0 (No Flood), 1 (Some Water), or 2 (Flooded). What matters here is the contract it exposes. The Layer 1 / Layer 2 contract is a per-sensor JSON record (Table 1), and every design choice that follows is made against this contract rather than against the implementation that produces it.

Layer 2 (Human Interface) runs on a base-station workstation, ingests Layer 1’s JSON artifacts, and provides two capabilities: (a) an interactive colour-coded flood map rendering the geo-referenced snapshot on a Leaflet map via Folium [19], and (b) on-demand stakeholder-tailored natural-language reports generated by GPT-5-mini [16].

The central design invariant is that data flows *strictly from Layer 1 to Layer 2*. This guarantees the safety-critical detection path remains unaffected by the non-deterministic generative component. The LLM is advisory: it interprets decisions already made, rather than influencing future ones.

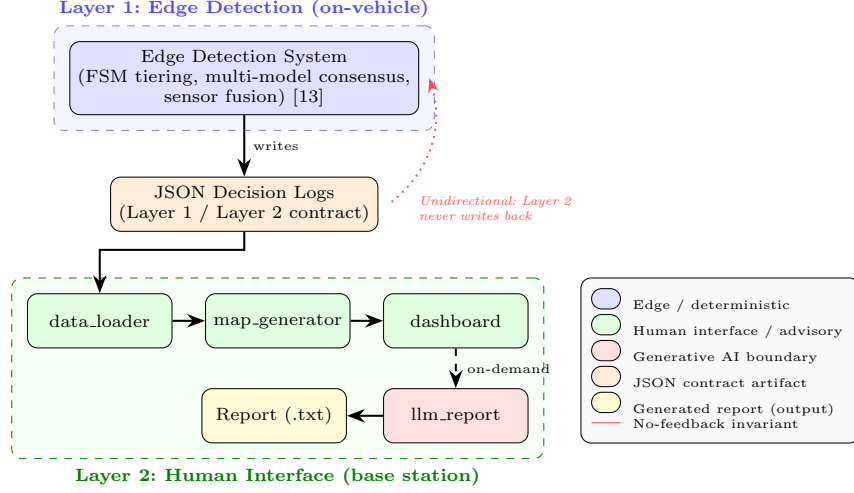


Fig. 1. Two-layer architecture. Layer 1 (edge detection, described in [13]) writes JSON decision logs that fix the Layer 1 / Layer 2 contract. Layer 2 (this paper’s contribution) consumes them for visualization and on-demand stakeholder-tailored LLM reporting. Data flows strictly downward. The unidirectional invariant prevents the generative layer from writing back to the safety-critical detection plane.

Table 1. Key per-sensor JSON record fields consumed by Layer 2 (a timestamp and detection-image are also present in the schema).

Field	Description
<code>camera_id</code>	Sensor / camera identifier
<code>location</code>	GPS coordinates [<code>lat</code> , <code>lon</code>]
<code>sensor_data</code>	Temperature (°C), humidity (%), pressure (hPa)
<code>sensor_anomalies</code>	ΔT , ΔRH , ΔP vs. diurnal baseline
<code>prediction</code>	Hazard classification: 0 / 1 / 2
<code>scores.combined</code>	Weighted flood score. Higher means stronger evidence

3.3 Dashboard Component Decomposition

Layer 2 is implemented as a modular Python application with five modules (Table 2). Figure 2 shows the running prototype. At runtime, `data_loader` reads JSON files into sensor-record dictionaries. `map_generator` renders them as colour-coded Folium markers. `dashboard` embeds the map, computes summary counts, and renders the UI. When the operator selects a stakeholder type and clicks *Generate AI Report*, `llm_report` constructs a prompt (§4), calls GPT-5-mini, displays the result, and writes a timestamped copy.

Table 2. Module decomposition of the human-interface layer.

Module	Responsibility
<code>config</code>	Deployment constants: colour-code mapping, data paths, classification thresholds.
<code>data_loader</code>	Parses JSON sensor files from <code>data/</code> and aggregates readings into an in-memory list of sensor records.
<code>map_generator</code>	Produces an interactive Folium [19] map with colour-coded markers (red = Flooded, orange = Suspicious, green = Normal) and click popups showing sensor data and detection images.
<code>dashboard</code>	Hosts the Dash [17] web application: embeds the Folium map, renders summary cards, provides a stakeholder dropdown, and exposes a <i>Generate AI Report</i> callback.
<code>llm_report</code>	Generative-AI boundary (§3.4): accepts sensor data + stakeholder type, constructs a role-specific prompt, calls GPT-5-mini, returns and persists report text.

3.4 LLM Integration as an Architectural Boundary

The `llm_report` module is the controlled entry point through which the generative component interacts with the system. The boundary is designed around four properties tied to the architectural drivers of §3.1. *On-demand invocation* (D-Cost) calls the LLM only when an operator explicitly requests a report, keeping the dashboard responsive and bounding API spend. *Statelessness* packages each report fresh from the JSON record and stakeholder type, with no conversation history across calls. This prevents context-window drift and makes the prompt bit-exact reconstructible from the JSON record and stakeholder type. *Auditability* (D-Rep) pairs every generated report with a timestamped copy alongside the source JSON record and the version-controlled prompt template, so the chain from edge decision to stakeholder report remains reconstructable despite LLM

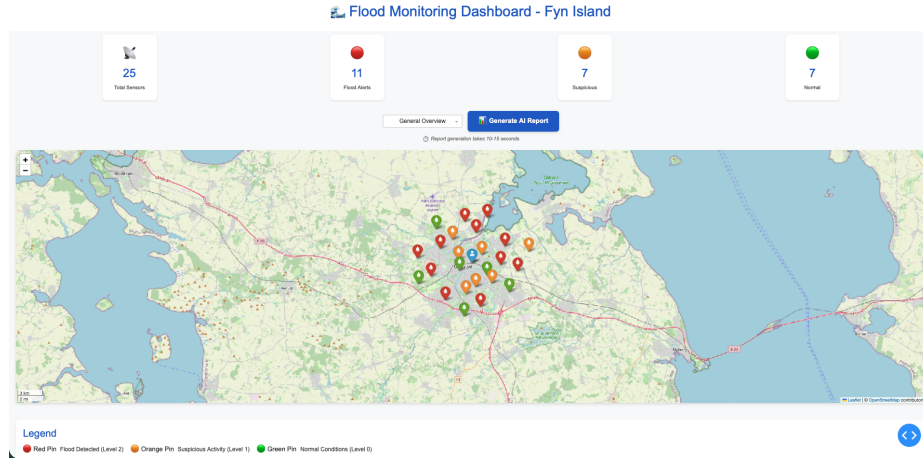


Fig. 2. Dashboard prototype showing the Fyn Island monitoring area. Colour-coded map pins indicate per-sensor flood classifications (red = Flooded, orange = Suspicious, green = Normal). Summary cards report aggregate counts. The *Generate AI Report* button triggers stakeholder-tailored report generation via GPT-5-mini.

phrasing variability. *Substitutability* (D-Sub) confines provider choice to a narrow interface (sensor data and stakeholder type in, report text out), so GPT-5-mini can be replaced by a different cloud provider or by a local model without touching dashboard code.

4 Stakeholder-Aware Report Generation

4.1 Prompt Configuration Architecture

Each of the six supported stakeholder types (Table 3) maps to a structured prompt configuration with three elements: 1) a *system prompt* assigning a domain-specific persona and report structure; 2) a *data injection template* serialising aggregated sensor records; and 3) a *focus directive* enumerating priority topics.

This design applies the Persona pattern [27] at the architectural level: each persona is a version-controlled configuration artifact, not an ad-hoc instruction. The architectural consequence is concrete: a new stakeholder type should be addable by a non-prompt-engineer (a domain expert) because the artifact has a stable schema and is reviewable independently of the data pipeline, dashboard, or LLM integration code. Nothing else is touched.

4.2 Data-to-Prompt Pipeline

When a report is requested, the module executes four steps.

Table 3. Stakeholder prompt configurations and report focus areas.

Stakeholder	Report Focus
General Overview	Executive summary, overall risk level, prioritised recommendations.
Farmer	Field accessibility, crop damage potential, livestock safety, immediate action items.
Agronomist	Soil health indicators, cross-sensor anomaly patterns, drainage and remediation advice.
Farm Manager	Vehicle and route safety, detection confidence trends, workforce scheduling.
Government Agency	Regulatory compliance, audit-ready documentation, inter-agency coordination.
Insurance Company	Geo-referenced damage evidence, quantified loss estimates, temporal claims records.

1. Aggregation. All sensor records are serialised into a structured text block. Each record is rendered as a compact summary: timestamp, sensor identifier, GPS coordinates, environmental readings, anomaly deltas (ΔT , ΔRH , ΔP), classification label, and combined score. Including the full sensor network lets the LLM synthesise spatial patterns such as clusters of flood alerts.

2. Template composition. The data block is inserted into the selected stakeholder’s prompt template, which specifies the persona (e.g., “You are an agronomist advising on post-flood soil remediation”), expected sections (risk assessment, affected areas, recommendations), and focus directive from Table 3.

3. API submission. The composed prompt is submitted to GPT-5-mini as a single-turn request: a system message (persona and instructions) and a user message (sensor data and generation directive). No multi-turn conversation is maintained, enforcing the statelessness property described in §3.4.

4. Persistence and presentation. The returned text is displayed in a modal overlay and simultaneously written to a timestamped `.txt` file for auditing.

4.3 Illustrative Output

Table 4 shows excerpts from the six persona reports generated from one sensor record: the JSON data block in the user message is held fixed, and only the prompt-template configuration (the system-message persona and focus directive) changes. For contrast, the farmer receives plain-language accessibility guidance while the insurer receives claims-oriented damage evidence, showing how the Persona configuration produces operationally distinct outputs without any change to the data pipeline.

Table 4. Excerpts generated from the same 25-sensor decision-log batch (Fyn Island, 2026-02-15; 11 stations at Level 2, score range 0.080 to 3.003). All six stakeholder personas are produced from one JSON input. Only the prompt-template configuration changes.

Stakeholder	Generated Excerpt
General Overview	“11 of 25 stations at Level 2; mean score 0.789. Localised but significant; immediate operational response recommended.”
Farmer	“Immediate no-go: expect deep standing or flowing water. Do NOT attempt to access nearby fields by vehicle.”
Agronomist	“ $\Delta P = -21$ hPa with humidity 88% indicates frontal-system passage; prolonged soil saturation expected. Prioritise drainage assessment within 48 h.”
Farm Manager	“Eleven Level-2 and seven Level-1 sites require route updates; reschedule field operations away from (55.394, 10.489) for the next 24 h.”
Government	“Audit entry: 11 Level-2 detections at 2026-02-15 17:09:43; actions assigned to Regional Emergency Co-ordinator; municipal notification pending.”
Insurance	“Flooding score 3.003 at (55.394, 10.489); expected total-loss for ground-floor contents; highest claim priority.”

5 Expert Architectural Review

This section reports a structured expert review of the architecture. The goal is to assess the soundness of the architectural design and the novelty of the two-layer pattern from a technical perspective.

5.1 Method

The evaluation answers a deliberately scoped question: *do the architectural claims C1 and C2 (unidirectional invariant, persona-as-configuration) hold under expert architectural review?* It does not assess whether the generated reports are accurate, actionable, or trusted by domain stakeholders; that is left to a future study with agricultural domain experts. The panel composition (architecture, ML, edge, and security expertise) and the dimensions reflect this scope.

Where applicable, each dimension maps to ISO/IEC 25010 quality characteristics and subcharacteristics [10]: *D2* Separation of Concerns (modifiability), *D3* Extensibility (modifiability, flexibility), *D4* Practical Viability (operability, reliability), and *D5* Generalisability (flexibility, reusability). Each dimension probes one or more drivers of §3.1: *D2* tests D-Det and D-Rep, *D3* tests D-Stk and D-Sub, *D4* tests D-Cost and D-Conn, and *D5* tests transfer of the pattern as a whole. *D1* (Architectural Novelty) is an originality check rather than a driver.

We conducted a structured walk-through session with four researchers whose complementary expertise spans the key architectural concerns of the system: (E1) an ML expert with experience in model compression and edge deployment; (E2) a security and LLM expert specialising in prompt robustness and trust boundaries; (E3) an edge computing expert with a background in IoT architectures; and (E4) a simulation and human-in-the-loop expert focusing on decision-support interfaces.

The session had three phases: (1) a live walk-through of the dashboard, map, and report generation; (2) independent ratings on the five quality dimensions above using a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree); (3) a moderated open discussion of the architecture’s strengths, weaknesses, and possible improvements, recorded as written notes by two of the authors and analysed through open thematic coding, yielding the four themes in the next subsection.

5.2 Results

Table 5 presents the ratings as median and range. Means are omitted because they over-state precision with $N=4$ on a 5-point Likert. The highest-rated dimension was *D2* (Separation of Concerns, median 5, range 4 to 5), reflecting strong agreement that the unidirectional data-flow invariant effectively isolates deterministic and non-deterministic concerns. *D3* (Extensibility, median 4.5, range 4 to 5) was also rated favourably. *D1* (Architectural Novelty, median 4, range 4 to 5) received uniformly positive ratings. E3 noted that formalising the Persona pattern as an architectural concern was the most distinctive aspect. *D4*

(Practical Viability, median 3.5, range 3 to 4) received more moderate ratings, with E1 and E3 noting dependence on connectivity and operator training. D5 (Generalisability, median 4, range 3 to 4) was positive but E4 observed that demonstrating transfer requires instantiation in at least one additional domain. Inter-rater spread did not exceed one Likert point on any dimension, indicating consistent assessment despite the small panel.

Table 5. Expert panel ratings (1 to 5 Likert scale). E1: ML, E2: Security/LLM, E3: Edge, E4: Simulation/HiL.

Dimension	E1	E2	E3	E4	Median	Range
D1: Architectural Novelty	4	4	5	4	4	4 to 5
D2: Separation of Concerns	5	5	4	5	5	4 to 5
D3: Extensibility	5	4	4	5	4.5	4 to 5
D4: Practical Viability	3	4	3	4	3.5	3 to 4
D5: Generalisability	4	4	4	3	4	3 to 4

5.3 Qualitative Findings

Thematic coding of the open discussion yielded four themes.

Strength: Trust boundary clarity. All panellists commended the architectural separation between the safety-critical edge path and the advisory LLM layer. E2 highlighted that the no-write-back invariant is critical given prompt-injection and hallucination risks, since the generative layer cannot corrupt edge decisions when its output is consumed by a human, not by the control plane.

Strength: Stakeholder extensibility. E1 and E4 noted that the prompt-configuration-only mechanism yields a favourable effort-to-value ratio, keeping the data pipeline and UI invariant while varying only the prompt template.

Concern: Absence of stakeholder validation. All panellists identified the lack of end-user evaluation as the most significant limitation. E4 emphasised that expert assessment does not substitute for empirical evidence about report accuracy and actionability, and E1 suggested that even a small-scale study with three to five stakeholders per role would substantially strengthen the claims.

Concern: Real-time gap. E3 and E1 raised the batch-mode limitation, suggesting MQTT-based streaming as a priority extension. Further observations included prompt-template version-stamping and on-device vision-language models.

5.4 Threats to Validity

External validity. The architecture has been instantiated in a single domain (agricultural flood monitoring), and generalisation remains to be demonstrated.

Internal validity. The expert panel comprised four researchers whose complementary expertise covers the system’s key architectural concerns. A larger panel could surface additional issues, and panellists aware of the project goals may introduce positive bias. *Construct validity.* The five dimensions were grounded post hoc in ISO/IEC 25010 categories (§5.1). *Scope.* The panel validates architectural, not end-user, properties. None of the panellists are agricultural stakeholders, and ratings are not evidence about report accuracy, persona differentiation, or stakeholder trust.

6 Discussion

6.1 Architectural Trade-Offs

Each trade-off corresponds to a driver of §3.1. *Latency asymmetry* (D-Det): edge decisions are per-frame with bounded tail latency; LLM reports tolerate multi-second cloud-API latency because they serve post-hoc interpretation, not real-time hazard detection. *Cost control* (D-Cost): on-demand invocation keeps a typical 2000 to 3000-token report under one cent at GPT-5-mini [16] pricing as of Q1 2026. *Connectivity asymmetry* (D-Conn): the edge system operates entirely offline during field traversal, while the LLM layer requires connectivity only at the base station after a run. Substitutability of `llm_report` (§3.4) provides a migration path to local models when base-station connectivity is unreliable. *Hallucination containment* (motivates §6.2): the insurance excerpt of Table 4 extrapolates beyond the schema, and the unidirectional invariant keeps such fabrications out of the control plane.

6.2 Generative Reliability as an Architectural Concern

The unidirectional invariant (§3.2) prevents LLM non-determinism from corrupting the control plane, but it does not prevent the more frequent failure mode in this class of systems: a confidently wrong report that a stakeholder acts on. Hallucination, extrapolation beyond input fields, and ungrounded recommendations remain on the safety surface. Aligned with multi-layered guardrail proposals for foundation-model-based systems [24], we treat generative reliability as a property of the integration boundary rather than of the LLM itself, and locate four established mitigation patterns on the architecture in Figure 1:

- M1: Input grounding by construction.** `llm_report` sees only the structured JSON record schema (Table 1), so free-text context cannot shift the prompt distribution. This attaches at the data-injection step (§4, step 1) and is the only mitigation already realised in the prototype.
- M2: Field-level attribution.** Prompt templates can require the LLM to cite the source field for each numerical claim (e.g., “*combined score 3.003* → *flooded, source field scores.combined*”). Attaches at the prompt-template seam, changing the persona contract without touching the data pipeline.

M3: Schema- or grammar-constrained generation. Reports can be generated against a per-stakeholder output schema via guided generation that constrains LLM token sampling to a regular expression or context-free grammar [28]. Eliminates fabricated fields by construction. Attaches at the API-submission step (§4, step 3).

M4: Abstention on missing input. The dashboard can refuse to invoke the LLM when required JSON fields are absent or anomalies exceed plausibility bounds, surfacing a structured “insufficient input” notice instead of a confident report. Attaches at the on-demand invocation seam (§3.4).

We do not claim to have solved hallucination, and M1–M4 are not exhaustive: retrieval-augmented grounding [6,1] and runtime guardrail layering [24] attach at the same seams. The boundary admits each mitigation as a configuration or middleware change, which is what makes the pattern transferable to other context-aware, smart cyber-physical systems with similar reliability surfaces.

6.3 Future Work

The most significant gap, identified by the expert review (§5), is empirical validation with actual domain stakeholders. A future study with farmers, agronomists, farm managers, government-agency representatives, and insurers would assess factual accuracy against the JSON record, actionability, and role-targeting appropriateness of the generated reports. Beyond this primary study, the following extensions are scoped briefly.

Autonomous persona selection. The versioned template repository (C2) is a substrate for autonomic selection [11,12,9]: a MAPE-K loop scoring template metadata against authentication context and recent JSON state, with fallback to the general-overview persona on low confidence. The selection policy adapts; the templates and the C1/C2 invariants do not.

Streaming ingest. Streaming JSON over MQTT would enable live monitoring (weakening D-Conn) and address the panel’s real-time gap, provided lossy delivery does not threaten D-Rep.

Mitigation prototypes. Among the patterns of §6.2, M3 (schema-constrained generation [28]) and M4 (abstention) are the obvious next prototypes: largest reliability gain for smallest implementation cost. Either would convert the seams analysis into a working demonstration.

Multi-LLM comparison and template provenance. A multi-LLM accuracy/cost study is the natural empirical follow-up to the stakeholder study above. Because D-Sub treats model selection as a configuration boundary, it belongs to the follow-up. Content-hashing prompt templates alongside the JSON schema would further strengthen D-Rep provenance traceability.

7 Conclusion

We presented *persona-as-configuration*, an architectural pattern that pairs a deterministic edge plane with a generative stakeholder-reporting layer through two

invariants: unidirectional consumption of structured edge decision logs (C1), and stakeholder adaptation as a versioned prompt-template artifact, not runtime improvisation (C2). We instantiated the pattern over a previously published edge-based standing-water detection system [13] and analysed how the integration boundary admits four standard generative-reliability mitigations (M1 to M4) as configuration- or middleware-level extension points. A structured expert review scoped to the architectural claims and tied to the ISO/IEC 25010 quality model rated the pattern favourably, strongest on separation of concerns (median 5) and extensibility (median 4.5). End-user evaluation with agricultural stakeholders is planned for future work. The pattern targets context-aware, smart cyber-physical domains where edge devices produce structured decision logs for heterogeneous stakeholders, such as predictive maintenance, clinical decision support, and autonomous-vehicle fleet operations. Source code, prompt templates, and synthetic decision-log traces are publicly available at <https://github.com/Oliver1703dk/generative-reporting-for-agricultural-floods>.

References

1. An, T., Zhou, Y., Zou, H., Yang, J.: IoT-LLM: A framework for enhancing large language model reasoning from real-world sensor data. *Patterns* **7**(1), 101429 (2026). <https://doi.org/10.1016/j.patter.2025.101429>
2. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*. Addison-Wesley Professional, Boston, MA, 4 edn. (2021)
3. Bucaioni, A., Weyssow, M., He, J., Lyu, Y., Lo, D.: A functional software reference architecture for LLM-integrated systems. In: *Proc. IEEE ICSA 2025 Companion (New and Emerging Ideas)*. pp. 1-5 (2025). <https://doi.org/10.1109/ICSA-C65153.2025.00006>
4. Dauda, A., Flauzac, O., Nolot, F.: A survey on IoT application architectures. *Sensors* **24**(16), 5320 (2024). <https://doi.org/10.3390/s24165320>
5. De Vito, G., Palomba, F., Ferrucci, F.: The role of large language models in addressing IoT challenges: A systematic literature review. *Future Generation Computer Systems* **171**, 107829 (2025). <https://doi.org/10.1016/j.future.2025.107829>
6. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023)
7. González-Potes, A., Martínez-Castro, D., Paredes, C.M., Ochoa-Brust, A., Mena, L.J., Martínez-Peláez, R., Félix, V.G., Félix, R.A.: Hybrid AI and LLM-enabled agent-based real-time decision support architecture for industrial batch processes: A clean-in-place case study. *AI* **7**(2), 51 (2026). <https://doi.org/10.3390/ai7020051>
8. Hong, J., Hong, Y.G., de Foy, X., Kovatsch, M., Schooler, E., Kutscher, D.: Internet of Things (IoT) edge challenges and functions. RFC 9556 (Apr 2024)
9. IBM Corporation: An architectural blueprint for autonomic computing. White Paper G507-2065-00, IBM Corporation, Hawthorne, NY, USA (2006)
10. International Organization for Standardization: ISO/IEC 25010:2023, systems and software engineering, systems and software quality requirements and evaluation (SQuaRE), product quality model. Standard, ISO, Geneva, Switzerland (2023), edition 2

11. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. *Computer* **36**(1), 41-50 (2003). <https://doi.org/10.1109/MC.2003.1160055>
12. Krupitzer, C., Roth, F.M., VanSyckel, S., Schiele, G., Becker, C.: A survey on engineering approaches for self-adaptive systems. *Pervasive and Mobile Computing* **17**, 184-206 (2015). <https://doi.org/10.1016/j.pmcj.2014.09.009>
13. Larsen, O.A., Moghaddam, M.T.: Edge-based standing-water detection via FSM-guided tiering and multi-model consensus. In: *Proc. IEEE Int. Conf. Software Architecture (ICSA)*, In Practice Track (2026), <https://arxiv.org/abs/2604.03308>, arXiv:2604.03308
14. Miller, T., Mikiciuk, G., Durlik, I., Mikiciuk, M., Lobodzińska, A., Śnieg, M.: The IoT and AI in agriculture: The time is now, a systematic review of smart sensing technologies. *Sensors* **25**(12), 3583 (2025). <https://doi.org/10.3390/s25123583>
15. Naikwade, S.S., Nannaware, Y.S., Narkhede, A.A., Navale, P.S., Nikampatil, A.S., Bhaskar, T.: IoT-based flood level monitoring system with real-time alert and dashboard visualization. *J. Sensor and Cloud Computing* **2**(2) (2025)
16. OpenAI: GPT-5 mini model card. <https://developers.openai.com/api/docs/models/gpt-5-mini> (2025)
17. Plotly Technologies Inc.: Dash: Analytical web apps for Python. <https://dash.plotly.com/> (2024)
18. Reiter, E., Dale, R.: *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK (2000)
19. Rob Story and the Folium contributors: Folium: Python data, Leaflet.js maps. <https://python-visualization.github.io/folium/> (2024)
20. Samadi, S.: The convergence of AI, IoT, and big data for advancing flood analytics research. *Frontiers in Water* **4**, 786040 (2022). <https://doi.org/10.3389/frwa.2022.786040>
21. Schreiber, W., White, J., Schmidt, D.C.: Toward a pattern language for persona-based interactions with LLMs. In: *Proc. 31st Conf. on Pattern Languages of Programs (PLoP)*. ACM (2024). <https://doi.org/10.64346/PLoP2024p27>
22. Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., Gupta, A., Han, H., et al.: The prompt report: A systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608* (2024)
23. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F., Dennison, D.: Hidden technical debt in machine learning systems. In: *Proc. NIPS*. pp. 2503-2511 (2015)
24. Shamsujjoha, M., Lu, Q., Zhao, D., Zhu, L.: Swiss cheese model for AI safety: A taxonomy and reference architecture for multi-layered guardrails of foundation model based agents. In: *Proc. IEEE ICSA 2025*. pp. 37-48 (2025), <https://ieeexplore.ieee.org/document/10978931>
25. Tao, Y., Tian, B., Adhikari, B.R., Zuo, Q., Luo, X., Di, B.: A review of cutting-edge sensor technologies for improved flood monitoring and damage assessment. *Sensors* **24**(21), 7090 (2024). <https://doi.org/10.3390/s24217090>
26. Tseng, Y.M., Huang, Y.C., Hsiao, T.Y., Chen, W.L., Huang, C.W., Meng, Y., Chen, Y.N.: Two tales of persona in LLMs: A survey of role-playing and personalization. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. pp. 16612-16631 (2024). <https://doi.org/10.18653/v1/2024.findings-emnlp.969>
27. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382* (2023)

28. Willard, B.T., Louf, R.: Efficient guided generation for large language models. arXiv preprint arXiv:2307.09702 (2023)
29. Xu, F., Ma, J., Li, N., Cheng, J.C.P.: Large language model applications in disaster management: An interdisciplinary review. *Int. J. Disaster Risk Reduction* **127**, 105642 (2025). <https://doi.org/10.1016/j.ijdrr.2025.105642>
30. Xu, W., Liu, M., Sokolsky, O., Lee, I., Kong, F.: LLM-enabled cyber-physical systems: Survey, research opportunities, and challenges. In: *Proc. IEEE Int. Workshop on Foundation Models for Cyber-Physical Systems and IoT (FMSys)*, at CPS-IoT Week (2024), <https://ieeexplore.ieee.org/document/10590269>
31. Yu, P., Teng, F., Zhu, W., Shen, C., Chen, Z., Song, J.: Cloud-edge-device collaborative computing in smart agriculture: Architectures, applications, and future perspectives. *Frontiers in Plant Science* **16**, 1668545 (2025). <https://doi.org/10.3389/fpls.2025.1668545>
32. Zhang, X., Cao, Z., Dong, W.: Overview of edge computing in the agricultural Internet of Things: Key technologies, applications, challenges. *IEEE Access* **8**, 141748-141761 (2020). <https://doi.org/10.1109/ACCESS.2020.3013005>
33. Zheng, M., Pei, J., Logeswaran, L., Lee, M., Jurgens, D.: When “a helpful assistant” is not really helpful: Personas in system prompts do not improve performances of large language models. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. pp. 15126-15154 (2024). <https://doi.org/10.18653/v1/2024.findings-emnlp.888>