FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Application of Asset Administration Shell for Scalable IIoT System Design

André Correia da Costa

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. António Lucas Soares

Second Supervisor: Prof. Filipe David Ferreira

July 22, 2025

Application of Asset Administration Shell for Scalable IIoT System Design

André Correia da Costa

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. José Magalhães Cruz

Referee: Prof. José Barbosa

Referee: Prof. António Soares

July 22, 2025

Resumo

O rápido desenvolvimento da Internet Industrial das Coisas, dos Sistemas Ciber-Físicos e dos Gêmeos Digitais trouxe um novo interesse ao mercado sobre a transformação digital. No entanto, apesar da intensidade teórica apresentada pela literatura atual, a digitalização continua a ser um desafio significativo, sobretudo em ambientes complexos do mundo real, caracterizados por arquiteturas fragmentadas, iniciativas multi-organizacionais, sistemas *legacy* e baixos níveis de integração tecnológica.

Esta dissertação aborda esta falha, apresentando soluções digitais modulares, escaláveis e interoperáveis e validando uma delas num ambiente de cultivo de algas. Motivado pelos obstáculos observados na estrutura tecnológica de uma iniciativa multidisciplinar que promove a aquacultura sustentável, este trabalho explora duas abordagens distintas para resolver a mesma deficiência de formas diferentes: uma arquitetura baseada em conectores que utiliza o Apache PLC4X e o Kafka Connect para a ligação de protocolos, e uma arquitetura baseada no Asset Administration Shell (AAS) para a representação digital semanticamente enriquecida e que utiliza por base o Eclipse BaSyx para a integração com uma infraestrutura IIoT.

A solução baseada em conectores centra-se na instalação rápida e na facilidade de integração com uma Plataforma IIoT. Esta abordagem estabelece uma comunicação bidirecional entre um servidor OPC UA e um broker de mensagens Kafka. Permite a ligação entre protocolos e uma integração total com a estrutura de dados da plataforma. Foi criada uma Interface de Utilizador de conectores personalizada para gerir e simplificar a configuração destes, apoiando a rápida implementação e a adaptabilidade do sistema.

Em alternativa, uma solução baseada no conceito do AAS introduz uma representação standardizada e semanticamente enriquecida dos ativos, tirando partido do Eclipse BaSyx para gerir os componentes do AAS e coordenar o intercâmbio de dados com as tecnologias operacionais. O desenvolvimento de uma aplicação RESTful Spring Boot facilitou a comunicação bidirecional entre os elementos AAS e um controlador lógico programável Siemens S7, permitindo a monitorização em tempo real e o controlo remoto do ambiente de cultivo.

Esta última solução foi implementada e validada num ambiente de cultivo de algas, resultando num ecossistema operacional totalmente digitalizado. Suporta a gestão do ciclo de vida dos ativos, a integração IIoT através de MQTT, o armazenamento de dados em séries temporais e a monitorização em tempo real do sistema através de ferramentas de painéis de visualização - demonstrando o seu potencial de reutilização em configurações de produção igualmente complexas.

As principais contribuições incluem o desenvolvimento de protótipos funcionais, modulares, escalares e personalizados e estratégias práticas de integração para infraestruturas *legacy*. As conclusões sublinham a importância de abordagens digitais reproduzíveis para acelerar a digitalização industrial. O trabalho conclui oferecendo perspectivas acionáveis e delineando futuras rotas de pesquisa para o avanço de soluções interoperáveis e compatíveis com a Indústria 4.0.

Palavras-chave: Asset Administration Shell, Conectores Kafka, Cultivo de Algas, Internet Industrial das Coisas, Interoperabilidade.

Abstract

The rapid development of the Industrial Internet of Things, Cyber-Physical Systems, and Digital Twins brought new interest into the market about digital transformation. However, despite the theoretical maturity provided by the current literature, digitalization remains a significant challenge, particularly in complex, real-world environments characterized by fragmented architectures, multi-organizational initiatives, legacy systems, and low levels of technological integration.

This dissertation addresses this gap by presenting scalable, modular, and interoperable digital solutions and validating one of them within an algae cultivation environment. Motivated by the constraints observed in the technological stack of a broader multidisciplinary initiative promoting sustainable aquaculture, this work explores two distinct approaches to address the same gap in different ways: a connector-based architecture using Apache PLC4X and Kafka Connect for protocol bridging, and an Asset Administration Shell (AAS)-based architecture for semantically enriched digital representation and leveraging Eclipse BaSyx for IIoT infrastructure integration.

The connector-based solution focuses on rapid deployment and ease of integration with an IIoT Platform. This approach establishes bidirectional communication between an OPC UA server and a Kafka message broker. It enables protocol bridging and seamless integration with the data structure of the Platform. A custom-developed Connector UI was created to manage and simplify the configuration of these connectors, supporting rapid deployment and system adaptability.

In contrast, the AAS-based solution introduces a standardized, semantically enriched representation of assets, leveraging Eclipse BaSyx to manage AAS components and coordinate data exchange with operational technologies. A custom RESTful Spring Boot application facilitates bidirectional communication between the AAS elements and a Siemens S7 Programmable Logic Controller, enabling real-time monitoring and remote control of the cultivation environment.

This latter solution was implemented and validated within an algae cultivation environment, resulting in a fully digitized operational ecosystem. It supports asset lifecycle management, IIoT integration through MQTT, time-series data storage, and live system visualization via dashboard tools—demonstrating its potential for reuse in similarly complex production settings.

Key contributions include the development of functional prototypes, modular, scalable, and customizable, and practical integration strategies for legacy infrastructures. The findings underscore the importance of reproducible digital approaches in accelerating industrial digitalization. The work concludes by offering actionable insights and outlining future research pathways for advancing interoperable, Industry 4.0-compliant solutions.

Keywords: Algae Cultivation, Asset Administration Shell, Industrial Internet of Things, Interoperability, Kafka Connectors.

UN Sustainable Development Goals

The research presented in this dissertation contributes to the United Nations Sustainable Development Goals (SDGs) through a twofold approach: (1) by advancing digital transformation in complex industrial environments and multilayered organizations through modular, interoperable, and scalable IIoT solutions; and (2) by directly supporting the sustainable innovation of aquaculture systems, particularly in algae cultivation. This intersection addresses technological innovation but also undertakes responsible production practices and supports food system resilience.

The project's outcomes—ranging from real-time monitoring and control infrastructures to semantic and standardized digital twins for asset lifecycle management—have practical implications for sustainable development across multiple sectors. The work has been embedded in a multi-organizational initiative focused on ecological aquaculture, further underscoring its alignment with long-term environmental, economic, and societal objectives. Table 1 presents the most relevant SDGs that this dissertation targets in a direct or indirect manner.

SGD	Target	Contribution	Performance Indicators and Metrics
9	9.5	Development of modular and scalable solutions, enabling digitalization in legacy infrastructure.	Integration of such solutions into legacy systems, multiorganizational environments, and production ecosystems.
17	17.16	Strengthens multi-institutional collaboration through participation in a European project. Encourages the development of open-source frameworks and reusable methodologies that can be scaled across sectors.	Number of collaborating institutions; quantity and quality of dissemination activities; joint workshops and technology transfers.
2, 13	2.4, 13.3	Promotes sustainable and carbon-efficient food production systems by digitizing and optimizing algae cultivation. Supports higher productivity, improved monitoring, and more precise control of inputs and outputs.	Energy savings per biomass unit; reduction in operational footprint; reproducibility of digital processes across different aquaculture setups; increase in digitally supported yields.

Table 1: Sustainable Development Goals addressed directly or indirectly by this Dissertation

Agradecimentos

Desde o início deste ano, embarquei numa viagem de crescimento pessoal - persegui novos objetivos, iniciei projetos que nunca imaginei iniciar, adotei hábitos que antes pareciam fora do meu alcance e esforcei-me por me tornar a melhor versão de mim. Não tem sido fácil. Não me refiro aos hábitos nem aos projetos, mas a manter-me mentalmente forte com tudo o que se tem passado.

O stress constante e a incapacidade de controlar todos os pormenores da minha vida estavam a desgastar-me. Mas optei por enfrentar esses desafios de cabeça erguida, redefinir prioridades e provar que podia fazer melhor. Essa escolha foi um tiro no escuro. Parei tudo - os projetos, os hábitos, os progressos que tinha feito desde janeiro - sem saber se seria bem sucedido. Mais uma vez, consegui alcançar os meus objetivos e também ultrapassá-los. Não posso celebrar esta conquista sem agradecer às pessoas que estiveram ao meu lado ao longo desta jornada.

Ao meu irmão. A pessoa mais importante da minha vida. Foi por tua causa que escolhi este caminho. Ensinaste-me, apoiaste-me, riste e choraste comigo e fizeste sacrifícios por mim. Mesmo longe, fazes a diferença em cada dia da minha vida.

Para a minha linda namorada, a minha fonte diária de motivação. Tens-me feito seguir em frente, inspirado a sonhar mais alto e a alcançar o que antes parecia inalcançável. Este ano também tem sido difícil para ti... És a razão dos meus sorrisos, da minha dedicação e da minha falta de paciência (estou a brincar!). O tempo que passamos juntos traz paz ao caos do meu dia a dia.

Aos meus pais. Pai e mãe, à vossa maneira, influenciaram profundamente a minha vida. O homem que sou hoje é um grande reflexo do que me ensinaram, da educação que me proporcionaram e da força que demonstraram nos momentos mais difíceis. Tenho orgulho em ser vosso filho e estou eternamente grato pelo amor e apoio que sempre me demonstraram.

Ao meu melhor amigo! A pessoa com quem mais me identifico. Começaste a tua própria jornada neste mesmo ano e, juntos, conseguimos tanto. As nossas conversas diárias e a força inabalável da nossa amizade trazem alegria e significado à minha vida.

Aos meus supervisores e colegas de trabalho do INESC TEC. A minha primeira experiência profissional foi muito além do que eu imaginava. Nunca pensei que encontraria tanta satisfação no trabalho. Deram-me oportunidades, conhecimentos e experiências que levarei para a frente com orgulho. Este ambiente de trabalho é a prova viva de que o trabalho árduo e a alegria podem - e devem - coexistir no mesmo ambiente.

Finalmente, à minha família alargada e aos meus amigos. Cada um de vós desempenhou um papel importante na minha vida. Espero que saibam que, de uma forma ou de outra, em algum momento, estiveram lá para mim e ajudaram a moldar a pessoa em que me tornei. Sem vós, nada disto teria sido possível.

Esta dissertação foi desenvolvida no âmbito do projeto INNOAQUA, financiado pelo programa de investigação e inovação Horizonte Europa da União Europeia (Grant Agreement No. 101084383). Os conhecimentos, o feedback e as colaborações no âmbito deste projeto constituíram uma base essencial para os desafios explorados e as soluções propostas nesta dissertação.

Acknowledgements

Since the beginning of this year, I've embarked on a journey of self-improvement—pursuing new goals, initiating projects I never imagined, adopting habits that once felt out of reach, and striving to become the best version of myself. It has not been easy. Not the habits nor the projects, but remaining mentally strong with everything that has been going on.

The constant stress and the inability to control every detail of my life were wearing me down. But I chose to face those challenges head-on, reprioritize, and prove to myself that I could do better. That choice was a leap into the unknown. I paused everything—the projects, the habits, the progress I had made since January—without knowing whether I would succeed. Once again, I managed not only to reach my goals but to surpass them. I cannot celebrate this achievement without expressing gratitude to the people who stood by me throughout this journey.

To my brother. The most important person in my life. You are the reason I chose this path. You've taught me, supported me, laughed and cried with me, and made sacrifices for me. Even from afar, you make the difference in each day of my life.

To my beautiful girlfriend, my daily source of motivation. You've kept me moving forward, inspired me to dream bigger, and reach for what once seemed unreachable. This year has been difficult for you too… You are the reason behind my smiles, my dedication, and some of my lack of patience (I'm kidding!). The time we spend together brings peace to the chaos of my days.

To my parents. Mom and Dad, in your own distinct ways, you've profoundly influenced my life. The man that I am today has a big reflection of what you've taught me, the education that you've provided me, and the strength that you demonstrated during your most difficult times. I am proud to be your son and forever grateful for your love and support.

To my best friend! The person that I most relate to. You began your own journey this same year, and together we've accomplished so much. Our daily conversations and the unshakable strength of our friendship bring joy and meaning to my life.

To my supervisors and colleagues from INESC TEC. My first professional experience has been beyond anything I imagined. I never thought I would find so much fulfillment in work. You've given me opportunities, knowledge, and experiences I'll carry forward with pride. This work environment is living proof that hard work and joy can—and should—coexist.

Finally, to my extended family and friends. Each of you has played a role in my life. I hope you know that, in one way or another, at some point, you were there for me—and helped shape the person I've become. Without you, none of this would have been possible.

Furthermore, this dissertation was developed in the scope of the INNOAQUA project, funded by the European Union's Horizon Europe research and innovation programme (Grant Agreement No. 101084383). The insights, feedback, and collaborations within this project provided an essential foundation for the challenges explored and the solutions proposed in this dissertation.

André Correia da Costa

“Nothing in life is worthwhile unless you take risks.
Fall forward. Every failed experiment is one step closer to success.”

Denzel Washington

Contents

List of Figures

List of Tables

Listings

List of Acronyms

AAS	Asset Administration Shell
AI	Artificial Inteligence
AML	AutomationML
API	Application Programming Interface
BoM	Bill of Material
CD	Concept Description
CPPS	Cyber-Physical Production Systems
CPS	Cyber-Physical Systems
CPU	Central Processing Unit
DT	Digital Twin
GIoT	Green Internet of Things
IEC	International Electrotechnical Commission
IDTA	Industrial Digital Twin Association
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
ISA	International Society of Automation
IT	Information Technology
I4.0	Industry 4.0
MES	Manufacturing Execution System
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
OPC UA	Open Platform Communications Unified Architecture
OT	Operational Technology
PdM	Predictive Maintenance
PLC	Programmable Logic Controller
RAMI 4.0	Reference Architectural Model Industrie 4.0
RBAC	Role Access Based Control
REST	Representation State Transfer
RQs	Research Questions
SDGs	Sustainable Development Goals
TCP	Transmission Control Protocol
UI	User Interface
VM	Virtual Machine

Chapter 1

Introduction

This chapter introduces the foundation and motivation for exploring emerging digital technologies aimed at improving industrial processes and systems integration. It sets the stage for the research problem, key questions, and hypothesis to further support the reason and guidance of this study.

1.1 Context and Motivation

The evolution of industrial paradigms through the fourth and fifth industrial revolutions has fueled a shift toward smart manufacturing, asset digitization, and human-machine collaboration. These advancements have broadened the scope for innovation across production sectors.

As part of this transformation, organizations are increasingly investing in digitalization to enhance operational efficiency, data-driven decision-making, and environmental sustainability [37]. Cyber-Physical Production System (CPPS) and Industrial Internet of Things (IIoT) solutions have emerged as key enablers of this shift, enabling tighter integration between Operational Technologies (OT) and enterprise-level systems.

Despite widespread adoption across traditional manufacturing domains, several industrial sectors remain underexplored in terms of digital integration. Algae cultivation is one such domain, where production environments are often managed manually and lack the benefits of scalable, automated, and interconnected systems [7]. Although algae, both micro- and macroalgae, hold high potential across sectors such as bioenergy, food, and environmental applications, the production processes remain fragmented and technologically underdeveloped in many regions, especially Europe [18, 64]. As a result, this sector presents a compelling use case for demonstrating how modular and interoperable digital solutions can be applied in complex, real-world environments.

1.2 Problem Statement

The rapid development of IIoT and CPPS has resulted in a growing body of literature proposing reference architectures, communication frameworks, and conceptual models. Many of these contributions offer valuable insights into how modern industrial systems might evolve, highlighting principles such as modularity, interoperability, and digitalization.

However, despite this theoretical maturity, the path to practical implementation often remains fragmented and domain-specific. Studies tend to focus on isolated components—such as protocol adapters, data models, or digital representations—without offering unified, validated workflows for deploying these technologies in operational environments. This is particularly evident when transitioning from controlled lab scenarios to real-world, multi-layered industrial systems involving legacy infrastructure, partner constraints, or organizational silos.

In such settings, challenges arise not only from the technical integration itself but from the lack of composable and reproducible blueprints that support scalable deployment. Even though different studies may complement each other, the absence of consolidated, practical guidance hinders replication and slows innovation in various sectors. This gap applies broadly to any context where innovation must be introduced into legacy systems or collaborative environments. To foster true digital transformation, there is a need for unfolded, comparative, and validated case studies that move beyond conceptual frameworks—offering practical guidance on how to integrate state-of-the-art industrial technologies in complex real-world scenarios.

1.3 Research Questions

Building upon the identified gap in the problem statement, this dissertation is guided by a set of research questions that shape the study's direction and define its scope.

RQ1: What are the key requirements and constraints when implementing/integrating digital-industrial solutions into complex legacy systems in real-world environments?

RQ2: Which state-of-the-art technologies offer scalable, modular, and interoperable solutions for such integration?

RQ3: How can these technologies be effectively implemented, validated, and evaluated in a practical use case to ensure robustness, usability, and long-term adaptability?

Together, these questions provide a structured framework for investigating the practical integration of advanced industrial technologies in real-world scenarios. They guide the exploration from understanding system-level requirements to identifying suitable technologies and, ultimately, to validating their implementation within a representative use case.

1.4 Hypothesis

This research hypothesizes that the development and integration of modular, scalable, and interoperable industrial solutions—grounded in CPPS, IIoT architectures, and state-of-the-art frameworks such as the Asset Administration Shell (AAS)—can effectively enhance digital integration within complex environments. These solutions, when designed with compliance to industrial standards and layered architectures, will demonstrate high compatibility with real-world legacy systems.

The **hypothesis** assumes that:

- An interoperable and standard implementation of Digital Twins (DTs) using AAS will enable effective integration across system layers, facilitating adaptable data processing.
- Leveraging interoperable technologies and communication protocols will ease integration with legacy infrastructures, reduce overhead, and improve maintainability and scalability.

These assumptions will be tested through the qualitative evaluation of two implementation strategies, validated in a real-world use case within an algae cultivation setting. Emphasis will be placed on architectural soundness, interoperability, and the clarity of implementation—addressing the current gap in practical and reproducible digital-industrial integrations.

1.5 Document Structure

This dissertation is structured to support the research focus on modular, interoperable, and scalable digital-industrial implementations. The document follows a logical progression from problem identification to applied experimentation and final reflections.

Chapter 1 introduces the research, presenting the motivation and background, defining the problem statement, and formulating the research questions that guide the investigation. It also outlines the intended contributions and overall structure of the work.

Chapter 2 reviews foundational concepts and technologies relevant to the study, including IIoT, CPPS, and DTs. Special attention is given to frameworks and technologies that support interoperability, such as Asset Administration Shells, alongside studies highlighting existing gaps in real-world implementation, further exploring digital approaches for algae cultivation.

Chapter 3 presents the adopted methodology, structured into exploratory research, technology experimentation, and validation phases. It defines the approach to requirement elicitation, technology assessment, and practical recognition in real-world conditions. It introduces the case study—an algae cultivation environment—describing its infrastructure, legacy constraints, and the multi-organizational context in which it operates. This chapter highlights the practical challenges that motivate the need for robust, adaptable digital solutions.

Chapter 4 details the proposed digital solutions, including their architectural design, integration strategy, and evaluation in terms of interoperability, scalability, and applicability within the case study, by comparing their strengths and limitations within the operational context.

Chapter 5 focuses on the digitization of the algae cultivation environment through the structured use of Asset Administration Shells. It explores the construction of digital twins for physical and software assets, the application of standardized Submodel templates, and the enrichment of asset semantics for machine-readability and long-term lifecycle management.

Finally, Chapter 6 concludes the dissertation, summarizing its contributions and presenting recommendations for future research and technological development in industrial digitalization.

Chapter 2

Literature Review

This chapter consolidates and critically examines existing literature to establish a foundation for the research. It begins by outlining the methodology used for identifying and selecting relevant sources that address the research questions defined in Section 1.3. The review is structured to prioritize studies related to industrial digitalization, including key enabling technologies, digital innovation, and a small understanding of domain-specific implementations in algae cultivation to contextualize the use case. This structured synthesis provides both theoretical and practical insights that inform the architectural decisions and experimental design of the proposed solutions.

2.1 Search Strategy and Analysis

The literature collection process for this study was designed to understand the state of digital innovation in multiple solutions and provide contextual relevance for the use case adopted in this research. The following sections outline the search strategy and the rationale behind it. Each of the main topics covered in the review is introduced with its relevance to the RQs. This is followed by a description of the document analysis process, including filtering and screening steps. The resulting body of literature serves as the foundation for the subsequent review and analysis.

2.1.1 Development of Search Query

The search strategy was developed to explore the technological foundations and frameworks essential for industrial digitalization, particularly those that gained momentum during the emergence of Industry 4.0 (I4.0). The selected documents from each query aimed to capture both the theoretical underpinnings and the practical applications of these technologies, ensuring a comprehensive review aligned with the research questions.

Figure 2.1 illustrates the key technological terms explored in the first two search queries, grouped by color. The first query focused on three core enablers of industrial transformation: **CPS/CPPS**, **IoT/IIoT**, and **DTs**. These technologies form the conceptual and operational backbone of many modern production systems and are heavily referenced in current literature due to their widespread application and relevance.

The second query was specifically designed to investigate the **Asset Administration Shell**, a standardized implementation of DTs rooted in the Industry 4.0 paradigm [33]. AAS was included due to its strong alignment with the research objectives—particularly its focus on modularity, semantic interoperability, and standardization. In addition to academic sources, original technical documentation and reference materials from the Platform Industrie 4.0 initiative were consulted to ensure a reliable and up-to-date understanding of AAS concepts.

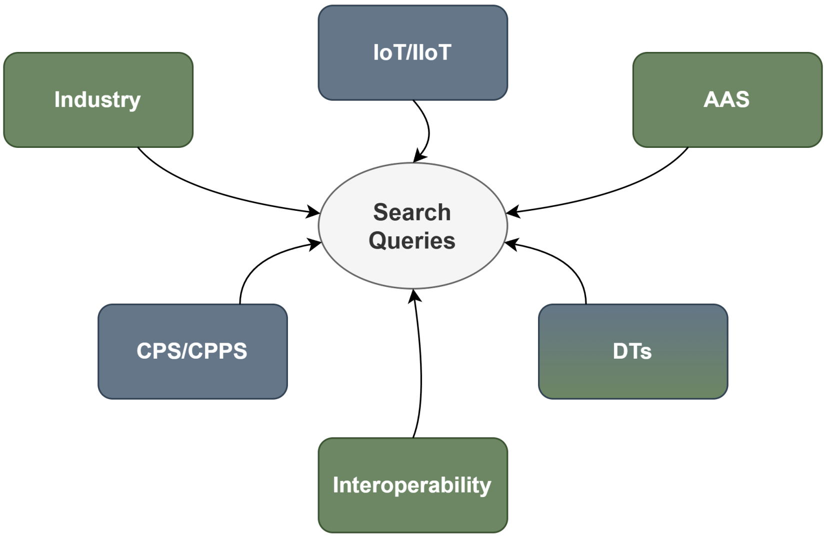

Figure 2.1: Main technological Terms included in the initial Search Queries

A third query was conducted, targeting algae-related studies that intersect with the aforementioned technologies. This final search ensured a better understanding of the current state of digital innovation in algae cultivation, highlighting the extent to which these solutions are being adopted or proposed in this emerging sector. The structure of this combined query is shown in Figure 2.2.

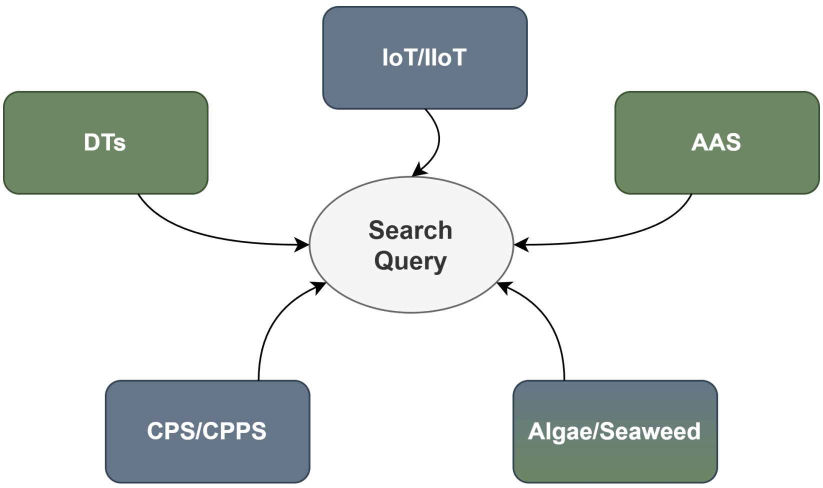

Figure 2.2: Search Query Technological and Algae related

The full search was executed on the *Scopus* database due to its extensive and multidisciplinary coverage of peer-reviewed literature. Each query produced an initial set of documents, which were then narrowed down through a preliminary relevance assessment based on title, abstract, and citation count. The final selection aimed to balance theoretical significance with practical applicability, covering both industry-standard frameworks and emerging use cases.

Every query was designed to contribute to a specific dimension of the investigation. The first two queries support RQ1 and RQ2 by identifying the technological enablers and standardization frameworks that define the integration landscape, including their capabilities, limitations, and industrial relevance. The third query provides context for evaluating how these technologies have been—or could be—applied and validated in real-world use cases, contributing in part to RQ3.

2.1.2 Organization and Filtering Process

The retrieved documents were organized and managed using *Zotero*, a reference management tool selected for its ability to streamline evaluation, annotation, and citation management. To ensure the relevance and quality of the selected studies, a multi-phase filtering process was employed:

- **Initial Screening:** Documents were initially reviewed based on their titles, abstracts, and keywords. This step aimed to exclude studies that were either irrelevant or tangentially related to the research objectives.

- **Full Text Accessibility:** After the initial screening, attempts were made to access the full text of the selected and remaining studies. Articles and book chapters that were unavailable due to access restrictions or other limitations were excluded from further analysis.

- **Full-Text Review:** The final phase involved a full-text reading and analysis of each document. This analysis focused on evaluating the relevance of the study's theme and objectives, ensuring that the selected studies contributed directly to the research and hypotheses.

The analysis screening process culminated in a set of studies. This collection comprised documents derived from the initial and technology-focused searches, original documentation, and more concentrated studies on algae cultivation. Together, this set of documents serves as a reference for future research and forms a comprehensive foundation for the review process chapter.

2.2 Foundational Technologies

This section introduces the exploration and assessment of the knowledge presented in the documents retrieved from the initial search, expressed in Figure 2.1. Given all the information available, it details the core industrial terms most relevant for this research. The studies provide explanations of each key concept and their role as enablers of innovation in ecosystems.

These technological concepts are often characterized by these studies as Industry 4.0 key enablers [68]. Besides their different definitions and purposes, in industrial environments they act as complementary technologies to deal with more complex systems [60]. It is important to understand that the concepts presented in the following subsections pursue common goals to improve, automate, and understand diverse ecosystems.

2.2.1 Internet of Things and Cyber-Physical Systems

The Internet of Things and Cyber-Physical Systems are cornerstone technological terms driving the digital transformation in industrial settings. These technologies represent the connection between the physical and digital worlds, forming the basis for real-time data collection, environment monitoring, engagement, and analysis in complex systems [45, 68]. Their integration transforms traditional practices and industrial processes into dynamic, adaptive, and **smart** systems [39].

IoT, a foundational topic of I4.0, represents a network of interconnected physical devices capable of collecting and transmitting data over the Internet [40]. Its application in manufacturing processes has evolved into the Industrial IoT, aiming to enhance adaptability and efficiency within production systems [68]. Its functionality is closely tied to Big Data, as it generates vast volumes of information for analysis and optimization, and its capabilities are further amplified when integrated with other Information Technologies (IT) [41, 60].

CPS, another main pillar of I4.0, enables the integration and interaction of cyber and physical components within open-loop or embedded systems, depending on the application requirements [39, 45]. In industrial contexts, this concept has evolved into CPPS, which focus on automating production environments by leveraging real-time data acquisition and system control. These systems facilitate seamless communication between physical assets and computational processes, driving efficiency and adaptability in manufacturing operations [54].

The reviewed studies offer a robust foundation of knowledge of these key enablers, highlighting their relevance and potential to address the research questions outlined in Section 1.3. This body of knowledge and further sections highlight the critical role of IoT and CPS in advancing the technological framework necessary for this study, particularly when combined with complementary technologies to enhance industrial applications.

2.2.2 Digital Twins

DTs enhance IoT and CPS capabilities by integrating with other I4.0 technologies [40]. At their core, DTs are virtual counterparts of physical entities, acting as highly accurate digital replicas that mirror the state and behavior of their physical part in real time [45, 55]. This capability enables DTs to act as a bridge between the physical and digital domains, such as CPS [59], fostering advanced monitoring, analysis, and control throughout the lifecycle of systems or products [54].

A DT surpasses the concept of a Digital Shadow, a digital representation of a physical entity without virtual control over the asset, by enabling bidirectional communication between the two entities. This allows DTs not only to simulate or analyze operations but also to control the physical system, making them dynamic for managing and optimizing operations [8].

One of the most relevant studies, frequently citepd in the retrieved literature, defines a digital twin as a mirror of its physical twin in real-time, continuously adapting to changes in the operational environment, accurately replicating and simulating the behavior of the physical system [59]. Additionally, their study explores the correlation between DT and CPS concepts, both of which are fundamentally rooted in cyber-physical integration. It examines the origins and evolution of these concepts, highlighting both their similarities and differences. While both share a foundation in bridging physical and cyber domains, CPS typically operates in a one-to-many configuration, where a single computational system manages multiple physical components. In contrast, DTs are distinguished by their one-to-one mapping, offering precise and high-fidelity virtual representations of specific physical entities or processes.

DTs are also distinguished by their lifecycle perspective, which spans the entire lifecycle of a system or a product [58]. Key functionalities derived from the reviewed studies can be grouped into four primary roles:

- **Monitoring and Visualization:** DTs enable continuous, remote, and real-time observation of physical systems, offering insights into operations through graphical interfaces and dashboards [54, 58].
- **Simulation and Prediction:** Using advanced computational models, DTs can simulate future behaviors, evaluate potential scenarios, and predict failures, thereby reducing risks and operational downtime [54].
- **Optimization and Decision Support:** DTs leverage real-time data to optimize processes, validate and refine operational strategies, and support informed decision-making through actionable insights [54, 55].
- **Lifecycle Management:** DTs document and manage the asset lifecycle, providing traceability, performance records, and communication between stakeholders to improve coordination and understanding [8, 58].

The adoption of DTs is not uniform across industries. Their roles and implementations depend on the specific requirements of each sector. Zeb et al. [68] emphasize the symbiotic relationship between DTs, IIoT, and CPS, where DTs leverage the connectivity of IoT devices and the computational infrastructure of CPS to create synchronized virtual models of physical systems.

The conceptual model of DTs is aligned with recent developments in standardization efforts, particularly within the framework of Industry 4.0. A key evolution in this domain is the introduction of the Asset Administration Shell, a standardized digital representation of an asset that formalizes the principles of DTs while ensuring interoperability across industrial systems [32]. This concept is grounded in the Reference Architecture Model for Industry 4.0 (RAMI 4.0), which provides the structural and semantic foundation for implementing AAS in real-world systems.

2.2.3 RAMI 4.0

The RAMI 4.0 is a three-dimensional framework developed by Plattform Industrie 4.0 to harmonize the various aspects of industrial digitalization. It combines elements of product lifecycle, enterprise hierarchy, and IT architecture to form a unified model that enables consistent digital transformation in manufacturing [34].

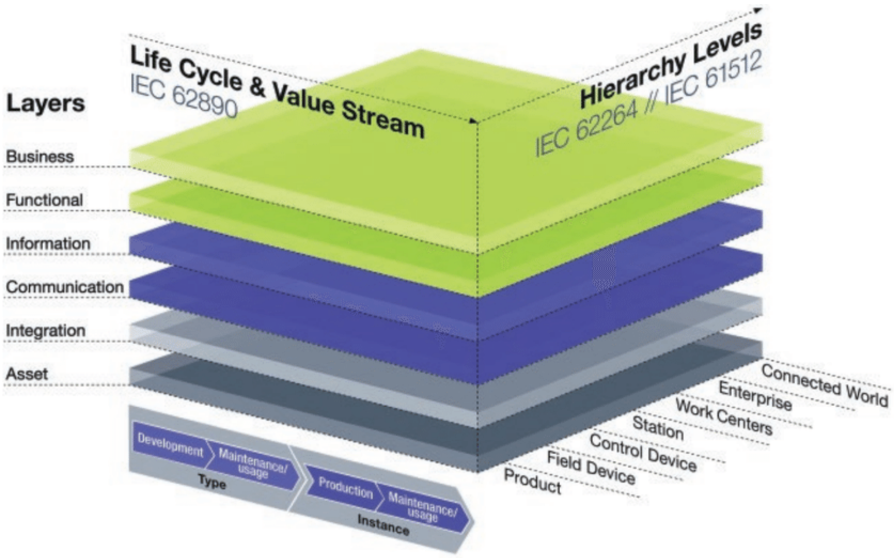

Figure 2.3: Reference Architecture Model for the Industry 4.0 (Source: [35])

Product Lifecycle This axis captures the asset's evolution from design to decommissioning, its end-of-life [34]. DTs and AAS are instrumental in supporting this lifecycle perspective, enabling continuous monitoring, reconfiguration, and data access throughout the asset's operational life. AAS instances are typically deployed and updated across the lifecycle stages, ensuring that all relevant data remains synchronized and accessible.

Hierarchy Levels It derives from the automation pyramid (as defined in International Electrotechnical Commission (IEC) 62264/International Society of Automation (ISA)-95 [30]), extending from field devices up to enterprise-level systems. RAMI 4.0 adapts this model to modern, decentralized environments by introducing the Connected World level at the top. AAS can be instantiated at various levels of this hierarchy—whether embedded in a field device or used as a digital service interface at higher levels [24].

Layers The third axis defines seven layers, from the physical asset up to the business layer:

- **Asset:** Represents the physical or logical object.
- **Integration:** Links the asset to its digital representation.
- **Communication:** Defines protocols and data exchange mechanisms.

- **Information:** Structures the data for semantic interpretation.
- **Functional:** Describes services and functions available to the system.
- **Business:** Embeds the logic for process rules, goals, and value generation.

In real-world manufacturing environments, asset diversity is significant—ranging from hardware (e.g., machines, sensors) and software (e.g., services, control logic) to production data and communication protocols. The RAMI 4.0 model accommodates this heterogeneity and provides a reference for building autonomous, standardized shop floor solutions [9]. Central to this approach is the Asset Administration Shell, which plays a key role in managing the heterogeneous systems present within industrial environments.

2.2.4 Asset Administration Shell

AAS, as a standardized digital representation of a physical or logical asset, gives uniform access to information and functions, thereby facilitating interoperability, transparency, and modular integration among the applications of a manufacturing company and industrial environments [33].

The AAS is composed of two main parts: a **header**, which includes identification data about the AAS instance and the associated asset, and a **body**, which contains detailed information structured in Submodels (Figure 2.4). The Submodels encapsulate context-specific data and behavior, enabling modular representation of different aspects of an asset (e.g., manufacturer, technical data, service interfaces, or lifecycle data).

The AAS **metamodel** is presented as a UML class diagram, as illustrated in Figure 2.5, defining all the building blocks that must be used to internally structure the AASs of every possible asset [9]. The information is structured into **Submodels**, an entity listing specific attributes, and Submodel Elements, of an asset. Being identifiable, it is usually referred to by an AAS or multiple AASs. Each Submodel is composed of **Submodel Elements**, which can include properties, functions, events, files, and collections. These elements are typed and may vary depending on the complexity of the asset being represented.

To ensure semantic interoperability, each Submodel or Submodel Element may include a *semanticId*, referencing an external semantic repository such as IEC 61360, eCl@ss, or custom ontologies [9, 12, 29]. This alignment with standardized vocabularies allows AAS-enabled systems to interpret and interact with data meaningfully and consistently across domains and platforms.

Additionally, the AAS can be categorized into three types, passive, reactive, and proactive, based on its behavior and degree of automation [67]:

1. A static data container, typically serialized in formats such as JSON, XML, RDF, or AutomationML (AML), used for cataloging and data exchange across systems or partners.
2. An interface-enabled AAS that supports API-based client-server communication. This model allows external applications to retrieve or update data via service endpoints.

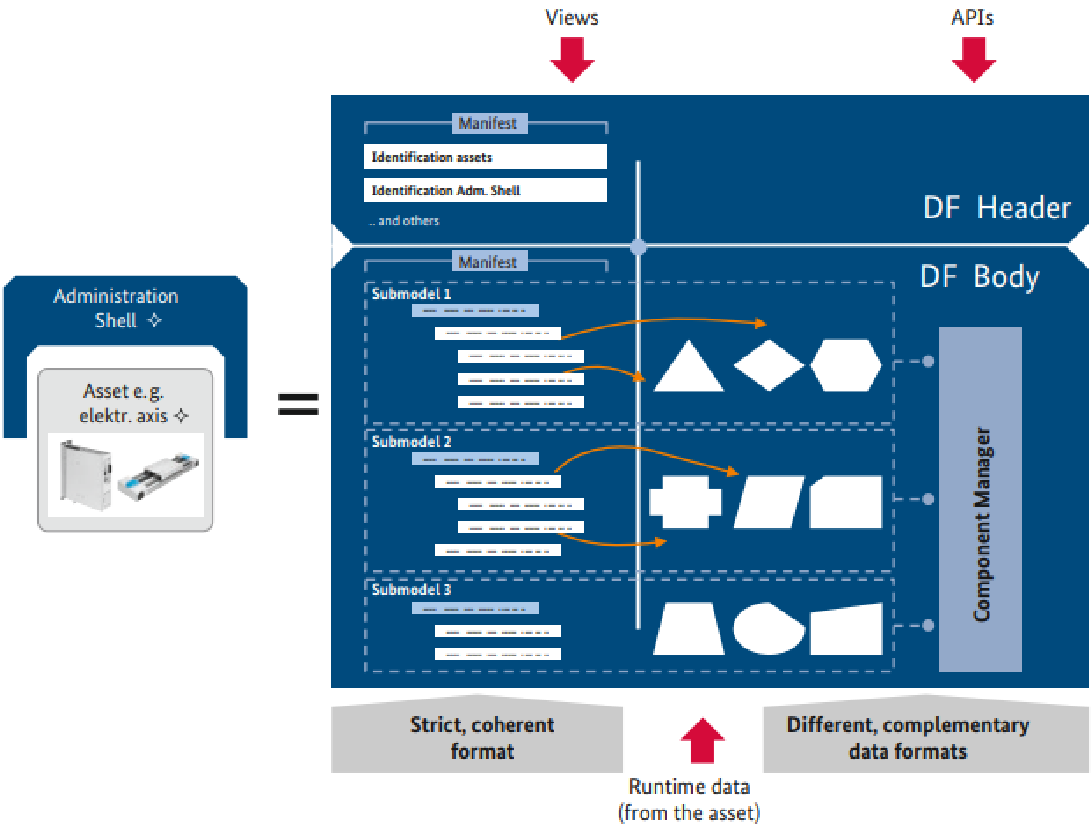

Figure 2.4: Asset Administration Shell General Structure (Source: [36])

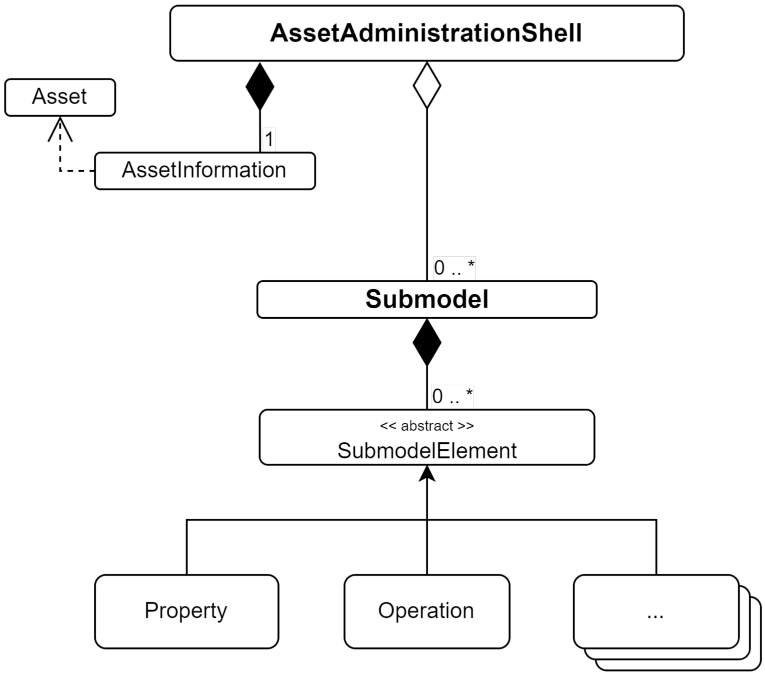

Figure 2.5: AAS Metamodel Simplified (Source: Industrie 4.0 [33] and Jacoby et al. [38])

3. The most advanced form of an AAS, capable of not only responding to queries but also performing autonomous functions such as local decision-making, behavior execution, and participation in distributed control logic with other assets' AASs.

Although passive and reactive AAS types are already supported by several open-source and commercial frameworks, the development of proactive AAS remains in development phases by such platforms, which underscores the relevance of further experimentation and validation efforts.

Together these technologies form the conceptual basis for industrial digitalization and asset digitization. They provide the necessary structure to support scalable, modular, and intelligent systems. This groundwork sets the stage for examining how such technologies are being adopted and innovated across different industrial sectors.

2.3 Digital Innovation across Industries

The collected documents examine the foundational technological enablers presented in Figure 2.1, detailing their meanings, industrial applications, and theoretical benefits as emphasized in the Section 2.2. These studies underline the interconnection and complementary nature of these technologies, highlighting their role in driving the ongoing transformative industrial ecosystems. Beyond logical definitions, this section outlines the practical potential, detailing how the advancements of the aforementioned technologies are leveraged to foster digital innovation across diverse industries, expressed by the collected set of documents. Additionally, the following content explores various approaches and architectures that integrate these key enablers with other emerging frameworks, focusing on their application within the manufacturing domain.

A summarized overview of these studies, including their technological focus and key contributions, is presented in Table 2.1, showcasing the application of these technologies across industries.

Before delving into the innovative approaches of these papers, it is important to consider the perspective provided by Kaur, Mishra, and Maheshwari [41] and Redelinghuys, Basson, and Kruger [54]. Since the outbreak of digitization, data has become a critical resource for businesses. Kaur, Mishra, and Maheshwari [41] point out that these technologies are designed to address the growing volumes of data generated daily by industrial operations. The ability to control the environment, perform analysis, and derive actionable insights from the data has become indispensable for modern systems and is enabled by the innovation provided by these technologies. Redelinghuys, Basson, and Kruger [54] emphasize the role of human-machine collaboration in creating intelligent business systems. They attribute this need to the increasing complexity brought by advancements in computational power, connectivity, and data volumes, underscoring the demand for technologies capable of effectively addressing these challenges.

2.3.1 Cross-Industry Applications of Foundational Enablers

To contextualize the focus of the following content, Zeb et al. [68] conducted a study that presents a Scopus analysis in 2021, evaluating publications on IoT, CPS, and DTs up to that year. Despite

DTs being introduced as early as 2003, their growth was initially deprived by insufficient technological advances and limited vision for their application. NASA's successful use of DTs in their space shuttle program demonstrated the concept's potential and catalyzed its broader adoption. From 2011 onward, DTs began to gain momentum with their adoption in various industries. Zeb et al. [68] further emphasize that, as of 2021, the number of publications on DTs had surpassed those of CPS and IIoT combined, reflecting the growing recognition of their transformative capabilities. This analysis is further reinforced by information presented from the studies collected, which place a notable emphasis on DTs and their implementations compared to other technologies, lending additional credibility to Zeb et al. [68] observation.

Redelinghuys, Basson, and Kruger [54] [55] introduce a six-layer architecture designed for implementing aggregated DTs. This architecture enables the integration of multiple DT instances into a unified system, where various DTs are consolidated into a single aggregated DT. Such an approach allows the simulation of complex operations involving multiple digital counterparts, enhancing the system's capacity for data collection, analysis, and decision-making. By streamlining operations and improving insight generation, this architecture supports more intelligent and efficient industrial systems. To achieve this functionality, the architecture employs OPCUA servers for secure and standardized data exchange, alongside IoT gateways facilitating information conversion between systems for each DT. These components ensure effective communication and understanding between the digital counterparts through message exchange methods. This allows DTs to monitor the system's status, analyze its performance, and send actionable insights to physical assets, enabling the system to react autonomously and intelligently.

Redelinghuys, Kruger, and Basson [55] mention the concept of "separation of concerns" within this framework. This principle involves breaking down complex industrial equipment into smaller, modular components to reduce the system complexity. By doing so, the architecture enhances flexibility and scalability, allowing individual DTs to focus on specific tasks. This modularity simplifies business-level analysis but also empowers each DT to make more informed and efficient decisions, ultimately contributing to the system's intelligence and automation capabilities.

Similar to Redelinghuys, Kruger, and Basson [55], other studies also introduce innovative architectures and data models that leverage the capabilities of IoT platforms for enhanced industrial applications. For example, Park, Woo, and Choi [52] propose a robust cloud-based DT manufacturing system underpinned by an interoperable data schema. This approach facilitates seamless integration across smart manufacturing systems, enabling real-time monitoring, virtual modeling, and process optimization. Their work emphasizes how digital technologies can enhance production flexibility and efficiency, addressing the diverse and evolving needs of modern industries.

Back to architectures, Mendonça et al. [45] explore methodologies for integrating communication standards and enabling technologies into DTs. By employing AutomationML and artificial intelligence (AI), they create advanced DT-driven solutions for predictive analytics and real-time decision-making. This framework enhances operational efficiency and facilitates predictive maintenance, underscoring the role of AI in harnessing actionable insights from industrial data.

The outcomes of deploying digital twins at edge and cloud levels have been further explored by Zeb et al. [68], who underline the importance of next-generation communication technologies. Their study illustrates how DTs can be implemented across multiple layers of communication and computing infrastructure, thereby optimizing industrial processes. This multi-level deployment strategy enables manufacturers to balance processing workloads between edge and cloud environments, improving system responsiveness and reliability.

Teng et al. [60] extend this discussion by delving into the interplay of emerging enabling technologies such as blockchain, edge computing, and cloud computing. Their analysis highlights the potential of these technologies to enhance data security, streamline resource allocation, and support industrial energy efficiency initiatives. They argue that such advancements are integral to building resilient and sustainable manufacturing ecosystems.

Furthermore, Javaid et al. [39] provide a comprehensive view of how CPS, IoT, and DTs collectively revolutionize industrial operations. Their tabular representation of over 30 applications details diverse advancements achieved through these technologies, including enhanced control, monitoring, and integration. Their work serves as a high-level theoretical resource, offering a stable foundation for understanding key concepts and correlations that drive digital innovation.

Despite these achievements, a critical observation persists across the literature: many studies fall short in sharing practical implementation details. As Kamath, Morgan, and Ali [40] and Teng et al. [60] argue, much of the research prioritizes theoretical modeling over actionable development insights, which poses a bottleneck for academic and industrial collaboration. This concern, previously discussed in the Problem Statement's Section 1.2, underscores the complexity of addressing such gaps, which require coordinated efforts and contributions from multiple industry sectors. Bridging this gap by fostering open-source platforms and sharing detailed implementation frameworks could accelerate the adoption and innovation of Industry 4.0 technologies.

2.3.2 Standardization and Interoperability in Digital Transformation

Regarding the standardization of manufacturing systems with heterogeneous sources of data/information, the Asset Administration Shell has been an element of study of robust but effective integration in complex industrial settings. The following studies illustrate how AAS can be leveraged to enhance automation, ensure system compatibility, and promote seamless data exchange across different levels of industrial environments.

One stream of research focuses on the interoperability—one of the foundational pillars of the AAS concept—especially with legacy systems and standardized communication models. Among the technologies associated with this goal, the **Open Platform Communications Unified Architecture (OPC UA)** is widely recognized as the most suitable communication protocol for Industry 4.0 environments [50]. Within the RAMI 4.0 model, OPC UA is positioned in the Communication Layer, where it facilitates secure, platform-independent, and service-oriented data exchange [35]. Defined by the **IEC 62541** standard [31], OPC UA supports information modeling, which makes it compatible with AAS's structured data model.

For instance, Cavalieri and Salafia [9] support OPC UA as a central enabler of I4.0-compliant architectures. In their study, AAS is implemented in a **Predictive Maintenance (PdM)** scenario through the creation of dedicated Submodels that monitor asset conditions, highlighting how domain-specific extensions can support proactive strategies, leveraging OPC UA communication protocol for data aquisiction. Another study proposes a transformation pipeline that converts AAS instances (`.aasx` files) into XML-based OPC UA structures. This process enables the integration and **aggregation of AAS** data across platforms and partners, fostering interoperability in distributed manufacturing environments [67].

The synergy between AAS and OPC UA has also been explored in control systems. A recent study by Cavalieri and Salafia [10] examine the communication between AAS and **Programmable Logic Controllers (PLCs)** for real-time actuation. PLCs are industrial digital computers governed by the **IEC 61131-3** standard [28], designed for the control of electromechanical processes. In this work, the internal logic and I/O structure of PLCs are mapped into AAS Submodels, with OPC UA enabling runtime bidirectional communication. This integration allows AAS-enabled systems to both monitor and control physical processes within a standardized digital twin framework.

Another line of research addresses the use of AAS in managing production environments and **Manufacturing Execution Systems (MES)**. MES platforms coordinate and monitor production workflows in real time, bridging the gap between enterprise-level planning and shop floor operations. Quadrini et al. [53] present a work about AAS's Submodel creation for reproducing an industrial laboratory environment. It leverages the technology PyI40AAS, a Python-based toolkit for modeling and instantiating AASs. Their implementation includes seven AAS types representing stations, products, work orders, and assembly lines. These components are organized hierarchically and interconnected to reflect the logic and dependencies typical of real manufacturing systems, effectively simulating a modular and interoperable MES.

Within production systems, flexible automation is increasingly supported by **capability-based engineering** [24]. Capabilities are treated as functional descriptions of what an asset can perform—such as a robotic arm's lifting strength or grip type—and are formalized within AAS Submodels. These capabilities are formalized within AAS Submodels, enabling intelligent asset discovery and autonomous orchestration. The study presents AAS-based templates that support this capability-driven approach, enabling proactive AAS behavior where assets dynamically self-organize based on operational needs.

Complementing these efforts, the **standardization of Submodels** has also received significant attention. several studies contribute to the standardization landscape by cataloging reusable Submodel templates and summarizing current practices. Abdel-Aty, Negri, and Galparoli [1] provide a comprehensive survey of reusable AAS Submodels drawn from both academic literature and predefined templates by Plattform Industrie 4.0 [33]. This work offers a valuable reference for developers and researchers aiming to adopt or customize AAS implementations, bridging theoretical design with practical deployment.

Collectively, these studies highlight the versatility of the Asset Administration Shell across a range of industrial applications. Whether enabling interoperability via communication standards, modeling asset capabilities, or facilitating system design, AAS provides a powerful framework for advancing standardization and practical digital transformation in manufacturing ecosystems.

Table 2.1: Summary of Key Works on Digital Technologies Across Industries

Technology Focus	**Key Contributions**	**Studies**
DTs, IoT, and OPC UA	Six-layer DT aggregation model with OPC UA and IoT gateways. Proposed a modular DT framework for improved data exchange, automation, and system intelligence.	[54, 55]
DTs and Cloud Computing	Cloud-based DT system with interoperable data schema. Real-time monitoring and virtual modeling for smart manufacturing.	[52]
DTs, AI, and AutomationML	DT-driven predictive analytics model. Enhanced operational efficiency and predictive maintenance.	[45]
DTs and Edge/Cloud Computing	Multi-layer DT deployment model (edge/cloud). Improved workload distribution between edge and cloud for optimized processing.	[68]
Blockchain and Edge/-Cloud Computing	Hybrid approach integrating emerging technologies into DTs. Enhanced security, resource allocation, and industrial energy efficiency.	[60]
AAS, Submodel Templates, and OPC UA	Implementation of AAS in Predictive Maintenance (PdM) using domain-specific Submodels. Leverages OPC UA as a standard communication protocol for Data Aquisition.	[9]
AAS and OPC UA	Pipeline for converting AAS (`.aasx`) files to XML and aggregating data via an OPC UA Server for cross-platform interoperability.	[67]
AAS, Submodel Templates, PLC, and OPC UA	Mapping of IEC 61131-3 PLC structures into AAS Submodels. Real-time actuation and control through OPC UA integration.	[10]
AAS, Submodel Templates, and MES	Simulates MES with seven interconnected AAS types for stations, products, and workflows in an industrial lab. Monitors and controls production execution through AAS and Submodel Types creation.	[53]
AAS, Submodel Templates, and Capabilities	Formalization of asset capabilities within Submodels. Enables dynamic discovery, orchestration, and proactive system behavior.	[24]
AAS and Submodel Templates	Survey of existing Submodel templates from literature and Plattform Industrie 4.0. Supports standardization and implementation guidance.	[1]

2.3.3 Technological Approaches in Algae-based Systems

In this section, algae studies are presented to provide a better understanding of the current impact of these preceding technologies on the use case and the object of study of the major project in which this dissertation is included. The use case is algae cultivation. Algae, ranging from single-celled organisms (microalgal) to large seaweed (macroalgal), offer a wide range of possibilities in various sectors [18]. Despite their similarities, microalgae often receive more attention due to their smaller size, rapid growth rate, and ease of mass production, making them a more accessible organism for research and industrial applications. The following studies include both organisms and explore how the mentioned key enablers may be leveraged in algae production.

Recent studies, such as El-Moustaqim et al. [46] and Lim et al. [44], highlight that the integration of high technology, autonomous processes, and intelligence into algae farming remains in its infancy. Significant gaps persist in the adoption of these innovations, despite their potential to revolutionize the industry. The complexity of algae cultivation, combined with the need for constant and real-time monitoring, underscores the importance of such integrations. These technologies facilitate rapid adjustments to maintain ideal growth conditions, ensuring sustainability and productivity [66]. Among the retrieved studies, much of the focus is on the integration of IoT technologies and Machine Learning (ML) algorithms to achieve these goals.

IoT, as a cornerstone of this research, is highly mentioned to ensure environmental monitoring, a critical requirement for preserving the optimal parameters necessary for algae cultivation [57]. Numerous studies propose conceptual ideas for integrating IoT technologies through connected sensor networks, emphasizing the potential of IoT to manage the environmental complexities of both micro- and macroalgae production. These studies highlight the key environmental parameters that must be monitored and controlled to ensure successful cultivation.

A study conducted by Nwoba et al. [49] serve as a vital element for understanding the variables and measurements essential for cultivating algae, particularly microalgae. This research categorizes the necessary parameters into three main groups: physical variables (e.g., light, temperature, and flow rate), chemical parameters (e.g., pH and nutrient concentrations), and biological parameters (e.g., cell density and growth rates). Nwoba et al. provide a detailed explanation of the priority levels for controlling these variables, along with specific optimal values, making this study a valuable resource for future research. Additionally, the study offers critical insights into how IoT sensor networks can be effectively integrated to monitor and manage these parameters, providing a practical foundation for the development of advanced cultivation systems.

Machine Learning, another cornerstone technological concept in algae cultivation. Many studies emphasize that the combination of IoT technologies with ML algorithms is crucial for advancing algae farming practices [46, 61]. These studies highlight the critical role of ML algorithms in predicting and optimizing algae growth by analyzing raw data and identifying patterns that promote optimal parameters and conditions for cultivation [66]. The integration of ML into IoT systems allows for data-driven insights, enabling smarter decision-making for increasing production, and more precise control over the growth environment.

Within the retrieved documents, studies by Yang et al. [66] and Oruganti et al. [51] stand out for their thorough review of ML algorithms applicable to algae cultivation. These works provide a detailed breakdown of various ML algorithms, outlining their objectives, functionalities, and potential applications in the field. By presenting the strengths and use cases of each algorithm, these studies equip readers with the knowledge necessary to select the most appropriate methods for their specific requirements. This detailed understanding serves as a useful contribution for designing systems that effectively integrate ML to optimize algae production and sustainability.

Some studies extend beyond conceptualization by presenting simple architectures to validate the potential of IoT and ML integrations in algae cultivation. For instance, Munoz et al. [47], Samuel and Favitri [57], and Le-Thi-Thu, Nguyen-Van-Duong, and Doan-Van [61] showcase proof-of-concept implementations in algae cultivation environments. These studies utilize a limited set of physical devices and straightforward architectures to collect, analyze, and visualize data, effectively demonstrating the feasibility of integrating IoT technologies in algae farming. The results provide useful insights for future research and further exploration of these technologies.

A particularly significant contribution comes from Wang et al. [65], which offer an in-depth analysis of IoT applications in microalgae cultivation. This study provides comprehensive theoretical knowledge, covering conceptual IoT applications and their role in optimizing algae production. Although it touches on ML as an auxiliary tool, the primary focus remains on IoT's potential to monitor and enhance the cultivation process. The information presented makes this study a cornerstone reference for understanding the integration of IoT in algae farming.

The number of studies focusing on both macroalgae and microalgae has been continuously increasing and is expected to grow further in the coming years [18]. However, a targeted literature search focusing on DTs and AAS within algae-related contexts yielded very limited results. This absence stands in contrast to other industrial sectors, where AAS has shown clear potential for enabling modular, interoperable, and automated systems. The lack of such frameworks in algae cultivation highlights a promising direction for future research, especially in designing scalable digital solutions that can support lifecycle management, real-time monitoring, and adaptive control in complex biological environments.

2.4 Summary

Chapter 2 provided a comprehensive review of the existing literature relevant to industrial digitalization, focusing on key enabling technologies such as IoT, CPS, DTs, and the AAS. The review began with a structured overview of the literature search and analysis methodology, ensuring alignment with the research questions guiding this dissertation.

The foundational technologies were explored in depth, outlining their definitions, functional roles, and interdependencies within modern manufacturing systems. Subsequently, they were examined at an implementation level, architectural design, and application sectors, providing a clear view of the current knowledge presented across the Literature.

Finally, the chapter turned to the algae cultivation sector to assess how these digital innovations are—or are not—being applied. While IoT and Machine Learning have seen growing adoption for monitoring and optimization, the application of more advanced frameworks such as DTs and AAS remains largely absent, presenting an opportunity for further research and understanding on how asset digitization could enhance production systems.

Chapter 3

Methodology & Domain Description

This chapter outlines the approach to mitigate the previously defined research problem and answer the RQs. The chapter is divided into two main parts. The first presents the **research methodology**, detailing the research approach to contextualize the upcoming chapters, followed by important considerations and the expected outcomes of this study. This methodology is designed to accommodate the complexities of integrating digital-industrial technologies into a multi-organizational aquaculture environment.

The second part of this chapter delivers a detailed **domain description**, focused on the algae cultivation environment that serves as the real-world use case for this study and the technology stack of the organization that this work is involved with. This includes both the biological dimension of algae as a sustainable aquaculture resource and the digital infrastructure that supports the solutions to be presented in the next chapter. Together, these elements form the foundation upon which the research process is executed.

As introduced in Section 1.2, the core research problem centers on the difficulty of **translating modular, standards-based digital frameworks into practical, interoperable solutions** within complex industrial environments. While the literature offers numerous conceptual architectures, few are validated in real-world settings where heterogeneous technologies, legacy systems, and multiple stakeholders must be coordinated.

This challenge becomes especially critical in **large-scale, multi-organizational contexts**, where successful system implementation depends on navigating organizational boundaries, integrating diverse expertise, and ensuring long-term sustainability. The project in which this study is embedded exemplifies this complexity. t brings together institutions, companies, and research teams with the shared objective of advancing digital transformation in algae production systems.

With this context established, the next section presents the methodology adopted to address the research questions and develop scalable, reproducible solutions within the described environment.

3.1 Methodology

To address the RQs and mitigate the identified challenges, this dissertation adopts a three-phase methodology: initial **exploration**, **technology experimentation**, and **validation** through testing and conclusive **evaluation**. This structured approach allows for an investigation of potential implementations within the defined scope of this study, focusing on the integration of the digital technologies into the use case environment. Each phase builds on the previous one, allowing the study to remain adaptable to real-world project constraints while remaining aligned with the overarching research objectives and the digitalization goals of the broader initiative.

The **exploration phase** responds primarily to RQ1, focusing on understanding the requirements and constraints of integrating digital-industrial technologies into legacy systems within a multi-organizational context. This phase combines technical investigation with collaborative stakeholder engagement to elicit key operational and system-level requirements. Through recurring meetings with cross-functional teams—including OT/IT specialists and field operators—this stage identifies essential production variables (relevant information to collect, process, and structure), assets/devices, data flows, and existing infrastructure dependencies. These insights form the foundation for architectural planning and ensure that the methodology is aligned with both organizational realities and user needs.

The second component of this phase—**technology exploration**—focuses on evaluating candidate technologies not only for their theoretical strengths but also for their interoperability, scalability, and fit within the current ecosystem. The objective is to identify suitable tools and frameworks for integration that directly inform RQ2. Since this work is embedded within a large organization, the methodology must consider already established technologies and infrastructures. By assessing available solutions and uncovering system-level constraints, this phase establishes the baseline for the experimentation to follow.

Following the exploratory stage, the research transitions into an **experimentation phase**, a critical step in problem-solving. As Farley [20] described in his book *"Modern Software Engineering"*, experimentation is ***"humanity's best problem-solving technique."*** To effectively address RQ2 and mitigate the identified gaps, prototyping and testing will be carried out to evaluate different technological solutions. This phase aims to compare alternative frameworks, software solutions, and integration approaches, assessing their advantages, limitations, and feasibility within the current manufacturing system, with the objective of uncovering the **"How?"**. The conceptual foundation for this phase is derived from the knowledge outlined in Sections 2.2 and 2.3, which provide some theoretical groundwork for both exploration and practical implementation.

The final phase, **validation and evaluation**, addresses RQ3, focusing on the deployment of selected technologies within the actual algae cultivation real-world environment. This stage is designed to assess the robustness, scalability, and long-term adaptability of the implementations described in Chapter 4. The primary objective is to evaluate system responsiveness and technical performance, ensuring that the solution aligns with the integration requirements identified earlier in the study. In addition to technical metrics, aspects such as usability, maintainability, and adaptability will be examined to enable a comprehensive assessment of the proposed architecture. Serving as a practical testbed, this phase also evaluates the hypothesis introduced in Section 1.4, determining whether the final system meets its intended objectives under real-world conditions.

Ethical Considerations This study is conducted within the framework of a European project involving multiple organizations, necessitating transparency and responsible research practices. Data will be anonymized to ensure privacy and confidentiality, and ethical approval will be obtained to comply with relevant regulations. Transparency regarding objectives, methods, and data-handling procedures will be maintained to ensure informed participation by all stakeholders.

Limitations The differing timelines between the broader project and this dissertation create constraints on implementation and validation processes. In addition, established infrastructures, protocols, and strategies may limit/restrict the scope and flexibility of this research.

Mitigation Plan To address potential challenges related to data accessibility and project timelines, the validation phase will focus on the conceptual implementation of the final architecture. Where access to live operational data is limited, data sampling and synthetic data generation will be employed to simulate realistic scenarios and evaluate the presented implementation. This approach ensures that sensitive data from participating organizations remains protected while still enabling a level of technical detail sufficient for reproducibility. By adopting this strategy, the study remains aligned with its research objectives and provides a transferable framework that can be applied to similar environments.

Expected Results From a researcher's perspective, this study aims to answer the research questions, validate the hypothesis through a structured methodology, and establish a foundation for future research on integrating high-tech solutions into diverse possible environments. From an engineering perspective, particularly in software engineering, the main goal is to develop, deploy, and present the best solution by employing engineering best practices and leveraging common knowledge learned throughout the course.

The focus is on exploring and experimenting with industrial enablers and evaluating their frameworks and best fit within the broader system. Following this methodology, after these stages, the objective is to employ a continuous deployment strategy into the production environment, testing and validating the chosen technologies onto the system. Ultimately, the goal is to present a detailed, modular, scalable, and reproducible solution for various industrial use cases.

3.2 Algae Cultivation Environment and Digitalization Effort

Following the methodology approach, this section introduces the real-world research context that underpins this dissertation—specifically, a European project aimed at advancing sustainable, circular, and digitally integrated aquaculture systems. Within this initiative, algae cultivation serves as both a biological and technological focal point, presenting unique challenges and opportunities for digital innovation. In this context, this process remains largely traditional, with very limited technological integration. Manual observation and human intervention are still the norm, and the facilities' environment imposes certain constraints on scalability, data acquisition, and system interoperability.

To fully understand the constraints, opportunities, and implementation logic behind the proposed digital solutions, this section explores the role of algae as a sustainable resource in aquaculture and its broader environmental and economic potential. The second outlines the existing technological infrastructure of the host organization and the system-level constraints that shape the implementation space for this study.

3.2.1 Algae as a Sustainable Aquaculture Resource

Algae, ranging from single-celled organisms to large seaweed, offer a wide range of possibilities in various sectors [18]. Their potential can be categorized into three main areas: **biorefinery** processes, contributions to the **circular economy**, and **environmental sustainability**. The biorefinery process involves converting algae biomass into biofuels, renewable energy, and valuable materials [65]. Algae also embrace the principles of the circular economy, promoting the reutilization of resources and minimizing waste generation [22]. Moreover, their ability to reduce greenhouse gases contributes significantly to mitigating the effects of global warming and climate change [56].

Biofuel & Bioenergy Algae are widely recognized as one of the third-generation biofuels due to their potential for sustainable energy production stemming from their carbon-neutral life cycle [51]. The process involves the absorption of sunlight and CO2 by the organism to produce biomass, which is then converted into biofuels. As emphasized by ElFar et al. [18], algae have gained significant momentum in recent years as a reliable source of energy.

Multipurpose Biomaterial Algae present immense potential as a versatile biomaterial, with applications spanning multiple industries. According to Munoz et al. [47], algae can be utilized in human and animal food manufacturing, pharmaceutical products, cosmetics, and biomedicine. Additionally, studies by Neo et al. [48] and Tummawai et al. [62], highlight algae's role as an effective and important protein source in plant-based nutrition. Certain algal species are composed of 40% to 60% protein, positioning algae as a critical resource to meet global dietary needs.

Ideal Candidate for Circular economy Macro- and microalgae are characterized by their rapid growth and efficient nutrient absorption, making them ideal candidates for circular economy applications [43]. Their cultivation has been integrated into systems designed to reduce waste and optimize resource utilization. As Kunapinun et al. [43] stated, ***"waste products of one serve as resources for another, enhancing system efficiency and sustainability."***. Microalgae cultivation in wastewater environments exemplifies this concept, enabling dual-purpose systems for water treatment and resource recovery while producing biomass [19].

Reducing Greenhouse Effect Both macro- and microalgae play an essential role in reducing greenhouse gases, with microalgae being particularly significant due to their abundance. Studies by Khor et al. [42] and Yang et al. [66], underscore the role of microalgae in photosynthesis, responsible for producing approximately 50% of the oxygen in Earth's atmosphere while using CO2 throughout the cultivation process. Furthermore, Gotovtsev [23] even highlights the potential for incorporating microalgal cultivation into smart cities and home systems to aid in air treatment.

Among the reviewed literature, an article stands out for its comprehensive coverage of the aspects highlighted in this section. Fabris et al. [19] offer an extensive review of algae's potentials, exploring detailed scenarios and applications. The study includes a user-friendly illustration that encapsulates the various areas of algae utilization and the purposes of their cultivation. This extensive review serves as a valuable resource, providing a deep understanding of the topic, making it an essential reference for exploring algae's diverse applications and benefits.

3.2.2 Research Context and System Constraints

This research is embedded within a broader initiative, which encompasses multiple demonstration projects targeting sustainable aquaculture through the integration of algae-based systems. These projects explore both microalgae and macroalgae applications. Specifically, this dissertation contributes to the development of a ***System for Monitoring and Adaptation in Real Time for Integrated Multi-Trophic Aquaculture*** demonstration, which integrates flatfish aquaculture with macroalgae cultivation. The system's goal is to reuse wastewater by treating aquaculture effluents through macroalgae bioremediation, all supported by advanced digital technologies.

The host institution for this dissertation plays a pivotal role in this demonstration, serving as the main technological partner. Its contributions span both software and hardware domains, with a specific focus on the algae cultivation environment. The organization's work includes the development and deployment of hardware equipment, data acquisition systems, real-time data processing, storage, and monitoring infrastructures. The integrated digital ecosystem that it provides forms the operational and analytical backbone of the aquaculture environment.

To contextualize the solutions presented in the following chapter and to answer the first Research Question of this study, it is essential to understand the current technological stack that this institution has developed and adopted in similar initiatives.

At the physical control level, **Programmable Logic Controllers** act as the system's central processing units (CPUs), executing **real-time automation logic**. For instance, in the algae cultivation ecosystem, these devices manage the behavior of actuators such as electrovalves, blowers, and pumps, using sensor feedback to maintain environmental conditions. The PLC implements predefined control logic with prioritization rules tailored to specific variables (e.g., water temperature, concentration of nutrients, pH).

Communication between PLCs and field equipment occurs via **fieldbus networks**, which support distributed, real-time control. These networks are further extended through fieldbus couplers, which localize connections within control cabinets and reduce wiring complexity. Each physical asset is digitally modeled within the PLC's internal memory via DataBlocks, where sensor readings and control states are mapped to specific memory addresses. This deterministic structure ensures reliability and traceability in environmental control.

To expose PLC data to external systems, this technological partner relies on an **OPC UA-compliant server**, a **Kepware KEPServerEX**. This platform interprets the PLC's internal node structure and publishes it via the OPC UA protocol. OPC UA is widely recognized as an industry-standard communication layer for digital systems and is explicitly referenced in the RAMI 4.0 framework as a core enabler of interoperability (see Figure 3.1).

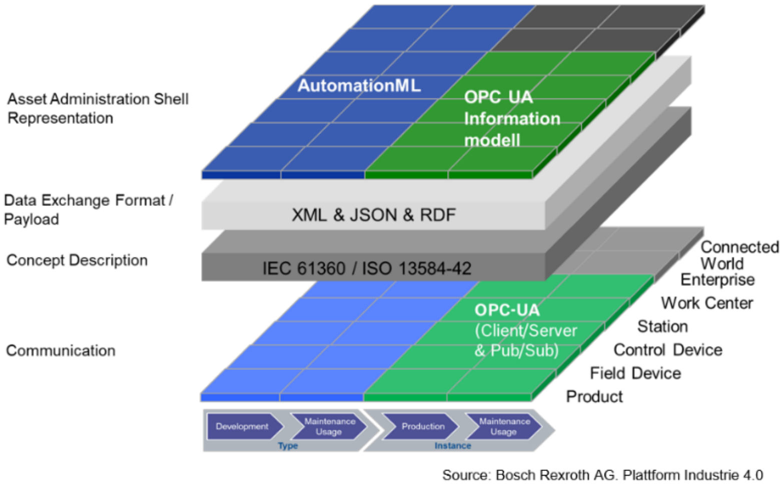

Figure 3.1: Exchange Data Formats for Asset Administration Shell [33]

According to a specification by Plattform Industrie 4.0, the use of OPC UA as a communication format is essential for bridging different layers and lifecycle phases in Industrie 4.0 systems. As the document highlights:

> *"Sharing information between different systems throughout the areas covered by the entire RAMI 4.0 model is crucial for Industrie 4.0 applications. OPC UA has been targeted as a format for information models in the domain of production operations, but there is a need for other formats for the other areas and the interrelationships between them."* Industrie 4.0 [33]

The AAS, for instance, supports multiple communication and serialization formats—including OPC UA, AutomationML, XML, JSON, and RDF—to cover diverse industrial requirements. Each serves a distinct purpose across lifecycle phases, enhancing interoperability and system cohesion.

Within the organization's developments, its digital ecosystem is structured around an ***"IIoT Platform"*** with strong support for data processing, storage, and visualization. As illustrated in Figure 3.2, it integrates several interconnected components to help with these processes:

- **Kafka message broker:** Acts as the data ingestion point for the platform. Following a publish/subscribe model, Kafka captures real-time data and organizes it into topics aligned with the node structure defined in the Thing Manager.

- **Thing Manager:** Comprising the Thing Registry and Thing Replica services, this module manages digital twin representations. Nodes are structured into variables, events, and aggregations, accessible through REST APIs for CRUD operations and visualization.
- **TimescaleDB:** A time-series database used for data persistence and efficient historical querying, supporting analytics and system optimization tasks.
- **API Gateway:** Mediates access to the platform's internal services. It provides an access point via REST for external services, bringing a robust level of scalability and modularity, like a microservices architecture presents.
- **Grafana:** One of the external add-on services, it provides user-friendly dashboards and visualization tools for real-time system monitoring, insight delivery, and even remote control.

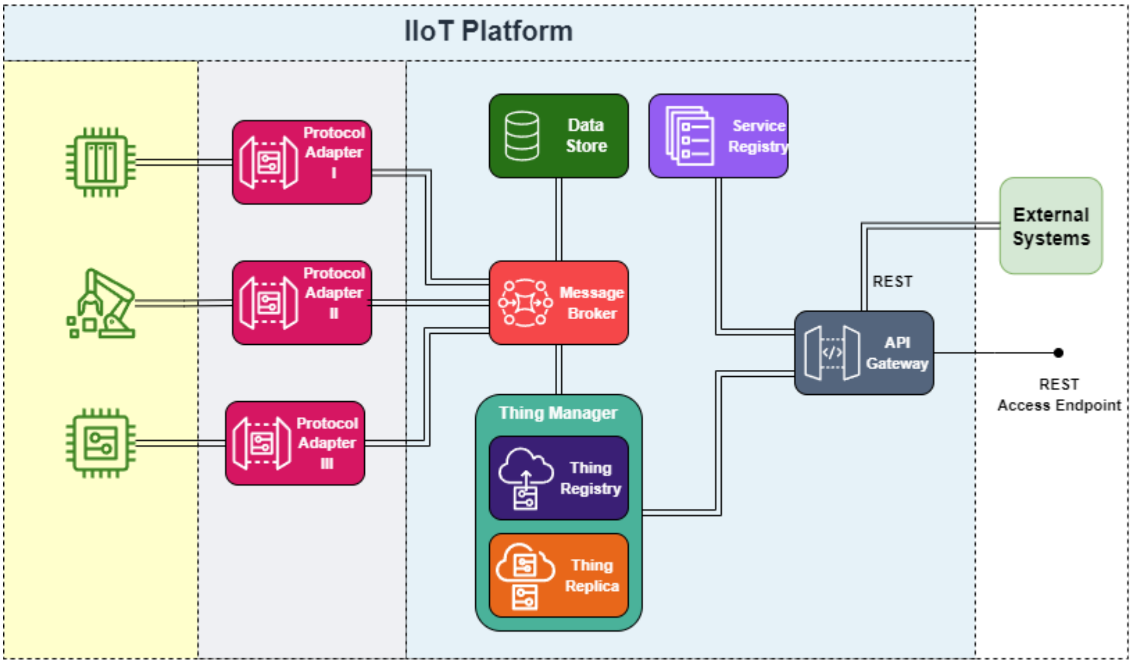

Figure 3.2: The Organization's *IIoT Platform*

Despite the maturity of these individual components, in the overall architecture and the technological approach, there is a critical integration issue. The connection between the **PLC instance** or the **OPC UA server** and the **IIoT Platform**, specifically **Kafka**, remains uncovered. This missing link restricts real-time data flow, limits system responsiveness, and prevents remote access and automated control—key capabilities for advancing intelligent and adaptive aquaculture operations.

The subsequent chapter will focus on this **integration challenge**, presenting a range of solutions that were explored, implemented, and evaluated in a production environment. These solutions will be assessed based on their interoperability, modularity, and long-term scalability and adaptability within the complex aquaculture system.

Chapter 4

Scalable Digital Solutions

This chapter presents and evaluates two technological solutions explored to address the interoperability challenge described in Chapter 3. The focus of this chapter is on detailing scalable, modular, and adaptable solutions that can be replicated in other environments under study or similar digital-industrial systems. Emphasis is placed on addressing real-world challenges in integrating OT with IT and industrial platforms while ensuring long-term digital transformation initiatives. The content that follows offers valuable information concerning RQ1 and RQ2.

Building upon the technological background and research context established previously, this chapter introduces and describes the design, implementation, and deployment of two digital solutions. These efforts, while aligned with the broader goals of the European project, are not narrowed down into the algae use case. Succinct presentations may be found in Table 4.1.

4.1 Bridging OPC UA and Kafka via PLC4X Connectors

The core challenge presented in Section 3.2, in the existing system architecture, was the lack of a native bridge between the operational-level technologies, such as the PLC and OPC UA server, and the Kafka message broker within the IIoT Platform. This interoperability issue is common in industrial environments where legacy systems use protocols not natively compatible with modern data platforms. There are several initiatives developing frameworks to overcome this problem, like Apache Kafka with its implementation ***Kafka Connect***, which focuses on bringing Kafka together with other technologies [6]. Specifically, these implementations are called Kafka Connectors.

Kafka Connectors are plug-and-play components designed to enable the flow of data between Kafka and external sources or sinks [6]. For instance, within the aforementioned digital solution, Kafka connectors are already in use to bridge protocols such as MQTT and Kafka:

- **Source connectors** retrieve data from external systems (e.g., MQTT or OPC UA) and publish it to Kafka topics.
- **Sink connectors** perform the inverse operation, consuming data from Kafka topics and writing it to external systems.

This section presents two specific connectors, source and sink, between OPC UA and Kafka. It leverages the implementation from [4]. These connectors act as a bridge between an OPC UA server and a Kafka message broker, enabling bidirectional communication between OT and the IIoT Platform. These connectors are configured using JSON files and rely on a functional deployment of the following components: Kafka Message Broker, Kafka Connect, an OPC UA server, and optionally, an interface for managing the connectors described in Subsection 4.1.4. The underlying configuration and setup of these components is essential for the implementation of the following approach. However, it is outside the scope of this dissertation.

4.1.1 General Configuration

Both the source and sink connectors share a similar configuration structure, defined via a JSON file. Listings 4.1 & 4.2 present the general characteristics and configuration properties that manage the connection to the OPC UA server from which data is being retrieved or published, the Kafka broker and its specific topics, and finally the intermediary transformations to align the data coming from one data model to another.

Listing 4.1: General Source Connector Configuration

```
name: <connector-name>
config: {
    - connectorClass
    - sources
    - sources Addresses
    - their jobs and kafka topics associated
    - jobs specifications
        - jobs tags and server node associated
    - kafka properties
    - set of transformers
}
```

Listing 4.2: General Sink Connector Configuration

```
name: <connector-name>
config: {
    - connector class
    - sinks
    - sinks Addresses
        - each sink represent a unique task
        - kafka topic associated with sink
    - sinks tags and server node associated
    - kafka properties
    - set of transformers
```

```
}
```

The main components include a ***name***, which is a unique identifier for the connector instance; the ***config***, a nested structure containing the rest of the properties of the connector; and the ***sources*** or ***sinks***, which are a list of the OPC UA servers to connect to. Within the defined source or sink, it is necessary to provide the ***server address*** and the ***tags*** to read from or write to, as represented in Listing 4.3. The necessary parameters of the *connectionString* and the server nodes, represented by the tags, should be formatted as per PLC4X specifications [5]. The remaining properties are related to Kafka, such as its address and topics, and data transformations.

Listing 4.3: Server Address and Node Tags

```
"sinks": "KEPServer",
"sinks.KEPServer.connectionString":"opcua:{transport}://{ip-address}:{port}?{
    options}",
"sinks.KEPServer.tags": "temperature"
"sinks.KEPServer.tags.temperature": "ns={namespace-index};[s|i|g|b]={Identifier};{
    Data Type}"
```

The next subsections will describe each connector configuration in greater detail, along with the role of transformers and the User Interface (UI) used to manage and monitor connectors.

4.1.2 Source & Sink Connectors

The Source Connector is defined by the class ***org.apache.plc4x.kafka.Plc4xSourceConnector***. It is responsible for retrieving data from one or more OPC UA servers and publishing this data to Kafka topics. This connector supports multiple sources and is capable of reading from several node tags simultaneously. The retrieval operations are divided into discrete units called ***jobs***, each of which functions independently. This job-based model significantly enhances scalability and modularity, as it allows the system to execute multiple data retrieval tasks in parallel without mutual interference. In these initial properties, the maximum number of concurrent tasks/jobs and the default Kafka topic are required.

Listing 4.4 introduces an example of a source connector configuration file. For each server instance, it's necessary to define the ***connectionString***, the ***pollInterval*** to connect to the server, and the number of ***retry*** attempts, which are also presented in the sink connector, and ***jobReferences***, which point to the jobs defined below. In this implementation, the Transmission Control Protocol (TCP) is used as the transport protocol. The actual IP address and port are omitted here for confidentiality, and connection options may vary depending on the OPC UA server configuration.

Listing 4.4: Source Connector Configuration

```
{
    "name": "opcua-source-connector",
    "config": {
        "connector.class": "org.apache.plc4x.kafka.Plc4xSourceConnector",
        "default.topic": "{kafka-topic}",
        "tasks.max": "1",
        "sources": "kepserver",
        "sources.kepserver.connectionString": "opcua:tcp://{ip-address}:{port}?discovery=false&tcp.keep-alive=true&tcp.default-timeout=15000",
        "sources.kepserver.pollReturnInterval": "5000",
        "sources.kepserver.retries": 5,
        "sources.kepserver.jobReferences": "temperature",
        "sources.kepserver.jobReferences.temperature.topic": "{kafka-topic}",
        "jobs": "temperature",
        "jobs.temperature.interval": "1000",
        "jobs.temperature.pollReturnInterval": "3000",
        "jobs.temperature.tags": "temperature",
        "jobs.temperature.tags.temperature": "ns=2;s={Path/To/ServerNode};REAL",
        "transforms": "flatten, ReplaceField, addTimestamp",
        "transforms.ReplaceField.exclude": "timestamp,expires",
        "transforms.ReplaceField.renames": "tags.temperature:v_temperature",
        "transforms.ReplaceField.type": "org.apache.kafka.connect.transforms.ReplaceField$Value",
        "transforms.addTimestamp.fields": "ts",
        "transforms.addTimestamp.type": "pt.inesctec.jcustenborder.TimestampNowField$Value",
        "transforms.flatten.delimiter": ".",
        "transforms.flatten.type": "org.apache.kafka.connect.transforms.Flatten$Value",
        "bootstrap.servers": "localhost:9092",
        "key.converter": "org.apache.kafka.connect.storage.StringConverter",
        "value.converter": "org.apache.kafka.connect.json.JsonConverter",
        "key.converter.schemas.enable": "false",
        "value.converter.schemas.enable": "false",
        "errors.log.enable": "true",
        "errors.log.include.messages": "true",
        "errors.tolerance": "all"
    }
}
```

Each job contains the ***interval*** to perform the task, the ***pollReturnInterval***, the time of waiting for server response, and a list of ***server nodes*** to read from. Server nodes were typically extracted using client tools such as **UA Expert**, which provides a graphical interface to browse and retrieve tags paths [63]. A single job can retrieve data from multiple server nodes, and the output message will encapsulate all the values read during that cycle. The Kafka topic for each job is defined in the ***jobReference*** of the source instead of on the job declaration. Kafka topics and the server nodes paths are also omitted intentionally, but those are defined in Apache PLC4X [4] documentation.

The Sink Connector shares a similar structure but performs the inverse operation. It consumes data from Kafka topics and writes it to one or more OPC UA nodes. Unlike the Source Connector, which organizes tasks into jobs, each sink instance is tied to a single Kafka topic and operates as an independent task. While a sink can write to multiple OPC UA node tags, it does not support batching or grouping of tasks as the source connector does, as observed in Listing 4.5. This difference in task structuring reflects the directional nature of the connectors:

- Source connectors emphasize broad, parallel data acquisition.
- Sink connectors focus on targeted, deterministic actuation or command delivery based on incoming Kafka messages.

Listing 4.5: Sink Connector Configuration

```
{
    "name": "opcua-sink-connector",
    "config": {
        "connector.class": "org.apache.plc4x.kafka.Plc4xSinkConnector",
        "topics.regex": "{kafka-topic}",
        "tasks.max": 1,
        "sinks": "KEPServer",
        "sinks.KEPServer.connectionString": "opcua:tcp://{ip-address}:{port}?
            discovery=false",
        "sinks.KEPServer.pollReturnInterval": 5000,
        "sinks.KEPServer.retries": 5,
        "sinks.KEPServer.topic": "{kafka-topic}",
        "sinks.KEPServer.tags": "value",
        "sinks.KEPServer.tags.value":"ns=2;s={Path/To/ServerNode};REAL",
        "transforms": "JsonToString,JsonParser,RenameField,HoistField,insert,
            insertTimestamp,castValues",
        "transforms.JsonToString.type": "pt.inesctec.JsonToString$Value",
        "transforms.JsonToString.field.name": "data",
        "transforms.JsonParser.type": "pt.inesctec.JsonObjectParser$Value",
        "transforms.RenameField.type": "org.apache.kafka.connect.transforms.
            ReplaceField$Value",
        "transforms.RenameField.exclude": "u, ts",
        "transforms.RenameField.include": "v",
        "transforms.RenameField.renames": "v:value",
        "transforms.HoistField.type": "org.apache.kafka.connect.transforms.
            HoistField$Value",
        "transforms.HoistField.field": "tags",
        "transforms.insert.type": "org.apache.kafka.connect.transforms.
            InsertField$Value",
        "transforms.insert.static.field": "expires",
        "transforms.insert.static.value": 0,
        "transforms.insertTimestamp.type": "org.apache.kafka.connect.transforms.
            InsertField$Value",
```

```
        "transforms.insertTimestamp.static.field": "timestamp",
        "transforms.insertTimestamp.static.value": 0,
        "transforms.castValues.type": "org.apache.kafka.connect.transforms.
            Cast$Value",
        "transforms.castValues.spec": "timestamp:int64, expires:int64",
        "bootstrap.servers": "localhost:9092",
        "key.converter": "org.apache.kafka.connect.storage.StringConverter",
        "value.converter": "org.apache.kafka.connect.json.JsonConverter",
        "key.converter.schemas.enable": "false",
        "value.converter.schemas.enable": "false",
        "errors.log.enable": "true",
        "errors.log.include.messages": "true",
        "errors.tolerance": "all"
    }
}
```

Besides the previous terms, some Kafka-related parameters are also essential in the configuration: ***bootstrap.servers***, specifies the Kafka broker address; the ***key.converter*** and ***value.converter***, define the serialization format for keys and values; and finally the ***key.converter.schemas.enable*** and ***value.converter.schemas.enable***. The latter 2 options enable or disable the use of schemas defined by PLC4X [4]. In this implementation, schema support is disabled due to its complexity. However, even without schema enforcement, the data structure must conform to a basic expected format, highlighted in Listing 4.6, which influences the required data transformers.

Listing 4.6: Example of PLC4X OPC UA Basic Message Format

```
{
    "tags": {
        "temperature": 25,
    },
    "timestamp": 1748882259000,
    "expires": 0
}
```

4.1.3 Transformers

Transformers are widely employed in software systems to facilitate interoperability between differing data formats or schemas, particularly in scenarios involving heterogeneous communication protocols. In the context of this work, transformers play a crucial role in adapting the data format produced by the OPC UA Source Connector to match the schema requirements of the IIoT Platform digital. Although Kafka is a schemaless technology, it uses the format shown in Listing 4.7 given that it is compatible with the Thing Replica.

As outlined in Listing 4.6, the raw data retrieved from the OPC UA server does not conform to the schema expected by the IIoT Platform. This discrepancy necessitates the application of a transformation pipeline capable of restructuring the data into a compatible format. The transformation process occurs within the Kafka Connect configuration of the Source Connector and is implemented using a chain of predefined and custom transformers. In the current setup, three transformers are applied sequentially:

- **flatten:** This transformer restructures the incoming message by converting its hierarchical JSON structure into a flat format using a specified delimiter. For instance, a nested JSON object such as *"tags": "temperature": 25* becomes ***"tags.temperature": 25***. This simplification aids downstream processing and filtering.
- **ReplaceField:** Once flattened, the message is refined using this transformer, which allows for the selective removal and renaming of fields. In this case, irrelevant metadata fields are excluded, and the remaining keys are renamed to align with the IIoT Platform's schema. The temperature value is thus reformatted as ***"v_temperature": 25***.
- **addTimestamp:** This is a custom transformer developed internally by the research institution. It appends a timestamp field, *"ts"*, to the message, denoting the precise moment of data acquisition or message processing. This timestamp is crucial for maintaining temporal accuracy within the time-series database and supporting historical analytics.

Listing 4.7: Kafka's Message Output

```
{
    "v_temperature": 25,
    "ts": $timestamp,
}
```

The final output of this transformation pipeline is a minimalistic and platform-compliant message format, as illustrated in Listing 4.7. It aligns with IIoT Platform's digital twin representation and ensures compatibility with its data storage, processing, and visualization components.

Although the discussion above focuses on transformations within the Source Connector, a similar process is applied in the Sink Connector. However, in that case, the transformations operate in reverse—taking a Kafka-compliant message and adapting it to the tag-based data structure required by the OPC UA server. Transformers can either be custom-developed, as demonstrated, or sourced from publicly available libraries such as those provided by Confluent [11]. Their modular design allows developers to tailor data flows to the unique requirements of each integration, ensuring flexibility and maintainability across evolving digital ecosystems.

4.1.4 Connector UI

To streamline the management of Kafka connectors within the IIoT Platform, a web-based UI has been developed. While initial developments leveraged the open-source *landoop/kafka-connect-ui*, limitations in compatibility and long-term maintenance necessitated the design of a new custom solution, since the project has not been actively updated in over five years.

This new interface was implemented using *Node.js*, *Next.js*, and *React*, chosen for their scalability, modern frontend capabilities, and ease of integration with existing web technologies. The primary objective of this application is to provide an **accessible environment for managing Kafka Connectors** without requiring manual edits of complex JSON configuration files. By enabling interaction through **structured web Forms**, this solution simplifies the **creation**, **monitoring**, and **editing** of both source and sink connectors—particularly important for operational teams who may not be familiar with Kafka's internal APIs or JSON formatting.

Figure 4.1 presents the homepage of the Connector UI. The interface clearly segregates source and sink connectors, displaying them on the left and right panels, respectively. Each connector entry includes essential information such as the connector name, class type, and the number of running tasks. Users can perform key management operations—such as restarting tasks, deleting connectors, or viewing configurations—directly from this interface.

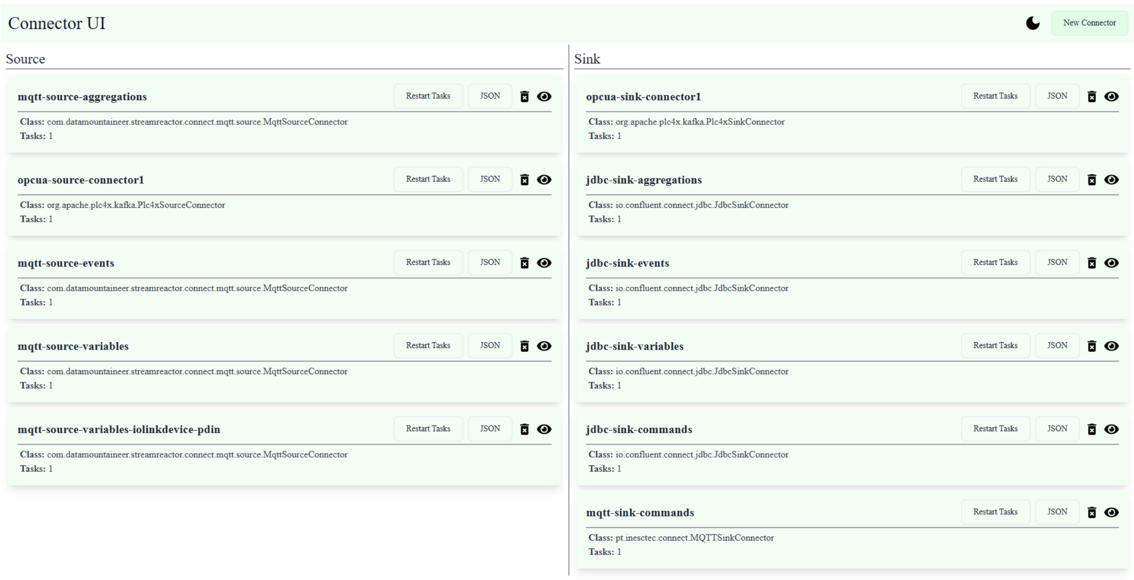

Figure 4.1: Main Page of the Connector User Interface

Importantly, each connector's configuration can also be viewed in a structured format, offering a more intuitive alternative to raw JSON. The interface includes built-in forms that support editing and creation of connectors, as shown in Figure 4.2. These forms guide the user through each required field, offering inline descriptions to clarify terminology and valid input formats.

Although this application was initially developed to support the specific OPC UA–Kafka connectors used in the current study, it was designed with extensibility in mind. The backend architecture supports dynamic loading of other connector classes, allowing the UI to scale alongside the

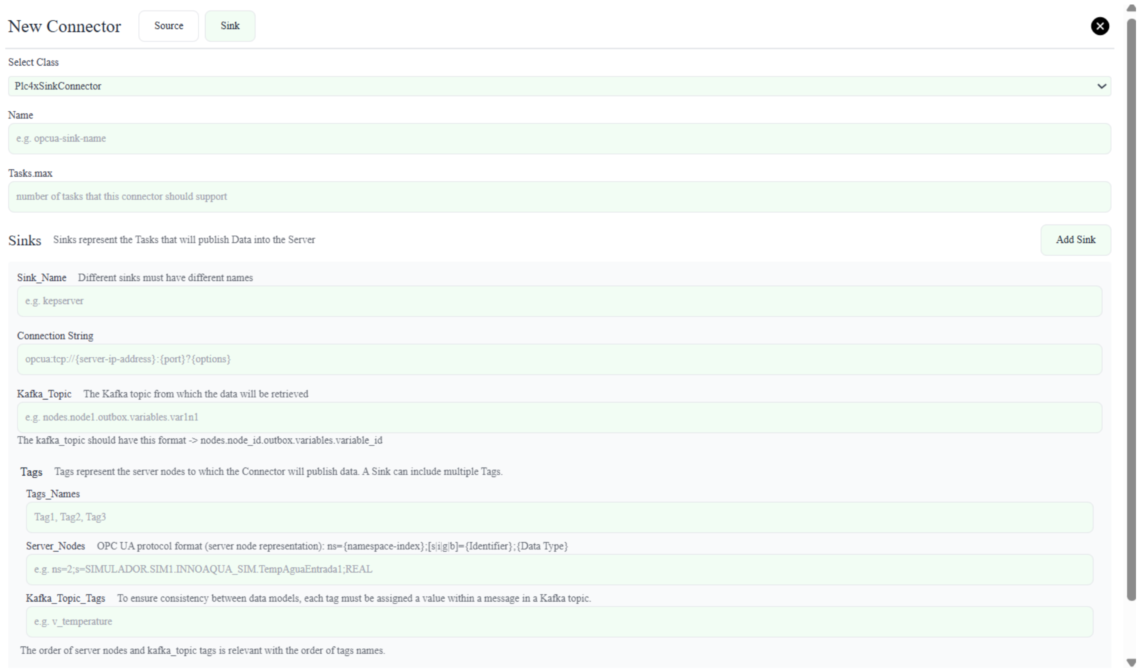

Figure 4.2: Connector UI - Add New Connector Popup

evolving needs of the IIoT Platform. In future iterations, additional connector types and configuration templates can be integrated, reinforcing the modular and adaptable nature of this solution.

4.1.5 Conclusions

This section introduced a connector-based solution specifically designed to address the interoperability challenge identified in Chapter 3, namely the absence of a native bridge between OPC UA and the Kafka-based IIoT Platform. While the implementation is narrowly focused on the use case at hand, the broader value lies in the connector-based architecture itself.

The key contribution of this approach is its modularity and scalability. By abstracting communication between heterogeneous standards—such as OPC UA and Kafka—into configurable connectors, this method enables flexible and reusable data pipelines across diverse industrial environments. The connector framework supports concurrent execution of multiple independent tasks (jobs), making it well-suited for real-time applications. Informal testing conducted on a personal computer demonstrated its ability to handle approximately 34 tasks per second, highlighting its efficiency even in non-specialized hardware settings.

Moreover, the development of a dedicated Connector UI significantly enhances usability. This interface simplifies the creation, configuration, and monitoring of connectors, reducing the technical barrier for operational teams and promoting faster adoption. The UI's architecture also allows for future extension to other connector types, reinforcing the solution's long-term adaptability.

In summary, this proof-of-concept not only bridges two specific systems but also illustrates the viability of adopting a connector-based strategy to achieve scalable, modular, and maintainable integration within complex industrial ecosystems.

4.2 AAS Infrastructure & IIoT Environment

The second solution explored in this dissertation addresses the interoperability concern through the lens of Asset Administration Shell. As discussed in Chapter 2, AAS is a foundational element in the realization of I4.0 architectures that acts as a digital representation of assets, providing a standardized interface for accessing and managing asset-related data [33]. Its importance lies in its capacity to enable interoperable systems, especially in complex environments composed of heterogeneous technologies.

Regarding the challenges previously presented, in the reviewed literature, multiple studies emphasized the pivotal role of OPC UA concerning cross-partner data exchange [9], data aggregation in multi-layered platforms [67], and integration within control systems [10]. These insights validate the strategic relevance of the AAS concept in addressing the limitations of the organizational infrastructure under observation.

To bring this vision into practice, this research adopts Eclipse BaSyx, an open-source and actively developed implementation of the AAS. With broad support for Java, Python, C++, and C#, Eclipse BaSyx provides a robust suite of components for the realization of digital twins, along with tools for communication, configuration, visualization, and integration [17]. The following subsections present the key architectural components, deployment approach, and configuration patterns leveraged in this solution.

4.2.1 Eclipse BaSyx & AAS Components

The Eclipse BaSyx stack is developed as part of the BaSys 4.0 initiative by Fraunhofer IESE [21], which aims to deliver reference implementations for key I4.0 concepts. The middleware supports complete digitalization of the shop floor by enabling seamless integration between OT and IT.

BaSyx developments include several essential components to manage, visualize, and interact with AASs and their Submodels. In this implementation, these components were deployed using Docker containers, configured primarily via explicit configuration files or Docker environment variables. For deployment, the **BaSyx Starter Kit** by Eclipse Foundation [16], a lightweight orchestration tool based on Docker Compose, was used to facilitate initial setup, which provides similar configuration files as the ones presented below and the Docker Compose file.

AAS Environment stands as the central orchestrator component that aggregates the **registries and repositories for AASs and Submodels**. The configuration file presented in Listing 4.8 contains properties of each component and additional features relevant for the final outcome. These properties could have been defined via **Docker environment variables** instead.

Listing 4.8: AAS Environment Configuration File

```
server.port=8081
spring.application.name=AAS
```

```
basyx.backend = InMemory

basyx.environment=file:aas
basyx.cors.allowed-origins=*
basyx.cors.allowed-methods=GET,POST,PATCH,DELETE,PUT,OPTIONS,HEAD

basyx.aasrepository.feature.registryintegration = http://aas-registry:8080
basyx.submodelrepository.feature.registryintegration = http://sub-registry:8080
basyx.externalurl = http://host.docker.internal:8081
# basyx.externalurl = http://localhost:8081

spring.servlet.multipart.max-file-size=500MB
spring.servlet.multipart.max-request-size=500MB

mqtt.clientId=AAS-Env-8081
mqtt.hostname=mosquitto
mqtt.port=1883
basyx.aasrepository.feature.mqtt.enabled=true
basyx.submodelrepository.feature.mqtt.enabled=true
```

The ***basyx.backend***, specifies the backend storage for persisting AAS data. Regarding this attribute, there are a lot of options, and further on in this document, its relevance will be understood. The upload endpoint or ***basyx.environment*** points to the directory or file containing AAS data in AASX, JSON, or XML formats. The URLs for the **integration between repository and registry** components resolve using Docker container names as hostnames, ensuring service discovery in containerized environments. The ***externalurl*** endpoint declares the endpoint used to expose the AASs and Submodels via HTTP. In this configuration, there are 2 options for the *externalurl* due to the deployment environment chosen. The ***mqtt.client*** enables MQTT-based eventing, which forwards incoming AAS data to an MQTT broker, further discussed in Subsection 4.2.3.

AAS and Submodel Registries provide RESTful APIs for registering and locating AAS and Submodel descriptors. Listing 4.9 showcases a minimal configuration, which can be extended with additional features. Each registry exposes a Swagger UI for testing and documentation.

Listing 4.9: AAS and Submodel Registry Configuration Files

```
---
basyx:
  cors:
    allowed-origins: "*"
    allowed-methods: "GET,POST,PATCH,DELETE,PUT,OPTIONS,HEAD"
```

AAS and Submodel Repositories offer full CRUD capabilities over AASs and Submodels through REST APIs. Typically accessed via ***/shells*** and ***/Submodels*** paths, these services are tightly coupled with the AAS Environment, the listing 4.8, and operate on the same ***server.port***, illustrated in Figure 4.3.

Concept Description (CD) Repository supports the **semantic definition layer** of the AAS, storing structured vocabularies and definitions. It enables meaningful interpretation of AAS Properties and supports ontology-based structuring. It is also present in the Swagger interface as the previous repositories, shown in Figure 4.3.

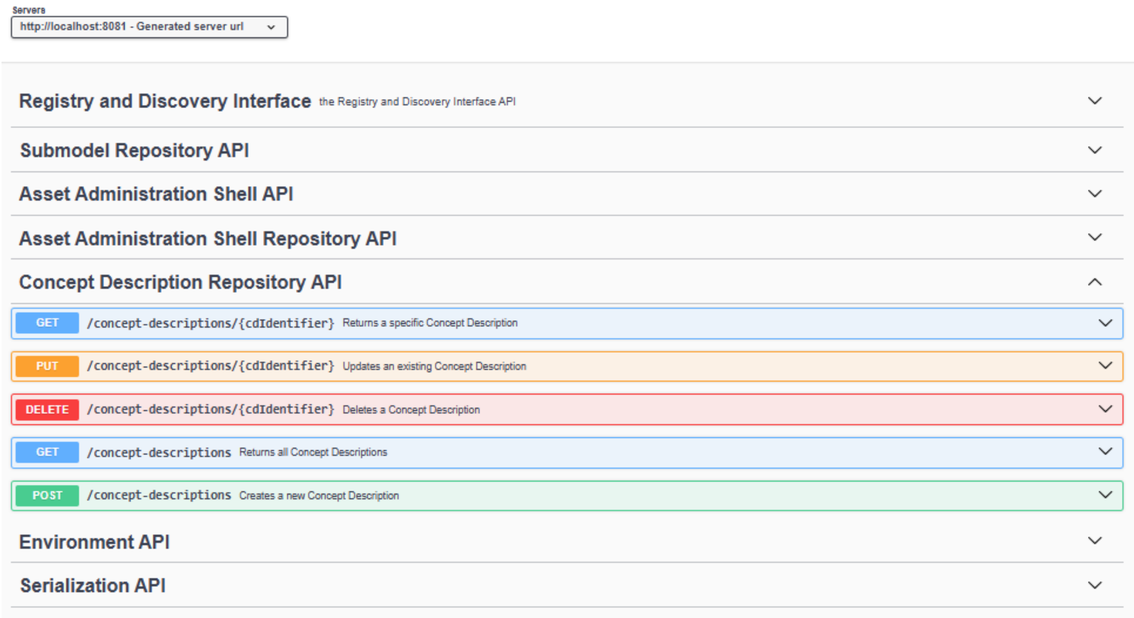

Figure 4.3: Swagger UI with CRUD Operations for AASs, Submodels, and CDs

AAS Discovery Service enables identification of AAS instances based on their associated asset identifiers. This is useful when working with hierarchical structures such as Bills of Material (BoM), facilitating efficient navigation between related assets. In addition to other components, it requires on configuration, as seen in Listing 4.10, and provides a Swagger UI for such discovery. To access Swagger UIs, simply add ***/swagger-ui/index.html*** to the component's endpoint, like *http://localhost:8084/swagger-ui/index.html*.

Listing 4.10: AAS aDiscovery Configuration File

```
server.port=8081
spring.application.name=AAS Discovery Service
basyx.aasdiscoveryservice.name=aas-discovery-service

basyx.backend=InMemory
```

```

basyx.cors.allowed-origins=*
basyx.cors.allowed-methods=GET,POST,PATCH,DELETE,PUT,OPTIONS,HEAD
```

AAS Web UI is a *Vue.js*-based front-end application offering an intuitive interface for interacting with AASs. It enhances usability by allowing users to:

- Visualize AAS trees and structures.
- Edit Properties and invoke Operations.
- Access semantic documentation and graphical views.

This UI complements the Swagger-based APIs by making the interaction experience more accessible, especially for non-technical users. Figure 4.4 illustrates the main interface layout.

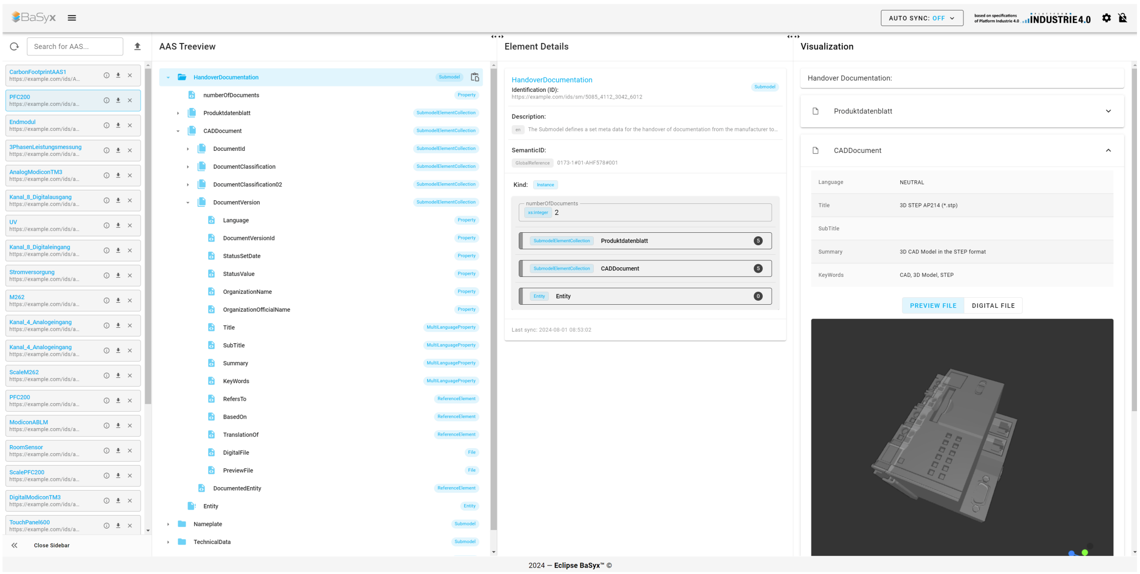

Figure 4.4: AAS Web UI (Source: Eclipse BaSyx [13])

DataBridge Component acts as a central enabler of interoperability regarding the challenge presented. It allows seamless data integration between AASs and external systems using various industrial protocols, including **OPC UA**, MQTT, Kafka, and **PLC drivers via Apache PLC4X**. BaSyx supports bidirectional communication only with MQTT. Only incoming data flows are permitted with other protocols. It may be deployed via Docker, and its setup typically consists of three configuration files:

- **AAS Data Sink:** Defines the Submodel endpoint and the ID short of Submodel Element where data will be written.
- **Data Consumer:** Specifies protocol-specific parameters for consuming external data.

- **Routes Configuration:** Connects consumers to sinks, applying optional transformations and selecting either polling (timer) or event-driven modes.

The following subsection establishes the personalized approach to the gap that was previously discussed, as well as the protocols that were selected, the developments that have occurred in the context of the protocols, and the testing scenarios that will be used to verify the successful implementation of these protocols.

4.2.2 Operational Technologies and AAS Integration Solution

This subsection details the route adopted to enable bidirectional communication between OT systems and the AAS. Nonetheless, given the scalability and interoperability provided by this paradigm, additional alternatives can be explored.

To recall the integration choices accessible, an **OPC UA server** was available, which was linked to the current DataBlock structure of a **Siemens S7-1200 PLC**. Before making a decision, both approaches were evaluated using the Eclipse BaSyx Databridge Component. These options were researched and attempted; nonetheless, due to the restricted availability of outgoing data flows, neither was a viable alternative for providing remote system control. Other technologies may address this issue, but owing to time constraints and the effort involved in this middleware, an adaptation was rejected.

Therefore, a Spring Boot RESTful application was created to handle the bidirectional connection directly with the PLC. The application is written in Java, which is the major programming language for Eclipse BaSyx developments. It uses **Apache Camel** along with **Apache PLC4X** for connection, routing, and data flow, with a **Siemens S7 driver** chosen for compatibility with the PLC unit [2, 3]. Moreover, the software benefits from **BaSyx REST API** endpoints to retrieve or write to AAS Submodel Elements.

The application is hosted on **port 8089** and is designed to be portable across environments, binding to **0.0.0.0** to allow external access. It builds upon more advanced configurations grounded by one of Eclipse BaSyx's examples, bypassing the default setup [15].

This implementation exposes four core endpoints that facilitate start/stop operations for both reading and writing tasks:

- POST **/start-continuous-reading**
- POST **/stop-continuous-reading**
- POST **/start-continuous-writing**
- POST **/stop-continuous-writing**

Each request is structured with parameters configured as ***Operational Variables***, the payload format that aligns with AAS's **Operation** Submodel Element convention, and is designed to be seamlessly managed via the AAS Web UI. These arguments include the target AAS **Submodel endpoint**, the **Submodel Element's short ID** (which is the name of the Property), the **PLC tag**, and, in the case of the reading operation, the **polling interval**, as demonstrated in Figure 4.5. As a result of this development choice, the invocation of such Operations is managed directly within this BaSyx interface, which simplifies the whole process since the Submodel endpoint parameter requires its encoded format that may be copied directly on the interface. This structure's function is to **attach a Submodel Element to a tag inside a PLC DataBlock** so that a value can be read for historical monitoring or changed remotely via the digital replica.

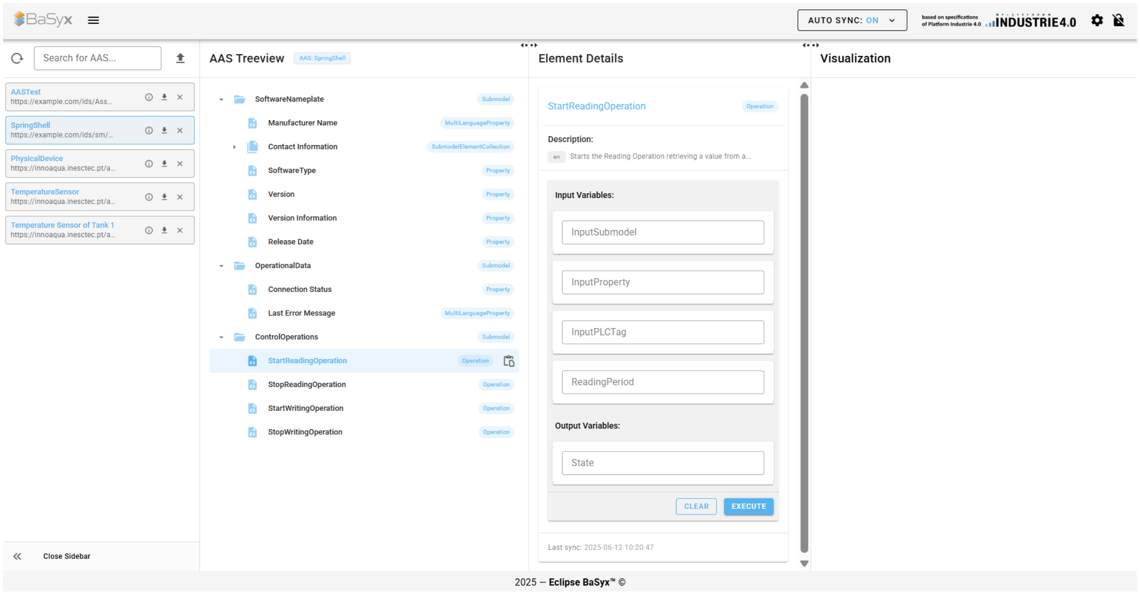

Figure 4.5: AAS of Spring Application on AAS Web UI

Due to Siemens S7 PLCs lacking native support for event-driven mechanisms, read operations are implemented using a polling approach at user-defined intervals. In contrast, write operations leverage the MQTT Eventing mechanism provided by Eclipse BaSyx. An MQTT broker is deployed alongside the BaSyx components to emit update events. As detailed in Listing 4.8, the AAS Environment includes an MQTT client that subscribes to element-level update topics of the following form:

Topic: "sm-repository/sm-repo/submodels/" + EncodedSubmodelURL + "/submodelElements/" + SubmodelElementName + "/updated"

Upon receiving a message, a **singleton MQTT client** within the Spring application parses the data—originally received as a *String* from the AAS—and converts it into a format compatible with the target PLC tag. The tag format adheres to PLC4X conventions, exemplified below.

%DB<DataBlock>:<Offset>:<DataType> (Source: Apache PLC4X [3])

To manage ongoing reading and writing operations, the application uses map structures to track active tasks, preventing duplication and enabling graceful shutdowns. Since these operations are ongoing processes, a ***ThreadPoolExecutor*** service is established to distribute load among available *worker threads* for optimized performance. This ***ExecutorService*** continually submits new tasks to the thread pool due to the polling mechanism and upcoming events of reading and writing operations, respectively.

A **lock-based mechanism** ensures thread safety during connection setup and operation execution. Listing 4.11 provides an example of this logic, showing how the connection state is validated and re-established if necessary. Before returning the endpoint, the *Class* in charge of the singleton connection checks for connectivity. If there is a problem in any operation, the connection is restarted and shared among all workers.

Listing 4.11: PLC Connection Code Snippet

```
public static synchronized Plc4XEndpoint getConnection() {
    plcLock.lock();
    try {
        if(!isConnected) {
            if(endpoint == null) {
                System.out.println("Connecting to PLC.");
                endpoint = new Plc4XEndpoint(PLC_CONNECTION_STRING, new Plc4XComponent());
            } else {
                endpoint.close();
                endpoint = new Plc4XEndpoint(PLC_CONNECTION_STRING, new Plc4XComponent());
            }
            endpoint.setupConnection();
            isConnected = true;
        }
        return endpoint;
    } catch (Exception e) {
        System.out.println("Error connecting to PLC: " + e.getMessage());
        isConnected = false;
        e.printStackTrace();
        return null;
    } finally {
        plcLock.unlock();
    }
}
```

Additionally, to enhance resilience, the application implements a state machine-based fault detection mechanism for the PLC connection. In scenarios involving network disruptions (e.g., VPN/WiFi failures) or internal timeouts, the connection state is flagged as invalid. This triggers a reinitialization of the connection, ensuring minimal downtime and reliable long-term operation. A simplified version of this logic is presented in Listing 4.12.

Listing 4.12: Write Operation Code Snippet

```
public static void safeExecuteWrite(Map<String, Map<String, Object>> tags, String
    plcTag, Object valueToWrite) throws Exception {
    plcLock.lock();
    PlcWriteRequest writeRequest = endpoint.buildPlcWriteRequest(tags);

    CompletableFuture<? extends PlcWriteResponse> future = writeRequest.execute();

    try {
        PlcWriteResponse response = future.get(5, TimeUnit.SECONDS);
        if (response.getResponseCode(plcTag) != PlcResponseCode.OK) {
            System.out.println("Error writing to PLC: " + response.getResponseCode(
                plcTag));
        }
    } catch (Exception e) {
        e.printStackTrace();
        setConnectionStatus(false);
        throw e;
    } finally {
        plcLock.unlock();
    }
}
```

In conclusion, this specially designed application combines Apache PLC4X connectivity, Eclipse BaSyx API endpoints for AAS data administration and MQTT eventing features, and AAS Operations and its design, which can and should be accessed through the AAS Web UI, to bridge one of the OT integration possibilities with AAS. It makes bidirectional data flow possible, which is a significant accomplishment for the subsequent stages: monitoring and preservation to offer field operators observability and remote control of the whole system.

4.2.3 Integration Architecture and Data Flow Environment

This subsection details the IIoT environment developed to support the validation of the AAS-based integration strategy presented in the previous sections. The primary goal was to simulate a production-grade infrastructure capable of demonstrating real-time data acquisition, semantic representation via AAS, and dashboard monitoring through industry-aligned tools.

While the original intention was to integrate directly with the organization's IIoT Platform, specifically its Kafka message broker, this path required aligning the AAS-generated data schema with the platform's internal structure and topics. Due to the time-intensive nature of this integration, an alternative environment was designed. This setup preserves the interoperability, monitoring, and remote control capabilities required for the use case while enabling functional validation and observability in a self-contained deployment.

The Eclipse BaSyx Databridge component, as previously observed with the OPC UA and PLC4X protocols, provides Kafka integration only as a data consumer. In other words, while it supports subscribing to Kafka topics and ingesting data into the AAS environment, it does not offer mechanisms for producing or routing AAS-generated data to Kafka or other heterogeneous sinks. The design focus of the Databridge is on ingestion and transformation into AAS-compliant structures, not on outbound interoperability.

Still, the AAS Environment component offers several **backend storage** configurations via the ***basys.backend*** property. These include:

- **In-Memory** Hash Map Storage.
- **MongoDB** that supports JSON document storage and querying but does not meet the efficiency demands of time-series data flows.
- **Kafka + In-Memory or Kafka + MongoDB**, which represent the most scalable and IIoT-compliant configuration, enabling decoupled data streams through Kafka.

Given the strategic alignment with the Platform and the Kafka-based message broker, the Kafka storage option would ideally serve as the bridge between AAS and the IIoT ecosystem. However, integration requires alignment of AAS structures with message schemas, internal stream processing pipelines, and validation within the platform's data model. Time constraints and technical complexity led to the establishment of an alternative yet scalable environment for testing and deployment, postponing these options for future considerations Section 6.2.

As a workaround, the AAS MQTT eventing feature was leveraged to build an end-to-end data pipeline. This mechanism enables the automatic publication of AAS Property updates to corresponding MQTT topics. This system architecture includes the following components:

- **MQTT Broker:** A lightweight broker was deployed alongside AAS components to handle all event-based messaging. When an AAS Property is updated (either manually or via PLC write operations), BaSyx triggers a message to a structured MQTT topic.
- **Telegraf:** Telegraf serves as the intermediary pipeline component, with a configuration outlined in Listing 4.13. Its MQTT consumer plugin subscribes to selected MQTT topics and formats the incoming data for time-series ingestion. It also applies basic filtering, transformation, and routing rules, ensuring that only relevant metrics reach the database.
- **InfluxDB:** Selected for its native time-series optimizations, InfluxDB enables efficient storage and historical data querying. It organizes data into "buckets", each corresponding to a logical data stream. In this deployment, a bucket named "readings" was used to store all telemetry data emitted from the AAS infrastructure.
- **Grafana:** For monitoring and observability, Grafana was added to the stack. Its InfluxDB connector was configured to retrieve historical data, allowing real-time dashboard construction. The dashboards facilitate performance tracking of the system, validation of PLC data propagation, and visual inspection of AAS value consistency. It also enables remote adjustments of cultivation parameters through the setpoints, as demonstrated in Figure 4.6.

Listing 4.13: Telegraf Configuration File

```
# Configuration for telegraf agent
[agent]
  debug = true
  interval = "10s"
  round_interval = true

  metric_batch_size = 1000
  metric_buffer_limit = 10000

  collection_jitter = "0s"
  flush_interval = "10s"
  flush_jitter = "0s"
  precision = ""

  omit_hostname = true

[[outputs.influxdb_v2]]
  urls = ["http://influxdb:8086"]
  token = "<token-defined-in-docker-compose>"
  organization = "<org>"
  bucket = "readings"
  timeout = "5s"

[[inputs.mqtt_consumer]]
  servers = ["tcp://mosquitto:1883"]
  topics = [
    "telegraf/host01/cpu",
    "telegraf/+/mem",
    "sm-repository/sm-repo/submodels/#", # every topicstarting with this prefix
  ]
  qos = 0
  client_id = "telegraf"
  data_format = "json_v2"

  [[inputs.mqtt_consumer.json_v2]]
    [[inputs.mqtt_consumer.json_v2.tag]]
      path = "idShort"
    [[inputs.mqtt_consumer.json_v2.field]]
      path = "value"
      rename = "current_value"
      type = "float"
```

To validate this environment, a test case was implemented starting on the PLC. A DataBlock was configured with two tags: **Heartbeat**, an integer that increments from 0 to 59 in a loop, changing every second, and **Write**, a mutable register to which AAS Operations can write. An AAS was instantiated containing a Submodel with two Properties corresponding to these tags.

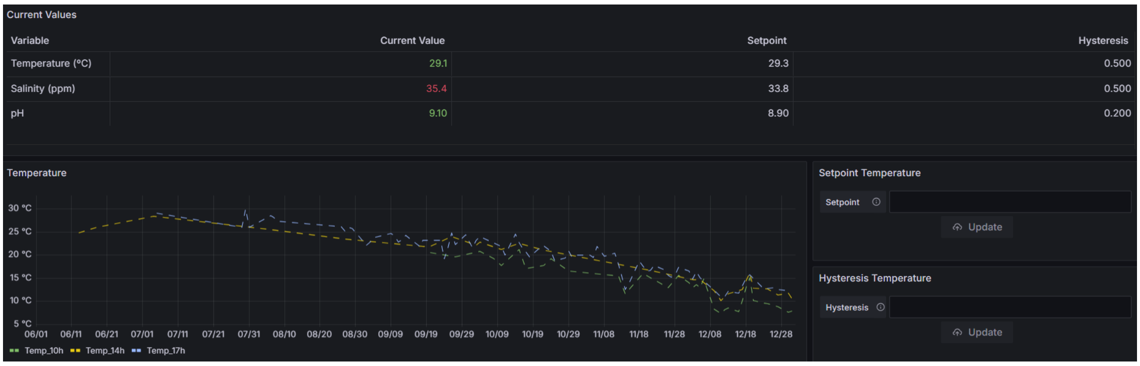

Figure 4.6: Grafana Dashboard for Monitoring and Remote Control

Using the Spring Boot integration layer described in Section 4.2.2, reading and writing operations were started through the AAS Web UI. The **StartReadingOperation** initiated a polling routine that updated the AAS Property every second. This update, in turn, triggered the MQTT eventing mechanism, sending the new value to a predefined topic (e.g., *.../Heartbeat/updated*). Telegraf subscribed to this topic and forwarded the messages into the readings bucket of InfluxDB. Grafana then visualized these values on a dashboard, confirming the end-to-end functionality of the system. Figure 4.7 illustrates this loop. A controlled interruption in the WiFi connection was also introduced to observe the system's recovery behavior. Upon network re-establishment, the reconnection mechanism restarted the polling process, and the data pipeline resumed seamlessly.

Figure 4.7: Grafana Dashboard with Heartbeat Example

This IIoT environment was deployed on a **Virtual Machine** (VM) accessed via a VPN-secured SSH connection. The architecture includes a **Reverse Proxy**, used to expose key web-based UIs over HTTP/s with port routing and security rules. Grafana access is protected via credentials and user roles, while the AAS Web UI uses **Role-Based Access Control** (RBAC) to limit operations and AAS editing rights. All services—AAS components, MQTT, Telegraf, InfluxDB, and Grafana—were deployed using **Docker** for modularity, replication, and maintenance simplicity. Figure 4.8 presents an architectural overview of the deployed IIoT environment.

Although the solution presented does not yet integrate with the technological institution's IIoT Platform, this alternative environment demonstrates a multi-service event-driven approach grounded in I4.0 principles. This setup enables both real-time observability and remote interaction with physical assets. The architecture showcases how an AAS-compliant stack can be deployed,

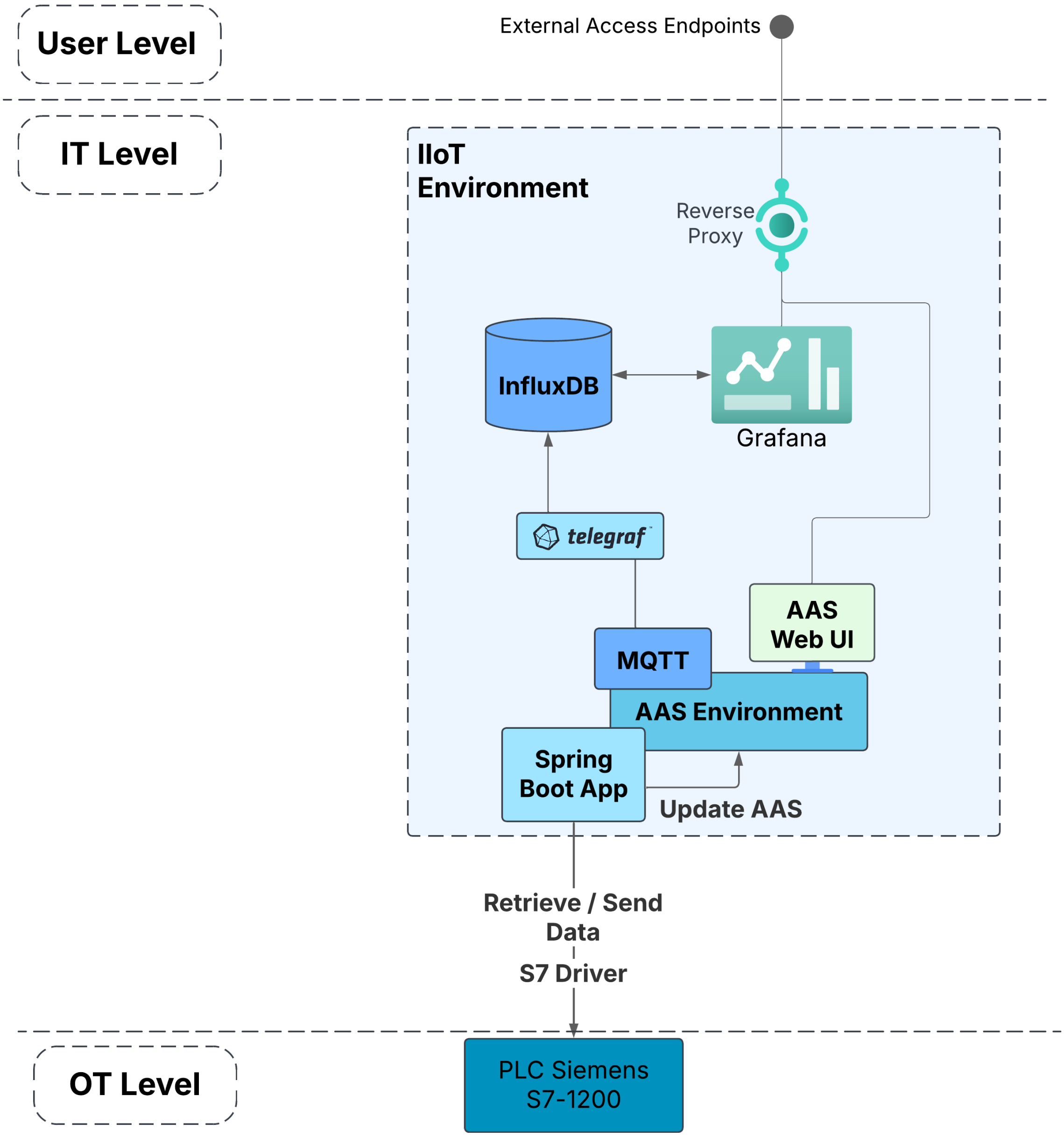

Figure 4.8: IIoT Architecture

monitored, and scaled while preserving interoperability and future integration potential. The following chapter builds upon this digital approach, digging into the creation and structure of the AASs and their role in digitally representing the algae cultivation ecosystem.

4.3 Conclusions

This chapter presented two distinct approaches for enabling scalable digital solutions in industrial environments, particularly in the context of IIoT integration and real-time system monitoring. The following table summarizes both approaches and highlights the core differences, advantages, and suitability of each one:

Aspect	Connector-Based Approach	AAS-Based Approach
Core Concept & Summary	Uses source and sink connectors (via Apache PLC4X) to bridge OPC UA and Kafka. Suitable for modular, rapid integration within existing Kafka-based environments.	Builds on the interoperability of the AAS standard. Leverages Eclipse BaSyx API Endpoints, AAS Components, and a custom-developed REST Application for PLC communication.
Interoperability Strategy	Protocol connector-based; maps OPC UA structure into Kafka messages using transformers.	Semantic and standardized digital representation for data exchange.
Platform Integration	Natively aligns with organization's IIoT Platform (Kafka); easy deployment with a low entry barrier.	More complex integration; Kafka backend support exists but is not yet adopted in this work.
Data Flow Mechanism	Bi-directional via source/sink polling; message transformation through Kafka connectors.	Bi-directional; polling for reads and MQTT-based eventing for writes.
Best Suited For	Fast, lightweight integrations within known infrastructures or pilot setups.	Long-term, scalable, and standardized environments requiring full asset digitalization.

Table 4.1: Comparison of Connector-Based and AAS-Based Approaches

Chapter 5

Asset Administration Shells for Algae Cultivation Environment

This chapter presents the **digitization of the algae cultivation environment** through the design and deployment of AASs. While Chapter 4 focused on the interoperability benefits of the AAS-based infrastructure, it did not dig into the formal digital modeling of the production assets. This chapter fills that gap by defining how physical and cyber-physical components are semantically structured and represented through AASs, ensuring a **lifecycle-oriented**, modular, and interoperable digital twin system, offering helpful insights on answering RQ3, about the validation of the previous AAS-based and IIoT infrastructure implementation into a real-world ecosystem.

The cultivation environment includes a variety of asset types: **physical devices** such as sensors, blowers, valves, and pumps; biological assets like the **algae** itself; composite systems such as **cultivation tanks**, which house multiple sensors and actuators; and software components including the custom **Spring Boot application** that facilitates communication with the PLC system. Each of these components must be correctly modeled to enable semantic integration, data interoperability, and traceability across the system's operational lifespan.

This chapter is structured into two main sections. First, it examines the key information structures and semantic layers underlying the AAS model. Then, it explores the process of building digital twins by deploying and connecting AASs in the context of the algae cultivation ecosystem.

5.1 AAS Structures, Semantics, and Templates

A fundamental concept in the AAS framework is the distinction between **types** and **instances**. A type defines a general template or blueprint for an asset class, specifying its structure, expected Submodels, and semantic annotations. In contrast, an instance represents a specific, real-world manifestation of that asset, adhering to the schema defined by its corresponding type.

To illustrate this, Figure 5.1 shows a sample type-instance relationship in the AASX Package Explorer, a tool used for managing and visualizing AAS packages [25]. For example, the type *PhysicalDevice* defines a generic structure for representing any sensor or actuator. It defines how the information should be structured and which Submodel Elements should be presented. The instance *TemperatureSensor* is a specific instantiation tied to a temperature sensor. It follows, or more specifically is ***conformant to***, the type *PhysicalDevice*. Furthermore, it defines the information that is constant throughout all temperature sensor instances.

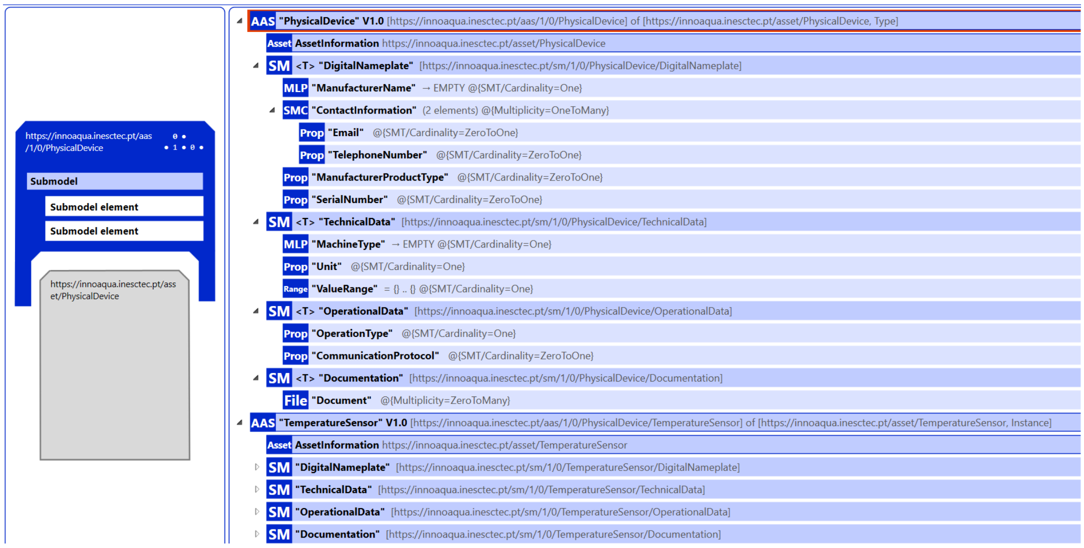

Figure 5.1: AAS of *PhysicalDevice* and *TemperatureSensor* on AASX Package Explorer

The system under study includes several physical temperature sensors deployed across different tank locations. As visualized in Figure 5.2, two specific instances—*TemperatureSensorTank1* and *TemperatureSensorTank5*—were derived from the original *TemperatureSensor* instance and are conformant to another AAS type, the *PhysicalDeviceTank*, similar to the former ones. Both instances maintain the core attributes of the parent model via the **derivedFrom** relationship while presenting location/environment-specific data and operational values.

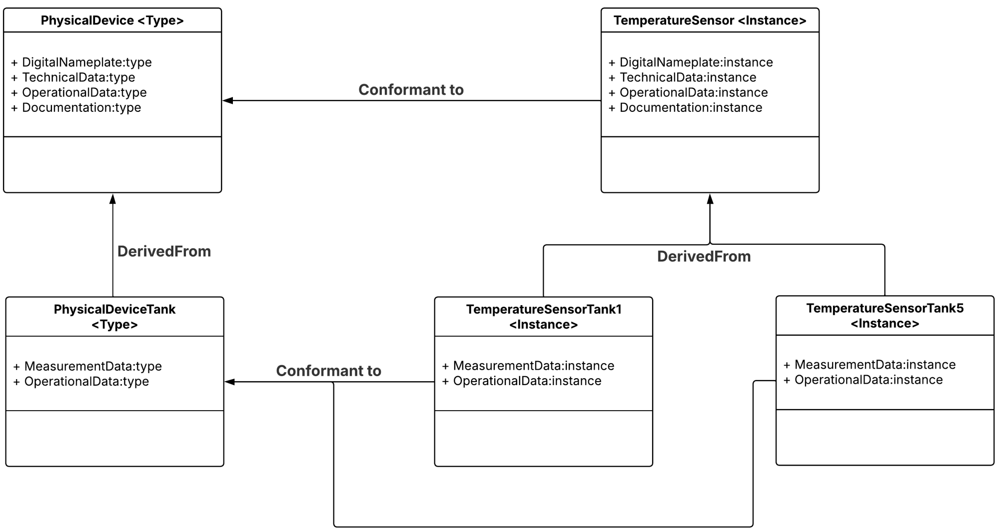

Figure 5.2: Physical Device and Temperature Sensor UML Diagram

Each AAS is composed of one or more Submodels, which define the internal structure and content of an AAS's data. To ensure interoperability and semantic alignment, several organizations—**Plattform Industrie 4.0**, the **Industrial Digital Twin Association** (IDTA), and **ZVEI**—have proposed standardized Submodel templates for common asset classes and industry applications.

This topic was introduced earlier in the literature review. In particular, Abdel-Aty, Negri, and Galparoli [1] presented several design visions for Submodel creation within different applications and multiple situations. It highlighted the benefits of using standardized templates. In the present work, the *PhysicalDevice* type includes four Submodels, three of which are based on officially recommended templates: **DigitalNameplate**, **TechnicalData**, and **Documentation/HandoverDocumentation**. These templates provide semantic consistency and interoperability by defining standard attributes and relationships.

Therefore, these templates were examined beforehand, and Submodel components were chosen, duplicated, and modified to fit the use case. Along with these came the **Concept Descriptions** and their semantic references. CD links the elements of Submodels to global semantic definitions, enabling machines to interpret and process asset-related information unambiguously [33]. Platform Industrie 4.0 states that it is best practice to incorporate the CDs into the AAS Environment rather than merely referencing them globally. In addition to being retrieved from the Submodel templates, CDs can be sourced externally from semantic repositories such as ECL@SS and IEC CDD [12, 29] or even custom-created when domain-specific attributes are needed.

When creating CDs, Submodels, or AASs, unique identifiers are requested, typically expressed as URIs or IRDIs. Following recommendations from Platform Industrie 4.0, this study adopted a structured format [33]:

```
https://<domain>/<aas|sm|cd>/<version>/<revision>/<elementName>
```

An important feature of the AAS information model visible in the figure 5.1, is the ability to attach Qualifiers to Submodel Elements. Qualifiers provide metadata that describes constraints, conditions, or auxiliary information relevant to an element's interpretation. There are three main types of Qualifiers:

- **ValueQualifier** applies to runtime values with `kind="Instance"`. Describes expected properties of the live value of a Property.
- **ConceptQualifier** describes aspects of the concept you're semantically linking to. Can define constraints, rules, and behavior on the semantic concept side.
- **TemplateQualifier** applies to elements of `kind="Template"`. Says how the template should be used, typically structural rules.

In this work, qualifiers served two primary purposes. A *TemplateQualifier* was applied to Submodel templates to indicate cardinality and conformance rules, aiding in the generation of derived Submodels. A *ConceptQualifier* was added to **Operation** elements to support **Invocation Delegation**, a feature provided by Eclipse BaSyx [14]. Specifically, the **invocationDelegation** type was used to redirect execution to a specified REST endpoint, as discussed in Chapter 4.

5.2 Building Digital Replicas via AAS

This section outlines the AAS components for the digital representation of the algae cultivation environment. It presents the information and structure for each of them and their correlation. The necessary AAS types and instances presented in this section showcase the minimal viable ecosystem's structure, providing a foundation for scaling into similar applications.

The physical devices enable the monitoring and control of the cultivation process. As previously mentioned, a generic AAS type, *PhysicalDevice*, was created to represent any field-level hardware, including sensors, valves, pumps, and other actuators. As seen in Figure 5.1, the type comprises four key Submodels.

- A **DigitalNameplate** based on the ZVEI template, which includes manufacturer data, serial number, product type, and contact information [70];
- **TechnicalData**, partly inspired by the ZVEI template, yet customized to include dimensional and physical properties of the device [69];
- **OperationalData**, which models the status and characteristics of device communication (e.g., protocol, read/write capabilities);
- **Documentation**, aligned with the IDTA handover documentation template, to include external references and manuals [26].

Instances that follow this template inherit the defined Submodels and attributes from the *PhysicalDevice* type. This guarantees consistency across physical assets while allowing flexibility through the instance-level specification of values.

In the deployed scenario, the temperature sensors installed on each tank were modeled as distinct AAS instances. Each of them derives from the *TemperatureSensor* instance and is conformant to the *PhysicalDeviceTank*, as illustrated in Figure 5.2. Each sensor instance includes an additional Submodel named **MeasurementData**, which holds runtime-relevant Properties such as the current temperature value, the defined setpoint, and the hysteresis (acceptable variation). This data can be directly accessed or updated via *Delegated Operations* or externally linked endpoints.

Some AASs represent unique software components and therefore do not require a separate type. One such case is the **SpringShell**, the digital representation of a software component responsible for mediating communication between PLCs and AAS environments. The SpringShell AAS includes a Submodel with four Operations to start and stop read/write processes, a **SoftwareNameplate** Submodel from IDTA [27], similar to the DigitalNameplate but with software-specific Properties, and a **ConnectionStatus** to verify the state of the communication pipeline.

To represent the cultivated product, two AAS types were created: *Seaweed* and *SeaweedTank*. The *Seaweed* type captures the general biological characteristics of a species, using a **BiologicalInfo** Submodel that includes Properties such as species name, strain, taxonomy, and common name. Instances of this type, such as the AAS named *Ulva* presented in Figure 5.3, represent specific strains that are cultivated within the environment.

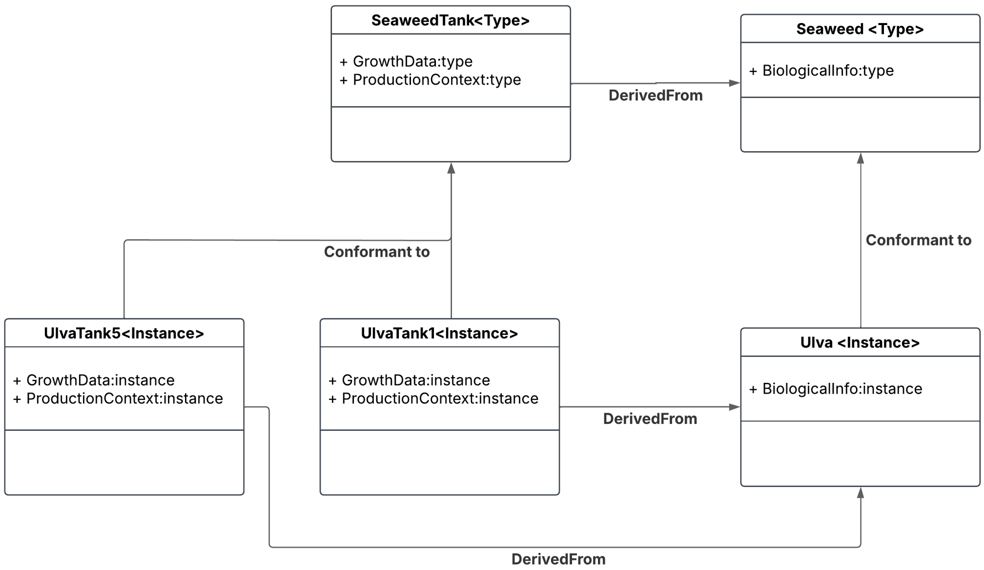

Figure 5.3: Seaweed AASs Types and Instances—UML Diagram

The *SeaweedTank* type extends the *Seaweed* AAS structure to include growth monitoring and batch production data. It contains two additional Submodels: **GrowthData**, which models biomass concentration, growth rate, cultivation phase, and timestamps; and **ProductionContext**, which tracks metadata about the cultivation process, such as batch ID, start time, expected harvest time, and a reference to the cultivation tank. Instances such as *UlvaTank1* and *UlvaTank5* derive from this type and represent specific cultivation batches of seaweed within Tank 1 and Tank 5, respectively. Each is linked to its corresponding tank via a *Reference Element*.

Tanks themselves are modeled as composite I4.0 components. The AAS type *Tank* structures information relevant to each cultivation tank. It includes Submodels for the tanks' specifications, documentation for clarity, and to represent the cultivation environment. The **CultivationEnvironment** Submodel is used to reference other AASs associated with the tank, including physical devices (sensors, actuators) and the cultivated seaweed batch. This allows the tank to act as a digital aggregator, dynamically representing its internal composition and environmental state. To finalize the ecosystem, two instances were created, *Tank1* and *Tank5*, that reference the corresponding temperature sensors and seaweed instances. This whole environment can be visualized in the UML Diagram on Appendix A.1.

5.3 Conclusions

To conclude, this chapter demonstrated the process of digitally replicating a real-world algae cultivation environment using AAS and the AASX Package Explorer framework. Building on the interoperable infrastructure explored in the previous chapter, this one translated the physical and software assets into structured, standardized, and semantically enriched DTs.

The approach adopted follows the principles outlined by Plattform Industrie 4.0 and related standardization bodies. Where applicable, predefined templates for Submodels such as Digital-Nameplate, TechnicalData, and Documentation were employed to ensure consistency and semantic alignment. In parallel, custom Submodels were developed to address domain-specific requirements, particularly in the context of seaweed cultivation and tank-based production management.

While many of the standardized Submodels offered a useful starting point, a significant portion of the representation—especially concerning algae biology and cultivation context—required custom modeling. This reflects a broader insight of the chapter: that the AAS framework is flexible enough to accommodate both highly standardized industrial devices and the unique features of emerging sectors such as algae-based aquaculture.

The AAS ecosystem created in this study acts as a fundamental layer for upcoming data interchange, monitoring, remote control, and optimization tasks within the larger framework of the background described in Chapter 3. It is hoped that this will serve as a starting point for upcoming initiatives in this field.

Chapter 6

Conclusions and Further Research

This dissertation set out to explore the challenges and opportunities of implementing scalable digital solutions in complex industrial environments, with a focus on algae cultivation systems. Motivated by the need for more actionable and reproducible industrial digitalization models, this work sought to bridge the gap between conceptual reference architectures and their actual deployment in real-world scenarios. As reviewed in the Problem Statement, although the literature on IIoT and Industry 4.0 concepts is increasingly mature, many studies fall short of providing modular, implementable, and context-aware solutions—particularly in domains where legacy infrastructure, multi-organizational collaboration, and heterogeneous technologies coexist.

6.1 Contributions

To explore how digital transformation could be achieved under real-world constraints, two distinct architectural strategies were proposed and implemented:

1. **Connector-Based Architecture** leveraged the Apache PLC4X and Kafka Connect ecosystem to facilitate modular and rapid integration of OPC UA data streams with a Kafka Message Broker within an IIoT Platform. A custom-developed Connector UI further enhanced usability by abstracting connector configuration into intuitive forms.

2. **AAS-Based Architecture** embraced the Asset Administration Shell framework as defined by Plattform Industrie 4.0, aiming to provide a semantically rich and interoperable digital representation of the algae cultivation environment. Using Eclipse BaSyx and a custom REST-based application, a robust infrastructure was established to enable bidirectional communication between physical assets and their digital twins. A complete environment was built to support AAS instantiation, Submodel templates, semantic enrichment via Concept Descriptions, and dynamic runtime data synchronization through MQTT-based eventing and time-series storage with InfluxDB.

Together, these implementations demonstrated key findings regarding the Research Questions:

- RQ1: ***What are the key requirements and constraints when implementing/integrating digital-industrial solutions into complex legacy systems in real-world environments?***

  A deep contextual understanding is required of both **operational constraints** and **organizational roles**. Integration must respect infrastructure limitations while promoting gradual scalability and interoperability.

- RQ2: ***Which state-of-the-art technologies offer the most scalable, modular, and interoperable solutions for such integration?***

  The study validates Kafka-connector ecosystems for rapid integration and the AAS framework for long-term, semantically robust architectures. Each presents distinct strengths, tailored to different phases and scales of digital transformation.

- RQ3: ***How can these technologies be effectively implemented, validated, and evaluated in a practical use case to ensure robustness, usability, and long-term adaptability?***

  Through the hands-on deployment of digital replicas via **AAS type and instance creation**, this study demonstrated a robust and structured approach to modeling **real-world assets** across **physical**, **software**, and **biological** domains. Runtime interaction with the AAS environment—via polling and event-based mechanisms—enabled synchronized control and monitoring. The integration of **semantic references**, **lifecycle data**, and **operational behavior** within each AAS allowed for validation not only at the system level but also in terms of usability and alignment with long-term maintenance and scalability.

As part of the research contributions, a scientific article based on the AAS-based integration architecture within the algae cultivation environment was developed and submitted during the dissertation process. The article, titled ***Enhancing Sustainability and Efficiency in Algae Cultivation Through IIoT and Asset Administration Shell***, was peer-reviewed, accepted, and presented at the ***Flexible Automation and Intelligent Manufacturing (FAIM2025)***. Feedback received from reviewers and discussions during the conference provided external validation of the AAS-based solution in production scenarios. The full article is included in Appendix B. This step not only enhanced the academic rigor of the work but also ensured alignment with current research discourse and practical relevance in the field of industrial digitalization.

Despite the advances made, this work has limitations. The AAS integration with the organization's IIoT Platform was not realized due to time constraints and the complexity of aligning semantic models with the existing Platform's schema. The operational deployment was confined to small environment validations, limiting operational stress evaluation.

6.2 Future Work

While this study tackled key integration challenges and successfully demonstrated two alternative architectures, it also opens several promising directions for future research and development.

On the connector-based side, the developed Connector UI could be extended into a broader **multi-protocol connector management platform**. This would support **rapid deployment and dynamic configuration** of source/sink connectors across various protocols, enriched with additional capabilities such as advanced logging, error handling, and real-time monitoring using tools like Grafana or Prometheus. Such enhancements would transform the solution into a more production-ready infrastructure, suitable for operational IIoT environments.

For the AAS-based approach, further work could focus on extending protocol support beyond the existing Eclipse BaSyx ecosystem, enabling broader bidirectional communication capabilities and alignment with other open-source AAS frameworks. Deeper integration with industrial platforms, particularly via **Kafka-based backends**, remains an important milestone within this project. In addition, co-developing and **standardizing new Submodel templates** in collaboration with industry-specific bodies could help fill current semantic and structural gaps, especially for **production planning**, **quality assurance**, or **biological process** monitoring.

From a broader perspective, this dissertation provides a methodological foundation for scaling digital transformation initiatives in traditional sectors. Its principles and tooling can be replicated or adapted to contexts such as smart farming, aquaculture, or decentralized energy systems.

In conclusion, this work contributes both technically and conceptually to the field of industrial digitalization. By bridging high-level frameworks with practical deployments, it offers a tangible reference for engineers, researchers, and industry professionals. As the transition to I4.0 accelerates, such grounded, replicable studies will be vital in translating vision into sustainable impact.

References

[1] T.A. Abdel-Aty, E. Negri, and S. Galparoli. "Asset Administration Shell in Manufacturing: Applications and Relationship with Digital Twin". In: *IFAC-PapersOnLine* 55 (2022). Issue: 10, pp. 2533–2538. DOI: 10.1016/j.ifacol.2022.10.090. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85144486380&doi=10.1016%2fj.ifacol.2022.10.090&partnerID=40&md5=feb91ff30f74ecf5491e7943d6a2be

[2] Apache Camel. *Apache Camel PLC4X Component*. https://camel.apache.org/components/4.10.x/plc4x-component.html. Accessed: 2025-06-11. 2022.

[3] Apache PLC4X. *Apache PLC4X - Siemens S7 Protocol Driver*. https://plc4x.apache.org/plc4x/latest/users/protocols/s7.html. Accessed: 2025-06-11. 2022.

[4] Apache PLC4X. *Apache PLC4X Kafka Integration*. Accessed: 2025-06-04. 2019. URL: https://plc4x.apache.org/plc4x/latest/users/integrations/apache-kafka.html.

[5] Apache PLC4X. *Apache PLC4X OPC UA Protocol Guide*. Accessed: 2025-06-04. 2019. URL: https://plc4x.apache.org/plc4x/latest/users/protocols/opcua.html.

[6] Apache Software Foundation. *Apache Kafka Documentation*. Accessed: 2025-06-04. 2011. URL: https://kafka.apache.org/documentation/.

[7] Rita Araújo et al. "Current status of the algae production industry in Europe: an emerging sector of the blue bioeconomy". In: *Frontiers in Marine Science* 7 (2021), p. 626389.

[8] M. Borth, J. Verriet, and G. Muller. "Digital twin strategies for SoS: 4 challenges and 4 architecture setups for digital twins of SoS". In: *2019 14th Annual Conference System of Systems Engineering, SoSE 2019*. 2019 14th Annual Conference System of Systems Engineering, SoSE 2019. 2019, pp. 164–169. DOI: 10.1109/SYSOSE.2019.8753860. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069768676&doi=10.1109%2fSYSOSE.2019.8753860&partnerID=40&md5=d698a8dccd6efc41eb6caf10744930a9.

[9] S. Cavalieri and M.G. Salafia. "A model for predictive maintenance based on asset administration shell". In: *Sensors (Switzerland)* 20.21 (2020), pp. 1–20. DOI: 10.3390/s20216028. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85093969254&doi=10.3390%2fs20216028&partnerID=40&md5=efaf4e84f7a41f4d7b0bd9b87aea419a.

[10] Salvatore Cavalieri and Marco Giuseppe Salafia. "Asset Administration Shell for PLC Representation Based on IEC 61131-3". In: *IEEE Access* 8 (2020). Cited by: 60; All Open Access, Gold Open Access, Green Open Access, pp. 142606–142621. DOI: 10.1109/ACCESS.2020.3013890. URL: https://www.scopus.com/inward/record.

uri?eid=2-s2.0-85090044142&doi=10.1109%2fACCESS.2020.3013890&partnerID=40&md5=0671dec55871a411e7b3531885e5fd94.

[11] Confluent, Inc. *Confluent Kafka Connect: Single Message Transforms*. Accessed: 2025-06-04. 2014. URL: https://docs.confluent.io/kafka-connectors/transforms/current/overview.html.

[12] eCl@ss. *eCl@ss – The Standard for Product and Service Classification*. Accessed: 2020-08-01. 2020. URL: http://www.eclass.eu/.

[13] Eclipse BaSyx. *AAS Web UI – Eclipse BaSyx Component*. https://wiki.basyx.org/en/latest/content/user_documentation/basyx_components/web_ui/index.html. Accessed: 2025-06-11. 2024.

[14] Eclipse BaSyx. *AAS Web UI – Eclipse BaSyx Component*. https://wiki.basyx.org/en/latest/content/user_documentation/basyx_components/v1/aas-server/features/operation-delegation.html. Accessed: 2025-06-11. 2024.

[15] Eclipse BaSyx. *Eclipse BaSyx Java SDK Operation Delegation Example*. https://github.com/eclipse-basyx/basyx-java-server-sdk/blob/main/examples/BaSyxOperationDelegation/exampleOperationService/src/main/java/org/eclipse/digitaltwin/basyx/examples/operationdelegation/Application.java. Accessed: 2025-06-11. 2024.

[16] Eclipse Foundation. *Eclipse BaSyx Starter Kit – Getting Started*. https://basyx.org/get-started/introduction. Accessed: 2025-06-11. 2024.

[17] Eclipse Foundation. *Eclipse BaSyx Wiki*. https://wiki.basyx.org/en/latest/. Accessed: 2025-06-11. 2024.

[18] Omar Ashraf ElFar et al. "Prospects of Industry 5.0 in algae: Customization of production and new advance technology for clean bioenergy generation". In: *Energy Conversion and Management: X* 10 (2021), p. 100048. ISSN: 2590-1745. DOI: https://doi.org/10.1016/j.ecmx.2020.100048. URL: https://www.sciencedirect.com/science/article/pii/S2590174520300209.

[19] M. Fabris et al. "Emerging Technologies in Algal Biotechnology: Toward the Establishment of a Sustainable, Algae-Based Bioeconomy". In: *Frontiers in Plant Science* 11 (2020). DOI: 10.3389/fpls.2020.00279. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85082687869&doi=10.3389%2ffpls.2020.00279&partnerID=40&md5=8f4c59fb8d3913b8a8f819a89af7188b.

[20] David Farley. *Modern Software Engineering: Doing What Works to Build Better Software Faster*. First Edition. Boston, MA: Addison-Wesley Professional, 2021. ISBN: 9780137314911.

[21] Fraunhofer IESE. *BaSys 4.0 – Basic System Industry 4.0*. https://www.iese.fraunhofer.de/en/customers_industries/digitalisierung-produktion/industrie40/basys-4-0-.html. Accessed: 2025-06-11. 2016.

[22] J. Gayathri et al. "Generating an IOT Based Knowedgebase to Analyze The Microalgae Growth". In: *Proceedings of the 2024 10th International Conference on Communication and Signal Processing, ICCSP 2024* (2024), pp. 104–108. DOI: 10.1109/ICCSP60870.2024.10543913. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85195998379&doi=10.1109%2fICCSP60870.2024.10543913&partnerID=40&md5=99d243d9fa53d055c3276c2efddde252.

[23] P. Gotovtsev. “How IoT can integrate biotechnological approaches for city applications-review of recent advancements, issues, and perspectives”. In: *Applied Sciences (Switzerland)* 10.11 (2020). DOI: 10.3390/app10113990. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85087035337&doi=10.3390%2fapp10113990&partnerID=40&md5=a05a4958717744a2fa784940f9b6f67f.

[24] Y. Huang, S. Dhouib, and J. Malenfant. “An AAS Modeling Tool for Capability-Based Engineering of Flexible Production Lines”. In: *IECON Proceedings (Industrial Electronics Conference)* 2021-October (2021). DOI: 10.1109/IECON48115.2021.9589329. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85119507003&doi=10.1109%2fIECON48115.2021.9589329&partnerID=40&md5=6350c3c7f9673ab7ba6c54e4dbf162e6.

[25] IDTA. *AASX Package Explorer*. https://github.com/admin-shell-io/aasx-package-explorer. Accessed: 2025-06-15. 2020.

[26] IDTA. *Handover Documentation (Version 1.2)*. https://github.com/admin-shell-io/submodel-templates/tree/main/published/Handover%20Documentation/1/2. Accessed: 2025-06-15. 2023.

[27] IDTA. *Nameplate for Software in Manufacturing (Version 1.0)*. https://github.com/admin-shell-io/submodel-templates/tree/main/published/Software%20Nameplate/1/0. Accessed: 2025-06-15. 2023.

[28] IEC-61131. *Programmable controllers - Part 1: General information*. en. International Standard. May 2003. DOI: 9782889128877. URL: https://webstore.iec.ch/en/publication/4550.

[29] IEC-61360. *Standard data element types with associated classification scheme*. en. International Standard. July 2017. DOI: 9782832245811. URL: https://webstore.iec.ch/en/publication/28560.

[30] IEC-62264. *Enterprise-control system integration Part 1: Models and terminology*. en. International Standard. May 2013. DOI: 9782832208335. URL: https://webstore.iec.ch/en/publication/6676.

[31] IEC-62541. *OPC Unified Architecture - Part 1: Overview and concepts*. en. International Standard. Nov. 2020. DOI: 9782832290767. URL: https://webstore.iec.ch/en/publication/61109.

[32] IEC-63278. *Asset Administration Shell for Industrial Applications - Part 1: Asset Administration Shell Structure*. en. International Standard. Dec. 2023. DOI: 9782832276792. URL: https://webstore.iec.ch/en/publication/65628.

[33] Plattform Industrie 4.0. *Details of the Asset Administration Shell - Part 1*. en. May 2022. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf?__blob=publicationFile&v=1 (visited on 05/23/2025).

[34] Plattform Industrie 4.0. “Network-based Communication for Industrie 4.0: Proposal for an Administration Shell”. en. In: *Federal Ministry for Economic Affairs and Energy* (Nov. 2016). URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/network-based-communication-for-i40-it-summit.pdf?__blob=publicationFile&v=1.

[35] Plattform Industrie 4.0. *RAMI4.0 – a reference framework for digitalisation*. en. 2018. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf?__blob=publicationFile&v=7.

[36] Plattform Industrie 4.0. *Structure of the Administration Shell Continuation of the Development of the Reference Model for the Industrie 4.0 Component*. Apr. 2016. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.pdf?__blob=publicationFile&v=1.

[37] Proaction International. *Industry 5.0: The Next Industrial Revolution is People-Centric*. Accessed: 2024-10-03. 2021. URL: https://blog.proactioninternational.com/en/industry-5.0-the-next-industrial-revolution-is-people-centric.

[38] M. Jacoby et al. "An approach for realizing hybrid digital twins using asset administration shells and apache streampipes". In: *Information (Switzerland)* 12.6 (2021). DOI: 10.3390/info12060217. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107233310&doi=10.3390%2finfo12060217&partnerID=40&md5=2551b060d6dce2abd89cc1950ac8ce9e.

[39] M. Javaid et al. "An integrated outlook of Cyber–Physical Systems for Industry 4.0: Topical practices, architecture, and applications". In: *Green Technologies and Sustainability* 1.1 (2023). DOI: 10.1016/j.grets.2022.100001. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85204470386&doi=10.1016%2fj.grets.2022.100001&partnerID=40&md5=f952b3122a7469ad4fbc510fe2a7d396.

[40] V. Kamath, J. Morgan, and M.I. Ali. "Industrial IoT and Digital Twins for a Smart Factory : An open source toolkit for application design and benchmarking". In: *GIoTS 2020 - Global Internet of Things Summit, Proceedings* (2020). DOI: 10.1109/GIOTS49054.2020.9119497. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85087431079&doi=10.1109%2fGIOTS49054.2020.9119497&partnerID=40&md5=6f3bc651b400739066558083706e3352.

[41] M.J. Kaur, V.P. Mishra, and P. Maheshwari. "The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action". In: *Internet of Things*. Springer Nature Switzerland AG 2020, 2020, pp. 3–17. DOI: 10.1007/978-3-030-18732-3_1. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85081533676&doi=10.1007%2f978-3-030-18732-3_1&partnerID=40&md5=dac7600db5b56aa547a81469039cacde.

[42] J.G. Khor et al. "Automated Cultivation System for Microalgae: Growth Factors and Control". In: *Current Nutrition and Food Science* 18.9 (2022), pp. 776–779. DOI: 10.2174/1573401318666220421132428. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85137459448&doi=10.2174%2f1573401318666220421132428&partnerID=40&md5=3db82fdcb648210e4887c1748483bf32.

[43] A. Kunapinun et al. "Advancing Seaweed Cultivation: Integrating Physics Constraint in Machine Learning for Enhanced Biomass Forecasting in IMTA Systems". In: *Remote Sensing* 16.23 (2024). DOI: 10.3390/rs16234418. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85211950048&doi=10.3390%2frs16234418&partnerID=40&md5=65618090da9e7a50281f108cc27d8788.

[44] H.R. Lim et al. "Evaluation of Real-Time Monitoring on the Growth of Spirulina Microalgae: Internet of Things and Microalgae Technologies". In: *IEEE Internet of Things Journal* 11.2 (2024), pp. 3274–3281. DOI: 10.1109/JIOT.2023.3296525. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85165244075&doi=10.1109%2fJIOT.2023.3296525&partnerID=40&md5=0f43dabef71a49190326e7c5c58cd436.

[45] R.D.S. Mendonça et al. "Digital Twin Applications: A Survey of Recent Advances and Challenges". In: *Processes* 10.4 (2022). DOI: 10.3390/pr10040744. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85129045758&doi=10.3390%2fpr10040744&partnerID=40&md5=d6008352656fed53be2b09b925b189ee.

[46] Khadija El-Moustaqim et al. "Enabling Smart Agriculture Through Integrating the Internet of Things in Microalgae Farming for Sustainability". en. In: *Smart Internet of Things for Environment and Healthcare*. Ed. by Mourade Azrour et al. Cham: Springer Nature Switzerland, 2024, pp. 209–222. ISBN: 978-3-031-70102-3. DOI: 10.1007/978-3-031-70102-3_15. URL: https://doi.org/10.1007/978-3-031-70102-3_15 (visited on 11/30/2024).

[47] M. Munoz et al. "An IoT Platform for Data Management in an Industrial-Scale Microalgae Cultivation Plant". In: *IEEE Access* 10 (2022), pp. 127128–127139. DOI: 10.1109/ACCESS.2022.3226334. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85144021356&doi=10.1109%2fACCESS.2022.3226334&partnerID=40&md5=a2e12b001c5f03e5fd4458a78cbaccaf.

[48] Y.T. Neo et al. "Smart systems in producing algae-based protein to improve functional food ingredients industries". In: *Food Research International* 165 (2023). DOI: 10.1016/j.foodres.2023.112480. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85146722753&doi=10.1016%2fj.foodres.2023.112480&partnerID=40&md5=e2e0644fcde76667b7b63386f4c734db.

[49] E.G. Nwoba et al. "Process control strategies applied to microalgae-based biofuel production". In: *3rd Generation Biofuels: Disruptive Technologies to Enable Commercial Production*. 2022 Elsevier Inc., 2022, pp. 105–134. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85138072393&doi=10.1016%2fB978-0-323-90971-6.00031-0&partnerID=40&md5=3c810c42cf9ec1754af98aa74a2e6552.

[50] OPC Foundation. *OPC Unified Architecture (OPC UA)*. Accessed: 2025-05-23. 2024. URL: https://opcfoundation.org/about/opc-technologies/opc-ua/.

[51] R.K. Oruganti et al. "Artificial intelligence and machine learning tools for high-performance microalgal wastewater treatment and algal biorefinery: A critical review". In: *Science of the Total Environment* 876 (2023). DOI: 10.1016/j.scitotenv.2023.162797. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85149854857&doi=10.1016%2fj.scitotenv.2023.162797&partnerID=40&md5=06d4250db58b903e68629176ef233100.

[52] Y. Park, J. Woo, and S. Choi. "A Cloud-based Digital Twin Manufacturing System based on an Interoperable Data Schema for Smart Manufacturing". In: *International Journal of Computer Integrated Manufacturing* 33.12 (2020), pp. 1259–1276. DOI: 10.1080/0951192X.2020.1815850. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091011080&doi=10.1080%2f0951192X.2020.1815850&partnerID=40&md5=87ab6c8da6b3f4a787478187f87fdcbc.

[53] W. Quadrini et al. “Asset Administration Shell as an interoperable enabler of Industry 4.0 software architectures: a case study”. In: *Procedia Computer Science* 217 (2022), pp. 1794–1802. DOI: 10.1016/j.procs.2022.12.379. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85163844912&doi=10.1016%2fj.procs.2022.12.379&partnerID=40&md5=9f0dad0b2f565312352f44ea02f5f954.

[54] A.J.H. Redelinghuys, A.H. Basson, and K. Kruger. “A six-layer architecture for the digital twin: a manufacturing case study implementation”. In: *Journal of Intelligent Manufacturing* 31.6 (2020), pp. 1383–1402. DOI: 10.1007/s10845-019-01516-6. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85076271545&doi=10.1007%2fs10845-019-01516-6&partnerID=40&md5=c04eafdfaad710f209949342d25514bc.

[55] A.J.H. Redelinghuys, K. Kruger, and A. Basson. “A six-layer architecture for digital twins with aggregation”. In: *Studies in Computational Intelligence*. Vol. 853. Studies in Computational Intelligence, 2020, pp. 171–182. DOI: 10.1007/978-3-030-27477-1_13. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85070603892&doi=10.1007%2f978-3-030-27477-1_13&partnerID=40&md5=d6781508b6f788499dc6602bbe191326.

[56] A. Salam. “Internet of things in agricultural innovation and security”. In: *Internet of Things*. Springer Nature Switzerland AG 2020, 2020, pp. 71–112. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077448041&doi=10.1007%2f978-3-030-35291-2_3&partnerID=40&md5=7b5590c6f1a67f250b58b77902e3b518.

[57] C.I. Samuel and D. Favitri. “IoT-based of Seaweed Cultivation Employing a Low-cost Autonomous Experimental Buoy”. In: *Proceeding of 15th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2021* (2021). DOI: 10.1109/TSSA52866.2021.9768270. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85130582862&doi=10.1109%2fTSSA52866.2021.9768270&partnerID=40&md5=8ef65de9d26bb04083fa25e9d433ba25.

[58] G.N. Schroeder et al. “Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange”. In: *IFAC-PapersOnLine* 49 (2016). Issue: 30, pp. 12–17. DOI: 10.1016/j.ifacol.2016.11.115. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85006391498&doi=10.1016%2fj.ifacol.2016.11.115&partnerID=40&md5=307c6612411e9e0c79343c8c66d04126.

[59] F. Tao et al. “Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison”. In: *Engineering* 5.4 (2019), pp. 653–661. DOI: 10.1016/j.eng.2019.01.014. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068798049&doi=10.1016%2fj.eng.2019.01.014&partnerID=40&md5=ec57c7be2349b8ebd06ed854c6c1b94b.

[60] S.Y. Teng et al. “Recent advances on industrial data-driven energy savings: Digital twins and infrastructures”. In: *Renewable and Sustainable Energy Reviews* 135 (2021). DOI: 10.1016/j.rser.2020.110208. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85089574345&doi=10.1016%2fj.rser.2020.110208&partnerID=40&md5=f0b0f56cc153c0bf23384bf52c88ef30.

[61] N. Le-Thi-Thu, T. Nguyen-Van-Duong, and T. Doan-Van. “Combining Artificial Intelligence with the Internet of Things in Microalgae Cultivation”. In: *Lecture Notes on Data Engineering and Communications Technologies*. Vol. 148. Springer Nature Switzerland AG 2022, 2022, pp. 156–166. DOI: 10.1007/978-3-031-15063-0_14. URL: https://

www.scopus.com/inward/record.uri?eid=2-s2.0-85136802641&doi=10.1007%2f978-3-031-15063-0_14&partnerID=40&md5=3b3eb5520662a5f60a1874583cc9d7

[62] T. Tummawai et al. "Application of Artificial Intelligence and Image Processing for the Cultivation of Chlorella sp. Using Tubular Photobioreactors". In: *ACS Omega* (2024). DOI: 10.1021/acsomega.4c05971. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85209366373&doi=10.1021%2facsomega.4c05971&partnerID=40&md5=3b33736b6a13a7c01239e8fdfa7492fa.

[63] Unified Automation. *UA Expert – OPC UA Client from Unified Automation, V1.7.2*. Accessed: 2025-06-04. 2025. URL: https://documentation.unified-automation.com/uaexpert/1.7.2/html/index.html.

[64] Mauro Vigani et al. "Food and feed products from micro-algae: Market opportunities and challenges for the EU". In: *Trends in Food Science & Technology* 42.1 (2015), pp. 81–92.

[65] Kexin Wang et al. "How does the Internet of Things (IoT) help in microalgae biorefinery?" In: *Biotechnology Advances* 54 (Jan. 2022), p. 107819. ISSN: 0734-9750. DOI: 10.1016/j.biotechadv.2021.107819. URL: https://www.sciencedirect.com/science/article/pii/S0734975021001257 (visited on 11/15/2024).

[66] C.-T. Yang et al. "Machine learning in microalgae biotechnology for sustainable biofuel production: Advancements, applications, and prospects". In: *Bioresource Technology* 413 (2024). DOI: 10.1016/j.biortech.2024.131549. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85205318714&doi=10.1016%2fj.biortech.2024.131549&partnerID=40&md5=05600c40e7640424845effff2886ee52.

[67] X. Ye et al. "An Industry 4.0 Asset Administration Shell-Enabled Digital Solution for Robot-Based Manufacturing Systems". In: *IEEE Access* 9 (2021), pp. 154448–154459. DOI: 10.1109/ACCESS.2021.3128580. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85120499823&doi=10.1109%2fACCESS.2021.3128580&partnerID=40&md5=2ed388f4c8c544e4b36fd1d3db875bb4.

[68] S. Zeb et al. "Industrial digital twins at the nexus of NextG wireless networks and computational intelligence: A survey". In: *Journal of Network and Computer Applications* 200 (2022). DOI: 10.1016/j.jnca.2021.103309. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85122979494&doi=10.1016%2fj.jnca.2021.103309&partnerID=40&md5=8973ac26a5216a1dd77c7df94be6d3b9.

[69] 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: 2025-06-15. 2020.

[70] 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: 2025-06-15. 2020.

Appendix A

A1. UML Diagram of the Cultivation Environment

This appendix presents the complete UML diagram of the algae cultivation environment as digitally modeled using the Asset Administration Shell (AAS) framework. The diagrams detail the type-instance hierarchy, structural relationships, and submodel compositions for all core components, including physical devices (e.g., sensors, actuators), composite assets (e.g., tanks), and biological entities (e.g., seaweed species and batches).

Figures A.1 and A.2 illustrate the digital twin ecosystem's modular structure, highlighting how individual AASs are linked through semantic references and conform to predefined templates. These visualizations support the architecture described in Chapter 5 and emphasize the scalability and interoperability of the proposed solution.

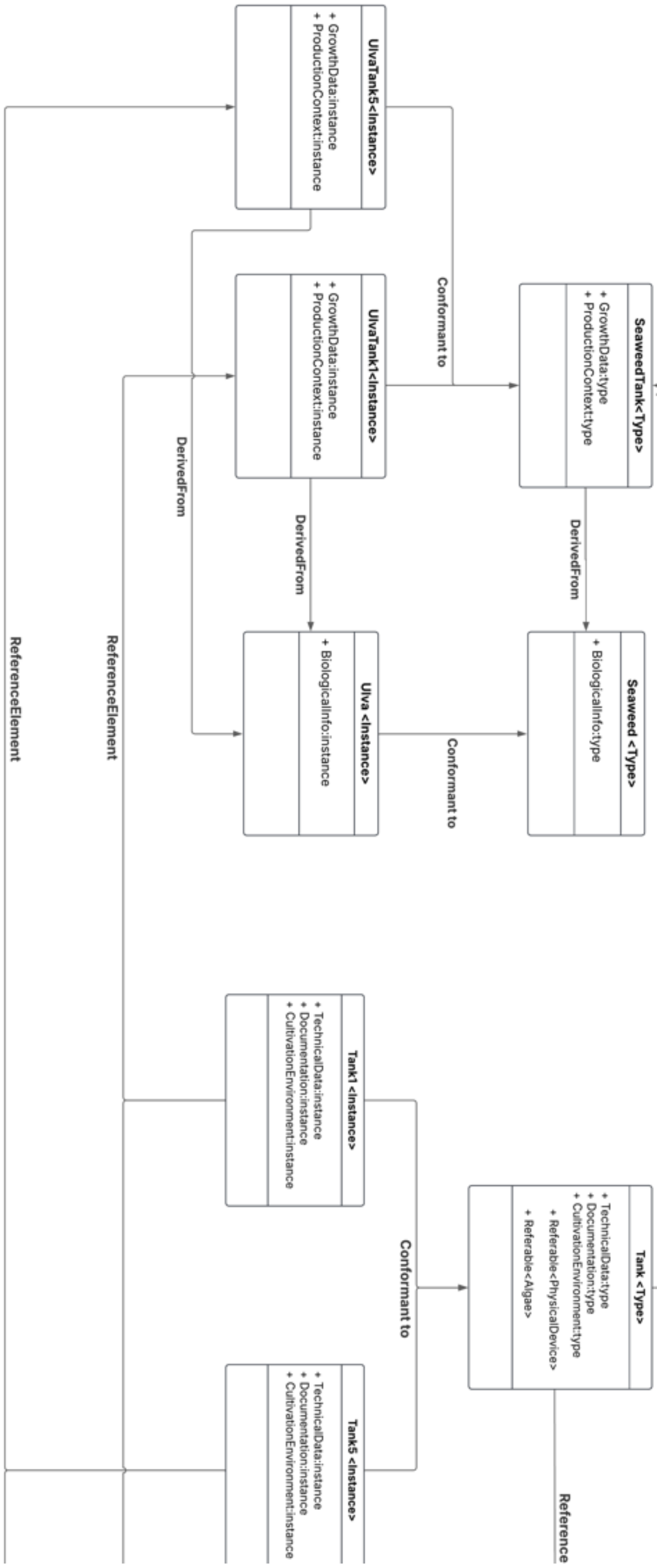

Figure A.1: UML Diagram Part 1 of the Cultivation Environment

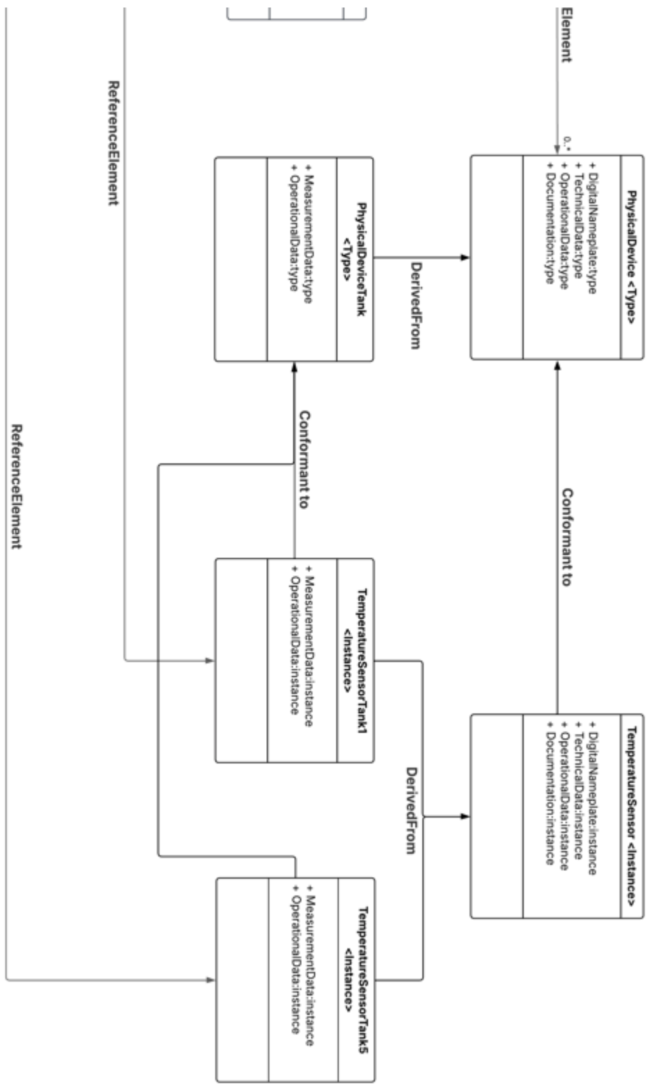

Figure A.2: UML Diagram Part 2 of the Cultivation Environment

Appendix B

A2. Research Paper—FAIM2025

The following pages exhibit a research paper entitled ***"Enhancing Sustainability and Efficiency in Algae Cultivation Through IIoT and Asset Administration Shell"***, elaborated and accepted for ***The 34th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2025)***.
Status: Accepted & Presented[1]

[1] Accepted and presented at the FAIM2025 conference that occurred on 21-24 of June in New York.

Enhancing Sustainability and Efficiency in Algae Cultivation Through IIoT and Asset Administration Shell

André Costa, Filipe Ferreira, and António Lucas Soares

Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência, Porto
Faculdade de Engenharia da Universidade do Porto, Porto
andre.c.costa@inesctec.pt, filipe.d.ferreira@inesctec.pt,
antonio.l.soares@inesctec.pt

Abstract. Algae cultivation is increasingly recognized for its potential in sustainable production due to algae's ability to produce biomass efficiently with a low environmental footprint. However, optimizing algae cultivation operations presents significant challenges, particularly in maintaining precise environmental conditions and minimizing resource use. Traditional methods often lack the adaptability to respond swiftly to changing environmental variables, leading to operational inefficiencies and resource waste. This article explores a multi-layered Industrial digital solution that integrates Asset Administration Shell technology with Internet of Things solutions to address and analyze these challenges and improve system resilience and operational outcomes in macroalgae production. The implementation includes an interconnected field solution based on sensor and actuator networks, edge processing, and integration of industrial communication protocols, which are deployed to tackle distinct operational needs within the algae production environment. Additionally, by leveraging Digital Twins implementation, data persistence with a MongoDB, and dashboard visualization through Grafana, this approach provides the field operators with near real-time monitoring, predictive adjustments, and optimized process management, directly contributing to scalable production. It is hypothesized that this integrated approach will improve cultivation performance and operational efficiency, demonstrating the potential of advanced industrial solutions to support the future of sustainable algae production systems.

Keywords: Industrial Internet of Things, Asset Administration Shell, Algae Cultivation, Sustainability, Operational Efficiency

1 Introduction

As industrial revolutions and smart manufacturing continue to evolve, companies are increasingly investing in their initiatives to optimize production and to support decision-making and environmental sustainability [9]. Cyber-Physical Production System (CPPS) and Industrial Internet of Things (IIoT) solutions are being adopted across various sectors to achieve these goals. Nevertheless, there are areas that remain underexplored and should be studied for the aforesaid integration, such as algae cultivation [3].

Algae, encompassing both micro-algae and macroalgae, has gained increasing importance due to its potential in a variety of high-value applications [7]. In the food industry, algae are used as functional ingredients, while they also hold promise in bioenergy production, such as biofuels, and carbon capture efforts [7]. Despite the substantial opportunities algae offer, their cultivation, especially in Europe, remains underdeveloped compared to more advanced regions like Asia, signaling a pressing need for innovative, integrated solutions to optimize production and environmental monitoring [15].

The production of macroalgae, in particular, presents unique challenges. Unlike conventional industrial processes, the cultivation is highly dynamic, influenced by continuous environmental changes and interdependent parameters. Traditional manual cultivation methods require constant supervision and periodic sampling, making it difficult to accurately monitor and control key variables [16]. The lack of automated monitoring and adaptive control mechanisms introduces inefficiencies, limiting the scalability and sustainability of production.

This study proposes a digital manufacturing solution that leverages state-of-the-art industrial technologies to address these challenges. The proposed architecture is structured into two levels: **Operational Technology** (OT) and **Information Technology** (IT). The OT level consists of a network of field devices connected to a Central Processing Unit (CPU), enabling edge computing for low-latency processing and real-time control of cultivation parameters.

The primary focus of this study is the IT level, specifically the implementation and integration of **Asset Administration Shell** (AAS), a standardized approach to Digital Twins (DTs) for asset representation and management [1]. This research explores how AAS can enhance interoperability, automation, and decision-making within algae cultivation environments. The study further details the architecture and development of the AAS framework, demonstrating its integration with other digital components and highlighting the benefits of standardized digital replica implementations in industrial ecosystems.

While technical implementation is a core aspect, the overarching objective remains the optimization of algae production in a local facility, guiding architectural and development decisions throughout the study. The core architecture and communication layers have been validated, confirming seamless data exchange between OT and IT components. However, further work is required to strengthen system robustness, integrate real-world environmental data, and refine the system using historical analysis and machine learning techniques.

The remainder of this paper is structured as follows: Section 2 reviews the existing research on algae cultivation and AAS applications. Section 3 discusses the research approach and methodology to mitigate the identified challenges. Section 4 details the proposed solution, architectural decisions, and implementation aspects. Finally, Section 5 discusses the study's contributions, identifies areas requiring further refinement, and outlines directions for future enhancements.

2 Related Work

To align with the focus of this study, this section reviews existing digital approaches in algae cultivation, highlighting the advancements and existing limitations in the field, and explores the concept of Asset Administration Shell and its role in industrial settings.

2.1 Digital Technologies in Algae Cultivation

Despite the growing interest in algae production, the integration of high technology, autonomous processes, and intelligent manufacturing remains in its infancy [6][11]. The complexity of environmental conditions and the demands for real-time monitoring present significant challenges to manual methods, making autonomous data-driven solutions crucial [16]. IoT technologies and Machine Learning (ML) have emerged as key enablers to address these challenges.

IoT plays a pivotal role in environmental monitoring, ensuring that key physical, chemical, and biological parameters crucial for algae growth are maintained [13]. Studies suggest that sensor networks and automated control systems could significantly improve cultivation efficiency, reducing the dependency on manual intervention. A study by Nwoba et al. [12] provides a comprehensive breakdown of essential variables, categorizing them into physical (e.g., light, temperature, flow rate), chemical (e.g., pH, nutrients), and biological (e.g., cell density, growth rates) parameters. Their work offers a reference model for integrating IoT-based monitoring systems in algae farming.

Beyond IoT, ML techniques are increasingly being explored to optimize algae growth. Studies show that combining IoT with ML enables predictive analytics and adaptive control, improving algae yield and sustainability [6]. Notably, Yang et al. [16] provide an extensive review of ML algorithms applicable to algae cultivation, offering insights into their strengths and use cases for optimizing production efficiency, resource utilization, and environmental adaptation.

Despite growing research, digital innovations have predominantly focused on microalgae due to their rapid growth and scalability [7]. Consequently, macroalgal farming has yet to benefit from equally robust digital frameworks that offer detailed implementation strategies. Addressing this gap, this study explores how digital technologies, in addition to IoT and ML, particularly AAS, can integrate diverse industrial innovations to improve large-scale algae production.

2.2 Asset Administration Shell

The Asset Administration Shell concept, introduced as part of the Industry 4.0 Reference Architecture model (RAMI4.0), serves as a standardized digital representation of physical and virtual assets [1]. As a DT, it acts as the asset's virtual counterpart, encompassing all relevant information throughout its entire lifecycle [2]. The AAS organizes this information hierarchically through submodels, where each submodel encapsulates specific characteristics or functionalities of the asset, ensuring structured and modular data representation [1].

Beyond asset representation, the AAS serves as an interoperability framework, providing a standardized method for structuring and exchanging asset-related data [4]. Schäfer et al. [14] emphasize its role in facilitating seamless integration across heterogeneous systems, allowing assets from different manufacturers or software ecosystems to communicate through a unified digital model.

This study focuses on leveraging the interoperability capabilities of AAS within an IIoT environment to optimize algae cultivation processes. The implementation adopts Eclipse BaSyx solution, enabling AAS instance management, submodel creation, and cross-platform integration, ensuring not only digital asset representation and real-time control but also scalability and long-term adaptability for future technological advancements in sustainable algae production.

3 Methodology

The methodology of this study follows a structured, iterative approach, ensuring a robust, scalable digital infrastructure for algae cultivation. The key phases are:

1. **Requirements Elicitation:** This phase involved on-site observations and discussions with aquaculture stakeholders to identify critical operational needs. This included defining relevant sensor data (e.g., pH, temperature, nutrient levels), understanding communication limitations between devices, and identifying control dependencies between variables. These insights guided the specification of system functionalities and performance expectations.
2. **Architectural Design and Component Selection:** Architectural decisions were guided by the need for modularity, scalability, and interoperability. Siemens PLC was selected for its industrial-grade reliability and native support for multiple protocols. Data processing between OT & IT is based on AAS and PLC connection, data structuring and storage through AAS, and monitoring tools enhancing the environment visualization and control.
3. **Experimentation & Validation:** An iterative process is conducted to refine the integration of OT, AAS, and IIoT infrastructure. Mock data are used to test the system's interoperability. This step validates the data communication, consistency, and system functionality before deployment and under safe conditions. Through continuous testing and refinement, this methodology ensures a robust and reliable digital solution before full-scale deployment.

4 Industrial AAS-based Digital Solution

This section provides a detailed examination of the overall architecture slightly covered in Section 1. It highlights the integration of the Eclipse BaSyx AAS implementation with the OT level and the broader IIoT infrastructure.

4.1 Operational Technology Overview

A clear understanding of the OT level is key to grasping the system's architecture. The OT infrastructure consists of a network of edge devices deployed across the algae cultivation facility, spanning multiple production containers. These field assets, including sensors and actuators, are connected via two communication methods to a Programmable Logic Controller (PLC) S7-1200, which commonly acts as the CPU for operational control and edge computing [8].

Most devices interface with the PLC through a Fieldbus network, with Fieldbus Couplers assigned to each container for localized data acquisition and processing. Additionally, certain devices utilize Bluetooth communication, linking to a local edge node that hosts an MQTT broker and a lightweight AAS implementation. This auxiliary node facilitates ***soft*** **real-time** - with minimal latency - data streaming, enhancing connectivity for wireless devices.

These devices are coupled with microcontrollers that publish sensor readings to MQTT topics at predefined intervals, following the Publish and Subscribe communication method adopted by MQTT. These messages are received by a MQTT broker running on the local edge node, where an AAS instance listens and updates the digital replicas of field assets upon new messages. The remaining communication between the AAS and the PLC follows the same pattern as the interaction between the Fog node and the OT level, discussed in Subsection 4.2.

The PLC maps each physical asset to a dedicated memory block, Data Block, and executes a closed-loop control to autonomously regulate production parameters. This control loop dynamically adjusts the system by analyzing real-time variable values against predefined set points, ensuring optimal cultivation conditions. Actuators such as water and air valves are adjusted based on system priorities and environmental demands. Operators can manually modify set points using either a Human-Machine Interface (HMI), local interfaces, or through digital replicas within the AAS, which enable remote adjustments.

4.2 AAS & OT Integration

The integration of AAS into the digital solution for algae cultivation serves multiple purposes, with interoperability being the most crucial, as previously highlighted in Sections 1 and 2. Eclipse BaSyx plays a key role as a middleware component [14], facilitating seamless data exchange between OT and IT levels of the architecture. This integration is visually depicted in Figure 1.

The AAS environment is deployed on an intermediary node at the **Fog level**, positioning the digital replicas at the heart of the IT infrastructure. This Fog node is responsible for managing the AAS representation of the aquaculture environment, ensuring near real-time synchronization between physical and virtual assets. Additionally, it acts as a control interface by establishing bidirectional communication with the PLC, allowing both monitoring and actuation.

To enable this interaction, a **Spring Boot application** was developed to create a ***RESTful API***, implementing key Read/Write functionalities for exchanging data with the PLC's Data Blocks. This API allows external components to retrieve real-time sensor data from the PLC or modify system parameters, contributing to enhanced automation and remote management.

The AAS communication pipeline, encompassing Spring API, is built using Apache Camel, an open-source integration framework, combined with a Siemens

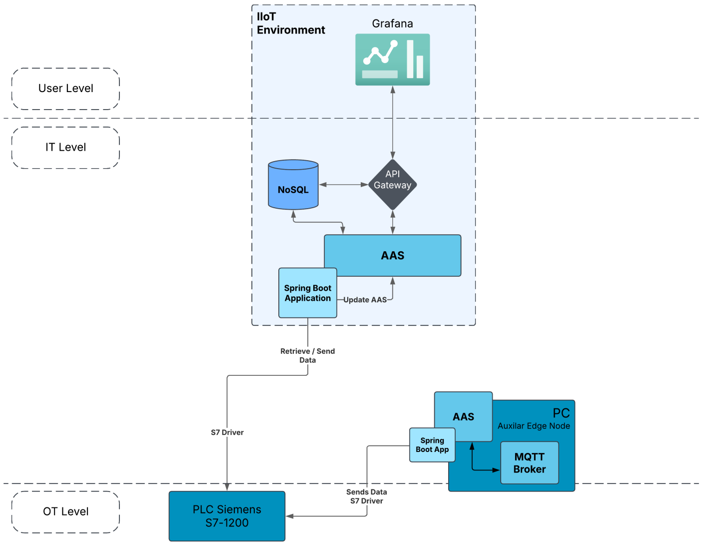

Fig. 1. OT and IT Integration

S7-Driver to establish direct interaction with the PLC and the built-in REST APIs provided by Eclipse BaSyx for AAS management. The system's REST API requests are classified into read/write operations, requiring three key parameters:

Submodel: Defines the set of SubmodelElements to be accessed.
Property: Specifies the exact parameter to be read or modified.
Data Block: Identifies the PLC memory location for the requested operation.

These operations can be invoked through multiple channels; however, BaSyx's **AAS Web UI** [5] provides the most intuitive approach, leveraging AAS's structure to facilitate standardized interaction. This ensures compatibility across industrial settings while maintaining a high level of automation and data integrity.

4.3 AAS & IIoT Platform Integration

The IIoT infrastructure in this digital solution is primarily supported by the Asset Administration Shell development. Leveraging its hierarchical structure to enhance data organization, interoperability provided by Eclipse BaSyx implementations, and connectivity of OT and IT components.

To support long-term data storage and analytics, an in-memory AAS storage [10] is coupled with a NoSQL MongoDB, ensuring the preservation of operational data for historical and real-time analysis. Grafana is employed as the monitoring tool, enabling visualization of key performance indicators (KPIs) and facilitating trend identification, anomaly detection, and predictive optimization. This combination of IoT tools, along with ML-based analytics, as mentioned in Section 2, enables operators to refine setpoints and improve sustainable production.

As shown in Figure 1, Grafana operates at the user level, requiring a dedicated access point to communicate with the core IIoT infrastructure. A centralized API Gateway facilitates this interaction, enabling real-time data retrieval and seamless integration with digital replicas. This approach ensures long-term scalability and sustainability, supporting future adoption of external services.

The technical implementation outlined in this section connects the OT field operations, AAS-based middleware, and IIoT infrastructure into a cohesive digital solution. Each component, from localized control and data collection to AAS-driven interoperability and finally to real-time monitoring and data analytics, has been integrated and tested with mock data and a controlled environment.

5 Conclusion & Future Work

This study presents the ongoing development of an AAS-based digital solution for algae cultivation, integrating IIoT technologies to enhance operational efficiency and sustainability. The foundational architecture, communication mechanisms, and IIoT structuring have been established and validated. The integration of OT components, AAS, and IIoT infrastructure has been tested using mock data, ensuring seamless data flow and interoperability across system layers.

Nonetheless, additional improvements are needed to study system behavior in real-world conditions. Future efforts will focus on reinforcing robust control logic in OT development, exploring historical data analytics and ML algorithms to optimize production using real algae cultivation data, and scrutinize safety data concerns. These efforts will enhance system adaptability, enabling dynamic optimization of cultivation parameters in response to environmental variations.

Beyond system development, further research is needed to explore scalable AAS frameworks tailored to biological production processes and the long-term impact of digitalization on the efficiency of macroalgal farming.

6 Acknowledgements

This publication has received funding from the European Union's Horizon Europe research and innovation action through the project INNOAQUA (Grant Agreement No. 101084383). The authors extend their gratitude to INESC TEC for their valuable support and contributions to this work.

References

1. 4.0, P.I.: The asset administration shell: Implementing digital twins for use in industrie 4.0 (2019). URL https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/VWSiD%20V2.0.html. Accessed: 2025-04-12
2. Abdel-Aty, T., Negri, E., Galparoli, S.: Asset Administration Shell in Manufacturing: Applications and Relationship with Digital Twin. In: 10th IFAC Conference on Manufacturing Modelling, Management and Control, MIM 2022, vol. 55, pp. 2533–2538 (2022). Issue: 10

3. Araújo, R., Vázquez Calderón, F., Sánchez López, J., Azevedo, I.C., Bruhn, A., Fluch, S., Garcia Tasende, M., Ghaderiardakani, F., Ilmjärv, T., Laurans, M., et al.: Current status of the algae production industry in europe: an emerging sector of the blue bioeconomy. Frontiers in Marine Science **7**, 626,389 (2021)
4. Cavalieri, S., Salafia, M.: A model for predictive maintenance based on asset administration shell. Sensors (Switzerland) **20**(21), 1–20 (2020)
5. Eclipse BaSyx: BaSyx AAS Web UI Documentation (2025). URL https://wiki.basyx.org/en/latest/content/user_documentation/basyx_components/web_ui/index.html. Accessed: 2025-02-27
6. El-Moustaqim, K., Mabrouki, J., Azrour, M., Hadine, M., Hmouni, D.: Enabling Smart Agriculture Through Integrating the Internet of Things in Microalgae Farming for Sustainability. In: M. Azrour, J. Mabrouki, A. Alabdulatif, A. Guezzaz, F. Amounas (eds.) Smart Internet of Things for Environment and Healthcare, pp. 209–222. Springer Nature Switzerland, Cham (2024)
7. ElFar, O.A., Chang, C.K., Leong, H.Y., Peter, A.P., Chew, K.W., Show, P.L.: Prospects of industry 5.0 in algae: Customization of production and new advance technology for clean bioenergy generation. Energy Conversion and Management: X **10**, 100,048 (2021)
8. Folgado, F.J., Calderón, D., González, I., Calderón, A.J.: Review of industry 4.0 from the perspective of automation and supervision systems: Definitions, architectures and recent trends. Electronics (Switzerland) **13**(4) (2024). Cited by: 53; All Open Access, Gold Open Access
9. International, P.: Industry 5.0: The next industrial revolution is people-centric (2021). URL https://blog.proactioninternational.com/en/industry-5.0-the-next-industrial-revolution-is-people-centric. Accessed: 2025-01-11
10. Jacoby, M., Jovicic, B., Stojanovic, L., Stojanović, N.: An approach for realizing hybrid digital twins using asset administration shells and apache streampipes. Information (Switzerland) **12**(6) (2021)
11. Lim, H., Khoo, K., Chew, K., Teo, M., Ling, T., Alharthi, S., Alsanie, W., Show, P.: Evaluation of Real-Time Monitoring on the Growth of Spirulina Microalgae: Internet of Things and Microalgae Technologies. IEEE Internet of Things Journal **11**(2), 3274–3281 (2024)
12. Nwoba, E., Chuka-Ogwude, D., Vadiveloo, A., Ogbonna, J.: Process control strategies applied to microalgae-based biofuel production. In: 3rd Generation Biofuels: Disruptive Technologies to Enable Commercial Production, pp. 105–134. Woodhead Publishing (2022)
13. Samuel, C., Favitri, D.: IoT-based of Seaweed Cultivation Employing a Low-cost Autonomous Experimental Buoy. In: 15th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2021 (2021)
14. Schäfer, S., Schöttke, D., Kämpfe, T., Lachmann, O., Zielstorff, A.: Synchronizing Devices Using Asset Administration Shells. In: Innovative Intelligent Industrial Production and Logistics., vol. 1855 CCIS, pp. 70–92 (2023)
15. Vigani, M., Parisi, C., Rodríguez-Cerezo, E., Barbosa, M.J., Sijtsma, L., Ploeg, M., Enzing, C.: Food and feed products from micro-algae: Market opportunities and challenges for the eu. Trends in Food Science & Technology **42**(1), 81–92 (2015)
16. Yang, C.T., Kristiani, E., Leong, Y., Chang, J.S.: Machine learning in microalgae biotechnology for sustainable biofuel production: Advancements, applications, and prospects. Bioresource Technology **413** (2024)