A network architecture for scalable end-to-end management of reusable AI-based applications

Flávio Brito*, Josué Castañeda Cisneros*, Neiva Linder *, Roberto Riggio†
Estefanía Coronado‡ §, Javier Palomares‡, Jovanka Adzicl¶, Javier Renart‖,
Anders Lindgren**, Miguel Rosa††, and Per Ödling ‡‡
*Ericsson AB, Stockholm, Sweden Email: flavio.brito, josue.castaneda.cisneros, neiva.linder@ericsson.com
†Università Politecnica delle Marche Home, Rome, Italy Email: r.riggio@staff.univpm.it
‡i2CAT Foundation, Barcelona, Spain Email: estefania.coronado, javier.palomares@i2cat.com
§ High Performance Networks and Architectures, Universidad de Castilla-La Mancha, Albacete, Spain
Email: estefania.coronado@uclm.es
¶ Telecom Italia S.p.A., Turin, Italy Email: jovanka.adzic@telecomitalia.it
‖ ATOS Research and Innovation Department (ARI), Atos Spain S.A.E., Madrid, Spain; Email: fco.renart@atos.net
** RISE Research Institutes of Sweden AB, Stockholm, Sweden Email: anders.lindgren@ri.se
†† Aerotools, Madrid, Spain Email: miguel.rosa@aerotools-uav.es
‡‡ Lund University, Lund, Sweden; Email: per.odling@eit.lth.se

***Abstract*—Artificial intelligence (AI) is a key enabler for future 6G networks. Currently, related architecture works propose AI-based applications and network services that are dedicated to specific tasks (e.g., improving performance of RAN with AI). This approach offers a unique way to collect data, process it, and extract features from data for each AI-based application. However, this dedicated approach also creates AI-silos that hinder the integration of AI in the networks. In other words, such AI-silos create a set of AI-models and data for AI-based applications that only work within a single dedicated task. This single task approach limits the end-to-end integration of AI in the networks and brings costs for network operators. In this work, we propose a network architecture to deploy AI-based applications, at different network domains, that prevents AI-silos by offering reusable data and models to ensure scalable deployments and lower costs for operators. We describe the architecture, provide workflows for the end-to-end management of AI-based applications, and show the viability of the architecture through multiple use-cases.**

Keywords*—Reusable AI-based applications, E2E network management, AI-Native Networks, 6G

I. Introduction

Artificial intelligence (AI) has become a promising enabler for future 6G networks. Such networks will deploy many AI-based applications that leverage data to make autonomous decisions for all network layers. This improves automation and reduces the cost for network operators. Such is the goal of the so-called AI-Native networks [1], [2]. While the precise definition of the term AI-Native is still under discussion, the goal of integrating multiple AI-based solutions, where AI is a natural part of the functionality, in terms of design, deployment, operation, and maintenance, into the network is a common denominator between different related works [3]–[5]. Such type of networks requires efficient end-to-end management of these AI-based network services and applications. To this end, new network architectures have been proposed recently to realize the AI-Native vision.

Two architecture proposals have been published as a step towards AI-Native networks [2], [6]. However, both architectures use dedicated data pipelines for each AI-based application. This means that each AI-based solution has a unique way to collect data, process it, and extract useful features from such data. This conforms to the current usage of AI-based solutions where they focus on a single layer/application to optimize network resources [7]. This way of managing the lifecycle of such solutions creates AI-silos. These silos are the set of tools, features, and data that exclusively work only for a unique and dedicated AI-based application. For example, several AI-based applications have been proposed for RAN automation that exploits data to deploy RAN at different environments [8], [9]. In both works this data, required to train models for the AI-based application, is tailor fit to meet the specific conditions of a given deployment [10]. Such data comes from a dedicated data pipeline for each application. This means that data needs to be collected twice and each AI-based application (i.e., machine learning model) needs to be managed independently. This results in inefficient resource utilization for both the data and the model lifecycle management. To circumvent this inefficient deployment of AI-based solutions with AI-silos and dedicated pipelines, we propose a network architecture that exploits shared pipelines to ensure reusable network services and applications.

The architecture proposed enables scalable end-to-end management of AI-based applications. Unlike the previous published architecture papers that focus on either (i) a specific domain with generalized models, (ii) end-to-end domains with dedicated pipelines our paper considers shared pipelines for both data and models. This enables generalization of both data and models and prevents AI-silos. The proposed architecture enables collecting, processing, and delivering data and models to multiple applications. In other words, we reuse data and models to optimize resource consumption when deploying AI-based solutions in/for the network. To this end, we describe the architecture components, the workflow, and use-cases that illustrate the validity of our approach.

The paper is organized as follows. Section II presents the related work where we go more in depth about the dedicated pipelines solutions and how this approach does not align with expectations for future 6G networks. Section III describes the proposed architecture with all components to ensure end-to-end AI-based service deployments. Section IV provides workflows for AI-based deployment and migration using the architecture. Section V presents two use-cases that illustrate how the architecture enables future AI-based applications highlighting the benefits of having a shared data pipeline. Section VI concludes the paper and describes next steps. It is important to note that for the rest of the paper the term AI-based applications refers to applications in all layers of the network.

II. Related work

The next generation networks (e.g., 5G beyond, 6G) call for the so-called AI-Native paradigm where ”AI-native is the concept of having intrinsic trustworthy AI capabilities, where AI is a natural part of the functionality, in terms of design, deployment, operation, and maintenance [1]. Despite the term still being under discussion, it has set many expectations for future networks [2]. The new paradigm calls for an explosion of AI-based applications and network functions to support next generation services. Recently, several works have been published where authors try to close the gap between current AI paradigm, where many applications run simultaneously in each layer or per operation to a more data-driven and intelligence-driven network operation.

A high-level architecture to enable AI-Native paradigm in next generations networks was proposed [2]. The architecture considers resource, security, function, capability exposure, and orchestration layers. For intelligence in the network, it considers hierarchical and distributed one, in-depth convergence and connection, and a collaborative exposure of connection and intelligence capabilities. This includes the self-* properties of the network (e.g., self-configuration, self-management, self-optimization) and exposure for AI capabilities and services for third-party applications.

Another architectural work explored the close-loop automation framework to integrate different layers in different domains, such as the Far and Near Edge [6]. The architecture considers decentralized close loops to integrate AI in 6G via network intelligence orchestration layer. With this layer, the architecture coordinates the network intelligence instances deployed across the end-to-end infrastructure, including beyond edge micro-domains. This approach ensures the correct execution of closed-loops in the multiple domains, as each AI-based application can have its lifecycle managed by the central orchestrator.

While both previous architectures consider the tighter integration of AI-based application and services to the network, there are some limitations. For the first work, the authors describe only at a high-level the properties of an architecture without describing the components that would enable such architecture. For example, how to balance the training cost and performance by applying intelligence in the network [2]. The second work also describes the design principles of the intelligence orchestration level at a high level, but leaves aspects as privacy and security for further discussion [6]. Moreover, both works consider dedicated data and AI models pipelines for each AI-based application. This means that each AI application is self-contained by obtaining data and processing it according to the AI model, creating an AI-silo. In other words, when a user needs to update the model, new data will be collected, processed, and used in a training phase. This has to be done for every AI-based application; thus, this approach is not scalable nor efficient since resources tend uniquely assigned to a single model application. This in turn becomes costly for operators. For example, handling thousands of AI-based network applications increases the operator's cost, as they need to collect new data for every application. We propose an architecture that does not consider such dedicated pipelines; and, by extension, AI-silos.

In our architecture we can reuse data and models to prevent AI-silos. For example, if two different AI-based network services need the same data but with distinct features (i.e., data processing is different), the architecture enables collecting the data only once. Furthermore, the architecture enables reusing models for similar applications. For example, an AI-based network service that runs in different edge domains will need to be slightly tuned to match the conditions of each domain. By reusing the models, an operator can reduce costs and time to deploy a service. This paper is a continuation of one previously published [11]. In that paper we introduced the concept of Artificial Intelligent Functions (AIF). The AIF enables AI-based network services/applications by encapsulating them inside the AIF. The services communicate using the AIF well-defined interfaces, as described in the framework presented in the paper. In this paper, we present the detailed architecture, with each component, for supporting AIFs enabling AI-based network services. For the AIF, we also introduce the AIF descriptor that describes how to deploy and manage the AIFs. This includes functional and non-functional requirements related to AI, data management, and hardware acceleration. Additionally, we describe workflows that describe how the architecture supports AIF deployment and migration to support multiple use cases.

Next, we describe our proposed architecture.

III. AI@EDGE architecture

The AI@EDGE architecture supports artificial intelligence in two approaches: “in-platform” and ”on-platform”. The ”in-platform” approach enables better usage of infrastructure resources through the network and service automation of AIFs. The ”on-platform” approach enables better end-user quality of experience through end-user application intelligence AIFs. In other words, we consider AIFs both for the network and application layer. To this end and to manage the AIFs, we propose an AIF descriptor.

The AIF descriptor captures the information related to the orchestration and lifecycle management. It describes the rules and requirements of an AIF module and considers the aspects related to deployment and management of MEC applications and the AI lifecycle management (e.g., collecting data, training models, updating models). Thus, the AIF descriptor contains a MEC descriptor, an operation management descriptor (e.g., a Helm Chart for Kubernetes

Network Function), and an AI-specific domain descriptor. The later considers deployment information, such as required environment (e.g., computation properties like CPU/GPU, cluster, and nodes properties), advertised metrics and KPIs, and data/model requirements.

All the previous requirements are managed by the different components of our proposed architecture. Thus, the architecture supports end-to-end system orchestration and management of third-party AIFs, as specified in life-cycle management workflows. Such orchestration is supported by a data pipeline that preserves privacy and security. To enable such functionality, the AI@EDGE architecture is composed of two layers: the Network and Service Automation Platform and the Connect-Compute Platform, as shown in Fig. 1. The first layer groups the components that provide the means to properly control and optimize the performance of the MEC and 5G Systems deployed at the near and Far Edge. The second layer combines cloud computing and virtualization, hardware acceleration, and a cross-layer, multi-connectivity-enabled disaggregated RAN into a single platform. Next, we describe the main components of the AI@EDGE architecture to support AIFs.

A. Network and Service Automation Platform

The Network and Service Automation Platform (NSAP) provides optimization and intelligence for the management of AIFs. For example, for onboarding and instantiation of AIFs, the Multi-Tier Orchestrator, and the Intelligent Orchestration Component, will efficiently allocate the resources needed by the AIFs. Additionally, the NSAP adds intelligence via the Non-Real Time RAN Intelligent Component which provides Non-Real Time intelligence in a RAN domain. We describe the NSAP in more detail in the following text.

The Multi-Tier Orchestrator (MTO) is the NSAP entry point for onboarding and instantiating AIFs. It enables communication with different orchestrators, such as MEC orchestrators and cloud based NFV orchestrators. It can also issue orchestration operations to nodes in different locations of the network, such as instantiation, migration, as specified in the AIF descriptor. For example, when the MTO receives a migration request it checks the AIF descriptor to meet the requirements present in the descriptor. Such requirements are the input to the Intelligent Orchestration Component (IOC) to seek the most suitable solution to the request. The IOC leverages functionalities of fault, security, and resource management for AIFs in run time. Additionally, the IOC can select the best placement of AIFs which require hardware acceleration. Once the decision is provided by the IOC, the MTO proceeds with the deployment process to the selected destination, communicating to the specific MEC Orchestrator required. This division of responsibilities between the MTO and IOC ensures an extensible and native intelligence for network management by the synergy of responsibilities: The MTO keeps track of the status of edge systems while the IOC offers an intelligent decision by levering on AI/ML models. The lifecycle management can be for a single network service or a slice. For the latter case, the Slice Manager supports the MTO.

The Slice Manager provides control over the lifecycle of network slices in the AI@EDGE platform. It creates slice instances and controls their lifecycle over multiple MEC systems and 5G systems. It enables Create, Read, Update, and Delete operations over slice instances, as defined in a slice descriptor. The architecture considers two ways of slicing: implicit and explicit. The implicit slicing follows a best-effort approach without guarantees. The explicit slicing uses a descriptor to define the main requirements of a slice, being able to pre-allocate resources at desired MEC systems, guaranteeing performance KPIs at this level. In addition, the Slice Manager relates the SLA requirement with the logical sliced network. To do so, it interacts with the MTO and RAN controllers to allocate needed resources for a slice. During this interaction, the IOC supports the decisions of the MTO; however, to meet the requirements up-to-date data needs to be available to the IOC. The architectural component that ensures such data is the Data Pipeline.

The Non-Real-Time RAN Intelligent Controller (non-RT RIC) is the key element to implementing non-RT intelligent closed-loop automation related to the 5G System at the NSAP level, and to manage the 5G System Platform level. The AI@EDGE architecture is aligned with O-RAN's view on AI/ML, where the AIF and the rAPPS/xAPPs can be related [12]. The latter is enabled through the implementation of the A1 interface between the non-RT and near-RT RICs, which enables the definition of policies to control xAPPs. In addition, some of the O-RAN's workflows, such as the AI/ML, are implemented in the AI@EDGE through the Data Pipeline.

The Data Pipeline is responsible for delivering up-to-date and relevant data to AIFs executing on the AI@EDGE platform. Since the IOC executes several AI/ML models, the Data Pipeline also offers data to the IOC. Moreover, since data is only half of the equation for a successful AI-Native network platform, the Data Pipeline also considers the lifecycle management of AI/ML models (e.g., instantiating, updating, and replacing). This enables the AI@EDGE architecture to support multiple types of AI and machine learning, such as supervised learning or reinforcement learning. To match the requirements of these learning types, the Data Pipeline separates the responsibilities of lifecycle management into different components: The Data Collector, the Data Processor, the Data Repository, the Model Manager, and Model Repository. The Data Collector receives data from multiple data sources required to train models, it also offers authentication functionality to prevent spurious data. The Data Processor cleans, filters, and prepares data for the AIFs. It can also provide more data when updating and/or replacing an AI/ML model for the AIF. Since some data might require time to obtain, the processor can store data in the Data Repository. The Data Repository enables the re-usability of data by many models. Such models can specify the meta-data of the AIF descriptor. The data from either the Data Processor or Repository can be used for model training; however, due to the high variability of models, the AI@EDGE architecture does not contain a generic model processor as a component. The AI@EDGE architecture enables the lifecycle management of these models via the Model Manager. The Model Manager is responsible for the evaluation of performance for one or multiple models. Such evaluation is either periodic or by events, as specified in the AIF descriptor, and monitored by the Model Manager. If the performance is not satisfactory,

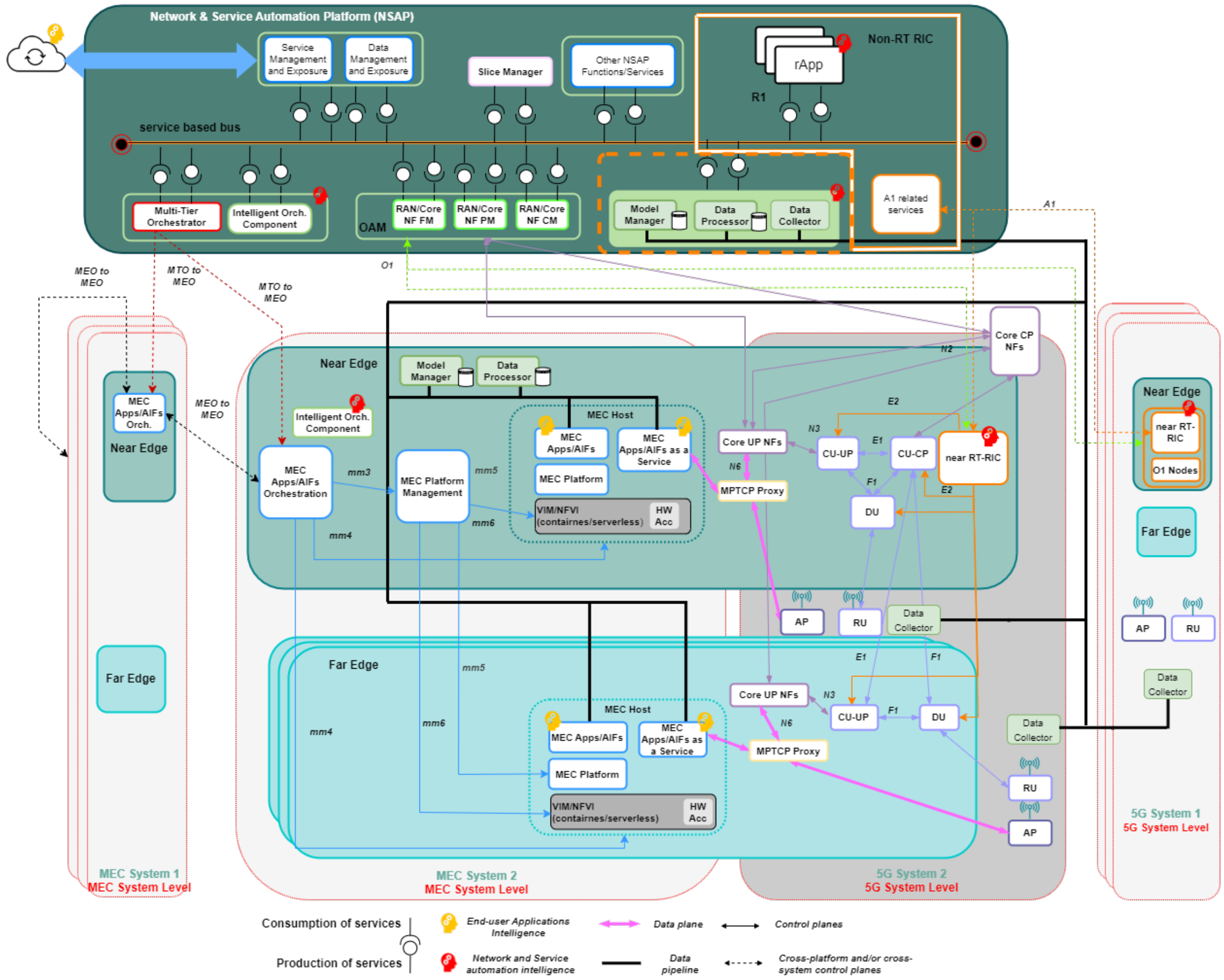

Fig. 1. AI@EDGE architecture to deploy AI-based applications. It considers Cloud, Near Edge and Far Edge domains.

the Model Manager triggers an update. Such update is stored in the Model Repository which contains metadata associated with the model. This enables model re-usability by different AIFs.

B. *Connect-compute platform*

The Connect-Compute Platform (CCP) facilitates the deployment and management of AIFs by using their descriptors. More specifically, the CCP ensures the necessary life-cycle operations for each AIF, and by extension of AI-based network services. To ensure this, the AI@EDGE architecture creates synergy between the NSAP entities and the CCP. For example, the Data Pipeline operates integrally with the CCP to ensure data is available at the AIFs. To ensure such integration with the NSAP, the AI@EDGE architecture groups the function by the CCP in two layers: MEC system components and 5G system components.

The European Telecommunications Standards Institute (ETSI) defines Multi-access edge Computing (MEC) standards as a network architecture concept that enables cloud computing capabilities for application developers at the edge of the network [13], [14]. Following the ETSI standard, the AI@EDGE architecture defines a MEC system with several blocks: the MEC Host, the MEC Platform Manager, the MEC APPs/AIFs Orchestrator (MEO), and either Near or Far Edge domains. The MEC Host offers the resources to deploy AIFs, including in as-a-service paradigm. It consists of: (i) a Virtualization Infrastructure Manager, responsible for the management of virtual resources, (ii) a MEC platform, which contains functionalities to run MEC applications on a particular infrastructure and enable to provide/consume MEC services, (iii) the MEC APPs/AIFs as a Service which exposes the APPs/AIFs to be consumed by a MEC platform or applications. The MEC Platform Manager is responsible for APPs/AIFs lifecycle management, FCAPS for the MEC platform, and network management (e.g., traffic rules and DNS configuration). The MEC APPs/AIFs Orchestrator manages the lifecycle of applications through the MEC platform manager, deploys AIFs, and selects appropriate MEC hosts for AIFs during instantiating such applications. The near and Far Edges contain the computing (virtual) infrastructure and, in the case of the Near Edge, hosts the main management entities of the MEC system.

The AI@EDGE architecture considers the network functions that form the virtualized 5G RAN and Core. To ensure interoperability with other standardization efforts, the 5G System is based on the O-RAN specification [12]. It enables network automation intelligence for the AIFs. These functions also are managed by the IOC in the Edge domain like at the NSAP.

The IOC will continuously monitor the different MEC systems, looking for anomalies or faults in specific AIFs and nodes. In case any anomaly or fault is detected in a system, the IOC could decide to migrate certain AIFs to balance the load between different nodes, always ensuring the minimum requirements of the AIFs specified in the descriptor are met. It is expected that the IOC includes several AI/ML models deployed in the architecture to solve placement or runtime issues in AIFs.

In the next section we describe how the AI@EDGE architecture components interact with each other through some workflow examples.

IV. AI@EDGE ARCHITECTURE WORKFLOWS

In this Section we describe the how to handle the end-to-end management of AIFs by describing common scenarios such as instantiating, migrating, and updating an AIF.

A. Workflow to instantiate a new AIF

This workflow describes how to deploy an AIF in the AI@EDGE architecture. Fig. 2 shows how the deployment is done. It begins with the Operation System Support (OSS) that receives a request to instantiate an AI-based application. Then, the OSS sends the request, which contains an AIF Descriptor File (AIFD), to the Multi-Tier Orchestrator (MTO). Then, the MTO stores the AIFD in the Database, which, in turn, returns to the MTO the Descriptor File ID. After this process, the MTO will select the best MEC system for the AIF deployment, according to the AIFD. To do that, the MTO sends a request to the Intelligent Orchestrator (IOC) to check what is the best MEC system to instantiate the AIF.

The IOC decides which MEC system is the most appropriate to deploy the AIF based on the status of each MEC system and the AIFD requirements. If a MEC system is chosen, then the IOC selects the Near Edge node of the respective MEC system to instantiate the desired AIF. This request is forwarded to the respective MEO and the MEC platform Management. The latter contacts the respective VIM to allocate the necessary resources. It is important to highlight that the AIF could be deployed on the Far Edge or the Near Edge depending on the requirements and the available resources. In this example, the AIF was deployed on the Near EDGE. The process to deploy it in the Far Edge is the same. If no option in the Near Edge is found, then the AIF should be deployed in the cloud. The process is similar as described in Fig. 2. But the NFVO located in the cloud is the entity that will receive the MTO request. Then, the NFVO will contact the VIM in the cloud to allocate the necessary resources.

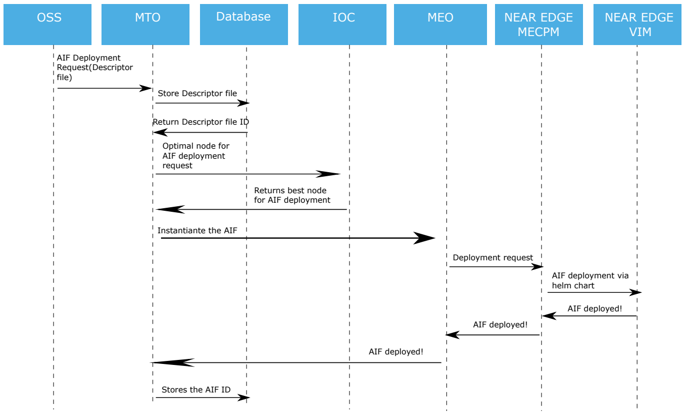
OSS
MTO
Database
IOC
MEO
NEAR EDGE MECPM
NEAR EDGE VIM
AIF Deployment Request(Descriptor file)
Store Descriptor file
Return Descriptor file ID
Optimal node for AIF deployment request
Returns best node for AIF deployment
Instantiante the AIF
Deployment request
AIF deployment via helm chart
AIF deployed!
AIF deployed!
AIF deployed!
Stores the AIF ID

Fig. 2. AI@EDGE AIF instantiation workflow.

B. Worfklow for AIF Migration to a different MEC system

After the AIF deployment, it is necessary to monitor the MEC system available resources. If, for some reason, the MEC system does not have the necessary resources to run the AIF anymore, the migration should be done. This case is illustrated by the workflow presented in Fig. 3 which describes on the left the cloud, in the middle square the MEC system 1 and in the right square the MEC system 2. The AIF is initially deployed in the MEC system 1.

The workflow starts with the VIM of the MEC system 1 sending the status of the resources to its MEO. The MEO realizes that the current MEC system cannot support the AIF and contacts the MTO requesting to migrate the AIF. The MTO then contacts the IOC to compute what is the best MEC system to migrate the AIF. The IOC starts this process by collecting the status of all MEC systems and then a specific MEC system is chosen which is, in this example, the MEC system 2. The IOC then sends to the MTO is the corresponding MEC system in which the AIF should be moved. The MTO then contacts the chosen MEC system by contacting the corresponding MEO. The MEO then contacts the MECPM that contacts the corresponding VIM of the Near Edge to allocate the necessary resources for the AIF deployment. After the deployment, a confirmation is sent to the MEO that forwards it to the MTO. The last step is the removal of the old AIF that was initially deployed in the MEC system 1.

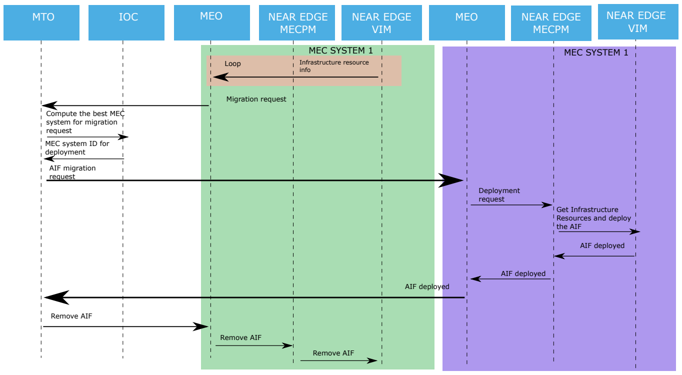

Fig. 3. AI@EDGE AIF migration workflow.

The migration workflow is a way to offer scalability to the proposed architecture as the resource management is optimized and, therefore, more AIFs can be deployed by using the workflow of Fig 2. However, for a good management of the available models inside the AIFs, it is important to handle the scenario where the model performance is decreased.

In the next workflows, we propose how the architecture supports models' replacement and retraining.

C. Workflow for Model Replacement

Fig. 4 describes how the proposed architecture supports model replacement for the AIF. It is important to note that the AIFD contains information to instantiate an AIF; however, it also contains indicators to be monitored about the AIF's state such as the prediction, performance, and confidence. The prediction contains the output of the model, the performance relates to the model, and the confidence indicates how precise a model is, respectively. This information is received by the Model Manager which associates metadata (such as a unique identifier and historical performance) to the AIF. With such information the Model Manager can ascertain whether the model running in an AIF needs to be replaced. Fig. 4 shows how to replace a model by comparing the current model's performance with the one in the model's meta-data. If the AIF's model underperforms, the Model Manager starts the replacement process.

The first step is to check in the Model Repository if there is a compatible model for replacement. By model replacement, we mean that there is a compatible model stored in the database that can be used to replace the current model. This replacement avoids training a new model from scratch which can take a long time. In this scenario, there is a compatible model. Then the Model Manager request to the current AIF to replace the model. After this replacement, the AIF register the new model with the necessary metadata and this information is stored in the Model Database.

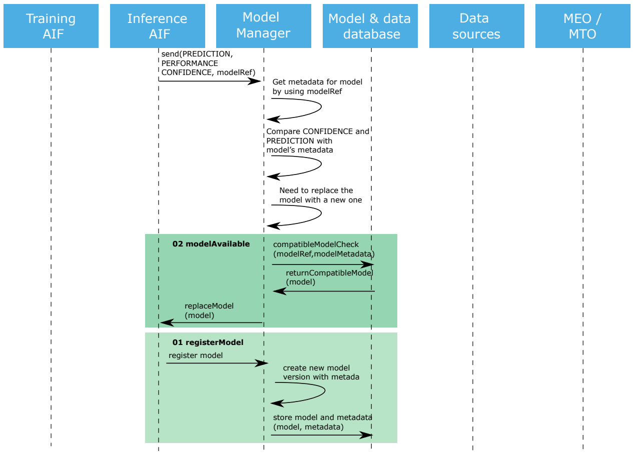
Training AIF
Inference AIF
Model Manager
Model & data database
Data sources
MEO / MTO
send(PREDICTION, PERFORMANCE CONFIDENCE, modelRef)
Get metadata for model by using modelRef
Compare CONFIDENCE and PREDICTION with model's metadata
Need to replace the model with a new one
02 modelAvailable
compatibleModelCheck (modelRef,modelMetadata)
returnCompatibleModel (model)
replaceModel (model)
01 registerModel
register model
create new model version with metada
store model and metadata (model, metadata)

Fig. 4. AI@EDGE Artificial Intelligence Function model replacement workflow. In this scenario a compatible model is found

The compatible model for replacement might not be available. Then, the Model Manager needs to trigger the model retraining process. This process is illustrated in the green box of the Fig. 5. After a compatible dataset is found, the Model Manager requests the Orchestrator to deploy a training AIF to process the compatible dataset. After this deployment, the model is trained, instantiated, and the old model replaced. Then, the new model is registered in the model database for future use.

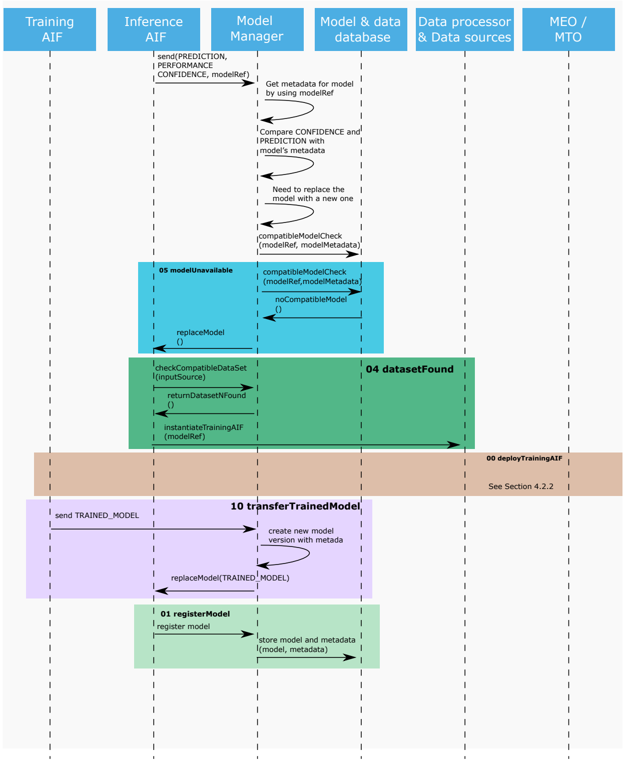
Training AIF
Inference AIF
Model Manager
Model & data database
Data processor & Data sources
MEO / MTO
send(PREDICTION, PERFORMANCE CONFIDENCE, modelRef)
Get metadata for model by using modelRef
Compare CONFIDENCE and PREDICTION with model's metadata
Need to replace the model with a new one
compatibleModelCheck (modelRef, modelMetadata)
05 modelUnavailable
compatibleModelCheck (modelRef,modelMetadata)
noCompatibleModel ()
replaceModel ()
checkCompatibleDataSet (inputSource)
04 datasetFound
returnDatasetNFound ()
instantiateTrainingAIF (modelRef)
00 deployTrainingAIF
See Section 4.2.2
10 transferTrainedModel
send TRAINED_MODEL
create new model version with metada
replaceModel(TRAINED_MODEL)
01 registerModel
register model
store model and metadata (model, metadata)

Fig. 5. AI@EDGE Artificial Intelligence Function model replacement workflow. In this scenario no compatible model is found.

The workflows in Figures 4 and 5 detail how the proposed architecture supports reusability of AI-based applications. By reusability, we mean AI-based applications reuse models and data by the architecture's shared pepipeline. The data and model pipeline presented supports feature extraction for similar applications. This means that the data can be collected once, and different features can be extracted to be consumed by similar applications. This avoids AI-silos, as both data and model are shared in our architecture. Furthermore, the migration workflows presented in Fig. 3 also provides scalability as an intelligent resources management can leads to more AIFs to be used in the network.

Next, we illustrate how AIF applications can be deployed using the proposed architecture through two uses cases that highlight the end-to-end service management.

V. Use cases

The proposed architecture supports a secure and reusable artificial intelligence platform for edge computing in beyond 5G networks. This includes a framework for closed-loop network automation that supports flexible and programmable pipelines for secure and reusable AI models, as well as a connect compute platform to create end-to-end slices supporting a diverse range of AI-enabled network applications. To this end, we highlight two use cases that illustrate how our proposed architecture supports AI-based applications and services facilitating the end-to-end management.

A. Use case 1: Virtual validation of vehicle cooperative perception

The use case considers multiple vehicles exchanging trajectory data in a roundabout, where traffic fluidity and safety are paramount. This is gathered at the network edge, and it will be used to build a view of the roundabout environment, using a digital twin that considers the roundabout with autonomous and human-driven vehicles, all supported by AIFs. The multiple AIFs will predict potential collisions and dangers from both autonomous and human-driven vehicles. Since this is a complex and costly scenario in the real world, the proposed architecture will support a digital twin with cooperative perception in the context of real and emulated vehicles. This means that the proposed architecture will support and optimize communication between human-driven and autonomous vehicles. Each autonomous vehicle drives with a reinforcement learning agent to coordinate the actions (e.g., driving, communicating). However, since the local execution of AI algorithms with direct vehicle to vehicle (V2V) has limitations, such as the possibility to identify and solve complex traffic situations like in a roundabout, the computing architecture will be required to ease the limitations of V2V. In other words, the vehicles will coordinate trough the network (i.e. the edge) in a vehicle to network approach (V2N).

The change of communications perspectives from traditional V2V short-range (e.g., 802.11p) to the V2N long-range (e.g., 5G) using the MEC platform of the proposed architecture adds latency in V2V communications but allows for wider communications between the vehicles. This means that the networks offloads coordination functions, deployed as AIFs, that allow for a more complete solution regarding scenarios such as the roundabouts. For this use case latency in the V2N communications will be under 2000ms, with positioning of 1.5 meters to deal with vehicle dynamics and movement. The vehicle density considered is 12000 vehicle per square kilometer. This means that a digital twin deployed with the architecture can simulate that density of vehicles. To this end the testbed for the use case is described next.

The use case considers two testbeds connected to the proposed architecture (i.e., the AI@EDGE architecture). The first testbed relies on the driving simulator at POLIMI in Milano connected to the AI@EDGE through a 5G Telematics BOX. This ensures that the simulator can send dynamic data to an Edge Node on which the Cooperative Perception AIF will execute. The second testbed is the validation site at CRF in Torino where a 5G emulator tests the 5G enabled automotive Telematic Boxes. This allows sending data to the Cooperative Perception AIF through the 5G emulator. It is important to note that the proposed architecture not only supports the Cooperative Perception AIF but also the orchestration, deployment, and migration MEC functions to support the roundabout scenario. The functionalities highlighted in the use case are local traffic outbreak on the edge, the extension of the digital twin with 5G connectivity, and the added intelligence to vehicles, through the AIFs, to coordinate maneuvers with both autonomous and human-driven vehicles.

B. Use case 2: Edge AI assisted monitoring of linear infrastructures using drones in Beyond Visual Line of Sight operation

The use case considers monitoring large areas of roads networks using drones in Beyond Visual Line of Sight (BVLOS) mode trough the 5G network supported by the proposed architecture. In this scenario, reliability and fluid data traffic are required to send telemetry, image, and video data with low latency. To support such requirements, the MEC systems based on AI and Edge Computing of the proposed architecture will support AIFs for: optimal monitoring, accelerating computational and modeling processes, and improving reliability and range of operation. This includes energy efficient functionalities to scan the infrastructure and environment, build a 3D model of the infrastructure, locate identified incidents, and send notifications to a human drone operator. Moreover, all data must send continuously to a central domain to improve decision making by the drone operator. Two AIFs are being deployed with the proposed architecture to support the automated incident detection: anomaly detection and 3D reconstruction AIFs.

The anomaly detection AIF can identify, detect, and locate anomalies in the videos acquired by the drone. To support such functionality the drone has on-board a data server that synchronizes data from the different sensors. The drone also includes a message bus broker to send synchronized data to the AIF. This synchronization is important for the anomaly detection; but it is fundamental for the 3D reconstruction AIF since images with their position and orientation need to have the same time reference. The anomaly detection AIF uses Detic as detection model and CLIP visual-language model for the search engine [15], [16]. The proposed architecture supports updating the Detic model through the Model Manager, as described in Section IV. This allows for accurate and up-to date anomaly detection model. When a drone operator wants to detect a concrete anomaly, such as "rolled truck on the road", the anomaly detection AIF will interpret natural language and identify when and where the anomaly described appears in the video sent by the drone. If such anomaly is detected, it will be sent to the 3D reconstruction AIF.

The 3D reconstruction AIF generates the third-dimensional model of the area where an anomaly has been detected. To achieve this, the drone will start flying around the area to generate a photo realistic representation of the environment by capturing position, orientation data at every instant of time. In other words, the operator will program the drone to perform a circular flight to obtain overlapped images. The communication functionality and management of the AIF is also supported by the proposed architecture. In particular, the outdoor scenario integrates both the 5Tonic and the proposed AI@EDGE architecture to provide assisted monitoring operation [17], [18]. This includes drone control communication (C2) and video latency of less than 50 and 100ms, respectively, and C2 signal packet loss of less than 1%. In addition, the support of lifecycle management of both data and AIF models by the proposed architecture.

C. How the AI@EDGE architecture supports more use-cases

The AI@EDGE architecture described in Figure 1 provides a data-driven approach with components such as the data collector, data processor, and different domain databases. This data-driven architecture provides data privacy by delivering the processed data to the desired AI-based network service that is consuming that data (i.e., the AIFs). Specifically, the data processor is the main element of the proposed system. The data processor manages all the data-driven aspects such as asking for the data, processing it and delivering it to the correct application. This allows reusing data for multiple uses-cases and applications by enabling efficient resource usage (e.g., collecting data once for many applications). This has synergy with scalability issues for next generation networks. Thus, to support the explosion of AI-based applications, the proposed architecture supports shared pipelines for multiple uses cases, instead of the traditional dedicated (data/model) pipelines for each AI-based network service. In our approach we provide data reusability but also AI model lifecycle management via the Model Manager, as described in Section III-A. This means that new use cases can reuse models from different applications with techniques such as transfer learning. For example, instead of having many different models for the RAN domain (which brings lots of complexity for network operators), our proposed architecture allows reusing general models that match closely the conditions of a specific RAN domain. Additionally, unlike related works that focus only on a single layer of the network, our proposed architecture considers different layers and domains, such as the Cloud, Edge, and Far Edge, simultaneously with current 5G and beyond functionalities. This means we support network functions and applications in end-to-end service, with artificial intelligence through standardized AIFs. Thus, the architecture supports new use cases from different verticals such as industrial IoT, aviation, among others.

VI. Conclusion

This paper presented the AI@EDGE architecture for end-to-end management for AI-based applications at the network and application layers. Unlike other works which consider dedicated pipelines that create AI-silos, our proposed architecture enables reusing data and models to ensure a scalable deployment of AI-based applications. We described how to instate, migrate, and replace such AI-based applications using the architecture's components. As next steps, we will implement a test bed and evaluate our proposed architecture supporting use cases for next-generation networks. This is a step towards the AI-Native network vision.

Acknowledgment

The authors would like to thank the AI@EDGE project. The AI@EDGE project (https://aiatedge.eu/) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N^{o} 101015922. This work has also been supported by the EU "NextGenerationEU/PRTR", MCIN, and AEI (Spain) under project IJC2020-043058-I.

References

[1] *Defining AI native: A key enabler for advanced intelligent telecom networks*, Accessed on May 22, 2023. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/ai-native.

[2] Y. Liu, Y. He, Y. Lin, and L. Tang, "Toward Native Artificial Intelligence in 6G," in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, IEEE, 2022, pp. 1–6.

[3] J. Hoydis, F. A. Aoudia, A. Valcarce, and H. Viswanathan, "Toward a 6g ai-native air interface," *IEEE Communications Magazine*, vol. 59, no. 5, pp. 76–81, 2021. DOI: 10.1109/MCOM.001.2001187.

[4] A. Moubayed, A. Shami, and A. Al-Dulaimi, "On end-to-end intelligent automation of 6g networks," *Future Internet*, vol. 14, no. 6, 2022, ISSN: 1999-5903. DOI: 10.3390/fi14060165. [Online]. Available: https://www.mdpi.com/1999-5903/14/6/165.

[5] W. Wu, C. Zhou, M. Li, *et al.*, "Ai-native network slicing for 6g networks," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 96–103, 2022. DOI: 10.1109/MWC.001.2100338.

[6] M. Camelo, L. Cominardi, M. Gramaglia, *et al.*, "Requirements and Specifications for the Orchestration of Network Intelligence in 6G," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, 2022, pp. 1–9.

[7] X. You, C. Zhang, X. Tan, S. Jin, and H. Wu, "AI for 5g: Research directions and paradigms," *Science China Information Sciences*, vol. 62, no. 2, p. 21 301, Feb. 2019, ISSN: 1674-733X, 1869-1919. DOI: 10.1007/s11432-018-9596-5. [Online]. Available: http://link.springer.com/10.1007/s11432-018-9596-5 (visited on 05/21/2023).

[8] D. Corcoran, E. Westerberg, H. Olofsson, *et al.*, "Ai-enabled ran automation," *Ericsson Technology Review*, vol. 2021, no. 10, pp. 2–12, 2021. DOI: 10.23919/ETR.2021.9904687.

[9] A. K. Bashir, R. Arul, S. Basheer, G. Raja, R. Jayaraman, and N. M. F. Qureshi, "An optimal multitier resource allocation of cloud ran in 5g using machine learning," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 8, e3627, 2019, e3627 ETT-18-0332.R2. DOI: https://doi.org/10.1002/ett.3627. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.3627. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3627.

[10] P. Soldati, E. Ghadimi, B. Demirel, Y. Wang, M. Sintorn, and R. Gaigalas, "Approaching ai-native rans through generalization and scalability of learning," *Ericsson Technology Review*, vol. 2023, no. 3, pp. 2–12, 2023. DOI: 10.23919/ETR.2023.10068317.

[11] R. Riggio, E. Coronado, N. Linder, *et al.*, "AI@EDGE: A Secure and Reusable Artificial Intelligence Platform for Edge Computing," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, IEEE, 2021, pp. 610–615.

[12] H. Lee, Y. Jang, J. Song, and H. Yeon, "O-RAN AI/ML Workflow Implementation of Personalized Network Optimization via Reinforcement Learning," in *2021 IEEE Globecom Workshops (GC Wkshps)*, 2021, pp. 1–6. DOI: 10 . 1109 / GCWkshps52748 . 2021 . 9681936.

[13] S. Kekki, W. Featherstone, Y. Fang, *et al.*, "MEC in 5G networks," *ETSI white paper*, vol. 28, no. 28, pp. 1–28, 2018.

[14] M. Etsi, "Multi-access edge computing (mec); framework and reference architecture," *ETSI GS MEC*, vol. 3, p. V2, 2019.

[15] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 350–368, ISBN: 978-3-031-20077-9.

[16] A. Raffio, D. Braga, S. Ceri, P. Papotti, and M. A. Hernandez, "Clip: A visual language for explicit schema mappings," in *2008 IEEE 24th International Conference on Data Engineering*, 2008, pp. 30–39. DOI: 10.1109/ICDE.2008.4497411.

[17] B. Nogales, I. Vidal, D. R. Lopez, J. Rodriguez, J. Garcia-Reinoso, and A. Azcorra, "Design and deployment of an open management and orchestration platform for multi-site nfv experimentation," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 20–27, 2019. DOI: 10.1109/MCOM.2018.1800084.

[18] B. Nogales, I. Vidal, D. R. Lopez, J. Rodriguez, J. Garcia-Reinoso, and A. Azcorra, "Design and deployment of an open management and orchestration platform for multi-site nfv experimentation," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 20–27, 2019.