UNICO I+D Project
6G-DATADRIVEN-02

6G-DATADRIVEN-02-E9

Revised System architecture

Abstract

This document presents the revised system design of the 6G-DATADRIVEN-02 architecture. It updates the main entities and building blocks present in the system architecture to procure automated and zero-touch deployment that leverages AI/ML mechanisms in Industry 4.0 scenarios. The document considers a pool of factory floors connected to a central cloud to perform Industry 4.0 related tasks using AI/ML and an autonomous orchestration loop. This deliverable focuses in specifying the federated AI techniques, data sources, interconnectivity, and process automation.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia

Document properties

Document number	6G-DATADRIVEN-02-E9
Document title	Revised System Architecture
Document responsible	Carlos J. Bernardos (UC3M)
Document editor	Constantine Ayimba, Carlos J. Bernardos (UC3M)
Editorial team	Constantine Ayimba, Carlos J. Bernardos (UC3M)
Target dissemination level	Public
Status of the document	Final
Version	1.0
Delivery date	30-11-2023
Actual delivery date	29-11-2023

Production properties

Reviewers	Antonio de la Oliva (UC3M)

Disclaimer

This document has been produced in the context of the 6G-DATADRIVEN Project. The research leading to these results has received funding from the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D programme.

Table of Contents

List of Figures

List of Acronyms

5GC: 5G Core

AI: Artificial Intelligence

API: Application Programming Interface

ARIMA: Autoregressive Integrated Moving Average

B5G: Beyond 5G

DT: Digital Twin

IoT: Internet of Things

LSTM: Long Short Term Memory

ML: Machine Learning

NR: New Radio

RU: Radio Unit

SFC: Service Function Chain

SDN: Software Defined Network

SQL: Structured Query Language

PNF: Physical Network Function

VLAN: Virtual Local Area Network

VNF: Virtual Network Function

Resumen Ejecutivo

Este documento describe el diseño actualizado del sistema para la arquitectura en 6G-DATADRIVEN-02, aprovechando la inteligencia artificial para operar infraestructuras en industrias conectadas. Utilizando la arquitectura inicial presentada en el entregable 6G-DATADRIVEN-02-E8 como base, realizamos ajustes funcionales para mejorar el funcionamiento del sistema. La arquitectura revisada combina a la perfección la inteligencia artificial con herramientas de recopilación de datos para agilizar las tareas de mantenimiento y producción en entornos industriales conectados.

Los principales resultados descritos en este entregable son:

- el diseño revisado de una entidad de inteligencia artificial en industria conectada;
- el diseño de una entidad destinada a la colección de métricas en entornos industriales;
- el entrenamiento federado de modelos de inteligencia artificial para mejorar el rendimiento de las tareas de producción y mantenimiento;
- el registro automático de dispositivos para integrarlos en el proceso de producción; y
- la automatización del despliegue de servicios de industria conectada.

De acuerdo con la arquitectura actualizada presentada en este documento, hemos realizado investigaciones adicionales, aprovechando el poder de la inteligencia artificial. Como ya se mencionó en el entregable 6G-DATADRIVEN-02-E8, las siguientes publicaciones corresponden a estos trabajos:

- una solución que permite usar inteligencia artificial para mitigar la interferencia inalámbrica en el control remoto de un brazo robótico (Milan Groshev J. M.-P., 2022) (Milan Groshev J. S.-P., 2022); y
- la formulación del problema de despliegue de servicios de robots para entornos de industria conectada (Applications, 2022).

El resto del documento está redactado en inglés, de cara a maximizar el impacto del trabajo realizado en este proyecto.

Executive Summary

This document outlines the updated system design for the architecture in 6G-DATADRIVEN-02, leveraging artificial intelligence for managing connected industries. Using the initial architecture presented in the 6G-DATADRIVEN-02-E8 deliverable as a baseline, we make functional enhancements to improve the operation of the system. The revised architecture seamlessly combines artificial intelligence with data collection tools to streamline maintenance and production tasks within connected industry settings

The main results described within the deliverable are:

- the updated design of an artificial intelligence entity for connected industry;
- a data collection entity designed to collect metrics in an industrial environment;
- the training of federated models of artificial intelligence to boost the production and maintenance tasks;
- the automatic registration of devices to integrate them in the production pipeline; and
- the automation of connected industry service deployment.

In accordance with the updated architecture presented in this document, we have conducted research on the connected industry, harnessing the power of artificial intelligence. As already mentioned in the 6G-DATADRIVEN-02-E8 deliverable, the following publications correspond to this work:

- a solution to mitigate the Wireless interference of remotely controlled robotic arms (Milan Groshev J. M.-P., 2022) (Milan Groshev J. S.-P., 2022); and
- the formulation of the problem related to the deployment of robotic services in connected industry (Applications, 2022).

1. Introduction

This document revises the proposed system architecture of 6G-DATADRIVEN-02. It revisits the components that enable zero-touch deployment of management of Industry 4.0 through AI/ML and updates selected components to make them more apt to the task.

The overall architecture considers a pool of connected factory floors, each with a modicum of autonomy. These are then connected to a central entity that performs a coordination function. Individual factory floors, considered as edge clouds, can exploit the data from end IoT devices to facilitate localised training of AI models and send models weights to the central entity. This document therefore defines a cooperative Edge and Cloud architecture that exploits 5G and B5G connectivity to stitch Industry 4.0 devices, AI/ML and data storage entities together in the Edge and Cloud.

This revised document focuses on federated learning for the AI processing architecture as the more sustainable, modular approach applicable to industry 4.0 use cases. It specifies the pipelines and constituent components that handle data coming from myriad sources and identifies the technologies that are available to perform such tasks.

Additionally, this document presents an architecture that is designed to achieve zero-touch automation of Industry 4.0 services. The proposed architecture autonomously allocates computing and network resources to run the trained AI/ML models that perform Industry 4.0 tasks. Moreover, upon the authentication of a new device on the factory floor, the proposed architecture detects it and triggers re-orchestration to include it in the production pipeline.

2. Updates in this revision

In this revised design, we include a message broker solution (Apache kafka) to handle the communication within the factory pool and the cloud central entity. This makes the data exchange more scalable since interfaces do not have to be defined from each factory floor on a peer to peer basis but only to the common data bus. This solution also reduces latency in the exchange of data between elements.

Furthermore, we eschew centralised AI training for the more modular federated learning which is privacy preserving. This further reduces the load on the centralised entity by reducing the amount of data that is exchanged with the connected factory floors.

3. Revised system architecture

In this section we describe a general system architecture to exploit the usage of distributed data in industrial environments.

We consider a pool of *factory floors*, equipped with a host of IoT and industrial devices as actuators, surveillance cameras, sensors, robots, etc. Connectivity within and between factory floors is enabled through 5G/B5G. Through the *5G/B5G connectivity*, IoT devices exchange production process data to Edge compute resources within the factory floor. The IoT data is gathered to infer management actions using *AI* methods. For example, the sensor data from a rotating machine may indicate that the it is vibrating more than usual an AI method/algorithm may then predict that it will fail prompting the decision to stop it. Harvested data could also feed a Digital Twin (*DT*) of the factory floor to improve process planning before live implementation.

At a higher level of the hierarchy, each factory can belong to a *pool of connected factories* that cooperate to improve machine behaviour prediction models, thereby enhancing the preventive maintenance procedure. The inter-factory connectivity also allows data exchange and coordination between local DTs and AI algorithms at each Factory Floor.

Additionally, every factory floor within the pool shares data with a *Central cloud entity* that has a global view of the Factory pool. The Central cloud entity is useful to do a global monitoring of a company/cooperative that controls a pool of factory floors. As the Edge premises of each factory floor, the Central cloud entity has a DT that has a global view of the factory pool status; and it also has a holistic AI that can cooperate with the AI of each factory floor to take smart decisions within the whole factory pool.

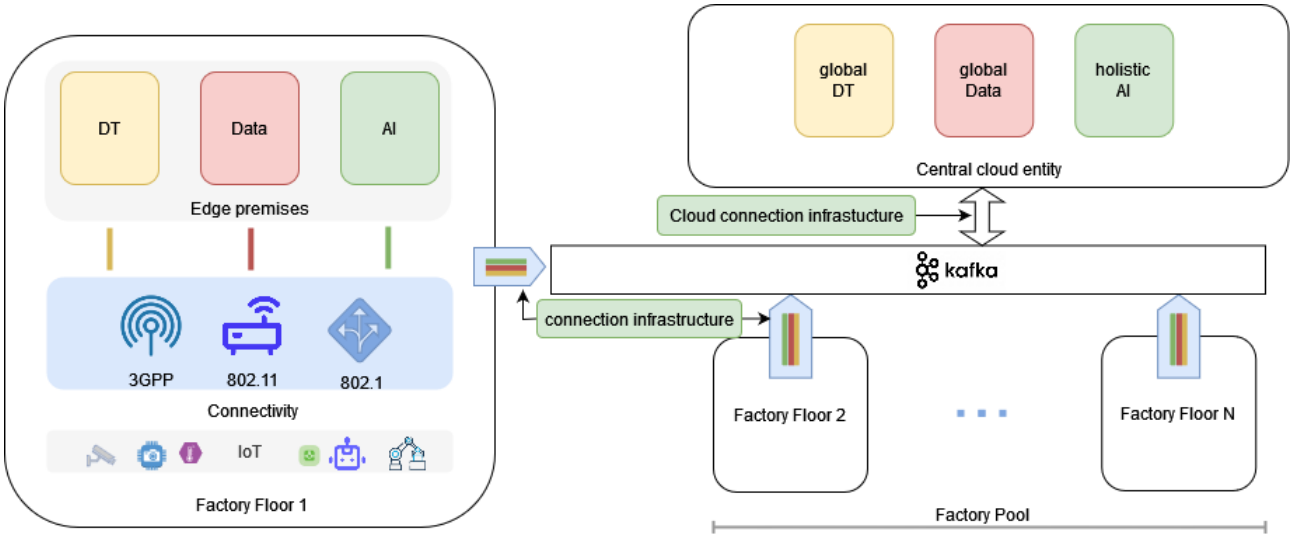

FIGURE 1: REVISED SYSTEM ARCHITECTURE

Figure 1 provides an overview of the general system architecture, that is, the main building blocks that compose the architecture for the exploitation of distributed data in industrial environments. In the following we enumerate and detail each element/entity within the general system architecture:

- *Factory floor*: we refer to a physical building where the machinery of a factory plant is.
- *IoT pool*: each factory floor hosts a set of IoT devices as cameras, sensors or actuators. The industrial machinery also belongs to the IoT pool, for it also belongs to the set of devices that produce data and information to recreate the status of the factory floor.
- *5G/B5G connectivity*: we refer to the set of networking devices that connect the IoT pool to the internet. In particular, it consists of a set of radio devices as NR antennas with their respective 5G Core (5GC) stack to process the incoming traffic. Additionally, the 5G/B5G connectivity comprises all the switches, routers, Ethernet, and fiber links that interconnect the 5G/B5G stack with the Edge premises and Central cloud entity.
- *Edge premises*: it is the set of servers and NFS that hosting the DT, Data, and AI functionality. The Edge premises could either be located within the factory floor (On premise Edge), at the operator premises (Fat Edge), or at any other levels (local, regional, etc.). The aim of the Edge premises is to provide computational capabilities to tackle the data handling and AI tasks to automate the factory floor tasks.
- *DT* entity: it is the set of software entities that reproduce a DT of the Factory Floor. It ranges from simulation software, numerical models, and visualization tools that mimic the behaviour of the IoT pool elements performing industrial tasks, e.g., a robotic arm.
- *Data entity:* we refer to the data storage software and files that contain historic and up-to-date information of the reports produced by the IoT pool within the factory floor. As an example, the Data entity may contain a database storing the number of goods produced by a factory line, and what is the health-status of the robotic arms involved in the process.
- *AI entity*: it contains the whole pipeline to perform AI tasks, ranging from data pre-processing scripts, AI/ML models for different tasks, and training environments to increase the AI/ML models accuracy. For example, the AI entity may host the pipeline to perform classification tasks that help to decide whether the produced goods pass or not the quality control check.
- *Kafka Message bus*: It provides a common set of software interfaces through which data can be reliably exchanged among factory floors and the central cloud entity
- *Connection infrastructure*: it includes both the physical and logical internet connectivity between the factory floors. Specifically, it is the set of fiber links, switches, routers, VLANs and IPsec tunnels that allow the connectivity between factory floors within the same pool,yet preserving the data privacy and integrity throughout the communication. Thanks to the inter-factory connectivity is possible to establish a connection to exchange information/data among the DT, AI and Data entities of different factory floors.
- *Cloud connection infrastructure*: same as the inter-factory connectivity, but stitching the Factory floor and Central cloud entity elements. Consequently, the exchange of information over it allows the factory floor to feed the Central cloud entity with reports on the data, DT and AI models' status.

- *Central cloud entity*: it is the set of servers, and data storage resources that contain an holistic and general view of the factory pool. The cloud central is typically a pool of computing and data resources that are located at some cloud facility with elastic capabilities that can grow according to the computational needs of the factory pool.
- *Global DT entity*: it contains a synced DT of the factory pool, i.e., the entity hosts also the set of simulation, models and visualization tools that sync with the Dts of each factory floor to produce a global DT. The Global DT can mimic the current and even future behaviour of the whole factory pool, hence, resulting in a real-time estimation of the performance of the industrial facilities.
- *Global Data entity*: as its factory floor counterpart, the Global Data Entity is the set of data storage mechanisms and files that gather the IoT reports of all the factory floors. The Global Data entity may be a mirror of the whole factory floor Data entities, or contain only the information of interest for both the global DT and holistic AI.
- *Holistic AI entity*: as the AI entity within the factory floor, the Holistic AI is the pipeline that allows the data pre-processing, defines the set of models, so as the training tools. The Holistic AI can communicate with the factory floor AIs through the Factory-to-cloud connectivity, and use their partial knowledge to assess active learning and federated learning procedures. Hence, it takes advantage of partial learning stages of the factory floors to boost the performance of a Holistic AI that would foresee a vast casuistic consisting on the aggregated knowledge of all the factory floors AI.

The aforementioned general system architecture gives an overview of the high level entities involved in the revised draft of our system architecture. The interaction between the constituent elements of our system is illustrated in Figure 1.

In the following sections we enter in detail into the description of the AI processing architecture, and the Network architecture. Both architectures detail how the AI entity and 5G/B5G connectivity stitch the entities within the general system architecture.

4. AI processing architecture

In this section we describe the AI processing architecture running at the Edge premises of a factory floor. The AI entity of each factory floor retrieves the data coming from the IoT devices to extract, process, and exploit useful information for the factory floor maintenance and automation tasks. The main goal of the AI entity is to assess the tasks that it has designated within the factory floor. Such tasks have an associated target such as detecting a defect in the factory lane, and a metric to measure the accuracy of how AI performs the task.

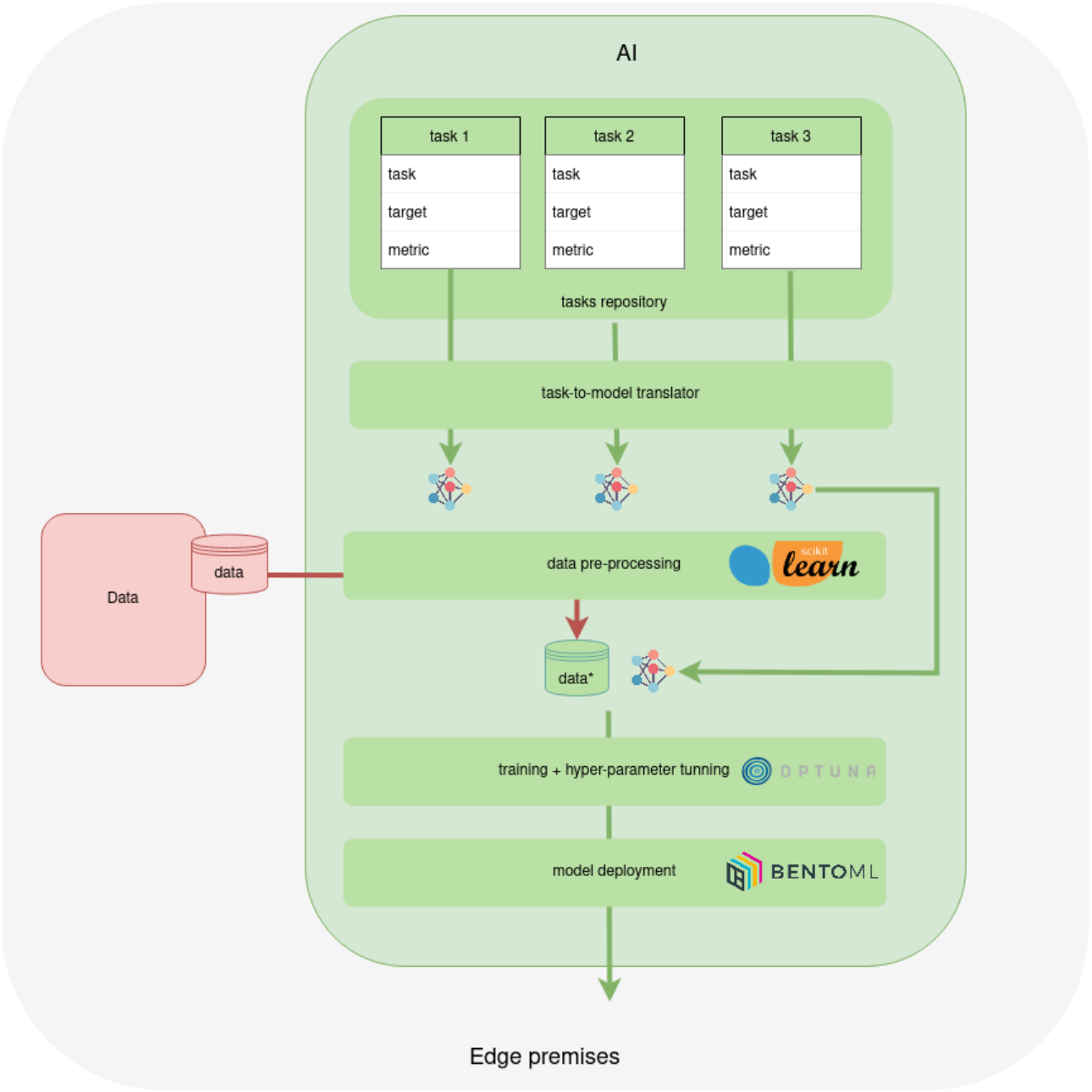

FIGURE 2: AI ENTITY ARCHITECTURE AT THE EDGE PREMISES OF A FACTORY FLOOR

Figure 2 illustrates the AI processing architecture building blocks. These are the tasks repository, task-to-model translator, data pre-processing, training+hyper-parameter tuning, and model deployment stage. These building blocks provide an end-to-end pipeline that takes as input data and tasks definitions, and deploys models to perform AI tasks in the factory floor.

In the following we describe each building block of the AI processing architecture:

- *Tasks repository*: it stores a list of tasks, each of them defined in a descriptor file (e.g., a YAML). The factory floor owner onboards in the tasks repository the list of tasks to perform at the factory floor. A task may be to do the quality check of the goods produced along a factory lane.
- *Task-to-model translator*: its duty is to translate a task into an AI/ML model that can perform such task. It achieves such translation by checking a list of pre-stored AI/ML models that suit the task to perform. For example, for a predictive maintenance task it will output a model as ARIMA or an LSTM neural network.
- *Data pre-processing*: prior to the execution/training of the AI/ML model it is necessary to do data pre-processing. This entity uses the scikit learn library to do data normalization, NaN filtering, and train/test splits. As a result, it produces a dataset that is ready to be fed to the AI/ML model.
- *Training+hyperparameter tuning*: it takes as input the pre-processed dataset and AI/ML model, and performs the corresponding training. Namely, it uses the optuna library to do hyperparameter tunning of the model. Optuna receives the metric to be optimized within the task, e.g., the F1-score to check whether the quality control accurately classified a piece with a problem within a production lane. Then, Optuna performs multiple training sweeps modifying hyperparameters of the chosen model, e.g., changing the number of LSTM neurons within a hidden layer of a neural network. The advantage of using Optuna is that it works with multiple AI/ML libraries for neural networks and statistics as pytorch, chainer, tensorflow, MXNet or Scikit-learn. Consequently, we can achieve an automated training process for a wide variety of AI/ML models for the factory floor.
- *Model deployment*: after the AI/ML model is trained inside the Training+hyperparameter tuning, the model deployment entity uses BentoML to put in production the trained model. Namely, a predictive maintenance ARIMA model is deployed by BentoML and receives HTTP requests with historic data to tell whether there will be an imminent failure or not.

Note that the AI architecture in Figure 2 shows the different steps within the Edge premises of the factory floor. However, Figure 1 illustrates that AI/ML entities of different factory floors may interact among them, and the cloud central entity have a holistic AI/ML that works over the pool of factory floors. In the following we detail both the interaction between AI/ML entities in the pool factory floors, and the holistic AI/ML in the cloud central entity.

4.1. Federated learning between factory floors

This section specifies how the AI/ML of different factory floors interact to achieve federated learning. By using federated learning, it is possible to preserve the privacy of the data produced by each factory floor while cooperating with others to achieve a common AI/ML model using the knowledge of all factory floors.

Figure 3 illustrates how the federated learning would work between the edge facilities of two factory floors. Both, factory floor 1 and factory floor 2, belong to a common pool for the exchange of information. For example, both factory floors may belong to an automotive consortium. Note that the Kafka bus in the figure is the same common bus for all factory floors depicted in Figure 1.

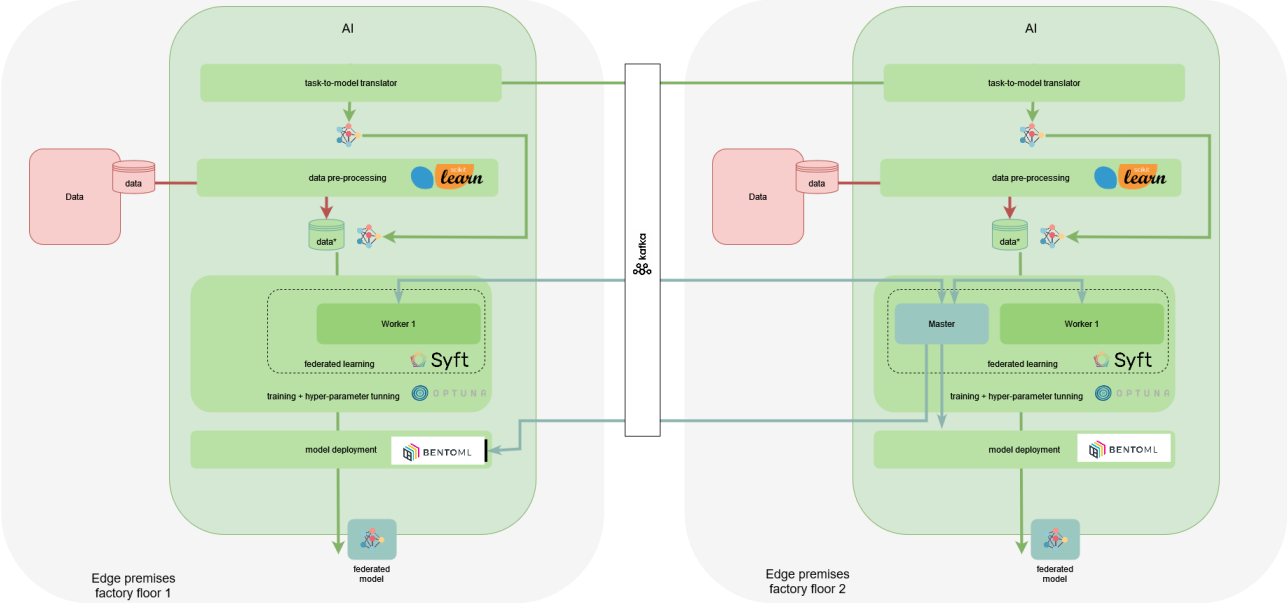

FIGURE 3: FEDERATED LEARNING BETWEEN FACTORY FLOORS

In the federated learning paradigm, multiple workers train the AI/ML model with local data, i.e., each factory floor trains at its edge premises a model using its own data. The training of each factory floor results in a local model that coordinates with other factory floors' models to share the learning. As a result, the federated learning yields a federated AI/ML model that is common to all factory floors although, yet with each being agnostic to others' data.

Figure 3 illustrates the federated learning with two factory floors involved in the procedure. With respect to Figure 4, we now enumerate all the steps (arrows) which explain the training stage:

1. the *task-to-model translators* of each factory floor agree on a common AI/ML model to use for a specific task. Such agreement happens over the inter-factory connectivity;
2. each AI entity uses the data locally stored within the factory floor, and selects the common AI/ML model agreed by both factory floors;

3. one of the factory floors is selected as the master in the *federated learning* training procedure (factory floor 2), and triggers the training stage at the *training+hyperparameter tuning* entity using the Syft library;
4. the master gathers the weight updates of each worker at every epoch, and average them to send weight updates to each worker. Note that at each epoch of the training stage optuna keeps track of the model performance to assist the master in the hyperparameter tuning;
5. once the training finished, the master forwards the common federated model to the *model deployment* entity of each factory floor, which deploys with BentoML and HTTP API to interact with the resulting AI/ML model.

Figure 4 illustrates the training for two factory floors. However it is worth remarking that the procedure applies for as many factory floors as needed. If all factory floors belong to the same pool, the master factory floor can synchronize with all of them to issue the federated training. The only difference is that it would have N workers, rather than the 2 workers of Figure 4

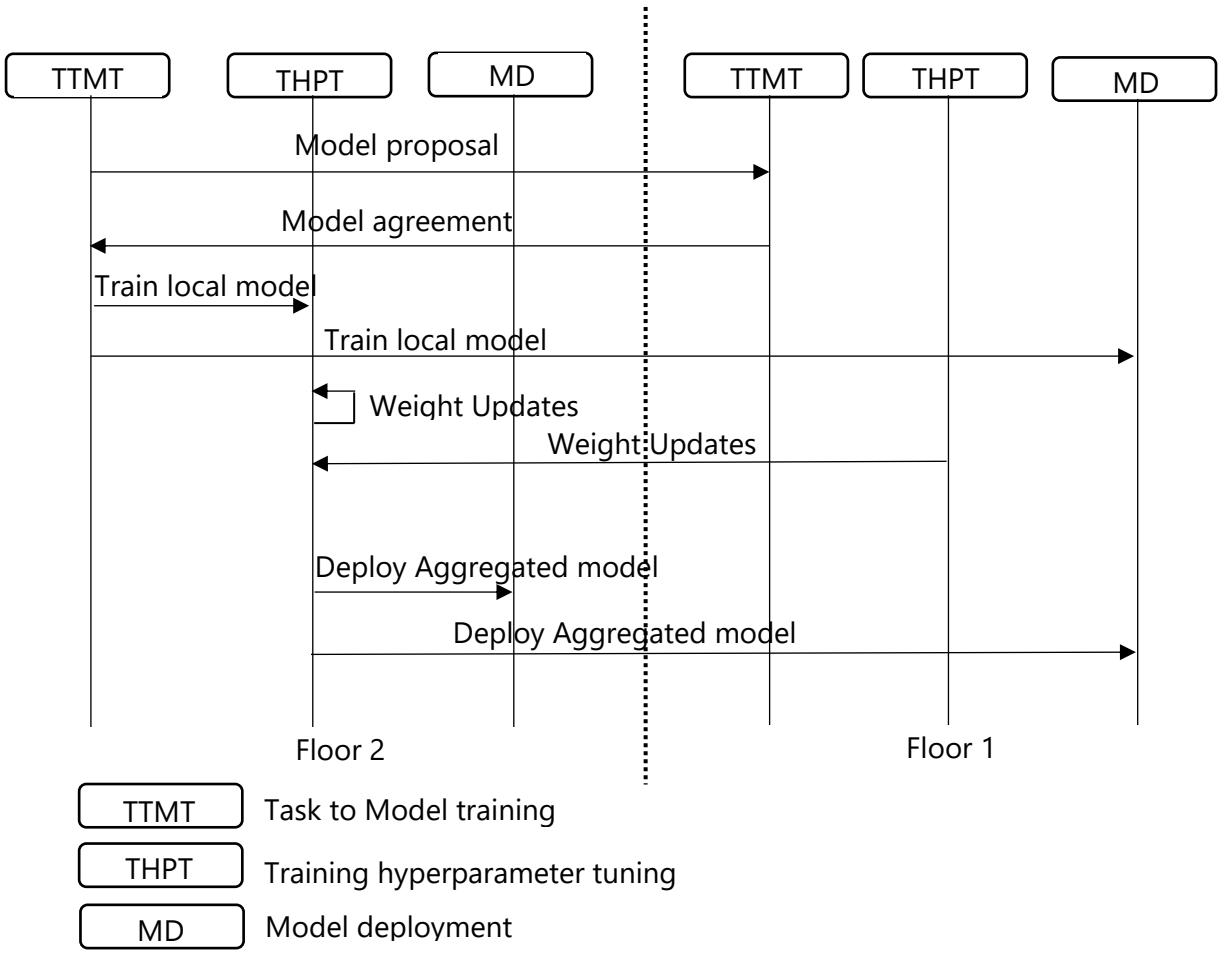

FIGURE 4: SEQUENCE DIAGRAM OF FLOOR TO FLOOR AI/ML TRAINING

4.2. Central cloud entity holistic AI

The cloud central entity has a holistic AI that serves for the purpose of training AI/ML models for all the factory floors. The central cloud entity is aptly be hosted at the headquarters with administrative control over all the factory floors. Individual factory floors are equipped with a modicum of compute resources and serve as edge-clouds that facilitate federated learning.

The central cloud entity establishes connections with individual factory floors through the Kafka bus, facilitating the exchange of data between the central cloud entity and factory floors. This connection also supports the exchange of AI/ML-related information, including trained models and training interactions.

The holistic AI trains a central AI/ML model using a federated learning approach; as detailed in the next subsection.

4.2.1. Federated AI/ML training at the central cloud entity

To carry out a federated learning procedure, the central cloud entity uses a procedure similar to the peer to peer one employed in floor to floor federated learning depicted in Figure 4.

First of all, the holistic AI agrees with the AI entities of each factory floor which is the AI/ML model that they will use within the federated learning stage. The central cloud entity does not require any data to perform the training. Its master node interacts with the workers of each factory floor to synchronize at the training stage. It exchanges with all of them the averaged weights received from the AI/ML models trained locally within each factory floor, and produces a federated AI/ML model that passes to the model deployment entity.

Finally, the BentoML at the central cloud entity exchanges the trained AI/ML model to the model deployment entities of each factory floor, so each requests the necessary tasks to the trained model, e.g., forecasts about imminent failures. This procedure is depicted in Figure 5 and the sequence diagram in Figure 6.

Note that this procedure presents two advantages with respect to centralized training. The first is that the data is locally pre-processed and only resides at each factory floor thus preserving privacy. Secondly, the data exchange with the holistic AI only carries weight updates, which will be presumably smaller than exchanging whole datasets.

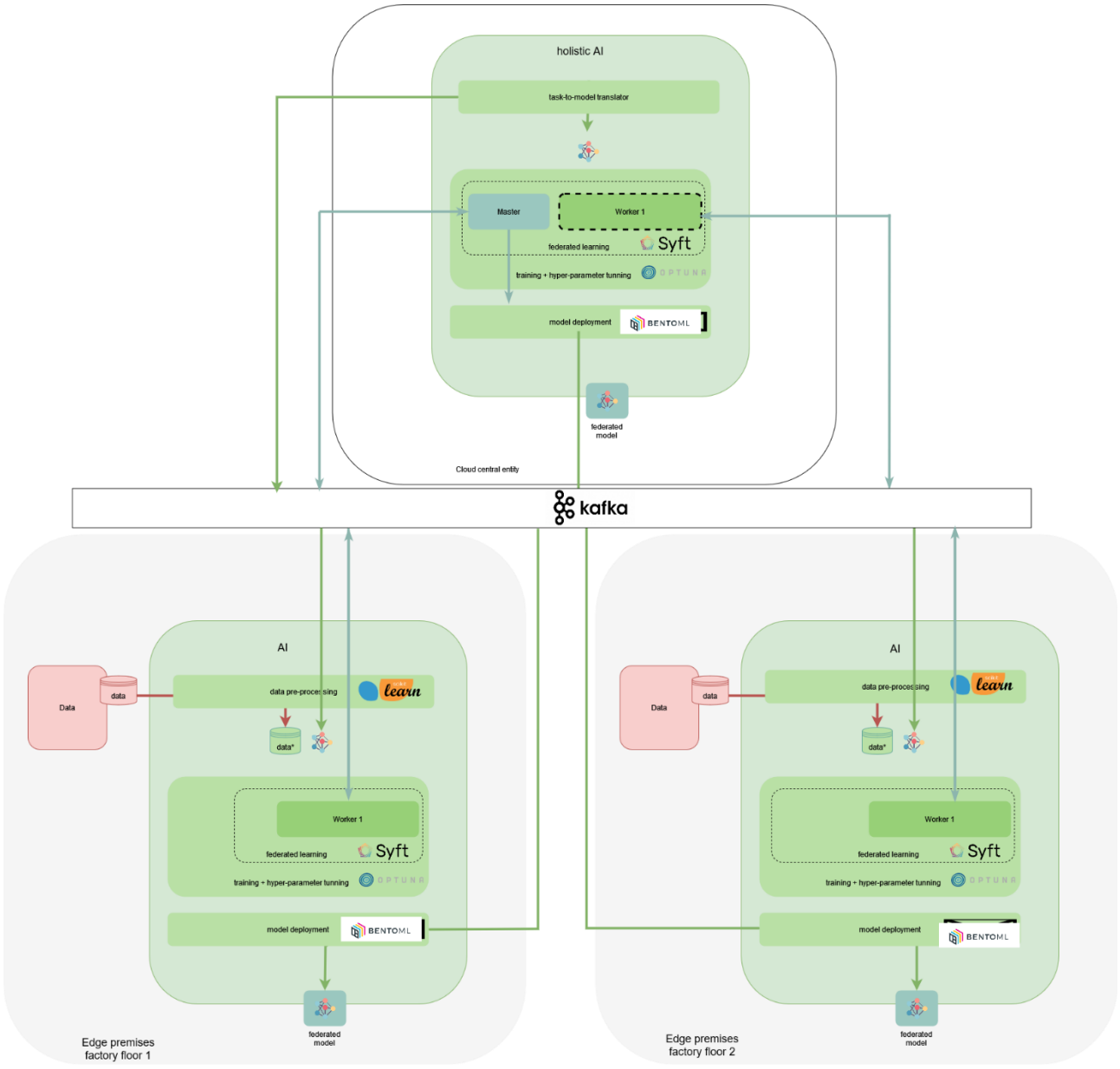

FIGURE 5: FEDERATED AI/ML TRAINING AT CENTRAL CLOUD ENTITY

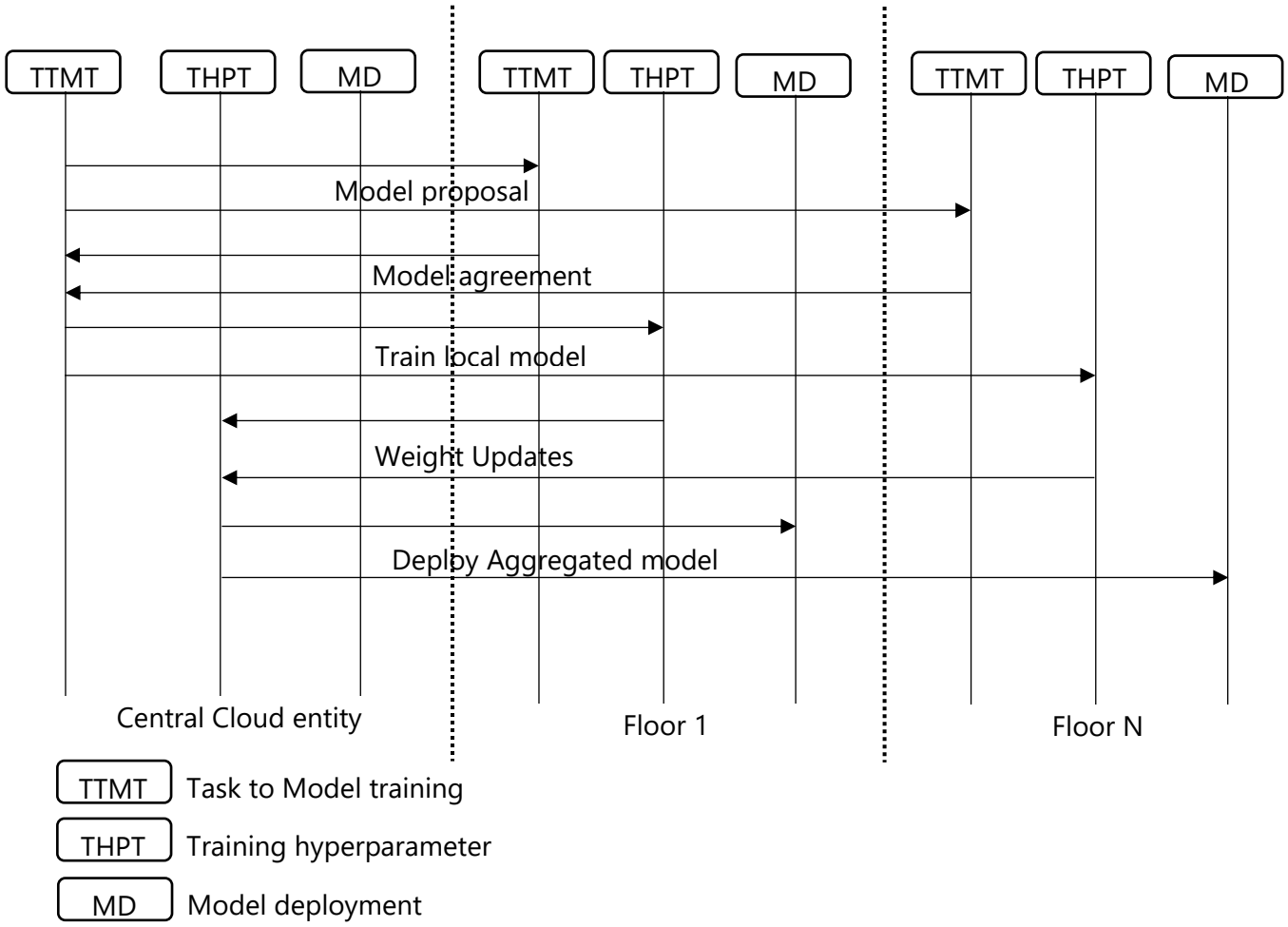

FIGURE 6: SEQUENCE DIAGRAM FOR FEDERATED AI/ML TRAINING AT CENTRAL CLOUD ENTITY

5. Data sources

Key to the functioning of the system is reliable data collection for training the AI models. The factory floor collects data of the IoT devices such as temperature, state of robotic arms, etc. All these metrics are reported to the Data entity within the factory floor for predictive maintenance tasks, error recovery, etc. Additionally, the Data entity of the factory floor keeps track of the devices available at its own premises: the network and IoT devices. To prevent non-authorized devices from trying to maliciously connect to the factory floor, there is an Authentication step within the factory floor.

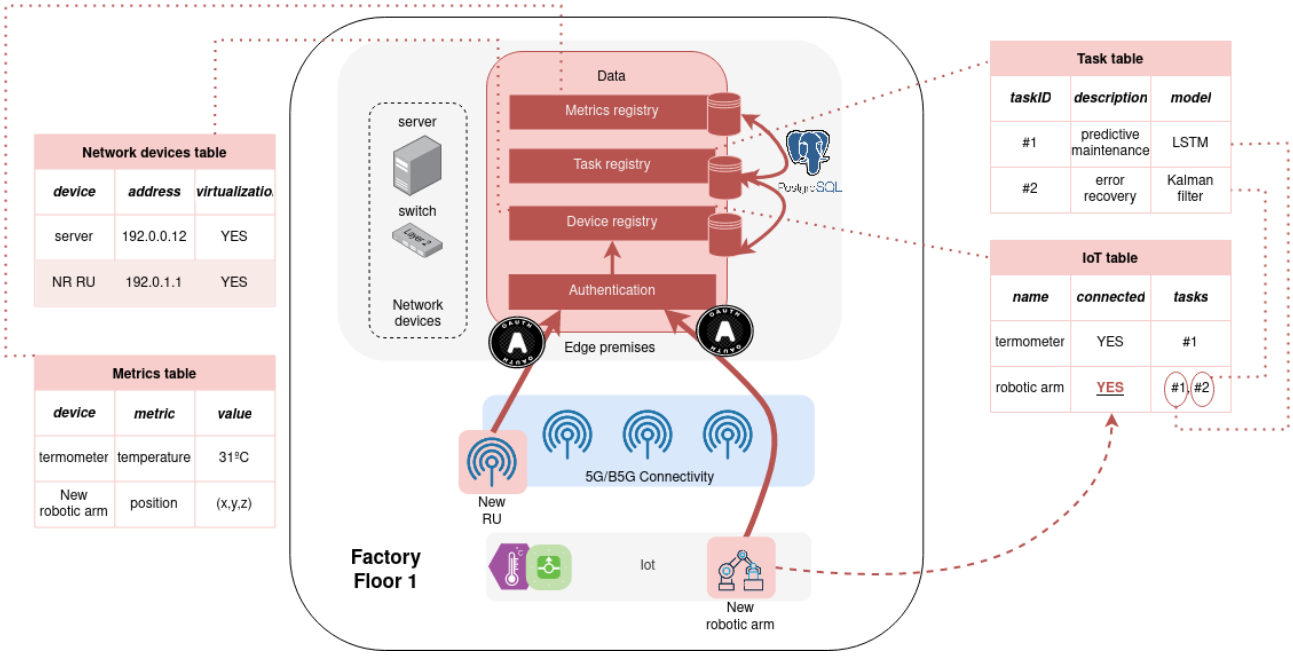

FIGURE 7: DATA SOURCES WITHIN THE FACTORY FLOOR

Figure 7 illustrates the data sources available within the factory floor, all of them captured by SQL tables stored in PostgreSQL at the factory floor premises. In the following we detail both the Authentication building block within the Data Entity, so as the registries holding the factory floor data sources:

- *Authentication*: this building block is the entry point for new devices attaching to the factory floor premises. Either if they are network or IoT devices (as the New RU and New robotic arm in Figure 7, respectively), they should do an OAuth challenge with the Authentication building block, so the latter verifies that they can operate at the factory floor.
- *Device registry*: this building block keeps track of the IoT and Network devices tables that store the factory floor devices. The network and IoT devices will only appear at such tables if the Authentication building block succeeded in the OAuth challenge. With the IoT table and Network devices tables it is possible to know which are the available devices to connect the IoT devices with the related AI/ML models as, e.g., the predictive maintenance. Furthermore, upon the attachment of new devices (as the New robotic arm in Figure 7) it is possible to report the orchestration AI/ML model (see Next section) the availability of a new IoT device

to consider it for a given task, e.g., the predictive maintenance. Regarding the network devices table, it contains records on the addresses of each device, and whether it has virtualization capabilities or not, so it is possible to deploy components on top of them in the orchestration stage.

- *Tasks registry*: this building block records the existing tasks within the factory floor, e.g.: the predictive maintenance, or error recovery shown in Figure 7. This information is kept within the tasks table, which also stores the AI/ML model used to perform the task. For example, the predictive maintenance model may be an LSTM neural network that was trained and deployed as specified in Section 3. It is worth mentioning that each task within the tasks table has a taskID that is associated to each IoT device in the IoT table. In such a manner, it is possible to have a one-to-many mapping of the task and associated devices, respectively. Specifically, Figure 7 shows that the New robotic arm is associated to the predictive maintenance, and its successful authentication leads to considering itself at the execution of the predictive maintenance task.
- *Metrics registry*: this building block keeps track of the metrics reported by the IoT devices at the Edge premises of the factory floor. It stores information related to the temperature, location of the robotic arms manipulators, etc. It is the content of the Metrics table of Figure 7 which keeps track of historic data used by AI/ML models to execute their tasks. For example, the temperature report may help a predictive maintenance AI/ML model to infer whether some IoT pieces may break soon.

It is worth mentioning that the Data sources at the factory floor may also exchange the data with the cloud central entity. In such a case, the global Data entity (see Figure 1) also has a metrics, tasks, and devices registry building block that collects data from the respective building blocks at the factory floor facilities. Consequently, the AI/ML models at the holistic AI can process global data to take better decisions of certain tasks.

Note that the data entity of each factory floor is isolated and only may share data with the central cloud entity. Therefore, the data is not exchanged among factory floors, which prevents data privacy leakages.

6. Automation

The primary objective of this document is to realize automation on factory floors through the proposed architecture. By automation, we mean achieving zero-touch deployment and AI-driven orchestration. Zero-touch deployment involves the seamless installation of services in a plug-and-play manner, eliminating the need for manual service configuration. For instance, a new robotic arm can automatically perform required tasks upon authentication without requiring manual setup. On the other hand, AI-based orchestration involves the intelligent management and coordination of network and computational devices (such as servers or IoT devices with storage or processing capabilities) to host services within the factory premises. As an example, an AI/ML model may determine that hosting the predictive maintenance model on a server within the Edge premises of a factory floor is optimal and can allocate the necessary bandwidth resources to connect the robotic arm to the server.

With the help of Figure 8, in the following sections we explain the components involved in the AI-based orchestration and zero-touch deployment.

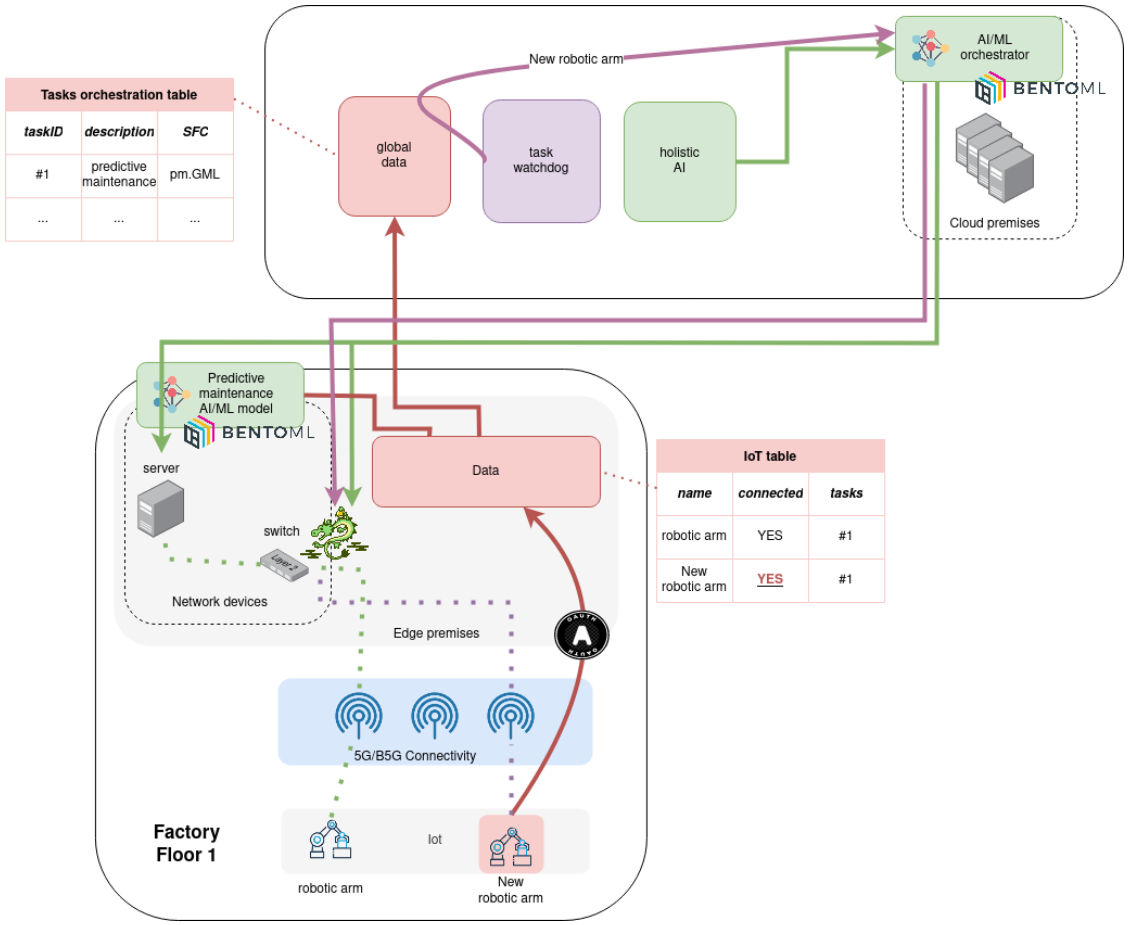

FIGURE 8: AUTOMATION INTERACTIONS

6.1. AI-based orchestration

For the initial system architecture, we consider that the AI-based orchestration resides in the central cloud entity, i.e., the orchestration of industrial services runs in the cloud and has a global view of the factory floors within the pool.

The global Data entity keeps track of a task orchestration table that specifies the Service Function Chain (SFC) of the deployed industrial service. The SFC is a graph with each node representing a Virtual Network Function (VNF) – as the predictive maintenance AI/ML model –, that interconnects with other VNFs or Physical Network Functions (PNFs) as a robotic arm. The SFC graph also holds the bandwidth and computing requirements of the service to deploy. For instance, the predictive maintenance SFC could be a tree graph of depth one with each leaf being the robotic arm, and the root being the AI/ML predictive maintenance model.

Every orchestrator task is delegated to the comprehensive AI for determining the deployment of the corresponding Service Function Chain (SFC). For instance, the input related to predictive maintenance is processed by a pre-trained AI/ML orchestrator embedded within the comprehensive AI. This AI/ML orchestrator undergoes training following the methodology outlined in Section 3. However, instead of utilizing data from metrics tables, it utilizes data sourced from the network devices table to ensure awareness of the available network resources. An approach worth considering involves defining a reinforcement learning AI/ML model that receives all tasks during the training phase and learns how to effectively orchestrate them.

Once the AI/ML orchestrator is trained and deployed at the cloud as a BentoML API, each task orchestration entry is fed to the model to deploy the related task SFC, e.g., the predictive maintenance. Figure 8 illustrates in green arrows the interactions between the BentoML, the Edge server, and the RYU SDN controller to deploy the Predictive maintenance AI/ML model (see the server within the factory floor), and allocate the network resources to interconnect it with the robotic arm (see the dotted green lines).

6.2. Zero-touch deployment

Upon completion of the training process for the AI/ML orchestrator (refer to Figure 8), the BentoML model takes in all orchestration task entries, facilitating their deployment. For instance, in the example depicted in Figure 8, the predictive maintenance task is autonomously deployed. The entire orchestration procedure is executed without human intervention, given that the orchestration tasks table encompasses all the Service Function Chains (SFCs) slated for deployment.

As a result, the deployment of AI/ML tasks is autonomously managed by the system architecture. However, the handling of scenarios involving new device authentication in the factory floor remains undefined. Figure 8 illustrates the procedure for such scenarios. To address this, we introduce the concept of the task watchdog—an entity that periodically monitors authenticated devices recorded

in the global data entity, prompting orchestration updates for each task. The subsequent explanation provides a detailed breakdown of this process.

1. Initially, a new robotic arm initiates an OAuth authentication to register on the factory floor. The data associated with the new device is exchanged with the global data entity of the central cloud entity, updating its IoT table.
2. Subsequently, the task watchdog identifies the authentication of the new robotic arm and determines the associated tasks. In the context of Figure 8, the new robotic arm is associated with the predictive maintenance task.
3. The task watchdog notifies the AI/ML orchestrator about the inclusion of the new robotic arm in the predictive maintenance task. Specifically, the AI/ML orchestrator, upon receiving the predictive maintenance Service Function Chain (SFC), recognizes the need to allocate traffic resources for connecting the new robotic arm with the AI/ML predictive maintenance running at the factory floor's Edge premises. Consequently, the AI/ML orchestrator issues instructions to the RYU controller at the factory floor, directing the creation of new routes and the allocation of bandwidth to interconnect the AI/ML predictive maintenance, the 5G/B5G RUs, and the new robotic arm (as indicated by the purple dotted lines in Figure 8).

The aforementioned interactions trigger orchestration updates (with the help of the task watchdog) on the authentication of new devices, hence, leading to a zero-touch deployment.

7. Summary and Conclusions

This revised deliverable outlines the system architecture designed for evaluating AI/ML automation in the connected industry. The architecture encompasses Edge premises within each factory floor, a central Cloud premise for aggregating and processing information from multiple factory floors in a pool. Each factory floor includes its Data and AI entities responsible for collecting and processing device metrics in the production pipeline. The proposed architecture prioritizes data privacy, collaborative AI/ML model training, and facilitates automated registration and integration of new devices into production tasks. Furthermore, it supports maintenance operations, simplifying the monitoring and surveillance of device statuses within the factory floor.

8. References

Applications, O. N. (2022). *Khasa Gillani, Jorge Martín-Pérez, Milan Groshev, Antonio de la Oliva, Carlos J. Bernardos, Robert Gazda.* arXiv.

Milan Groshev, J. M.-P. (2022). FoReCo: a forecast-based recovery mechanism for real-time remote control of robotic manipulators. *IEEE Transactions on Network and Service Management*, 12.

Milan Groshev, J. S.-P. (2022). Demo: FoReCo – a forecast-based recovery mechanism for real-time remote control of robotic manipulators. *SIGCOMM '22: Proceedings of the SIGCOMM '22 Poster and Demo Sessions* (pág. 2). Amsterdam: ACM.