

## LNI 4.0 Testbed Edge Configuration – Functional View

<b>LNI 4.0 Testbed Edge Configuration – Functional View</b>	<b>1</b>
<b>1 Overall Objectives of Testbed Edge Configuration</b>	<b>2</b>
<b>2 Purpose of this Document</b>	<b>3</b>
2.1 Target audience of this document	3
<b>3 The Framework for Functional Considerations in Edge Configuration</b>	<b>4</b>
3.1 Structure of the document	4
3.2 Greenfield and Brownfield Scenarios	4
3.2.1 Greenfield	4
3.2.2 Brownfield	4
3.3 Functional requirements derived from Usage View	4
3.3.1 Requirements for Edge Configuration System (Architecture)	5
3.3.2 Requirements for adding entities use cases	6
3.3.3 Requirements for removing entities use cases	6
3.3.4 Requirements for setting operational configurations use cases	6
3.3.5 Provision of entities	7
<b>4 Functional Workflows</b>	<b>8</b>
4.1 SystemIntegrator - Discovery and Registration	8
4.2 SystemIntegrator - App Deployment and Activation	10
4.3 PlantOperator - Maintenance Actions	11
4.4 EdgeProvider and AppDeveloper - App and Edge Runtime development and provisioning to the App Store	13
<b>5 Functions in Edge Configuration</b>	<b>14</b>
5.1 Detection & Discovery Functions	14
5.1.1 Entity Detection Functions	14
5.1.1.1 Field Device Detection	14
5.1.1.2 Edge Device Detection	15
5.1.2 Parameter Discovery Functions	15
5.1.2.1 Discovery of Field Device Parameters	16
5.2 Communication Functions	16
5.2.1 Device to Edge Communication Functions	17
5.2.1.1 Setting of Parameters for a Field Device by an Edge Device	17
5.2.1.2 Propagation of Data Endpoints	17
5.2.2 Protocol and Format Transformation Functions	18
5.2.3 Edge to IT Communication Functions	18
5.2.3.1 Propagation of a Detected Field Device	18
5.2.3.2 Propagation of Field Device Parameter Set	18
5.2.3.3 Forwarding of Field Device Target Parameter Values	19
This function serves as a forwarding of information from the edge management system to the edge device for setting values in the field device in another step.	19
5.2.3.4 Deploying an Edge Runtime	19
5.2.4 Edge Management System to Application Store Communication Functions	19
5.2.4.1 Propagation of an provisioned Application	19
5.2.4.2 Propagation of an updated Application	20
5.2.4.3 Propagation of a discontinued Application	20
5.2.4.4 Obtaining an application from the application store (for deployment)	20
5.2.4.5 Obtaining an Update of an application	20
5.3 Rights Management Functions	21
5.3.1.1 Making an Edge Runtime available	21
5.4 Configuration Functions	21
5.4.1 Selection Functions	21
5.4.1.1 Selection of an Edge Runtime	21
5.4.2 Parameterization Functions	21
5.4.2.1 Setting Field Device Parameters	21

5.4.3	Deployment Functions.....	2220
5.4.3.1	Deployment of an Application .....	2224
5.5	Provisioning Functions.....	2224
5.5.1	Runtime Provisioning Functions.....	2224
5.5.1.1	Provision of an edge runtime.....	2224
5.5.2	Application Provisioning Functions.....	2224
5.5.2.1	Provision of an application.....	2224
5.5.3	Application Runtime Functions.....	2324
5.6	Monitoring Functions.....	2322
5.6.1	Asset Management Functions.....	2322
5.6.2	Integrity Monitoring Functions.....	2322
5.7	Non-Functional Requirements .....	2322
5.7.1	Precondition Checks.....	2322
5.7.2	Acknowledgements .....	2322
6	Authors and Contributors .....	2422
6.1	Authors.....	2422
6.2	Contributors .....	2423

#### Contact:

Labs Network Industrie 4.0 e.V.

Ernst-Reuter-Platz 7

10587 Berlin, Germany

Phone: +49 (30) 36702177

E-Mail: [info@lni40.de](mailto:info@lni40.de)

## 1 Overall Objectives of Testbed Edge Configuration

The testbed Edge Configuration was established to prepare the standardization in the context of the manufacturing industry with respect to the emerging edge computing technology. The testbed does not address edge computing technology in itself but focuses on *edge configuration*. For this purpose, concepts will be developed, practically implemented and validated. The results and experiences will be made available to the standardization activities to feed them into the further or new development of standards.

From an architectural point of view, the testbed Edge Configuration is based on a layered architecture as shown in Figure 1:

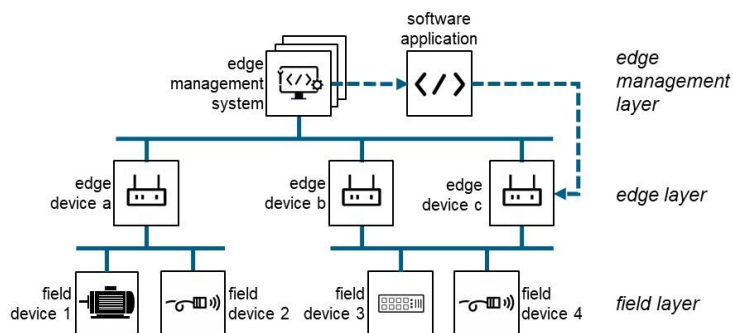


Figure 1. Layered architecture of the proposed testbed Edge Configuration

- The focus of the testbed Edge Configuration is on the configuration of the interaction between the edge and edge management layer. Currently there does not exist a suitable standard for this focus

and the testbed will develop proposals for this aspect in the form of functional primitives including parameter sets (functional view), which afterwards must be implemented (implementation view).

- In the interaction between field and edge layer, although today no standard has prevailed in practice, we currently see OPC-UA as a promising candidate to design this interaction in the future, both technologically and via companion specifications. For this reason, this interaction is *not* the focus of the testbed Edge Configuration (functional view).

## 2 Purpose of this Document

The purpose of this document is to describe the functional context of the testbed Edge Configuration. Thus, this document defines the contextual framework of all activities in the testbed and possibly derived standardization activities. In accordance with the Industrial Internet Reference Architecture [1] this document describes a so-called *functional view*, see Figure 2. We assume that this functional view remains stable throughout the life of the testbed Edge Configuration.

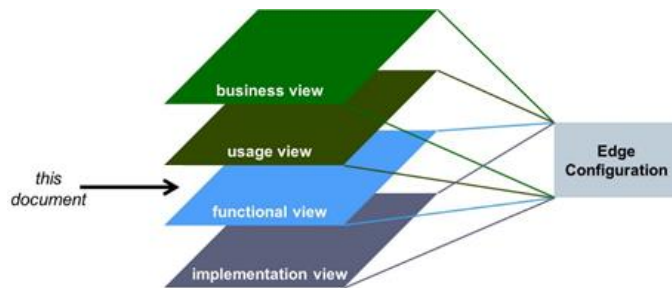


Figure 2. Classification of the document following the Industrial Internet Reference Architecture, see [1]

Regardless of this “stable” application context (see *usage view* document), various concepts (in the terminology in Figure 2 called a functional view) and solution approaches (in the terminology in Figure 2 called an implementation view) will be developed and discussed. The development of these concepts and solutions will be the “real” work in the testbed Edge Configuration. These concepts and solutions shall encompass options for solving specific requirements provided by the testbed partners. Therefore, the purpose of this document is helping to structure and orchestrate the various work efforts of the different partners based on a common view.

In addition, there is a description of a *business view*, which describes the business concerns of various companies with respect to the topic “edge configuration” [2]. This frames an independent perspective but has well-defined relationships to the second document which is the *usage view* with respect to the topic “edge configuration”. These relationships are described in the annex of [2].

### 2.1 Target audience of this document

This document serves as a bridge between the usage view and the implementation view on Edge Configuration. While no details on the technical implementation of Edge Configuration are given, the considerations in this document go beyond use case classification by describing functional interactions between the different actors in an Edge Configuration scenario in detail. The goal of this document is to provide a guideline for the detailed planning of an edge configuration system on an architectural level, before actual implementation can be considered.

Since the focus of this document is to provide planning guidelines for an edge configuration system, its main target audience is system architects trying to specify a concrete edge configuration system. System developers might find this document useful to understand the context of requirements set for the implementation of an edge configuration system. However, the document „Implementation View on Edge Configuration” focuses on the implementation of the edge configuration based this document and therefore addresses system developer’s needs.

## 3 The Framework for Functional Considerations in Edge Configuration

**Kommentiert [AGG1]:** Glossary from Usage View to be repeated here

### 3.1 Structure of the document

Chapter 3 analyses the use cases given in the Usage View document, also considering the business context described in the Business View, and derives functional requirements from them. Chapter 4 takes a look at typical interactions for the functioning of a system under consideration, from which necessary functionality is derived. Chapter 5 describes the functions identified to be necessary in greater detail using UML sequence diagrams.

### 3.2 Greenfield and Brownfield Scenarios

**Kommentiert [AGG2]:** Beispielszenarien mit Geräten/MES/Bussystemen als Illustrationen

When defining the functions of an Edge Management System and its systemic environment, many functions hinge on the capabilities of the systems involved. The Testbed Edge Configuration is striving for a clear-cut distinction for the described functions between those functions that are optional and "nice to have" and those that are considered mandatory for a fully functioning Edge Configuration. However, demanding mandatory functions from legacy systems can often lead to inconsistencies, as some legacy solutions may simply not be capable of fulfilling certain requirements for the desired functions. This section, therefore, establishes a definition for both greenfield and brownfield as scenarios for implementation and considers differences in the requirements postulated for each scenario.

#### 3.2.1 Greenfield

Greenfield scenarios are those scenarios where a production system is designed from scratch at the time that Edge Configuration is considered. The system might substitute for a decommissioned legacy system, but the whole engineering of the system must be independent of any pre-installed base to be considered greenfield.

Greenfield installations of edge solutions provide inherent breadth and depth of new technologies that can be applied. The lack of legacy solutions allows for a rigorous application of state of the art products. Hence, for a greenfield installation, a maximum of supported edge configuration functions can be demanded.

#### 3.2.2 Brownfield

Brownfield scenarios are those scenarios where a pre-existing solution is retrofitted to become IIoT capable. Often, brownfield scenarios will call for the installation of new hardware and/or software to establish connectivity. However, its distinguishing characteristic is that a great level of respect has to be paid to pre-existing definitions and solutions, be they installed or even only planned.

Brownfield installations are characterised by a large installed base that has to be taken into account for engineering edge solutions. It should be noted that brownfield IIoT installations are the prime area of application for edge gateway solutions. By their very nature, edge gateways provide a good way for retrofitting a brownfield installation.

### 3.3 Functional requirements derived from Usage View

This document ties into the working documents of the LNI4.0 Testbed Edge Configuration. The Usage View document of the Testbed Edge Configuration [2] provides a comprehensive description of the use cases considered inside the testbed. Naturally, these use cases have implications for the functions that an edge configuration solution has to provide. This chapter analyses the categories given for use cases in the Usage View document with regards to the functional requirements that can be derived from them.

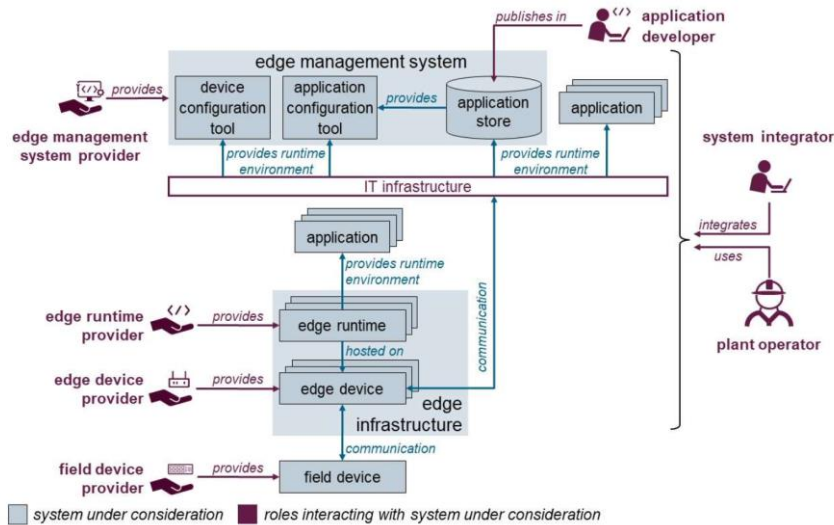


Figure 3

Figure 3: Overview of "system under consideration" and considered "roles" [2]

### 3.3.1 Requirements for Edge Configuration System (Architecture)

As described in Figure 3 and in the Usage View document [2], the testbed Edge Configuration considers the following entities with their corresponding functions:

- Edge management system as the runtime environment for device configuration and application configuration tools as well as an access tool for an application store
- Edge runtime as a part of the edge infrastructure which provides a runtime environment for applications
- IT infrastructure as a runtime environment for applications and for the edge management system
- Edge device as the host for edge runtimes
- Field device as the interface to the physical world
- Applications as objects of edge configuration

The IT infrastructure and the edge runtime both provide environments to run applications. While due to the disparity in resources the implementation of both may differ significantly, for the Functional View taken in this document, they can basically be considered equivalent.

The possible relationship between edge runtimes, edge devices and applications is illustrated in Figure 4.

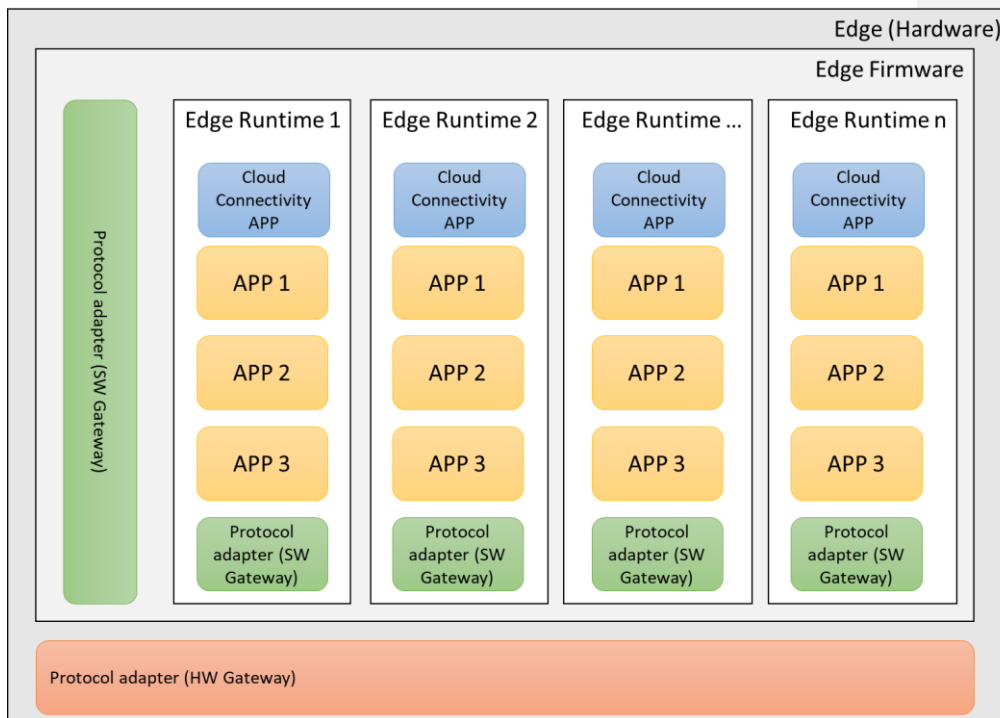


Figure 4 Overview of possible software/hardware relationships inside an edge device

### 3.3.2 Requirements for adding entities use cases

The description for adding a field device calls for detection of the asset in question (5.1) and the need to provide parameters. Furthermore, communication between edge and field device is described (5.2). The parameter setting is tied to specific roles (5.3).

The addition of an edge device calls for the detection of the device by an edge management system (5.1).

Adding an edge runtime calls for a selection process (5.4).

Adding an edge management system again calls for holistic detection functions (5.1).

### 3.3.3 Requirements for removing entities use cases

Removing an edge management system calls for extensive communication (5.2).

The act of removing an edge runtime encompasses a configuration function (5.4).

Removing an edge device demands communication (5.2) as well as configuration (5.4).

Finally, removing a field device again asks for communication (5.2) as well as configuration (5.4) functions. Advanced systems need monitoring functions (5.6) to operate autonomously.

### 3.3.4 Requirements for setting operational configurations use cases

Deploying an application encompasses both configuration (5.4) and runtime provisioning (5.5) functions.

## Unrestricted

Updating a deployed application again requires configuration (5.4) and runtime provisioning (5.5) functions.

So, does uninstalling a deployed application.

Configuration of data provisioning and consumption, however, is purely a configuration (5.4) task.

Updating an edge management system is also a configuration (5.4) task.

Updating an edge runtime can, if optional tasks are encompassed, pose the need for runtime provisioning functions (5.5), but in any case poses a configuration (5.4) function.

Update firmware requires first of all communication (5.2), then configuration (5.4) and might have demand for runtime provisioning functions (5.5).

### 3.3.5 Provision of entities

Provision of a field device has no direct functional impact on edge configuration but can benefit from configuration functions (5.4) as well as monitoring functions (5.6).

Provision of an edge device is functionally similar enough to the provision of a field device that the functional requirements on this level are the same.

Provision of an edge runtime has direct need for runtime provisioning functions (5.5).

Update of an edge runtime likewise needs runtime provisioning functions (5.5).

For the discontinuation of an edge runtime, runtime provisioning functions (5.5) need to be supplemented with communication functions (5.2) for notification transport.

The provision of an edge management system again is mostly not affected by edge configuration functions, but benefits from configuration (5.4) and monitoring (5.6) functions.

The update of an edge management system can be considered similar to the provision of an edge management system with regards to its functional requirements.

Discontinuation of an edge management system calls for communication functions (5.2) in order to forward notifications.

Provision of an application to the application store calls for specific functions in the field of runtime provisioning functions (5.5).

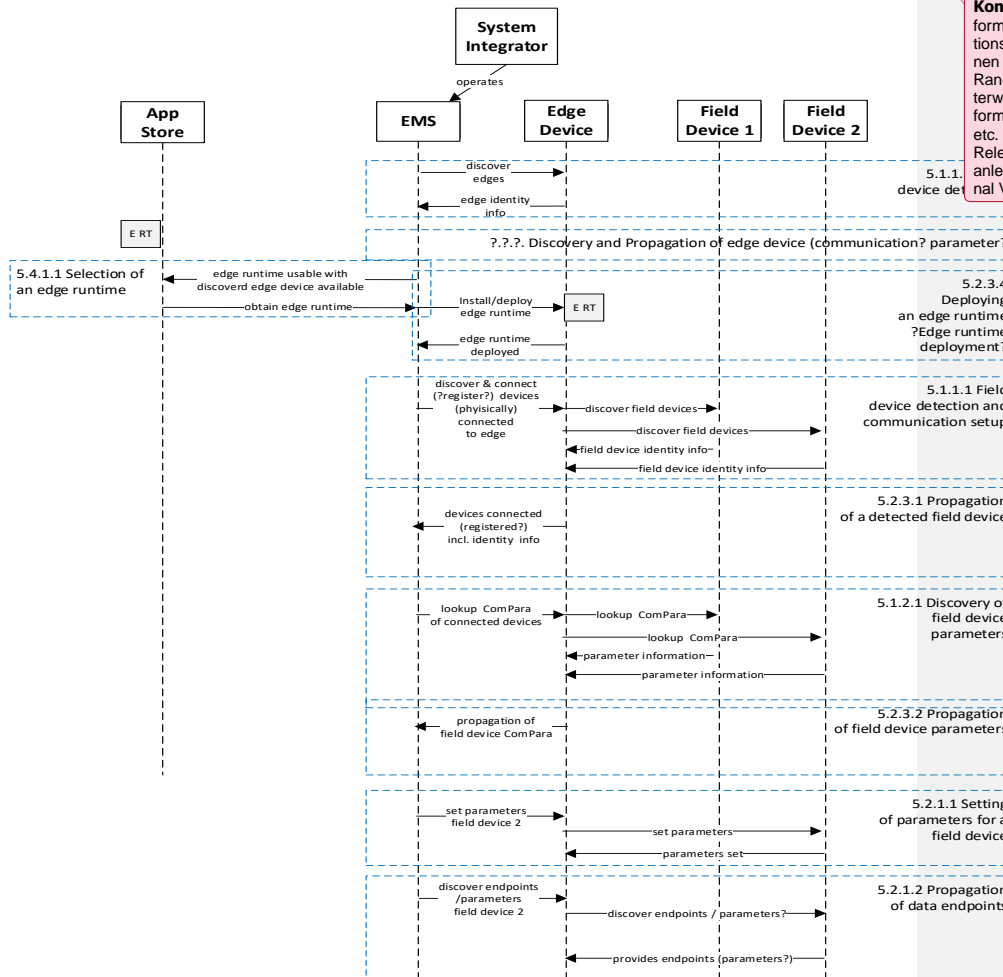
The update of an application has functional requirements similar to the provision of an application.

Discontinuation of an application is another case where communication (5.2) is needed.

The provision of an update of firmware finally benefits from configuration functions (5.4).

## 4 Functional Workflows

### 4.1 SystemIntegrator - Discovery and Registration



**Kommentiert [AGG3]:** Extend the user stories to more practical descriptions?  
Selection processes are not described in detail by us right now. There is demand to add this perspective.

**Kommentiert [AGG4]:** Idee Ulrich: Kapitel zum Thema Informationsmodell des Functional View, Dies kann die Informationsinfrastruktur katalysieren und das Handling der Informationen abstrahieren.  
Randbedingungen aus dem Usage View müssen konsequenterweise aufgefangen und erläutert werden, hierzu sind die Informationsmodelle wichtig. Recycling von ZVEI SW Submodel etc.  
Relevanz für die Standardisierung aus dem Testbett? Orthogonale Diskussion was wollen wir standardisieren sobald Functional View in dieser Version vorliegt.

This scenario illustrates a workflow considered typical for a system integrator tasked with setting up or adding to an edge computing scenario. More information about the context for such scenarios can be found in the documents on the Business View and the Usage View of Edge Configuration published by the Labs Network Industrie 4.0. In this scenario, the system integrator, being an entity outside the computing infrastructure, interacts with the system as a whole through the Edge Management System.

In a first step, in order to gain access to shop floor systems, the Edge Management System has to be made aware of any connected Edge Devices and how to reach them. This defines a functionality, the Edge device detection (5.1.1.2) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.



???

In order to set up application functionality on the available edge devices, the system integrator has to decide on an edge runtime. In this scenario, the system integrator has to select an edge runtime from the App Store of their choice. This defines a functionality, the Selection of an edge runtime (5.4.1.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

A selected edge runtime has to be deployed on those edge devices deemed fit for it by the system integrator. Apart from technical parameters, the application scenarios for individual edge devices will may influence the scope of deployment. On the other hand, for big installations, bulk deployment is likely a desirable possibility. The need for deployment defines a functionality, the Deploying an edge runtime (5.2.3.4) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

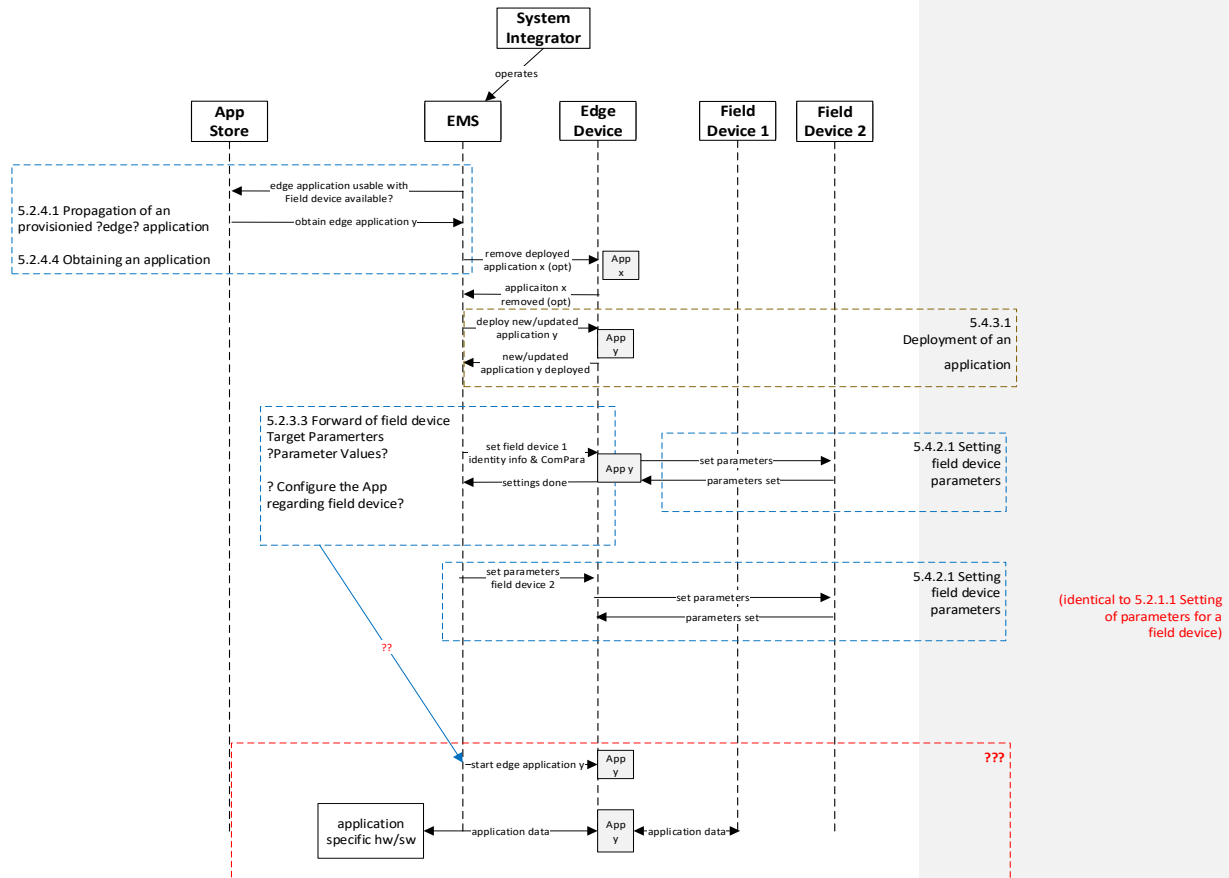
Once the Edge Devices are all set up, the system integrator can focus their efforts on handling the Field Devices. As with Edge Devices, first of all some kind of detection for connected Field Devices has to happen. This defines a functionality, the Field device detection (5.1.1.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

The information about detected Field Devices is first collected by the Edge Devices. In order to aggregate and consolidate the gathered information, it has to be passed on to the Edge Management System. This defines a functionality, the Propagation of a detected field device (5.2.3.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

The Edge Management System shall be the final arbitrator for communication parameters to be used by the devices in a managed shop floor. In order to allow the management of communication parameters, the Edge Management System has to collect information on the available and currently configured communication parameters. This defines a functionality, the Discovery of field device communication parameter (5.1.2.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

As the Edge Devices collect communication parameters from the Field Devices, these have to be collected and consolidated at the Edge Management System. This defines a functionality, the Propagation of field device parameter set (5.2.3.2) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

## 4.2 SystemIntegrator - App Deployment and Activation



This scenario elaborates a typical workflow for the deployment and subsequent activation of a distinguishable software application in the form of an app. Interactions are initiated from outside the system under consideration by a system integrator, who interacts through the Edge Management System.

Before the system integrator can become active, information about available applications has to be made available to the Edge Management System from the side of the App Store. This defines a functionality, the Propagation of a provisioned application (5.2.4.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden, Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

Based on the knowledge of available applications, the system integrator can go about obtaining the software they desire. This defines a functionality, the Obtaining an application (5.2.4.4) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden, Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

The newly acquired application has to be brought to an appropriate Edge Device in order to deploy it for execution. This defines a functionality, the Deployment of an application (5.4.3.1) function. Details on

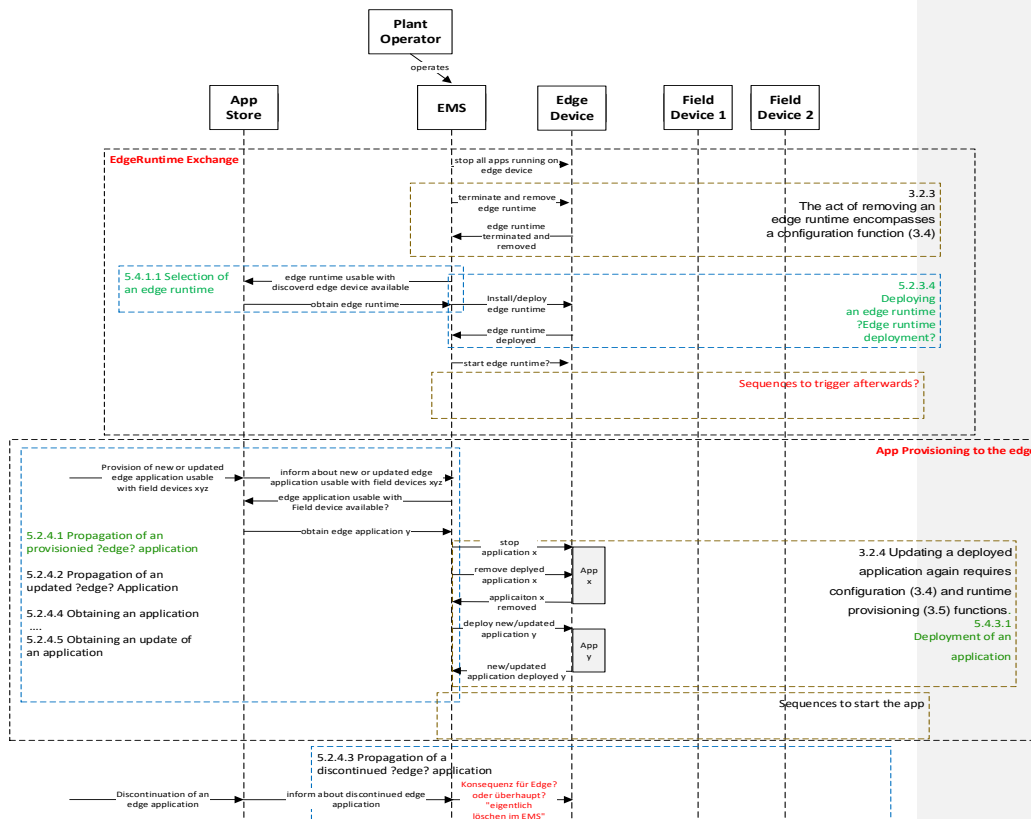
this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

For the purpose of this scenario, it is assumed that the application supports the system integrator in parameterizing a specific field device optimally. To ensure that optimizations defined by the manufacturer are actually utilized by the field device, an altered set of field device parameters has to be sent to the Edge Device for processing.

In the context of a new application, it is assumed that the system integrator has to change the field device parameters in order to ensure full functionality of the application, e.g. to acquire new data points for processing in the application. To this end, a change to the communication parameters for a field device has to be triggered. This defines a functionality, the Change field device parameters (5.4.2.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

???

### 4.3 PlantOperator - Maintenance Actions



This scenario considers typical maintenance actions to be performed by a plant operator, as far as they concern the system under consideration. The interactions between the operator and the system under consideration are again funneled through the Edge Management System. Activities covered in this scenario include the exchange of the edge runtime in utilization on an edge device and the addition of applications to the new runtime.

???

Once the old edge runtime is terminated, a new edge runtime has to be selected from the App Store. In this process, considerations for runtime compatibilities have to be taken into account. This defines a functionality, the Selection of an edge runtime (5.4.1.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

Next, the selected edge runtime has to be deployed on the edge device. The Edge Management System is considered to be the source of installation. The overall process defines a functionality, the Deploying of an edge runtime (5.2.3.4) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

???

With the exchange of the edge runtime concluded, the maintaining engineer can approach the task of redeploying applications to the system. There are two possible ways in which an application can be deployed to the system.

For one, a newly acquired application can be propagated to the Edge Management System. This defines a functionality, the Propagation of a provisioned application (5.2.4.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

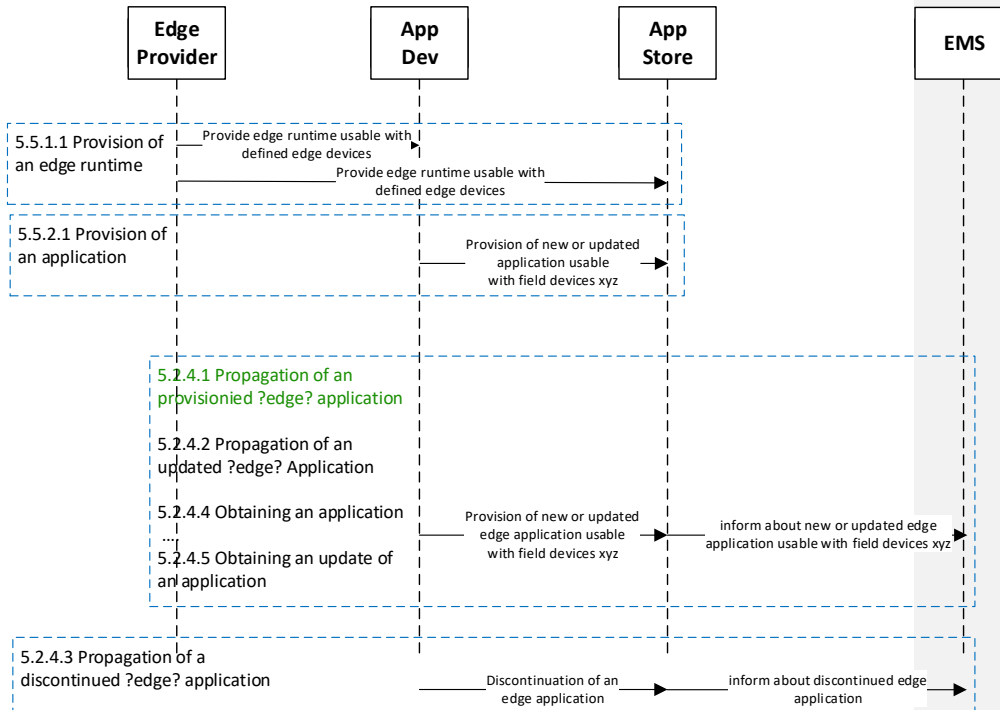
Also, an application can be loaded in the current version through an update channel. This defines a functionality, the Propagation of an updated application (5.2.4.2) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

For both previously discussed cases of application propagation from the App Store to the Edge Management System, the application has to be deployed on the edge device. For a newly acquired application, this defines the functionality Obtaining an application (5.2.4.4). Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

For the case of an update to be propagated to the edge device, the functionality defined is Obtaining an update of an application (5.2.4.5). Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

In order to maintain a proper listing of applications on the edge level, discontinued applications have to be removed from the pool of applications in the Edge Management System. This defines the function Propagation a discontinued application (5.2.4.3). Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden..** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

#### 4.4 EdgeProvider and AppDeveloper - App and Edge Runtime development and provisioning to the App Store



This scenario considers the necessary actions for developing software products such as edge runtimes and applications on the context of the system under consideration, namely the process of bringing developed software into the application store.

First of all, a newly developed edge runtime must be made available in the application store. This defines a functionality, the Provision of an edge runtime (5.5.1.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

As a next step, an application has to be made available in the application store as well. This defines a functionality, the Provision of an application (5.5.2.1) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

From the side of the developer, the next step would only be the discontinuation of an application. This defines a functionality, the Propagation of a discontinued Application (5.2.4.3) function. Details on this functionality can be found in the relevant section of chapter **Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden.** Details on the implementation of this functionality can be found in the Implementation View document for Edge Configuration published by the Labs Network Industrie 4.0.

## 5 Functions in Edge Configuration

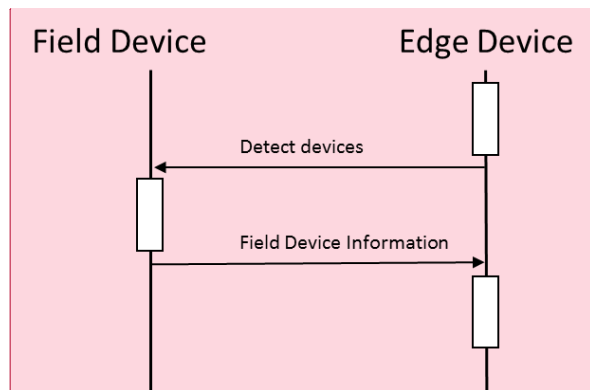
### 5.1 Detection & Discovery Functions

The area of detection and discovery functions is concerned with functionality that provides information about new or changed entities to any part of the system under consideration. The information provided can then be used by other functions to fulfill use cases given in the Usage View document. The process for discovery and detection is expected to be either automatic or semi-automatic as to ensure ease of use. If the addition of information is solely in the hands of a system integrator role, without support from a part of the system under consideration, it leaves the focus area of this document (even though this case will certainly happen in brownfield installations).

#### 5.1.1 Entity Detection Functions

Entity Detection functions deal with the identification of hitherto unknown or unacknowledged entities inside the system under consideration. Information gained from these detection functions is critical for any form of automated onboarding process. The system under consideration will have to provide an identification mechanism to make detected entities addressable inside the system.

##### 5.1.1.1 Field Device Detection



This function specifically deals with the detection of newly installed or reconnected field devices ~~or field devices~~ in a network where an edge device is newly connected. After the edge device is connected, it publishes its presence in the network and collects the responses from neighbouring devices in its internal registry database. ~~In the given overall architecture, this detection function has to be performed by an edge device.~~

Object: field device

Subject: edge device

Context: A field device is either newly connected or reconnected to an automation network containing at least one edge device OR an edge device is added to an automation network. (No functional preconditions.)

Result: The field device is identified for further handling in corresponding Adding Entities use cases.

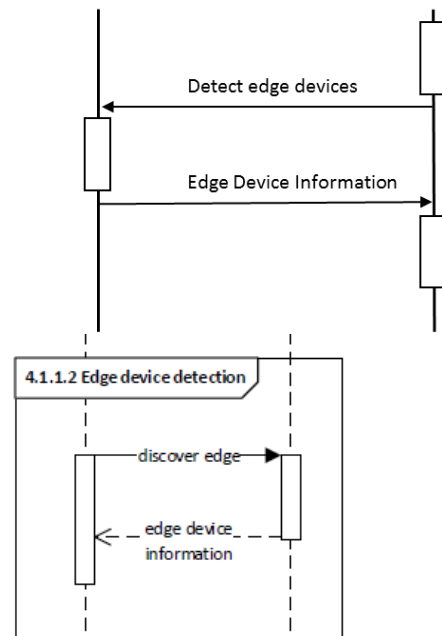
**Kommentiert [AGG5]:** User Story Generation: Wittenstein Beispiel heranziehen und zweites Beispiel von kuka jeweils komplett runterdeklinieren/aufschreiben. Daraus können wir (hoffentlich) Schlüsse ziehen. Ulrich ist interessiert die User Stories gegenzulesen. Ebenso Herr Heinzer von Schmersal.

**Kommentiert [AGG6]:** Feature of the field device to bootstrap/self-register to the registry to be added.

**Kommentiert [AGG7]:** IBM: Onboard ist nur Adressdatenverteilung, Aushandeln der Fähigkeiten findet separat statt. Onboard im Netz ist auch spannend! (DHCP oder nicht? Etc.) Broker/Bootstrap Service in der Mitte dazwischenschalten?

### 5.1.1.2 [Edge Device Detection](#)

#### Edge Device      Edge Management System



This function makes sure that an edge management is informed about any newly connected edge devices. After the edge device is connected, it publishes its presence in the network and the EMS database can register the presence of the newly connected edge device in its edge device registry database.

Object: edge device

Subject: edge management system

Context: When a new edge device is connected to an edge management system, it has to be detected in order for the edge management system to manage it. (No functional preconditions.)

Result: The edge management system has knowledge of the edge device and is able to manage it according to other functions.

### 5.1.2 Parameter Discovery Functions

The Discovery of Parameters encompasses those functions that deal with the retrieval of available and currently set parameters for the entities in the system under consideration. These functions are supportive of the Configuration functions discussed below insofar as they provide starting information for any kind of configuration process that would edit the parameters of an entity.

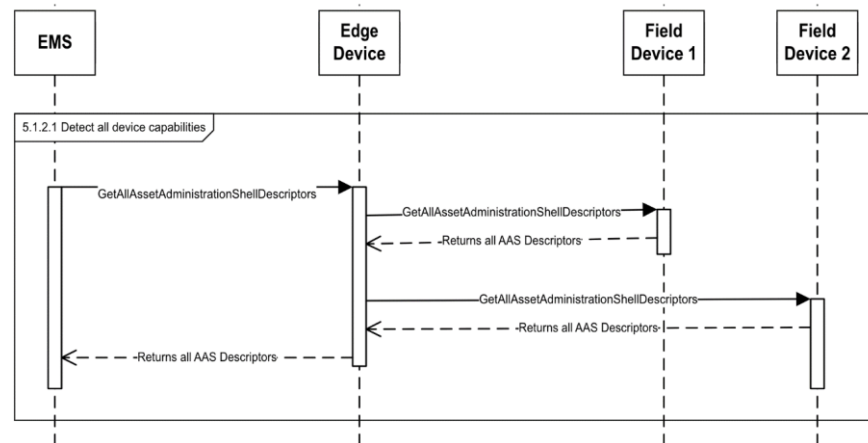
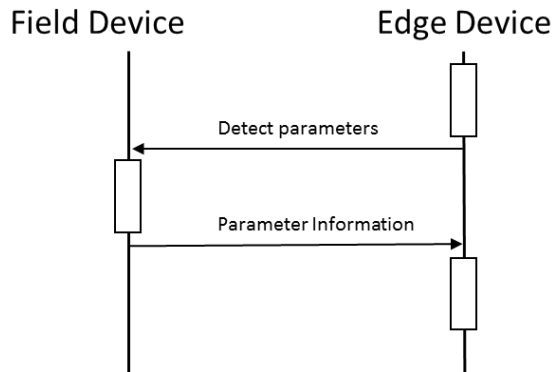
**Kommentiert [AGG8]:** Feature of the edge device to bootstrap/self-register to the registry to be added.

**Kommentiert [AGG9]:** Optionally, this may work exactly the other way around. How can we catch ambiguities like this?

**Kommentiert [AGG10]:** Implementation will be an internal registry for edge devices that has to be filled through either a poll by the EMS or the bootstrap self-registering of the edge devices.

Unrestricted

### 5.1.2.1 Discovery of Field Device Parameters



This function recovers a set of parameters that the field device needs for performing its expected duties in the automation process.

Object: Field Device

Subject: Edge Device

Context: Especially when a field device is being newly commissioned (see 5.1.1.1) to the overall automation system, the parameters it needs have to be retrieved in order to allow system integrators to set them (see 5.2.1.1).

Result: A set of parameters for a field device is available at the edge device.

## 5.2 Communication Functions

Communication functions allow the transfer of information between different entities in the system under consideration. Characteristically, while the transferred data may change, the information is typically unchanged by a communication function. Manipulation of information will usually be the domain of other functional categories.

**Kommentiert [AGG11]:** Only relevant for brownfield. In greenfield, field devices will always be configured through applications, as the basic connectivity is handled by bootstrap mechanisms.

"5.1.2.1 Detect communication capabilities"

Better:

"5.1.2.1 Detect all device capabilities"

Sequence diagram:

IDEA to be used for modification of this function: Demand that there is a URL provided for device interaction for brownfield configuration interfaces (typically small webservers these days, worst case online PDF with instructions for pin configurations etc.).

"Devices should support an AAS interface behind a uniformly defined URL as per Plattform I4.0 definitions. (AAS in detail V2 Part 2)"

Call to GetAllAssetAdministrationShellDescriptors  
Returns all AAS Descriptors

Unrestricted

Labs Network Industrie 4.0 e.V.

Ernst-Reuter Platz 7

16

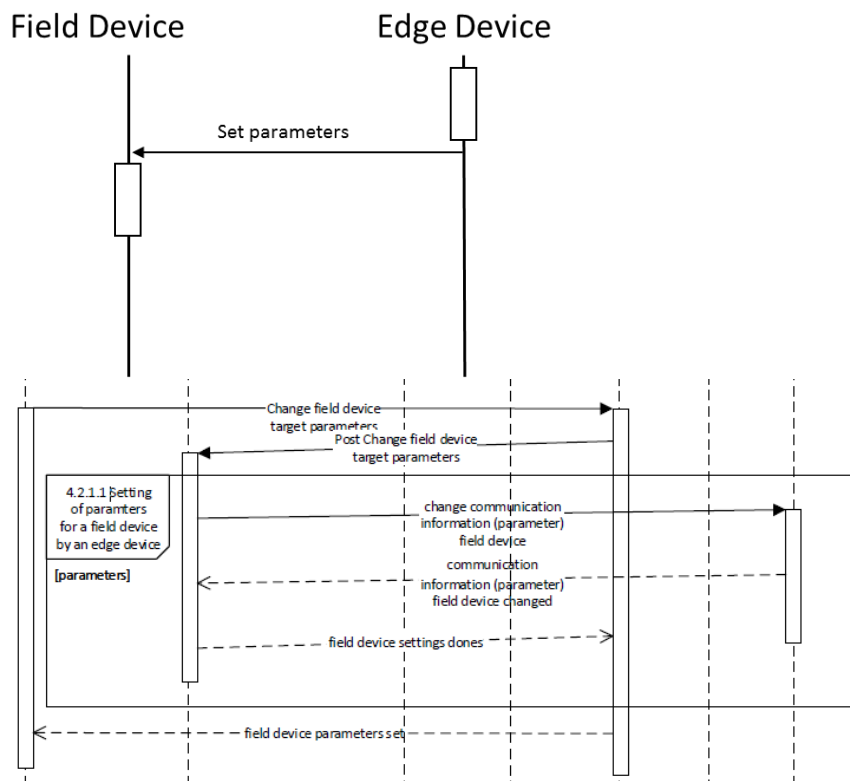
10587 Berlin



## 5.2.1 Device to Edge Communication Functions

Device to Edge Communication encompasses those data transmission functionalities that happen between field devices and edge devices. This type of communication might encompass hard real-time traffic, but it might also encompass acyclic communication. Characteristically, data endpoints provide the basis for communication relationships covered by these functions.

### 5.2.1.1 Setting of Parameters for a Field Device by an Edge Device



This function enables edge devices to set parameters on the connected field devices, typically having received these from an edge management system.

Object: Field Device

Subject: Edge Device

Context: If an edge device receives parameters to be set on a field device, it has to propagate these parameter values to the device in question and make sure that they are incorporated (5.2.3.2).

Result: The field device has changed its parameters to the transported values.

### 5.2.1.2 Propagation of Data Endpoints

This function allows a field device to propagate its contained data endpoints to a connected edge device for further processing.

Object: Edge Device

Subject: Field Device

**Kommentiert [AGG12]:** Might this be actually done by an application rather than an edge device? Is this still necessary for brownfield scenario support? For cases where the AAS request from 5.1.2.1 is not available e.g.? This would mandate an alternate sequence for 5.1.2.1 which would encompass this function in a more complex engineering process. Andreas to make a proposal how to handle the brownfield scenario for this. See sequence diagram by Thiago.

**Kommentiert [AGG13]:** Data endpoints would be part of the capabilities we handle in 5.1.2.1. Connectors have to map brownfield endpoints to capabilities as per 5.1.2.1. Other endpoints will already be capabilities. Separate propagation would then be redundant and can be removed. Description elsewhere that capabilities replace data endpoints for our functional considerations?!

## Unrestricted

Context: Data endpoints are not defined in edge devices in an explicit manner. However, edge devices have to obtain these data endpoint so that a communication connection is established.

Result: The edge device receives the data to be exchanged.

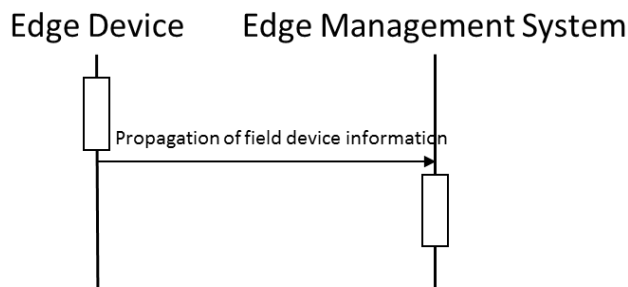
### 5.2.2 Protocol and Format Transformation Functions

The Transformation of Protocols and Formats can become necessary when communicating between different entities in a system under consideration. Transformation functions typically do not alter the information being transferred, but often alter the actual data transmitted.

### 5.2.3 Edge to IT Communication Functions

Edge to IT Communication handles all those communication relationships that exist between entities on the edge layer and entities on the Edge management layer. This kind of communication will usually preclude hard real-time traffic.

#### 5.2.3.1 [Propagation of a Detected Field Device](#)



This function propagates information gathered by an edge device about a connected field device towards the edge management system.

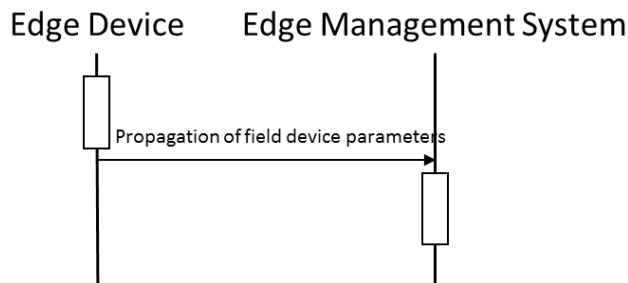
Object: Edge Management System

Subject: Edge Device

Context: The edge device has newly acquired information about a connected field device (5.1.1.1) that it did not yet synchronise with the edge management system.

Result: The edge management system has incorporated the field device into its representation of connected devices.

#### 5.2.3.2 [Propagation of Field Device Parameter Set](#)



## Unrestricted

In order for a field device to be commissioned, its working and communication parameters have to be set by the system integrator. After an edge device has retrieved the according parameter sets, this function deals with the forwarding of said information to the edge management system

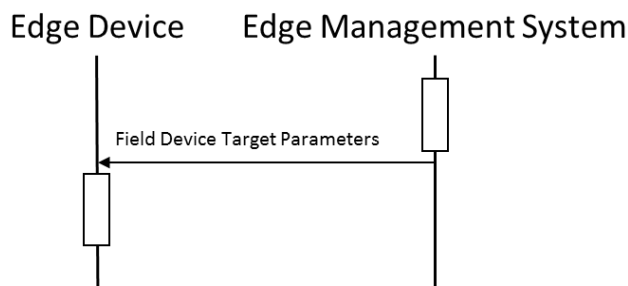
Object: Edge Management System

Subject: Edge Device

Context: After an edge device has retrieved a set of necessary parameters from a field device (5.1.2.1), this information can then be forwarded to the edge management system for interfacing towards the system integrator.

Result: The edge management system has received the necessary parameters for a field device and has interconnected this information in its data storage for further processing.

### 5.2.3.3 Forwarding of Field Device Target Parameter Values



This function serves as a forwarding of information from the edge management system to the edge device for setting values in the field device in another step.

Object: Edge Device

Subject: Edge Management System

Context: Whenever an Edge Management System ends up having target values for field device parameters that are not currently the actuals in the device, sending the target values to a connected edge device is necessary.

Result: The edge device has the target values for the field device's parameters.

### 5.2.3.4 Deploying an Edge Runtime

This function covers the deployment of an edge runtime to an edge device after it has been selected.

Object: Edge Device

Subject: Edge Management System

Context: A system integrator has selected an edge runtime for deployment to an edge device and the edge management system has been supplied with the appropriate access rights.

Result: The edge device runs the edge runtime.

## 5.2.4 Edge Management System to Application Store Communication Functions

Edge Management System to Application Store Communication handles all those communication relationships that exist between entities in the application store and entities on the Edge management layer. This kind of communication will usually encompass acyclic and event-based communication.

### 5.2.4.1 Propagation of an provisioned Application

This function propagates information about a new application towards the edge management system.

**Kommentiert [AGG14]:** Obtain edge runtime missing? Or in Selection covered?

## Unrestricted

---

Object: Edge Management System

Subject: Application Store

Context: A developer has provisioned an application to the application store and would like to make them available for use in the edge management system.

Result: The edge management system lists the new application

### [5.2.4.2 Propagation of an updated Application](#)

This function propagates information about an updated application towards the edge management system.

Object: Edge Management System

Subject: Application Store

Context: A developer has updated an application to the application store and makes them available for usage in the edge management system.

Result: The edge management system lists the updated application

### [5.2.4.3 Propagation of a discontinued Application](#)

This function propagates information about an application that has been discontinued towards the edge management system.

Object: Edge Management System

Subject: Application Store

Context: A developer has discontinued an application to the application store. The corresponding application is no longer available in the edge management system.

Result: The edge management system lists the updated application

### [5.2.4.4 Obtaining an application from the application store \(for deployment\)](#)

This function covers the act of obtaining an application from the application store for further deployment to an edge runtime.

Object: Application

Subject: Edge Management System, Application Store

Context: A system integrator has selected an application for deployment to an edge runtime and the edge management system has been supplied with the appropriate access right, for example after successful transactions, to obtain the application from the application store.

Result: The edge management system has the installation binaries/files

### [5.2.4.5 Obtaining an Update of an application](#)

This function covers the act of obtaining an application's update from the application store.

Object: Application itself and its update.

Subject: Edge Management System, Application Store

Context: A system integrator has selected an application to be updated in an edge runtime and the edge management system has been supplied with the appropriate access right, for example after successful transactions, to obtain the application's update from the application store.

Result: The edge management system has the installation binaries/files of the update

## 5.3 Rights Management Functions

Rights Management is necessary to allow Configuration functions that access data or parameters considered worthy of protection from unauthorized access. These functions provide a basis for assigning and checking access rights.

### 5.3.1.1 [Making an Edge Runtime available](#)

When an edge runtime is selected to be downloaded to an edge device, it first needs to be made available to the edge management system that is to perform the deployment.

Object: Edge Management System

Subject: Application Store

Context: A system integrator has selected an edge runtime from an application store and now wants to deploy it to an edge device.

Result: The edge management system is set up for deploying the edge runtime to an edge device.

## 5.4 Configuration Functions

Configuration functions are a category of functionality that allows manipulation of a system under consideration to make it fulfill a purpose given by external roles aiming at utilizing the system. These functions are at the core of what this document is trying to achieve. Every configuration function shall have an impact on the function of the system under consideration as a whole.

### 5.4.1 Selection Functions

Selection functions are those configuration tasks that focus on picking anything from the overall list of possibilities. Typical examples include the selection of an edge runtime that an application is supposed to be run on, or the selection of a data endpoint with output data to be transferred to another data endpoint as input data.

#### 5.4.1.1 [Selection of an Edge Runtime](#)

For an edge runtime to be installed on an edge device, it has to be selected from an application store.

Object: Edge Runtime

Subject: System Integrator

Context: The system integrator wants to commission an edge runtime to an edge device.

Result: An edge runtime is made available for download to an edge device.

### 5.4.2 Parameterization Functions

The Parameterization of entities encompasses changes to any sort of working parameter to be performed in order to alter the entities function. This may include communication parameters such as IP addresses or application enablement through defining available memory for an edge runtime, e.g.

#### 5.4.2.1 [Setting Field Device Parameters](#)

Either after a new field device is connected or when the system integrator deems changes necessary for system-external reasons, the parameters that a field device receives for its operation can be set by the system integrator in the edge management system.

Object: Edge Management System

Subject: System Integrator

Context: This function is performed either when a new field device is connected or when the working parameters of a field device need to be changed during operation or maintenance.

Result: The edge management system receives a new set of target parameters for a field device which it can propagate down to the edge and field device through according communication functions.

**Kommentiert [AGG15]:** Removal Functions missing!

### 5.4.3 Deployment Functions

The Deployment of entities (applications) encompasses the initial installation and configuration of the application in question. The deployment of applications has to set a lot of parameters to ensure the correct running of the application in question. Application deployment functions are in place to guarantee the successful setting of these working parameters and for setting up the application environment.

#### 5.4.3.1 [Deployment of an Application](#)

This function provides the capability of deploying a selected application in an edge device using the edge management system. The deployment of an application also includes the configuration of data exchange and initial parameters.

Object: Application

Subject: Plant Operator

Context: This function can only be triggered if the target edge device supports the deployment of an application.

Result: The application can be executed with initial parameter on the edge runtime. The edge management system is notified after the selected applications has been successfully deployed and started.

## 5.5 Provisioning Functions

### 5.5.1 Runtime Provisioning Functions

The Runtime Provisioning functions are those functions provided by a runtime to its applications. As applications can be almost arbitrarily complex pieces of software, the runtime they are supposed to be run on has to provide a certain set of functionalities that ensures seamless execution.

#### 5.5.1.1 [Provision of an edge runtime](#)

The function provision of an edge runtime provides the fundamental runtime environment to be hosted on an edge device so that applications can be deployed, updated, and executed.

Object: Edge runtime

Subject: Edge runtime provider

Context: Edge runtime must be tested before being provided into an application store of an edge management system. The provision of the edge runtime must consider the requirements of the edge device as well as the applications to be deployed in it.

Result: Edge devices hosting an edge runtime are capable of deploying, updating, and executing applications.

### 5.5.2 Application Provisioning Functions

The installation of applications has to set a lot of parameters to ensure the correct running of the application in question. Application Provisioning functions are in place to guarantee the successful setting of these working parameters and for setting up the application environment.

#### 5.5.2.1 [Provision of an application](#)

The function provision of an application sets up the working parameters and the application environment.

Object: Application

Subject: Application developer

Context: Application must be tested before being provided into an application store of an edge management system.

Result: The application can be offered in the application store and can be bought, deployed, and executed on an edge device or IT infrastructure.

### 5.5.3 Application Runtime Functions

While some applications can be run almost independently of their surrounding infrastructure, many need accesses to specific resources provided by their host system. The provision of such access is the task of Application Runtime functions.

## 5.6 Monitoring Functions

While configuration tasks aim at changing certain elements of a system under consideration, outside roles need means to identify the demand for a change of configuration in the first place. Monitoring functions allow the supervision of the workings of a system under consideration and provide alarms to any changes in the setup that require interaction by an affected outside role.

### 5.6.1 Asset Management Functions

Asset management deals with the economic handling of systems with regards to the status of their entities. In the context of a system under consideration, Asset Management functions can be utilized to handle changes to entities that are covered by the use cases given in the Usage view document.

### 5.6.2 Integrity Monitoring Functions

Not all changes to a system under consideration will always be planned events. Sometimes, malfunction, or unforeseen outside influence, can have an impact on the system. Integrity Monitoring functions allow outside roles who hold stakes in the system to detect any such unplanned events and to trigger proper event handling procedures.

## 5.7 Non-Functional Requirements

The following requirements are mandatory attributes of each of the functions discussed in the other subsections of this chapter. Since they hold true for all functions, they are not explicitly mentioned in each function's description.

### 5.7.1 Precondition Checks

For each function that is being executed, the preconditions according to the expected context given and according to any security policy have to be checked. If any of the preconditions for executing the function are not met, a warning should be propagated through the system-inherent mechanisms. Execution should be prevented in order to avoid destabilizing conditions in the overall system.

### 5.7.2 Acknowledgements

For each function that is being executed, an acknowledgement should be exchanged between Subject and Object. When the Object has to send a payload response as part of the functions, the acknowledgement should simply acknowledge receipt. Where the state of the object is to be adjusted, the changed state should be acknowledged towards the subject of the function.

## 6 Backlog

Handling of capabilities to be extended: Applications provided to an app store have to bring their filter criteria through capability detection with them. The handling of this information has to happen on every level and we should discuss this in the different functions.

### References

- [1] Industrial Internet Consortium, "Industrial Internet Reference Architecture".
- [2] Labs Network Industrie 4.0 Testbed Edge Configuration, "Usage View of the Testbed Edge Configuration," Labs Network Industrie 4.0, 2019.

Formatiert: Überschrift 1

Formatiert: Englisch (Vereinigtes Königreich)

Formatiert: Englisch (Vereinigtes Königreich)

## **67 Authors and Contributors**

### **6.47.1 Authors**

Dr. Andreas Graf Gatterburg, Hilscher

Dr. Ulrich Löwen, Siemens

Markus Rentschler, Balluff

Bernd Vojanec, Wittenstein

Dr. Dominik Rohrmus, LNI 4.0

### **6.27.2 Contributors**

Volker Heinzer, Schmersal

Bernd Fiebiger, KUKA

Detlef Tenhagen, Harting

Alexander Willner, Fraunhofer FOKUS