IIP-Ecosphere Whitepaper

IIP-Ecosphere Platform Requirements (Usage View)

(Version 1.0)

Heiko Stichweh, Christian Sauer, Holger Eichelberger

White Paper IIP-2021/001

IIP-Ecosphere

Next Level Ecosphere for Intelligent Industrial Production

Disclaimer

The contents of this document has been prepared with great diligence. Although the information has been prepared with the greatest possible care, there is no claim to factual correctness, completeness and/or timeliness of data; in particular, this publication cannot take into account the specific circumstances of individual cases.

Any use is therefore the reader's own responsibility. Any liability is excluded.

This document reflects only the views of the authors at the time of publication only. The Federal Ministry for Economic Affairs and Energy or the responsible project agency are not liable for the use of the information contained herein.

Publication: March, 2021

DOI: 10.5281/zenodo.4485801

Gefördert durch:

Bundesministerium
für Wirtschaft
und Energie

Executive Summary

This Whitepaper describes a shared view on the IIP-Ecosphere platform, which was developed as a core technical contribution of the IIP-Ecosphere Think Thank “Platforms”, to foster and complement the requirements collection of the platform, based on this shared view on envisioned platform functionality. Following the Industrial Internet Reference Architecture [IIRA], this Whitepaper describes the IIP-Ecosphere platform from the Usage Viewpoint. The Usage View on the IIP-Ecosphere platform that we discuss in this document represents the common view of all partners involved in the design, the subsequent implementation and, finally, the operations of the platform based on the voice of the prospective users of the platform in the IIP-Ecosphere community. The Usage View plays the role of a set of high-level usage-based requirements to drive the more detailed requirements collection to define the Functional View on the platform.

The Usage View presented in this Whitepaper was collected in terms of a series of workshops with all interested project partners. Based on discussions of partners with interests in the platform and the platform architecture, i.e., the Think Tank “Platform” and the work package “AI Accelerator”, a joint vision of the IIP-Ecosphere is created. Based on the precursor work on the related edge configuration and management by the Lab Networks Industry 4.0 testbed “edge configuration” [LNI-UV], the authors used the joint vision as basis for creating a proposal of the system entities, the roles and the activities documenting the interaction to be supported by the IIP-Ecosphere platform. This proposal was discussed, as stated, with interested project partners in terms of a series of workshops, namely a general requirements workshop and several focus workshops. All workshops were online workshops, due to the Corona situation at the time of the workshops, using online collaboration techniques, such as interactive whiteboards, with two to three hours of interaction with the stakeholders and intermediate feedback phases between the workshops. All workshops first introduced the approach and provided space for the participants to discuss, improve and develop the system entities, the roles and, in particular, the activities. The shared Usage View is also aimed at supporting further discussions between the different Think Tanks of the IIP-Ecosphere platform, which are the Think Tanks: “Platforms”, “Data”, AI and Production” and “Business Models”. The shared Usage View established in this document therefore provides a basis for deriving/validating the functional and quality requirements of the overall platform and, thus, enables the subsequent work on the development of the concepts and solutions established in the shared Usage View presented in this Whitepaper.

Due to time and resource limitations, it was important to prioritize the topics to be covered. For this current version of the Usage View, we jointly decided to focus on application building, distribution and AI services, as these topics strongly correlate with the technical foundations of the platform to be developed. For this focus, we describe a System under Consideration with 18 entities, 19 roles and 67 activities in this Whitepaper. As also important topics like interconnections between platform instances, security and data protection are currently not in this focus, we plan to initiate a second round of workshops and to update this Usage View accordingly.

1 Introduction

1.1 Motivation and goals

The goal of this document is to capture and establish a shared view on the IIP-Ecosphere platform, which will be developed as a core technical contribution of the IIP-Ecosphere Think Tank "Platforms", and to foster and complement the requirements collection of the platform, based on this shared view on envisioned platform functionality. Following the Industrial Internet Reference Architecture [IIRA], this document describes the IIP-Ecosphere platform from the Usage Viewpoint, see Figure 1.

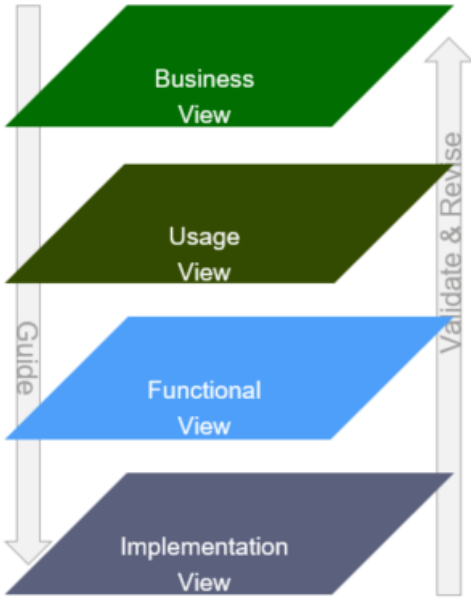


Figure 1: Industrial Internet Architecture Viewpoints [IIRA]

The Usage View on the IIP-Ecosphere platform that we discuss in this document represents the common view of all partners involved in the design, the subsequent implementation and, finally, the operations of the platform based on the voice of the prospective users of the platform in the IIP-Ecosphere community. The Usage View plays the role of a set of high-level usage-based requirements (given in terms of usage activities), which, in turn, drive the more detailed requirements collection[1] to define the Functional View on the platform, which consists of the functional and quality requirements as well as the architecture of the IIP-Ecosphere platform. Subsequently the solution approaches, which form the Implementation View, will also be based (indirectly) on the collective requirements of all partners.

The other direction, a description of the IIP-Ecosphere platform from the Business Viewpoint is pragmatically foreseen for future work in collaboration with the Think Tank "Business Models". Therefore, the Usage View presented in this document will act as foundation and input from a more technical side, but also to develop future business models for an AI-based Industrial Internet of Things (IIoT) platform, i.e. to support one of the core contributions of the Think Tank "Business Models".

This Usage View was collected in terms of a series of workshops with all interested project partners. Based on discussions of partners with interest in the platform and the platform architecture, i.e., the Think Tank "Platform" and the work package "AI Accelerator", a joint vision of the IIP-Ecosphere based

[1] In fact, the functional requirements collection was done before starting the Usage View, but both will be synchronized to form a coherent version of architecture (functional viewpoint) and implementation (viewpoint).

on the overall project concept and the grant agreements was created. Based on the precursor work on the related edge configuration and management by the LNI4.0 testbed "edge configuration" [LNI-UV], the authors used the joint vision as basis for creating a proposal of the system entities, the roles and the activities documenting the interaction to be supported by the IIP-Ecosphere platform. This proposal was discussed with interested project partners in terms of a series of workshops, namely a general requirements workshop and several focus workshops. All workshops first introduced the approach and provided space for the participants to discuss, improve and develop the system entities, the roles and, in particular, the activities. Due to time and resource limitations, it was important to prioritize the topics to be covered. For this version of the Usage View, we jointly decided to focus on application building, distribution and AI services, as these topics strongly correlate with the technical foundations of the platform to be developed. As also important topics like interconnections between platform instances, security and data protection are currently not in this focus, we plan to initiate a second round of workshops and to update this Usage View accordingly.

The shared Usage View established in this document therefore will provide a basis for deriving/validating the functional and quality requirements of the overall platform and, thus, enable the subsequent work on the development of the concepts and solutions established in the shared Usage View. A further goal of establishing a shared Usage View of the platform is to define the scope of the subsequent work (what can and what cannot/shall not be realized within the lifetime of IIP-Ecosphere) based on a shared understanding of the concepts within the platform for all partners. In other words, the Usage View may also contain also usage scenarios of the platform that will be scoped out for now. Although these scenarios are then not subject to the realization work, it is important to collect and describe them, as they can form the basis for future work of the IIP-Ecosphere community after the project's lifetime or for other IIoT platform works.

1.2 Interaction with other initiatives

This version of the Usage View was collected from the partners in the IIP-Ecosphere project. In addition, but also for future versions, e.g., including topics that were not in focus for this version, interaction with other initiatives are important:

- The LNI Usage View on edge configuration in terms of the approach taken to obtain the results but also the specific results presented in [LNI-UV] provided a relevant basis for this work. In the counter direction, initial discussions with the LNI (ensuring that re-using the approach and the results in our context is possible) showed interest in the extension of the original work that we do in IIP-Ecosphere. Thus, we believe that compiling this document is a basis for passing back our view into the work of the LNI.
- The German Standardization Roadmap Industry 4.0 [ZVEI I4] and the Industrial Internet Reference Architecture [IIRA] provided a basis for the LNI Usage View, and so we considered the work done there also as a basis for our work on the Usage View of the IIP-Ecosphere platform.
- Discussions between IIP-Ecosphere and other research-projects in this field like BMWi DaPro[2] or BMWi BaSys 4.0/4.2[3] provide a broader insight into the field and other works on deploying (edge) services, describing standardized interfaces or even enabling the use of AI services.
- First realizations of edge management systems in the sense of [LNI] have been developed. One example is the IBM Edge Application Manager[4], which relies on (Docker) containers as

[2] http://dapro-projekt.de/
[3] https://www.basys40.de/, https://www.iese.fraunhofer.de/en/innovation_trends/industrie40/basys42.html
[4] https://www.ibm.com/support/knowledgecenter/SSFKVV_4.1/getting_started/overview_ieam.html

well as their management through Kubernetes or Open Shift. However, the focus of [1] is on the edge and container management rather than on the services running on the edges, and so important topics like definition and management of data flows, data endpoints, their connections to sensors or to the user interface (UI, HMI) are typically out of scope. In contrast, we are convinced that applications in such systems are inherently distributed and consist of services that also must be managed and so we extend in this whitepaper the [LNI] edge management approach into such directions.

Furthermore, this version shall provide the basis for further discussion with stakeholders interested in the IIP-Ecosphere community. Also from this point of view, it is important to not consider this work as finished requirements, rather than an initial set to be extended by a second round of workshops, but also refined by discussions with the stakeholder community.

1.3 Structure of the document

As stated, the Usage View on the IIP-Ecosphere platform given in this document was established according to the recommendations of the German Standardization Roadmap Industry 4.0 [ZVEI I4]. Following their recommendation to base Usage Views of systems on the Industrial Internet Reference Architecture [IIRA], the Usage View given in this document comprises the following elements:

- The definition of the “System under Consideration”, being the IIP-Ecosphere platform.
- The definitions of all “Roles” involved in the System under Consideration, where “Roles” are actors outside of the System under Consideration who interact with the System under Consideration in varying form.
- The Definition of all “Activities” that are executed when “Roles” interact with the System under Consideration. The core element of an activity description is formed by a workflow of individual tasks that are executed by roles.

The present document, having described motivation and goals of the document and interactions with other initiatives, is further structured as follows: In Section 2, a brief overview on the IIP-Ecosphere platform is given in terms of the envisioned high-level services of the platform. The purpose of this overview is to provide a background as well as a motivation for the discussion of requirements derived from the Usage View. Section 3 then gives an overview of the System under Consideration and, following the approach described above, defines the concepts identified in the shared view. Section 4 then defines the Roles that interact with the System under Consideration, i.e., the IIP-Ecosphere platform. In Section 5, the subsequent Activities are defined, describing how the Roles, defined in Section 4, can interact with the IIP-Ecosphere platform. Section 6 discusses how the presented material fits into the foreseen scope of the work in IIP-Ecosphere, i.e., performs an initial scoping with respect to the project grant contracts and the available resources. The document then concludes with a summary in Section 7.

2 Overview the IIP-Ecosphere platform

In this section, we give a brief overview of the IIP-Ecosphere platform in terms of the envisioned high-level services. This overview shall provide a background as well as a motivation for the following discussion of Usage View requirements. In particular, we focus here on the big picture rather on technical solution candidates that may be used to realize the platform. Technical aspects, as well as detailing functional and quality requirements will be topics of a subsequent document.

Figure 2 visualizes (a kind of mind map) the services to be provided by the IIP-Ecosphere platform. This vision was developed by the joint working group "Architecture" in IIP-Ecosphere. This working group mainly consists of the partners of the work package Think Tank "Platforms" (University of Hildesheim/Software Systems Engineering, University of Koblenz-Landau) and the work package “AI Accelerator” (represented by Siemens, KI-Protect, Lenze, Bitmotec), but also encompasses interested or associated partners such as Phoenix Contact.

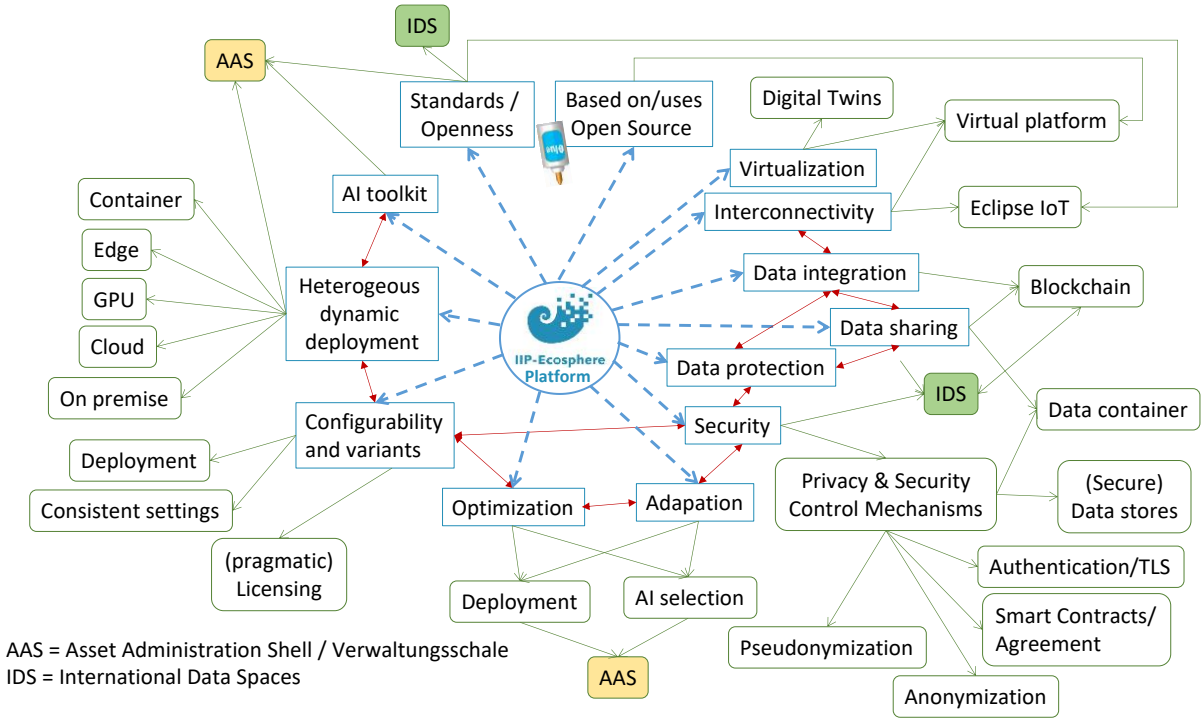


Figure 2: MindMap overview of the envisioned platform services.

The IIP-Ecosphere platform consists of (discussed counter clockwise following the inner cycle in Figure 1):

- **AI toolkit**: An open and extensible AI toolkit containing reusable AI building blocks in the form of services. The platform will provide interfaces and mechanisms to distribute AI services to execution resources and supply them with data and to make the output available to subsequent components - for data exploration, training and for runtime prediction. The specific blocks, methods, services, entities etc. are described via digital representations, e.g. in the form of the Asset Administration Shells (AAS) [AAS-D], which offer a standardized manufacturer-independent / interoperable description with regard to their properties, capabilities and interfaces. The interfaces shall be defined jointly with the relevant work packages of the project, in particular the demonstrators, the Think Tank “AI and production” and the AI Accelerator.

- **Heterogeneous dynamic deployment**: A mechanism for heterogeneous, dynamic deployment of selected platform functionalities such as AI building blocks to different resources. The platform performs the packages and distribution of the services or components based on information such as service orchestration, assurances and resource requirements. This information is provided by the services, which, in turn, describe themselves in terms of AAS. Analogously, the potential target resources, such as edge devices or GPU servers, describe themselves in terms of their AAS. As deployment units, containers (e.g. Docker container[5]) are used. Primarily the IIP-Ecosphere platform aims at on-site/on-premise installations. A cloud-based deployment of the containers is considered optional.

- **Configurability and variants**: For systematic and consistent configurability, the IIP-Ecosphere platform will use variant management techniques [LSM]. This allows determining even before the execution of the platform whether the respective platform configuration/installation is consistent and executable and, thus, simplifies installation and maintenance tasks. Since the configuration also describes the (distributed) applications and their required components, i.e., the orchestration, the configuration also enables the model-based application-driven construction of the deployment units and the binding of components against the technical infrastructure.

- **AAS**: The static configuration described above is supplemented by (dynamic) information from the AAS of the used components. This allows reasoning on and optimization of the heterogeneous deployment based on available resources. As one aspect, the optimization contributes an initial allocation of services/components to platform resources when starting an application.

- **Adaptation**: Moreover, the optimization of the deployment can also be done at runtime and, thus, induce one aspect of self-adaptation. As a complementing mechanism, we envision the runtime adaptation of deployed services, in particular AI methods, to re-configure the methods or to substitute them by alternatives that are better suited in the current situation. One example for such a situation is the replacement of a production machine, which supplies more precise data. Subsequently, automated replacement of an AI method (implementation) working on that data can lead to better predictions.

- **Security**: Various security mechanisms, including authentication and transport layer security (TLS). In addition, the platform will offer various storage services (data lakes) with different quality characteristics (including secure or encrypted storage) and will make them available to all components, especially services deployed to edge devices. Furthermore, (external) data connections shall optionally be secured by means defined in the references architecture [IDS-RAM] of the International Data Spaces (IDS) Association.

- **Data protection**: Mechanisms for data protection. Article 25 of the DPA (Data Protection by Technology Design and by Privacy Friendly Presettings, PbD (Privacy by Design) requires that appropriate technical and organizational measures must be integrated into a system both at the time the means for processing are determined and at the time of the current processing. Example measures are e.g. Security & Privacy Controls [NIST-800-53] for effectively implementing data protection principles such as data minimization.

[5] https://www.docker.com/

- **Data sharing**: IIP-Ecosphere wants to foster data sharing among partners in the ecosystem. Thus, the platform shall provide means to support data sharing, in particular in interrelation with data protection and security (e.g., through smart contracts or data usage agreements).

- **Data integration**: Services for runtime data integration, i.e., combining and augmenting machine and sensor data with other production-relevant data like order information, floor plans, factory construction information, etc. The integrated data may be fed into AI methods to improve the predictions.

- **Interconnectivity**: Protocols for interconnectivity on different levels so that data from underlying platforms or machines can be made available to all services/components running on the platform. This includes protocols from Industry 4.0 enabling the connection to machines and already installed platforms. Examples are the use of the OPC-UA [OPC-UA] communication framework via the MQTT protocol on the "southern" side of the platform, communication of different various instances of the IIP-Ecosphere platform at different sites (horizontal interconnectivity) as well as connections to control systems (SCADA, MES) or visualizations ("northern" interconnectivity).

- **Virtualization**: Virtualization of the platform and its components. Fundamentally, the IIP-Ecosphere platform is designed as a virtual platform, i.e., it is not intended to replace existing Industry 4.0 platforms [SEA+] rather than basing on and complementing existing services, protocols and installations. Furthermore, the IIP-Ecosphere platform itself shall be virtualized, ranging from the containers of the dynamic and heterogeneous deployment up to the central components of the platform. Where possible, the IIP-Ecosphere platform shall also provide means to integrate digital twins [VDW-DZ].

3 System under Consideration

In this section, we present the System under Consideration in terms of the parts / assets and entities that are forming the system and the relations between parts / assets and entities within the system. Therefore, we present the System under Consideration as follows:

- A rather high-level technical view of relevant parts / assets and roles that will form the IIP-Ecosphere platform. This view on the System under Consideration may be understood as a kind of **proto-architecture** with relevant, logical entities (can be understood as high-level components or services) and their interaction with the actors of the system (roles). However, it is important that we do not aim at defining an architecture of the platform here, as the focus is on the interaction of actors with the system rather than architectural styles or interfaces. In other words, we avoid taking architectural decisions here, which will be made based on the combination of the Usage View and functional/quality requirements on the level of the Functional View. In the opposite direction, the proto-architecture here shall be used as a collection of elements to be detailed by the later platform architecture.
- A description of the main classes of **entities** in the System under Consideration such as objects or components like edge device, edge runtime, the application(s) running on the platform, etc.
- The **roles** classifying and characterizing the system-relevant aspects of the stakeholders/actors interacting with the system. As usual, a stakeholder can play one or multiple roles at the same time and, further, a stakeholder can be a human, an organisation or an external system.

Implicitly, through the selection of entities, this section already paves the way for the system scope in Section 6, i.e., questions like “What’s in? What’s out? Who interacts with the system?” In Section 4, we detail the roles and in Section 5 the activities that roles may perform on the entities.

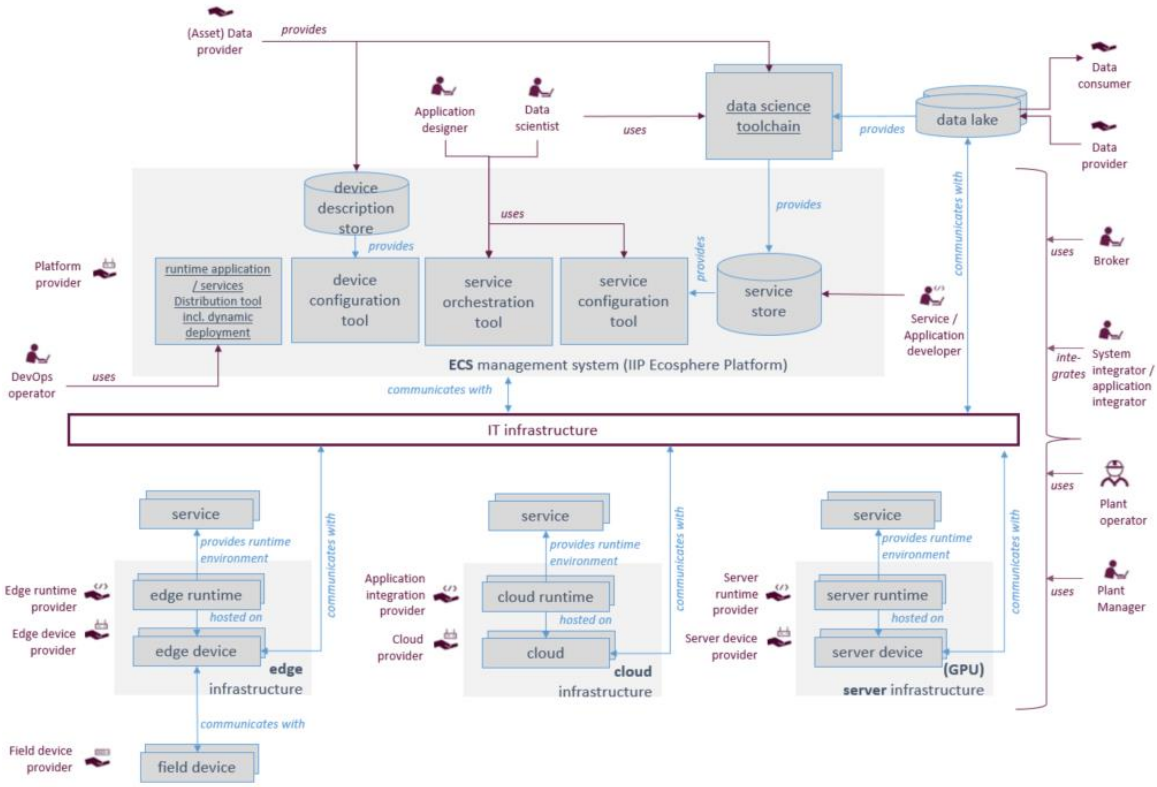


Figure 3: Proto-architecture of the System under Consideration

Figure 3 illustrates the "System under Consideration" in terms of a proto-architecture drawing in the style of the LNI edge configuration Usage View [LNI-UV]. An enlarged version to Figure 3 is also available as Appendix 1. The considered entities are coloured in grey, the considered roles are coloured in purple, and the interactions between the System under Consideration and roles are drawn as purple arrows. Areas indicating systems of the IIP-Ecosphere platform, such as the ECS management system, where "ECS" stands for Edge/Cloud/Server, or the Edge Cloud Server devices/infrastructures, are shaded in light grey, covering all entities that belong to the indicated systems. Data flows between Entities are drawn as light blue arrows, which can be directed flows or bi-directional flows, indicated by the arrowheads. The underlying IT-infrastructure of the System under Consideration, connecting the Edge Cloud Server devices/infrastructure with the IIP-Ecosphere platform is depicted as a white box with a purple outline, as it is neither an entity nor role in the System under Consideration.

3.1 Field device

Field devices are physical computing resources with real-time communication capabilities. Field devices communicate with edge devices, can be configured via parameters and the firmware of a field device can be updated. [LNI-UV]

Note that we apply a broad understanding of the term "field device". We call each device that does not support the activities as described in this paper a field device. Thus, even a programmable logic controller, which for example does not support the deployment of applications, is called a "field device" in this context. [LNI-UV]

3.2 Edge device

Edge devices are physical computing resources with capabilities for communication and several edge runtimes to be deployed on the edge device. An edge device formulates its assertions with respect to edge runtimes deployed on the edge device. Edge devices also can be configured by parameters and

the firmware of edge devices can be updated. Edge devices can be connected to field devices and for each connected field device there is a data connector, provided by the service operating on the edge device, representing the communication capabilities between field and edge device, please see Section 3.8 "Service" for further information on the handling of data connections within a service. This data connector can be configured by an ECS management system. [LNI-UV]

3.3 Edge runtime

Edge runtimes are system software applications, which are hosted on edge devices and provide a runtime environment allowing applications to be deployed and executed. An edge runtime formulates its assertions with respect to applications deployed on the edge runtime and their requirements with respect to an edge device. [LNI-UV] Edge runtimes may be optimized with respect to the resources and services that (a kind of) edge devices can offer.

The set of all edge runtimes hosted on edge devices is called the edge infrastructure. [LNI-UV]

3.4 Server device

Server devices are physical computing resources with capabilities for communication and one or more server runtimes to be deployed on the server device. We consider server devices as dedicated machines that are hosted on premise or externally by a different organization (a "hoster") and are at least partially administered by the organization operating the edge and field devices. Server devices also can be configured by parameters and the server runtime can be updated.

3.5 Server runtime

Server runtimes are system software applications hosted on server devices to provide a runtime environment allowing applications or services as part of a distributed application to be deployed and executed. A server runtime formulates its assertions with respect to applications deployed on the server runtime and their requirements with respect to a server device, e.g., whether specialized resources such as GPUs (graphics processing units) are offered and how they can be shared. Technically, a server runtime shall offer the same functionality as an edge runtime (same interface), however, the binding to the underlying technology may differ or it may be optimized for the specific (kind of) server device, e.g., a different container technology than on an edge device may be offered. The set of all server runtimes hosted on server devices is called the server infrastructure.

3.6 Cloud

A public cloud is a set of resources (CPU, GPU, TPU, etc.)[6] managed by a cloud provider under certain level of Quality of Service (QoS) and billing agreements. Various types of clouds do exist, e.g., public clouds running on the premise of a cloud provider, private clouds where on premise resources are used to host and administer a cloud infrastructure, or hybrid clouds mixing aspects of public and private clouds. Aspects like QoS, accessibility, transfer latency and billing differs among these types. At its core, a cloud can be seen as a set of server devices running additional software layers, in particular to enable elasticity, resilience or virtualization. However, these aspects like transfer latency, QoS and billing models may imply significant differences in practically executing services or applications. As typically, for a user cloud facilities occur as a single instance, for the remainder of this Whitepaper we will refer to any cloud facilities as a cloud device.

[6] Example resources for AI processing such as CPU = central processing unit, GPU = graphics processing unit, TPU = tensor processing units, etc.

3.7 Cloud runtime

Cloud runtimes are system software applications hosted on cloud devices to provide a runtime environment allowing applications to be deployed and executed. A cloud runtime formulates its assertions with respect to applications deployed on the cloud runtime and their requirements with respect to a cloud device, e.g., whether specialized resources such as TPUs are offered, how they can be shared or how the interaction between virtual machines and containers of different tenants can be taken into account. Technically, a cloud runtime shall offer the same functionality as an edge or a server runtime (same interface), however, the binding to the underlying technology may differ or it may be optimized for the specific (kind of) cloud device, cloud provider, the virtualization technology and constraints, or the billing model of a certain cloud provider. The set of all cloud runtimes hosted on cloud devices is called the cloud infrastructure.

3.8 Service

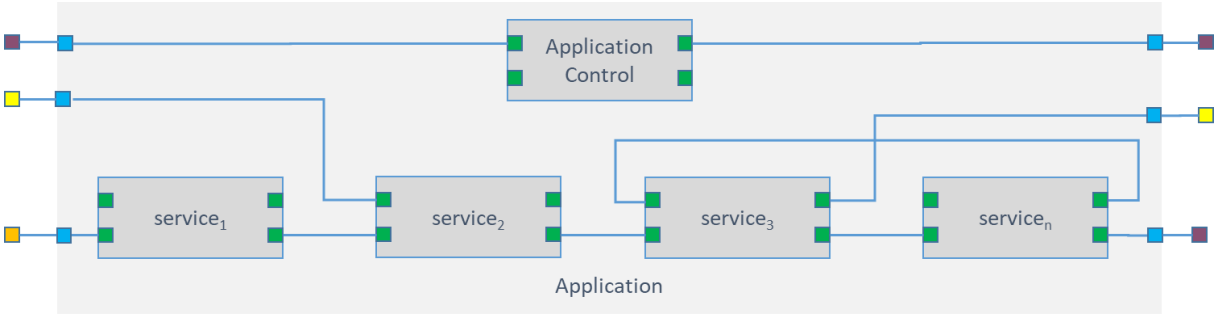
A service is a singular distributable functional unit that offers a specific functionality[7]. Each service can be configured and deployed on an ECS system, where "ECS" stands for Edge/Cloud/Server, via technological standards (e.g. container, web service, etc.) in an executable or callable way. Applications (see also Section 3.9) are always composed of one or more interconnected services and therefore services form the basis for applications. Hence, any service in production within the IIP-Ecosphere platform is also always an application. The capabilities, characteristics, properties and interfaces of a service are described by an information model / digital representation. In IIP-Ecosphere, we plan to use a standardized and device-independent information model for this purpose, more specifically the Asset Administration Shell (AAS) [AAS-D].

A service as a part of an application is always executable within an edge, server or cloud runtime. The execution of a service can be container-based, as a deployable stand-alone application with defined API, as a web service or a remote procedure call, e.g., an OPC-UA method. Services are versioned and can be updated to newer versions or reset to earlier versions of themselves. Services can communicate with the IT-infrastructure, for example other services or infrastructure services, and with field devices via their associated edge or the edge runtime, respectively. Services shall be compliant with common industry standards such as [SOA, TOSCA] with regard to their interfaces and information models (representation) to be able to be orchestrated into applications and to be deployed as distributed applications.

Services can have required and optional consumed/provided input/output data. A data connector is a logical entity with the purpose to exchange data. Data connectors are either defined in a service / an application or they are induced by a field device / data lake as sources or sinks, see Figure 4. More specifically, each service provides data connectors, i.e., input/output interfaces for data, in order to specify and safely perform data exchange. ECS devices, where "ECS" stands for Edge/Cloud/Server, and IT-infrastructure do not have explicitly defined data endpoints/connectors, but they have the capability to automatically generate the necessary communication connections depending on the (distributed) deployment of the application. The data connector therefore also stores and provides all

[7] We follow the usual terminology of Service Oriented Architectures, in particular those from the OASIS SOA reference architecture and [SOA] for deployment [TOSCA] naming connections between services "relations" or templates of configured services and relations. However, no terms for the "endpoint" are defined there. We will use the term "data connector" for a logical data input/ouput capability with associated properties, while we use the term "endpoint" for the underlying physical/network connection to the data. A data connector may consist of multiple endpoints.

necessary information on endpoint connections of a service / an application and their respective configurations.



- *data connector (interface) induced from a field device (e.g. sensor)*
- *data connector (interface) induced from a service*
- *data connector (interface) induced from an application*
- *data connector (interface) induced from a sink or source (e.g. HMI, business process, MES)*
- *data connector (interface) induced from a data lake (e.g. ERP-system)*

Figure 4: Data connectors of an application

Within the platform, monitoring tasks can be implemented as services and be provided for platform-wide use, analogue to the provision and use of (AI) services (akin injected probes may be possible). Using monitoring services can hence provide any monitoring task, such as heartbeat functions of assets, storage capacity monitoring or the monitoring of the functional state of applications, within the platform. The monitoring tasks may communicate with central tasks, e.g., to notify about processing problems. Besides such a central collector service, the platform may offer (central) platform services, such as data lakes, authentication, role-based access, data synchronisation, etc.. Platform services are provided by the platform and hence do not form specific applications, as an application within the IIP-Ecosphere platform is an executable deployed to the production environment of a user of the IIP-Ecosphere platform.

An application consists of at least one service or can be formed by orchestrating several (AI, monitoring, platform) services into an application. The application will always provide all necessary information on the needs of its service(s), in the form of relations of services of an application between themselves and to external entities and capacities, such as computational- storage- or communication-capacities, the services interact with and/or rely on.

3.9 Application

An application is an executable software solution consisting of a number of configured and orchestrated (combined) services. The services of applications can be deployed and executed in a distributed way on edge devices, cloud devices, server devices and any other computational resource accessible to the ECS management system. The (description of the) software programs manifesting the services are provided via the service store.

Services forming an application can be generic, customizable or application/customer specific. Each service of an application is executable and deployable, for example in a container[8], as a stand-alone

[8] We name here the potential cases while IIP-Ecosphere will probably focus on container-based deployment.

application with a specific API, as a web service or as an OPC-UA Method [OPC-UA]. Applications have data connectors, which are defined by the provider of an application and may be bound to data sources or external interfaces. Applications formulate their requirements with respect to a runtime or IT-infrastructure, for example, requirements with respect to real-time operation of the application or required resources. Due to the distributed nature, typically only parts of an application, consisting of one or multiple services, run on an individual device. The combined capabilities an application provides and the interfaces it uses are described in a standardised interface description, such as the information model of the Asset Administration Shell (AAS) [AAS-D].

However, applications may not only run successfully, but they may also fail or be in different state (e.g., following an application state machine), which shall be exhibited through the application interface. The state of an application may depend to a certain degree on the states of the connected services. As discussed in Section 3.8 on Services, an application shall provide facilities to monitor its execution and diagnose its current state, in particular through monitoring services.

The ECS management system and the IT-infrastructure of the platform are provide and implement the communication between services (of an application), HMI devices, such as dashboards, and data connectors, e.g., on field devices.

3.10 Service store

The service store provides services and offers information on all services, which are available via the IIP-Ecosphere platform. The service store describes a service with regard to the abilities of the service, its assurances, interfaces, properties, available configurations, requirements, compatibility to other services and the possibilities of its combination with other services. The service store makes this information available to higher system levels, such as search engines, for example for a service broker. The service store is part of the ECS management system.

Further, the service store hosts and provides information on applications, composed from services, in the format of combined solution patterns of services (application templates for typical, frequently occurring use cases). The service store provides information on the compatibility of services and the feasibility of specific combinations of services, based on the stored solution patterns of services and the available information on service compatibility.

3.11 Service configuration tool

The service configuration tool provides configuration capabilities for services. The service configuration tool uses information on services from the service store, to support and validate the parametrisation and correct configuration of services by application designers and data scientists. The service configuration tool does so by providing available configurations to select from or by providing information on the requirements and interfaces of services to ensure the services are configured in compliance with their requirements and interfaces. The service configuration tool also allows the setup of generic services to specific tasks and the connection of generic services to specific data sources by an application designer or data scientist. The service configuration tool also uses information provided by the device configuration tool to ensure a configuration of a service is compliant with the configuration parameters of a target device, the service is to be deployed on. The service configuration tool is part of the ECS management system.

It is to be noted that the service configuration tool is exclusively responsible for the (logical) task of configuring individual services. The combination of several services into an application and the

subsequent configuration of any cumulative configuration of the combined services within an application is managed within the service orchestration tool of the platform, as described below.

3.12 Service orchestration tool

The service orchestration tool focuses on the (logical) task of connecting configured services to an application. As an output, an application is offered including relevant application functionalities (e.g., control state machine or error handling) and a digital information model (AAS) which includes all relevant information of the individual services as well as the composite application.

The service orchestration tool uses information provided by the service configuration tool and the service store (e.g. information about services or application templates), to suggest possible combinations of services to the application designer or data scientist for orchestrating an application from multiple individual services available from the service store. The service orchestration tool also validates possible combinations of services with regard to their compatibility, overall requirements with regard to computational effort, execution time and other criteria of the overall application being orchestrated from the services. Furthermore, the service orchestration tool validates that the individual configurations of the services to be orchestrated are not conflicting, e.g., with respect to service versions or service dependencies. Furthermore, the service orchestration tool inserts additional relevant functionality to control and use the entire application (e.g. a distributed control state machine or error handling). The service orchestration tool is part of the ECS management system.

3.13 Data lake

The data lake provides interfaces and access to storage facilities of production process data from a business/factory as well as related/derived data, which is relevant for the processing within (AI) services of the IIP-Ecosphere platform. The data lake consists of ERP/MES databases as well as of locale data, such as a platform customer's production data.

The production data can originate from all kinds of sources, e.g., field devices via edge devices, the IT-infrastructure, databases of the ERP/EMS Systems of a business or from locale, on premise, databases, the AI, etc. The data lake may offer additional functionality to integrate data (based on information from the service configuration or service orchestration). Furthermore, the data lake provides interfaces to data consumers, e.g., to import external data such as weather data or stock training datasets or to export own data for the use by other stakeholders.

The connection- and access-management to the different databases the data lake consists of is handled by using the respective information from the relations connecting the data lakes to the services. We assume, that ultimately the technical connection and access management may be handled by the IT-infrastructure in the platform, e.g., through respective platform services.

3.14 Device description store

The device description store is a storage containing information on devices or assets, provided by the developer or operator of a device, for all devices that are available in the System under Consideration. Such devices are field devices, edge devices, cloud devices, server devices or data lakes. The device description store describes a device with regard to the abilities of the device, its interface, available configurations, requirements, communication protocols, message formats used, capabilities with regard to data storage, memory, computational capabilities and further criteria, such as minimum response time, time criticality etc. The device description store makes this information available to the device configuration tool. The device description store is part of the ECS management system.

3.15 Device configuration tool

The device configuration tool focuses on the (logical) task of configuring devices using information on services, applications and devices to support and validate the correct distribution of the application, including the specific configuration of devices for an application. The device configuration tool does so by providing available configurations to select from or by using information on the device from the device description storage to validate possible configurations and to prevent invalid configurations of a device. The device configuration tool also supports the configuration or re-configuration of devices to accommodate new or changed applications and services that are to be deployed or updated on the target devices and hence is used by application designers or data scientists. The device configuration tool is part of the ECS management system.

3.16 Runtime application / service distribution tool

The runtime application / service distribution tool supports the (logical) task of managing the dynamic deployment of applications and services, which can be a distributed deployment of applications and services to the specific edge/cloud or server infrastructure via the general IT-infrastructure. The runtime application / service distribution tool also manages adaptations of indicated applications and services at runtime, for example, managing updates of applications and services, re-configurations of running services, deployment of alternative services or migration of services/containers. For both tasks, the distributed deployment and the adaptation, it provides general testing and pre-deployment testing facilities for the applications and services to be deployed or for the adaptations or re-configurations to be deployed for running applications and services. The runtime application / service distribution tool is part of the ECS management system.

3.17 Edge/Cloud/Server (ECS) Management System (IIP-Ecosphere Platform)

An edge/cloud/server (ECS) management system is a software program deployed on an IT-infrastructure. An ECS management system provides configuration capabilities for field devices and edge/cloud/server devices, a store for providing services, edge/cloud/server runtimes and firmware, a device description store and configuration and deployment capabilities for applications, services, edge runtimes and firmware, which are provided via the service store of the ECS management system. [LNI –UV]

The IIP-Ecosphere platform encompasses an ECS management system, as described above, which provides additional capabilities for intelligent industrial production. These additional capabilities are:

- The provision of AI services (to be composed into applications), see Sections 3.8, 3.9 and 3.10.
- The provision of information about the available AI services to service brokers.
- Support for IIoT communication protocols.
- Facilities to provide information to / receive information from external data lakes.
- Connection facilities to connect to Data Science tool chains for the data exploration for AI applications and services and the training and testing of AI applications and services.
- A distribution tool to define/control the distribution and deployment of applications and services at runtime as well as runtime adaptation of applications and services and the (pre-deployment) testing of applications and services

3.18 Data Science tool chain

A Data Science tool chain, though for flexibility reasons not being an integral part of the ECS management system and also not of the IIP Ecosphere platform, is a key component in the development of AI applications, AI application templates or AI services, used by data scientists for the

development of AI applications and services. Data Science tool chains are tool chains maintained by individual developers of AI applications and services and hence these tool chains can vary significantly. This variation is the reason why the IIP-Ecosphere platform/ECS management system does not aim at providing an integrated, uniform, Data Science tool chain but rather provides facilities to interact with various Data Science tool chains and support these for different developers/providers of AI applications.

However, the general purposes of Data Science tool chains are reviewed briefly at this point to provide the basis of the Activities related to Data Science tool chains, which are described in Section 5.2.

The main purposes of a Data Science tool chain are:

- The collaborative exploration of data to be used in AI applications and AI services
- Data gathering and incorporation
- Data pre-processing
- Data augmentation (data visualisation)
- Statistical analysis of data
- The provision of intermediate results for further exploration and planning with customers
- The training of models for AI applications and services
- The testing of models for AI applications and services, including competitive model testing
- Continuous application of models on new data
- Parameter tuning for models

The output of a Data Science tool chain is an (orchestrated) AI-based application or AI service that can be configured as a generic template that is then available via the service store or as a specific AI application / AI service. A specific Ai application / AI service can be deployed, for example in container-based manner, directly in a customer system, provided as compiled executable for direct deployment to a customer's hardware or existing AI environment or it can be provided as a web-service[9].

[9] As in Section 3.16, we name here potential cases while IIP-Ecosphere will probably focus on selected forms.

4 Roles

Roles interact from outside with the entities of the System under Consideration. Depending on the purpose to be described in the Usage View, a role may be a technical system, a human, an organization or even a combination of this. Thus, a specific company with various organizations may implement several roles.

As our work is based on [LNI-UV], we have taken over basic roles from there and will indicate this appropriately. In addition, we have identified further roles as described in the following sections.

4.1 Field device provider

The field device provider is an organization, which develops, manufactures and finally provides physical field devices and associated technical systems. As the vendor of one or more field devices, the device manufacturer is primarily interested in marketing the field device. In addition, the field device provider is interested in information on the usage of the field device to identify potential for optimizing the design and functionality of the field device. The field device provider provides the firmware (including software functionalities) and it is in the responsibility of the user of a field device to install the current, latest, firmware version on the field device. [LNI-UV]

4.2 Edge device provider

The edge device provider is an organization, which develops, manufactures and finally provides physical edge devices and associated technical systems. As the vendor of such devices, the device manufacturer is concerned with the usability and capabilities of the edge device and the hosting capabilities of different edge runtimes. To increase interoperability of automation devices, the edge device provider has a high interest in using and setting standards for field device onboarding and management as well as edge runtime deployment. The edge device provider provides the firmware (including software functionalities) and it is in the responsibility of the user of an edge device to install the current, latest, firmware version on the edge device. [LNI-UV]

4.3 Edge runtime provider

The edge runtime provider is mainly a software development organization providing an edge runtime, which can be deployed on an edge device. The edge runtime provider is interested in providing a system software application that can be deployed and managed on edge devices easily. [LNI-UV]

The edge runtime provider also (should) be interested in providing the facilities for the execution of AI enabled applications and (AI) services, provided via the IIP-Ecosphere platform, within the respective edge runtime.

The IIP-Ecosphere project acts as an edge runtime provider, supplying support runtimes to, for example, support basic functionalities of the platform such as container management and resource monitoring (probably in close collaboration with edge vendors in the IIP-Ecosphere ecosystem).

4.4 Cloud provider

The cloud provider is an organization, which hosts and offers a cloud consisting of various compute resources. A cloud provider is concerned with matching resource offers and demands, while smoothly running compute resources. In the past, cloud providers have defined pseudo-standards on their own, like QoS, machine classes, regional zones, virtual machine types, data transfer mechanisms or billing procedures. However, recently also interoperability among clouds to prevent vendor lock-ins or even among applications running in clouds become appealing [GAIA-X]. Typically, a cloud provider obtains server devices from a server device provider (see Section 4.6). A cloud provider is responsible for the

qualities of the offers, the smooth operation of the cloud as well as for basic administrative services that are otherwise typically in the responsibility of the user, e.g. to install the actual firmware version, drivers or the operating system on a server device.

In contrast, the server device provider develops and offers the BIOS/firmware (including software functionalities) as well as specific system drivers. It is in the responsibility of the cloud provider to install the current, latest, firmware version, drivers and operating system on the hosted server devices.

4.5 Application integration provider (service/application developer)

The application integration provider is mainly a software development organization providing applications for cloud infrastructures. Here, in particular, the application integration provider is responsible for a cloud runtime, which can be deployed on a cloud. The application integration provider is interested in providing a system software application that can be easily deployed and managed on clouds, in particular, if feasible, integrated with the own provided software.

4.6 Server device provider

The server device provider is mainly an organization, which develops, manufactures and finally provides physical server devices and associated technical systems. As the vendor of such devices, the manufacturer is concerned with the usability and capabilities of the device and the hosting capabilities of different runtimes. The server device provider provides the BIOS/firmware (including software functionalities) as well as specific system drivers and it is in the responsibility of the user of a server device to install the current, latest, firmware version, drivers and operating system on the server device.

4.7 Server runtime provider

The server runtime provider is mainly a software development organization providing a server runtime, which can be deployed on a server. The server runtime provider is interested in providing a system software application that can be deployed and managed on server devices easily.

4.8 Platform provider

The platform provider is a software development organization providing a platform that is compliant with the described ECS management system, for example, the IIP-Ecosphere platform running on some IT-infrastructure.

4.9 Data scientist

A data scientist uses a Data Science tool chain (external to the System under Consideration, see Section 3.18) for data exploration in the context of AI services and applications, the training of models for AI services and applications and the testing of developed AI services and applications. A data scientist further uses the service configuration tool of the ECS management system to configure newly developed applications and services as well as to re-configure existing applications and services. A data scientist also uses the service orchestration tool of the System under Consideration to orchestrate applications from a combination of services available in the service store.

4.10 DevOps operator

The DevOps operator is a human actor who uses the runtime application / service distribution tool to manage the dynamic and distributed deployment of applications and services from the ECS management system to edge, cloud or server devices. The DevOps operator further uses the runtime application / service distribution tool for the coordination of the deployment of software updates to edge, cloud or server devices.

4.11 Data provider

A data provider is a person, an organization or a system that provides data to the System under Consideration, in particular the data lake. The provided data can be production data (in the sense of data sharing, e.g. ERP or MES system), but also data, such as weather forecasts, supply-chain status, test data sets, or general knowledge data, which can be integrated in the development of applications and services.

4.12 Data consumer

A data consumer is a person, an organisation or a system, e.g., an underlying IIoT platform, which receives data from the System under Consideration, in particular the data lake. The consumed data can be production data (in the sense of data sharing) or augmented production data (after integrating data received from a data provider). In addition, other available data may (if permissible) be passed on to a data consumer, such as weather data or forecasts, supply-chain status, test data sets, or general knowledge data from applications and services.

4.13 Asset data provider

An asset data provider is an organization and/or customer that provides device descriptions of specific assets to the device description store of the platform, as well as data throughout the whole lifecycle, for example production data of a specific asset, to the external Data Science tool chain(s). The provided data includes type (general information of device type) and instance data (individual device information, like serial number, production date, hardware version, current state, etc.). Both provisions serve the goal to enable the development of applications and services for the specific asset by the application designer and the data scientist.

4.14 Application designer

An application designer is a human actor developing and providing an application (e.g. application template) to the platform. The application designer can use services provided by the service store to orchestrate an application from services, using the service orchestration tool of the System under Consideration. For the orchestration of an application from services, the application designer further uses the service configuration tool of the System under Consideration to configure individual services for their use within the application. The application designer uses the service orchestration tool of the System under Consideration again for connecting configured services among each other and with any data connectors relevant for the application.

4.15 Service developer

The service developer is a human actor developing and providing services, in particular AI services, which can be deployed and executed on an edge, cloud, server device over the IT-infrastructure. The service developer must provide his services via the service store provided by the System under Consideration. The service developer can use (AI) models provided by a data scientist via the service store.

4.16 Broker

A broker is either a provider of brokerage services or a technical facility that uses information on applications and services available via the platform and information about a problem a customer/user of the platform wants to solve by an application or service from the platform. The broker performs a match between the descriptions of available applications and services, for example their application domain or typical problem category and the problem description provided by the customer/user of the platform to recommend the best directly applicable application or service available via the

platform to solve the customers/users problem. If such a direct match is not possible, the broker supports the customer/user in his search for the most suitable available application template or service template to be adapted to the customers/users problem with the least effort needed for the adaptation.

4.17 Plant operator, system integrator and application integrator

While the prior roles are primarily organizations concerned with providing the different applications and services that constitute the System under Consideration, the roles of plant operator, system integrator and application integrator are primarily human actors (belonging to some organization and supported by some technical systems) that acquire these products to operate and utilize them. The plant operators, system integrators and application integrators are interested in minimizing efforts for different tasks, i.e. operation, adaptation and optimisation of a plant (plant operator, application integrator) as well as engineering and commissioning (system integrator). We distinguish these roles, because there are more technical skills required from a system integrator / application integrator for installing and commissioning than “only” application specific configuration skills from a plant operator. [LNI-UV]

The tasks of a system integrator or application integrator, referred jointly to as a system/application integrator from here on, can include the provision of interfaces as well as the configuration of gateways and the overall setup of communication channels between the deployed application and the monitoring / continuous development facilities with which the application needs to communicate, for example the runtime application / service distribution tool. For these tasks, the service configuration tool and the device configuration tool of the ECS management platform are used, as well as the local configuration of ECS devices and runtimes.

4.18 Plant manager

The plant manager is a human role in the organization operating the plant. The plant manager (e.g. executive board, controlling or even service management) is concerned with the smooth operation of the plant utilizing the IT-infrastructure via the applications to monitor or control the current operations. The plant operator acts under supervision of the plant manager.

4.19 IT-infrastructure

The IT-infrastructure is a technical system. We assume that this infrastructure is provided and operated by some IT-infrastructure operators or the platform provider. We also assume that the IT-infrastructure has capabilities to explore connected edge/cloud/server devices; this includes automatic detection including connection and disconnection of edge/cloud/server devices. In the case of mobile edge devices, this relates to automatic detection as soon as such a mobile device is in the “reach” of the IT-infrastructure or leaves it again; but this also includes a status monitoring to what extent an edge/cloud/server device is still “alive” or can be reached. An IT-infrastructure formulates its assertions with respect to applications deployed on the IT-infrastructure. [LNI-UV]

From a logical point of view, we divide the IT-infrastructure into an area that is owned by the plant operator, who operates the field devices and edge devices, and an area that is not owned by the plant operator, e.g., a cloud. If applications are deployed on the area, which the plant operator does not own, and require access to applications deployed on the area, which the plant operator owns, or on one of the plant operator’s edge devices, the plant operator must grant the access in order for the applications to work properly. [LNI-UV]

Security within the IT-infrastructure is realized in terms of the platform-wide security of the IIP-Ecosphere platform, e.g., through a server installation or a platform service. Hence, various security

mechanisms for authenticating and securing platform-internal/external communications are assumed to be in place. In this version of the Usage View, we also assume that this underlying platform-wide security also covers any need for data protection within the IT-infrastructure.

Furthermore, within the IT-infrastructure, platform services are provided that support all aspects of monitoring tasks, such as the monitoring of operational states of applications, heartbeat functions for assets or the monitoring of available capacities, such as storage facilities, or centralized error handling.

5 Activities

In this section, we elaborate the activities that roles can perform on the entities of the System under Consideration. While some activities have been taken over from [LNI-UV] as indicated below, the IIP-Ecosphere partners have contributed many specific activities for the IIP-Ecosphere platform during the conducted workshops.

In several locations, we will refer to edge devices, server devices and cloud. However, to keep the description short, we will just refer to ECS devices, implicitly naming the cloud as “cloud device” but actually referring to the introduced term “Cloud”. Similarly, we refer to the three types of runtime software as ECS runtime, while actually meaning the corresponding instance for the actual context. Furthermore, only the system/application integrator is mentioned as a representative of the two roles system integrator and application integrator.

As done in [LNI-UV], we assume that field devices are exclusively connected to edge devices. In field device specific activities, we thus refer explicitly to field/edge devices. In some cases, tasks also refer to local devices without correspondence to cloud devices, i.e., we then refer to an edge/server device.

5.1 Activities related to the ECS management system

In this section, all activities that can occur within the ECS management system are described in an initial form. A selection of activities is based on activities described in the Usage View document LNI 4.0 Testbed Edge Configuration – Usage View [LNI-UV]. For each activity that is based on the LNI 4.0 Testbed Edge Configuration – Usage View, it is stated if the activity originates from the 4.0 Testbed Edge Configuration – Usage View or if it is an entirely new activity within the IIP-Ecosphere platform. If an activity originates from the 4.0 Testbed Edge Configuration – Usage View it is further stated if the activity is an original, none-adapted activity from the 4.0 Testbed Edge Configuration – Usage View or if it was slightly adapted from the 4.0 Testbed Edge Configuration – Usage View or if it was heavily adapted from the 4.0 Testbed Edge Configuration – Usage View. The information on the origin and grade of adaptation of an activity is always given as the last comment in each activity description.

All activities are described in the following format: A list of roles and/or events that can trigger an activity explicitly or implicitly is given. This list of triggers for the activity is followed by a workflow description, listing the individual tasks of the workflow of the activity, whereas tasks can be optional, indicating their necessity only in certain situational-contexts in which the activity can occur. After the workflow description the effects of the activity are listed, as well as any constraints that apply on the execution of the activity are given. The activity description ends with a list of any comments related to the activity.

5.1.1 Adding entities

This section describes activities that occur during the addition of entity instances, namely field devices, ECS devices, ECS runtimes and ECS management systems, to the IIP-Ecosphere platform.

5.1.1.1 Adding a field device

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1: Physical installation of field device including connecting to ECS devices and some specific engineering: role system/application integrator
- Task 2: Detection and acknowledgement of the field device by connected ECS devices (onboarding of the field device on the ECS layer): executed automatically

- Task 3: Detection and acknowledgement of the field device by the ECS management systems connected to the affected ECS devices (onboarding of the field device on the ECS management layer): executed automatically
- Task 4: Provision of necessary parameters (including but not limited to communication parameters) to users of ECS management systems: executed automatically
- Task 5: Setting of these parameters from the ECS management systems: role system/application integrator (or plant operator)
- Task 6: Creation of the corresponding data connectors in the connected ECS devices: executed automatically
- Task 7: Initiation of download of parameter values to field device: role system/application integrator (or plant operator)
- Task 8: Download of parameter values to field device and notification about result (successful, etc.): executed automatically
- Task 9: Acknowledgement: role system/application integrator

Effects:

- Reduction of the effort for commissioning by allowing the setting of parameters from an ECS management system. This includes the onboarding processes executed automatically as well as streamlined commissioning processes based on a common ECS management system.

Constraints:

- The connectivity between field devices and edge devices must comply with some “standard” provided by the edge devices.
- The connectivity between edge devices and to an ECS management system must comply with some “standard”, which does not exist yet.

Comments:

- The actual onboarding mechanisms on the field layer can vary and are not subject to the work of the IIP-Ecosphere project.
- The principles of onboarding mechanisms to ECS management systems are subject to the work of the IIP-Ecosphere project.
- Task 1 is not necessary if an already network-connected field device is additionally connected to another edge device.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.1.2 Adding an ECS device

Triggers: Explicitly triggered by role system/application integrator in case that a physical installation (including the communication link) is necessary or triggered by the IT-infrastructure in case the IT-infrastructure detects a new device

Workflow:

- Task 1 (optional): Physical installation of edge/server device including connectivity to IT-infrastructure (here also for cloud devices) and connection to field devices: role system/application integrator

- Task 2: Detection and acknowledgement of the edge/server devices (including automatic detection of relevant communication parameters and creation of the corresponding data connectors, onboarding of the devices on the edge/device layer): executed automatically
- Task 3: Detection and acknowledgement of ECS device by the ECS management systems (onboarding of edge devices on the ECS management layer): executed automatically
- Task 4 (optional): Acknowledgement: role system/application integrator

Effects:

- Reduction of the effort for commissioning by allowing the setting of parameters from an ECS management system. This includes the onboarding processes executed automatically as well as streamlined commissioning processes based on a common ECS management system.

Constraints:

- The connectivity to ECS devices must comply with some “standard”.
- The connectivity to an ECS management system must comply with some “standard”, which does not exist yet.

Comments:

- The principles of onboarding mechanisms to an ECS management system are subject to the work of the IIP-Ecosphere project.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.1.3 Adding an ECS runtime

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1 (optional): Depending on the capabilities of the ECS device and requirements of the ECS runtime to the underlying device, the ECS device is brought into a defined state and maybe taken out of regular operation: role system/application integrator
- Task 2: Selecting an ECS runtime from the application store of the edge provider or the service store provided by an ECS management system: role system/application integrator
- Task 3: Deploying the selected ECS runtime on the ECS device using the ECS management system: role system/application integrator
- Task 4: Starting the execution of the ECS runtime on the ECS device using the ECS management system: role plant operator
- Task 5 (optional): Depending on the capabilities of the ECS device and requirements of the ECS runtime to the ECS device, the ECS device is brought into an operative state: role system/application integrator
- Task 6: Acknowledgement: role system/application integrator

Effects:

- Reduction of the effort for adding and removing an edge runtime by using an ECS management system.

Constraints:

- The ECS device must provide the capabilities requested by the ECS runtime.

Comments:

- Because this activity may require a system understanding of the ECS device and the ECS runtime, we assume that the system/application integrator must perform the activity.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.1.4 Adding an ECS management system

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1: Evaluation if the ECS management system is compatible and compliant to all standards of the IIP-Ecosphere platform and to further established industry standards: role system/application integrator
- Task 2: Installation of the ECS management system on IT-infrastructure: role system/application integrator
- Task 3: Detection and acknowledgement of all connected ECS devices (including automatic detection of relevant information of ECS devices, which includes in the edge device case the information on the underlying field devices as well, onboarding of ECS devices on the ECS management layer): executed automatically
- Task 4 (optional): Acknowledgement: role system/application integrator

Effects:

- No vendor lock-in of a provider of an ECS management system.

Constraints:

- The connectivity to an ECS management system must comply with some “standard”, which does not exist yet.

Comments:

- In IIP-Ecosphere, the ECS management system will be an integral part of the IIP-Ecosphere platform, i.e., installing the ECS management system implies installing the (central parts of) the IIP-Ecosphere platform.
- The principles of onboarding mechanisms to an ECS management system are subject to the work of the IIP-Ecosphere project.
- The ECS management system will collect all application specific data connectors managed by the connected ECS devices and make them manageable.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.2 Removing entities

This section discusses activities that remove entity instances, namely field devices, ECS devices, ECS runtimes and ECS management systems, from the System under Consideration.

5.1.2.1 Removing an ECS management system

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1: Notification of removal to all connected ECS devices: executed automatically after triggering
- Task 2 (optional): Acknowledgement to connected ECS devices: role system/application integrator

Effects:

- No vendor lock-in of a provider of an ECS management system.

Constraints:

- The connectivity to an ECS management system must comply with some “standard”, which does not exist yet.

Comments:

- The principles of offboarding mechanisms to an ECS management system are subject to the work of the IIP-Ecosphere project.
- The removal should happen without any disturbance of overall organizational processes. After removal of an ECS management system, the adding of a new ECS management system follows the process as described in the activity 5.1.1.4 “Adding an ECS management system ”.
- The management of application specific data connectors is independent from the used ECS management system. Thus, removing an ECS management system has no impact on the usage of application specific data connectors. However, information on the created connectors may have to be transferred among the concerned instances of the ECS management system.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.2.2 Removing an ECS runtime

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1 (optional): Depending on the capabilities of the ECS device and requirements of the ECS runtime to the ECS device the ECS device is brought into a defined state and maybe taken out of regular operation: role system/application integrator
- Task 2: Removing the ECS runtime using the ECS management system: role system/application integrator
- Task 3 (optional): Depending on the capabilities of the ECS device and requirements of the ECS runtime to the ECS device the ECS device is brought back into an operative state: role system/application integrator
- Task 4: Acknowledgement: role system/application integrator

Effects:

- Reduction of the effort for adding and removing an ECS runtime by using an ECS management system.

Constraints:

- Before removing an ECS runtime, all applications deployed on the ECS runtime should be uninstalled.

Comments:

- Because this activity may require a system understanding of the ECS device and the ECS runtime, we assume that the system/application integrator must perform the activity.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.2.3 Removing an ECS device

Triggers: Explicitly triggered by role system/application integrator in case that a physical de-installation (including the communication link) is necessary or triggered by the IT-infrastructure in case the IT-infrastructure detects that an ECS device is not connected any longer.

Workflow:

- Task 1: Notification of removal to all connected ECS management systems: executed automatically after triggering
- Task 2: Notification of removal to all affected applications deployed on the IT-infrastructure: executed automatically
- Task 3: Engineering of communication between ECS device and connected field devices (removing communication from field device to specific ECS device, offboarding of ECS devices): role system/application integrator
- Task 4 (optional): Acknowledgement to connected ECS management systems: role system/application integrator

Effects:

- Reduction of the effort for commissioning by allowing the setting of parameters from an ECS management system. This includes the offboarding processes executed automatically as well as streamlined commissioning processes based on a common ECS management.
- There may be now disconnected field devices and there may be now data connectors of applications deployed and running on the IT-infrastructure, which are no longer supplied with data or data provided by the applications is no longer consumed.

Constraints:

- The connectivity to field device must comply with some “standard” and it is assumed that this is provided by the ECS device.
- The connectivity to an ECS management system must comply with some “standard”, which does not exist yet.

Comments:

- The principles of offboarding mechanisms from an ECS management system are subject to the work of the IIP-Ecosphere project.
- The removal of an edge device could have the implication that some field device cannot communicate any longer with an edge device. It is in the responsibility of the system/application integrator to take care of this, e.g. by connecting the field device to another edge device, see activity “Adding a field device” system/application integrator.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.2.4 Removing a field device

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1: Deinstallation of field device and engineering of communication between field device and connected ECS devices (removing communication from field device to ECS devices, offboarding of field devices): role system/application integrator
- Task 2: Notification of removal to all connected ECS management systems: executed automatically after trigger
- Task 3: Removing all data connectors in applications on ECS devices formerly connected to the field device: executed automatically
- Task 4: Disabling of configuration capabilities of ECS management system with respect to the removed field device (offboarding of field devices on ECS management layer): executed automatically
- Task 5 (optional): Acknowledgement: role system/application integrator

Effects:

- Reduction of the effort for commissioning by allowing the setting of parameters from an ECS management system. This includes the offboarding processes executed automatically as well as streamlined commissioning processes based on a common ECS management.
- Because of removing data connectors there may be now data connectors of applications, which are no longer supplied with data or data provided by the edge devices is no longer consumed.

Constraints:

- The connectivity to the field device must comply with some “standard” and it is assumed that this is provided by the edge device.
- The connectivity to an ECS management system must comply with some “standard”, which does not exist yet.

Comments:

- The principles of offboarding mechanism for an ECS management system are out of the scope of the IIP-Ecosphere project.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.3 Provision of entities

This section discusses activities that occur during the provision of entity instances, namely field devices, ECS devices, ECS runtimes and ECS management systems, applications and updates for firmware for the IIP-Ecosphere platform. This section further describes activities that occur during the updating of an edge runtime, discontinuation of an edge runtime, updating an ECS management system, discontinuation of an ECS management system and updating of an application.

5.1.3.1 Provision of a field device

Triggers: Explicitly triggered by role field device provider

Workflow:

- Task 1: Developing and providing a field device complying with a “standard” between field device and edge layer: role field device provider

Effects:

- A system/application integrator or plant operator can buy a field device from the device manufacturer complying with the “standard” between field device and edge layer.

Constraints:

- We assume that the edge device provider provides a “standard” for connecting field devices to his edge devices and assume that in the future this could be based on an OPC-UA companion specification. Therefore, field device provider should comply with such OPC-UA companion specifications in the future.

Comments:

- This activity not (yet) adapted, it is the original from the LNI Edge Usage View.

5.1.3.2 Provision of an edge device

Triggers: Explicitly triggered by role edge device provider

Workflow:

- Task 1: Developing and providing an edge device complying with the “standard” between field device and edge layer and the “standard” between edge and edge management layer: role edge device provider

Effects:

- A system/application integrator or plant operator can buy an edge device from the edge device provider complying with the “standard” between field device and edge layer and the “standard” between edge and edge management layer.

Constraints:

- The edge device provider must provide a “standard” for connecting field devices to his edge devices. We assume that in the future this will be based on OPC-UA companion specifications. Therefore, the “standard” between edge devices and field devices is out-of-scope of the testbed.
- The edge device provider should deliver his edge device by default with a pre-installed edge runtime.

- The connectivity between edge devices and an edge management system must comply with some “standard”, which does not exist yet. The requirements, concepts and solution approaches for such a “standard”.

Comments:

- This activity is not (yet) adapted, it is the original from the LNI Edge Usage View.

5.1.3.3 *Provision of an ECS runtime*

Triggers: Explicitly triggered by role edge runtime provider

Workflow:

- Task 1: Developing and testing an ECS runtime: role ECS runtime provider
- Task 2: Providing the ECS runtime in the service store of an ECS management system: role ECS runtime provider

Effects:

- A user of an ECS runtime can now use the ECS runtime to be installed on ECS devices.

Constraints:

- It is in the responsibility of the user of an ECS runtime to install the current ECS runtime version on the ECS device.
- It is in the responsibility of the ECS runtime provider to notify potential users of the ECS runtime adequately.

Comments:

- From a conceptual point of view, an ECS runtime is a specific application: It is system software, which must be deployed on ECS devices to enable them so that applications can be deployed. Depending on the capabilities of the ECS device and requirements of the ECS runtime to the ECS device, however, certain precautions must be taken during installation and specific system understanding may be necessary.
- This activity was slightly adapted from the LNI Edge Usage View.

5.1.3.4 Discontinuation of an ECS runtime

Triggers: Explicitly triggered by role ECS runtime provider

Workflow:

- Task 1: Notification of potential users of the discontinuation of the ECS runtime: role ECS runtime provider

Effects:

- A user of an ECS runtime will no longer be supported by updates of the ECS runtime.

Constraints:

- It is in the responsibility of the user of an ECS runtime to take appropriate countermeasures.

Comments:

- This activity was slightly adapted from the LNI Edge Usage View.

5.1.3.5 Provision of an ECS management system

Triggers: Explicitly triggered by role ECS management system provider

Workflow:

- Task 1: Developing an ECS management system complying to the “standard” between ECS and ECS management layer: role ECS management system provider
- Task 2: Providing the ECS management system on the IT-infrastructure: role ECS management system provider

Effects:

- A system/application integrator or plant operator can buy an ECS management system from the ECS management system provider complying with the “standard” between ECS and ECS management layer.

Constraints:

- None

Comments:

- The connectivity between an ECS management system and ECS devices must comply with some “standard”, which does not exist yet. The requirements, concepts and solution approaches for such a “standard” are the main subject of the testbed.
- This activity was slightly adapted from the LNI Edge Usage View.

5.1.3.6 Discontinuation of an ECS management system

Triggers: Explicitly triggered by role ECS management system provider

Workflow:

- Task 1: Notification of potential users of the discontinuation of the ECS management system: role ECS management system provider

Effects:

- A user of an ECS management system will no longer be supported by updates of the ECS management system.

Constraints:

- It is in the responsibility of the user of an ECS management system to take appropriate countermeasures.

Comments:

- This activity was slightly adapted from the LNI Edge Usage View.

5.1.3.7 Provision of an application

Triggers: Explicitly triggered by the roles data scientist or application designer

Workflow:

- Task 1: Developing and testing an application (customer specific or generic template) complying to the standards of the service orchestration tool (e.g. interfaces, information

models), based, possibly partial, on existing services available from the service store: role data scientist or application designer
- Task 2 (optional): Providing a semantic description of the application (e.g. for the broker) and a reference to an available source of a dataset for testing the application: role data scientist or application designer
- Task 3: Providing the application in the service store of the platform: role data scientist or application designer

Effects:

- The application can be combined with further value-adding services, such as metrics for evaluation, databases, notification services, evaluation of data maturity analytics.
- Customer-specific distributed and dynamic deployment of the application on target runtimes.
- Announcement of the new application to the broker can lead to the acquisition of new customers, the ability to solve old problems, known to the broker, that have yet been unsolvable by the newly introduced application.

Constraints:

- Applications are sometimes customer specific. They include sensitive data and should only be provided to the customers accordingly.
- The applications requirements, such as computational or storage capabilities, must be met by the operational environment of the application in the platform.

Comments:

- Standards of the service orchestration tool have to be developed.
- This activity was slightly adapted from the LNI Edge Usage View.

5.1.3.8 Provision of an update of firmware

Triggers: Explicitly triggered by role field device provider or ECS device provider

Workflow:

- Task 1: Developing and testing a firmware update: role field device provider or ECS device provider
- Task 2: Providing the firmware update in the service store of an ECS management system: role field device provider or ECS device provider

Effects:

- A user of a field or ECS device can now initiate an update of the firmware of the device.

Constraints:

- It is in the responsibility of the user of a device to install the current firmware version on the device.
- It is in the responsibility of the field or ECS device provider to notify potential users of the field or ECS devices adequately.

Comments:

- From a conceptual point of view, firmware is a specific application: It can be deployed also on field devices and typically, the device must be shut down in the case of updating firmware, whereas applications on ECS devices can be deployed without a need for shutdown of the ECS device.
- This activity was slightly adapted from the LNI Edge Usage View.

5.1.4 Provision of a service and application template

In this section, activities that are related to the provision of a service are described. All activities discussed here apply to all kinds of services that can be affected by a user, in particular AI services or application-specific monitoring services. This includes similar kinds of services that are shipped with the platform as default implementations. However, underlying basic services provided by the platform such as authentication belong to the ECS management system/platform/IT-infrastructure and cannot be affected by the user, as they are required for the basic operations of the ECS management system/platform.

We do not discuss individual deployment/removal/update operations for services, as we believe that the exclusive conduction of such activities can disrupt the consistency of a running application. However, such operations may pave the way for long-running evolving systems and smooth transitions while manufacturing processes (partially) can continue. Thus, we allow for such strategies in Activity 5.1.6.2 Updating an (distributed) application.

5.1.4.1 Provision of a service

Triggers: Explicitly triggered by the roles data scientist or service developer

Workflow:

- Task 1: Developing and testing a single service (customer specific or template-based) complying with the standards of the service configuration tool (e.g. interfaces, information models): role data scientist or service developer
- Task 2 (optional): Providing general restrictions on the distribution of the service, e.g., that sensitive information/code is included and the service may only be deployed on premise or even on certain devices. Such restrictions can be deeply anchored to a service in such a way that the restrictions cannot be overwritten/changed via the service orchestration tool.
- Task 3 (optional): Providing a semantic description of the service (e.g. for the broker) and parameter settings (for AI services, e.g., a reference to an available source of a dataset for testing the service): role data scientist or service developer
- Task 4: Providing the service in the service store of the platform and setting the initial service version: role data scientist or service developer
- Task 5: Logging and documenting the provision of the service: role plant operator, DevOps operator
- Task 6: Communicate the provision of the service to all relevant roles: role plant operator, DevOps operator

Effects:

- The new service is available in the service store and can be combined with other services from the service store into new orchestrated applications.

- The service can be combined with further services from the service store, as well as with other value-adding services, such as monitoring, metrics for evaluation, databases, notification services, evaluation of data maturity analytics, etc.
- Announcement of the new service to the broker can lead to the acquisition of new customers, the ability to solve old problems, known to the broker, that have yet been unsolvable, by the newly introduced service , especially for new AI services.
- Documentation of the provision of the service and communication to all relevant roles is available.

Constraints:

- Services, if customer specific and not generic, can include sensitive data and should only be provided according to the restrictions given in Task 2.
- The requirements of the service, such as computational or storage capabilities, must be met by the operational environment of the service in the platform. This pre-requisite will be checked during deployment.

Comments:

- Standards of the service store have to be developed.
- The deployment of services happens as part of the application deployment of the application containing the services.
- This activity is new and not part of the LNI Edge Usage View.

5.1.4.2 Provision of a an application template

Triggers: Explicitly triggered by the roles data scientist or application designer

Workflow:

- Task 1: Developing and testing an application template as a generic template, for example for a specific kind of AI problems, such as classification, by orchestrating available and newly developed services or their parametrisation(s) (e.g. AI model(s)) into an application template. The orchestration of the services must comply with the standards of the service orchestration tool (e.g. interfaces, information models) and be based on existing services available from the service store: role data scientist or application designer
- Task 2: Pre-configure all services of the application template for which enough information is already available, based on the generic nature of the template, such as minimum storage requirement of the application template.
- Task 3 (optional): Providing a semantic description of the application template (e.g. for the broker) and a reference to an available source of a dataset for testing the application template: role data scientist or application designer
- Task 4: Providing the application template in the service store of the platform: role data scientist or application designer

Effects:

- The application template can be used to fully develop it into customer specific applications.
- Announcement of the new application template to the broker can lead to the acquisition of new customers, the ability to solve old problems, known to the broker, that have yet been unsolvable, by the newly introduced application template.

Constraints:

- The application template can only be developed and provisioned based on available services within the IIP-Ecosphere platform.

Comments:

- Standards of the service orchestration tool have to be developed.
- This activity is new and not part of the LNI Edge Usage View.

5.1.4.3 Discontinuation of a service

Triggers: Explicitly triggered by role service developer

Workflow:

- Task 1: Notification of potential users of the discontinuation of the service: role service developer
- Task 2: Flagging the service in the service store as discontinued: role service developer
- Task 3: Logging and documenting the discontinuation of the service: role plant operator, DevOps operator
- Task 4: Communicate the discontinuation of the service to all relevant roles: role plant operator, DevOps operator

Effects:

- A user of the service will no longer be supported by updates of the service.
- The service is no longer listed/available in the service store.
- Applications using that service cannot be deployed after this activity. However, actually running applications may not be terminated, as operations on the service store shall not affect installed versions.
- Documentation of the discontinuation of the service and communication to all relevant roles.

Constraints:

- It is in the responsibility of the user of the discontinued service to take appropriate countermeasures, e.g. switching to a different service.

Comments:

- Discontinuing a service must not implicitly require removal from the service store as for documentation purposes retired services may be kept in the store. Then, a service is not listed/available although not deleted. However, ultimate deletion is possible and from a service-usage perspective leads to the same effects.
- This activity was slightly adapted from the LNI Edge Usage View.

5.1.4.4 Defining monitoring parameters for services

Triggers: Explicitly triggered by the role application designer, data scientist, service developer, system/application integrator, plant operator, DevOps operator

Workflow:

- Task 1: During the development of an application/service, including the development of AI models, the scope and value ranges of the parameters to be monitored are defined and are validated with regard to the technical possibility of implementing their monitoring with regard to, for example, time criticality or data volume associated with each parameter to be monitored : role application designer, data scientist, service developer
- Task 2 (optional): During the development of an AI model possible heuristics for the parameters to be monitored can be developed: role data scientist, service developer
- Task 3 (optional): Definitions of parameters, their scope and value ranges are validated by the customer that will use the service, analogue to Activity 5.2.2.7 “Provision of intermediate results for further model development with customers” and are made available in the service configuration tool, via the service store: role application designer, service developer, data scientist
- Task 4: During operation, in case of changes in the production environment the service operates in, monitoring parameters for a service may need to be added, altered or removed, if possible this should be done by the operator of the service and be updated in the service configuration tool, via the service store: role plant operator, DevOps operator, in case it is not possible for an operator to make such changes Task 5 needs to be performed
- Task 5 (optional): If an operator role, such as plant operator or DevOps operator, is unable to adjust, add or remove monitoring parameters for a service in production it is necessary to refer the service to a developer role, such as application designer, service developer or data scientist, to have the necessary adjustments to the monitoring parameters implemented, similar to Activity 5.2.2.7 “Provision of intermediate results for further model development with customers”
- Task 6 (optional): Addition, removal and changes of parameters need to be updated / communicated to any visualisation / monitoring / analysis component that is intended to use the new parameter or uses the changed parameter or used the removed parameter. All changes to parameters need to be updated in the service configuration tool, via the service store: role application designer, data scientist, service developer, system/application integrator, plant operator, DevOps operator

Effects:

- In development, the monitoring parameters are established and defined with regard to their characteristics, such as scope and value ranges.
- In operation, the monitoring parameters are updated or revised, including the removal, addition and replacement of parameters.

Constraints:

- For all parameters defined, it has to be ensured that the necessary data for these parameters is available to the service via the ECS management system.

Comments:

- Monitoring can be implemented in various forms, e.g., through probe injection, service code or specific (generic or application-specific) monitoring services. We refer here to monitoring parameters regardless of their realization.

- A standard for the description of monitoring parameters needs to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.1.5 Service configuration and orchestration of an application from services

In this section activities are described that occur during the orchestration of an application from available services, the configuration of the applications and its services, customisation and adaption of an application and its services to specific production environments, for example specific hardware and the generation of a final deployable of an application. The activities described in this section occur during the process of composing or orchestrating an application, including AI applications, from services, including AI services, within the IIP-Ecosphere Platform, involving its service store, service orchestration tool and service configuration tool.

5.1.5.1 Configuring services for use in an application

Triggers: Explicitly triggered by the role application designer, system/application integrator

Workflow:

- Task 1: Retrieve the service requirements for all required services from the service store via the service orchestration tool: role application designer
- Task 2: Retrieving information on the current ECS-characteristics of the prospective area of deployment of the application via the IT-infrastructure: role application designer
- Task 3: All services to be used in the application are configured to meet the required ECS-characteristics of the planned area of deployment for the application with regard to computational capacity, storage capacity, minimum characteristics, such as response times of communication links, by allocating resources and / or setting up communication links. This task follows the activity Task 2: The application requirements are checked against the ECS-characteristics of the ECS devices it is going to be deployed on, following activity 5.1.7.1"Checking for application requirements vs. current ECS-characteristics": role system/application integrator
- Task 4: Any service interdependencies are reviewed and subsequently active services on which the services of the application will depend are re-configured: role system/application integrator

Effects:

- All services of an application are configured and any potential active services on which the application depends are re-configured.

Constraints:

- All requirements to the ECS-characteristics of the planned area of deployment of the application must be met.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.1.5.2 Defining notification triggers (conditions) for monitoring parameters

Triggers: Explicitly triggered by the role application designer, data scientist, service developer, system/application integrator, plant operator, DevOps operator

Workflow:

- Task 1: During the development of an application/service, including the development of AI models, value ranges of single parameters individually as well as constellations of value ranges of parameters (composed conditions) that will trigger notifications are defined and are made available in the service configuration tool, via the service store: role application designer, data scientist, service developer
- Task 2 (optional): The definition of the (automatic) trigger values for dynamic adaptation of services is done and are made available in the service configuration tool, via the service store: role application designer, data scientist, service developer
- Task 3 (optional): The automatic (dynamic) actions to adapt a service, if a trigger is activated, are defined and are made available in the service configuration tool, via the service store: role application designer, data scientist, service developer, system/application integrator, plant operator, DevOps operator
- Task 4 (optional): Definitions of value ranges of single parameters individually as well as constellations of value ranges of parameters (composed conditions) are validated by the customer that will use the service, analogue to Activity 5.2.2.7 “Provision of intermediate results for further model development with customers”: role application designer, service developer, data scientist
- Task 5: During Operation, in case of changes to the production environment the service operates in, value ranges of single parameters individually as well as constellations of value ranges of parameters (composed conditions) may need to be added, altered or removed, if possible this should be done by the operator of the service: role plant operator, DevOps operator. In case it is not possible for an operator to make such changes Task 6 needs to be performed.
- Task 6 (optional): If an operator role, such as plant operator or DevOps operator, is unable to adjust, add or remove value ranges of single parameters individually as well as constellations of value ranges of parameters (composed conditions) for a service in production it is necessary to refer the service to a developer role, such as application designer, service developer or data scientist, to have the necessary adjustments to value ranges of single parameters individually as well as constellations of value ranges of parameters (composed conditions) implemented, similar to Activity 5.2.2.7 “Provision of intermediate results for further model development with customers”, all changes to parameters need to be updated and made available in the service configuration tool, via the service store
- Task 7 (optional): Addition, removal and changes of value ranges of single parameters individually as well as constellations of value ranges of parameters (composed conditions) need to be updated / communicated to any visualisation / monitoring / analysis component that is intended to use these notification triggers or uses the changed notification triggers or used the removed notification triggers all changes to parameters need to be updated in the service configuration tool, via the service store: role application designer, data scientist, service developer, system/application integrator, plant operator, DevOps operator

Effects:

- In development, the notification triggers are established and defined with regard to their characteristics, such as trigger threshold values and/or value ranges.
- In operation, the notification triggers are updated or revised, including the removal, addition and replacement of notification triggers.
- In operation, changes to notification triggers are communicated/integrated to/with all relevant components and roles of the platform, such as visualisation tools and/or monitoring service.
- Upon a trigger, one or more automatic (dynamic) actions will be executed, if the corresponding trigger values are met.
  Optional: One or more automatic (dynamic) actions to adapt a service will be executed, if the corresponding trigger values are met.

Constraints:

- For all notification triggers defined, it has to be ensured that the necessary data for these parameters is available to the service via the ECS management system.
- The platform/ECS management system/IT-infrastructure must provide a mechanism/service to receive the notifications, to pass them on to humans or to handle them automatically/autonomously.
- An automatic (dynamic) action for adapting a service can only be defined from a registry of available actions (services), for example delivered by the service store.

Comments:

- Monitoring can be implemented in various forms, e.g., through probe injection, service code or specific (generic or application-specific) monitoring services. We refer here to monitoring parameters regardless of their realization.
- A standard for the description of monitoring parameters needs to be defined.
- Defined triggers may involve adaptation actions of the platform, e.g., changing data connections, optimizing the deployment, migrating or replacing services.
- This activity is new and not part of the LNI Edge Usage View.

5.1.5.3 Managing service interdependencies

Triggers: Explicitly triggered by the role application designer, data scientist, service developer, system/application integrator, plant operator, DevOps operator, distributed application management system provider. Implicitly triggered by automated changes of services, for example automated fallbacks of services

Workflow:

- Task 1: During the design/composition of a new application or new service the dependencies of all services within the new application or the new service need to be identified and ensured that these interdependent services are available either via the platform or in the target production environment of the new application/service: role application designer, data scientist, service developer
- Task 2: During the Deployment of an application or service to a specific production environment any dependencies of the services within the application or the service to be deployed that are specific to the production environment need to be updated (customised) via the service configuration tool for the application or service being deployed. Any such

customisations/changes need to be stored in the Service store: role system/application integrator

- Task 3: In the case of changes of applications or services in production any resulting changes to the dependencies of the services within the application or the service need to be implemented via the service configuration tool and stored in the service store: role plant operator, DevOps operator
- Task 4: Any changes to dependencies of applications or services being deployed or changed while in production need to be communicated to all relevant roles: system/application integrator, plant operator, DevOps operator
- Task 5: Any changes to dependencies of applications or services being deployed or changed while in production need to be indicated to all subsequent dependent applications or services: role system/application integrator, plant operator, DevOps operator
- Task 6: All changes to service interdependencies need to be updated in the service store and subsequently in the service orchestration tool and service configuration tool: role system/application integrator, plant operator, DevOps operator

Effects:

- All initial interdependencies between services are documented and available via the service store and subsequently in the service orchestration tool and service configuration tool.
- Any changes of interdependencies are communicated and are made available via the service store.
- It is ensured that any application/service being deployed has all its necessary service interdependencies satisfied.
- It is ensured that no changes to services occur that either rely on non-available service interdependencies or would negate an existing service interdependency.
- It is ensured that services are only removed if the removal of the service does not violate any service interdependencies.

Constraints:

- The service interdependencies must be described in a format compliant to the standardised information representation used within the platform to enable their (automatic) integration.
- Any deployment or change to applications and services can only be performed after a prior check of the service interdependencies of the affected application/service. This ensures the availability of all services the application/service depends on (transitive to further service interdependencies).
- Any service removal can only be performed after a check for interdependencies (again transitive) of other applications/services to the service to be removed. This ensures that no other application or service in production is depending on the service to be removed.
- Prior to automatic changes/removals/additions of services the constraints with regards to service interdependencies, as described above, need to be checked automatically.

Comments:

- A standard for the description of service dependencies needs to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.1.5.4 *Composing an* application *from configured* services

Triggers: Explicitly triggered by the role application designer

Workflow:

- Task 1: Retrieval of information on the available services from the service store: role application designer
- Task 2 (optional): Use the support of the problem solution match provided by the broker for the information retrieval for tasks 1 and 2: role application designer
- Task 3: Retrieval of information on data resources from the data lake: role application designer
- Task 4 (optional): For applications for a specific operational environment: Retrieval of information on data sources from the intended operational environment of the application, to be provided by the roles plant operator, plant manager, system/application integrator or DevOps operator: role application designer
- Task 5: Retrieval of information on the structure of data connections from the ECS management system for generic applications: role application designer
- Task 6 (optional): For applications for a specific operational environment: Retrieval of information on the structure of data connections for a specific production environment of the application, to be provided by the roles plant operator, plant manager or system/application integrator or DevOps operator: role application designer
- Task 7: Retrieve information on possible ECS device configurations relevant for the application from the device description store: role application designer
- Task 8: Composing the application from the relevant configured services by defining relations between the services using the service orchestration tool: role application designer
- Task 9 (optional): For applications for a specific operational environment: If any services to be used in the application have not yet been configured, the services are configured using the service configuration tool: role application designer, system/application integrator, DevOps operator
- Task 10: Validation that the composed application is not violating any functional, dependency, communication, or storage constraints: role this validation is done automatically within the IIP-Ecosphere platform
- Task 11: Validation that the composed application is compliant with all information-representation and communication-standards used within the IIP-Ecosphere platform: role application designer and (partially) this validation shall be done automatically within the IIP-Ecosphere platform
- Task 12 (optional): Validate any non-automatically checked constraints of the application: role application designer
- Task 13 (optional): The validation of an application can include testing of the application using digital twins, similar to the corresponding activities during pre-deployment testing: role system/application integrator, DevOps operator
- Task 14: For generic applications: Publish the application on the service store: role application designer

Effects:

- A new application is available and validated, either as a generic application being available from the service store, or as a new application for a specific operational environment to be deployed to this environment.

Constraints:

- The new application must be compliant with all information-representation and communication-standards used within the IIP-Ecosphere platform.

Comments:

- The standards for information-representation- and communication in the platform need to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.1.5.5 Configuring the data connectors of an application

Triggers: Explicitly triggered by the role system/application integrator, plant operator, DevOps operator. Implicitly triggered by automated changes of an application, for example automated fallbacks of applications or one or more of their services or adaptive replacement of system requiring dynamic connection substitutions.

Workflow:

- Task 1 (optional): If the data connectors of an application are to be connected to new ECS device or subsequent field device this task is mandatory as the initial task of this activity, see activity 5.1.3.8 “Configuring an ECS device” for a full description of this task: Connecting data connectors of ECS devices (connected to some hardware points of measurements of a field device) resp. data connectors of applications using an ECS management system: role plant operator
- Task 2: Retrieve data requirements of the application via the service orchestration tool, the application management system: role system/application integrator, plant operator
- Task 3 (optional): Retrieve the data requirements of the application via the runtime application / service distribution tool: role DevOps operator
- Task 4: Retrieve the data storage requirements for the application: role system/application integrator, plant operator, DevOps operator
- Task 5: Allocate the necessary data storage capacities from existing ECS devices or, if these are not sufficient / have not existed yet, from the ECS providers: role system/application integrator, plant operator, DevOps operator
- Task 6: Establish the data source connections, such as field devices via ECS devices: role system/application integrator, plant operator, DevOps operator
- Task 7: Establish data sink connections, such as ECS device long term storage, ECS storage or the data lake: role system/application integrator, plant operator, DevOps operator
  Task 8: Establish the connections between the data connectors and the application/services, for example further applications or services that perform analyses on the data generated by the application: role system/application integrator, plant operator, DevOps operator

Effects:

- The application has its necessary data storage allocated and connected.
- The application has access to the data from the ECS device it is operating on.
- Data flows to further applications or services are connected to the application.

Constraints:

- For all data connections the necessary access rights and credentials need to be provided.

Comments:

- This activity was heavily adapted from the LNI Edge Usage View [LNI-UV].

5.1.5.6 Creating an application *from a template*

Triggers: Explicitly triggered by the role system/application integrator, DevOps operator

Workflow:

- Task 1: Retrieval of information on the available applications from the service store: role system/application integrator, DevOps operator
- Task 2: Selection of a suitable generic application (template): role system/application integrator, DevOps operator
- Task 3: Retrieval of information on data sources from the intended operational environment of the application, to be provided by the roles plant operator, plant manager, system/application integrator or DevOps operator: role system/application integrator, DevOps operator
- Task 4: Retrieval of information on the structure of data connections for the specific production environment of the application runtime, to be provided by the roles plant operator, plant manager or system/application integrator or DevOps operator: role system/application integrator, DevOps operator
- Task 5: Retrieve the necessary information on ECS device configurations relevant for the target specific application runtime from the device description store: role system/application integrator, DevOps operator
- Task 6: If any services to be used in the target specific application runtime have not yet been configured, the services are configured using the service configuration tool: role system/application integrator, DevOps operator
- Task 7: Validation of compatibility of the target specific application runtime with the ECS devices of the intended operational environment on tool, using information from the device description store: role system/application integrator, DevOps operator
- Task 8: Validation that the target specific application runtime is not violating any functional-, dependency-, communication-, or storage-constraints: role this validation is done automatically within the IIP-Ecosphere platform
- Task 9: Validation that the target specific application runtime is compliant with all information-representation and communication-standards used within the IIP-Ecosphere platform: role system/application integrator, DevOps operator and partially this validation is also done automatically within the IIP-Ecosphere platform
- Task 10 (optional): Validate any non-automatically checked constraints of the target specific application runtime: role system/application integrator, DevOps operator
- Task 11 (optional): The validation of an application can include testing of the application using digital twins, similar to the corresponding activities during pre-deployment testing: role system/application integrator, DevOps operator
- Task 12: Compile the target specific application runtime of the application: role system/application integrator, DevOps operator

Effects:

- A new target specific application runtime is available for deployment.

Constraints:

- The new target specific application runtime must be compliant with all information-representation and communication-standards used within the IIP-Ecosphere platform.

Comments:

- The standards for information-representation and communication in the platform need to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.1.6 Setting up operational configurations

This section deals with bringing applications into some operation state, e.g., deploying applications and their services.

5.1.6.1 Deploying an (distributed) application

This activity describes initial basic tasks for the deployment of an (distributed) application. There are optional additional tasks for the deployment of an (distributed) application, beyond establishing its initial operational state, such as pre-deployment simulation, which are described in activity 5.1.7.5.

Triggers: Explicitly triggered by the system/application integrator, DevOps operator

Workflow:

- Task 1: For activation of a new application an initial version number needs to be assigned to the application and made available in the service store, for activation of an existing application the version numbers of available applications must be retrieved, together with information of capabilities, requirements of these versions from the service store, via the service orchestration tool (application consisting of several services) or the service configuration tool (single service): role system/application integrator
- Task 2: The application requirements are checked against the ECS-characteristics of the ECS devices it is going to be deployed on, following Activity 5.1.7.1 "Checking for application requirements vs. current ECS-characteristics"
- Task 3 (optional): Depending on the capabilities of the application and interdependencies of the application , the application's interdependencies, for example other services it is going to interact with, are brought into a defined state that allows the registration / integration of the application to be deployed with these interdependencies: role system/application integrator, DevOps operator
- Task 4 (optional): Perform a pre-deployment simulation of the deployment of the application: role system/application integrator, DevOps operator
- Task 5: Parametrize the application to be deployed, using the and change / update the configuration of all the application's interdependencies: role system/application integrator, DevOps operator
- Task 6: Establish all necessary communication links and data access needs and storage capacities of the application to be deployed for all its interdependencies: role system/application integrator, DevOps operator
- Task 7: Deploy the application to the ECS devices of the production environment via the IT-infrastructure or in case of the role DevOps operator via the runtime application / service distribution tool: role system/application integrator, DevOps operator

- Task 8 (optional): Inform all relevant roles of the deployment of the (distributed) application: role system/application integrator, DevOps operator

Effects:

- The application is deployed to the ECS devices of the production environment and running after deployment.

Constraints:

- The operational environment of the application within the platform must meet the capabilities requested by the application.

Comments:

- For activation of a new application, the initial version number of the application is made available via the service store.
- This activity was heavily adapted from the LNI Edge Usage View [LNI-UV].

5.1.6.2 Updating an (distributed) application

Triggers: Explicitly triggered by the role DevOps operator, plant operator, application designer

Workflow:

- Task 1 (optional): In the context of the provision of an update for an application: Developing and testing an update of an application: role application designer
- Task 2 (optional): In the context of the provision of an update for an application: Providing the update of an application in the service store of an ECS management: role application designer
- Task 3: For updating an existing application, the version number of the application active in a production environment needs to be retrieved from the ECS devices in the production environment on which the service operates. Furthermore the version numbers of available versions (alternative or newer applications) need to be retrieved from the service store, via the service orchestration tool (application consisting of several services) or the service configuration tool (single service): role system/application integrator
- Task 4: For partial updates of an application: Bring the application in a stable state, for example “paused” in which it can be updated, for replacing an entire application with a new version of itself stop the application to be updated.
- Task 5: Any additional or changed requirements of application, induced by the update, are checked against the ECS-characteristics of the ECS devices it is going to be deployed on, following activity 5.1.7.1 "Checking for application requirements vs. current ECS-characteristics"
- Task 6(optional): Remove the old version of the application from the ECS devices of the production environment via the IT-infrastructure or in case of the role DevOps operator via the Runtime application / service distribution tool.
- Task 7 (optional): Deploy the new version of an entire application to the ECS devices of the production environment via the IT-infrastructure or in case of the role DevOps operator via the Runtime application / service distribution tool.
- Task 8 (optional): Remove a part of the application to be updated from the ECS devices of the production environment via the IT-infrastructure or in case of the role DevOps operator via the Runtime application / service distribution tool.

- Task 9 (optional): Deploy the partial application update for the part of the application to be updated on the ECS devices of the production environment via the IT-infrastructure or in case of the role DevOps operator via the Runtime application / service distribution tool.
- Task 10 (optional): Test the updated application in the production environment: role DevOps operator, plant operator
- Task 11 (optional): Communicate the update of the application to all relevant roles: role DevOps operator, plant operator

Effects:

- An operational application is replaced by an updated version of the application.
- In case of the development of an update of an application: A user of the application can now use the update of the application.

Constraints:

- An application update cannot introduce incompatible data parameters or computational, communication, storage requirements within the platform. If it does, it may be considered a replacement of the application, with the subsequent necessary activities to accommodate the additional data parameters, computational, communication, storage requirements of the application within the platform.
- Advanced versions may involve partial replacement or even runtime in-place substitution of services by adaptation. We do only describe the basic principles here.
- Although Task 6-9 are marked optional, at least one of them is mandatory depending on the supported update strategy (see comments).
  In case of the provision of an update for an application, it is in the responsibility of the application designer to notify users of the application adequately.

Comments:

- Task 6-9 allow for removing/re-deploying the entire application as a basis. ECS management system may also allow for a series of incremental changes, either when the application is down or even at runtime while the application (partially) continues running. Such incremental changes may allow for a smooth evolution of an application over time while manufacturing continues. However, such evolutions may be seen sceptical from a practitioner's point of view and, thus, forbidden or just permissible in terms of incremental human approvals. We do not prescribe any strategy or preference here, but it is important that the validation in Task 5 is done before, either to decide for the strategy or to reject unsupported/disruptive incremental changes.
- This activity was heavily adapted from the LNI Edge Usage View [LNI-UV].

5.1.6.3 Uninstalling an (distributed) application

Triggers: Explicitly triggered by the system/application integrator, DevOps operator

Workflow:

- Task 1 (optional): Depending on the capabilities of the application and interdependencies of the application , the application is brought into a defined state and maybe taken out of regular operation: role system/application integrator, DevOps operator
- Task 2 (optional): If the application is not going to be re-used within the platform, remove the application from the service store: role system/application integrator, DevOps operator

- Task 3: Remove the application from the ECS devices of the production environment via the IT-infrastructure or in case of the role DevOps operator via the runtime application / service distribution tool: role system/application integrator, DevOps operator
- Task 4 (optional): If the application was removed from the service store, announce the removal of the application to the broker: role system/application integrator, DevOps operator

Effects:

- The application is stopped and removed from the ECS devices of the production environment.
- (Possible): The application is removed from the service store.

Constraints:

- The removal of an application is only possible if no other operational service or application is dependent on the application to be removed.

Comments:

- This activity was heavily adapted from the LNI Edge Usage View [LNI-UV].

5.1.6.4 Updating the ECS management system

Triggers: Explicitly triggered by role plant operator, ECS management system provider

Workflow:

- Task 1 (optional): In the context of the provision of an update for an ECS management system: Developing and testing an update of an ECS management system: role ECS management system provider
- Task 2 (optional): In the context of the provision of an update for an ECS management system: Providing the update of an ECS management system: role ECS management system provider
- Task 3: Updating the ECS management system using the ECS management system: role plant operator

Effects:

- In case of the development of an update of an ECS management system: A user of an ECS management system can now install the update of the ECS management solution.
- The previous version of the ECS management system will be replaced by the new version.

Constraints:

- When updating an ECS management system, the configuration of the overall “System under Consideration” should be preserved, for example the concrete parameter values, the installed ECS runtimes or the deployed applications.
- It is in the responsibility of the user of an ECS management system to install the actual ECS management system.
- In case of the provision of an update to an ECS management system, it is in the responsibility of the ECS management system provider to notify users of the ECS management system adequately.

Comments:

- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.6.5 Updating an ECS runtime

Triggers: Explicitly triggered by role plant operator, ECS runtime provider

Workflow:

- Task 1 (optional): In the context of the provision of an update for an ECS runtime: Developing and testing an update of an ECS runtime: role ECS runtime provider (optional)
- Task 2 (optional): In the context of the provision of an update for an ECS runtime: Providing the update of an ECS runtime in the service store of an ECS management system: role ECS runtime provider
- Task 3 (optional): Depending on the capabilities of the ECS device and requirements of the ECS runtime to the ECS device the ECS device is brought into a defined state and maybe taken out of regular operation: role plant operator
- Task 4: Updating the ECS runtime using the ECS management system: role plant operator
- Task 5 (optional): Depending on the capabilities of the ECS device and requirements of the ECS runtime to the ECS device the ECS device is brought into an operative state: role plant operator
- Task 6 (optional): Acknowledgement: role system/application integrator

Effects:

- In case of the development of an update of an ECS runtime: A user of an ECS runtime can now use the update of the ECS runtime to be installed on ECS devices.
- The previous version of the ECS runtime will be replaced by the new version.

Constraints:

- It is in the responsibility of the user of an ECS runtime to install the current ECS runtime version on the ECS device.
- In case of the provision of an update to an ECS runtime, it is in the responsibility of the ECS runtime provider to notify users of the ECS runtime adequately.

Comments:

- We assume that – unlike the initial installation of an ECS runtime – the update does not require a dedicated system understanding and therefore the plant operator can execute the activity.
- Depending on the capabilities of the ECS device and requirements of the ECS runtime to the ECS device, however, certain precautions must be taken during the installation of an update. However, we assume that for an installation of an update of an ECS runtime no specific system understanding is necessary.
- This activity was slightly adapted from the LNI Edge Usage View [LNI-UV].

5.1.6.6 Update firmware

Triggers: Explicitly triggered by role system/application integrator or plant operator

Workflow:

- Task 1: The affected field or edge device is brought into a defined state and thereby taken out of regular operation: role system/application integrator or plant operator
- Task 2: The current version of the firmware provided in the application store of the edge management system is installed on the field or edge device, for this purpose usually the device must be shut down: role system/application integrator or plant operator
- Task 3: The field or edge device is brought into a defined state so that it can resume regular operation: role system/application integrator or plant operator

Effects:

- The firmware of the field or edge device is updated.

Constraints:

- When updating firmware, the configuration of the device should be preserved, for example the concrete parameter values, the installed edge runtimes or the installed applications. If, for technical reasons, compatibility restrictions must be considered, the system/application integrator or plant operator should be informed in an adequate way.
- The overall “System under Consideration” must be prepared so that individual devices may not operate for a certain period.

Comments:

- It is in the responsibility of the system/application integrator or plant operator to ensure that the update of firmware does not affect the overall functionality of the “System under Consideration”. This concerns especially the analysis of dependencies between different versions of firmware.
- This may be supported by runtime operations of the ECS management system, e.g., by migrating services before the firmware update.
- This activity was slightly adapted from LNI Edge Usage View [LNI-UV].

The following activities are not based on activities described in the LNI-Edge Usage View document and are new activities that occur within the IIP-Ecosphere platform.

5.1.6.7 Replace an ECS device

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1: The current configuration of the ECS device is retrieved, either form the device or via the device configuration tool: role system/application integrator
- Task 2: Following Activity 5.1.2.3 the ECS device is removed: role system/application integrator
- Task 3: Following Activity 5.1.1.2 the replacement ECS device is added: role system/application integrator
- Task 4 (optional): If the ECS device replaced a malfunctioning ECS device the applications / services that were operating on the malfunctioning ECS device are re-deployed to the replacement device: role system/application integrator

Effects:

- The replacement device is operational with the same configuration as the device that it replaced.

Constraints:

- This activity is only applicable for the replacement of hardware in ECS devices.
- Reusing existing configurations is only possible if the necessary back-up functions were set up for the ECS device.

Comments:

- Configuring the data connectors of an application.
- This activity is new and not part of the LNI Edge Usage View.

5.1.6.8 Restore an ECS device

Triggers: Explicitly triggered by role system/application integrator, plant operator

If backup data of a device is to be used for the restoration of an ECS device, this activity assumes that a backup of an existing configuration for the ECS device is available from the ECS management system. Furthermore, it is assumed that backup data of any operational data the device was operating on is available from either the Data lake or long term storage within the ECS management system.

Workflow:

- Task 1 (optional): The current configuration of the ECS device to be restored is retrieved, either form the device or via the device configuration tool: plant operator, role system/application integrator
- Task 2 (optional): Retrieve recent operational data of the ECS device from the data lake, ECS device long term storage or historian databases: role system/application integrator, plant operator
- Task 3: Any application or service running on the ECS is brought into a stable state that allows for the restoration of the application or service into an operational state: role system/application integrator
- Task 4: Following Activity 5.1.1.2 the ECS device is (re-)added: role system/application integrator, plant operator
- Task 5 (optional): The restored ECS device is configured using the previously retrieved configuration: role system/application integrator, plant operator
- Task 6 (optional): Restore the retrieved operational and recent data to the restored ECS device via the ECS management system and IT-infrastructure: system/application integrator, plant operator
- Task 7: The restored device is newly configured: role system/application integrator, plant operator

Effects:

- The field / ECS device is restored to an operational state.
- In case of using backup data of the device: The ECS device is restored to an operational state with minimal data loss.

Constraints:

- The connectivity between the field ECS device and the ECS management system must still be available.
- Retrieving and restoring operational and historic data is only possible if the necessary back-up / logging functions were set up for the ECS device that needs to be restored.

Comments:

- The ECS management system must comply with some “standards”, which still need to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.1.6.9 Remote management of ECS devices (further administrative actions)

Triggers: Explicitly triggered by role DevOps operator, plant operator, system/application integrator

Workflow:

- Task 1: Establish a connection to the ECS device via the ECS management system and the IT-infrastructure, including the provision of the necessary credentials / access rights: role DevOps operator, plant operator, system/application integrator
- Task 2: Perform any further administrative actions on the ECS device, not yet described in specific Activities.
- Task 3: Close the remote connection to the ECS device via the ECS management system and the IT-infrastructure: role DevOps operator, plant operator, system/application integrator

Effects:

- The intended actions on an ECS device have been executed remotely.

Constraints:

- The ECS device needs to be accessible, hence a restart and or update, in case of failure of an ECS device may not be possible.

Comments:

- The ECS management system must comply with some “standards”, which still need to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.1.7 Additional activities for (distributed) applications

This section describes activities that occur during the Deployment of a (distributed) application, during the setup of an application for an operational environment, during updates of an operational application and during the removal (uninstallation) of an operational application.

5.1.7.1 *Checking for application requirements vs. current ECS-characteristics*

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1: Retrieving the application requirements from the application description in the service store: role system/application integrator
- Task 2 (optional): Retrieving configuration requirements from the service configuration tool: role system/application integrator
- Task 3: Retrieving information on the current ECS characteristics of the prospective area of deployment of the application via the IT-infrastructure: role system/application integrator
- Task 4: Depending on the ECS characteristics of the area of deployment, the application is deployed or not. If not Task 5 follows: role system/application integrator
- Task 5 (optional): Insufficient or incompatible ECS-characteristics of the planned area of deployment for the application are amended, for example by allocating further resources or setting up further communication links, to enable the deployment of the application: roles system/application integrator, edge runtime provider, application integration provider (cloud), server runtime provider

Effects:

- If the ECS characteristics match the requirements of the application, the application can be deployed.
- If the ECS characteristics are not matching the requirements of the application, the ECS-characteristics can either be amended or another application can be considered by the system/application integrator.

Constraints:

- All information on the current ECS-characteristics of the platform, relevant to the application, must be available in the necessary standardised format.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.1.7.2 Simulating the Integration of ECS devices for an application

Triggers: Explicitly triggered by role system/application integrator in case that a physical device installation (including the communication link) is scheduled.

Workflow:

- Task 1: Preparing a simulation of the ECS device, including simulated connectivity to the IT-infrastructure and simulated connection to field devices based on the information available from the device description storage, device configuration tool and the IT-infrastructure: role system/application integrator
- Task 2: Simulating and evaluating the impact of the integration of the ECS device (simulated onboarding, including installation of an ECS runtime): role system/application integrator
- Task 3: Decision on the physical integration of ECS device, based on the outcome of the simulated integration: role system/application integrator

Effects:

- Prevention of the integration of unsuitable ECS devices.
- Possibility to test for optimisation potential by integrating optimal ECS devices, based on their simulation.

- Allocation of optimised amounts of data storage based on simulation.
- Optimised configuration of ECS devices before, after deployment, based on simulation of different configurations of ECS devices.
- Establishing an estimate of the latency of communication with other, existing, systems, for example, data transmission rates.
- Establishing an estimate of the ECS runtime requirements (latency of replies/solutions of the device/system) for later allocation of sufficient resources before the physical deployment of the ECS device.

Constraints:

- The connectivity to an ECS management system must comply with some “standard”, which does not exist yet.

Comments:

- The principles of onboarding mechanisms to an ECS management system, which are to be simulated, are subject to the work of the IIP-Ecosphere project.
- As we do not deploy services/applications here, we have to estimate the information based on functionality of the ECS runtime, which shall be prepared for such simulations.
- This activity is new and not part of the LNI Edge Usage View.

5.1.7.3 Simulating the deployment or update of services and applications

Triggers: Explicitly triggered by role system/application integrator or DevOps operator in case that the deployment of a new application or the update of an operational service or application is scheduled.

Workflow:

- Task 1: Preparing a simulation of the service or application or the updated version of the operational service or application. Such a simulation can be based on digital twins of the relevant devices and services / applications. The simulation includes simulated connectivity to IT-infrastructure and simulated connection to ECS devices and the ECS management system, based on the information available from the service store, service configuration tool, service orchestration tool, runtime application / services distribution tool, IT-infrastructure and ECS management system: role system/application integrator, DevOps operator
- Task 2 (optional): In case of an update of a service or application: Depending on the capabilities of the service or application and interdependencies of the application , the application’s interdependencies, for example other services it is going to interact with, are brought into a defined state that allows the registration / integration of the application to be deployed with these interdependencies: role system/application integrator, DevOps operator
- Task 3: Configure the service or application to be deployed and change / update the configuration of all the service’s or application’s interdependencies: role system/application integrator, DevOps operator
- Task 4: Configure any ECS devices that need to be set up or re-configured for the deployment of the service or application, using the device configuration tool: role system/application integrator, DevOps operator
- Task 5: Establish all necessary communication links and data access needs and storage capacities of the service or application to be deployed for all its interdependencies: role system/application integrator, DevOps operator

- Task 6: Perform the pre-deployment simulation of the deployment of the service or application: role system/application integrator, DevOps operator
- Task 7: Decision on the deployment or update of the service or application, based on the outcome of the simulated deployment or update: role system/application integrator, DevOps operator

Effects:

- Prevention of the deployment of unsuitable services or applications or the deployment of defective updates to services or applications.
- Possibility to test for optimisation potential through simulation of the deployment of potentially better, alternative, services or applications.
- Allocation of optimised amounts of ECS device capabilities for a service or application, based on their simulated deployment and simulated operation.
- Optimised configuration of services and applications, based on simulation of different configurations of a service or application.

Constraints:

- Sufficient information on the service or application, its configuration space, as well as the data it operates on must be available for accurate simulation of the operation of the service or application.
- The operational environment of the application within the platform must meet the capabilities requested by the application.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.1.7.4 Setting up the necessary resources for an application

Triggers: Explicitly triggered by role system/application integrator

Workflow:

- Task 1: Retrieving the application requirements form the application description in the service store: role system/application integrator
- Task 2 (optional): Retrieve configuration requirements from the service configuration tool: role system/application integrator
- Task 3: Retrieving information on the current ECS characteristics of the prospective area of deployment of the application via the IT-infrastructure, following Activity 5.1.7.1 "Checking for application requirements vs. current ECS-characteristics": role system/application integrator
- Task 4: Sufficient and compatible ECS characteristics of the planned area of deployment for the application are set up, with regard to computational capacity, storage capacity, minimum characteristics, such as response times of communication links, by allocating resources and / or setting up communication links, to enable the deployment of the application: roles system/application integrator, edge runtime provider, application integration provider (cloud), server runtime provider, edge device provider, cloud provider, server device provider

Effects:

- The application can be deployed.

Constraints:

- It is in the responsibility of the plant manager or plant operator to provide the necessary access rights and capabilities needed to meet the requirements of the application.
- The principal ability of an application to connect to data connectors must be verified by the ECS management system, such as, for example, data types or inputs versus outputs.

Comments:

- If ECS characteristics already exists, for example for the case of the deployment of an application in an existing production environment, Activity 5.1.3.3 can be performed to establish if the application can be deployed immediately or if further amendments are need to be made to the ECS-characteristics of the production environment before the application can be deployed.
- This activity is new and not part of the LNI Edge Usage View.

5.1.7.5 Integrating external container registries

This activity interacts with container registries providing information of the various containers that are prepared for certain hardware/software combinations, e.g., a certain Linux distribution for ARM processors. Accessing such registries may require user-data (logins, passwords) or certificates management.

Triggers: Explicitly triggered by the roles system/application integrator

Workflow:

- Task 1: Establishing a connection to the external container registry, as a form of external platform: role system/application integrator
- Task 2: Registering new or accessing existing containers with the necessary credentials: role system/application integrator
- Task 3 (optional): Establishing any necessary (automated) connections between a service within the IIP-Ecosphere platform and the external container registry for automated access, providing the necessary credentials: role system/application integrator

Effects:

- An external container registry is accessible from within the IIP-Ecosphere platform.
- New / existing containers can be retrieved / registered in the external container registry.

Constraints:

- The necessary credentials for establishing the connection to the external container registry must made available.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.1.7.6 Changing the state of an application

Triggers: Explicitly triggered by the role system/application integrator, plant operator or DevOps operator. Implicitly triggered by automatic actions, for example triggered by changes in the production environment of a deployed application.

Workflow:

- Task 1: After the initial deployment of a new application the state of the application, after following the Activitiy 5.1.6.1, is changed to active (activated): role system/application integrator
- Task 2: For the change of the state of an active application, e.g., to “maintenance” or “suspended”, the application must be brought to a stable condition, for example halting some processes: role plant operator, DevOps operator or, in case of an implicitly triggered change of state by the ECS management system via the IT-infrastructure
- Task 3: The application’s state is changed to the intended new state. In case of the deactivation of an application this includes the resolution of any affected interdependencies the application has: role plant operator, DevOps operator
- Task 4 (optional): The change of the state of the application is communicated to all relevant roles / entities within the platform.

Effects:

- The state of the application is changed.

Constraints:

- The change of an application’s state to deactivate requires the assurance that no other active service or application is depending on the application.

Comments:

- Some state changes may happen implicitly at runtime, e.g., to perform safe adaptations, depending on the capabilities of the ECS management system. Thus, we do not prescribe potential system states here.
- This activity is new and not part of the LNI Edge Usage View.

5.1.7.7 Restoring an operational application

Triggers: Explicitly triggered by role plant operator, DevOps operator. Implicitly triggered by monitoring facilities if the malfunction of a runnable application is detected.

Workflow:

- Task 1 (optional): Retrieve recent operational data of the runnable application from the data lake, ECS device long term storage or historian databases: role plant operator, DevOps operator
- Task 2: Retrieve the most recent version of the runnable application via the service store or via the runtime application / services distribution tool: plant operator, DevOps operator
- Task 3: Re-deploy a previous version of the runnable application, known as functional and stable to the relevant ECS devices via the ECS management system: roles plant operator, DevOps operator

- Task 4 (optional): Restore the retrieved operational and recent data to the re-deployed runnable application via the ECS management system and IT-infrastructure: roles plant operator, DevOps operator

Effects:

- A working runnable application is restored, optional with restored recent data.
- Optional: A fallback to a stable version of the runnable application is performed, providing an operational environment.

Constraints:

- Retrieving and restoring operational and historic data is only possible if the necessary back-up / logging functions were set up for the runnable application that needs to be restored.

Comments:

- The ECS management system must comply with some “standards”, which still need to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.1.7.8 Visualising the results of an application

To be useful for humans, the results of an application including states, monitoring notifications and deployment to ECS devices shall be subject to visualization (dashboard, human machine interface) to establish a continuous feedback loop from the production data via the HMI components to the plant operator or DevOps operator. However, a visualization can be application-independent or consist (at least to some degree) of application visualization components, e.g., specific gauges or value interpretations.

Triggers: Explicitly triggered by the roles DevOps operator, plant operator, plant manager

Workflow:

- Task 1: The scope and nature of the data to be visualised, if not predefined by the application that is monitored, needs to be specified: role DevOps operator, plant operator, plant manager
- Task 2: The necessary communication links with the data storage facilities, services and the monitoring/notification management are established, providing the necessary credentials and access right via the ECS management system, IT-infrastructure
- Task 3: The specified data is retrieved either from ECS long term storage, service or data storage facility, such as the data lake via the IT-infrastructure
- Task 4: The data is visualised via customised or generic application dashboards

Effects:

- The DevOps operator, plant operator, plant manager are enabled to monitor specific aspects of a production environment.

Constraints:

- Access rights and credentials must be provided to access the data to be visualised.
- Access rights and credentials may have to be provided even for the dashboard.

Comments:

- Standards of access rights representations and generic representations via dashboards have to be defined.
- Standards for the information representation of data / services have to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.2 Activities related to AI services and processes

This section discusses all activities related to the development, provision, deployment and use, in production systems, as well as offline, of AI applications / AI services. The section starts with the discussion of activities that occur within the Data Science tool chain, which focuses on the development and enhancement of AI models for the use within AI applications and AI services. For the activities described in this section AI models, such as a trained neural network, are parametrising the AI service they are part of and enable the AI service to operate. Hence, AI models are parametrisations of an AI service and are thus an integral part of an AI service. If an AI model, used within an AI service, is changed / updated or an AI model is removed, replaced or added, it implies that a re-parametrization of the corresponding AI service is performed and this an updated version of the AI service is made available, employing the updated AI model(s).

5.2.1 Activities within the Data Science tool chain related to data exploration

This section describes all activities related to processes within the Data Science tool chain to perform data exploration processes, implement AI models for use in AI services and evaluating the processes and AI models within the Data Science tool chain.

5.2.1.1 Creating a data exploration processes

Triggers: Explicitly triggered by the role data scientist

Workflow:

- Task 1: Requesting an exploration environment (hardware) for example, “GPU, medium amount of memory, etc.”: role data scientist
- Task 2: Provide the data for exploration (in case of malfunctioning without impact on the running system): role plant operator
- Task 3: Use available tools for testing the data, including external tools: role data scientist
- Task 4: Evaluate the data structure and data quality: role data scientist
- Task 5: Decide if pre-processing or feature engineering is to be done: role data scientist
- Task 6: Establish a mapping between target scope and input (granularity): role data scientist
- Task 7: Pre-selection of learning techniques: role data scientist
- Task 8: Definition of a quality criteria: data scientist
- Task 9: Documenting the exploration process: data scientist

Effects:

- A data exploration process for a specific new AI model/service is performed, yielding the necessary data for the development, refinement of a AI model/service.

Constraints:

- It is in the responsibility of data providers, such as plant manager, plant operator to provide accurate and up to date data for the data exploration process.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.1.2 Data pre-processing

This activity involves the refinement, normalization and addition of data as well as feature engineering based on new or changed data.

Triggers: Explicitly triggered by the role data scientist

Workflow:

- Task 1: The available data, provided by the roles asset data provider, data provider or plant operator, is refined to suit the techniques of the exploration process: role data scientist
- Task 2: The refined data is normalised: role data scientist
- Task 3: Any suitable further data, available via the partners in the collaborative exploration process, is added to existing the data: role data scientist, service developer, application designer, asset data provider, plant operator, plant manager
- Task 4: Based on the results of Tasks 1,2 and 3 further features of the data to be included in the next iteration of the data exploration process: role data scientist, asset data provider, plant operator, plant manager, service developer, application designer, system/application integrator

Effects:

- An initial or refined data set is available for use in the development and / or training of AI models within the Data Science tool chain.

Constraints:

- Sensitive data needs to be handled appropriately with regard to necessary anonymisation, access rights to the data for different roles in the data exploration process.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.1.3 Data visualisation

Triggers: Explicitly triggered by the roles data scientist, service developer, application designer or system/application integrator

Workflow:

- Task 1: Scope of the visualisation is agreed upon: role data scientist, service developer, application designer and system/application integrator
- Task 2: Goals of the visualisation, for example the highlighting of specific data features, are agreed upon: role data scientist, service developer, application designer and system/application integrator
- Task 3: The data is visualised in the agreed scope in a form that implements the agreed goals: role data scientist

Effects:

- Based on the visualisation of the data further refinements of the data can be planned.
- Based on the visualisation of the data further needs for data gathering can be identified.
- New insights from the visualisation of the data can be integrated in the development of AI models and / or services and applications.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.1.4 Statistical analysis of data

Triggers: Explicitly triggered by the roles data scientist, service developer, application designer or system/application integrator

Workflow:

- Task 1: Scope of the analysis is agreed upon: role data scientist, service developer, application designer and optional system/application integrator
- Task 2: Goals of the analysis, for example metrics or threshold values and statistical methods to be used are agreed upon: role data scientist, service developer, application designer and optional system/application integrator
- Task 3: The data is analysed in the agreed scope with the agreed methods in a form that implements the agreed goals: role data scientist

Effects:

- Based on the analysis of the data further refinements of the data can be planned.
- Based on the analysis of the data further refinements of AI models, based on the analysed data, can be identified.
- New insights from the analysis of the data can be integrated in the development of new AI models and / or services and applications.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.1.5 Provision of intermediate results for further exploration planning

Triggers: Explicitly triggered by the roles data scientist, service developer, application designer or system/application integrator

Workflow:

- Task 1: Intermediate results of the data exploration are presented to the roles plant operator, plant manager or system/application integrator: role data scientist

- Task 2: Provision of feedback to the role data scientist comprising the level of sufficiency of the data explored so far and possible further data sources to be included in the next iteration of the data exploration process: roles plant operator, plant manager, system/application integrator
- Task 3: Planning of the next iteration of the data exploration process based on the given feedback: role data scientist

Effects:

- Based on the provision of intermediate results of the data exploration process the scope of the data can be refined.
- Based on the provision of intermediate results of the data exploration process further necessary data sources can be accessed from the data lake, asset data provider, data provider or by capturing of further necessary data via the ECS management system.
- Based on the provision of intermediate results of the data exploration process the service developer, application designer or system/application integrator is enabled to refine requirements for an application or service in development or being integrated.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.2 Activities within the Data Science toolchain related to AI model training/evaluation

This section discusses activities that occur during the initial training and evaluation of AI models, such as neural networks. It is noteworthy that these training and evaluation activities take place before the provision and deployment of the AI models into a production environment, hence are referred to as “offline” training and evaluation activities.

5.2.2.1 Data augmentation

Triggers: Explicitly triggered by the roles data scientist, service developer, application designer

Workflow:

- Task 1: The scope of the augmentation is agreed upon: role data scientist, service developer, application designer
- Task 2: Goals of the augmentation, for example the modification of specific data features or the addition of newly created synthetic data are agreed upon: role data scientist, service developer, application designer
- Task 3: The data is augmented in the agreed scope in a form that implements the agreed goals: role data scientist
- Task 4: The augmented data is evaluated by training or re-training the AI model with the augmented data: role data scientist

Effects:

- Based on the augmentation of the data and the training or re-training of the AI model with the augmented data, the AI model becomes more stable during the training.

- Based on the augmentation of the data and the training or re-training of the AI model with the augmented data, the AI model be trained on more generalized data.

Constraints:

- None

Comments:

- It can be argued that data augmentation is an optional part of the training itself and thus it is not justified to see it as a separate activity. However, in the context of the IIP-Ecosphere platform it is intended that different developers and/or data scientist will access AI models from other developers and/or data scientist and the data these AI models were trained on and thus want to find further possibilities for the augmentation of the data on which these AI models were trained.
- This activity is new and not part of the LNI Edge Usage View.

5.2.2.2 Training of AI models

Triggers: Explicitly triggered by the role data scientist

Workflow:

- Task 1: Obtain suitable data sets from the data lake, asset data provider, data provider for training: role data scientist
- Task 2: Organise data sets in training and testing sets: role data scientist
- Task 3: Select a suitable training method: role data scientist
- Task 4 (optional): Based on Activities 5.2.3.4, 5.2.3.5, 5.2.3.6 change the training method or neural network structure: role data scientist
- Task 5: Run the training process until sufficient quality / accuracy is reached which is evaluated by Activities 5.2.4.4 and 5.2.4.5: role data scientist

Effects:

- An initial trained AI model, which meets sufficient levels of quality / accuracy, is available either for deployment or for evaluation by stakeholders.

Constraints:

- Only insufficient data is available, if so, return to further collaborative data exploration.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.2.3 Testing of AI models

Triggers: Explicitly triggered by the role data scientist

Workflow:

- Task 1: Select suitable testing methods to the initial AI model: role data scientist
- Task 2: Select suitable metrics for the testing methods: role data scientist
- Task 3: Select suitable test data set(s) (data scope) from the available data: role data scientist

- Task 4: Apply the selected testing methods with the selected metrics on the selected test data set: role data scientist
- Task 5 (optional): Apply suitable testing methods to existing AI models based on feedback from the service developer or application designer: role data scientist

Effects:

- A measure of the quality / accuracy of the trained AI model is available for review by the service developer or application designer.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.2.4 Evaluate the training process by scoring the process against different metrics

Triggers: Explicitly triggered by the role data scientist or if based on feedback, then implicitly triggered by the roles service developer or application designer

Workflow:

- Task 1: Define metrics for the scoring of the training process: role data scientist
- Task 2 (optional) Define metrics based on feedback input from the service developer or application designer: role data scientist
- Task 3: Evaluate the AI model against the selected Metrics: role data scientist

Effects:

- An estimate of the suitability and efficiency of the selected training process is available.
- Based on the estimate of the suitability and efficiency of the training process an alternative approach can be selected or the existing training process can be refined further.

Constraints:

- None

Comments:

- It can be argued that evaluating the training process is already an integral part of the training itself and thus it is not justified to see it as a separate activity. However, in the context of the IIP-Ecosphere platform it is intended that different developers and/or data scientist will access AI models and their training processes from other developers and/or data scientist and thus want to evaluate the training processes used for these AI models separately.
- This activity is new and not part of the LNI Edge Usage View.

5.2.2.5 Evaluate the training process against different AI model structures

Trigger: Explicitly triggered by the role data scientist or if based on feedback, then implicitly triggered by the roles service developer or application designer

Workflow:

- Task 1: Select suitable different AI model structures: role data scientist
- Task 2: Train the selected different AI model structures without selecting a different training method or different training data set: role data scientist
- Task 3: Test the outcome of the training process, following Activity 5.2.2.3: role data scientist

Effects:

- An evaluation of the possible benefit / improvement of the AI model, using a different AI model structure, is available.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.2.6 Evaluate the training process against different training structures

Triggers: Explicitly triggered by the role data scientist or if based on feedback, then implicitly triggered by the roles service developer or application designer

Workflow:

- Task 1: Select suitable different training structure: role data scientist
- Task 2 (optional): Alter the existing training structure: role data scientist
- Task 3: Train the AI model using the altered or alternative training structure: role data scientist
- Task 4: Test the outcome of the training process, following Activity 5.2.2.3: role data scientist

Effects:

- An evaluation of the possible benefit / improvement of the AI model, using an altered or alternative training structure is available.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.2.7 Provision of intermediate results for further AI model development

Triggers: Explicitly triggered by the roles data scientist, service developer, application designer

Workflow:

- Task 1: Intermediate results of the data exploration are presented to the roles plant operator, plant manager or system/application integrator: role data scientist
- Task 2: Provision of feedback to the role data scientist comprising the level of sufficiency of the data explored so far and possible further data sources to be included in the next iteration

of the data exploration process: roles plant operator, plant manager, system/application integrator
- Task 3: Planning of the next iteration of the data exploration process based on the given feedback: role data scientist

Effects:

- Based on the provision of intermediate results of the data exploration process the scope of the data can be refined.
- Based on the provision of intermediate results of the data exploration process further necessary data sources can be accessed from the data lake, asset data provider, data provider or by capturing of further necessary data via the ECS management system.
  Based on the provision of intermediate results of the data exploration process the service developer, application designer or system/application integrator is enabled to refine requirements for an application or service in development or being integrated.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.2.8 Continuous application of the AI model on new data

Triggers: Explicitly triggered by role data scientist, system/application integrator, if new data from a new task, that is suitable for the AI model to be applied to, is available or if an existing task, the AI model is applied to, was modified. This activity can also be triggered if new, freely available additional data becomes available from data providers that can be used to improve the AI model.

Workflow:

- Task 1: Acquire new data from new or modified tasks that the AI model is applied to: role data scientist, system/application integrator
- Task 2 (optional): Pre-process and / or augment the new data before applying the AI model: role data scientist
- Task 3: Evaluate the quality / accuracy of the AI model operating on the new data: role data scientist, system/application integrator

Effects:

- Continuous optimization of productive AI models.
- Possible introduction of additional features of the AI model, based on newly available data.

Constraints:

- New raw data may need to be pre-processed, augmented by the role data scientist, see Task 2, before the system/application integrator can apply the AI model on the new data.

Comments:

- If significant shortfalls in the quality / accuracy of the AI model being applied on the new data is detected a new revision of the AI model or an entirely new AI model needs to be developed.

- This activity is new and not part of the LNI Edge Usage View.

5.2.2.9 Tuning (re-calibration) of the AI model parameters

Triggers: Explicitly triggered by the roles system/application integrator either with the goal of optimising an AI model in use or with the goal of countering / mitigating AI model shift over time, in case of automated monitoring of the quality / accuracy of the AI model. This activity can also be triggered implicitly if the quality / accuracy of an AI model falls below specific quality- / accuracy-thresholds.

- Task 1: Detect the need / opportunity of optimisation of the AI model in use or, if no automatic monitoring of the AI model's quality / accuracy is in place, detect the occurrence of AI model shift with deteriorating results from the AI model in use: role system/application integrator
- Task 2: Supply information on planned optimisations to the data scientist: role system/application integrator
- Task 3 (optional): If no automatic monitoring of the AI model quality / accuracy is in place: Supply information of the AI model shift and its possible reasons to the data scientist: role system/application integrator
- Task 4: Based on the supplied information on optimisation or AI model shift, re-calibrate or introduce / remove parameters to the AI model: role data scientist
- Task 5: Evaluate the re-calibrated AI model, following Activity 5.2.3.3: role data scientist
- Task 6: Once a sufficient new quality / accuracy is reached re-deploy the enhanced / adjusted AI model on the ECS devices of the production environment: role system/application integrator

Effects:

- Continuous optimisation of productive AI models.
- Mitigating, amending deteriorating AI model quality / accuracy by re-calibration of the AI model.

Constraints:

- None

Comments:

- The role data scientist is not assumed to be a trigger for this activity, as the AI model, when it is deployed for the first time is assumed to be validated by the data scientist to be of sufficient quality / accuracy before deployment.
- This activity is a key activity to react to the phenome of AI model-shift, where an initially accurate AI model starts to deliver deteriorating results due to small changes in the environment it is applied to, for example wear of machinery over time in a production environment.
- This activity is new and not part of the LNI Edge Usage View.

5.2.2.10 Backup of labelled data sets (for further learning/training)

Triggers: Explicitly triggered by the roles plant operator. Implicitly triggered by automatic backup intervals

Workflow:

- Task 1: Scope of labelled data set is set: role plant operator

- Task 2: Necessary storage capacity in the data lake or cloud infrastructure is allocated and acquired via the IT-infrastructure: role plant operator and cloud provider
- Task 3: Labelled data is annotated with metainformation and stored in the data lake: role plant operator

Effects:

- A new labelled data set is available for further learning and testing activities.

Constraints:

- The volume of labelled data to be stored must be adequate to the available bandwidth and data storage capacities of the ECS devices on which the service or application operates.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.3 Use of AI applications / AI services within the platform

This section describes the activities that occur during the updating of an AI model/service within an AI application and the replacement of an AI model/service within an AI application.

5.2.3.1 Update an AI service within an AI application

Triggers: Explicitly triggered by the role data scientist or service developer when an existing service was improved and is still compatible regarding its interface/semantic with the deployed version, implicitly triggered if an AI model deteriorates or changes in the environments of a productive AI model occur. This activity could also be used implicitly/automatized to adapt the applications autonomously.

Workflow:

- Task 1: Provision of an updated service increasing the service version: role data scientist, service developer
- Task 2: Provide configuration information of the updated service, e.g., the parameter settings to a co-evolved AI model, to the service orchestration tool and the service configuration tool: role data scientist, service developer
- Task 3: Make the service (implicitly a referencing parameter setting, e.g., to an AI model) available to the system/application integrator via the service store and increment the service version number: role data scientist: data scientist, service developer
- Task 4: Announce the availability of the updated service and information on its capabilities and requirements to the broker: role data scientist, service developer
- Task 5: Configure the updated service for its deployment on the ECS devices of a production environment: role system/application integrator, DevOps operator
- Task 6: Deploy the updated service on the ECS devices of the production environment planned for the service, if it is a tailored service for a specific task: role system/application integrator, DevOps operator
- Task 7: Logging and documenting the update of the service: role plant operator, DevOps operator
- Task 8: Communicate the update of the service to all relevant roles: role plant operator, DevOps operator

Effects:

- Continuous (potentially autonomous) optimization of productive AI models.
- Correction of defective or deteriorating services (implicitly AI models).
- Documentation of the update of the service and communication to all relevant roles.

Constraints:

- An updated service must be interface-compliant with its already deployed predecessor.
- An updated service shall be with its already deployed predecessor regarding the execution semantics, for AI services including AI model semantics, predictions.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.3.2 Replace an existing AI service

Replacing an AI model/service in an application or on an ECS device is only possible if the AI model/service is compliant with the interfaces and semantic structure of the AI model/service that is to be replaced, as it otherwise it would need to change also the application in which the AI model/service is being used. In that case, an update to the application would be necessary.

Triggers: Explicitly triggered by the role data scientist or service developer when an entirely new AI model is available, implicitly triggered if an AI model deteriorates or changes in the environments of a productive AI model occur

Workflow:

- Task 1: Provide configuration information of the new AI model to the service orchestration tool and the service configuration tool: role data scientist, service developer
- Task 2: Make the new AI model available to the system/application integrator via the service store: role data scientist: data scientist, service developer
- Task 3: Announce the availability of the new AI model and information on its capabilities and requirements to the broker: role data scientist, service developer
- Task 4: Remove the existing AI model from the ECS devices of the production environment the existing AI model was applied on: role system/application integrator
- Task 5: Configure the replacement AI model for its deployment on the ECS devices of a production environment: role system/application integrator
- Task 6: Deploy the replacement AI model on the ECS devices of the production environment planned for the AI model, if it is a tailored AI model for a specific task: role system/application integrator

Effects:

- Continuous optimisation of productive AI models is applied.
- Correction of defective or deteriorating AI models is ensured.

Constraints:

- The replacement AI model must comply with the data input, analysis output interfaces the replaced AI model used within the operational environment in the platform.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.4 Analysis and prediction of the performance and accuracy of AI applications

This section describes the activities that enable and perform the monitoring of the performance and accuracy of AI services and their AI models within an AI application.

5.2.4.1 Provision of metrics for evaluating an application or service

Triggers: Explicitly triggered by the roles data scientist or application designer or service developer

Workflow:

- Task 1: Developing and testing of metrics / prediction quality for AI services (customer specific or generic template) complying to the standards of the service orchestration tool (interfaces, information AI models): role data scientist or application designer or service developer
- Task 2: (optional): Providing a semantic description of the AI application or AI service (e.g. for the broker) and a reference to an available source of a dataset for testing the application or service : role data scientist or application designer or service developer
- Task 3: Providing the metrics / prediction quality in the service store of the platform: role data scientist or application designer or service developer

Effects:

- Enhanced ability to combine the service or application with other services and applications.
- Evaluation of the service or application with the ability to estimate the added-value, monetisation potential, for customer specific as well as generic services and applications.
- Possibility to provide generic services for the analysis of computational effort needed for a service or application.
- Possibility to provide generic services for further analytics, such as data maturity, AI model accuracy etc.

Constraints:

- Metrics are usually customer specific. They include sensitive data (costs, evaluation criteria) and should only be provided to the customers accordingly.

Comments:

- The standards for the service store are to be developed.
- This activity is new and not part of the LNI Edge Usage View.

5.2.4.2 Analysis of metadata to detect deviation or AI model drift

Triggers: Explicitly triggered by the roles data scientist, DevOps operator (plant operator)

Workflow:

- Task 1: The scope and nature of the metadata to be used in the analysis needs to be specified: role DevOps operator, plant operator, plant manager
- Task 2: The necessary communication links with the data storage facilities and / or external data providers are established, providing the necessary credentials and access right via the IT-infrastructure, “Data Access Management System”

- Task 3: The specified data is retrieved from data storage facility, such as the data lake and / or from external data providers via the IT-infrastructure: role data scientist, DevOps operator (plant operator)
- Task 4: The data used within suitable analysis methods to detect deviation of a service / application or AI model drift role: data scientist, DevOps operator (plant operator)

Effects:

- Potential deviations of a service / application or AI model drift are detected.
- Corrections to either the configuration of a service / application can be made or the service / application can be replaced or physical causes, causing the deviation, can be solved by maintenance tasks.

Constraints:

- This activity is limited to non-time critical services and applications.

Comments:

- Standards of metadata descriptions / formats need to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.2.4.3 Monitoring the prediction quality of the service/application

This activity describes just the monitoring of the prediction quality of a service/application; it is not actively testing a service/application for further development. The result of the monitoring in this activity enables the roles plant operator and DevOps operator to trigger further development of the service/application, for example due to deteriorating prediction quality being observed, based on the monitoring described in this activity.

Triggers: Explicitly triggered by the roles plant operator, DevOps operator

Workflow:

- Task 1 (optional): Use either the Activity 5.2.11.3 or 5.2.11.4 for manual analysis of the prediction quality of a service or application: role DevOps operator, plant operator
- Task 2: A scope for stored monitoring data or an interval for continuous data is set: role DevOps operator, plant operator
- Task 3: The necessary communication links with the data storage facilities and / or the connection to continuous data feeds of ECS / field devices are established, providing the necessary credentials and access right via the IT-infrastructure, “Data Access Management System”
- Task 4: The specified data is retrieved from data storage facilities or read from continuous data feeds of ECS / field devices and provided to services / applications which implement the monitoring of the prediction quality of the service / application monitored: role DevOps operator, plant operator

Effects:

- Deviations in prediction quality of a service or an application can be detected and further steps to either reconfigure or replace the service or application, or the DevOps operator or plant operator can initiate the reconfiguration or replacement of the AI model used by the service/application.

Constraints:

- Any monitoring parameters must be compliant with the standards used for information representation within the platform.

Comments:

- Standards of prediction quality need to be defined.
- This activity is new and not part of the LNI Edge Usage View.

5.2.4.4 Comparison of several AI models

This activity occurs during the evaluation of an AI model by the use of challenger AI models, being altered or different AI models applied to the same task as the AI model being evaluated, to ensure a stable quality of the AI model in use within the service/application.

Triggers: Explicitly triggered by the role data scientist or if based on feedback, then implicitly triggered by the roles service developer or application designer

Workflow:

- Task 1: Select suitable different AI model from a suitable repository, for example the service store: role data scientist, service developer or application designer
- Task 2: Integrate the AI model in an existing service or application: roles data scientist, service developer, application designer
- Task 3: Test the outcome of the integrated alternative AI model in the service or application against the AI model in operation, following Activity 5.2.2.3: roles data scientist, service developer, application designer
- Task 4: If sufficient improvement by the use of an alternative AI model in a service or application is detected the new, alternative, AI model is integrated in the service or application in operation (see Activity 5.2.4.3)

Effects:

- An evaluation of the possible benefit / improvement of the service or application by using a different AI model as operational AI model in the service or application.

Constraints:

- Challenger AI models should not exceed the computational / storage requirements of the operational AI model. If challenger AI models exceed these requirements the allocation of additional capabilities within the platform is necessary, should the challenger AI model be chosen to replace the existing AI model.

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.4.5 Evaluating the Quality of services/applications

Triggers: Explicitly triggered by the roles application designer, service developer, plant operator

Workflow:

- Task 1: Select suitable testing methods to evaluate the service / application : role application designer, service developer
- Task 2 (optional): Supply suitable data, testing methods and metrics to the application designer, service developer: role plant manager
- Task 3: Select suitable metrics for the testing methods: role application designer, service developer
- Task 4: Select suitable test data set(s) (data scope) from the available data: role application designer, service developer
- Task 5: Apply the selected testing methods with the selected metrics on the selected test data set: role application designer, service developer
- Task 6 (optional): Provide information on the quality of a service / application to the broker: role application designer, service developer

Effects:

- A measure of the quality of the service / application is available for review by the service developer, application designer or plant operator.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.5 Using AI services and AI applications manually, offline use

This section discusses activities that can occur if an AI service/application is used manually, for example by a data scientist or service developer, comparable to the use of a standalone AI application, in a way comparable to an assistance system in an offline, meaning not directly linked to live production data, scenario.

5.2.5.1 Classification based on the use of existing, stored, data

This activity describes the use of existing, stored, data from past production, to be used for classification AI model development or refinement by tasks as, for example: pattern finding in existing image data, pattern matching with data lake data, etc.

Triggers: Explicitly triggered by the role data scientist to establish new or refine existing classification methods or to establish new patterns

Workflow:

- Task 1: Establishing the necessary scope and kind of data: role data scientist
- Task 2: Establish connection to the data provider, for example the data lake or long term ECS storage: role data scientist (this includes also the acquisition of the necessary access rights and provision of credentials)
- Task 3: Establish connections to the data provider of a pre-defined scope and kind of data, for example the data lake or long term ECS storage

- Task 4: Retrieval of the (pre-) defined scope and kind of data: role data scientist
- Task 5: Use of the data to perform the classification or other specific tasks, such as searching for new patterns: role data scientist

Effects:

- The outcome of the classification or pattern search task can be used to refine or develop. existing or new classification methods.
- Newly found patterns can be integrated in future AI models or classification methods.

Constraints:

- None

Comments:

- This activity is new and not part of the LNI Edge Usage View.

5.2.5.2 Accessing historical data (historian databases) for AI tasks

This activity describes the access of historical data, for example from historian databases, for offline AI tasks such as prediction, classification and recommendation of further development / refinement of AI services.

Triggers: Explicitly triggered by the role data scientist in the context of the development of a new AI model or implicitly triggered by a service or application

Workflow:

- Task 1: Establishing the necessary scope and kind of historical data: role data scientist
- Task 2: Establish connection to the data provider, for example the data lake or long term ECS storage: role data scientist (this also includes also the acquisition of the necessary access rights and provision of credentials)
- Task 3: (for service or applications): Establish connections to the data provider of a pre-defined scope and kind of historical data, for example the data lake or long term ECS storage
- Task 4: Retrieval of the (pre-) defined scope and kind of data by a service or application or by the role: data scientist
- Task 5: Use of the data to perform prediction, classification or recommendation task within the service or application or by the role: data scientist

Effects:

- The role data scientist can apply the outcome of the prediction, classification or recommendation task to the task of a service or application or to the development / testing of an AI model under development.

Constraints:

- None

Comments:

- Services or applications must be provisioned with the necessary access credentials / access right to automatically retrieve data from historian databases or long-term storage within ECS devices.
- This activity is new and not part of the LNI Edge Usage View.

5.2.6 Further notes on AI and services

Additional to the AI-related activities described above two questions were identified with regard to the AI-related activities. For these two questions, IIP-Ecosphere may identify (initial) solutions, which may lead to further activities and requirements in a subsequent version of this whitepaper. The first of the two questions is:

In the context of distributed, cloud-based, deployment of AI services, which today are currently mainly monolithic solutions, do additional auxiliary conditions or constraints exist that need to be considered/fulfilled, when deploying an AI application, consisting of a number of separate AI services in a distributed way, such as, for example, the deployment to several different cloud runtimes?

Those auxiliary constraints could, for example, be formed by having to ensure that the allocated computational and memory resources in an ECS runtime are adequate for the AI service deployed into the ECS runtime. The second question is the question on general challenges with regard to the use of generic AI services and AI applications composed from them is:

The question how generic an AI service actually can be?

Today most AI solutions are mostly custom tailored, closely adapted solutions to a very specific problem. The question that needs to be answered here is how generic AI services and the subsequent AI applications can be made equally customisable, for example in the form of “80%”-solutions for specific kinds of AI techniques, which only need “20%” customisation to be applicable to a customer’s problem.

A probable solution to using generic AI services could be the use of “wizards”, which means configuration tools to configure the problem that an AI service to be developed is intended to solve before the AI service is then developed from a generic template AI service.

A way to further address the problem of generic AI services being too generic for specific problems and specific environments, in which they are intended to operate, for example specific production environments, is the idea to “compartmentalise” the service store. Large-scale customers of the IIP-Ecosphere platform could be allowed to have their own version of the service store in which they accumulate services, which are already customised to their production environment, and to common categories of problems the customer usually needs to solve with the services. This could have synergies with properties restricting the distribution, e.g., due to legal or IPR reasons.

6 System scope

While the implicit requirements of several activities must clearly be covered by the subsequent development, i.e., functional requirements, architecture and implementation/testing, other activities can be understood as delighters that would be nice to have in a future platform, but which are potentially not part of the set of core functionalities. In this section, we classify the activities discussed here with respect to the system scope, in particular based on the grant contracts of IIP-Ecosphere and the available resources in the project. Therefore, we use three scope categories, namely “must”

(activity/entity shall be realized with high priority), “nice-to-have” (optional for realisation), and “out of scope” (delighter, potential part of future development outside the project’s lifetime). Table 1 summarizes the initial scoping decisions and provides comments to clarify the individual decisions.

Activity/Entity	Scope	Comment
5.1 Activities related to the ECS Management System		
5.1.1 Adding entities (see LNI-UV)		
5.1.1.1 Adding a field device	Out of scope	Field level
5.1.1.2 Adding an ECS device	Must	
5.1.1.3 Adding an ECS runtime	Must	
5.1.1.4 Adding an ECS management system	Must	As part of installing the IIP-Ecosphere platform
5.1.2 Removing entities (see LNI-UV)		
5.1.2.1 Removing an ECS management system	Can	As part of uninstalling the IIP-Ecosphere platform, not really high priority
5.1.2.2 Removing an ECS runtime	Optional	Adding/Running is of higher priority
5.1.2.3 Removing an ECS device	Optional	Adding/Running is of higher priority
5.1.2.4 Removing a field device	Out of scope	Field level
5.1.3 Provision of entities		
5.1.3.1 Provision of a field device	Out of scope	Field level
5.1.3.2 Provision of an edge device	Out of scope	Edge device provider
5.1.3.3 Provision of an ECS runtime	Must	
5.1.3.4 Discontinuation of an ECS runtime	Optional	Adding/Running is of higher priority
5.1.3.5 Provision of an ECS management system	Must	
5.1.3.6 Discontinuation of an ECS management system	Can	As part of uninstalling the IIP-Ecosphere platform, not really high priority
5.1.3.7 Provision of an application	Must	
5.1.3.8 Provision of an update of firmware	Can	May be done via an edge-specific service, but not of high priority
5.1.4 Provision of a service and application template		
5.1.4.1 Provision of a service	Must	
5.1.4.2 Provision of a an application template	Optional	Basic functionality is of higher importance, anyway important
5.1.4.3 Discontinuation of a service	Optional	Basic functionality is of higher importance, anyway important
5.1.4.4 Defining monitoring parameters for services	Must	
5.1.5 Service configuration and orchestration of an application from services		
5.1.5.1 Configuring services for use in an application	Must	

5.1.5.2 Defining notification triggers (conditions) for monitoring parameters	Must	
5.1.5.3 Managing service interdependencies	Must	Represents a more complex but important case
5.1.5.4 Composing an application from configured services	Must	
5.1.5.5 Configuring the data connectors of an application	Must	
5.1.5.6 Creating an application from a template	Optional	Basic functionality is of higher importance, anyway important
5.1.6 Setting up operational configurations		
5.1.6.1 Deploying an (distributed) application	Must	
5.1.6.2 Updating an (distributed) application	Must	At least remove/re-install
5.1.6.3 Uninstalling an (distributed) application	Must	
5.1.6.4 Updating the ECS management system	Must	Updating the installed platform with a next iteration/sprint
5.1.6.5 Updating ECS runtime	Must	Updating the installed platform with a next iteration/sprint
5.1.6.6 Update firmware	Optional	Not in the primary scope, may be realised through a (device-specific) service.
5.1.6.7 Replace an ECS device	Can	Can be done via SSH in the extreme case.
5.1.6.8 Restore an ECS device	Can	Can be done via SSH in the extreme case.
5.1.6.9 Remote management of ECS devices (Further administrative actions)	Can	Can be done via SSH in the extreme case.
5.1.7 Additional activities for (distributed) applications		
5.1.7.1 Checking for application requirements vs. current ECS-characteristics	Must	
5.1.7.2 Simulating the integration of ECS devices for an application	Optional	Interesting and relevant, potentially out of resources.
5.1.7.3 Simulating the deployment or update of services and applications	Optional	Interesting and relevant, potentially out of resources.
5.1.7.4 Setting up the necessary resources for an application	Optional	If there is a way to have more resources in the project…
5.1.7.5 Integrating external container registries	Can	We start with a fixed set of basic containers
5.1.7.6 Changing the state of an application	Optional	From user perspective, “must” from platform perspective

5.1.7.7 Restoring an operational application	Optional	Can be achieved by other operations, in initial versions by application restart
5.1.7.8 Visualising the results of an application	Optional	No foreseen resources in contract, we will try anyway
5.2 Activities related to AI services and processes		
5.2.1 Activities within the Data Science tool chain related to data exploration		
5.2.1.1 Creating a data exploration processes	Optional	AI services must run, even without exploration support
5.2.1.2 Data pre-processing (refinement, normalising, addition of data, feature engineering)	Optional	AI services must run, even without exploration support
5.2.1.3 Data visualisation	Optional	See 5.1.7.8
5.2.1.4 Statistical analysis of data	Optional	AI services must run, even without exploration support
5.2.1.5 Provision of intermediate results for further exploration planning with customers	Optional	AI services must run, even without exploration support
5.2.2 Activities within the Data Science tool chain related to AI model training and evaluation		
5.2.2.1 Data augmentation	Optional	AI services must run, even without exploration support
5.2.2.2 Training of AI models	Must	
5.2.2.3 Testing of AI models	Must	
5.2.2.4 Evaluate the training process by scoring the process with the use of metrics	Optional	AI services must run, even without exploration support
5.2.2.5 Evaluate the training process against different AI model structures	Optional	AI services must run, even without exploration support
5.2.2.6 Evaluate the training process against different training structures	Optional	AI services must run, even without exploration support
5.2.2.7 Provision of intermediate AI model results for further planning of AI model development/refinement with customers	Optional	AI services must run, even without exploration support
5.2.2.8 Continuous application of the AI model on new data	Optional	AI services must run, even without exploration support
5.2.2.9 Tuning (re-calibration) of the AI model parameters	Optional	AI services must run, even without exploration support
5.2.2.10 Backup of labelled data sets (for further learning/training)	Optional	AI services must run, even without exploration support
5.2.3 Use of AI applications / AI services within the platform		
5.2.3.1 Update an AI service within an AI application	Must	At least remove/re-install
5.2.3.2 Replace an existing AI service	Optional	Correlates with adaptive platform functionality
5.2.4 Analysis and Prediction of the performance and accuracy of AI applications		

5.2.4.1 Provision of metrics for evaluating an application or service	Must	
5.2.4.2 Analysis of metadata to detect deviation of the service/application or AI model drift	Optional	AI services must run, even without exploration support
5.2.4.3 Monitoring the prediction quality of the service/application	Must	
5.2.4.4 Comparison of several AI models	Optional	AI services must run, even without exploration support
5.2.4.5 Evaluating the quality of services/applications	Optional	AI services must run, even without exploration support
5.2.5 Using AI services and AI applications manually, offline use		
5.2.5.1 Classification based on the use of existing data, stored, data	Can	AI services must run, even without exploration support
5.2.5.2 Accessing historical data (historian databases) for AI tasks	Can	AI services must run, even without exploration support
Entities of the IIP-Ecosphere platform		
Problem solution matching tool, broker	Out of scope	In more general form realised in terms of the “AI solution catalog” by the “AI accelerator” of IIP-Ecosphere.
Data Science tool chain	Out of scope	As suggested in partner discussions, it is more advisable and flexible, if the Data Science tool chain could use platform interfaces and services rather than being deeply integrated.
Data lake	Must	
Field device	Out of scope	IIP-Ecosphere, as a virtual platform, is extending to Edge devices but does not cover the level of field devices.
Edge device (ECS device)	Must	
Edge runtime (ECS runtime)	Must	
Server device	Must	
Server runtime	Must	
Cloud	Optional	Not mentioned in the IIP-Ecosphere grant agreement but of interest in the Industry 4.0 domain, e.g. GAIA-X
Cloud runtime	Optional	Not mentioned in the IIP-Ecosphere grant agreement but of interest in the Industry 4.0 domain, e.g. GAIA-X
Service	Must	
Service store	Must	
Service configuration tool	Must	
Service orchestration tool	Must	
Application	Must	
Edge/Server/Cloud (ECS)	Must	

Management system		
Device description store	Must	
Device configuration tool	Must	
Runtime application / service distribution tool	Must	

Table 1: Scoping of activities and entities.

It is important to understand that the scope discussed here is initial, i.e., it may change in the subsequent development (e.g., it might even become narrower). However, it is important to provide an initial classification, also avoid unrealistic expectations based on the collected activities. As already mentioned in Section 1.1, also entities/activities that we scope out here are important findings, as they may stipulate further development, e.g., in standardisation, in the upcoming IIP-Ecosphere community or in other platform development activities.

7 Summary

IIoT and Industry 4.0 platforms currently form the software foundation of complex manufacturing systems. Within the heart of the ongoing initiatives in the field of Industry 4.0 lies the integration of Artificial Intelligence into such systems, which will allow for new opportunities, but will also cause challenges for all involved disciplines. In this Whitepaper, we reported on our works to capture and establish a shared view on the IIP-Ecosphere platform, which will be developed as a core technical contribution of the IIP-Ecosphere Think Thank "Platforms", and to foster and complement the requirements collection of the platform, based on this shared view on envisioned platform functionality. Following the Industrial Internet Reference Architecture [IIRA], we established a first Usage View of the IIP-Ecosphere platform and performed intensive requirements collection for the upcoming platform, starting from an initial System under Consideration with 5 entities, 7 roles and 27 activities, this Usage View was further developed in terms of a series of workshops with all interested project partners. Based on discussions of partners with interested in the platform and the platform architecture, i.e., the Think Tank "Platform" and the work package "AI Accelerator", a joint vision of the IIP-Ecosphere platform was established and thoroughly discussed with all involved partners and stakeholders. The Workshop and discussions lead to the current initial description of the IIP-Ecosphere platform's Usage View as a System under Consideration, consisting of 18 Entities, 19 roles, as well as 43 edge-server-cloud activities and 24 AI-specific activities, as presented in this Whitepaper.

As our next steps, we plan to iteratively develop the IIP-Ecosphere platform and to face the research challenges on the way, i.e., we will refine our usage view requirements towards functional requirements, develop an architecture, realize the platform relying on as many existing Open Source components as feasible and perform experiments. Establishing the Usage View of the IIP-Ecosphere already lead to a significant increase in the number of described requirements for the platform as well as indicating new areas of requirements to be further elaborated.

Overall the established, shared, Usage View in this document provides an agreed basis for further deriving/validating the functional and quality requirements of the overall platform and, thus, enables the subsequent work on the development of the concepts and solutions established in the shared Usage View. A further goal of establishing a shared Usage View of the platform was to define the scope of the subsequent work (what can and what cannot/shall not be realized within the lifetime of IIP-Ecosphere) based on a shared understanding of the concepts within the platform for all partners. This goal was met in our work on the Usage View. Additional to the activities described in this Whitepaper, 26 further potential activities within the IIP-Ecosphere platform were identified and described, which are not currently listed in this Whitepaper, but will be worked on in our future work on the platform.

Another next step in the project together with the think tank "Business Models" will be the description of the IIP-Ecosphere platform from the Business Viewpoint. Therefore, the Usage View presented in this Whitepaper will act as foundation and input from a more technical side and will support the development of future business models for an AI-based Industrial Internet of Things (IIoT) platform.

While probably not all entities and activities will be part of the platform to be developed in IIP-Ecosphere, we believe that the requirements and concepts sketched in this usage view and the subsequent requirements will be a helpful foundation for future work on platforms within and outside IIP-Ecosphere.

The authors would like to thank all partners who participated and contributed to the IIP-Ecosphere Usage View workshops and also contributed to establishing the Usage View of the IIP-Ecosphere platform presented in this Whitepaper.

8 References

[AAS-D] Plattform Industrie 4.0, Die Verwaltungsschale im Detail, 2019, https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/verwaltungsschale-im-detail-präsentation.html

[GAIA-X] Federal Ministry for Economic Affairs and Energy (BMWi), GAIA-X: Technical Architecture, https://www.bmwi.de/Redaktion/EN/Publikationen/gaia-x-technical-architecture.pdf

[IDS-RAM] International Data Spaces Association, Reference Architecture Model, Version 3.0, April 2019, https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf

[IIRA] The Industrial Internet Reference Architecture Technical Report, https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf

[LNI-UV] LNI 4.0 Testbed Edge Configuration – Usage View, https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/LNI4.0-Testbed-Edge-Configuration_UsageViewEN.pdf

[LSM] F. van der Linden, K. Schmid, E. Rommes, Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering, Springer, 2007

[NIST-800-53] Joint Task Force, Security and Privacy Controls for Information Systems and Organizations, Draft NIST Special Publication 800-53, rev. 5, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5-draft.pdf

[OPC-UA] Website of the OPC Foundation, Specification of the OPC Unified Architecture https://opcfoundation.org/about/opc-technologies/opc-ua/

[RAMI] Reference Architecture Model Industrie 4.0, https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.

[SEA+] C. Sauer, H. Eichelberger, A. S. Ahmadian, A. Dewes, J. Jürjens, Aktuelle Industrie 4.0 Plattformen – Eine Übersicht, IIP-Ecosphere Whitepaper IIP-2020/001, 2020 (available until January via the platform survey, later also publicly available)

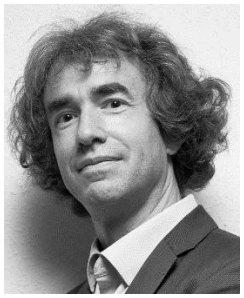
[ZVEI-AAS] ZVEI, Beispiele zur Verwaltungsschale der Industrie 4.0-Komponente – Basisteil, 2016, https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/November/Beispiele_zur_Verwaltungsschale_der_Industrie_4.0-Komponente_-_Basisteil/Beispiele-Verwaltungsschale-Industrie-40-Komponente-White-Paper-Final.pdf

[ZVEI I4] Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation, Federal Ministry for Economic Affairs and Energy (BMWi), https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2017/Juni/Industrie_4.0_Plug_and_produce/Industrie-4.0-_Plug-and-Produce-zvei.pdf

About the authors

Dr. Heiko Stichweh is head of the department Innovation at Lenze, which is an innovative supplier of drive systems and automation solution. His research interests focus on mechatronics, drive technologies, automation systems as well as digital twin and condition monitoring. He studied power systems engineering at the University of Clausthal, where he also received his PhD in 2004.

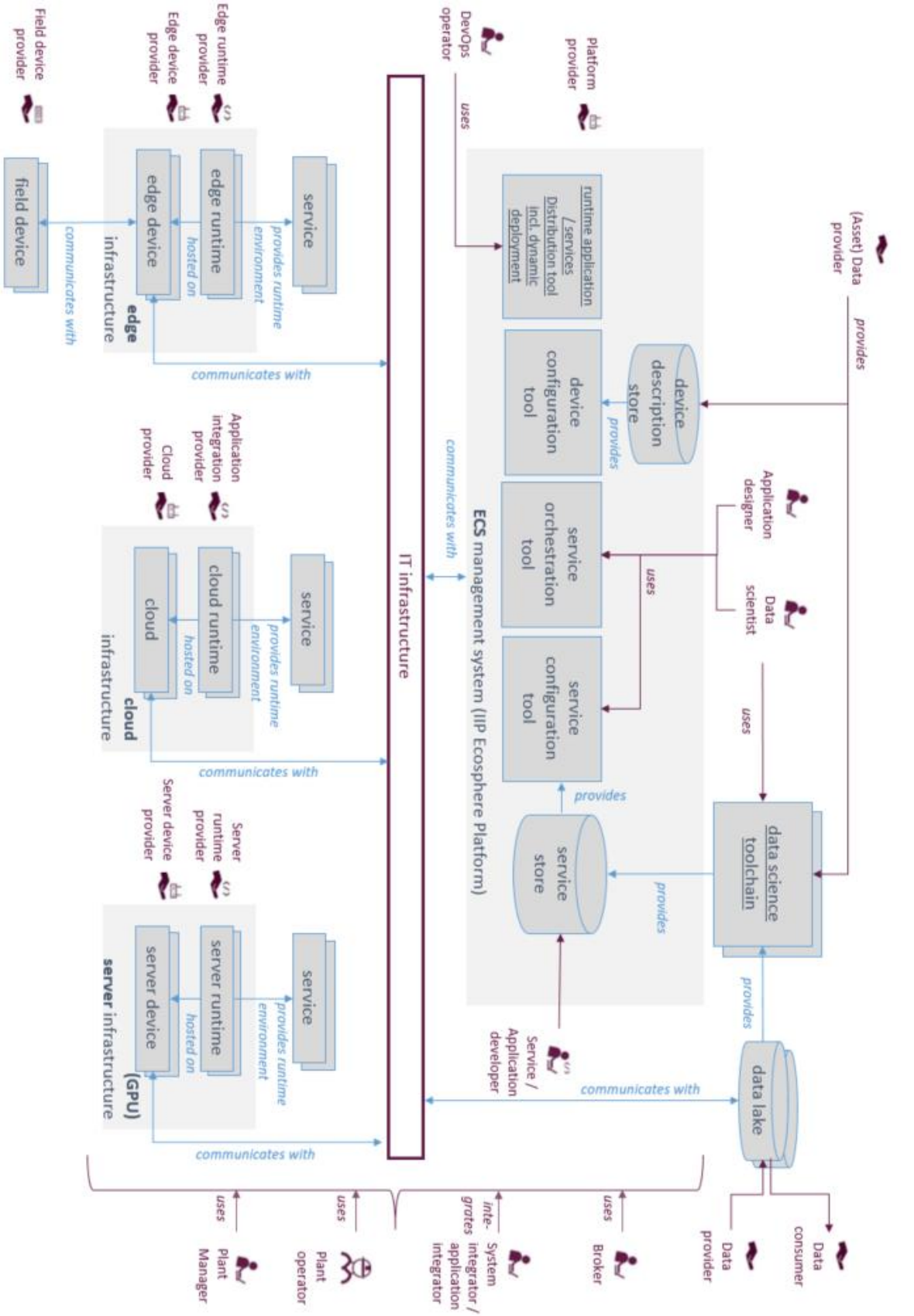
Dr. Christian Severin Sauer is a postdoctoral researcher in the Software Systems Engineering group at the Institute of Computer Science, University of Hildesheim. His research interests focus on domain knowledge elicitation and modeling for explanatory and context-sensitive AI applications. He studied at the University of Hildesheim and received his PhD in Computer Science from the University of West London. During his PhD, he investigated and developed methods for knowledge elicitation and knowledge modeling for explanatory and context-sensitive AI applications.



Dr. Holger Eichelberger is deputy head of the Software Systems Engineering group at the Institute of Computer Science at the University of Hildesheim. He conducts research in the areas of software product lines, model-based engineering, performance monitoring, and performance analysis. In particular, he is interested in the integration of these areas to create adaptive software systems. In IIP-Ecosphere he leads the think tank "Platforms" as well as the AI Accelerator. He studied computer science at the University of Würzburg, where he received his PhD on the automatic layout of UML diagrams.

Photographer: Daniel Kunzfeld

Appendix 1


(Asset) Data provider
provides
Platform provider
DevOps operator
uses
runtime application / services Distribution tool incl. dynamic deployment
device configuration tool
device description store
provides
ECS management system (IIP Ecosphere Platform)
communicates with
IT infrastructure
Application designer
Data scientist
uses
service orchestration tool
service configuration tool
provides
service store
data science toolchain
provides
data lake
Service / Application developer
communicates with
Field device provider
Edge device provider
Edge runtime provider
field device
edge device
edge runtime
service
hosted on
provides runtime environment
communicates with
edge infrastructure
Cloud provider
Application integration provider
cloud
cloud runtime
service
hosted on
provides runtime environment
communicates with
cloud infrastructure
Server device provider
Server runtime provider
server device
server runtime
service
hosted on
provides runtime environment
communicates with
server infrastructure (GPU)
Data provider
Data consumer
Broker
System integrator / application integrator
integrates
Plant operator
Plant Manager
uses