D4.14 – i4Q Edge Workloads Placement and Deployment

WP4 – BUILD:
Manufacturing Data Analytics for Manufacturing Quality Assurance

i4Q has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958205.

Document Information

GRANT AGREEMENT NUMBER	958205	ACRONYM	i4Q
FULL TITLE	Industrial Data Services for Quality Control in Smart Manufacturing		
START DATE	01-01-2021	DURATION	36 months
PROJECT URL	https://www.i4q-project.eu/		
DELIVERABLE	D4.6 – i4Q Edge Workloads Placement and Deployment (V2)		
WORK PACKAGE	WP4 – BUILD: Manufacturing Data Analytics for Manufacturing Quality Assurance		
DATE OF DELIVERY	CONTRACTUAL 31-Dec-2022	ACTUAL	30-Dec-2022
NATURE	Other	DISSEMINATION LEVEL	Public
LEAD BENEFICIARY	IBM		
RESPONSIBLE AUTHOR	Benny Mandler (IBM)		
CONTRIBUTIONS FROM	Benny Mandler; Nir Naaman; Maroon Ayoub; Nir Rozenbaum (IBM)		
TARGET AUDIENCE	1) i4Q Project partners; 2) industrial community; 3) other H2020 funded projects; 4) scientific community		
DELIVERABLE CONTEXT/ DEPENDENCIES	This document presents a technical overview of the AI Workload Placement and Deployment (i4Q^{EW}) component. This document is the second and final iteration scheduled for M24 (Dec. 2022)		
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None		
READING NOTES	None		
ABSTRACT	i4Q^{EW} is a toolkit for deploying and running AI workloads on edge computing environment, prevalent in manufacturing facilities, using a Cloud/Edge architecture. i4Q^{EW} provides interfaces and capabilities for running different workloads on different industrial devices, efficiently on the edge, including placement and deployment services. AI workloads at the edge are later used by the analysis components to run their inference close to the data sources. Target deployment environments may be very heterogeneous and dynamic, thus deployment needs to take a variety of criteria into consideration. The environment is dynamic thus re-deployment of the entire workload or the adaptation of the underlying model may be required while the workload is running. Deployment shall be based on well-known orchestrators, such as Kubernetes.		

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	07-Nov-2022	ToC	Table of Contents creation	IBM
0.2	01-Dec-2022	Draft	First draft available for internal review	IBM
0.3	12-Dec-2022	Internal Review	Internal review	EXOS, AIMPLAS
0.4	19-Dec-2022	Draft	Incorporate comments from external review	IBM
1.0	30-Dec-2022	Final Document	Final quality check and issue of final document	CERTH

Disclaimer

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ACRONYMS

ACM	Advanced Cluster Management
AI	Artificial Intelligence
CNC	Computerized Numerical Control.
DSS	Decision Support System
EW	Edge Workload
IoT	Internet of Things
K3s	Lightweight Kubernetes
K8s	Kubernetes
LRT	Line Reconfiguration Tool
NFR	Non-Functional Requirement
ML	Machine Learning
OCM	Open Cluster Management
RHAM	Red Hat Advanced Cluster Management
UI	User Interface
VM	Virtual Machine

Executive summary

Deliverable D4.6 V2 presents a technical overview of the **i4Q** Edge Workloads Placement and Deployment solution. This deliverable provides an overview of the unique operational environment of AI on the edge, diving into the specific capabilities provided by this task to other i4Q solutions and pilots.

i4Q^{EW} is a toolkit for deploying and running AI workloads on edge computing environment, prevalent in manufacturing facilities, using a Cloud/Edge architecture. i4Q^{EW} provides interfaces and capabilities for running different workloads on different industrial devices, efficiently on the edge, including placement and deployment services. The design and implementation of this solution tackles the challenges of scalability and heterogeneity into account right from the beginning.

There is a tight coordination required between this task and the Scalable Policy-based Model Distribution from Cloud to Edge (implemented in the solution i4Q^{AI}: AI Models Distribution to the Edge solution), such that the AI workload and model shall meet and collaborate in the correct target edge nodes.

This solution resides at the back-end infrastructure level, providing capabilities to be used by other solutions and pilots. The design is influenced by internal functionality at the core of the solution as well as interfaces and capabilities required by potential clients. This solution supports the placement and deployment of AI workloads of many kinds on edge nodes running several varieties of orchestrators based on the underlying heterogeneous devices.

For the first release of this solution the main collaboration would be with the manufacturing Line reconfiguration (LRT) solution provided by UPV in the context of the FACTOR pilot. This solution shall deploy the AI workloads and application provided by the LRT solution. As a second stage we aim for the solution to be incorporated within the generic pilot as well.

A recorded video demonstrating the current version is being made available on the project repository in SharePoint and may be available for dissemination purposes.

This document i4Q D4.6 v2 is an update of v1 of D4.6., for this reason it contains information of the 1st version together with the updates developed in this 2nd version.

Document structure

Section 1: is comprised of a general description of the operational environment (cloud-edge multi-tier environment and challenges) and specifically the **i4Q AI Edge Workloads Placement and Deployment** component, providing an overview and a list of features provided by this solution. It is addressed to additional solutions and end-users who may be interested in making use of this $i4Q^{EW}$ Solution from within another solution or a pilot.

Section 2: Contains the technical specifications of the **i4Q AI Edge Workloads Placement and Deployment** component, providing high level view and its accompanying architecture diagram. It is addressed to software developers who may want to interact with this solution.

Section 3: Details the implementation status of the **i4Q AI Edge Workloads Placement and Deployment** component, explaining the current status.

Section 4: Provides the conclusions.

APPENDIX I: Provides the detailed web documentation that accompanies this solution.

1. General Description

1.1 Overview

T4.3 (Scalable Policy-based Model Distribution from Cloud to Edge) and its respective deliverable (D4.5 - i4Q AI Models Distribution to the Edge) which summarizes the work done in task 4.3 are highly inter-related with T4.4 (AI Workload Placement and Deployment) and the corresponding deliverable (D4.6 - Edge Workloads Placement and Deployment). Thus, for completeness several of the sections are included in both deliverables. This mostly applies to background information providing details on the technological area and the challenges tackled by these tasks. The specific implementation related sections are separated between both these deliverables.

Edge computing is establishing itself as the architecture of choice for many industries across many use cases. There are several important reasons for this trend, such as processing data close to the source, thus reducing latency of the computation response, especially when working in real-time. In addition, network bandwidth is conserved, serving together as a first line of defence to tackle big data challenges by handling data locally as much as possible, rather than serve as a pipeline for all data to be transmitted to the cloud and be processed there. In such an architecture the vast majority of data is processed at the edge and only a sub-set, such as feedback related information, is shared with the cloud. Security and privacy are addressed well in such an environment by keeping and processing the information local without being externalized and sent to the cloud or a central data centre. Such an architecture may prove to be beneficial also for abiding by data related regulations, such as restrictions on the data storage and processing locations. This architecture supports also working in a disconnected mode.

There are several unique features that are prominent in such as an environment, chief among these is the scale, heterogeneity of computing platforms, and support for a disconnected mode of operation. The end goal is to develop an efficient, large-scale hybrid cloud edge infrastructure for edge computing, supporting AI operations at the edge by handling AI models and workloads lifecycle. As briefly mentioned above, there are several reasons and advantages for having such a hybrid edge computing paradigm:

- Reduce the amount of data sent to the cloud. A significant portion of the data processing can be performed at the edge, close to where the data is generated, thus dramatically reducing the amount of data that has to be sent to the cloud. This allows more data to be processed at a lower cost in less time.
- Ability to work while disconnected from the cloud. This is a big advantage (and sometimes a must) in many use cases where communication with the cloud is unreliable or is not always available.
- Privacy and security. In certain cases, some of the data must remain local and cannot be sent to the cloud, for example, due to regulatory reasons. In other cases, users may want to minimize the data exchange with the cloud to reduce the security risks involved with data in motion.
- Faster reaction time, lower latency. In several use cases a fast reaction is important and the ability to process the data on the edge and react immediately is important.

Applications involving big data and/or IoT are examples of applications that can greatly benefit from a hybrid edge computing environment. Edge computing brings a set of challenges, such as being able to work in a resource constrained environment, management and monitoring,

being able to work in disconnected mode, heterogeneity of devices and more. We focus on some of the key challenges involved in creating a scalable, manageable, hybrid edge computing, concentrating on AI on the edge capabilities.

Edge computing adds another tier of data processing to cloud processing. We define an architecture for a multi-tier hybrid edge computing platform that can efficiently service multiple types of applications and use cases with different requirements. The platform addresses issues such as the management of edge nodes, distribution of model updates, smart workload placement, and more.

Edge computing is becoming prevalent in many sectors, including smart manufacturing. Factories may start experimenting at a small scale, but in large and distributed production environments such capabilities are becoming a requirement. To support this trend, we designed the AI on the edge component in a scalable manner to support distributed installations, while ensuring that small scale scenarios are supported as well.

Edge computing typically involves large-scale deployments on a wide range of heterogeneous devices. Managing the artifacts that need to run in such an infrastructure is complex and hinders the growth of this field. We intend to alleviate the management of AI models and workloads in an edge environment.

Managing a large number of edge nodes is a key aspect of edge computing. Edge nodes are diverse and come in a variety of specifications and supported capabilities. However, a clear trend is to run containerized workloads on the edge as well. One of the obstacles in front of this vision is the limited support for easy deployment of AI ingredients on target edge machines. This is an issue being tackled in this task. We aim to reduce deployment complexity of AI models and workloads in edge scenarios. Such AI models tend to change based on incoming feedback from the running system. Thus, an extension of this task shall support the re-deployment of AI models based on updated information concerning the efficiency of the currently running model and available resources. Taking into account both the model running in real-time and the state of resources on the respective edge nodes. This forms a loop from deployment through feedback leading to re-training and re-deployment. AI models and workloads exhibit different lifecycle models. Workloads tend to be more stable with a longer lifecycle loop, while associated AI models tend to be updated at a higher rate, to improve its results based on updated feedback obtained from the field.

The potentially large number of edge nodes presents challenges in terms of configuration, management, and monitoring of the AI workloads and the environment. We design and implement a component which is responsible for providing a central control point for the management of AI workloads and models running on the edge. This central point will be accessible from the cloud and will enable the user to specify the rules that govern the location in which certain computations should take place.

This solution provides services to additional solutions and pilots for deploying AI workload on target edge devices and clusters. AI models have different life cycle behaviour and pattern than AI workloads and thus they require differential treatment, including the possibility to place them at correct locations at the edge, monitor their effectiveness, and be able to receive feedback that may lead to retraining and redistribution of refined models to the edge, as well as the update of AI applications making use of these models. AI workloads are much more stable in nature and need to be re-deployed relatively rarely, thus the requirements and modes of operations are somewhat different between components handling AI modes and AI workloads. A sub-set of these capabilities shall be demonstrated within the FACTOR use case (for more details please see 3.2).

The AI workload placement mechanism considers the edge environment along with user preferences and policies to decide where to deploy and run each workload. Automatic workload placement enables more applications to utilize Edge Computing and significantly improves workload performance and cost efficiency. The limited edge resources present challenges for adjusting AI workloads and models for the edge. We devise the capability to run edge workloads on a cluster rather than on single node, using for example orchestrators such as an underlying Kubernetes cluster. Such capability shall support High Availability of the component, as well as enhanced sustained throughput and load distribution. An example for this need is a production line in which several edge nodes are used to handle overall capacity required for mission critical image processing at the edge.

This task aims at alleviating deployment of AI workloads on the edge by introducing flexible and scalable automatic lifecycle management of AI on the edge components. Target deployment environments may be very heterogeneous and dynamic, thus deployment needs to take a variety of criteria into consideration. The environment is dynamic thus re-deployment of the entire workload may be required while the workload is running. Interfaces and capabilities to run different workloads on different devices shall be pursued. A Cloud/edge architecture provides efficient and flexible management of edge workloads. Deployment is based on well-known orchestrators, such as Kubernetes. AI at the edge enables AI computation logic and operations close to the data source.

Edge platforms are expected to improve the efficiency of edge devices while alleviating the obstacles for developers and administrators when managing the complex hybrid environment. Infrastructure for supporting edge applications based on a microservices approach is presented. The current envisioned use cases in the pilots may be rather small scale at the moment but we envision advanced features to cater for future large-scale scenarios which represent current and future directions and trends envisioned also in the smart manufacturing space. Once the technology is demonstrated on a single machine, requirements for scale are expected to follow soon, requiring support for more workloads, more machines, more models, more production lines, more plants, etc. Even if at a first step modest requirements are foreseen, we intend to build and use scalable technologies (such as ACM and K8s on the edge) to be present and future ready.

The main capabilities of this solution are demonstrated in conjunction with the Line reconfiguration tool (LRT) solution in the FACTOR use case. Need for scale in this arena can be seen for example in the IBM-FORD joint effort on defect identification across many plants and devices. The solution was deployed at several plants and embedded in multiple inspection points per plant. As a consequence and aftermath of a small-scale demonstration, the technology and deployment are expected to be expanded to additional plants and use cases.

In essence, the objective of this solution is to help deploy and operate AI workloads and applications close to the source of the data and its associated model, such that the entire AI workload-model combination works well at the edge, providing low latency responses to dynamically changing environments. This solution supports industrial companies in the quest for autonomous operation in manufacturing environments, taking advantage of AI. Consequently, the production rate can be increased, while waste and cost can be reduced. Thus, the main outcome of this solution is a workloads distribution and deployment component designed for high scale edge computing targeting manufacturing and industrial applications. The main challenges and characteristics are scalability, automation, flexibility, and ease of use workloads distribution mechanism. This capability should support integration with an overall AI model lifecycle management. Furthermore, integration and coordination with model distribution is targeted to ensure workloads and their AI models meet at the edge.

1.2 Related Work

Multi-Cluster management is a key aspect of a cloud-edge architecture. Large scale deployments are driving this trend. One reason for the increased demand in scale is the evolution of edge computing, which is growing rapidly with more and more workloads moving or extended to the edge to benefit from efficient use of bandwidth, low latency, and enhanced privacy and security. This means that far edge use cases now include a large number of edge nodes hosting a variety of workloads. In addition, Kubernetes has established itself as the platform of choice for running applications and services. Kubernetes is thus replacing many other platforms (e.g., Docker, and OpenStack) due to the uniform management it provides across different environments (such as public cloud, private cloud, near edge, and far edge). In addition, reduced footprint clusters are being produced, enabling Kubernetes (K8s) clusters to be deployed in more environments. This includes far edge scenarios which, until recently, didn't seem suitable for K8s. Offerings such as k3s (https://k3s.io), take this approach to the extreme and offer K8s versions that can run on devices with very limited resources such as a Raspberry Pi with less than 1GB of memory. This opens up the door for using K8s as the underlying orchestrator in many edge use cases (such as smart manufacturing, and more). Large scale edge management frameworks are being explored. For example, Open Cluster Management (https://github.com/open-cluster-management) offers comprehensive multi-cluster management capabilities with extensions to address large scale far edge use cases.

1.3 Features

The main features provided by this solution include:

- Take placement decisions and deploy AI workloads from the cloud (or a data centre) to the edge, where they are expected to use locally available models. The AI workloads distribution is coordinated with the model distribution and deployment mechanism to ensure that the right workloads get to interact with the corresponding AI model.
- Help deploy AI workloads at the edge in scale. To enable that, a policy-based deployment pattern can be used via associating policies, rules, and labels with the potentially different target nodes.
- Manages lifecycle of AI workloads, from creation at the cloud, initial deployment at the edge, and re-deployment when necessary. It is expected that workloads are more stable and shall be redeployed at a much lower rate than the models themselves. In addition, the characteristics of AI workloads and models is different, for example the workloads are expected to be smaller than the models. Finally, when not needed anymore the deletion of the workload is supported as well.
- Support policy-based placement mechanism that eases the task of the administrator by enabling the specification of rules for eligible targets in a simplified manner.
- Support GitOps based mode of operation. The interaction point between the AI application developer and the application administrator and deployer is achieved via Git. New resources that get pushed into Git are automatically retrieved to be deployed on the target nodes associated with the requested labels.
- Support the deployment of workloads with Red Hat Advanced Cluster Management for Kubernetes (RHACM). ACM serves as the basis for enabling deployment operations at scale.

- Support AI workloads packaged in a container. The orchestration engine is in charge of running the workload, and takes care of corresponding lifecycle operations, such as rolling updates.
- Operate seamlessly well for small to very large multi-site infrastructure common in smart manufacturing environments.
- Support lightweight orchestration engines such as k3s. Thus, it can be deployed and made operational over a variety of host devices and architectures, from high to low footprint artefacts.

These features are demonstrated in the video showcasing current capabilities[1].

[1] https://ibm.box.com/s/1ibaz47q33ydj331cfptr89fhd4hcaym

2. Technical Specifications

2.1 Overview

Edge computing faces challenges in scale and heterogeneity. A clear enabling trend is to run containerized workloads on the edge as well, introducing more opportunities for uniformly managing clouds to edge workloads. While many platforms still use container runtimes, such as Docker, Kubernetes is increasingly becoming the platform of choice for both the cloud and the edge nodes. As Kubernetes extends to the edge there is a growing demand to significantly increase the scale of multi-cluster management. This trend is expected to rapidly grow due to the combination of the following factors

- Kubernetes has established itself as the platform of choice for running applications and services. Kubernetes is thus replacing many other platforms (e.g., Docker, and OpenStack) due to the uniform management it provides across different environments (public cloud, private cloud, near edge, far edge).
- Edge computing is growing rapidly with more and more workloads moving or extended to the edge to benefit from efficient use of bandwidth, low latency, and enhanced privacy and security. This means that far edge use cases now include large numbers of edge nodes hosting a variety of models and workloads.
- Reduced footprint clusters managers are being offered which allow Kubernetes clusters to be deployed in more and more locations with varied resources. This includes far edge scenarios which, until recently, didn't seem suitable for Kubernetes. This opens up the door for using Kubernetes as the platform in many edge use cases.

We make use of the K8s support throughout the design, development, and deployment of these solutions.

Our research targets large scale edge platforms. An edge node can be anything from a tiny edge device to a multi-server edge cluster. We see a growing demand for such large-scale platforms across a variety of industries including telecom, manufacturing, automotive, industry 4.0, retail, IoT, and more. The scale of the platform mandates new and innovative solutions to a variety of management and automation aspects including lifecycle management for the edge infrastructure, and management of applications and services running on the edge. Special focus is given to Kubernetes edge clusters as Kubernetes becomes the platform of choice for both cloud and edge.

Our primary goal is to provide a scalable control plane capable of managing a large number of AI workloads and their associated models within distributed Kubernetes edge clusters.

This task addresses the challenge of managing large scale Kubernetes based edge computing platforms. In our solutions, edge nodes are operated as Kubernetes clusters which can be low footprint single-node clusters or large multi-node clusters. A scalable control plane is designed and implemented to control and manage the workloads that run on these edge nodes. The control plane shall address the multiple management aspects that are required for such edge platforms. These aspects include lifecycle management of the Kubernetes clusters, lifecycle management of the workloads running on the edge nodes along with their models.

This task aims to provide a distributed resource orchestrator and manager to enable AI at the edge. Kubernetes is mainly used, along with ACM and potential extensions where needed, to orchestrate and manage efficiently available resources in local edge devices and across

distributed clusters of edge devices. Deployment shall be pursued based on resource allocation policies. AI applications and models lifecycle management are pursued.

This task serves as a central enabler for being able to analyze data at the edge, relying on low latency and high data thruput. Provided capabilities help to de-couple edge applications from the underlying infrastructure, thus enabling the placement and deployment on different target nodes, by using universal tools such as K8s.

This task addresses the challenge of orchestrating large scale Kubernetes-based edge computing platforms. A scalable control plane is designed and implemented to control the edge nodes and workloads that run on these edge nodes. The control plane addresses AI workload and associated models lifecycle management.

This AI on the edge component addresses the life cycle management of workloads and AI models used by the pilots and i4Q use cases. This component enables efficient placement and deployment of AI based workloads and their associated AI models, on the edge, close to the data source locations. It further provides automatic deployment, updates, and monitoring of workloads and AI models, rather than a manual approach. This is more suited for a large and dynamic environment in which the deployment-feedback-retraining-redeployment cycle should be supported in an automatic manner. The AI on the edge component provides centralized management of all the edge workloads and the AI models they are using. A single Control-Center, which may run in the cloud or on-premises, can control workloads across multiple machines, production-lines, and plants which significantly simplifies management tasks, reduces errors, and cuts down costs. The AI on the edge component is generic and can work with a large variety of workloads, AI models, and edge infrastructures. For example, the same mechanism can be used for an application that processes sensors data (heat, vibrations, etc.) and an application that processes high-resolution images for automated visual inspection. Thus the component is not restricted to a specific solution but is rather a generic infrastructure mechanism that serves different components.

The main objective of i4Q^{EW} is to address the management of AI workloads in a hybrid cloud-edge manufacturing environment. On the one hand it supports the creation of AI applications using standard local or cloud native toolchains, while administering these artifacts and handling their lifecycle in a scalable manner. The main goal of this solution is to automatically handle the deployment of the AI workloads on the right target nodes at scale. The AI workload placement and deployment is coordinated with the model distribution mechanism to ensure that the right set of AI models is made available for the workload that uses them.

i4Q^{EW} is designed to operate seamlessly well for small to very large multi-site infrastructure common in smart manufacturing environments. Thus, the scope may span from a small local plant to multiple sites organizations. This is achieved by employing a policy-based placement mechanism that eases the task of the administrator by enabling the specification of rules and labels for eligible targets in a simplified manner. The end result of this solution is a new application (or a version thereof), operational on all designated managed clusters.

The architecture is multi-tiered with an overall management component running in the cloud, and an agent running on its behalf on every managed cluster, together the lifecycle management of AI workloads is supported across multiple locations in a dynamic environment. The intention is to enable such management at scale via a policy-based distribution mechanism. There is a tight coordination between the AI model deployment solution and the workload distribution mechanism. At the cloud layer tools are provided for the development of AI workloads. Once the developer is satisfied, the application is handed off to the cloud-based manager component. The management component communicates with an agent running on

participating nodes. The management component determines the correct set of managed clusters to be the designated targets on which the applications need to be deployed and executed. The local actions in the corresponding managed cluster are performed with the help of a locally deployed agent. Through this interaction the edge node receives the AI workload in question, either directly or through a link to git in which the application resides and can be retrieved. This is the mode of operation supported in the current release of the solution. Once the application is installed and deployed in its target, it can start executing and interacting with the corresponding models which are available. Local orchestrators agents take care of required local operations, such as rolling updates of the corresponding artifacts.

2.2 Architecture Diagram

Figure 1 presents a high-level architectural view of the AI on the edge solutions, and their interaction with their environment, at the different layers of operation. These components operate at the cloud level, the edge level, and the factory floor. At the cloud level resides the multi-cluster management component, which is in charge of managing all managed clusters in a scalable manner. At this layer resides also the development and authoring tools for AI models and workloads. Once a new version of one of these components is ready to be deployed, the corresponding artifacts are pushed into a monitored Git repository. The Git-repository serves as the contact point between the developers of the AI workloads and models and their respective edge deployment. Thus, the AI artifact developers can concentrate on their task and not worry about the artifacts deployment and execution. From this point, in a GitOps manner, the Advanced Clusters Management (ACM) notifies the relevant managed clusters on which components need to be deployed, that new resources are available. That causes the managed cluster in turn to pull the corresponding Git resources from the corresponding Git repository. While consuming the new resources, the managed cluster, via the standard Kubernetes mechanisms pulls resources from the Container registry (AI workload and model containers) and activates the required containers for the AI workload and model. All these components reside and operate at the edge layer. At the factory level there are instrumented machines, equipped with IoT sensors and actuators, that continuously provide data to be analysed by the edge components of AI workloads and models. The workload may automatically control a machine and automatically apply configuration suggestions, or can provide suggestions to an operator controlling the machine. Finally, data makes its way backwards as well from the machines at the factory level all the way to the AI models developers (data scientists) in the cloud. This feedback information is used to evaluate the efficiency of the model and to potentially retrain the model to obtain better results at the inference level. Once that is done the whole deployment cycle takes place as well culminating with a new version of the workload being deployed at the chosen targets. In a similar manner, but less frequent there can be changes that need to be applied to the AI workloads themselves. In this case, once again, ACM notifies the managed clusters that a new version of an artifact has been push to Git. The agent running on the managed cluster pulls the new version of the application and proceeds with a local rolling update, to obtain the new version running without any service interruption.

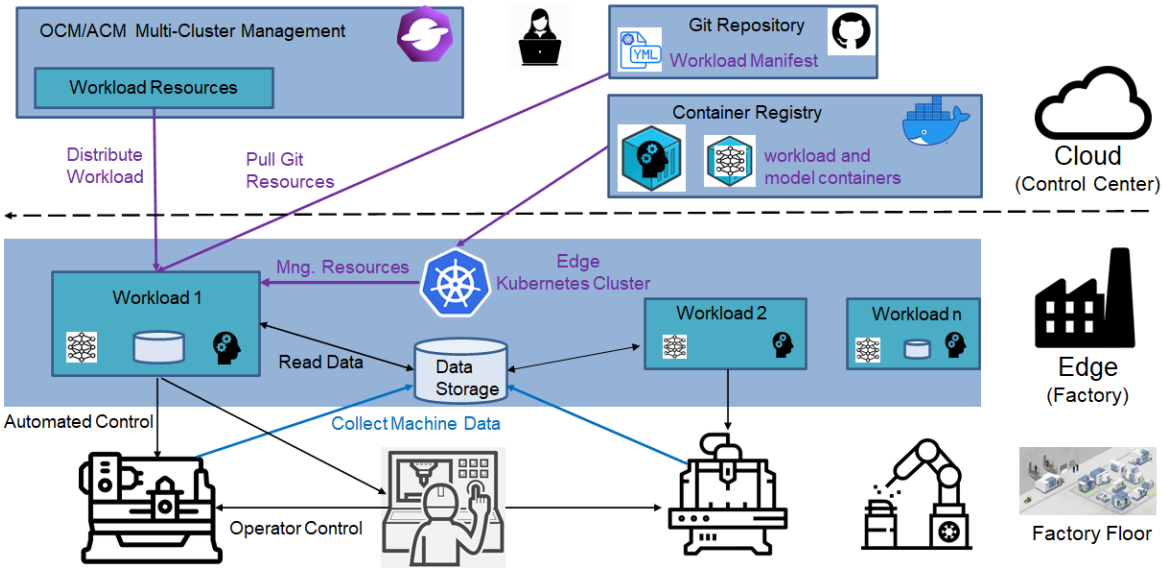

Figure 1. AI on the edge architecture

This solution includes different processes, which are mapped to the Platform Tier and to the Edge Tier of the i4Q Reference Architecture (see more details in D2.7). This solution shall operate at various levels of the architecture, a short analysis follows.

i4Q[EW] mainly resides at the Platform Tier (see **Figure 2**), but relies on services at the edge tier, and enables interaction with the enterprise tier. At the platform tier the most related components are the orchestrator management, and the distribution services. The distribution services cover the placement and deployment of workloads at the edge. Workload placement and deployment shall ease the orchestration within the entire platform. At the edge tier this solution is mapped to the distributed computing service, enabling the operation across a cloud and edge environment. Finally, at the enterprise tier there's a mapping with the "orchestration and distribution designer", which uses the orchestration management to carry out operations in the field.

Input consists of a workload to be deployed at the edge. Output consists of a running edge workload and updated view of the state at the edge.

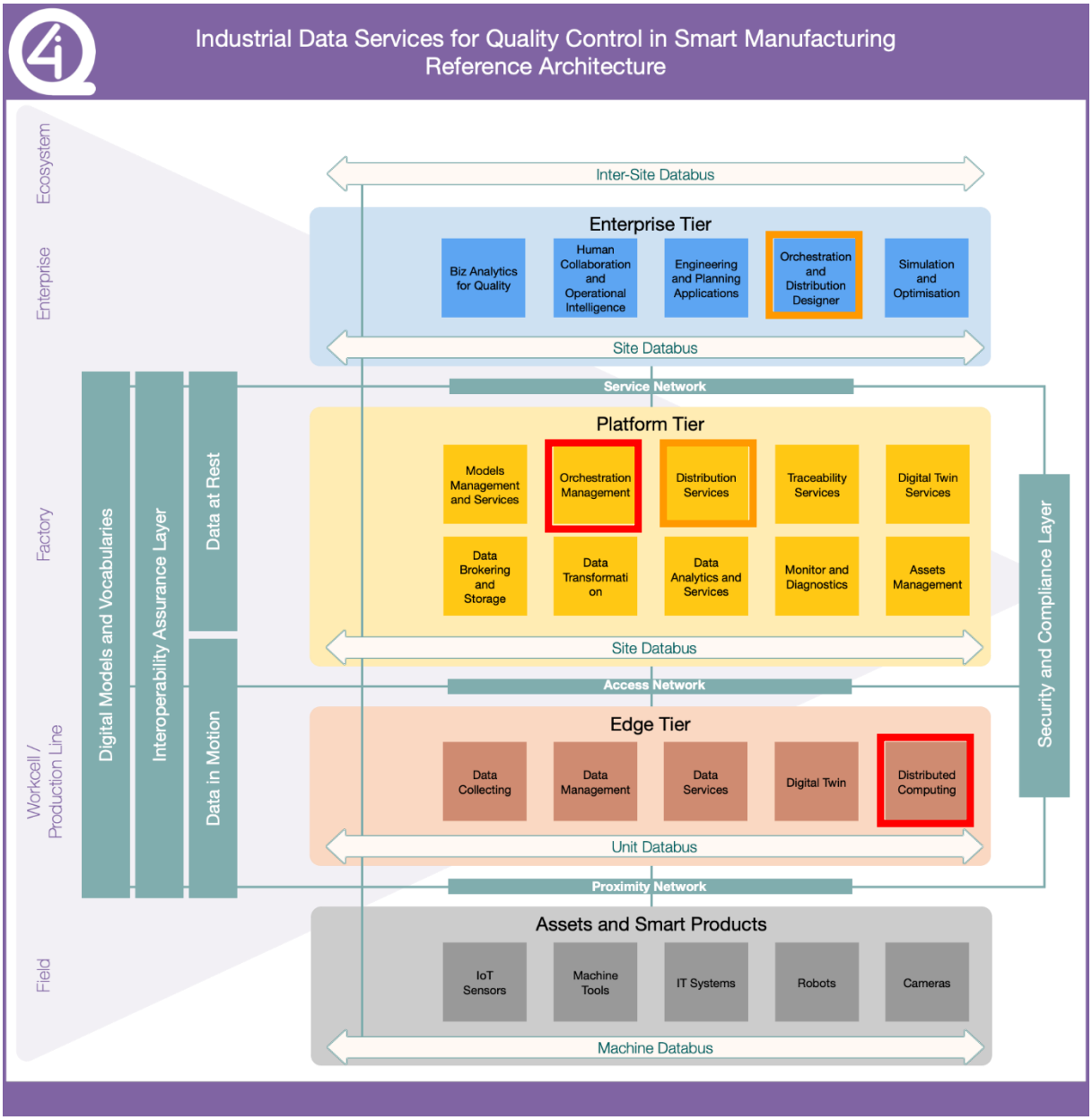

Figure 2. i4Q[RA] mapping with i4Q[EW]

Correspondence with the reference architecture:

- **Platform Tier:** “orchestration management" is used for the AI workloads management implementation. This shall serve as the hook between the enterprise tier authoring tools and attracts, to the actual edge-based workload deployment, while handling the corresponding specific life cycle of workload deployment and redeployment upon need. This component relies on the distribution service at the platform tier for placing workloads at the designated edge locations. It shall interface with the "orchestration and distribution designer" for creating and updating the edge workloads to be deployed and redeployed upon need.

- **Edge Tier:** The solution mapping to the “Distributed Computing” subcomponent, enabling AI workloads handling on the edge. This allows to deploy, run, and redeploy AI workloads on the edge upon need. This enables processing at the edge and aided by the monitoring component, send concise and useful data back to the cloud-based operator to be able to analyze the state of the workloads and models.
- **Enterprise Tier**: this solution relies on the orchestration and distribution designer to feed the initial workload, handle incoming feedback, and provide updated versions of the workload to be redeployed upon need. Any problems in communication between the edge and the cloud will slow down this process. Thus, there is a need to take into account the velocity and heterogeneity of data.

2.3 Technology background:

The main background technology we rely on is the Open Cluster Management (OCM) https://github.com/open-cluster-management-io (OCM is open source under Apache 2.0 License). This technology aims to ease the distributed control over various Kubernetes clusters. OCM uses a hub and spoke architecture for scalability. The hub serves as a control point for all managed clusters. An agent is installed in the managed cluster to manage locally operations directed to it from the hub. We rely on this technology and augment it to handle AI models and workloads.

Red Hat Advanced Cluster Management (RHACM) is the commercial product built on top of the open source OCM. ACM controls clusters and applications from a single console using policies across multiple clusters at scale. Managing K8s nodes and clusters at scale is the main benefit provided and used by this solution. In addition, infrastructure for application lifecycle management is provided. We build on top of these capabilities features corresponding to models and AI workload lifecycle management. Deployment of applications based on policies and labels serves as an additional base capability we use and enhance to handle the artifacts coming out of this solution. The end result contributed to capabilities for automating the enterprise workloads management at scale. ACM provides the basis for edge deployment on scale, thus helping to integrate edge computing with a cloud architecture and environment.

The technology we developed for scalable multi-cluster management (hub of hubs) serves as a basis for the design at scale in a cloud - edge environment. The hub of hubs component is another example of enhancements added on top of OCM.

3. Implementation Status

3.1 Current implementation

The major goal of this task is to provide a scalable, flexible, and easy-to-use method for deploying AI workloads and their associated models in a scalable cloud- edge environment. Currently an advanced version of the solution has been developed, presented, and demonstrated to the consortium members, and during the first review. The current version relies on ACM and was demonstrated to work with various kinds of artifacts, using various kinds of orchestration engines, running on different hardware configurations. The current capabilities have been mutually agreed upon with the LRT solution owners, who are going to be the first solution to interact with this technology. The combined capabilities shall be used and demonstrated first in the FACTOR pilot.

The currently implemented capabilities are summarized below:

- A new AI workload or a new version of a running workload is pushed to the designated Git repository. The management component obtains this information and determines which are the suitable destinations to host the new workload.
- The designated target managed clusters are notified and proceed to pull the new resources from the Git repository.
- The local agent on the target machine takes care of the local deployment (or rolling update) of the new workload.
- Administrators get to edit and transmit to the management component a dynamic set of policies and labels to determine the appropriate target managed clusters.
- This capability further demonstrates a flexible manner in which to select targets and deploy artifacts on different managed clusters, using policies and labels (Demonstrate flexible placement, for example using labels).
- Placement rules based on labels have been further demonstrated such that a change in labels causes a changed placement and target deployment edge node change. Thus, an application may be deleted from a certain managed cluster, and on the other hand being deployed on a new managed cluster.
- Further capabilities include the distribution of an application to a group of edge nodes, and another to a different group of edge nodes.
- Workloads are typically deployed as a docker container.
- Once the capabilities have been demonstrated using a single target managed cluster, the demonstration has been expanded to support several managed clusters as target deployment options.

The bottom part of the list represents areas in which we have advanced since the previous version of this deliverable (D4.6 submitted at M18)

- Demonstrate joint capabilities with LRT solutions on the target FACTOR environment. Deploy AI workload on a machine close to the data source at the factory floor. The demonstrator show an instance created by LRT, which is pushed to a GitHub repository. The AI workload mechanism comes into play and determines the target managed clusters to be used. We see in the demonstrator managed clusters spanning multiple environment are put into play, including 2 cloud providers (IBM and AWS), a VM within the FACTOR server running in their environment, and a local VM running on our local environment.
- Demonstrate different underlying orchestrators (k8s, k3s). Different flavors of the orchestrator based on the host machine and its resources.
- Use k3s or other low footprint standard K8s as the edge cluster. Demonstrate that it works as good as a k8s installation.
- Allow the edge cluster to run in multiple kinds of target managed clusters, such as in a target VM provisioned for the pilot (specifically within the FACTOR premises), or in a container.
- Flexibly control separately models and workloads
- Basic capability to update workloads in some target nodes, and abide by a flexible target selection mechanism.
- Flexibility in distributing versions of the model and of the workload to different clusters.
- End-to-end Demonstration: Show dynamic, flexible, and scalable nature of the solution, including target selection and deployment of various artifacts in a flexible manner.

3.1.1 Solution features mapping with user requirements

Requirements from the i4Q[EW] solution are presented in the table below.

Entity	Requirement	i4QEW Features	Comments (Group of tasks)
Biesse pilot	Cloud-edge architecture	Provide the infrastructure for collaboration between the clod and the edge tiers.	Deploy-redeploy cycles
AI workload generating solutions (such as LRT) that wish to deploy their artifacts on the edge	Interface with different solutions. AI workload distribution.	Expose interfaces and tools for deployment of AI workloads on the edge (via GitOps). Policy based placement	Deploy-redeploy cycles.

Entity	Requirement	i4QEW Features	Comments (Group of tasks)
FACTOR pilot	Low latency interaction between the sensorized machines and the corresponding AI workload.	AI workloads deployment to the edge	Deploy-redeploy cycles
Additional solutions	Ability to choose the set of desired targets for artifacts being created by different solutions.	AI workloads deployment to the edge	Policy based placement

Table 1. Solution featured mapped to requirements

As can be seen in **Table 1** above, the requirements are mostly of the non-functional kind, even though some functional requirements are necessary to fulfil the requirement. The main requirements are geared towards the possibility to operate efficiently in real-time, reducing latency, enhancing data privacy and security as a whole, conserving network resources, while being able to operate also in a network disconnected mode. The function requirements, such as the support for flexible deployment of AI workloads at the edge are geared to fulfil such NFRs.

3.2 Role in the corresponding pilot

This solution shall be demonstrated as a part of the FACTOR pilot. The FACTOR pilot focuses on improving the efficiency of the manufacturing process, mainly in two dimensions, namely, the quality of manufactured pieces, and the overall efficiency of the production process. The main method is to incorporate more digitization instrumentation and measurable automation into the process. The end goals are to reach the zero defects in manufacturing goal (P4_BP01), as well as to reduce the number of times a machine must be stopped and the overall time machines are doing non-productive work (P4_BP02). To achieve these goals, quality control applications that monitor the production process and adjust it to achieve better efficiency are introduced. These applications typically employ AI/ML solutions to analyse the collected data and decide on the best actions. Due to a variety of reasons, including the real-time nature of the control process, security concerns, and connectivity requirements, the quality control applications should run on the edge, i.e., in the factory close to where the data is being generated. Management of the edge applications and the AI/ML models they require for their operation is an important aspect of the overall tasks required to achieve the pilot's goals.

There is a good match between the intention of the pilot and the capabilities provided by the AI on the edge support suite. At this stage we shall look at cloud native technologies for easier installation, flexibility, and edge friendliness. A first i4Q solution to make use of the AI on the edge components is the production Line Reconfiguration Tool (LRT), being developed by UPV. AI on the edge components shall support various kinds of artifacts, such as enable workloads to dynamically load required models from shard storage, or be wrapped in an http server as a microservice, deployed via helm charts, and so on. AI on the edge components shall cover a wide range of target scale, from full-fledged K8s / OpenShift environments to small scale such as k3s or k3d which are easily deployed, edge friendly, and can run on a single VM.

The AI on the edge incorporation will help in some of the fundamental aspects of the pilot such as the collection of large amounts of data, serving as an enabler and input for gathering

relevant parameters and interactions among them; recommending config and parameters changes. At the background there is a continuous feedback loop from the edge to the cloud to enable producing enhanced AI models, which in turn need to be redeployed to the correct target locations. The AI on the edge component is a key pillar for digitation and automation aspects of the foreseen pilot. The edge component can be a part of a solution to improve the manufactured product inspection process; increasing the rate in which such inspections are performed, improving the accuracy of the inspection, as well as being capable of predicting ahead of time that the process is not moving in the right direction and attempt to alter it in real-time, while the process is running, to avoid defects in the manufactured items and reduce the rate of machine stops. These processes can save a lot of time and money for the manufacturing entity. The AI on the edge can be an integral part of both the manufactured goods production and the whole production process improvement, focusing mainly on machine stops.

In the pilot, sensor data acquisition as close as possible to the CNC tool is supported. For example, the change in time of parameters such as tool temperature and power consumption is of interest. The AI on the edge components serve as an enabler for such an architecture. Another mode of interaction between the technology and the pilot calls for each time the machine is stopped, for the solution to take into account the cause of the stop and may cause readjustments to the model. This in turn will make use of the AI on the edge components for deploying the revised model to the required target nodes.

3.2.1 Demonstration

The current demonstration of this solution includes the following capabilities: The main goal of this demonstration is to illustrate the deployment of workloads and AI models to edge Kubernetes clusters or single nodes running some version of Kubernetes. Differential lifecycle management of workloads and AI Models (update, configure, delete) is supported as well. In the cloud, the main management entity controlling all operations is based on ACM / OCM, and servers as a central management of all participating edge clusters. Underlying orchestrators used are based on Kubernetes, and may run different variants based on the characteristics of the target nodes. We demonstrated the capabilities on K8s as well as K3s.

All capabilities are shown at the recorded demonstration.

3.3 Next developments

Enclosed are several avenues which we intend to pursue towards the final version of the software per se on M24, and including possible additional development within the project life-time, as a part of WP6, in the integration phase and as a part of the generic pilot activities.

These items need to be prioritized thus not all activities shall be implemented and demonstrated within the confines of one of the pilots.

- Flexibly control separately models and workloads – add more flexibility to the differential deployment tooling.
- More advanced capability to update workloads in some target nodes, and abide by a flexible target selection mechanism.
- More advanced capability for flexibility in distributing versions of the model and of the workload to different clusters.

- Based on Git scale, a central hub may grab resources and deploy on managed clusters, or each managed cluster grabs directly from Git. Currently we employ a pull model in which the managed clusters pull themselves the required resources from Git. An alternative approach shall be considered in which the hub grabs the resources and distributes it to the selected targets.
- Work with OCM rather than ACM – open source vs the product version of it.
- Scalability tests – see how many managed cluster and managed entities we can sustain.
- Demonstrate scaling up the initial demonstration environment to multiple sites
- Connect to additional i4Q solutions and pilots: Integration: - adjust to collaborate with partners

3.4 History

Version	Release date	New features
V0.1	01/04/2022	Initial release, simple model deployment
V0.2	01/05/2022	Support multiple managed clusters as targets
V0.3	01/06/2022	Label based deployment
V0.4	30/06/2022	Integration in the FACTOR environment
V0.5	15/11/2022	Differential treatment of workloads and models
V1.0	15/12/2022	Refresh E2E demonstration

Table 2. i4Q[EW] History of versions

4. Conclusions

Capabilities, design and interactions supported by an AI edge component were detailed and demonstrated, for handling the distribution and deployment of AI workload within a hybrid cloud-edge environment, within the confined of a manufacturing facility. Such capabilities address many personas, for example developers, and administrators of edge platform vendors. Developers and administrators can make use of the edge at scale friendliness provided by the described capabilities. In turn this may attract edge platform providers, not only to make their platform more attractive to their customers, but also to enhance the utilisation of their underlying resources. Thus, Edge Software providers will enjoy enhanced application deployment and efficient life cycle management, edge-based AI application developers will be able to focus on the logic of their specific service and not worry about the heavy machinery at the infrastructure level that deals with placement and deployment issues in a complex target environment.

A first version of this technology has been used in the FACTOR pilot, paving the way for additional scalable deployment targets.

Appendix I

i4Q Edge Workloads Placement and Deployment (i4Q^{EW}) web documentation can be accessed online at: https://i4q.upv.es/14_i4Q_EW/index.html without putting the information in the deliverable as it is already in the web page.