AN AI-READY SMART ADAPTER ARCHITECTURE FOR INTEGRATING HETEROGENEOUS AGRICULTURAL IOT SYSTEMS ACROSS THE IoT COMPUTING CONTINUUM

Dener Silva[1], Alexandre Heideker[1], Reinaldo Bianchi[1], Carlos Kamienski[2]
[1] Centro Universitário FEI, Brasil
[2] Universidade Federal do ABC, Brazil

A paper from the Proceedings of the
17th International Conference on Precision Agriculture and 11th Brazilian Congress on Precision and Digital Agriculture
13-15 July 2026
Porto Alegre, Brazil

Abstract.

The growing adoption of Internet of Things (IoT) technologies in smart agriculture has increased the heterogeneity of sensing infrastructures, communication protocols, and distributed computing resources. Integrating these heterogeneous environments while enabling controlled interaction with artificial intelligence (AI) services remains a significant challenge. This paper presents a Smart Adapter architecture designed to operate across the IoT Computing Continuum, providing a multiprotocol intermediary layer between agricultural data sources and IoT platforms. The proposed approach performs telemetry normalization, validation, contextual enrichment, and selective event generation while decoupling AI-assisted analysis from the primary data ingestion path. An experimental evaluation was conducted using realistic irrigation-management and anomaly-detection scenarios. The results demonstrate that the proposed architecture successfully integrates heterogeneous telemetry streams, maintains low local resource consumption, and enables selective AI-assisted orchestration without disrupting platform operations. The Smart Adapter provides a practical foundation for combining deterministic monitoring with contextual AI-assisted analysis in real-world smart agriculture deployments.

Keywords.
Smart Agriculture; Internet of Things; IoT Computing Continuum; Artificial Intelligence Integration;
Smart Adapters.

1 Introduction

The increasing adoption of Internet of Things (IoT) technologies in smart agriculture has enabled the deployment of distributed sensing infrastructures capable of monitoring environmental conditions, irrigation systems, soil characteristics, and crop-related operational variables in real

The authors are solely responsible for the content of this paper, which is not a refereed publication. Citation of this work should state that it is from the Proceedings of the 17th International Conference on Precision Agriculture and 11th Brazilian Congress on Precision and Digital Agriculture. EXAMPLE: Last Name, A. B. & Coauthor, C. D. (2026). Title of paper. In Proceedings of the 17th International Conference on Precision Agriculture and 11th Brazilian Congress on Precision and Digital Agriculture (unpaginated, online). Monticello, IL: International Society of Precision Agriculture and Brazilian Congress on Precision and Digital Agriculture.

time [1], [2]. Modern agricultural environments increasingly combine heterogeneous sensing devices, multiple communication technologies, and distributed computing infrastructures operating across the IoT Computing Continuum [3], [4]. In practical deployments, telemetry may originate from MQTT streams, LoRaWAN gateways, Zigbee devices, SDI-12 environmental sensors, and proprietary vendor-specific protocols [1], [5].

Despite the growing availability of agricultural IoT platforms, interoperability across heterogeneous agricultural infrastructures remains a significant challenge. Many IoT platforms provide efficient telemetry ingestion and device-management services but frequently rely on tightly coupled processing pipelines and platform-specific integration mechanisms [6]. This limitation complicates the integration of heterogeneous telemetry sources and restricts the incorporation of external intelligent analytics services without modifications to core platform components.

At the same time, recent advances in artificial intelligence (AI) and large language models (LLMs) have created new opportunities for contextual agricultural analytics and intelligent decision-support systems [7], [8]. AI-assisted agricultural systems can support irrigation management, anomaly detection, environmental monitoring, and contextual operational recommendations using both historical telemetry and real-time environmental conditions. However, continuously forwarding large volumes of telemetry to external AI services may introduce significant latency, infrastructure overhead, and operational costs, particularly in distributed edge–cloud agricultural environments.

This paper proposes a Smart Adapter architecture to integrate heterogeneous agricultural telemetry into a centralized IoT platform while keeping AI inference outside the main ingestion path. Positioned between field data sources and platform services, the adapter performs protocol abstraction, normalization, validation, contextual enrichment, and rule-based event detection before forwarding data downstream. Rather than coupling analytics directly to telemetry ingestion, the architecture triggers AI-assisted workflows only when operational conditions indicate that additional analysis is necessary.

We evaluate the architecture in irrigation-demand estimation, contextual environmental monitoring, and anomaly-detection scenarios. The evaluation addresses three questions: whether heterogeneous telemetry can be normalized and integrated consistently, whether the added processing remains lightweight on local infrastructure, and whether event-driven orchestration enables AI use without disrupting telemetry continuity. Unlike a previous study by the authors that directly compared structured control and LLM-based orchestration in a related smart irrigation testbed, the focus here is the intermediary architecture itself: its ability to absorb telemetry heterogeneity, preserve ingestion continuity, and support selective AI triggering with limited local overhead.

The main contributions of this paper are:

(i) a Smart Adapter architecture for integrating heterogeneous agricultural telemetry across the IoT Computing Continuum;

(ii) a selective AI-triggering mechanism that decouples telemetry ingestion from AI inference workflows; and

(iii) an experimental evaluation demonstrating interoperability preservation, low local infrastructure overhead, and AI-assisted recommendation support under realistic agricultural scenarios.

The remainder of this paper is organized as follows. Section 2 presents related work involving agricultural IoT interoperability, edge–cloud architectures, and AI-assisted agricultural systems. Section 3 describes the proposed Smart Adapter architecture and orchestration workflow. Section 4 presents the experimental methodology and evaluated scenarios. Section 5 discusses the experimental results, and Section 6 concludes the paper and discusses future research directions.

2 Related Work

2.1 Heterogeneous Agricultural IoT Environments

Smart agriculture environments are inherently heterogeneous due to the diversity of sensing devices, communication technologies, data formats, and deployment conditions found in real-world agricultural systems. Agricultural monitoring infrastructures may integrate weather stations, soil sensors, irrigation controllers, water-level monitors, and telemetry devices distributed across geographically dispersed areas. These devices often rely on distinct communication technologies, including LoRaWAN, Zigbee, Wi-Fi, serial interfaces, and domain-specific protocols such as SDI-12 [1], [9].

Although IoT platforms provide important services for device management, message routing, data ingestion, and storage, integrating heterogeneous agricultural devices often requires protocol-specific gateways, custom adapters, or platform-specific modifications [10]. This increases maintenance complexity and reduces portability across deployments. Data representation heterogeneity further complicates interoperability, since sensor measurements may use different naming conventions, units, timestamps, and payload structures. As a result, normalization mechanisms are essential to support downstream analytics, historical data management, and AI-assisted workflows.

2.2 IoT Computing Continuum and Smart Data Processing

The IoT Computing Continuum has emerged as an important architectural paradigm for distributed IoT systems, distributing computation across multiple layers such as edge devices, mist nodes, fog infrastructures, and cloud services [3], [5]. In agricultural environments, this distribution is particularly relevant because sensing infrastructures are geographically dispersed and often operate under intermittent connectivity, limited bandwidth, and constrained computational resources.

Edge and fog nodes positioned closer to sensors can perform lightweight preprocessing, filtering, contextual validation, and local event detection before forwarding data to centralized services [5], [6]. This strategy reduces unnecessary data movement, improves responsiveness, and supports scalable data processing across distributed infrastructures. However, deciding which processing functions should remain close to the data source and which should be delegated to cloud or AI services remains an open architectural challenge in agricultural IoT deployments [9], [4].

2.3 AI Integration in Agricultural IoT Systems

Artificial intelligence technologies are increasingly being explored in smart agriculture to support irrigation management, anomaly detection, crop monitoring, predictive analytics, and operational decision support. Recent advances in large language models have expanded these possibilities by enabling contextual interpretation, explanation, and recommendation generation from heterogeneous agricultural data [7], [8].

Despite this potential, directly integrating AI services into IoT ingestion pipelines introduces important challenges. AI inference is typically more computationally expensive and slower than conventional telemetry ingestion, especially when large models or remote cloud-based inference services are used [4], [7], [8]. Moreover, most agricultural telemetry represents stable operational conditions that do not require advanced AI reasoning. Continuously forwarding all sensor data to AI services may therefore increase latency, bandwidth usage, token consumption, and operational cost without necessarily improving decision quality.

Prior work has addressed IoT interoperability, IoT Computing Continuum, and AI applications in agriculture, but fewer studies examine how heterogeneous telemetry can be normalized, validated, enriched, and selectively routed to AI services without placing inference in the primary ingestion path. This paper addresses this gap by proposing an intermediary Smart Adapter that normalizes and validates heterogeneous telemetry before platform ingestion, while selectively escalating only operationally relevant events to AI services. In contrast to approaches that directly embed AI into the ingestion pipeline, the proposed architecture treats AI as an optional contextual analysis layer over deterministic event-driven monitoring.

3 Proposed Architecture

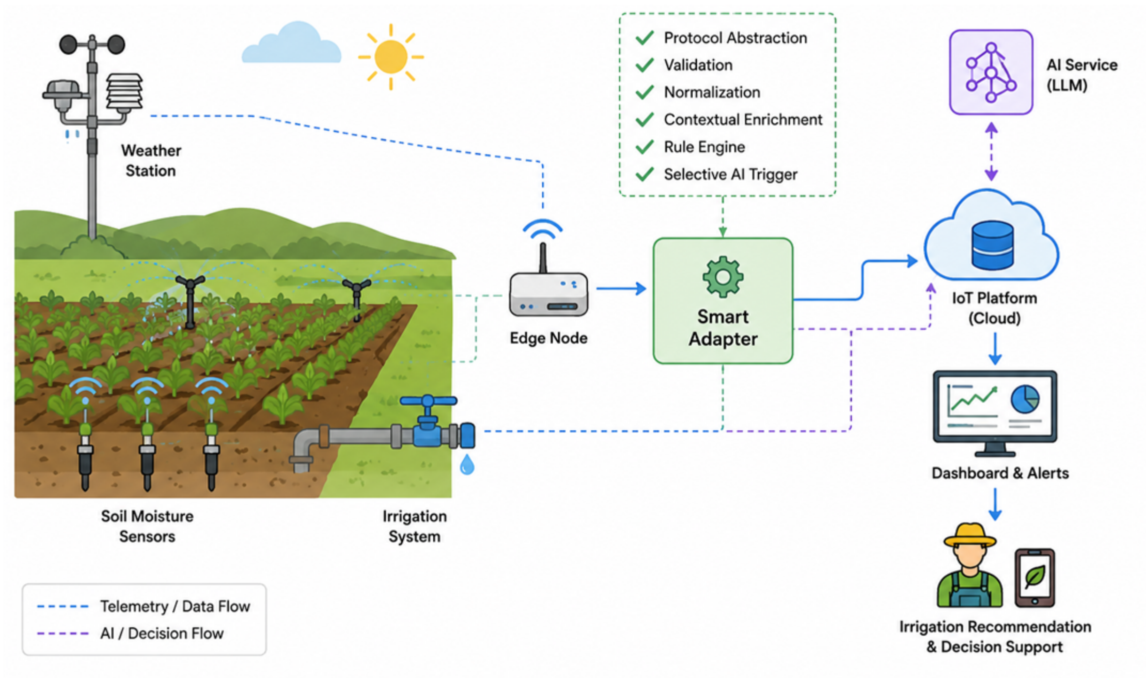

Figure 1 AI-ready smart agriculture architecture overview

3.1 Architectural Overview

Figure 1 presents the proposed architecture, which connects heterogeneous agricultural sensing environments to centralized IoT platform services and external AI-assisted decision support through an intermediary Smart Adapter operating across the IoT Computing Continuum.

The architecture is organized into four layers: (i) heterogeneous agricultural data sources, (ii) the Smart Adapter, (iii) IoT platform services, and (iv) optional AI-based analytics. Field devices generate raw environmental data through different communication technologies and payload formats. The adapter intercepts these streams and applies protocol abstraction, validation, normalization, contextual enrichment, and rule-based event analysis before sending processed data to the IoT platform.

In practice, this intermediary layer supports communication interfaces commonly found in agricultural deployments, including serial telemetry, MQTT streams, LoRaWAN gateways, Zigbee devices, and SDI-12 sensors. Its role is not merely to connect protocols, but to convert heterogeneous measurements into a unified representation that can be ingested by downstream services without platform-level changes.

After preprocessing, normalized telemetry is forwarded through standard ingestion interfaces to the IoT platform, which remains responsible for persistence, historical storage, visualization, and retrieval. AI services are invoked outside this path. When the adapter detects a relevant condition or anomaly, it emits an event that can trigger downstream analytical workflows.

In addition to environmental monitoring telemetry, the architecture also supports irrigation-related operational workflows. The irrigation subsystem may both provide operational telemetry and receive recommendation-oriented decision support information generated from contextual AI-assisted analysis. In the proposed implementation, AI services provide irrigation recommendations rather than fully autonomous actuation, preserving human supervision over agricultural operations.

This separation is central to the proposal because it prevents AI latency and inference variability from propagating into routine telemetry ingestion. As a result, the platform can continue storing and serving operational data even when external analytical services are slow or temporarily unavailable.

3.2 Smart Adapter Design

The Smart Adapter provides interoperability across heterogeneous agricultural sensing environments while introducing lightweight edge intelligence. Internally, it is organized into five stages: protocol integration, parsing and normalization, validation, contextual enrichment, and rule-based event generation.

The protocol integration stage handles communication with heterogeneous devices and gateways. Different communication adapters can be incorporated depending on the deployment requirements, allowing the ingestion of telemetry originating from serial interfaces, MQTT brokers, LoRaWAN infrastructures, Zigbee gateways, and other agricultural communication technologies. This modular design enables the integration of new protocols without modifications to platform services.

After reception, raw telemetry is processed by the parsing and normalization stage. Sensor-specific payloads are converted into a unified representation compatible with downstream services. In the proposed implementation, normalized telemetry is represented using SenML-compatible structures, allowing standardized representation of sensor measurements, timestamps, units, and contextual metadata.

The validation stage performs consistency verification and lightweight semantic checks before data forwarding. This stage verifies measurement ranges, malformed payloads, invalid sensor states, and anomalous operational conditions. Instead of silently forwarding inconsistent telemetry, the adapter can generate warning events and contextual notifications when invalid or suspicious measurements are detected.

Next, the contextual enrichment stage augments telemetry with additional metadata associated with sensors, deployment context, and operational conditions. This enrichment process may include sensor identifiers, measurement categories, scenario tags, protocol metadata, or deployment-specific attributes required for downstream analytics and AI inference workflows.

Finally, the rule-based event generation stage performs lightweight edge intelligence operations. Instead of forwarding all telemetry streams to AI services, the adapter evaluates configurable operational rules capable of identifying relevant situations such as threshold violations, abnormal environmental conditions, or sensor anomalies. When a relevant event is detected, the adapter generates a lightweight AI-trigger notification using MQTT-based messaging.This event-driven strategy enables external AI workflows to be invoked selectively while preserving the efficiency of routine telemetry ingestion.

3.3 Data Flow and AI Triggering Mechanism

The proposed data flow begins with telemetry generation at agricultural sensing devices. Raw measurements are transmitted to the Smart Adapter through protocol-specific communication channels. After normalization and validation, telemetry is forwarded to the IoT platform for storage and historical management.

Simultaneously, the adapter evaluates incoming telemetry against predefined operational rules. These rules may include threshold violations, environmental anomalies, irrigation-related conditions, or sensor consistency checks. If no relevant event is detected, telemetry continues through the normal ingestion pipeline without triggering additional processing stages.

When a rule violation occurs, the adapter publishes a lightweight MQTT alert containing contextual information about the detected event. These alerts are intentionally compact and contain only the information required to identify the operational condition and associated sensor context. This design minimizes communication overhead while enabling downstream orchestration services to react to relevant situations.

An external orchestration component subscribes to adapter-generated alerts and is responsible for interacting with AI inference services. Upon receiving an alert, the orchestrator retrieves relevant historical telemetry from the IoT platform, constructs contextual prompts, and selectively invokes external AI services when deeper analysis is required.

This architecture avoids continuous AI processing over raw telemetry streams. Instead, AI-assisted analysis is invoked only when edge-level rules identify operationally relevant conditions. This reduces unnecessary external inference requests, lowers communication overhead, and improves scalability across distributed edge–cloud environments.

3.4 AI-Ready Design Principles

The proposed architecture was designed following three main principles: interoperability, decoupled intelligence, and selective AI activation. Interoperability is achieved through protocol abstraction and unified data normalization mechanisms capable of integrating heterogeneous agricultural devices without requiring modifications to centralized platform services. The adapter isolates protocol-specific complexity from the remaining infrastructure, simplifying integration and deployment processes.

Decoupled intelligence refers to the separation between telemetry ingestion and AI inference workflows. AI services are treated as optional analytical components rather than mandatory elements of the ingestion path. This separation preserves operational stability even when AI services become unavailable, overloaded, or inaccessible.

Selective AI activation minimizes unnecessary computational and communication overhead by ensuring that AI services are triggered only when operational conditions justify additional analysis. Lightweight edge intelligence mechanisms operating inside the Smart Adapter identify contextually relevant events and selectively activate external AI workflows.

Together, these principles support scalable agricultural IoT infrastructures that can integrate heterogeneous telemetry and incorporate AI-assisted analysis without compromising operational efficiency.

4 Experimental Methodology

4.1 Experimental Infrastructure

The experimental environment was designed to reproduce a realistic distributed smart agriculture deployment operating across the IoT computing continuum. The objective of the experiments was not to evaluate raw platform scalability or stress-test the IoT infrastructure itself, but rather to analyze how the proposed Smart Adapter architecture enables controlled AI integration while preserving interoperability and decoupling heterogeneous agricultural systems from AI-specific processing workflows.

The experimental deployment used two Ubuntu-based virtual machines and one external AI inference node. VM1 hosted the core IoT infrastructure, including the Magistrala IoT platform, MQTT broker services, and the Smart Adapter responsible for telemetry normalization, validation, contextual enrichment, and selective alert generation. VM2 hosted the agricultural scenario simulator, which emulated heterogeneous telemetry sources and environmental conditions. AI inference was executed remotely through an Ollama deployment running the qwen3:8b model.

VM1 was configured with 4 vCPUs and 8 GB RAM, while VM2 used 2 vCPUs and 4 GB RAM. The distributed deployment was intentionally maintained lightweight in order to reproduce realistic edge/fog agricultural environments where computational resources are limited and AI services may operate remotely through constrained communication paths.

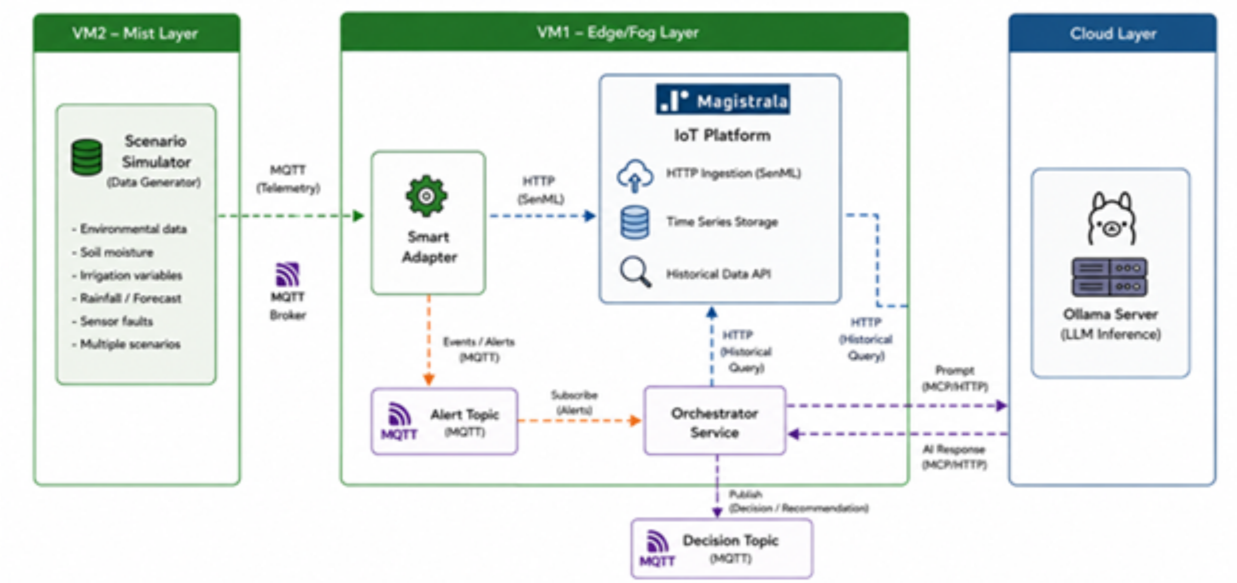

Figure 2 Experimental testbed architecture used during the evaluation, including distributed scenario generation, Smart Adapter processing, Magistrala IoT services, event-driven orchestration, and remote AI inference through an Ollama-based LLM service.

Figure 22 illustrates the experimental testbed architecture used during the evaluation, including distributed scenario generation, Smart Adapter processing, Magistrala IoT services, event-driven orchestration, and remote AI inference through an Ollama-based LLM service.

The agricultural scenario simulator continuously generated telemetry streams representing heterogeneous agricultural devices and environmental measurements. Telemetry was transmitted through MQTT topics and consumed by the Smart Adapter, which transformed heterogeneous payloads into normalized SenML representations before forwarding the resulting telemetry into Magistrala through HTTP ingestion endpoints.

The Smart Adapter implemented local validation, contextual enrichment, and rule-based filtering mechanisms. Instead of forwarding all telemetry to AI services, the adapter selectively generated lightweight MQTT alerts whenever relevant operational conditions or anomalies were detected. This design minimizes unnecessary AI invocations and reduces both computational cost and external inference usage. In practical deployments, continuously forwarding all telemetry to external large language model services would significantly increase operational costs, token consumption, and end-to-end latency.

Although the current evidence is reported from controlled experiments using Magistrala, the Smart Adapter was not conceived as a purely hypothetical component. It was derived from a production-oriented multiprotocol intermediary already used in the SMART project to connect heterogeneous telemetry sources to centralized platform services. The present paper does not report a longitudinal field evaluation of that operational deployment; instead, it uses controlled experiments to assess the architectural feasibility, local overhead, and selective AI-triggering behavior of the intermediary design.

4.2 Agricultural Scenario Definition

The experiments focused on realistic smart agriculture irrigation-management scenarios involving environmental monitoring, contextual irrigation analysis, anomaly detection, and AI-assisted recommendation workflows. The historical agricultural telemetry datasets were configured based on previous experience with smart irrigation and agricultural IoT deployments, including SWAMP, COSMIC-SWAMP, and the ongoing SMART project [14]–[16]. These initiatives provided practical reference points for defining the variables, telemetry patterns, and operational conditions represented in the generated scenarios. Historical datasets were generated and persisted inside the IoT platform before the execution of the experiments. The resulting datasets emulate environmental variation and irrigation-related operational conditions over time, including air temperature, relative humidity, atmospheric pressure, wind speed, rainfall indicators, soil moisture conditions, and evapotranspiration-related variables.

In addition, contextual agricultural indicators derived from the FAO-56 Penman–Monteith methodology were incorporated into the generated datasets in order to provide realistic evapotranspiration-based irrigation context. Instead of relying exclusively on threshold-based irrigation rules, the experimental scenarios incorporated contextual variables such as estimated reference evapotranspiration (ET_0), crop evapotranspiration indicators, effective rainfall estimates, and baseline irrigation depth recommendations.

Multiple operational scenarios were modeled in order to evaluate distinct agricultural conditions and AI-triggering behaviors. The evaluated scenarios are summarized in Table 1.

Table 1. Agricultural operational scenarios evaluated during the experiments.

Scenario	Operational Condition	Expected Behavior	AI Trigger
C1	High ETc, dry soil, and no rainfall forecast	IRRIGATE_HIGH	Yes
C2	Moderate ETc and moderate soil moisture	IRRIGATE_MEDIUM	Yes
C3	Rain forecast detected	NO_IRRIGATION	Yes
C4	Stable environmental conditions	NO_IRRIGATION	Minimal or none
C5	Invalid environmental sensor values	SENSOR_CHECK	Yes
C6	Inconsistent telemetry measurements	SENSOR_CHECK	Yes
C7	High-frequency telemetry without anomaly	No AI invocation expected	No

These scenarios were designed to reflect plausible agricultural operating conditions rather than synthetic benchmark workloads. This choice allowed the evaluation to examine interoperability, contextual retrieval, selective AI activation, and recommendation generation under conditions closer to deployment practice.

The methodology also preserves temporal and contextual continuity across experimental executions. Instead of regenerating entirely independent historical datasets for each run, the experiments maintained a persistent historical telemetry base while selectively modifying the most recent environmental conditions and operational states associated with each scenario. This allows the orchestration layer and AI services to operate over consistent historical agricultural context while still evaluating different operational situations such as irrigation-demand variation, rainfall conditions, and anomalous sensor behavior.

4.3 Experimental Workflow

The Smart Adapter performs parsing, normalization, contextual enrichment, validation, and rule-based event analysis before forwarding telemetry into Magistrala. During validation, it evaluates incoming telemetry against predefined operational conditions, including invalid sensor ranges, anomalous environmental values, irrigation-related conditions, and contextual inconsistencies.

Whenever a relevant operational condition is detected, the Smart Adapter generates a lightweight MQTT alert describing the event in a structured format. Instead of forwarding raw telemetry alone, the alert already includes contextual information about the affected entity, the detected condition, measured values, threshold violations, timestamps, and anomaly characteristics. This allows downstream orchestration and AI services to operate over structured contextual events rather than heterogeneous raw payloads.

An orchestration component subscribes to these alert topics and coordinates the subsequent AI-assisted analysis workflow. Upon receiving an alert, the orchestrator retrieves recent historical telemetry associated with the affected agricultural entity from Magistrala through controlled integration interfaces. In the evaluated implementation, the Model Context Protocol (MCP) was used as a structured contextual retrieval interface between the orchestrator and the platform telemetry services, preventing the inference logic from directly depending on platform-specific APIs.

The retrieved telemetry is transformed into a structured contextual prompt containing environmental conditions, evapotranspiration indicators, rainfall information, irrigation-related variables, and historical operational context. The generated prompt is selectively forwarded to the external Ollama deployment for contextual agricultural analysis, and the resulting AI response is then published to a dedicated MQTT topic containing irrigation recommendations and decision-support information. The execution of irrigation actions themselves remains outside the scope of the Smart Adapter architecture.

The proposed architecture intentionally separates AI-assisted analysis from agricultural actuation workflows. In the evaluated implementation, downstream applications may consume recommendation topics and translate irrigation recommendations into operational actions according to deployment-specific requirements. This preserves deployment flexibility across edge, fog, and cloud environments.

A key property of the workflow is that AI inference remains fully decoupled from the primary telemetry ingestion path. As a result, delays in AI processing do not interrupt telemetry normalization, forwarding, or persistence within the IoT infrastructure.

4.4 Evaluation Metrics and Factors

The evaluation focused on interoperability behavior, selective AI activation, performance overhead, recommendation consistency, and AI integration readiness. Table 2 summarizes the evaluated metrics and experimental factors considered during experiments.

Table 2. Evaluated factors and metrics considered during the experiments.

Category	Metric	Description
Interoperability	Successful normalization rate	Percentage of heterogeneous payloads successfully normalized into unified SenML representations
Performance	Context retrieval latency	Time required to retrieve historical telemetry from the IoT platform
Performance	LLM inference latency	Time required for remote AI inference execution
Performance	End-to-end latency	Total alert-to-decision delay
Reliability	Recommendation consistency	Agreement between AI recommendation and expected scenario behavior
Reliability	Invalid response rate	Percentage of malformed or inconsistent AI outputs
Resource Usage	CPU utilization	CPU consumption during orchestration and AI-assisted workflows
Resource Usage	RAM utilization	Memory consumption during orchestration and telemetry processing
AI Efficiency	Selective invocation rate	Percentage of telemetry streams forwarded to AI services

Performance overhead was evaluated through end-to-end latency measurements collected across multiple stages of the workflow. The measured intervals included: 1) telemetry injection to Smart Adapter event detection; 2) alert publication to orchestration reception; 3) contextual retrieval time from the IoT platform; 4) prompt construction overhead; 5) LLM inference latency; 6) decision publication delay; and 7) complete alert-to-decision time.

The experiments additionally evaluated irrigation recommendation consistency across predefined agricultural scenarios. Expected irrigation behaviors were defined according to the generated environmental conditions and Penman–Monteith-derived contextual variables. AI-generated recommendations were then compared against the expected operational behavior associated with each scenario.

Because the proposed architecture relies on selective AI activation rather than continuous inference, the experiments also assessed whether the Smart Adapter reduced unnecessary external AI calls by generating alerts only under operationally relevant conditions. This aspect is particularly important in distributed edge–cloud deployments, where remote inference can introduce substantial latency and cost.

An important architectural characteristic of the proposed approach is the coexistence of deterministic event-processing mechanisms and selective AI-assisted orchestration. The Smart Adapter can independently generate threshold-based alerts, anomaly notifications, and operational events without necessarily triggering external AI inference. In the evaluated workflow, only operationally relevant events are escalated to the AI orchestration layer.

This design enables lightweight deterministic monitoring to coexist with contextual AI-assisted analysis. As a result, routine operational conditions may be handled locally through structured rules and alerts, while more complex or context-dependent situations can selectively trigger LLM-based reasoning. Importantly, all generated events remain persisted within the historical telemetry infrastructure, allowing future contextual analysis even when immediate AI escalation is not performed. This makes the architecture suitable for hierarchical decision workflows, where simple events are handled locally and complex situations are escalated to AI-assisted analysis.

4.5 Statistical Analysis

All experimental scenarios were executed repeatedly under the same infrastructure conditions in order to reduce the impact of transient system variations and stochastic inference fluctuations. Each experimental scenario was executed thirty times, and all reported results include 95% confidence intervals. Latency measurements, resource consumption values, and irrigation recommendation consistency results are therefore presented as mean values accompanied by confidence intervals, enabling the analysis of variability and execution stability across experimental runs.

The statistical analysis focused primarily on: 1) variability of end-to-end latency; 2) inference delay fluctuations under different operational conditions; 3) resource consumption stability; 4) consistency of irrigation-related recommendations; and 5) variability introduced by remote AI inference execution. This methodology enables the evaluation of execution stability, latency variability, and recommendation consistency across repeated AI-assisted orchestration workflows while preserving reproducibility and statistical validity.

5 Results and Discussion

5.1 Experimental Scenario Validation

Across all evaluated scenarios, the architecture consistently integrated heterogeneous telemetry into the IoT platform. The adapter normalized incoming payloads into SenML-compatible representations before ingestion, showing that protocol diversity and payload variability can be absorbed at the intermediary layer rather than at the platform core.

The evaluated scenarios reproduced distinct agricultural conditions involving irrigation demand, rainfall-aware decision context, environmental anomalies, and sensor inconsistencies. More importantly, they showed that AI-assisted orchestration can be reserved for situations that carry operational relevance, instead of being applied continuously to all incoming telemetry. This behavior is central to the proposed architecture because it demonstrates that selective triggering is not only a design principle, but an executable mechanism validated under multiple conditions.

The experiments also confirmed that telemetry ingestion remained operationally independent from AI execution throughout all runs. Even when inference took longer, ingestion, normalization, and persistence continued without interruption. This result matters because it validates the architectural decision to keep AI outside the primary data path: the system continues to fulfill its monitoring role even when analytical services are slower or less predictable.

No telemetry normalization failures were observed across the evaluated executions. All heterogeneous payloads generated by the experimental scenarios were successfully translated into SenML-compatible representations before persistence within Magistrala. This result indicates that the intermediary adapter layer was capable of absorbing protocol and payload variability without requiring modifications to the IoT platform ingestion pipeline.

No malformed AI responses or structurally invalid orchestration outputs were observed across the evaluated runs. All generated recommendations remained compatible with the expected orchestration workflow and decision-publication mechanism.

5.2 End-to-End Latency Analysis

The latency analysis was used to distinguish the cost of local orchestration from the cost of remote AI execution. Rather than asking only whether the workflow is fast, this analysis clarifies where delay is introduced and whether the proposed intermediary layer becomes a bottleneck. The measured intervals therefore make it possible to separate integration overhead from inference overhead across the full alert-to-decision path.

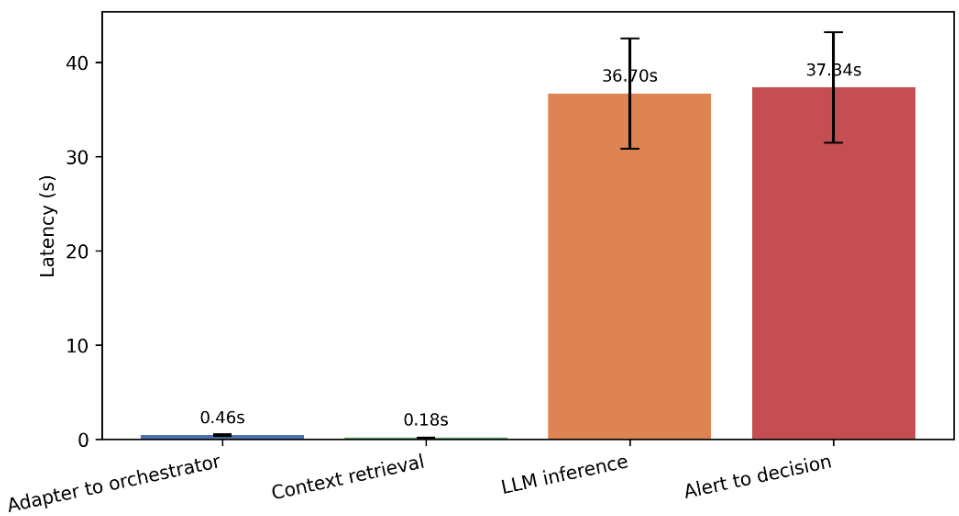

Figure 3 Latency breakdown across the AI-assisted agricultural orchestration pipeline.

As illustrated in Figure 3, the adapter introduced only a small fraction of the total workflow delay. Adapter-to-orchestrator communication averaged 0.46 ± 0.10 s, context retrieval averaged 0.18 ± 0.01 s, while remote LLM inference averaged 36.70 ± 5.87 s. Taken together, these results indicate that neither normalization nor orchestration handoff is the principal constraint of the evaluated workflow.

The dominant latency component was remote LLM inference. This means that the main performance trade-off of the architecture is not introduced by the integration layer, but by the decision to invoke external AI analysis when relevant events occur. That distinction is important: it suggests that future optimization efforts should prioritize inference placement, model choice, and orchestration strategy rather than further minimizing telemetry normalization overhead, which was already comparatively small.

The measurements therefore support the architectural claim that AI-assisted analysis can be added without destabilizing routine telemetry handling. Because inference remains outside the ingestion path, variability in AI response time does not propagate into persistence and monitoring functions. In practical terms, this preserves the usefulness of the IoT infrastructure even when analytical services operate with higher or less predictable latency.

Across the evaluated runs, the complete alert-to-decision delay remained sufficiently stable for the tested scenarios, despite the expected variability of remote large language model inference. The workflow also produced contextual recommendations using historical telemetry retrieved from the platform, indicating that the event-driven design is able to combine stored operational context with on-demand AI analysis when a relevant condition is detected.

5.3 Local Infrastructure Resource Consumption

The experiments additionally evaluated CPU and memory consumption associated with the local orchestration infrastructure. Because AI inference was executed remotely through an external Ollama deployment, these measurements isolate the local cost of telemetry ingestion, normalization, contextual processing, and orchestration coordination. This distinction is important because it separates the computational footprint of the proposed architecture from the substantially heavier cost of model execution.

Local resource consumption remained stable across the evaluated scenarios. Because the scenarios primarily changed contextual telemetry conditions and prompt-construction workflows, the effect on CPU and memory usage was limited.

The Magistrala infrastructure represented the dominant local resource consumer due to telemetry ingestion, persistence, API processing, and database-related operations. Across the evaluated executions, the Magistrala stack presented mean CPU utilization between 37.6% ± 7.5% and 39.0% ± 8.1%, while mean memory consumption remained between 1.74 ± 0.02 GiB and 1.81 ± 0.03 GiB (95% confidence interval).

The Smart Adapter maintained low operational overhead throughout all evaluated scenarios. Mean CPU utilization remained between 3.1% ± 0.9% and 3.4% ± 1.1%, while memory consumption remained between 34.0 ± 1.3 MiB and 36.0 ± 1.4 MiB during telemetry normalization, contextual enrichment, validation, and selective event-generation workflows (95% confidence interval).

The orchestration component responsible for contextual retrieval and AI workflow coordination also remained lightweight during the experiments. Mean CPU utilization remained below 1% (0.6% ± 0.2%), while memory consumption remained between 28.0 ± 0.2 MiB and 29.0 ± 0.3 MiB across the evaluated executions (95% confidence interval).

Overall, the resource measurements indicate that the proposed architecture can support event-driven AI orchestration with limited local overhead. The main practical implication is that introducing the adapter does not materially shift infrastructure cost toward the edge or fog layer under the evaluated conditions. Instead, the dominant cost remains associated with remote AI execution, reinforcing the usefulness of selective triggering as a mechanism for controlling when that cost is incurred.

5.4 Discussion

Taken together, the results suggest that the proposed architecture can address two recurring constraints in agricultural IoT deployments: heterogeneous telemetry integration and the need to use AI selectively rather than continuously. Within the evaluated scenarios, the intermediary layer absorbed protocol and payload diversity, while the event-driven workflow limited AI invocation to operationally significant conditions.

The main trade-off revealed by the evaluation is that remote inference, not the adapter, dominates end-to-end latency. This indicates that performance tuning in similar deployments should prioritize inference placement, model choice, and orchestration policy, while the separation of AI from ingestion preserves monitoring continuity even when analytical services are slower or less predictable.

Selective triggering proved to be a practical control mechanism: stable operating conditions did not generate unnecessary external inference requests, while relevant thresholds, anomalies, and contextual inconsistencies did. The experiments confirmed that AI invocation occurred only after adapter-generated alerts, rather than for every telemetry message persisted in the platform. Consequently, routine telemetry remained within the standard ingestion workflow, while only operationally relevant situations triggered contextual AI-assisted analysis.

A further contribution of the architecture is the combination of deterministic monitoring with selective AI-assisted escalation. Routine or low-impact conditions can be handled through local rules and alerts, while more complex situations can trigger deeper contextual analysis without replacing existing monitoring mechanisms. An important implication of these results is that the proposed architecture does not require replacing deterministic monitoring mechanisms with AI-based reasoning. Instead, deterministic rules can provide fast, low-cost detection of known operational conditions, while AI services can be selectively activated for contextual interpretation, recommendation generation, or anomaly analysis. This hybrid strategy is particularly suitable for agricultural IoT deployments, where most telemetry is routine but selected events may benefit from deeper contextual reasoning.

AI invocation occurred only after adapter-generated alerts. Stable telemetry conditions did not trigger external inference requests, confirming that the proposed architecture can selectively escalate only operationally relevant situations to AI-assisted analysis while preserving routine telemetry processing within the standard ingestion path.

6 Limitations and Threats to Validity

The results reported in this paper should be interpreted as evidence of architectural feasibility under controlled conditions rather than as a definitive benchmark across all agricultural IoT deployments. The evaluated scenarios were designed to isolate interoperability behavior, local overhead, and selective AI-triggering dynamics in operationally motivated situations, but they do not cover the full variability of field conditions, device failures, network instability, or long-term deployment effects.

The interoperability evidence presented here is representative but not exhaustive. Although the Smart Adapter was derived from a production-oriented multiprotocol component already used in the SMART project, the formal evidence reported in this paper comes from controlled experiments rather than from a longitudinal field study. Accordingly, the results support the plausibility of the intermediary design, but broader validation across additional devices, vendor-specific payloads, and deployment conditions remains necessary.

A direct comparison between structured control and LLM-based orchestration was previously explored by the authors in a related smart irrigation study [16]. The present paper does not repeat that full comparison because its objective is narrower: to evaluate the intermediary architecture for heterogeneous telemetry integration and selective AI triggering. In addition, the reported results reflect one orchestration design, one external inference setup, and one model configuration. Different models, placements, and network conditions may lead to different latency, cost, and reliability profiles.

7 Conclusion and Future Work

This paper presented a Smart Adapter architecture for integrating heterogeneous agricultural IoT systems across the IoT Computing Continuum while keeping AI inference outside the primary telemetry ingestion path. The intermediary layer performs protocol abstraction, normalization, validation, contextual enrichment, and event generation before data reaches centralized IoT services.

The evaluation indicated that the architecture preserves interoperability across the heterogeneous telemetry sources represented in the controlled scenarios while supporting contextual agricultural workflows for irrigation, rainfall-aware decisions, and anomaly detection. Selective AI triggering limited external inference to operationally relevant situations, and performance results indicated that the dominant source of end-to-end delay was remote large language model inference rather than the integration layer itself.

By isolating AI workflows from telemetry ingestion, the architecture preserves monitoring continuity even when analytical services are slower or temporarily unstable. The results also indicate that contextual variables derived from the FAO-56 Penman–Monteith methodology can be incorporated coherently into AI-assisted irrigation workflows.

Future work includes adaptive orchestration mechanisms that dynamically choose between local, fog, and cloud-based inference according to latency, connectivity, compute availability, and cost. Additional directions include multi-agent coordination, distributed AI orchestration, integration with Model Context Protocol (MCP)-based workflows, and broader evaluation with anomaly detection, predictive analytics, and more autonomous irrigation scenarios.

Overall, the reported results support the feasibility of the proposed Smart Adapter as an architectural pattern for agricultural IoT systems that must integrate heterogeneous telemetry while using AI in a controlled manner. Its main contribution is not simply to connect devices, but to separate interoperability concerns from analytical escalation in a way that preserves operational continuity and keeps AI-related cost under explicit control. These results suggest that AI-ready agricultural infrastructures should not rely on continuous AI inference, but rather on intermediary layers capable of combining deterministic monitoring, historical telemetry persistence, and selective AI-assisted escalation.

Acknowledgments

This work was funded by the São Paulo Research Foundation (FAPESP), grants #2024/15527-9 and #20/05152-7. It was also supported by the National Natural Science Foundation of China (NSFC) under the SMART project.

References

[1] T. Domínguez-Bolaño, O. Campos, V. Barral, C. J. Escudero, and J. A. García-Naya, “An overview of IoT architectures, technologies, and existing open-source projects,” Internet of Things, vol. 20, p. 100626, Nov. 2022, doi: 10.1016/j.iot.2022.100626.

[2] A. K. Saini, A. K. Yadav, and Dhiraj, “A Comprehensive review on technological breakthroughs in precision agriculture: IoT and emerging data analytics,” European Journal of Agronomy, vol. 163, p. 127440, Feb. 2025, doi: 10.1016/j.eja.2024.127440.

[3] A. Al-Dulaimy et al., “The computing continuum: From IoT to the cloud,” Internet of Things, vol. 27, p. 101272, Oct. 2024, doi: 10.1016/j.iot.2024.101272.

[4] F. B. Oliveira, M. Di Felice, and C. Kamienski, “IoTDeploy: Deployment of IoT Smart Applications over the Computing Continuum,” Internet of Things, vol. 28, p. 101348, Dec. 2024, doi: 10.1016/j.iot.2024.101348.

[5] L. Kong et al., “Edge-computing-driven Internet of Things: A Survey,” ACM Comput. Surv., vol. 55, no. 8, pp. 1–41, Aug. 2023, doi: 10.1145/3555308.

[6] M. Noura, M. Atiquzzaman, and M. Gaedke, “Interoperability in Internet of Things: Taxonomies and open challenges,” Mob. Netw. Appl., vol. 24, no. 3, pp. 796–809, Jun. 2019, doi: 10.1007/s11036-018-1089-9.

[7] M. T. Kuska, M. Wahabzada, and S. Paulus, “AI for crop production – Where can large language models (LLMs) provide substantial value?,” Comput. Electron. Agric., vol. 221, p. 108924, Jun. 2024, doi: 10.1016/j.compag.2024.108924.

[8] D. De Clercq, E. Nehring, H. Mayne, and A. Mahdi, “Large language models can help boost food production, but be mindful of their risks,” Front. Artif. Intell., vol. 7, Oct. 2024, doi: 10.3389/frai.2024.1326153.

[9] D. Urdu et al., “Aligning interoperability architectures for digital agri-food platforms,” Comput. Electron. Agric., vol. 224, p. 109194, Sep. 2024, doi: 10.1016/j.compag.2024.109194.

[10] E. Roccatello, A. Pagano, N. Levorato, and M. Rumor, “State of the Art in Internet of Things Standards and Protocols for Precision Agriculture with an Approach to Semantic Interoperability,” Network, vol. 5, no. 2, p. 14, Apr. 2025, doi: 10.3390/network5020014.

[11] C. Kamienski, J.-P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T. Salmon Cinotti, R. Filev Maia, and A. Torre Neto, “Smart Water Management Platform: IoT-based precision irrigation for agriculture,” Sensors, vol. 19, no. 2, p. 276, 2019, doi: 10.3390/s19020276.

[12] P. Stowell, C. Kamienski, A. Torre-Neto, and L. Rabello, “COSMIC-SWAMP: Internet of Things Irrigation Platform using Neutron and Capacitive Probes,” 2025 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Bologna, Italy, 2025, pp. 401–405, doi: 10.1109/MetroAgriFor66923.2025.11512396.

[13] D. Silva, A. Heideker, E. Ferreira, C. Kamienski, and R. Bianchi, “Controle Interoperável no Contínuo Computacional de IoT com Agentes de IA,” in *Anais do XLIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2026, pp. 477–490, doi: 10.5753/sbrc.2026.19803.