

Design and Formalization of a Distributed Intelligent Edge-Fog-Cloud Architecture for Industry 4.0 Systems

Hafsa AMALIK*, Hicham MEDROMI

Research Laboratory in Networks-Computer Science-Telecommunications & Multimedia (RITM)-System Architecture Team (EAS), National High School of Electricity and Mechanics (ENSEM), Hassan II University, Casablanca, Morocco

Abstract—Industry 4.0 systems increasingly require distributed decision-making capabilities to manage heterogeneous resources, large volumes of data, and time-sensitive operational constraints. This study proposes a distributed intelligent Edge-Fog-Cloud architecture combining a structural seven-component intelligent-entity model, a distributed cognition mechanism based on the Perceive, Analyze, Decide, and Act (PADA) cycle, hierarchical coordination, and graceful degradation. The contribution lies in the consistent integration of these mechanisms across Edge, Fog, and Cloud decision levels. The framework is illustrated through a proposed deployment scenario derived from the OCP Benguerir phosphate-processing facilities. The present study provides an architectural and qualitative analysis; it does not report physical deployment, prototype, simulation, or quantitative performance evaluation. Responsiveness, scalability, availability, and resilience are therefore treated as design objectives that require future empirical validation.

Keywords—Industry 4.0; distributed intelligence; edge-fog-cloud architecture; cyber-physical systems; PADA; multi-agent systems; graceful degradation; IIoT

I. INTRODUCTION

Industry 4.0 is reshaping industrial systems through the integration of Cyber-Physical Systems, Industrial Internet of Things technologies, and intelligent data-driven services [1], [2]. Modern industrial environments are now highly connected. They generate large volumes of operational data and demand decisions that are both faster and more adaptive. Traditional centralized architectures struggle to meet these requirements, particularly in terms of responsiveness, scalability, and resilience [3], [4].

Research on the digital transformation of industry has followed three main directions. The first builds on the convergence of Cyber-Physical Systems (CPS), the Industrial Internet of Things (IIoT), and cloud services [1], [2]. By tightly coupling physical assets with computational and communication capabilities, CPS provide a solid foundation for smart manufacturing. They enable real-time monitoring, process automation, and adaptive control [5], [6]. The later integration of cloud platforms and digital twin technologies extended these capabilities toward global supervision and long-term optimization [7], [8].

The second direction emerged as industrial environments grew more complex. Multi-Agent Systems (MAS) appeared as a natural answer to the limitations of centralized control. They

offer distributed decision-making and cooperative behavior among heterogeneous resources [9], [10]. Despite this flexibility, their integration with IIoT infrastructures and their ability to sustain consistent multi-level coordination remain limited at industrial scale [10], [11].

These limitations motivated a third direction, based on Edge-Fog-Cloud architectures. Such architectures distribute computation across several layers. Processing is brought closer to physical assets, which reduces latency and improves scalability [3], [4], [12]. Their effectiveness has been demonstrated in predictive maintenance and smart manufacturing applications [8], [13]. However, most of these solutions remain centered on communication and data processing infrastructure. The organization of distributed intelligence, the formal representation of intelligent entities, and the integration of resilience mechanisms are rarely treated as first-class architectural concerns [5], [7]. This gap motivates the framework proposed in this work. A detailed positioning of the proposed architecture with respect to these approaches is provided in Section IV.

Recent studies have extended this continuum through collaborative task offloading, cloud-fog-edge reference frameworks, implementation-oriented industrial architectures, resilience analyses, and quantitative performance models [14], [15], [16], [17], [18]. These contributions confirm the importance of distributing computation and coordination across heterogeneous tiers. Nevertheless, explicit entity abstractions, cognition cycles, decision subsidiarity, and graceful degradation are not always integrated within a single architectural representation, which motivates the structural framework developed in this study.

This study proposes a distributed intelligent architecture built on four pillars: a formal model of intelligent entities, a distributed cognition mechanism based on the PADA cycle (Perceive, Analyze, Decide, Act), hierarchical coordination, and resilience through graceful degradation. The applicability of the framework is illustrated through a proposed deployment scenario derived from the OCP Benguerir phosphate-processing facilities. No physical deployment or quantitative validation is claimed.

The main contributions are: 1) a multi-layer Edge-Fog-Cloud architecture where intelligence is an intrinsic component; 2) a formal seven-component entity model; 3) a distributed PADA cognition model; 4) hierarchical coordination and

*Corresponding author.

resilience mechanisms; and 5) an industrially grounded deployment scenario based on the OCP Benguerir phosphate-processing facilities.

The remainder of the study is organized as follows. Section II presents the proposed architecture. Section III presents the industrial case study and the proposed deployment scenario. Section IV provides a comparative analysis with existing approaches. Section V discusses the findings, and Section VI concludes the study.

II. PROPOSED DISTRIBUTED INTELLIGENT ARCHITECTURE

A. Design Principles

The proposed architecture is built around three design principles that address the limitations of existing Industry 4.0 frameworks [3], [4], [5].

The first principle is distributed responsibility, where perception, analysis, decision-making, and action capabilities are distributed among autonomous entities across all system levels. This organization improves responsiveness and reduces dependence on centralized infrastructures, which are increasingly unable to meet the real-time requirements of modern industrial environments [11].

The second principle is multi-level coordination, whereby local decisions remain aligned with broader operational objectives through structured coordination mechanisms. This approach preserves the flexibility required for local autonomy while ensuring consistency with system-wide goals [5].

The third principle is interoperability and modularity, through which the framework relies on standardized communication protocols and a modular deployment strategy. This facilitates progressive integration within heterogeneous industrial environments that typically combine legacy equipment with modern digital technologies [12].

Based on these principles, the architecture is organized into three complementary layers, namely Edge, Fog, and Cloud, which together provide the foundation for a distributed intelligent system capable of combining local responsiveness, coordinated decision-making, and operational resilience within a unified Industry 4.0 framework.

B. Multi-Level Edge-Fog-Cloud Architecture

The architecture proposed in this work adopts a multi-level Edge-Fog-Cloud organization designed to distribute intelligence according to the temporal and operational requirements of industrial decision-making. As illustrated in Fig. 1, the architecture is structured around three complementary layers that collectively support perception, analysis, coordination, and optimization functions across the industrial system.

This organization is motivated by the observation that industrial decisions differ significantly in terms of urgency, scope, and computational complexity. Certain events require immediate reactions at the equipment level, while others depend on a broader operational context or long-term analysis. A centralized architecture is often unable to efficiently address these heterogeneous requirements because all information must be transmitted to a common processing point before decisions

can be made. In contrast, the proposed architecture distributes responsibilities across multiple decision horizons while maintaining global system coherence [3], [5].

The Edge layer represents the operational level of the architecture. It is directly connected to physical assets and industrial equipment through sensors and actuators. At this level, intelligent entities continuously monitor operational parameters such as vibration, current consumption, pressure, flow rate, conveyor speed, and product quality indicators. The primary objective of the Edge layer is to ensure real-time responsiveness by processing information locally and executing immediate corrective actions whenever necessary. This capability is particularly important in industrial environments where delays of only a few seconds may lead to equipment damage, production losses, or safety risks [3], [19].

The Fog layer acts as an intermediate coordination level between local operations and global supervision. Rather than focusing on individual equipment, Fog nodes maintain a contextual view of an entire production area. They aggregate information originating from multiple Edge entities, correlate events, execute predictive analytics functions, and coordinate operational decisions within their respective zones. In the proposed deployment scenario, three Fog nodes are mapped to the three major phases of the OCP Benguerir production line: destoning, screening, and loading. This organization allows operational decisions to benefit from a broader context while preserving the responsiveness required by industrial processes [4].

The Cloud layer provides strategic and system-wide intelligence. It is responsible for global supervision, long-term optimization, model management, and knowledge extraction from historical operational data. Unlike Edge and Fog layers, which primarily focus on immediate and medium-term decisions, the Cloud layer operates over longer time horizons and supports continuous improvement of the industrial system. It also coordinates distributed learning processes and maintains a global representation of production activities, equipment conditions, and operational performance [1].

A key characteristic of the proposed architecture is the existence of bidirectional information flows between the three layers. Operational data generated at the Edge level are progressively aggregated and contextualized as they move toward the Fog and Cloud layers. Conversely, optimization policies, model updates, and strategic decisions generated at higher levels are propagated downward and adapted to local operational conditions. This continuous exchange of information enables the architecture to combine local autonomy with global coordination.

Unlike many existing Edge-Fog-Cloud solutions that focus primarily on data processing infrastructure, the proposed architecture explicitly integrates decision-making, coordination, and resilience mechanisms within the architectural model itself. This approach transforms the architecture from a simple communication hierarchy into a distributed intelligent system capable of supporting complex Industry 4.0 applications while maintaining operational continuity and scalability.

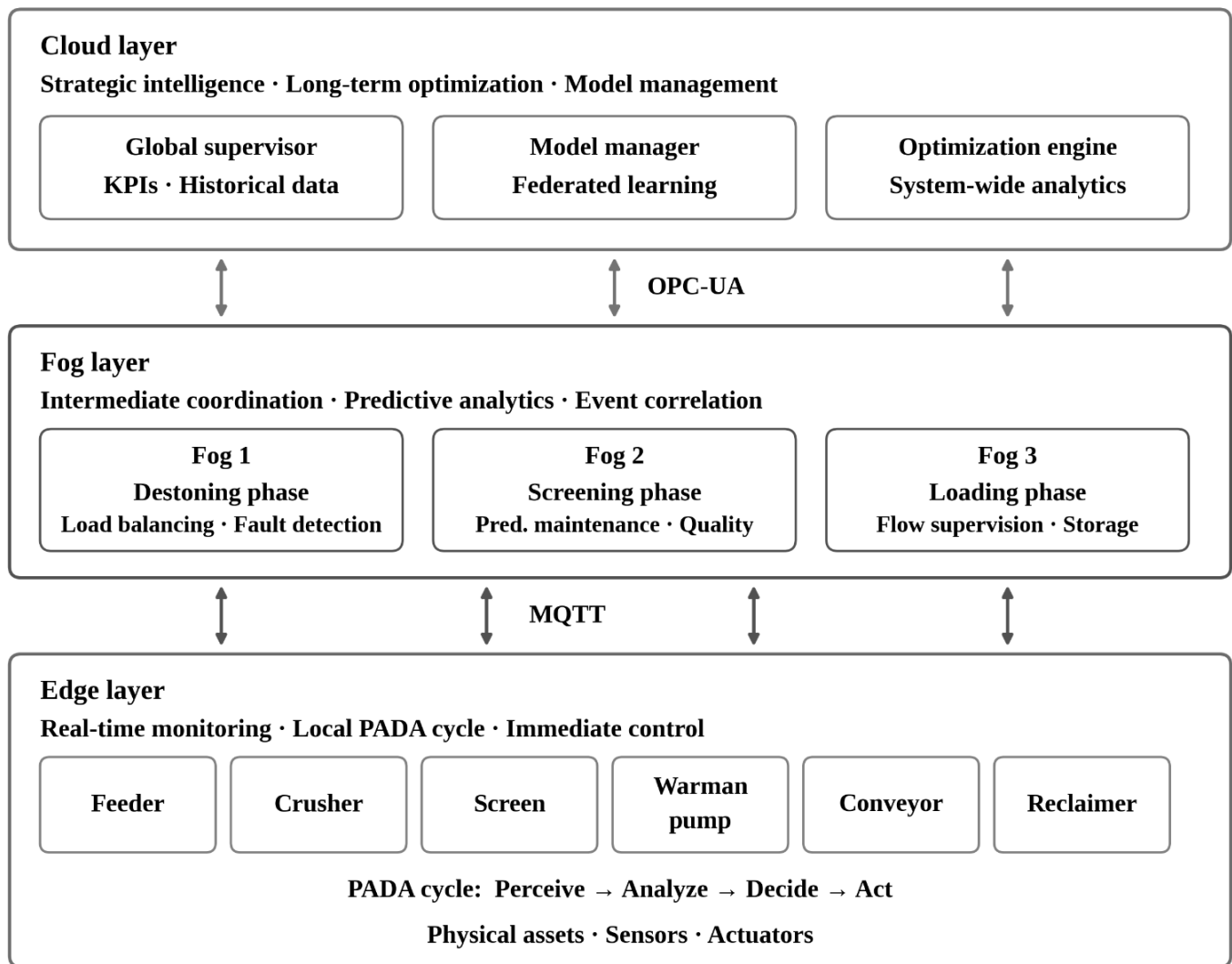


Fig. 1. Distributed edge-fog-cloud architecture.

C. Formal Modeling of Intelligent Entities

One of the limitations frequently observed in existing Industry 4.0 architectures concerns the lack of formal models capable of explicitly describing intelligent industrial entities and their role within distributed environments. Although concepts such as Cyber-Physical Systems, Industrial Internet of Things, and Multi-Agent Systems are widely adopted in the literature, the internal structure of intelligent components is often described at a conceptual level, making implementation, comparison, and reproducibility more difficult [5], [6]. This issue becomes particularly important in distributed architectures where multiple autonomous entities interact continuously across different decision layers.

To address this limitation, the proposed architecture introduces a formal model of intelligent entities that serves as the fundamental building block of the distributed intelligent system. The objective of this model is to provide a unified representation applicable to all components of the architecture, regardless of whether they operate at the Edge, Fog, or Cloud level. Such a representation facilitates system design,

interoperability, scalability, and future extensions while maintaining a consistent architectural framework.

In the context of this work, an intelligent entity is defined as an autonomous component capable of perceiving its environment, processing information, making decisions, executing actions, and interacting with other entities through standardized communication mechanisms. This definition encompasses both physical and software components. For example, an Edge node attached to a feeder, a predictive maintenance service deployed at the Fog layer, and a federated learning module operating in the Cloud can all be represented using the same formal model.

Formally, each intelligent entity is represented by the following seven-component tuple:

$$E = (Id, P, S, D, A, C, I) \quad (1)$$

where, (Id) denotes the identity of the entity, (P) represents perception mechanisms, (S) defines the internal state, (D) corresponds to decision-making functions, (A) represents executable actions, (C) denotes communication capabilities, and

(I) describes integration constraints. As illustrated in Fig. 2, the proposed model organizes the behavior of each intelligent entity around seven complementary dimensions that collectively

capture its operational, cognitive, and interactional characteristics. The model provides a common abstraction layer that can be applied consistently across the entire architecture.

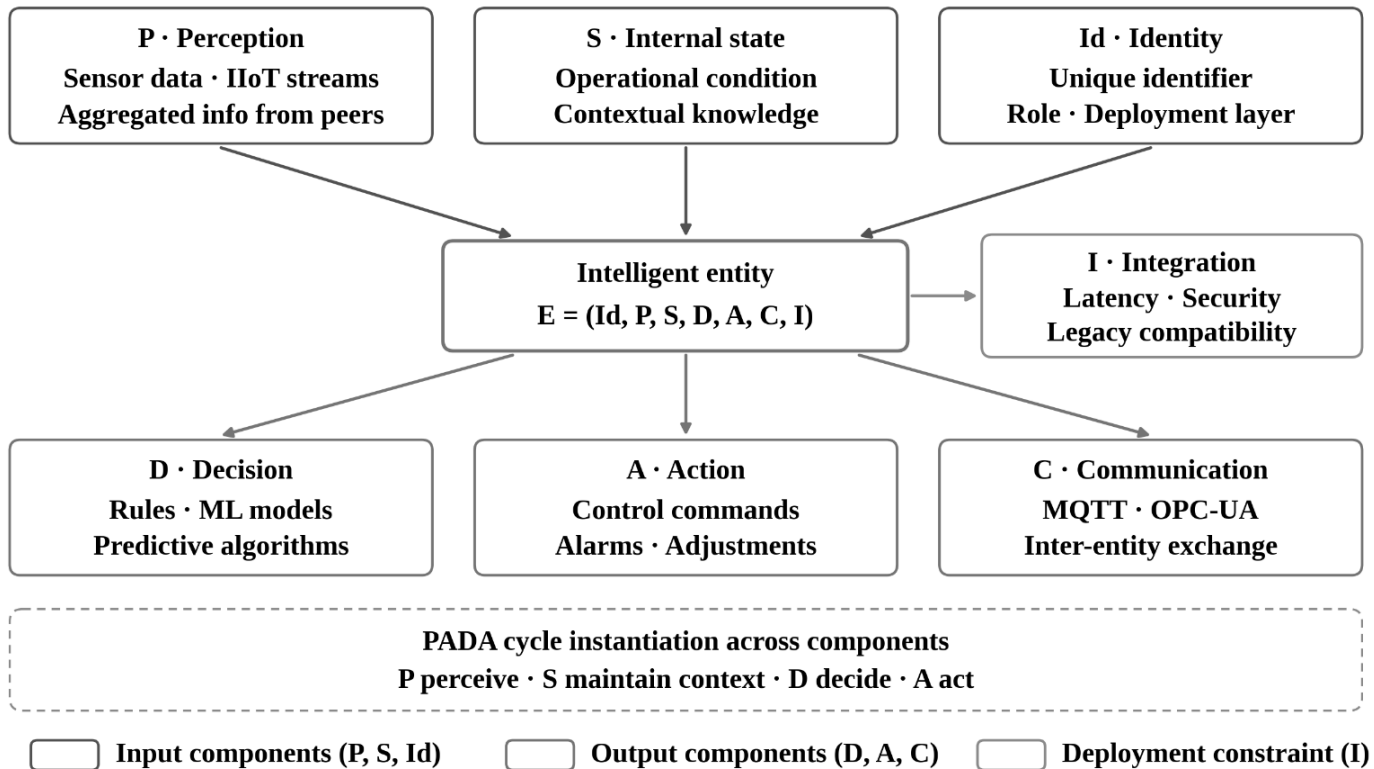


Fig. 2. Formal intelligent entity model.

The identity component ((Id)) uniquely identifies each entity within the architecture. Beyond simple identification, this component enables traceability, resource management, and coordination among distributed entities. It also facilitates the organization of entities according to their functional role and deployment layer.

The perception component ((P)) represents the mechanisms through which an entity acquires information from its environment. Depending on the architectural level, perception may involve direct acquisition of physical measurements from sensors, reception of processed data from neighboring entities, or access to aggregated information originating from higher layers. Perception constitutes the entry point of the intelligent decision cycle and provides the information required for subsequent analysis [19], [20].

The state component ((S)) captures the current operational condition of the entity. It includes both instantaneous observations and contextual information derived from previous interactions. Maintaining an internal state allows entities to preserve operational context and supports adaptive behavior in dynamic industrial environments.

The decision component ((D)) contains the mechanisms responsible for transforming information into decisions. These mechanisms may rely on predefined rules, machine learning models, predictive algorithms, or hybrid approaches depending on the complexity of the task and the computational resources

available at each architectural layer. The explicit representation of decision capabilities is particularly important in distributed intelligent systems because it clarifies where and how decisions are generated [10].

The action component ((A)) defines the set of operations that an entity can perform once a decision has been produced. Actions may include local control commands, alarm generation, parameter adjustments, maintenance recommendations, or communication with other entities. Through this component, decisions are translated into observable effects on the industrial environment.

The communication component ((C)) represents the interfaces and protocols that enable information exchange among distributed entities. Communication plays a central role in maintaining coordination across Edge, Fog, and Cloud layers. It allows entities to share operational information, synchronize their activities, and contribute to collective decision-making processes [3], [4].

Finally, the integration component ((I)) captures the constraints that govern the deployment and operation of the entity within an industrial environment. These constraints may include latency requirements, computational limitations, security policies, interoperability standards, and compatibility with legacy industrial systems. By explicitly incorporating integration aspects into the model, the architecture addresses one of the major challenges of Industry 4.0 adoption, namely the

coexistence of modern intelligent services with existing industrial infrastructures [5], [19].

More precisely, *Id* is drawn from the set of unique entity identifiers; *P* maps observable inputs to perceived information; *S* belongs to the entity state space and contains current and retained context; *D* maps perceived information and state to an admissible decision; *A* maps a selected decision to an executable action; *C* defines the permitted messages, interfaces, and communication protocols; and *I* contains integration constraints, including latency, resource, interoperability, security, and safety requirements.

At the architectural level, entity behavior is represented as a transition system whose principal states correspond to Perceive, Analyze, Decide, Act, Degraded, and Recovery. A valid perception updates the entity state and enables analysis; an analysis result enables a decision when its confidence and validity conditions are satisfied; an authorized decision enables an admissible action; and completion or rejection of the action returns the entity to perception. Communication loss or node failure triggers a transition to Degraded, whereas restored connectivity triggers Recovery and controlled resynchronization before normal operation resumes.

The representation is governed by four architectural invariants: each entity has a unique identity; every executed action is traceable to an originating state and decision; an action must satisfy the constraints contained in *I* before execution; and loss of higher-level connectivity must not disable the locally designated critical functions. These invariants define requirements for a future executable specification. They are not presented here as mechanically verified properties.

The proposed model provides a consistent structural representation for intelligent entities participating in the architecture. By bringing perception, state management, decision-making, action execution, communication, and integration constraints into a single framework, it establishes a basis for distributed coordination across operational levels. The tuple does not constitute a complete formal verification framework: detailed data types, protocol semantics, executable transition guards, temporal properties, and proof obligations remain to be specified in implementation-oriented formalism.

D. Distributed Cognition Model

The effectiveness of a distributed intelligent architecture depends not only on the way computational resources are organized but also on how cognitive functions are distributed throughout the system. While many Industry 4.0 architectures incorporate advanced analytics, machine learning algorithms, or autonomous agents, the cognitive processes underlying decision-making often remain implicit. As a result, it becomes difficult to understand how information is transformed into actions and how decisions generated at different architectural levels remain coherent [5], [6], [21].

To address this challenge, the proposed architecture adopts a distributed cognition model based on the PADA cycle, which consists of four successive phases: Perceive, Analyze, Decide, and Act. The model provides a unified decision-making framework that can be instantiated across all intelligent entities regardless of their role or deployment layer. Rather than

concentrating cognitive functions within a centralized controller, cognition is distributed among Edge, Fog, and Cloud entities according to their operational responsibilities and decision horizons.

The PADA model is directly connected to the formal intelligent entity model introduced in Section II-C. The perception component ((P)) supports information acquisition, the state component ((S)) maintains contextual knowledge, the decision component ((D)) generates decisions, and the action component ((A)) executes the selected responses. Through this integration, intelligent entities become autonomous cognitive units capable of interacting with their environment while remaining coordinated within the overall architecture.

The first phase of the cycle, Perceive, is responsible for acquiring information from the environment. Depending on the architectural level, perception may involve physical measurements collected from sensors, aggregated operational information received from neighboring entities, or global performance indicators obtained from higher-level services. The quality and relevance of the perceived information directly influence the effectiveness of subsequent cognitive processes.

The second phase, Analyze, transforms raw observations into meaningful knowledge. This phase may involve data filtering, event correlation, anomaly detection, predictive analytics, or machine learning inference. The objective is not merely to process information but to contextualize it and extract actionable insights. Several studies have emphasized the importance of contextual analysis in distributed industrial systems, particularly when multiple heterogeneous data sources must be combined to support decision-making [10], [12].

The third phase, Decide, generates an appropriate response according to the current state of the entity and the operational objectives of the system. Decisions may be derived from expert rules, optimization criteria, predictive models, or hybrid approaches. Because decision-making responsibilities are distributed across multiple architectural levels, decisions are generated locally whenever possible and escalated only when a broader operational context is required.

The final phase, Act, translates decisions into operational actions. Depending on the layer involved, actions may correspond to equipment control commands, alarm generation, maintenance recommendations, policy dissemination, or model updates. Through this phase, cognitive processes produce measurable effects on the industrial environment and contribute directly to system performance.

Fig. 3 illustrates how the PADA cycle is instantiated across the three architectural layers of the proposed framework.

PADA is conceptually related to established feedback and cognitive cycles. MAPE-K separates monitoring, analysis, planning, and execution around an explicit knowledge base [21]; OODA emphasizes rapid observation, orientation, decision, and action; and Sense-Plan-Act links perception to planning and control. PADA does not claim to replace these models or to introduce an entirely new feedback-loop principle. Its contribution in this work is architectural: the same Perceive-Analyze-Decide-Act structure is instantiated at Edge, Fog, and Cloud levels and coupled with decision subsidiarity, escalation,

and graceful degradation. Compared with MAPE-K, contextual knowledge is represented within the entity state S rather than as a separate shared component, while the Decide phase may

include rule selection, planning, or optimization according to the layer concerned.

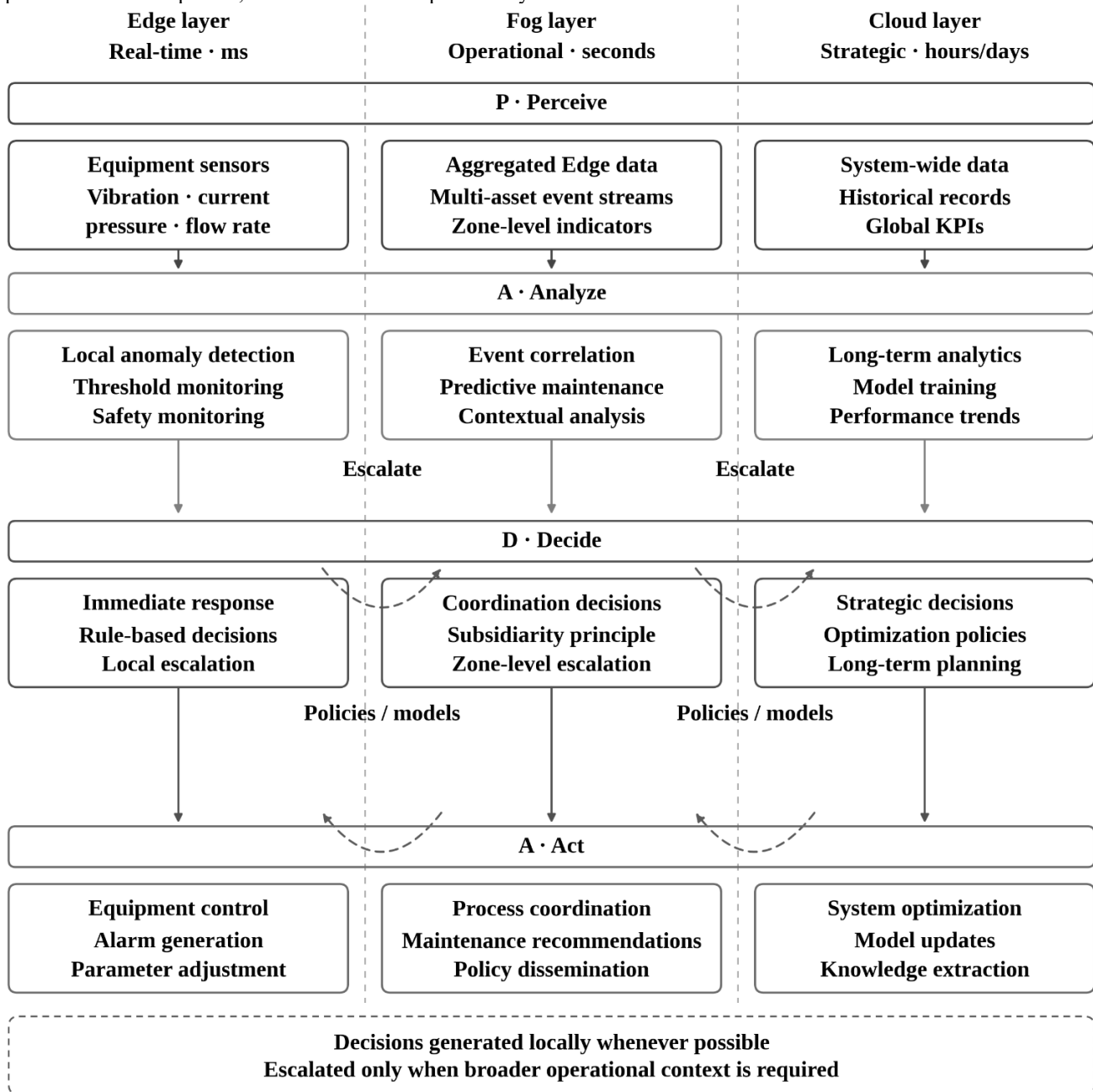


Fig. 3. PADA cycle distribution across the three-layer architecture.

The transition from one PADA phase to the next is conditional rather than purely sequential. Perception proceeds to analysis only when input validity requirements are met; analysis proceeds to decision when sufficient context or confidence is available; decision proceeds to action only when authorization, safety, and integration constraints are satisfied. Otherwise, the entity may retain the event, request additional context, escalate it to a higher layer, or enter a safe degraded state. These rules provide interaction semantics at the architectural level, while

application-specific guards and thresholds remain implementation parameters.

A distinctive characteristic of the proposed approach is the distribution of cognition across multiple temporal horizons. At the Edge layer, the PADA cycle operates under strict real-time constraints and focuses on immediate operational events. Typical examples include anomaly detection, local safety monitoring, and equipment supervision.

At the Fog layer, cognition supports contextual analysis, predictive maintenance, event correlation, and operational coordination. Finally, at the Cloud layer, cognitive processes are oriented toward strategic optimization, knowledge extraction, model management, and long-term planning.

This hierarchical organization enables the architecture to combine local responsiveness with global consistency. Immediate decisions can be taken close to the physical process while strategic decisions benefit from a broader operational perspective. Consequently, the proposed architecture avoids the limitations of purely centralized systems while preserving coordination among distributed intelligent entities.

By introducing a common cognitive framework across all decision levels, the PADA model provides a coherent foundation for distributed intelligence, facilitating scalability, adaptability, and resilience in complex Industry 4.0 environments.

E. Coordination and Communication Mechanisms

The effectiveness of a distributed intelligent architecture depends not only on the capabilities of individual entities but also on their ability to exchange information and coordinate decisions. In Industry 4.0 environments, where cyber-physical systems, intelligent services, and connected devices operate simultaneously, maintaining consistency between local autonomy and global objectives becomes a major challenge [5], [6], [10].

To address this requirement, the proposed architecture adopts a hierarchical communication framework spanning the Edge, Fog, and Cloud layers. Rather than relying on centralized control, coordination is achieved through structured information exchanges and distributed decision-making processes.

Communication between Edge entities and Fog nodes is implemented through the MQTT protocol. Its lightweight publish-subscribe model makes it particularly suitable for Industrial Internet of Things environments characterized by resource-constrained devices and continuous data exchanges [10]. Edge entities publish operational data, alarms, and equipment states, while Fog nodes subscribe to the information relevant to their production area. This mechanism improves scalability and reduces communication overhead.

The Fog layer acts as the primary coordination level of the architecture. By aggregating information originating from multiple Edge entities, Fog nodes perform event correlation, predictive analytics, and operational supervision. This intermediate layer provides a broader operational context while preserving the responsiveness required for industrial processes [12], [13].

Communication between Fog and Cloud layers relies on OPC Unified Architecture (OPC-UA), a widely adopted standard for industrial interoperability and information exchange [22], [23]. Through OPC-UA, supervisory information, analytical models, and optimization policies can be exchanged in a structured and secure manner.

Coordination is further reinforced through decision subsidiarity. Local events are handled by Edge entities when the required information and authority are available, whereas

decisions requiring a broader context or additional resources are escalated to Fog or Cloud levels [7], [24]. Each decision message is expected to carry its origin, scope, priority, timestamp, validity interval, and confidence. Safety-critical local actions take precedence over optimization requests; conflicting non-critical decisions are suspended and escalated to the responsible Fog node; and unresolved cross-zone conflicts are escalated to the Cloud supervisory level. These rules define a deterministic coordination policy rather than a distributed consensus protocol. To provide an implementation-oriented specification of this policy, Algorithm 1 describes local decision execution, priority-based conflict resolution, escalation, temporary Fog-level function takeover, and state reconciliation after connectivity restoration. The design of a distributed consensus protocol among equivalent Fog nodes remains outside the scope of this study.

Algorithm 1 specifies how each Edge entity processes observations and determines whether a decision can be executed locally or must be escalated. Local execution is permitted only when the decision satisfies the entity's authorization and safety constraints. When communication is unavailable, the entity activates its authorized degraded mode and buffers decisions and state updates. The procedure also defines deterministic priority rules for conflict resolution, temporary Fog-level takeover following an Edge-node failure, and state reconciliation after connectivity restoration. Although this specification improves the reproducibility of the coordination mechanism, its correctness and runtime performance remain to be evaluated through implementation and controlled experiments.

Algorithm 1: Distributed PADA Decision, Escalation, and Recovery Procedure

Initialize entity states S , decision rules R , authorization policies AP , priority levels PL , communication status CS , local state version V_E , and Fog state version V_F

Compute the initial connectivity and operational status of each Edge and Fog entity.

While the industrial system is operational, do

For every Edge entity E_i do

 Acquire and validate observation o_i .

Update internal state S_i using the validated observation.

 Execute the Perceive and Analyze phases of PADA.

Compute candidate decision d_i using R and S_i

 Assign a priority according to safety, urgency, deadline, and operational impact.

If d_i is locally authorized and satisfies the safety constraints, then

 Execute d_i at the Edge level.

 Record the action, timestamp, priority, and state version.

Else if the Edge-Fog connection is available, then

Escalate d_i and its metadata to the associated Fog node.

If the Fog node detects conflicting decisions, then

 Select the safety-critical decision.

 Otherwise, select the decision with the highest priority.

 When priorities are equal, select the earliest valid decision.

 Escalate unresolved cross-zone conflicts to the Cloud

End if

Else

 Activate the authorized degraded operating mode.

 Buffer decisions and state updates with timestamps and sequence numbers.

End if

If E_i becomes unavailable, then

 Reassign its critical monitoring functions to the associated Fog node.

 Suspend functions that cannot be executed safely by the Fog node.

End if

If communication is restored, then

 Transmit buffered records in sequence-number order.

 Remove duplicate or invalid records.

 Compare V_E with V_F

If a state conflict is detected, then

 Preserve previously executed safety-critical actions.

 Apply the most recent authorized consistent state.

 Record the reconciliation result in the audit log.

End if

 Resume normal PADA operation.

End if

End for

End while

A production implementation must complement this coordination policy with mutual authentication, role- or attribute-based authorization, encryption in transit, key management, audit logging, trust assessment, compromised-node detection, privacy controls, and safety interlocks. Fog-based security mechanisms can support distributed threat detection in large-scale IIoT environments [25]. These controls are architectural requirements associated with the integration component I; their implementation and security evaluation are outside the scope of the present study.

F. Resilience and Graceful Degradation

Resilience is a fundamental requirement for distributed industrial architectures, where communication failures, node outages, and infrastructure disruptions can compromise operational continuity [26]. Rather than assuming permanent availability of all system components, the proposed architecture adopts a graceful degradation strategy that preserves essential functions under partial failures. This approach is grounded in the distributed nature of the PADA cycle: since intelligent entities operate autonomously, they retain local perception, analysis, decision, and action capabilities even when connectivity with higher layers is interrupted [8], [26], [27].

Three degradation levels are considered. At the first level, Edge entities lose connectivity with their Fog node. In this case, they switch to autonomous mode, buffer operational data locally, and maintain critical functions such as anomaly

detection and safety monitoring until communication is restored. At the second level, Fog–Cloud connectivity is lost. Fog nodes continue operating using locally available information and previously received models, preserving event correlation, predictive maintenance, and coordination functions. Strategic optimization is temporarily suspended, but operational continuity is maintained [6], [8]. At the third level, a Fog node fails entirely. Neighboring Fog nodes dynamically assume coordination responsibilities for the affected zone, preventing isolated production areas [8], [26].

Once normal conditions are restored, the architecture initiates a progressive resynchronization process, redistributing buffered data, updated models, and optimization policies across all layers. Fig. 4 illustrates these three degradation levels and their corresponding recovery mechanisms.

The considered failure model covers communication-link loss and crash-stop unavailability of an Edge, Fog, or Cloud component; Byzantine behavior and compromised nodes are not modeled. Failures would be detected through configurable heartbeat timeouts, missed-message counts, and local health checks, with threshold values determined during implementation. Buffered records should carry timestamps, sequence numbers, and model or policy versions. Recovery should use idempotent replay, duplicate suppression, and version checks; when concurrent updates conflict, the responsible Fog node should retain safety-critical local actions and reconcile non-critical state according to the most recent valid policy. These mechanisms are specified as design requirements and have not yet been experimentally evaluated for consistency, recovery time, or availability.

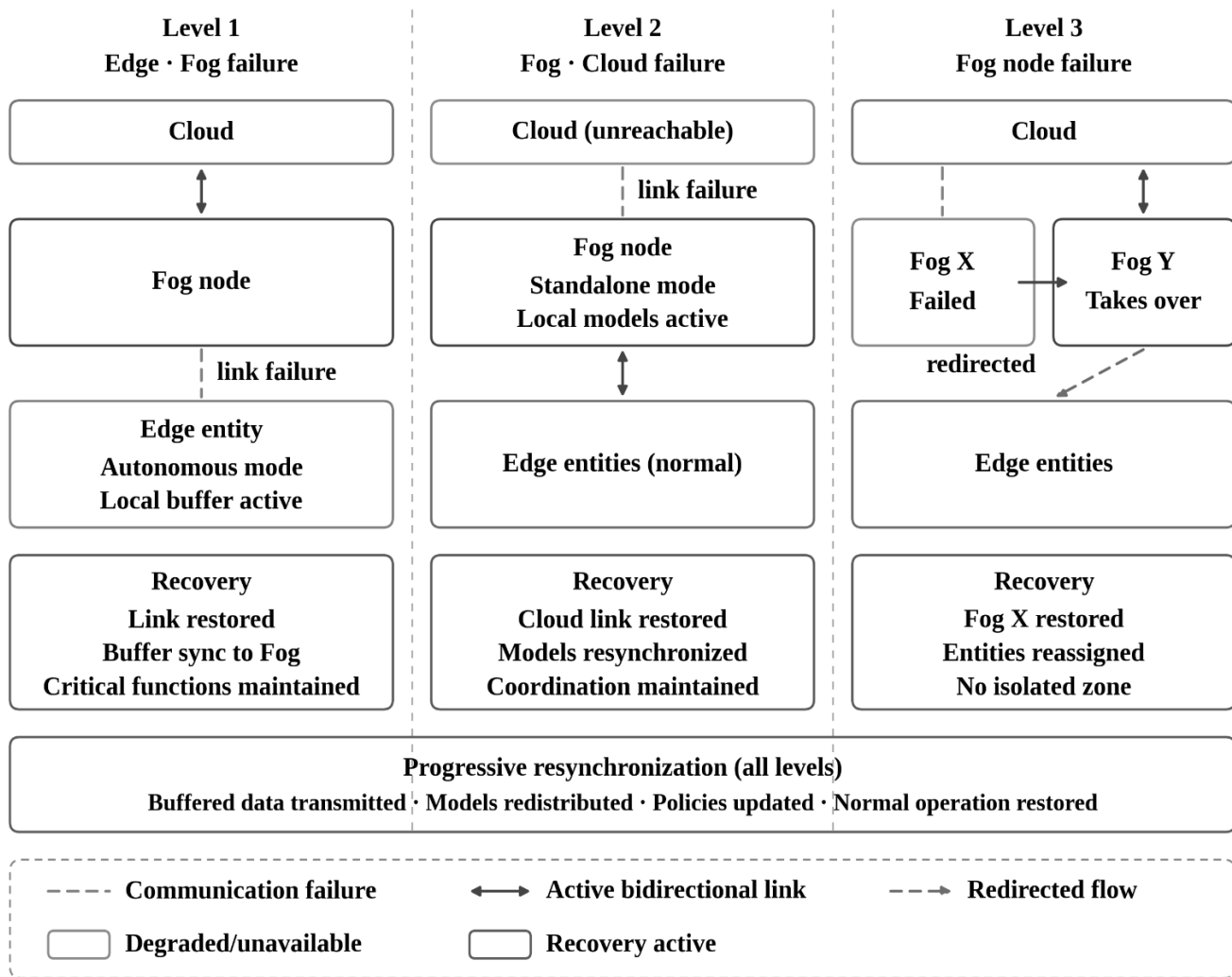


Fig. 4. Three-level graceful degradation and recovery mechanisms.

III. INDUSTRIAL CASE STUDY AND PROPOSED DEPLOYMENT SCENARIO

The fixed phosphate-processing facilities of the OCP Benguerir mining site are adopted as an industrial case study for defining the proposed deployment scenario. The architecture has not been physically deployed at the site, and this study does not use operational logs, field measurements, or observed system behavior as validation evidence. The production line is represented through three interconnected phases: destoning, screening, and loading. Their equipment, data sources, decision horizons, and continuity requirements provide an industrially grounded context for the architectural mapping [2], [5], [23], [28].

In the proposed mapping, Fog 1 would supervise feeders, crushers, and conveyors in the destoning phase, focusing on load balancing and fault detection. Fog 2 would coordinate screens, hydrocyclones, and Warman pumps in the screening phase, supporting predictive maintenance and quality monitoring [29]. Fog 3 would manage material-flow supervision and storage coordination during loading. This three-node organization follows the operational boundaries represented in the case study and is not a record of installed Fog infrastructure [8], [30].

In this scenario, intelligent Edge entities would be associated with the main assets of each phase and would execute local PADA cycles for monitoring, anomaly detection, and authorized immediate control. At the Cloud level, historical data and performance indicators would be consolidated for global supervision and long-term optimization. Table I summarizes this proposed mapping between production phases, equipment, and architectural components [5], [8], [31].

An illustrative data flow begins when a vibration sensor associated with a Warman pump produces a new observation. The Edge entity perceives and validates the measurement, updates its internal state, analyzes it against the configured anomaly rule or model, and either records a normal condition or issues an authorized local alarm. Events requiring wider context are published to Fog 2, which correlates them with other screening assets and may generate a maintenance recommendation. Relevant summaries can subsequently be transmitted to the Cloud for historical analysis and model management. This example explains the intended information flow; it is not an operational trace collected from the OCP site.

TABLE I. PROPOSED DEPLOYMENT OF INTELLIGENT ENTITIES FOR THE OCP BENGUERIR CASE STUDY

Production Phase	Main Equipment	Intelligent Edge Entities	Fog Node	Primary Functions
Destoning	Feeders, Crushers, Conveyors	Equipment-level entities	Fog 1	Monitoring, anomaly detection, local control
Screening	Screens, Hydrocyclones, Warman Pumps	Equipment-level entities	Fog 2	Predictive maintenance, coordination, quality monitoring
Loading	Conveyors, Reclaimers, Storage Silos	Equipment-level entities	Fog 3	Material flow supervision, operational coordination

IV. COMPARATIVE ANALYSIS

To position the proposed architecture within the current state of the art, this section compares it with three representative families of Industry 4.0 architectures identified in the literature, namely Cyber-Physical System (CPS)-based architectures, Multi-Agent System (MAS)-based architectures, and conventional Edge-Fog-Cloud architectures. The comparison is based on six descriptive dimensions: multi-level intelligence, entity representation, explicit cognition cycle, decision subsidiarity, graceful degradation, and illustration or validation basis. The comparison results are summarized in Table II.

CPS-based architectures [5], [23], [24] provide robust integration between physical assets and computational services, with a strong emphasis on connectivity, interoperability, and data management. However, they do not explicitly define a formal model for intelligent entities, nor do they distribute cognition across multiple architectural levels. MAS-based approaches [10], [11] naturally support distributed decision-making through autonomous agents but lack a unified representation applicable to heterogeneous industrial components deployed across Edge, Fog, and Cloud layers. Furthermore, large-scale coordination and integration with IIoT infrastructures remain challenging. Conventional Edge-Fog-Cloud architectures [3], [4], [13], [32] efficiently distribute computation to improve latency and scalability, but intelligence, decision-making, and resilience are generally implemented at the application level rather than being embedded into the architecture itself.

The proposed architecture combines a structural entity representation, a distributed PADA cycle, decision subsidiarity, and a conceptually specified degradation strategy. Table II indicates whether these capabilities are explicit, partial, or implementation-dependent within each architectural family. This qualitative comparison does not establish architectural completeness or performance superiority. Individual implementations may provide capabilities that are not captured by the selected dimensions.

TABLE II. COMPARISON OF REPRESENTATIVE INDUSTRY 4.0 ARCHITECTURES

Feature	CPS	MAS	Conventional EFC	Proposed framework
Multi-level intelligence	Partial	Supported	Partial	Explicit
Entity representation	Varies	Agent-based	Usually implicit	Seven-component model
Explicit cognition cycle	Usually implicit	Partial	No	PADA
Decision subsidiarity	Partial	Supported	Partial	Explicit
Graceful degradation	Implementation-specific	Implementation-specific	Partial	Conceptual
Illustration or validation basis	Varies	Varies	Varies	OCP conceptual case

V. DISCUSSION

The proposed architecture integrates intelligent entity representation, distributed cognition, hierarchical coordination, and graceful degradation within a unified Edge, Fog, and Cloud framework. The seven-component model clarifies the responsibilities of each entity, while PADA connects perception and internal state to decisions and actions. However, this architectural integration does not demonstrate better latency, availability, or scalability. These properties depend on the implementation, workload, network conditions, and available resources.

Distributing PADA across the three levels presents several trade-offs. Edge entities can process data close to industrial equipment and maintain selected functions during communication failures. This improves local autonomy but increases processing, storage, update, and safety requirements. At the Fog level, data aggregation provides a broader operational context. However, coordination and state synchronization may increase communication overhead as the number of entities grows. Recent studies confirm that task allocation and resource management influence the performance of collaborative Edge, Fog, and Cloud architectures [14], [15], [18].

The combined use of MQTT and OPC UA also introduces integration constraints. It may require protocol adaptation, shared information models, quality of service configuration, and duplicate message management. A physical implementation reported in [16] confirms the feasibility of combining local acquisition, preprocessing, and cloud transmission. Nevertheless, its experimental context differs from the industrial scenario considered in this work.

Decision subsidiarity creates a balance between local responsiveness and global consistency. An Edge entity may act before receiving a higher-level decision. Subsequent synchronization may then reveal conflicting states or policies. The proposed priorities, escalation rules, buffering, versioning, and resynchronization mechanisms provide an architectural basis for managing this issue.

Algorithm 1 specifies deterministic rules for conflict resolution, escalation, temporary Fog-level function takeover, and post-reconnection synchronization. Their implementation as executable software and the development of a distributed consensus protocol among equivalent Fog nodes remain future work.

Graceful degradation allows selected functions to continue during communication or node failures. This principle is consistent with existing work on fault-tolerant degradation and resilient distributed architectures [17], [27]. Its effectiveness must nevertheless be evaluated through failure injection and metrics such as availability, recovery time, message loss, and data consistency.

Security is also essential in distributed IIoT environments. Authentication, authorization, encryption, trust management, audit logging, and compromised node detection should be included in future implementation. Fog-based mechanisms can support distributed threat detection [25], but they also introduce processing and communication overhead.

The comparison in Table II is qualitative and does not demonstrate the superiority of the proposed architecture. Similarly, the OCP Benguerir case study represents a conceptual mapping rather than a physical deployment. No operational data, logs, or field measurements from OCP are used as validation evidence.

A. Limitations and Future Validation

This study is limited to architectural design, structural formalization, and qualitative analysis. Although the seven-component tuple defines the principal responsibilities of an intelligent entity, it does not yet constitute a complete executable or mechanically verified formal model. Detailed data types, temporal constraints, transition guards, proof obligations, and protocol semantics remain to be defined using an implementation-oriented formalism.

Algorithm 1 provides an implementation-oriented pseudocode specification of the PADA coordination and recovery mechanisms. However, it has not yet been implemented as executable software or evaluated under runtime conditions. The OCP Benguerir case represents a proposed architectural mapping without operational data, logs, or field measurements. Authentication, authorization, encryption, trust management, compromised-node detection, privacy protection, and industrial safety controls are identified as requirements but have not been implemented or evaluated. Consequently, the present results should not be interpreted as evidence of operational performance, availability, resilience, or scalability.

Validation should proceed in successive stages. The first stage should provide an executable specification of the entity transitions and architectural invariants. The second stage should implement a small-scale prototype or simulation of the distributed PADA cycle. The third stage should introduce controlled Edge-Fog, Fog-Cloud, and Fog-node failures. The evaluation should measure end-to-end latency, decision time, availability, recovery time, communication overhead, CPU and memory utilization, message loss, data consistency, and scalability under normal and injected-failure conditions.

A subsequent industrial pilot would then be required to compare experimental results with operational measurements and assess interoperability, safety, cybersecurity, and long-term reliability under field conditions.

VI. CONCLUSION

This study proposed a distributed intelligent Edge-Fog-Cloud architecture for Industry 4.0 systems. It integrates a structural seven-component entity representation, a distributed PADA cognition cycle, decision subsidiarity, and a three-level graceful degradation strategy. The contribution is architectural and qualitative. It organizes these mechanisms within a common representation across the Edge, Fog, and Cloud levels.

The seven-component tuple makes entity responsibilities and constraints explicit, while PADA links perceived information and internal states to decisions and actions at different temporal and operational horizons. The proposed transition rules, invariants, escalation metadata, and recovery requirements strengthen the specification. However, they do not constitute a complete executable or mechanically verified model.

The qualitative comparison positions the framework relative to broad CPS, MAS, and conventional Edge-Fog-Cloud architecture families without claiming universal superiority. Its main value lies in making entity representation, distributed cognition, decision subsidiarity, and graceful degradation explicit within a unified design. The associated resource requirements, coordination overhead, protocol integration, state reconciliation, and security costs remain to be quantified.

The OCP Benguerir case study illustrates a proposed mapping of intelligent entities to a realistic phosphate-processing context. It does not represent a physical deployment or provide operational evidence of performance. The applicability of the framework to other industrial domains must therefore be examined through implementation and comparative evaluation.

Future work will develop an executable prototype or simulation environment for the proposed architecture. The evaluation will implement conflict resolution, task reassignment, Fog-node takeover, post-reconnection synchronization, and security controls. It will measure latency, availability, recovery time, communication overhead, resource utilization, data consistency, and scalability under normal and injected-failure conditions. A subsequent industrial pilot will be required to assess interoperability, safety, and operational effectiveness under field conditions.

DECLARATIONS

The authors declare no competing financial interests or personal relationships that could have influenced this work.

REFERENCES

- [1] L. Da Xu, E. L. Xu, and L. Li, "Industry 4.0: State of the art and future trends," *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 2941–2962, 2018, doi: 10.1080/00207543.2018.1444806.
- [2] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *J. Ind. Inf. Integr.*, vol. 6, pp. 1–10, Jun. 2017, doi: 10.1016/J.JII.2017.04.005.

- [3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.
- [4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*, Association for Computing Machinery, Aug. 2012, pp. 13–16. doi: 10.1145/2342509.2342513.
- [5] L. Monostori et al., "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, no. 2, pp. 621–641, Jan. 2016, doi: 10.1016/J.CIRP.2016.06.005.
- [6] D. G. S. Pivoto, L. F. F. de Almeida, R. da Rosa Righi, J. J. P. C. Rodrigues, A. B. Lugli, and A. M. Alberti, "Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review," *J. Manuf. Syst.*, vol. 58, pp. 176–192, Jan. 2021, doi: 10.1016/J.JMSY.2020.11.017.
- [7] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, "Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison," *Engineering*, vol. 5, no. 4, pp. 653–661, Aug. 2019, doi: 10.1016/j.eng.2019.01.014.
- [8] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," *Journal of Intelligent Manufacturing*, vol. 31, pp. 127–182, 2020, doi: 10.1007/s10845-018-1433-8.
- [9] T. Borangiu, D. Trentesaux, A. Thomas, P. Leitão, and J. Barata, "Digital transformation of manufacturing through cloud services and resource virtualization," *Comput. Ind.*, vol. 108, pp. 150–162, Jun. 2019, doi: 10.1016/J.COMPIND.2019.01.006.
- [10] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 979–991, Oct. 2009, doi: 10.1016/j.engappai.2008.09.005.
- [11] P. Leitão, S. Kamouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart Agents in Industrial Cyber-Physical Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016, doi: 10.1109/JPROC.2016.2521931.
- [12] A. Yousefpour et al., "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.
- [13] F. H. Khoso, A. Lakhan, A. A. Arain, M. A. Soomro, S. Z. Nizamani, and K. Kanwar, "A microservice-based system for Industrial Internet of Things in fog-cloud assisted network," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7029–7032, 2021, doi: 10.48084/etasr.4077.
- [14] V. Barbosa et al., "Performance Evaluation of IoT-Based Industrial Automation Using Edge, Fog, and Cloud Architectures," *Journal of Network and Systems Management*, vol. 33, Art. no. 15, Jan. 2025, doi: 10.1007/s10922-024-09893-x.
- [15] Z. Li et al., "A reference framework for the digital twin smart factory based on cloud-fog-edge computing collaboration," *J. Intell. Manuf.*, vol. 36, no. 5, pp. 3625–3645, Jun. 2025, doi: 10.1007/s10845-024-02424-0.
- [16] R. Dogea, X. T. Yan, and R. Millar, "Implementation of an edge-fog-cloud computing IoT architecture in aircraft components," *MRS Commun.*, vol. 13, no. 3, pp. 416–424, Jun. 2023, doi: 10.1557/s43579-023-00364-z.
- [17] M. Naiouf, E. Rucci, F. Chichizola, and L. De Giusti, Eds., *Cloud Computing, Big Data & Emerging Topics, Communications in Computer and Information Science*, vol. 1828. Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-40942-4.
- [18] Z. Lin et al., "Edge-fog-cloud hybrid collaborative computing solution with an improved parallel evolutionary strategy for enhancing tasks offloading efficiency in intelligent manufacturing workshops," *J. Intell. Manuf.*, vol. 36, no. 7, pp. 4635–4662, Oct. 2025, doi: 10.1007/s10845-024-02463-7.
- [19] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014, doi: 10.1109/TII.2014.2300753.
- [20] E. A. Lee, "Cyber-physical systems - Are computing foundations adequate?" position paper presented at the NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin, TX, USA, Oct. 2006.
- [21] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003, doi: 10.1109/MC.2003.1160055.
- [22] W. Mahnke, S. H. Leitner, and M. Damm, *OPC unified architecture*. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-540-68899-0.
- [23] A. W. Colombo et al., Eds., *Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach*. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-05624-1.
- [24] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015, doi: 10.1016/j.mfglet.2014.12.001.
- [25] A. Ometov, O. L. Molua, M. Komarov, and J. Nurmi, "A survey of security in cloud, edge, and fog computing," *Sensors*, vol. 22, no. 3, Art. no. 927, 2022, doi: 10.3390/s22030927.
- [26] S. Colabianchi, F. Costantino, G. Di Gravio, F. Nonino, and R. Patriarca, "Discussing resilience in the context of cyber-physical systems," *Comput. Ind. Eng.*, vol. 160, p. 107534, Oct. 2021, doi: 10.1016/J.CIE.2021.107534.
- [27] Y. Lin, S. Kulkarni, and A. Jhumka, "Automation of fault-tolerant graceful degradation," *Distrib. Comput.*, vol. 32, no. 1, pp. 1–25, Feb. 2019, doi: 10.1007/s00446-017-0319-x.
- [28] T. Zvarivadza et al., "On the impact of Industrial Internet of Things (IIoT) - mining sector perspectives," *Int. J. Min. Reclam. Environ.*, vol. 38, no. 10, pp. 771–809, 2024, doi: 10.1080/17480930.2024.2347131.
- [29] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006, doi: 10.1016/j.ymssp.2005.09.012.
- [30] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "A critical review of smart manufacturing and Industry 4.0 maturity models: Implications for small and medium-sized enterprises," *Journal of Manufacturing Systems*, vol. 49, pp. 194–214, 2018, doi: 10.1016/j.jmsy.2018.10.005.
- [31] V. Alcácer and V. Cruz-Machado, "Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems," *Engineering Science and Technology, an International Journal*, vol. 22, no. 3, pp. 899–919, Jun. 2019, doi: 10.1016/J.JESTCH.2019.01.006.
- [32] S. Medina, D. Montezanti, L. Gómez D'Orazio, F. Garay, A. De Giusti, and M. Naiouf, "Distributed architectures based on edge computing, fog computing and end devices: A conceptual review incorporating resilience aspects," in *Cloud Computing, Big Data & Emerging Topics, Communications in Computer and Information Science*, vol. 1828. Cham, Switzerland: Springer, 2023, pp. 31–44, doi: 10.1007/978-3-031-40942-4_3.