Providing the Key Ingredients of an Edge PaaS for Supporting and Facilitating the development of Smart Energy Applications

Giovanni Di Orio[1] and Pedro Maló[1]

[1] NOVA School of Science and Technology, Portugal
gido@uninova.pt

Abstract. The monitoring and control of Critical Energy Infrastructure (CEI) is nowadays entrusted to Smart Grids (SGs). SGs rely on massive data and services to provide "awareness" about the status of the system. To do that distributed computing schemes have been applied based on decentralized communications, data collection, extractions, loading and analysis. These schemas are totally aligned with the Edge Computing (EC) paradigm. EC is an emerging paradigm that provides capabilities for processing and analyzing data away from the cloud, at the edge of the network closer to the source of the data. It offers multiple benefits including improved application performance, network latency reduction, and data locality. These characteristics reinforce EC is expected to have great impact on SG. However, a crucial aspect in implementing EC is company's foundational technology to really progress in cyber, digital, and cloud moves for SGs. The authors strongly believe that the foundation for the successful implementation of cloud/edge-based solutions strictly depends on employing new core architectures based on modern advanced cloud-native solutions, i.e., patterns, tools, techniques, and technologies derived from cloud-based design. As a result of this statement an Edge Platform-as-a-Service (PaaS) has been designed, developed, and deployed and used as the foundation of a flexible data platform at the Edge made up of fast-deployable, open source, and free-to-use PaaS services.

Keywords: Smart Grids, Edge Computing, Edge PaaS, Cloud-Native, DevOps, Industry 4.0, Asset Administration Shell.

1 Introduction to the Research

Traditionally the power gids have been operated using a centralized automation architecture where the primary monitoring, control and protection algorithms run at a central server [1]. In such a context, centralized cloud computing model has been easily applied to centralize functions, storage and network management [2]. However, the centralization of data from geo-distributed devices and equipment to cloud data centers, for computing centralized decisions and control orders is not without flaws especially with regard to performance, reliability and scalability [3]. Against this background, and taking into account the support of the new technologies such as big data, the 5-th generation wireless technology (5G), Internet-of-Things (IoT), internet applications, and Artificial

Intelligence (AI), we are assisting to the creation of more and more distributed multi-cloud environments comprised of a network of larger and smaller virtualized infrastructure nodes [4]. These "smaller" environments – often referred as edge clouds, edge computing (EC) or fog computing – are highly virtualized platforms that provides computing, storage, and networking services between physical devices and traditional cloud computing data services [5]. By employing the EC model to SGs, the computational requirements can be moved to the edge of the network leading to the design and deployment of SmartLocalGrids (SLGs). The way SLGs are created and the way they are developed (in terms of building blocks) are still an open issue and require a strong technology foundation, digital transformation strategy and holist agenda for tech transformations. As a matter of fact, even if Edge and cloud are essentially using the same building blocks, the "Edge of the network" shows some typical characteristics – pointed in [5] – that make EC development a non-trivial task. Furthermore, tech transformations remain notoriously difficult and complex. Too often companies simply migrate systems to the cloud without implementing cyber strategy, agile, DevOps, scale and flexibility offered by both cloud and edge-based systems as confirmed in [6]. Since economic outcomes are directly dependent on technologies capabilities, then research and investigation on how to achieve integration between EC and SG is necessary as also confirmed in [3]. With this in mind, the main purpose of this paper is to present a specific strategy, approach and technological development for EC clouds applied to industry in general, and energy domain in particular. Specifically, the paper presents the work done by the authors to provide an Edge PaaS for supporting smart energy applications. Cloud-native approaches and technologies have been investigated, and its principles used with the objective of coming with a "recipe" – that comprises guidelines, principles, and technological stack – for delivering the envisioned Edge PaaS.

2 What to Consider When Building an Edge PaaS

In this section paradigms, architectural models, tools, and technologies are presented that need to be considered when building an Edge PaaS. Firstly, the authors define Fog/Edge computing, secondly an overview of the main reference models is given to establish objectives, goals, guidelines, and a common language, and finally the main cloud-native principles that are necessary to execute the envisioned edge-based operating model while reinventing the technology delivery model.

2.1 Fog/Edge Computing

Edge/Fog Computing can be defined as an "highly virtualized platform that provides compute, storage, and networking service between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of the network" [5]. This computing model has been created to optimize cloud computing systems to face the new requirements imposed by digitalization and wider dissemination of IoT devices, by performing data analytics and processing at the edge of the network

near the source of the data [7]. The main advantages for the adoption of edge computing can be summarized as the following [3], [8], [4], [5]:

- Low Latency: moving the processing capabilities to the edge of the network can alleviate service latency while improving the decision-making process.
- On-line analytics: EC is positioned to play a fundamental role in the ingestion and processing of data close to the source.
- Decentralization: the presence of EC clouds can alleviate the core nodes of the cloud while weaken their dependencies on the cloud data centers.
- Distribution: supporting the geographical distribution of devices by delivering distributed EC clouds (composed by edge nodes) with computing and storage resources.
- Awareness: EC delivers support for a better localized decision-making process and end-user experiences by placing computing services closer to these locations.
- Interoperability: EC resources and services implicitly require the cooperation between different providers. This implies that EC must be able to interoperate.

2.2 Reference Models and Architectures

Edge/Fog Computing Reference Architecture. The edge computing reference architecture has been designed and released by the Edge Computing Consortium [9] (EEC) to promote in-depth industry coordination, accelerate innovation, and boost the application of edge computing. The reference architecture is built around four pillars: i) to describe connection and interaction between physical and digital world; ii) to provide a distributed architecture and platform; iii) to provide the development and the deployment&operation frameworks; and iv) to describe the connection with the cloud. On the same path, the OpenFog Consortium – created in 2015 – promotes fog computing adoption in various capacities and fields. The consortium released the reference architecture for Fog Computing [10] to help business leaders, software developers, silicon architects, and system designers create and maintain both hardware and software elements that are necessary for the fog computing. The reference architecture has been built around the Secure, Cognition, Agility, Latency and Efficiency (SCALE) principles.

Reference Models for Industry 4.0. Within the Plattform Industrie 4.0 project, the working group "Reference architectures, standards and norms" has developed the Reference Architecture Model Industrie 4.0 (RAMI 4.0). It is a three-dimensional map aiming at fully describing the Industry 4.0 space. RAMI4.0 is building the whole digital transformation strategy on the top of the Asset Administration Shell (AAS) concept. AAS is a standardized digital representation of the asset. It is the cornerstone of the interoperability between the applications managing the manufacturing systems. The AAS represents an asset (typically physical) by through its digital models. These models describe various aspects and as well as technical functionality of the respective assets. In the same direction, the Open Industry 4.0 (OI4.0) Alliance [11] is aiming to enable the digital transformation required to advance the principles of Industry 4.0. The reference architecture proposed is built around the EC and Edge connectivity concepts. Key principles are open interfaces, open edge application and cloud application layer,

data custodianship, role-based authorization and algorithms at every level for each provider and subscriber [12].

2.3 Considered Cloud-Native Principles

Virtualization: Containerization. Virtualization is a method of partitioning a system into multiple environments so that the resources of a single computer can be shared by multiple virtual systems. An important objective of virtualization is to centralize administrative tasks while improving scalability and workloads [13]. Other benefits include improved resource utilization, lower energy consumption, reduced hardware costs and improved disaster recovery processes. Server virtualization is the most common type of virtualization. It involves partitioning a physical server into multiple "virtual" servers. Each virtual server can be used to run a separate operating system instance and applications. Server virtualization can be considered as the foundation and infrastructure backbone of the first generation of cloud computing [14]. Another concept that is strictly connected to virtualization is containerization. Containers are a lighter and more agile way of handling virtualization. Containerization is an approach to software development in which an application/service, its dependencies, and configuration are packaged as standardized units. Containers enable an application/service to run on any computing environment without requiring any additional code, configuration, or dependencies [14].

Orchestration: Operations and Management. Container orchestration and operation management is the process of managing and operating container-based systems. Container-based architectures are typically deployed on groups of hosts organized into clusters [15]. Container management platforms or orchestration platforms, help IT teams to manage the various moving parts of container environments, namely: i) scalability; ii) configurability; iii) maintainability; iv) distribution. Specifically, it allows IT teams to automatically scale their containerized applications up/down based on demand, distribute containers across clusters of hosts, manage networking configurations for containers, schedule tasks and deploy new containers in response to events. The orchestration and management layer for containers is today handled by a wide variety of technologies. Kubernetes (k8s/k3s) has become the de facto standard for large-scale production environments.

DevOps and NoOps. One fundamental strategy to embrace cloud-native approach is DevOps/NoOps. It is a practice commonly used in today's software development process and relies on the deep collaboration between the development team (who builds the application) and the operation team (who deploys the application). It involves a set of practices aimed to enable and/or enhance continuous improvement, automation, cross-functional teams' collaboration while guarantying the alignment with business needs and costumer expectations of applications.

3 Building the Edge PaaS

In section 2 the authors reviewed several research topics to clearly identify and clarify the general goal and scope of the proposed research. As the next activity, the authors want to use the provided baseline to help shaping an Edge PaaS for building a data platform at the Edge. In this section, main benefits and values, in adopting a "cloud-native" approach to EC, are presented. Secondly, the proposed Edge PaaS design principles and architecture that combines a lightweight PaaS with cloud-native software stack are introduced. Then the Edge PaaS implementation based essentially on Kubernetes and Docker is discussed. Finally, a prototype of this architecture is shown.

3.1 Benefits and Values

The successful development, deployment, management and evolution of applications for industrial systems is connected with employing modern advanced cloud-native solutions i.e. patterns, tools, techniques and technologies derived from cloud-based design, in one word: containerization. The requirements for digitization – in almost all kind of industries regardless the kind of software development – are today demanding for:

1. Scalable (e.g. evolvable) and modular software architectures.
2. Automation in software operation and development.
3. Highly resilient and available systems.
4. Improved security.
5. Real-time capabilities, faster responses, and fast data analytics.

These generic requirements are still a big issue and represent a challenge in both enterprise and embedded software development. However, they can be easily mapped and connected to the EC and cloud computing value propositions. Specifically, the adoption of a platform-based model enables the fast delivery of value to both business and customer by delivering an operating environment to support applications for solving business problems. This platform transformation is built on a strong technology backbone.

3.2 Edge PaaS Design Principles and Architecture

The proposed Edge PaaS has been designed based on a set of principles aimed to identify an edge-operating model for the specific application context. The authors extracted these principles from the reference models analyzed in section 2, and are summarized in **Table 1**. To fulfill the authors' vision and considering the design principles in **Table 1** a reference architecture has been designed for the envisioned Edge PaaS. The architecture is the result of the syntheses of the author's main findings from the reference models analysis, and it is aimed to promote the development and deployment of scalable edge solutions based on containerized applications and open communication protocols (see **Fig. 1**). Further details on each layer are given in **Table 2** together with a reference to the models from which they are extracted.

Table 1. Considered Design Principles

Design Principle	Description
Reduced System Heterogeneity and Simplified Cross-Platform Migration	Targeting open and interconnectable principles between systems, subsystems, services, and new and legacy systems enable interaction, simplifying integration of these systems
Data Ownership	Delivering trust through comprehensible data custodian principles
Security by Design	Integrating clear and comprehensible security concepts
Easy Asset On-boarding	Adopting standardized solutions and communication protocols
Interoperability	Supporting data, syntactic and semantic interoperability
Support for system lifecycle activities	Supporting all the activities of application lifecycle from design to development and deployment
Cloud-native software backplane	Supporting containerization, orchestration and easy management of resources and applications

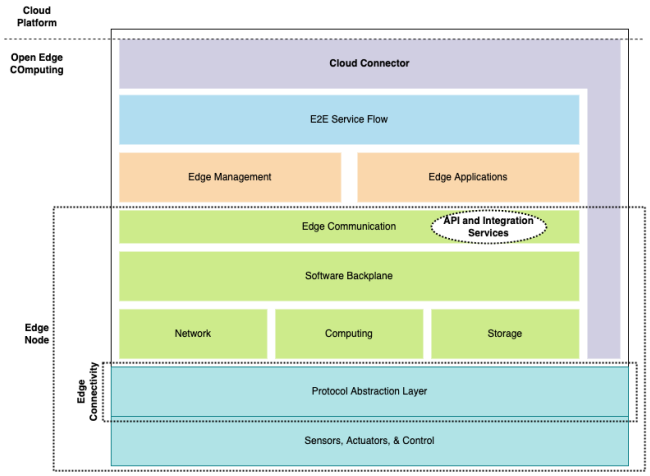

Fig. 1. Proposed Edge PaaS reference architecture (based on OpenFog and Edge Computing Consortium Reference Architectures)

Security Concerns. The adoption of EC introduces new security issues that need to be managed appropriately to avoid significant disruption and impact on business operations. There are many factors that make EC vulnerable to attack. First, decentralization makes EC difficult to secure from an enterprise perspective. Second, it relies on the public internet for connectivity which makes it easier for attackers to access resources remotely. Third, there is a lack of visibility into what edge nodes are connected and what types of data they are collecting and/or transmitting over the network. Considering the above concerns and as also confirmed in [10], security within the reference architecture is not a "one-size-fits-all" architecture but rather it describes all the necessary

Table 2. Reference Architecture: Layer Descriptions

Layer	Description	Reference
Cloud Connector	Guarantees secure communication pathways and data sharing between cloud and edge	OpenFog RA, ECC, OI4.0 RA
E2E Service Flow	Supports smart applications as the result of orchestrating services	ECC
Edge Management	Supports the unified management of edge infrastructure	ECC, OpenFog RA
Edge Applications	Any application running on the "Edge" and provided by Edge PaaS	ECC, OpenFog RA, OI4.0 RA
Edge Communication	Standardized layer to communicate with Edge Nodes	OI4.0 RA
Software Backplane	Run time environment for container-based system	OpenFog RA
Network/Computing/Storage	Capabilities of Edge Nodes	OpenFog RA, ECC
Protocol Abstraction	Provides the mechanisms to enable sensors, actuators and IoT in general to connect to Edge Nodes	OpenFog RA OI4.0 RA
Edge Node	The lowest level view of the reference architecture and actually connected to the physical layer	RAMI4.0 OpenFog RA

mechanisms that can be applied to make EC elements secure from hardware to software perspectives.

In the context of this paper security is not the focus, however some mechanisms have been employed and to enhance trustworthiness and reliability of the proposed data platform at the edge. Specifically, the authors' adopted the concept of "Industrial Demilitarized Zone" (IDMZ) adopted from [16], that provides a level of network segmentation that helps protect the internal networks. Servers and more specifically applications that needs to be exposed over the internet will be included within the IDMZ while any other resource and severs will be handled as restricted (no accessible from the outside of the network). Finally, the resources within the IDMZ will use a Virtual Private Network (VPN) for granting remote access.

3.3 Moving From Reference to Practicality

The reference architecture provides the structure of the Edge PaaS using a layered representation to highlight everything what matters. During the next sections the main objective is to plan how to morph the reference architecture to the needed architecture.

Technical Architecture and Building Blocks. Considering the reference architecture in **Fig. 1** it is now time to add details to it by considering both the functional and technological perspectives. In **Fig. 2** the reference architecture is broken down into smaller parts aimed to provide more concrete information about software, hardware components and technologies used. In particular, this technical architecture shows decisions

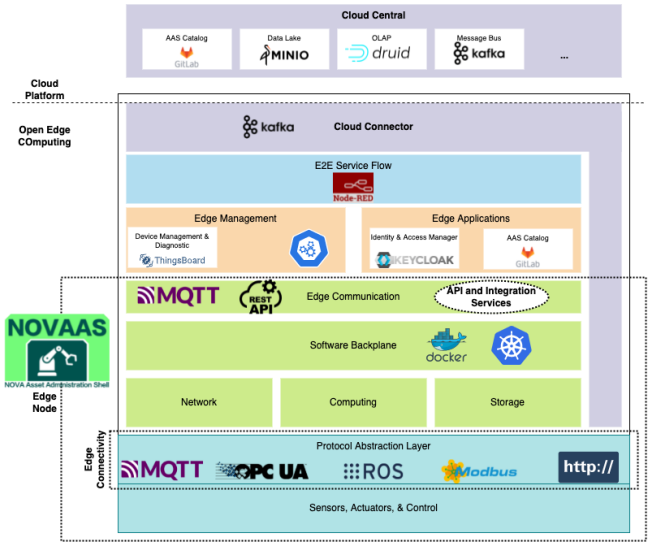

Fig. 2. Edge PaaS detailed architecture with main building blocks and technologies used

made by the authors about programming languages, frameworks, communication protocols, standards, and "must-have" features that is driving the edge-operating model.

3.4 Edge PaaS and Edge Data Platform Prototype

Implementation Context, Deployment View, and PaaS Services. To evaluate and validate the benefits of adopting an Edge PaaS based on cloud-native design principles a case study has been developed under the scope of the H2020-EU research project BD4NRG. One of the scientific and technological objectives of the project is the design and development of edge-oriented scalable architecture in smart energy grids. More specifically, BD4NRG aims at pushing forward big data management and processing towards the edge with a view to reduce some of the major barriers hampering big data & analytics full scale deployment in smart energy grid domain. One of the industrial partners wanted to develop newer tools and value-added services for improving the overall process of extracting, collecting, loading, and analyzing the data from the solar power plant. The payback will be the improvement of the overall system performance and efficiency thanks to data-informed decisions. In this case study, the authors suggested to reorganize IT operations by including advanced methodologies, strategies, and tools to improve overall system agility for faster software delivery and reduced applications development and management costs. In particular, the solution is built on an infrastructure backbone to ensure the implementation of a DevOps strategy. The strategy is built around Continuous Integration/Continuous Delivery (CI/CD) principles and requires a specialized IT architecture. The infrastructure, the tools and the CI/CD process for the test case (actually the Edge PaaS) are shown in **Fig. 3**. The Edge PaaS is managed by using the following toolchain:

- GitLab: is a web-based DevOps lifecycle tool that provides a Git repository manager using an open-source license, developed by GitLab Inc.
- Docker: is a set of PaaS products that use OS-level virtualization to deliver software in packages called containers.
- Kubernetes/K3S: is a highly available, Kubernetes distribution designed for production workloads in resource-constrained, remote locations or inside IoT appliances.
- Portainer: is a container-as-a-service solution that works with Docker and k8s/k3s.

A simulated environment has been created to run and test the Edge PaaS. This environment is made up of a cluster of four virtual machines that run Ubuntu 20.04. K3S is used to manage the cluster and orchestrate the Docker container-based environment. A graphical tool is used to communicate with K3S (i.e. Portainer). This tool allows to easily create .yml specifications and deploy new services/containerized applications within the PaaS. To enable the implementation of CI/CD pipeline the GitLab container registry is used. On the top of this infrastructure, a fully compliant I4.0 data platform at the Edge is running (see **Fig. 4**) based on the following PaaS services: i) Message Broker; ii) Identity & Access Management; iii) Client Application; and iv) AAS. The platform is designed using the broker topology model where assets events are broadcasted to a lightweight message broker. The component responsible for collecting the data from the physical asset is designed according to the I4.0 specifications and takes the name of AAS. The AAS will be also responsible to hold standards digital models for of the underlying physical asset. Finally, connected to the broker a client application is used to subscribe and ingest AAS events. The prototype provides an implementation of the AAS component based on NOVA Asset Administration Shell (NOVAAS) [17]. NOVAAS is an open-source AAS and will be responsible to provide – from one side – a "living instance" of the AAS itself, and – from the other side – a digitalization of the physical asset connected to it. NOVAAS is an execution environment for AAS designed by using the AASX package explorer tool. The AASX package explorer is an open-source tool that can be used to create a standardized digital description of an asset – also called "data image of the asset". This data image is a collection of submodels on several aspects related with the asset such as: operational data, documentation, security, etc.. In the context of this research, the *security,* and a simplified version of the *identification* submodels have been included and are used together with a custom *operational data* submodel. The *identification* submodel aims at interoperable provision of information describing essentially the manufacturer of the asset. The *security* submodel is used by the AAS itself to authenticate & authorize users before accessing asset related data. As soon as e new client application/user want to access a resource the AAS initiates interactions with the *Identity & Access Manager* service (implemented using Keycloack) to verify the access rights of the client. Finally, the *operational data* submodel gathers the so-called "live data" of the asset, i.e. the data related to asset variables and quantities that are changing over the time during the operation of the asset. These data can be accessed by using the API (request/ reply exchange pattern) or MQTT events. In the latter case, a pipeline is created, and data is pushed outside the boundaries of the AAS periodically. From now on, the Edge PaaS allows the management and evolution of the Edge data platform by simply providing deployment file for k3s.

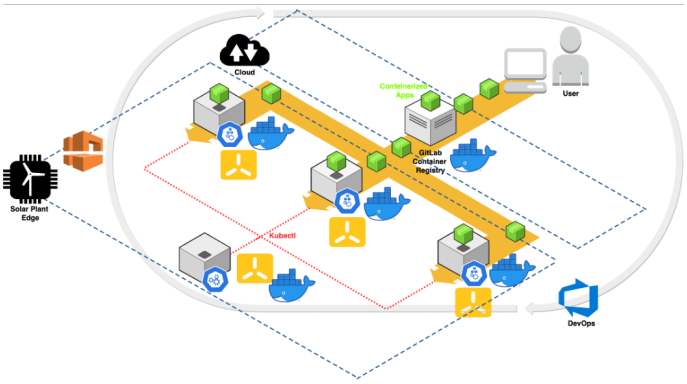

Fig. 3. Edge PaaS for DevOps CI/CD Strategy

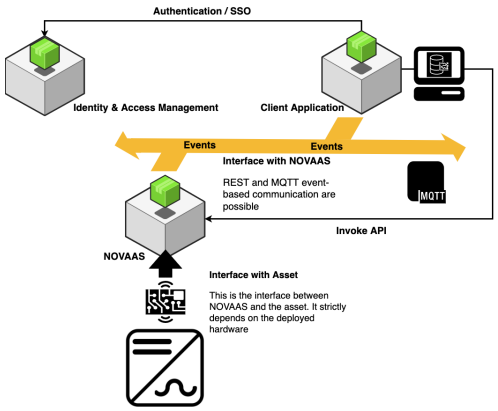

Fig. 4. The I4.0 Edge data platform used to show the Edge PaaS capabilities

CI/CD Pipeline: DevOps Pattern. One of the main advantages of the technologies adopted to deliver the Edge PaaS is that it is possible to easily integrate CI/CD pipelines within the Edge Data platform to set-up a perpetual-evolution model. To showcase this feature, an example based on NOVAAS is used. NOVAAS is a quite dynamic component, as a matter of fact after it is developed and deployed in production it could happen that something changes within the environment that – in turn – causes a change in NOVAAS. A possible scenario is the need to adapt the NOVAAS digital models due to a new security policy to be included within the *security* submodel, and/or to the inclusion of new quantities to the *operational data* submodel. In both the cases, it is necessary to update the internal NOVAAS's digital models and create a newer version of the component. To make this adaptation process almost transparent to the user a CI/CD pipeline has been developed (see **Fig. 5**). The developed CI/CD pipeline used the GitLab tool to create a *Shared Repository* for NOVAAS. This repository contains all the necessary files to containerize NOVAAS. Therefore, any change within the shared repository made by a developer of the development team a building phase is automatically triggered, and a newer container is generated by using docker. This newer image

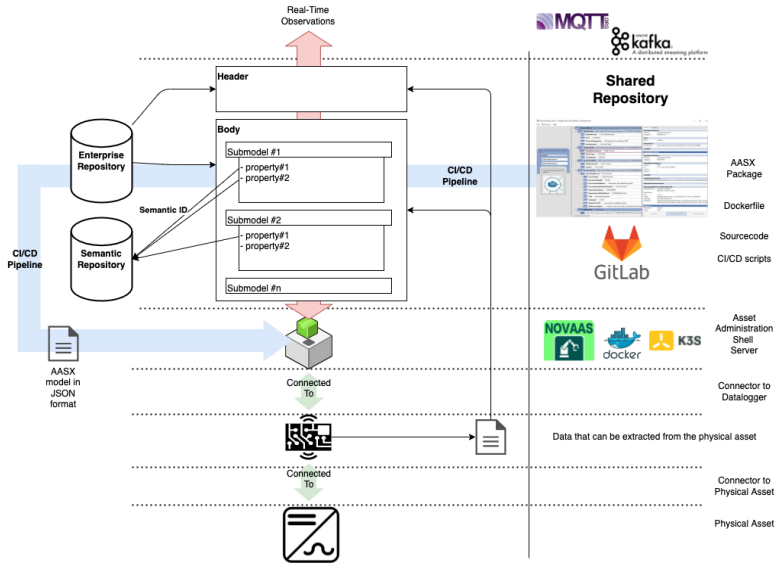

Fig. 5. CI/CD Pipeline Pattern "write once run-everywhere": edge-operating model

will be included within the container repository that is part of the *Shared Repository*. Once the image is ready, it is possible to update at any time the NOVAAS service in production (the one that is running within the Edge Data Platform) by simply using portainer and instructing k3s to recreate the container.

4 Final Remarks and Future Developments

Cloud-native principles are still at early stages when applied to EC. However, during this research the author observed and demonstrate that cloud-native principles have the potential to really promote the adoption of EC to industry (in this case SGs) as well as to facilitate the management of a container-based system while reducing significantly the time needed to deploy and evolve the system to face newer requirements. However, more research and work are still needed to improve and automatize the whole process of having an Edge PaaS and to deploy the necessary edge features on the top of it. Finally, the establishment of PaaS performance metrics is a necessary condition for boosting/encouraging the IT digital transformation while benchmarking and justifying the proposed approach. At the current stage of development, the following Key Performance Indicators (KPIs) will be used to measure the goodness of the approach: i) Time and effort to integrate a new application; ii) Time to redeploy an application; iii) Existing services re-use; iv) Time and effort to integrate business processes and event processors; and v) Ability to dynamically right-size the infrastructure and scalability. These KPIs will be used to measure, monitor and guide the progresses and to ensure the alignment between business and IT objectives and goals.

References

[1] G. D. Orio, G. Brito, P. Malo, A. Sadu, N. Wirtz, and A. Monti, “A Cyber-Physical Approach to Resilience and Robustness by Design,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 7, Art. no. 7, 34/31 2020, doi: 10.14569/IJACSA.2020.0110710.

[2] F. De la Prieta and J. M. Corchado, “Cloud Computing and Multiagent Systems, a Promising Relationship,” in *Intelligent Agents in Data-intensive Computing*, vol. 14, J. Kołodziej, L. Correia, and J. Manuel Molina, Eds. Cham: Springer International Publishing, 2016, pp. 143–161. doi: 10.1007/978-3-319-23742-8_7.

[3] C. Feng, Y. Wang, Q. Chen, Y. Ding, G. Strbac, and C. Kang, “Smart grid encounters edge computing: opportunities and applications,” *Advances in Applied Energy*, vol. 1, p. 100006, Feb. 2021, doi: 10.1016/j.adapen.2020.100006.

[4] C. Pahl and B. Lee, “Containers and Clusters for Edge Cloud Architectures – A Technology Review,” in *2015 3rd International Conference on Future Internet of Things and Cloud*, Aug. 2015, pp. 379–386. doi: 10.1109/FiCloud.2015.35.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things,” in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, New York, NY, USA, 2012, pp. 13–16. doi: 10.1145/2342509.2342513.

[6] A. Dhasarathy, I. Gill, N. Khan, S. Sekar, and S. Van Kuiken, “How to become ‘tech forward’: A technology-transformation approach that works,” McKinsey Technology, Nov. 2020.

[7] “The Implementation of a Cloud-Edge Computing Architecture Using OpenStack and Kubernetes for Air Quality Monitoring Application | SpringerLink.” https://link.springer.com/article/10.1007/s11036-020-01620-5.

[8] S. Chen, H. Wen, J. Wu, W. Lei, W. Huo, W. Liu, A. Xu, Y. Jiang, “Internet of things based smart grids supported by intelligent edge computing,” *IEEE access*, vol. 7, pp. 74089–74102, 2019.

[9] “Edge Computing Consortium.” http://en.ecconsortium.org/Index/index.html.

[10] OpenFog Consortium Architecture Working Group, “OpenFog Reference Architecture for Fog Computing.” Feb. 2017. [Online]. Available: https://www.iiconsortium.org/

[11] “Open Industry 4.0 Alliance,” *Open Industry 4.0 Alliance*. https://openindustry4.com/.

[12] Open Industry 4.0 Alliance, “Open Industry 4.0 Alliance White Paper.” 2020. [Online]. Available: https://openindustry4.com/

[13] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” National Institute of Standards and Technology, NIST Special Publication (SP) 800-145, Sep. 2011. doi: 10.6028/NIST.SP.800-145.

[14] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, “Cloud Container Technologies: A State-of-the-Art Review,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 677–692, Jul. 2019, doi: 10.1109/TCC.2017.2702586.

[15] V. Koukis, C. Venetsanopoulos, and N. Koziris, “okeanos: Building a Cloud, Cluster by Cluster,” *IEEE Internet Computing*, vol. 17, no. 3, pp. 67–71, May 2013, doi: 10.1109/MIC.2013.43.

[16] CISCO and Rockwell Automation, “Securely Traversing IACS Data Across the Industrial Demilitarized Zone.” Rockwell Automation, 2017. Accessed: Mar. 05, 2022. [Online]. Available: https://literature.rockwellautomation.com/idc/groups/literature/documents/

[17] G. di Orio, P. Maló, and J. Barata, “NOVAAS: A Reference Implementation of Industrie4.0 Asset Administration Shell with best-of-breed practices from IT engineering,” in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2019, vol. 1, pp. 5505–5512. doi: 10.1109/IECON.2019.8927081.