Leveraging Asset Administration Shells and Fog Computing for Scalable and Secure Smart Pool Management

André C. Costa[1], Rui Pinto[1,2], Gil Gonçalves[1,2]
Dept. of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal[1]
SYSTEC, ARISE, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal[2]
Email: up201905916@up.pt, {rpinto, gil}@fe.up.pt

***Abstract*—Maintaining optimal water quality in swimming pools is critical for ensuring safety, hygiene, and user comfort, yet traditional methods often prove time-consuming, error-prone, and operationally inefficient. This paper presents SmartPool, an advanced Cyber-Physical System (CPS)-based solution that automates swimming pool management through real-time monitoring, automated control, and Asset Administration Shell (AAS)-based Digital Twins. Unlike existing proprietary ecosystems, SmartPool leverages AAS as a standardized framework to ensure interoperability, scalability, and lifecycle-oriented management of diverse pool assets, moving beyond closed solutions. The system is built upon a robust five-level CPS architecture, strategically incorporating a Fog computing layer for distributed intelligence, hosting the AAS environment, and facilitating efficient edge processing. SmartPool measures critical water parameters such as pH, chloride, temperature, and water levels, provides real-time data visualization through a user-friendly dashboard, and enhances safety with integrated camera and recognition algorithms. This extended work details the comprehensive methodology, the enhanced architectural design, and the prototype validation, demonstrating the feasibility of this open, AAS-enabled approach to significantly improve efficiency, sustainability, and safety in pool operations. The research lays the foundation for scalable smart water management systems and promotes seamless integration into broader Industry 4.0 and smart city initiatives.**

***Keywords*-Pool System; Pool Maintenance; Automation; Control; Asset Administration Shell**

I. Introduction

This work builds upon a preliminary conference contribution that introduced the *SmartPool* concept as an automated Cyber-Physical System (CPS) for swimming pool monitoring and safety [1]. That initial study demonstrated the feasibility of integrating sensors, actuators, and real-time monitoring into a unified system for pool management. The present article extends that work by significantly deepening the architectural design, expanding the prototype implementation, and providing a more comprehensive validation and discussion of results.

Swimming pools are an increasingly common feature in modern environments, serving recreational, fitness, hospitality, and tourism purposes. With over 10.7 million public and residential pools in the United States alone [2], the global expansion of aquatic facilities places growing pressure on pool maintenance systems to meet expectations for safety, sustainability, and operational efficiency [3]. These demands are further intensified by frequent water consumption and chemical treatments, which impact both environmental sustainability and user well-being.

Ensuring water quality and safety is paramount. Inadequate pool maintenance is directly associated with health risks, including dermatological conditions, respiratory irritants, and waterborne disease outbreaks such as cryptosporidiosis and giardiasis [4]. These challenges highlight the need for intelligent, accountable, and data-driven pool management systems that go beyond isolated sensing and manual intervention.

While existing market solutions already incorporate automated control logic—typically through sensors and chemical dosing actuators—most offerings remain proprietary, closed, and vertically integrated. Such architectures limit interoperability, hinder data exchange with external platforms, and provide limited support for asset lifecycle visibility. Moreover, the fragmentation of sensor data, chemical consumption records, and equipment diagnostics prevents holistic oversight, particularly in multi-stakeholder or public pool environments.

Recent advances in CPS, the Internet of Things (IoT), and Digital Twin (DT) technologies [5], [6] offer a promising path forward. IoT enables continuous data acquisition from distributed sensors, CPS supports tight integration between physical and digital layers, and DTs facilitate simulation, forecasting, and lifecycle-oriented management. Among DT frameworks, the Asset Administration Shell (AAS) [7] has emerged as a standardized and semantically rich representation capable of unifying heterogeneous devices into interoperable systems [8]–[10].

In this context, this paper presents an extended version of *SmartPool*, a CPS-enabled prototype platform that applies AAS-based Digital Twins to swimming pool environments. The main goals of this work are to:

- Design an open, modular architecture for intelligent pool monitoring and control, supporting integration with IoT sensors and visual safety systems.
- Apply AAS-based DTs to pool assets—such as sensors, actuators, and tanks—to provide structured, interoperable digital representations.
- Develop and validate a working prototype that demonstrates real-time monitoring, automated actuation, and object recognition-based safety mechanisms.
- Explore the feasibility of lifecycle-oriented, standards-driven asset management in non-industrial water environments.

Unlike existing proprietary solutions, *SmartPool* leverages open-source middleware (*Eclipse BaSyx* [11], [12]) and a CPS-inspired architecture to support bi-directional communication between physical components and digital services [13]. Visual surveillance and object recognition enable proactive safety mechanisms, while AAS *Submodels* expose asset states and metadata, supporting transparency, traceability, and integration with smart building and smart city ecosystems.

Developed as an academic proof-of-concept under resource constraints, the system demonstrates the practical applicability of CPS and AAS principles to domains traditionally excluded from industrial digitalization. The research contributes to ongoing discussions on intelligent water management [14] by introducing a standards-aligned, modular framework that can be extended to public facilities, hospitality venues, and cyber-physical smart environments.

The remainder of this paper is structured as follows. Section II reviews relevant literature on automated pool systems and digitalization approaches. Section III presents the adopted research methodology. Section IV details the system architecture and components. Section V describes the prototype implementation and validation. Section VI discusses the results, limitations, and implications for real-world deployment. Finally, Section VII concludes the paper and outlines future research directions.

II. Related Work

Beyond dedicated smart pool systems, the principles of intelligent monitoring and control are increasingly vital across various water control and maintenance domains, including urban water networks [15], industrial wastewater treatment [16], and agricultural irrigation [17]. These broader "smart water management systems" share common challenges in efficiency, sustainability, and data-driven decision-making, which IoT, CPS, and DTs are actively addressing [18].

In these diverse water management areas, IoT technologies enable continuous sensing and data acquisition from a myriad of distributed points. For instance, urban water networks [15] deploy sensors to monitor water flow, pressure, leakage, and quality parameters like turbidity and chlorine residuals, ensuring efficient distribution and early detection of infrastructure issues. Industrial wastewater facilities [16] use sensors to track chemical composition, temperature, and pH levels to comply with environmental regulations and optimize treatment processes. Similarly, in agriculture [17], soil moisture, nutrient levels, and weather data are collected to inform precision irrigation, minimizing water waste. These systems often utilize various sensors for pH, temperature, turbidity, water level, and water flow.

CPS architectures [19] are employed to create closed feedback loops between the physical environment and digital control, enabling automated adjustments. For example, in smart irrigation, sensor data automatically triggers actuators (like valves or pumps) to deliver water precisely where and when needed. In industrial settings, real-time water quality data can activate dosing pumps or filtration systems to maintain desired parameters. The communication backbone frequently relies on protocols like MQTT [20] for efficient, real-time data exchange from edge devices. For specialized applications, such as underwater monitoring in reservoirs or rivers, technologies like IoT-LoRa [21] are explored, though they face challenges with water type and turbidity.

Furthermore, DTs provide advanced monitoring, predictive analysis, and lifecycle management capabilities in these sectors. By creating virtual replicas of physical water infrastructure—be it a section of a city's pipeline, a specific industrial treatment unit, or an agricultural field—operators can simulate scenarios, predict maintenance needs, and optimize resource allocation. These systems often leverage Edge and Fog computing paradigms [22], with devices like Raspberry Pis and Arduinos [23] serving as aggregation nodes or local processing hubs, offloading computational responsibilities from central cloud servers and enabling faster response times for critical events. The processed data can then be sent to the cloud for extra storage and computational analysis.

A. Literature Review

Modern swimming pool management has already reached a considerable degree of automation [24]–[27]. In practice, machine rooms are typically equipped with sampling points, where sensors continuously measure water quality parameters such as pH, chlorine concentration, and turbidity. These readings are fed into a controller with sufficient computational power to analyze the data and activate actuators that dose chemicals or adjust filtration cycles to maintain water quality within predefined setpoints. Major vendors also provide mobile applications and integrated ecosystems that enable companies and pool operators to remotely monitor and adjust pool parameters. However, these systems are proprietary and vertically integrated, often limiting interoperability, extensibility, and long-term integration with broader smart environments.

Academic research, on the other hand, has explored swimming pool management mostly through prototypes and small-scale experiments. These studies often emphasize either water quality monitoring or safety/drowning prevention, presenting diverse architectures and sensing approaches. To provide a structured overview, Table I summarizes the main contributions of relevant works across these domains. The following subsections briefly review each research stream.

a) ***Water Quality Monitoring:*** Monitoring water quality [34] is a fundamental requirement for ensuring hygiene, comfort, and safety in swimming pools. Parameters such as pH, turbidity, temperature, and water level must be continuously monitored to maintain optimal conditions and meet health regulations. Recent works have explored automated, sensor-based systems for real-time tracking of these metrics. For instance, Hamid *et al.* [28] proposed a Smart Water Quality Monitoring System (SWQMS) capable of tracking pH and temperature in real time. Their findings confirmed that while temperature fluctuates throughout the day, pH levels remain relatively stable, underscoring the need for automated monitoring. However, their system was limited in scope, tracking only a narrow set of parameters and lacking integration with broader pool management functions. Expanding on this, Lakshmikantha *et al.* [29] introduced a more comprehensive

TABLE I. Summary of related works in swimming pool monitoring and management.

Author / Year	Focus Area	Parameters / Sensors	Architecture / Technology	Limitations
Hamid *et al.* [28]	Water quality monitoring	pH, temperature	SWQMS prototype, automated tracking	Limited scope (only two parameters); lacks explicit standardization (AAS) for interoperability or a comprehensive multi-level CPS architecture
Lakshmikantha *et al.* [29]	Water quality monitoring	pH, turbidity, temperature, level, flow	Multi-sensor IoT prototype	Prototype stage, limited validation; does not address standardized asset representation (AAS) or a comprehensive multi-level CPS architecture for lifecycle management
Sangeetha *et al.* [30]	Safety / drowning prevention	Ultrasonic, PIR	SSPMS with alarms and drainage	Safety-focused, no water quality integration; proprietary approach (implied), lacks standardized interoperability (AAS) or a holistic multi-level CPS framework
Raj *et al.* [31]	Safety + monitoring	Temperature, level, intoxication detection	ESP32-CAM with IoT connectivity	Limited scalability, prototype; primarily focused on safety with specific hardware, no explicit mention of open standards like AAS for broader interoperability or a comprehensive multi-level CPS architecture
Christopher *et al.* [32]	Communication challenges	LoRa signals underwater	IoT-LoRa tests in various water types	Performance degradation in seawater; focuses on communication layer, does not address overall system architecture, standardized asset representation (AAS), or interoperability with other systems
Glória *et al.* [33]	IoT architectures	Multiple environmental sensors	Raspberry Pi + Arduino via MQTT	Generic, not pool-specific; while exploring IoT gateways, it does not explicitly integrate standardized digital twins (AAS) or a multi-level CPS architecture for complex asset management and interoperability across varied smart environments

prototype that included monitoring of turbidity, water level, and flow. While their solution was more versatile and adaptable to other water contexts such as industrial wastewater, it remained largely a prototype, with no deployment-focused validation, energy optimization, or discussion on connectivity and security challenges.

*b) **Safety and Drowning Prevention**:* Safety and drowning prevention [35] are crucial components of intelligent pool systems, especially given the high rates of drowning-related incidents, particularly among children. To address this, researchers have developed systems that fuse environmental sensing and visual analysis to detect emergency situations in real time. Sangeetha *et al.* [30] presented a Smart Swimming Pool Management System (SSPMS) combining ultrasonic and Passive Infrared (PIR) sensors for real-time drowning detection and emergency response. Although promising, their system was reactive rather than predictive and relied on basic threshold-based logic, limiting robustness in complex scenarios. Raj *et al.* [31] advanced this concept by integrating ESP32-CAM-based [36] image analysis to identify intoxicated individuals or drowning victims, alongside water level and temperature sensing. Their system also featured emergency notifications. However, this approach depends heavily on stable lighting conditions and the accuracy of basic computer vision models, which can be unreliable in dynamic pool environments. Furthermore, privacy concerns associated with visual surveillance were not thoroughly addressed, nor was the system evaluated under real-world operating conditions.

*c) **Communication and Connectivity**:* Effective communication is critical in distributed sensing systems, especially in aquatic environments where signal propagation can be degraded by factors like water salinity, turbidity, and interference [37]. The use of long-range, low-power protocols such as LoRaWAN and MQTT [38] is gaining traction due to their energy efficiency and robustness in constrained environments. Christopher *et al.* [32] investigated IoT-LoRa [21] signal behavior in water environments and found that pool water, due to its lower salinity, supports better signal transmission compared to seawater. Their findings help guide protocol selection based on fluid properties. However, their analysis focused primarily on static conditions and lacked insights into performance under user-generated interference (e.g., people swimming), device mobility, or real-time data throughput. Additionally, their study did not explore multi-hop or hybrid communication models, which are often necessary in complex installations.

*d) **Architectural Innovations**:* Modern smart pool systems increasingly benefit from hybrid architectures that leverage the complementary strengths of Edge and Cloud computing [39]. These enable real-time decision-making at the Edge while facilitating advanced analytics and long-term storage in the Cloud. Glória *et al.* [33] demonstrated such a layered architecture using a Raspberry Pi as an IoT gateway and Arduino-based sensor nodes for data acquisition. Data communication was handled via the MQTT protocol [20], providing a lightweight and reliable channel for telemetry. While this architecture offers flexibility and modularity, it lacks built-in security mechanisms, which are critical when dealing with safety-critical applications. Moreover, their work focused primarily on system architecture without delving into scalability, fault tolerance, or orchestration aspects. Later work by Andriulo *et al.* [22] explored more sophisticated Cloud–Edge integration, but the computational burden on local devices remained a constraint, especially for tasks involving visual processing or real-time inference.

B. Summary and Gap

The reviewed literature highlights important contributions to monitoring and safety in swimming pools. However, most academic studies adopt an experimental or prototype perspective, often neglecting the fact that commercial pool systems already provide closed-loop automation in practice. Moreover, the academic focus tends to shift toward safety mechanisms (especially drowning prevention) rather than continuous, standardized, and interoperable water quality management [40]. This creates a gap between industrial practice and academic research. While industry offers proprietary ecosystems, research opportunities remain in the application of open, standardized approaches that support lifecycle management of all pool assets—including sensors, actuators, water, and chemicals.

A significant limitation observed across many of these smart water solutions, akin to commercial smart pool systems, is their tendency to be proprietary, closed, and vertically integrated. This often limits interoperability, data sharing, and seamless integration with broader smart environments or third-party services. This fragmentation hinders comprehensive lifecycle management of heterogeneous assets and impedes the realization of truly interconnected ecosystems.

This is where the AAS concept, central to the *SmartPool* solution, offers a transformative approach. As a standardized implementation of Digital Twins, AAS represents a mechanism to unify heterogeneous devices and data under a common data model, ensuring interoperability, scalability, and long-term maintainability. In urban water management, each pump, valve, or sensor could have its own AAS, allowing different manufacturers' equipment to communicate and be managed uniformly. In industrial wastewater, each chemical dosing unit, filter, or monitoring probe could expose its functions and data through a standardized AAS, facilitating integration with factory-wide Industry 4.0 systems [41]. The *SmartPool* system, by leveraging AAS to digitalize pool assets and enable bi-directional communication with standardized interfaces, demonstrates how open and interoperable architectures can enhance transparency, scalability, and integration with broader Industry 4.0 and smart city initiatives, effectively laying the foundation for more interoperable smart water systems.

In this regard, the present work leverages the concept of AAS as the standardized digital representation of assets. By adopting an AAS-based architecture implemented through *Eclipse BaSyx* [11], [12], *SmartPool* demonstrates how pool automation can move beyond proprietary solutions toward interoperable, research-driven, and lifecycle-oriented management.

III. Methodology

The development of the *SmartPool* system followed a structured Design Science Research (DSR) methodology [42], widely used in CPS and IoT engineering contexts. This methodology was organized into three iterative and interdependent phases: (1) problem analysis and requirement elicitation, (2) system design and technology selection, and (3) implementation and prototype validation. Each phase was informed by engineering best practices, existing standards, and empirical feedback from iterative development cycles.

A. Requirements Elicitation and Problem Definition

The first phase focused on systematically identifying the functional and non-functional requirements for a smart pool management system. Functional requirements were derived from the domain-specific needs of aquatic facilities and centered on the following objectives:

- Continuous sensing of water quality parameters (e.g., pH, temperature, turbidity);
- Automated actuation mechanisms for water circulation and dosing;
- Real-time monitoring and alerts for safety-critical events (e.g., falls or drowning risks);
- Integration of a unified user interface for administrators and end-users.

Non-functional requirements were equally considered, emphasizing:

- Interoperability, to allow seamless integration of heterogeneous sensors and actuators;
- Scalability, to support deployment across pools of varying size and complexity;
- DT compatibility, through the use of standardized digital representations of assets;
- Security and dependability, ensuring reliable and confidential communication across components.

These requirements were elicited through a combination of domain analysis, literature review, and benchmarking against existing smart pool systems.

B. Architectural Design and Technology Selection

The second phase applied architecture-driven engineering to explore and define the technical backbone of the *SmartPool* system. A comparative analysis of sensors, microcontrollers, communication protocols, and middleware platforms was conducted. Technologies were evaluated against a matrix of selection criteria, including compatibility with open standards, cost-effectiveness, modularity, and support for Industry 4.0 paradigms, particularly DTs and edge-cloud integration.

This process led to the adoption of a CPS-inspired architecture with the following characteristics:

- Edge Layer: Microcontroller-based sensor nodes and actuators responsible for local control and data acquisition;
- Fog Layer: A local processing hub (Raspberry Pi) acting as an MQTT broker and pre-processing node;
- Cloud Layer: Centralized services for data persistence, advanced analytics, and integration with user interfaces;
- DT Layer: Adoption of the AAS concept through the *Eclipse BaSyx* middleware [11], [12], enabling the standardized digital representation of physical pool components.

Communication protocols were selected to meet the dual goals of lightweight messaging and secure interoperability. MQTT over Transport Layer Security (TLS) encryption [43] was chosen for sensor-to-fog and fog-to-cloud communication, ensuring encrypted, publish/subscribe-based messaging with low latency.

C. System Implementation and Prototype Validation

In the final phase, a fully functional proof-of-concept prototype was implemented to validate the proposed architecture and assess system behavior under real-world constraints. The prototype integrated:

- Real-time water quality monitoring using calibrated sensors;
- Automated actuation based on rule-based logic (e.g., activating filters or chemical dispensers);
- Live video processing for human presence detection;
- Secure data transmission across all CPS layers;
- AAS-based digital twins for each key pool component, enabling dynamic querying and status tracking.

Validation involved both qualitative and quantitative evaluation, including:

- System responsiveness: Time-to-alert and actuation latency were measured for safety events and water quality thresholds.
- Communication performance: Throughput and reliability of MQTT communication under typical network loads.
- DT operations: Correctness of lifecycle events and API interactions via the AAS interface.
- User feedback: Functional testing of the User Interface (UI) with simulated users to assess usability and visualization of sensor data.

These validation efforts demonstrated the feasibility and coherence of the architecture, and also helped identify areas for future optimization, particularly in terms of energy efficiency and Artificial Intelligence (AI)-based analytics.

IV. SmartPool Solution

SmartPool was conceived as a research-driven platform that combines existing principles of pool automation with open-source frameworks and standardized DT representations. The system integrates multi-parameter sensing, real-time actuation, and lifecycle-oriented digitalization through the AAS. In doing so, *SmartPool* moves beyond the proprietary solutions offered by industry vendors, demonstrating how open and interoperable architectures can enhance transparency, scalability, and integration with broader Industry 4.0 ecosystems.

The architecture follows a five-level CPS-inspired structure [19] (Figure 1), which organizes perception, communication, middleware, application, and business functions. This design ensures modularity and separation of concerns, while supporting interoperability across heterogeneous devices and services.

a) ***Perception Level:*** The physical layer integrates all assets responsible for data acquisition and actuation. Water quality sensors measure parameters such as turbidity, pH, chlorine concentration, and temperature, while environmental sensors capture auxiliary data (e.g., light intensity). A camera provides real-time video streams for safety monitoring. Actuators regulate chemical dosing, lighting, and alarms, enabling both automatic control and safety notifications.

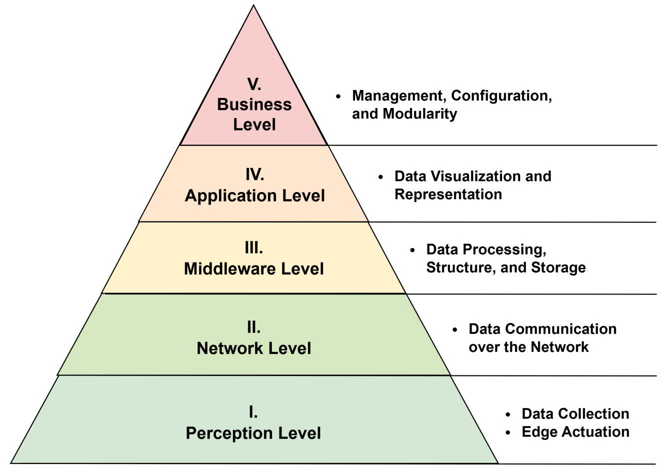

Figure 1. 5-Level Architecture

b) ***Network Level:*** This level manages connectivity between devices and higher layers. A central processing unit aggregates sensor data locally and coordinates actuation. Connectivity is supported by Wi-Fi [44] and communication protocols such as MQTT [45], OPC UA [46], [47], or Modbus TCP [48], ensuring compatibility with heterogeneous devices and infrastructures.

c) ***Middleware Level:*** Acting as the system's digital backbone, the middleware manages data processing, aggregation, and digitalization. The controller connects to a Fog node hosting **(i)** a message broker as intermediary for real-time data exchange, **(ii)** an AAS-based environment for standardized DT representations, and **(iii)** a time-series database for persistent storage. Each asset—including sensors, actuators, pool water, and dosing chemicals—is represented as an AAS instance, enabling lifecycle-oriented management.

Imagine every device in a pool – from the pH sensor to the filtration pump – doesn't just exist, but also carries a digital passport (its AAS). This passport isn't just a static document; it's a living, machine-readable record that contains all its essential information: its 'name' (identifier), technical specifications, real-time 'health' readings (sensor data), operational 'permissions' (actuator commands), and even its maintenance history.

Just like a country's border control system can quickly process individuals by scanning their standardized passports, the *SmartPool* system, using AAS, can seamlessly identify, monitor, and control any pool device, regardless of its manufacturer or specific communication protocol. Scaling 'SmartPool-as-a-Service' for multiple clients, it's like setting up a centralized agency that can issue, track, and manage these digital passports for thousands of pools. Each pool's collection of device passports ensures a uniform, secure, and interoperable way to manage everything, allowing for efficient, data-driven services on a large scale.

Through the AAS infrastructure, assets can be dynamically registered, monitored, and managed via standardized reposi-

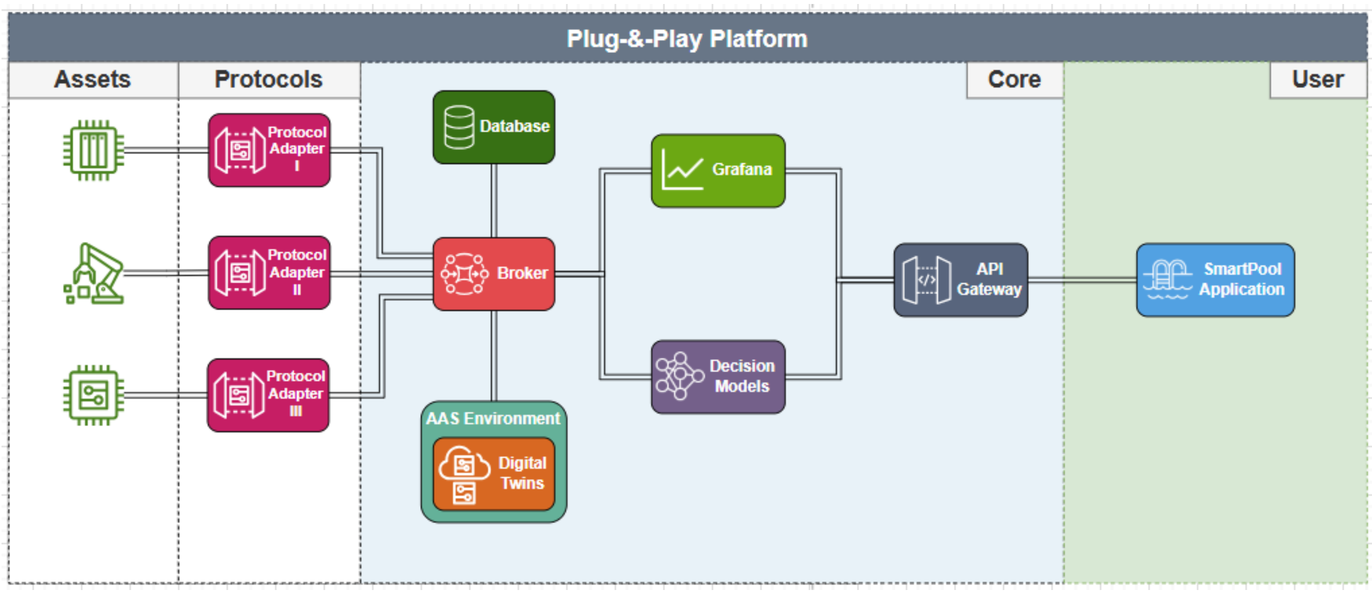

Figure 2. *SmartPool* System's Architecture.

tories and registries. The specification and representation of assets through AAS require several key considerations:

- **AAS Types and Instances:** The AAS framework distinguishes between *types* and *instances*. A type defines a general blueprint for an asset class, specifying its structure, mandatory *Submodels*, and semantic annotations. An instance, by contrast, represents a concrete, real-world manifestation of that asset, adhering to the schema defined by its corresponding type. Additionally, AAS types and instances are represented in an hierarchy deriving characteristics from parent representations [49].
- **Submodel Templates:** *Submodels* are collections of elements of information related to a specific aspect of an asset (e.g., technical data, digital nameplate, or lifecycle information) [49]. To ensure reusability and consistency, *Submodel Templates* are being defined and standardized by organizations such as ZVEI [50], the Industrial Digital Twin Association (IDTA) [51], and Plattform Industrie 4.0 [52]. These templates provide agreed structures for information exchange across domains, ensuring that assets of the same type can be represented in a consistent and machine-readable manner [53].
- **Semantic Descriptions:** To guarantee unambiguous interpretation of data, AAS elements must be semantically annotated. Semantic identifiers can be derived from established ontologies, data dictionaries, or classification systems such as IEC Common Data Dictionary (IEC CDD) and ECLASS [54], [55]. This semantic enrichment ensures that information is interoperable across systems and domains, supporting meaningful integration into wider ecosystems.
- **Interoperability:** Beyond digitalization, the AAS enables interoperability by providing standardized interfaces and semantics for asset communication. Achieving this requires collecting and harmonizing data from heterogeneous sources (e.g., sensors, controllers, and databases) into the AAS structure. Several open-source frameworks, including *Eclipse BaSyx* [56], CoreAAS, and PyI40AAS [57], actively support this integration by offering tools for modeling, registering, and exchanging AAS data.

d) ***Application Level:*** At this level, the *SmartPool* application provides real-time monitoring, historical dashboards, alerts, and control options. The application is designed to scale with the number of assets integrated into the system, supporting both individual pool owners and professional operators managing multiple facilities. Unlike commercial vendor-specific platforms, *SmartPool* emphasizes openness, allowing integration of safety features such as drowning detection (Section II) alongside water quality monitoring.

e) ***Business Level:*** The top layer focuses on decision support and lifecycle optimization. By leveraging AAS-based digital representations, *SmartPool* enables advanced analytics such as predictive maintenance, optimization of chemical usage, and integration with larger infrastructures (e.g., smart buildings or city-wide water management systems). This level transforms real-time monitoring into strategic insights for efficient, sustainable operation.

The components described across the five levels are illustrated in Figure 2, which depicts the *SmartPool* architecture and the interactions between its physical assets, middleware, and digital representations. The figure highlights how heterogeneous devices communicate through standardized interfaces, ultimately enabling interoperability, real-time monitoring, and lifecycle-oriented management of the pool system.

V. Prototype Validation

This section presents the proof-of-concept implementation of the *SmartPool* system introduced in the previous section.

The validation is structured into three parts: (i) monitoring and control of field assets, (ii) communication and digitalization of data into AAS-based DTs, and (iii) the *SmartPool* web environment created during implementation.

A. Monitoring and Control

As discussed in Section I, the prototype was constrained by limited hardware resources and the absence of laboratory conditions to fully replicate a pool environment. To represent the pool, a transparent plastic container was used, as shown in Figure 3.

The sensing setup included a temperature and humidity sensor (not waterproof, therefore positioned near the water surface), a light intensity sensor, and an ultrasonic sensor mounted at the top of the container to measure water level. The water level was calculated as the difference between the container height and the distance reported by the sensor.

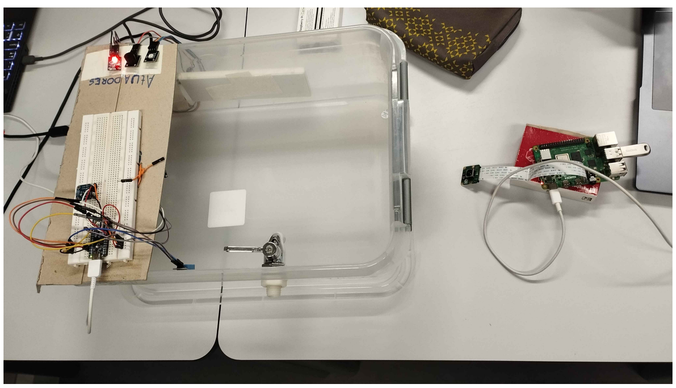

Figure 3. *SmartPool* Prototype Setup.

Actuation was represented by LEDs, which provided visual feedback when sensor readings exceeded predefined thresholds (e.g., red for low water level), and illuminated the container under low-light conditions. A buzzer simulated a safety alarm, alerting the user to the presence of entities such as a person or a pet near the water.

These devices were connected to microcontrollers responsible for data collection and actuation. Communication with the central node was established over Wi-Fi [44], using the MQTT protocol [45] to publish and subscribe data streams. Topics were organized hierarchically (e.g., `sensor/#` and `actuator/#`), with specific identifiers for parameters such as luminosity, distance, or temperature. For reliability, Arduinos [58] were configured with fault-tolerance mechanisms to attempt reconnection in case of Wi-Fi failures.

A Raspberry Pi acted as the primary processing and communication hub, equipped with a Raspberry Pi Camera Module 2 (Sony IMX219, 8 MP) [59] positioned above the container. In addition to capturing video, the Raspberry Pi executed object recognition using OpenCV [60] and the YOLOv8 model [61], detecting entities in the pool area and triggering alerts, thereby demonstrating the safety-enhancement capabilities of the system.

B. Communication and Digitalization

Sensor and actuator data were transmitted to a Fog node (a personal computer) hosting the middleware components. This included an MQTT broker for real-time data exchange and the *Eclipse BaSyx* framework [62] for AAS-based asset representation. The *BaSyx Databridge* [63] was used to map MQTT topics to AAS properties, enabling bi-directional event-based synchronization between physical assets and their digital counterparts. The *BaSyx Databridge* also supports integration with industrial protocols such as OPC UA, Kafka [64], and PLC drivers via *Apache PLC4X* [65], ensuring interoperability with heterogeneous systems.

For digitalization, AAS types and instances were created using the *AASX Package Explorer* [66]. Figure 4 shows the Unified Modeling Language (UML) model of AAS types and instances representing the prototype. The *PhysicalDevice* type, which includes *Submodels* such as *DigitalNameplate* [67] and *TechnicalData* [50], served as the basis for creating instances of each device.

Additional AAS types were defined for *Sensor* and *Actuator*, each with *Submodels* (*MeasurementData* and *ActuationData*, respectively). *Concept Descriptions* and semantic references were included to link *Submodel* elements to global semantic identifiers, ensuring machine-readable, unambiguous interpretation [49].

Data from the AAS instances were forwarded to InfluxDB via Telegraf [68], enabling persistent time-series storage. This allowed both real-time monitoring and historical analysis of the pool environment.

C. SmartPool Web Application

A user-friendly web application (Figure 5) was developed with React [69] and Vite [70]. The interface enables real-time visualization of sensor readings, historical dashboards (via Grafana [71]), alert notifications, and direct control of actuators. Notifications from the object detection system are also integrated through MQTT.

The design emphasized both usability and system functionality, ensuring operators can access key information quickly while maintaining the ability to act on system alerts. In this way, the *SmartPool* application illustrates how AAS-based digital representations can be made accessible to human operators in an intuitive form.

Finally, Figure 6 summarizes the complete prototype architecture, showing how the physical assets, middleware, AAS environment, and application layers are integrated to form a functioning end-to-end system.

VI. Discussion of Results

The implementation and validation of the *SmartPool* system demonstrated the feasibility and practical benefits of integrating CPS and standardized digital twin frameworks in the domain of aquatic facility management. The findings from the prototype not only confirm the functional validity of the proposed architecture but also highlight key contributions to

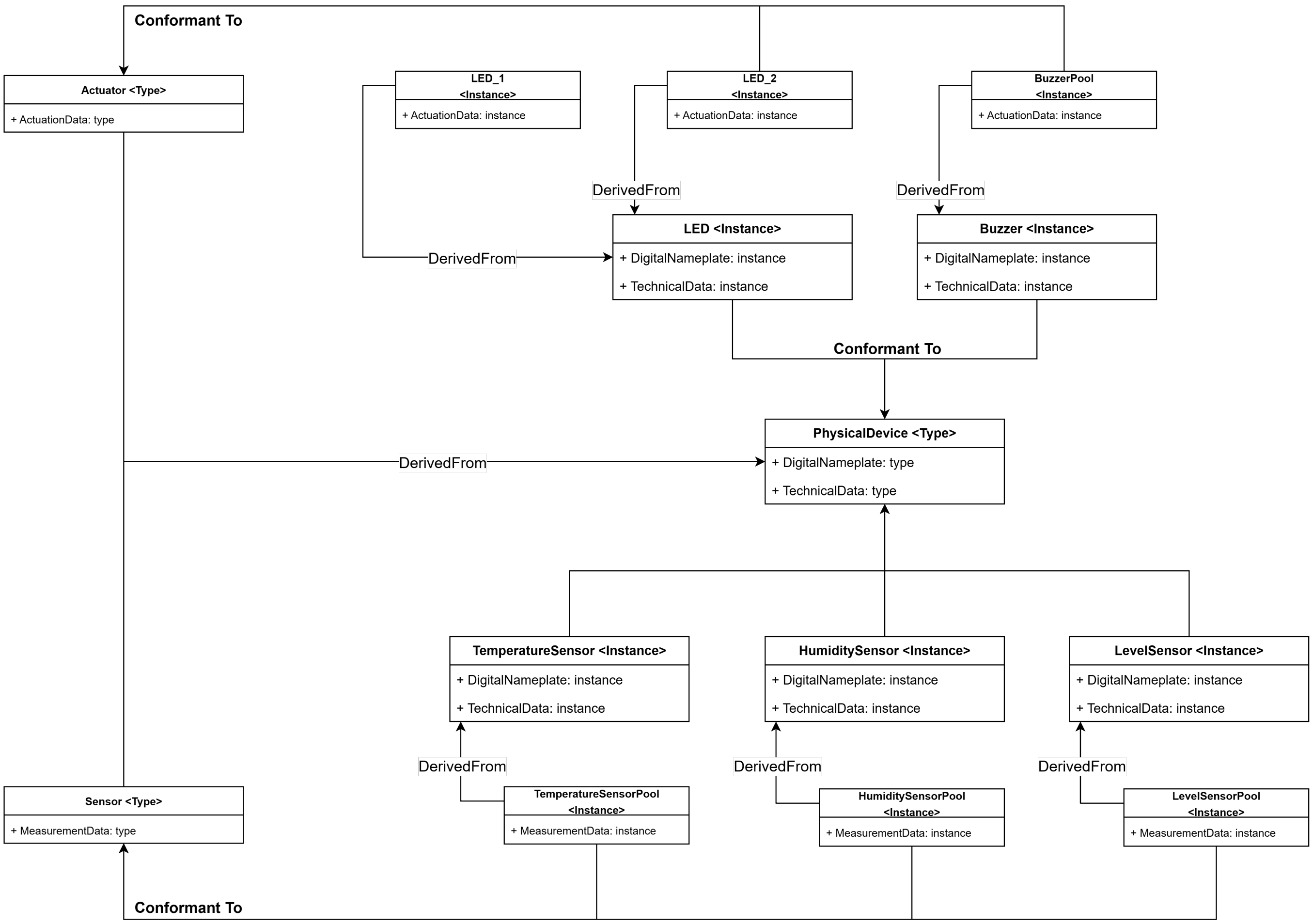

Figure 4. UML diagram of AAS types and instances for the prototype pool environment.

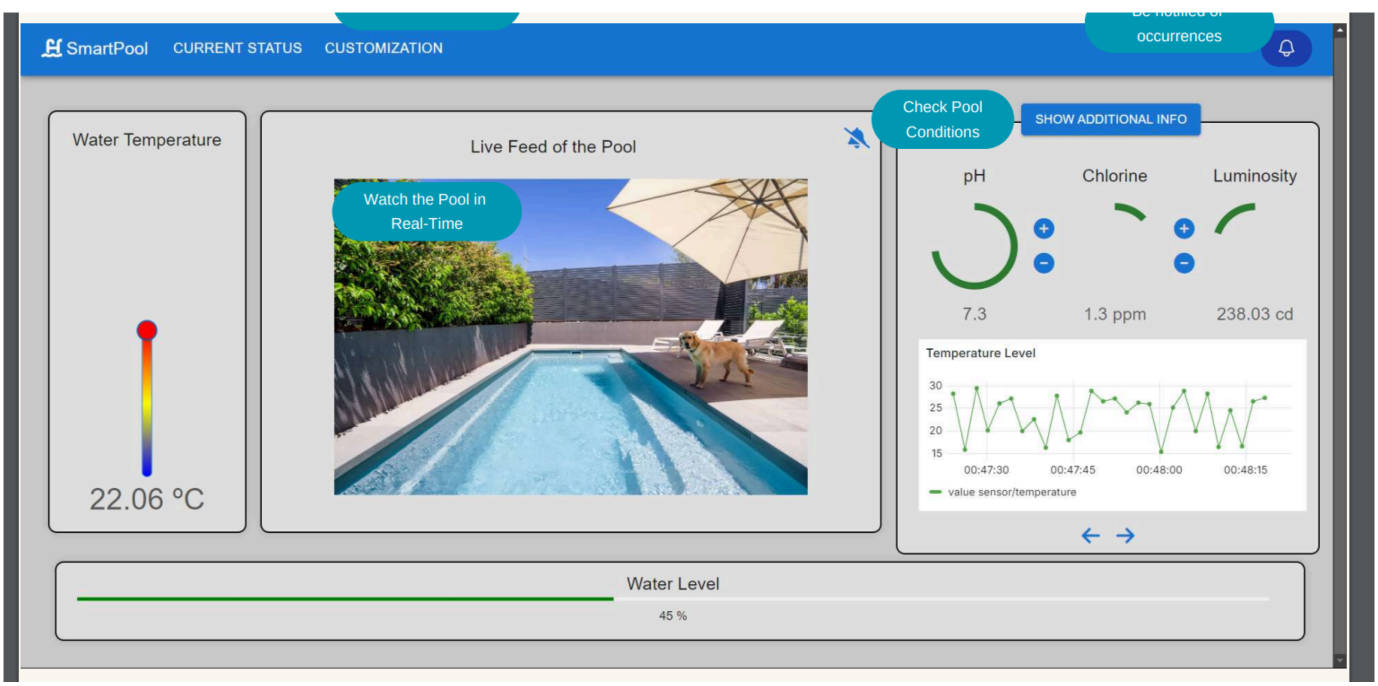

Figure 5. *SmartPool* Prototype Web Application.

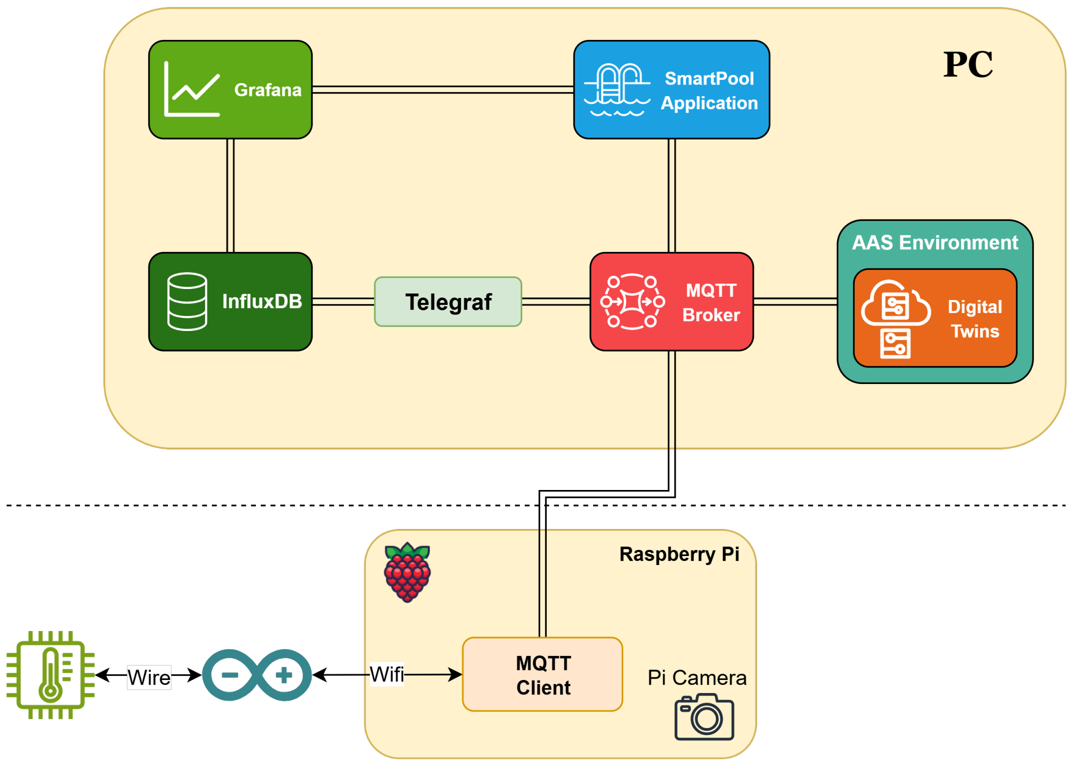

Figure 6. Prototypes System Architecture.

current knowledge and identify limitations that should be addressed in future work.

A. Insights from Prototype Validation

The experimental validation showed that the *SmartPool* architecture supports real-time sensing, actuation, and digital representation of pool components using lightweight, interoperable technologies. Key achievements include:

- Reliable data acquisition and actuation: The system demonstrated consistent and timely readings of water temperature and turbidity. Combined with automated rule-based actuation, this enables near real-time response to deviations in water quality, contributing to improved maintenance efficiency.
- Integration of visual safety monitoring: The edge-level video processing node detected human presence and movement in the pool area. While the current frame rate was limited (1 frame every 3 seconds), it proved sufficient to identify basic movement patterns for safety alerts, confirming the feasibility of edge-based drowning prevention mechanisms.
- DT representation with AAS: The deployment of AAS using the *Eclipse BaSyx* middleware enabled the standardized digital modeling of physical assets. This enhanced interoperability and reusability by allowing external applications to query asset status, historical logs, and live operational data via APIs.
- Secure communication via MQTT over TLS: The system maintained encrypted, authenticated communication between edge, fog, and cloud components, supporting secure sensor data handling in line with best practices for CPS and Industry 4.0 infrastructures.

These results support the core hypothesis that combining standardized DTs with CPS-based pool management offers a scalable, secure, and flexible solution for real-time monitoring and automation in aquatic environments.

B. Comparison with Existing Solutions

Compared to related works reviewed, *SmartPool* advances the state of the art in multiple dimensions:

- Water Quality Monitoring: Prior work (e.g., Hamid *et al.* [28] and Lakshmikantha *et al.* [29]) focused primarily on parameter tracking without integration into broader architectural frameworks. *SmartPool* extends this by embedding sensors within a layered CPS architecture, supporting actuation, lifecycle management, and cloud synchronization.
- Safety and Drowning Prevention: Unlike work such as Sangeetha *et al.* [30] and Raj *et al.* [31], which use sensor-triggered alarms or simple image capture, *SmartPool* leverages event-driven video processing at the edge, allowing more intelligent surveillance with reduced latency and bandwidth usage.
- Communication and Connectivity: Building on the insights from Christopher *et al.* [32], *SmartPool* demon-

strates effective use of MQTT over TLS within constrained aquatic environments, where traditional connectivity solutions often fail due to humidity, water reflection, or electromagnetic interference.
- Architectural Innovations: While prior IoT architectures (e.g., Glória *et al.* [33]) demonstrated the feasibility of decentralized control using Raspberry Pi and Arduino, *SmartPool* adds an additional layer of semantic interoperability via AAS, bridging the gap between raw data and contextualized asset management—a major leap toward Industry 4.0 integration.

SmartPool not only reproduces functionalities found in the literature but integrates them cohesively into a standards-based, modular, and future-proof architecture, a distinguishing feature in this domain.

C. Contributions and Limitations

This work advances both applied and theoretical understanding in the following ways:

- Demonstrates the application of AAS DTs beyond industrial production, validating its applicability in non-traditional CPS domains such as aquatic environments;
- Provides an end-to-end reference architecture for CPS-enabled aquatic management systems, combining Edge, Fog, and Cloud layers with secure data pipelines and modular control logic;
- Contributes an open and extensible prototype that can serve as a baseline for researchers exploring IoT-based safety, sustainability, and DT frameworks in smart environments;
- Offers a methodology that can be generalized to other safety-critical settings (e.g., public fountains, recreational facilities), reinforcing the role of CPS and interoperability standards in public infrastructure management.

Despite its contributions, *SmartPool* still presents several limitations:

- Limited AI Capabilities: While basic video analysis was implemented, more sophisticated ML-based behavior recognition (e.g., distress posture detection, fall prediction) requires both more powerful hardware and annotated datasets, which were outside the scope of this prototype.
- Frame Rate Constraints: The current processing pipeline limits video analysis to 3-second intervals, reducing responsiveness in fast-evolving drowning scenarios. Future versions should optimize the video subsystem or integrate hardware accelerators.
- Energy Consumption: No energy optimization was implemented at the prototype stage. Power-hungry components such as the camera and WiFi module may challenge long-term deployments in solar-powered or remote setups.
- Scalability: The system was tested with a single pool setup. Real-world scenarios with multiple pools and users may require load balancing, data deduplication, and improved concurrency handling across the middleware stack.
- Security and Identity Management: Although TLS and JSON Web Tokens (JWT) [72] were implemented, a full Public Key Infrastructure (PKI) [73] was not realized. Identity federation, certificate rotation, and fine-grained access control are planned as future enhancements.

VII. Conclusion & Future Work

This research introduced *SmartPool*, a modular and standards-based prototype platform for intelligent swimming pool management, combining CPS principles with AAS DT representations. By integrating real-time sensing, automated actuation, and edge-level visual surveillance within a secure and interoperable architecture, the system provides a novel approach to pool monitoring, safety, and lifecycle management.

The prototype validated the feasibility of applying CPS and Industry 4.0 paradigms in non-industrial contexts such as recreational aquatic facilities. Key outcomes included accurate real-time tracking of water quality parameters, responsive actuation logic, and the demonstration of digital asset models for pool components. Furthermore, the inclusion of edge-based visual monitoring reinforced the potential for enhancing safety in semi-supervised environments.

In addition to its technical achievements, *SmartPool* contributes new insights into:

- The application of AAS-based DTs beyond manufacturing environments.
- The use of lightweight, modular architectures to enable real-time control and monitoring in resource-constrained CPS deployments.
- The integration of safety-enhancing features, such as drowning detection and anomaly response, into a unified, standards-compliant middleware stack.

However, as with many academic proofs-of-concept, the system faces several known limitations. These include latency in edge-based video analysis, modest scalability, simplified environmental conditions, and the absence of fully implemented cybersecurity frameworks. These limitations underscore the challenges of balancing responsiveness, complexity, and energy efficiency in real-world deployments.

To build on these findings, several promising avenues for future research and development are proposed:

1) **Edge AI and Smart Perception:** Integrating optimized machine learning models (e.g., TinyML [74] or YOLOv5-tiny [75]) for local event recognition could improve detection capabilities while maintaining real-time responsiveness. This would enable behavior-based alarms (e.g., drowning postures, unauthorized access) and predictive maintenance based on sensor patterns.
2) **Secure and Trustworthy CPS:** Implementing a certificate-based PKI for fog and middleware nodes, along with TLS-secured MQTT communication and node-level authentication, will significantly enhance trust, integrity, and data confidentiality in distributed deployments. Additionally, availability concerns in edge scenarios must be addressed with fault-tolerant fallback routines, while data integrity mechanisms—such

as checksums, hash verification, and immutable logging—should be adopted to mitigate tampering risks.

3) **Energy Efficiency and Sustainability:** Exploring energy harvesting (e.g., solar panels) and ultra-low-power sensing technologies can reduce the environmental footprint and make the solution viable for remote or off-grid installations.
4) **Scalability and Multi-Pool Coordination:** Extending the platform to manage multiple pools simultaneously—or pools within large public/commercial facilities—will test its robustness and enable integration into broader smart building or smart city ecosystems.
5) **AAS-Integrated Analytics and Decision Support:** The AAS framework can be extended to include not only real-time parameters but also predictive indicators, alerts, and lifecycle metadata, transforming digital twins into decision-support agents rather than mere data containers.
6) **Control Loop Optimization and Adaptive Logic:** While the current system employs threshold-based actuation with fixed margins, future work could evaluate more sophisticated control strategies. These include PID or fuzzy logic control to ensure stable chemical dosing, model-predictive control (MPC) for resource optimization, and reinforcement learning approaches that adapt actuation parameters based on evolving environmental feedback.
7) **Broader Applicability in Water Environments:** The SmartPool architecture is not limited to recreational pools. Its modularity and asset-oriented design make it adaptable to other domains, such as aquaculture monitoring, thermal bath management, water purification facilities, and even environmental sensing in lakes or reservoirs. Future deployments could explore domain-specific adjustments to sensing, control, and safety logic to validate cross-context adaptability.

In conclusion, this research demonstrates the potential of integrating IoT, CPS, and AAS-based digital twins into intelligent water management systems. It contributes to both engineering practice and scientific knowledge by illustrating how lifecycle-oriented, secure, and interoperable architectures can be adopted in traditionally fragmented, non-industrial domains. As smart environments and sustainable infrastructure initiatives evolve, solutions like *SmartPool* can serve as foundational blueprints for scalable, intelligent, and context-aware water management.

Acknowledgment

This work is financially supported by national funds through the FCT/MCTES (PIDDAC), under the Associate Laboratory Advanced Production and Intelligent Systems – ARISE LA/P/0112/2020 (DOI 10.54499/LA/P/0112/2020) and the Base Funding (UIDB/00147/2020) and Programmatic Funding (UIDP/00147/2020) of the R&D Unit Center for Systems and Technologies – SYSTEC.

References

[1] A. Ávila *et al.*, "Smartpool: An automated cps-based system for real-time water quality management," in *INTELLI 2025, The Fourteenth International Conference on Intelligent Systems and Applications*, IARIA, 2025.

[2] RubyHome Blog, "Swimming pool statistics (2025)," 2025, [Online]. Available: https://www.rubyhome.com/blog/swimming-pool-stats/ (visited on 12/26/2025).

[3] A. Jemat, S. Yussof, S. S. Sameon, and N. A. Alya Rosnizam, "IoT-Based System for Real-Time Swimming Pool Water Quality Monitoring," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13051 LNCS, pp. 332–341, 2021, Cited by: 3. DOI: 10.1007/978-3-030-90235-3_29.

[4] A. Angdresey, L. Sitanayah, and V. J. A. Sampul, "Monitoring and Predicting Water Quality in Swimming Pools," *EPI International Journal of Engineering*, vol. 8, no. 2, pp. 119–125, 2020, Accessed from https://doi.org/10.25042/epi-ije.082020.05. DOI: 10.25042/epi-ije.082020.05.

[5] F. Tao, Q. Qi, L. Wang, and A. 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, 2019.

[6] A. Parnianifard *et al.*, "Digital-twins towards cyber-physical systems: A brief survey," *Engineering Journal*, vol. 26, no. 9, pp. 47–61, 2022.

[7] IEC-63278, *Asset Administration Shell for Industrial Applications - Part 1: Asset Administration Shell Structure*, en, International Standard, Dec. 2023. DOI: 9782832276792.

[8] M. Kaur, V. Mishra, and P. Maheshwari, "The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action," in *Internet of Things*, Springer, Cham, 2020, pp. 3–17. DOI: 10.1007/978-3-030-18732-3_1.

[9] A. Redelinghuys, A. Basson, and K. Kruger, "A six-layer architecture for the digital twin: a manufacturing case study implementation," *Journal of Intelligent Manufacturing*, vol. 31, no. 6, pp. 1383–1402, 2020. DOI: 10.1007/s10845-019-01516-6.

[10] S. Zeb *et al.*, "Industrial digital twins at the nexus of NextG wireless networks and computational intelligence: A survey," *Journal of Network and Computer Applications*, vol. 200, p. 103 309, 2022, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2021.103309.

[11] S. Karthik, D. Priya E.L., G. Anand K.R., and A. Sharmila, "IoT Based Safety Enhanced Swimming Pool with Embedded Techniques to reduce drowning accidents," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 843–847. DOI: 10.1109/ICOSEC49089.2020.9215247.

[12] S. Kannoth *et al.*, "Enabling smes to industry 4.0 using the basyx middleware: A case study," in *European Conference on Software Architecture*, Springer, 2021, pp. 277–294.

[13] P. Juhás and K. Molnár, "Key components of the architecture of cyber-physical manufacturing systems," *Industry 4.0*, vol. 2, no. 5, pp. 205–207, 2017.

[14] S. Ismail, D. W. Dawoud, N. Ismail, R. Marsh, and A. S. Alshami, "Iot-based water management systems: Survey and future research direction," *IEEE Access*, vol. 10, pp. 35 942–35 952, 2022.

[15] K. Joseph, A. K. Sharma, and R. Van Staden, "Development of an intelligent urban water network system," *Water*, vol. 14, no. 9, p. 1320, 2022.

[16] J. Y. Uwamungu *et al.*, "Future of water/wastewater treatment and management by industry 4.0 integrated nanocomposite

manufacturing," *Journal of Nanomaterials*, vol. 2022, no. 1, p. 5 316 228, 2022.

[17] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, "Iot-based smart irrigation systems: An overview on the recent trends on sensors and iot systems for irrigation in precision agriculture," *Sensors*, vol. 20, no. 4, p. 1042, 2020.

[18] M. Singh and S. Ahmed, "Iot based smart water management systems: A systematic review," *Materials Today: Proceedings*, vol. 46, pp. 5211–5218, 2021.

[19] D. G. Pivoto *et al.*, "Cyber-physical systems architectures for industrial internet of things applications in industry 4.0: A literature review," *Journal of Manufacturing Systems*, vol. 58, pp. 176–192, 2021.

[20] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of things: Survey and open issues of mqtt protocol," in *2017 International Conference on Engineering & MIS (ICEMIS)*, Ieee, 2017, pp. 1–6.

[21] S. Devalal and A. Karthikeyan, "Lora technology-an overview," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 2018, pp. 284–290.

[22] F. C. Andriulo, M. Fiore, M. Mongiello, E. Traversa, and V. Zizzo, "Edge computing and cloud computing for internet of things: A review," in *Informatics*, MDPI, vol. 11, 2024, p. 71.

[23] R. Singh, A. Gehlot, L. R. Gupta, B. Singh, and M. Swain, *Internet of things with Raspberry Pi and Arduino*. CRC Press, 2019.

[24] J. M. Marais, D. V. Bhatt, G. P. Hancke, and T. Ramotsoela, "A web-based swimming pool information and management system," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, IEEE, 2016, pp. 980–985.

[25] A. Alotaibi, "Automated and intelligent system for monitoring swimming pool safety based on the iot and transfer learning," *Electronics*, vol. 9, no. 12, p. 2082, 2020.

[26] Á. de la Puente-Gil, M. de Simón-Martín, A. González-Martínez, A.-M. Diez-Suárez, and J.-J. Blanes-Peiró, "The internet of things for the intelligent management of the heating of a swimming pool by means of smart sensors," *Sensors*, vol. 23, no. 5, p. 2533, 2023.

[27] G. Simões, C. Dionísio, A. Glória, P. Sebastião, and N. Souto, "Smart system for monitoring and control of swimming pools," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, IEEE, 2019, pp. 829–832.

[28] S. A. Hamid *et al.*, "IoT based Water Quality Monitoring System and Evaluation," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2020, pp. 102–106. DOI: 10.1109/ICCSCE50387.2020.9204931.

[29] V. Lakshmikantha *et al.*, "IoT based smart water quality monitoring system," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 181–186, 2021, International Conference on Computing System and its Applications (ICCSA- 2021), ISSN: 2666-285X. DOI: https://doi.org/10.1016/j.gltp.2021.08.062.

[30] A. Sangeetha *et al.*, "Smart Swimming Pool Management System (SSPMS) using IoT," in *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, 2023, pp. 840–846. DOI: 10.1109/ICIDCA56705.2023.10099729.

[31] K. J. S. Raj *et al.*, "Enhancing Pool Safety and Efficiency with an IoT Supported Monitoring System," in *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, 2023, pp. 1232–1237. DOI: 10.1109/ICPCSN58827.2023.00208.

[32] J. P. Christopher, S. D. Damayanti, and M. Suryanegara, "Investigating IoT-LoRa Technology for The Underwater System Application," in *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 2023, pp. 1–4. DOI: 10.1109/ICRAIE59459.2023.10468069.

[33] A. Glória, F. Cercas, and N. Souto, "Design and implementation of an IoT gateway to create smart environments," *Procedia Computer Science*, vol. 109, pp. 568–575, 2017, 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.05.343.

[34] M. Pule, A. Yahya, and J. Chuma, "Wireless sensor networks: A survey on monitoring water quality," *Journal of Applied Research and Technology*, vol. 15, no. 6, pp. 562–570, 2017.

[35] W.-C. Kao, Y.-L. Fan, F.-R. Hsu, C.-Y. Shen, and L.-D. Liao, "Next-generation swimming pool drowning prevention strategy integrating ai and iot technologies," *Heliyon*, vol. 10, no. 18, 2024.

[36] N. Cameron, "Esp32 microcontroller," in *ESP32 Formats and Communication: Application of Communication Protocols with ESP32 Microcontroller*. Berkeley, CA: Apress, 2023, pp. 1–54, ISBN: 978-1-4842-9376-8. DOI: 10.1007/978-1-4842-9376-8_1.

[37] F. P. F. Domingos, A. Lotfi, I. K. Ihianle, O. Kaiwartya, and P. Machado, "Underwater communication systems and their impact on aquatic life—a survey," *Electronics*, vol. 14, no. 1, p. 7, 2024.

[38] Lalhriatpuii, Ruchi, and V. Wasson, "Comprehensive exploration of iot communication protocol: Coap, mqtt, http, lorawan and amqp," in *International Conference on Machine Learning Algorithms*, Springer, 2024, pp. 261–274.

[39] D. Rosendo, A. Costan, P. Valduriez, and G. Antoniu, "Distributed intelligence on the edge-to-cloud continuum: A systematic literature review," *Journal of Parallel and Distributed Computing*, vol. 166, pp. 71–94, 2022.

[40] M. Elgorma *et al.*, "A review of methods for detecting and preventing drowning incorporating various techniques, devices and technologies," in *2nd International Conference on Electrical Engineering and Automatic Control*, 2024, pp. 1–6.

[41] H. Cañas, J. Mula, M. Díaz-Madroñero, and F. Campuzano-Bolarín, "Implementing industry 4.0 principles," *Computers & Industrial Engineering*, vol. 158, p. 107 379, 2021.

[42] J. Vom Brocke, A. Hevner, and A. Maedche, "Introduction to design science research," in *Design science research. Cases*, Springer, 2020, pp. 1–13.

[43] K. Keshkeh, A. Jantan, K. Alieyan, and U. M. Gana, "A review on tls encryption malware detection: Tls features, machine learning usage, and future directions," in *International Conference on Advances in Cyber Security*, Springer, 2021, pp. 213–229.

[44] S. S. Salwe and K. K. Naik, "Heterogeneous Wireless Network for IoT Applications," *IETE Technical Review*, vol. 36, pp. 61–68, 2019. DOI: 10.1080/02564602.2017.1400412.

[45] A. Banks and R. Gupta, "MQTT Version 3.1.1," 2014, [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html (visited on 12/26/2025).

[46] OPC Foundation, *Opc unified architecture (opc ua)*, Accessed: 2025-12-26, 2024.

[47] IEC-62541, *OPC Unified Architecture - Part 1: Overview and concepts*, en, International Standard, Nov. 2020. DOI: 9782832290767.

[48] Apache PLC4X, *Apache PLC4X Modbus Protocol Guide*, Accessed: 2025-12-26, 2019.

[49] P. Industrie 4.0, *Details of the Asset Administration Shell - Part 1*, en, May 2022.

[50] ZVEI, *Submodel Templates of the Asset Administration Shell*, https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2020/Dezember/Submodel_Templates_

of_the_Asset_Administration_Shell/201117_I40_ZVEI_SG2_Submodel_Spec_ZVEI_Technical_Data_Version_1_1.pdf, Accessed: 2025-12-26, 2020.

[51] E. Barnstedt *et al.*, "Open source drives digital twin adoption," *IIC J. Innov*, 2021.

[52] E. Tantik and R. Anderl, "Integrated data model and structure for the asset administration shell in industrie 4.0," *Procedia Cirp*, vol. 60, pp. 86–91, 2017.

[53] T. Abdel-Aty, E. Negri, and S. Galparoli, "Asset Administration Shell in Manufacturing: Applications and Relationship with Digital Twin," *IFAC-PapersOnLine*, vol. 55, pp. 2533–2538, 2022, Issue: 10. DOI: 10.1016/j.ifacol.2022.10.090.

[54] IEC-62264, *Enterprise-control system integration Part 1: Models and terminology*, en, International Standard, May 2013. DOI: 9782832208335.

[55] eCl@ss, *Ecl@ss – the standard for product and service classification*, Accessed: 2025-12-26, 2020.

[56] Eclipse Foundation, *Eclipse BaSyx Wiki*, https://wiki.basyx.org/en/latest/, Accessed: 2025-12-26, 2024.

[57] W. Quadrini, C. Cimino, T. Abdel-Aty, L. Fumagalli, and D. Rovere, "Asset Administration Shell as an interoperable enabler of Industry 4.0 software architectures: A case study," *Procedia Computer Science*, vol. 217, pp. 1794–1802, 2022. DOI: 10.1016/j.procs.2022.12.379.

[58] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd. Sebastopol, CA, USA: Maker Media, Inc., 2014, ISBN: 978-1449363338.

[59] M. Pagnutti *et al.*, "Laying the foundation to use raspberry pi 3 v2 camera module imagery for scientific and engineering purposes," *Journal of Electronic Imaging*, vol. 26, no. 1, pp. 013 014–013 014, 2017.

[60] OpenCV Team, "OpenCV: Open Source Computer Vision Library," 2025, [Online]. Available: https://opencv.org/ (visited on 12/26/2025).

[61] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, "A review on yolov8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*, Springer, 2024, pp. 529–545.

[62] E. Foundation, "Eclipse BaSyx Documentation," 2023, [Online]. Available: https://www.eclipse.org/basyx/ (visited on 12/26/2025).

[63] Eclipse BaSyx, *Databridge - Eclipse BaSyx Component*, https://wiki.basyx.org/en/latest/content/user_documentation/basyx_components/databridge/index.html, Accessed: 2025-12-26, 2024.

[64] N. Garg, *Apache kafka*. Packt Publishing, 2013.

[65] Apache PLC4X, *Apache PLC4X*, https://plc4x.apache.org/, Accessed: 2025-12-26, 2017.

[66] IDTA, *AASX Package Explorer*, https://github.com/admin-shell-io/aasx-package-explorer, Accessed: 2025-12-26, 2020.

[67] ZVEI, *THE DIGITAL NAMEPLATE CONSISTENT, SUSTAINABLE, FUTURE-PROOF, NETWORKED*, https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2020/November/Das_Digitale_Typenschild_-_ZVEI-Empfehlung/Digital_Nameplate.pdf, Accessed: 2025-12-26, 2020.

[68] InfluxData, "InfluxDB Documentation," 2025, [Online]. Available: https://docs.influxdata.com/influxdb/ (visited on 12/26/2025).

[69] Meta Platforms, Inc., "React: A JavaScript Library for Building User Interfaces," 2025, [Online]. Available: https://reactjs.org/ (visited on 12/26/2025).

[70] Evan You and Vite Contributors, "Vite: Next Generation Frontend Tooling," 2025, [Online]. Available: https://vitejs.dev/ (visited on 12/26/2025).

[71] S. Kirešová *et al.*, "Grafana as a visualization tool for measurements," in *2023 IEEE 5th International Conference on Modern Electrical and Energy System (MEES)*, IEEE, 2023, pp. 1–5.

[72] P. Mahindraka, "Insights of json web token," *International International Journal of Recent Technology and Engineering (IJRTE) ISSN*, pp. 2277–3878, 2020.

[73] J. Höglund, S. Lindemer, M. Furuhed, and S. Raza, "Pki4iot: Towards public key infrastructure for the internet of things," *Computers & Security*, vol. 89, p. 101 658, 2020.

[74] Y. Abadade *et al.*, "A comprehensive survey on tinyml," *IEEE Access*, vol. 11, pp. 96 892–96 922, 2023.

[75] T. Huang, M. Cheng, Y. Yang, X. Lv, and J. Xu, "Tiny object detection based on yolov5," in *Proceedings of the 2022 5th International Conference on Image and Graphics Processing*, 2022, pp. 45–50.