agrirouter Connectivity-Platform

Integration Guide (Part 1)

author(s): DKE-Data GmbH & Co. KG
last modification: 7/11/2018 5:12:00 PM
filename: DKE_agrirouter_Integration_Guide_Part_1_v2.3.docx
version: 2.3

Content

Table of Figures

Document history

version	date	author	Description
2.0	12.06.2018	DKE-Data GmbH & Co. KG	public release
2.1	26.06.2018	DKE-Data GmbH & Co. KG	Rewording of Inbox, Inbound Buffer, Outbox and Outbound Buffer to Routing, Inbox, Feed and Outbox Replace Picture of creating message; Mixed up 2 steps. Additionally, added chunking in the graphic Added Chapter for Certification Process
2.2	27.06.2018	DKE-Data GmbH & Co KG	Added Remark about needing only 1 developer account.
2.3	08.07.2018	DKE-Data GmbH & Co KG	Updates after Review - Added Chapter for general required technologies - Add chapters for description of CU onboarding processes - Added chapter for Reonboarding - Reformatted several parts

Review

version	Reviewer	Department/Company	Location	Rework remark
2.0	12.06.2018	DKE-Data GmbH & Co. KG		
2.2	27.06.2018	DKE-Data GmbH & Co. KG		
2.3	11.07.2018	DKE-Data GmbH & Co. KG		

1 Purpose

1.1 Purpose of this document

The purpose of this document is to create basic knowledge about the third-party integration with the agrirouter Connectivity-Platform. This document explains the process for common interactions – like onboarding, sending messages via commands or other integrations with the agrirouter. To ensure correct usage of the given functions, the document explains how to use the functionality. This document shall give a first overview of the agrirouter connectivity-platform integration.

1.2 Purpose of further documents

After signing the app and hardware provider contract, app and hardware providers will receive access to a more complete document with all URLs and information needed to integrate their software with the agrirouter. This more complete document will for example include a Postman Collection showing all commands.

Beside this document there will be already defined open source APIs available which ensure an easy and comfortable integration into the common environment of the agrirouter. See https://github.com/dke-data/

1.3 Purpose of “terms and short description”

Writing about the creation of software for agrirouter requires a very specific use of terms, as there are several functionalities and several layers of functionalities. Every chapter has a paragraph called “Terms and short description”, that shall show the terms new in this chapter and their relation to each other.

The several terms are described in the further paragraphs of each chapter so that – after reading the whole chapter – the short description should make good sense to the reader.

2 Epilog

Digitization in agriculture makes good progress. One unsolved problem so far are the different and customized developments that are used for the data exchange, for instance between machine and software products of different manufacturers. DKE-Data GmbH & Co. KG has a solution for this.

The company DKE-Data GmbH & Co. KG (DKE is the German abbreviation for digital communication and development) is currently developing a cross-manufacturer agrirouter Connectivity-Platform that is far more than merely hardware and software. Customers are making use of a service: safe and cross-product data transport.

3 General information

3.1 Terms and short description

This chapter will introduce and declare different new terms. The following text gives a short overview of what you will learn in this chapter and how the terms connect to each other. After reading the whole chapter, the following sentences should make good sense to you.

agrirouter is a neutral data exchange platform.

The data exchange shall be performed between different software products of different manufacturers.

End users of agrirouter are participants of the agricultural process like farmers and contractors.

The new terms will be described in more detail in the following paragraphs.

3.2 What is the agrirouter?

The agrirouter is a cloud solution, that allows farmers and contractors to transfer data between farming machinery from several manufacturers and several farming applications from different application providers. With central maintenance of data routing between endpoints, there is no need to configure network connections between all data sources and targets on each machine/ or in each application.

Figure 1 agrirouter Connectivity-Platform

3.3 Advantages for end users

Users can configure data transfer permissions centrally on the agrirouter Connectivity-Platform, to control what data can be exchanged between which endpoints. Data can be published on the agrirouter Connectivity-Platform and routed to interested and allowed endpoints.

The agrirouter Connectivity-Platform provides a common cross-vendor data exchange mechanism and protocol to exchange several common message formats, which can be implemented by different machine manufacturers and application providers. Examples for the common message formats are ISOXML, images and videos.

The transmission and evaluation of live telemetry data is getting more and more important. With the agrirouter Connectivity-Platform, machines can send telemetry data to the agrirouter, where data is automatically distributed to several applications, based on permissions and configuration. For live telemetry data, the agrirouter routes the content of the messages according to the user-defined permissions.

This way it can be achieved that one application gets only application rate data, for example, while another one gets only machine-related data such as fuel consumption.

Technically, the agrirouter Connectivity-Platform is a cloud system, which provides the messaging service and a web site with self-services for the different user types.

When the recipient of a message is offline, the message is buffered on the agrirouter until the recipient is online again. However, the message is not buffered forever. After a maximum buffer time, the buffered messages are removed from the buffer even if they are not yet delivered.

The agrirouter Connectivity-Platform stores additional data needed for rights management and command processing. This includes the list of endpoints attached to agrirouter account, and the technical device description of the connected agricultural machines. This information is needed for message routing. In addition, the agrirouter stores usage statistics data needed for billing.

The agrirouter also offers a technology to adjust manufacturer specific DDIs once they are standardized. This way, application providers do not have to care about manufacturer specific DDIs anymore, they can just use the standardized version even for older machines.

3.4 Advantage for App-, CU- and Telemetry providers

As provider of an application that could be connected to the agrirouter, you save a lot of time implementing and maintaining several interfaces to different endpoints of different brands.

The more providers are using the agrirouter solution, the better it is. With more and more different data providers entering the agrirouter, it becomes much easier to implement new data formats.

The compatibility with agrirouter and the usage of the market place gives your product an additional marketing aspect.

4 Accounts

4.1 Terms and short description

This chapter will introduce and declare multiple new terms. The following text gives a short overview of what you will learn in this chapter and how the terms connect to each other. After reading the whole chapter, the following sentences should make good sense to you.

agrirouter can handle multiple users.

A user can create an account at the agrirouter.

One account is meant to represent one company, e.g. a farm or a contractor.

The users of the agrirouter create end user accounts.

Application providers need a Developer Account.

The new terms will be described in more detail in the following paragraphs.

4.2 End-User (Farmers, Contractors, Consultants)

The end-user has access to the graphical user interface of the agrirouter for end users.

The user interface offers possibilities to handle the account:

- onboard new endpoints
- create groups of endpoints
- connect an account with other agrirouter accounts to exchange data
- setup routings to allow or disallow message delivery to specific endpoints or endpoint groups
- create endpoint records for service purpose
- modify his user interface (e.g. color scheme)
- view a list of news/updates for his account (e.g “account connected”, “endpoint onboarded”, etc.)

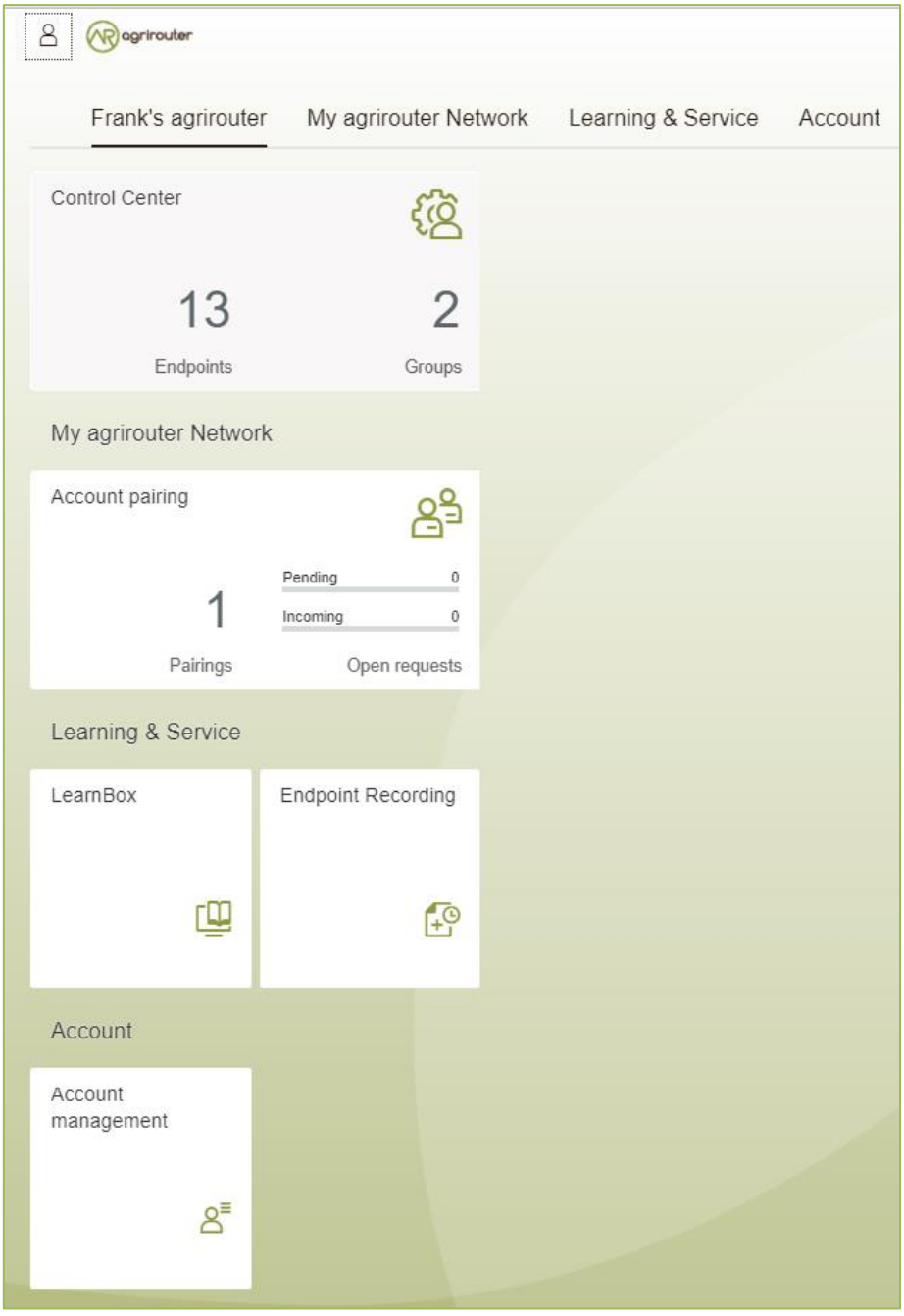

Figure 2 User Interface for End Users

4.3 Developer

The developer has access to the graphical user interface of the agrirouter for developer's users.

A developers account includes an end user account, therefore, the developer account has all functionalities of an end user account.

Remark
There is only 1 developer account needed **per company.** Testing the app; especially in the Quality assurance environment can be done from an ordinary end user account.

Additionally, developers can manage their own applications:

- Create new applications
- Modify the information for the application like name, description or symbol
- Add new software versions for Certification
- Manage the required capabilities of the application
- Block specific endpoints, e.g. to avoid misusage of applications
- Add testers to test an application before releasing to public
- Replay Endpoint Records
- Export metrics of agrirouter usage for billing purpose

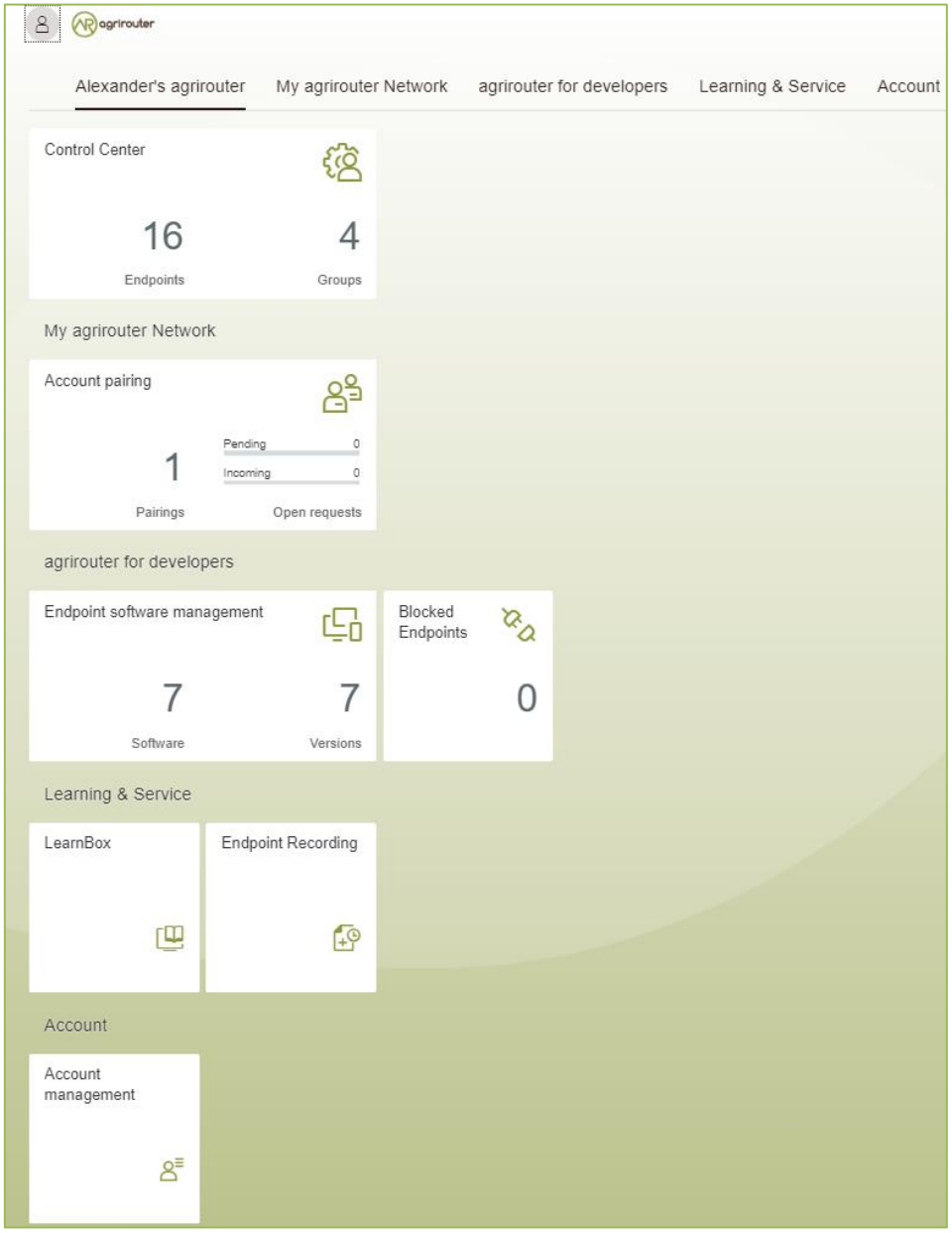

Figure 3 User Interface for Developers

4.4 Administrator

The administrator account is only accessible by DKE. DKE can manage application, technical message types and information types and certification.

5 Applications

There are different types of software, that communicate with the agrirouter. This software is called application, shorting is app. In general, agrirouter only communicates with software. Sometimes, the software might be bound with a hardware and therefore only be available to a customer with that hardware. Important for agrirouter communication however is the software communicating inside the hardware with the agrirouter.

5.1 Terms and short description

This chapter will introduce and declare multiple new terms. The following text gives a short overview of what you will learn in this chapter and how the terms connect to each other. After reading the whole chapter, the following sentences should make good sense to you.

Applications can be created to be connected to the agrirouter.

An application is a software product.

An application can be a communication unit, a telemetry platform or a farming software.

The new terms will be described in more detail in the following paragraphs.

5.2 Communication Units

Communication units (CU) are mobile applications, mostly running on embedded systems or mobile devices to establish a connection to the agrirouter “in the fields”. CUs are used to collect information like machine data or geo positions.

CUs are usually but not necessary connected to machines via ISOBUS and collect their data during the common ISOBUS interaction.

CUs should have a stable internet connection to provide live data. Otherwise, a CU should buffer generated messages to send them, once the internet connection returns.

Like any other endpoint, CUs are able to request information from the agrirouter, e.g. to inform a driver about other machines positions (fleet management). The more important job of CUs however is to provide data to agrirouter.
A list of compatible CUs can be found here: https://my-agrirouter.com/en/marketplace/telemetry-units/.

Please note, that there might be additional CUs in development.

Important note:

Communication units can also be tablets or other devices and should meet the following **base requirements**:

Every single device has to have internet connectivity and must be able to communicate via HTTP(S) with REST or via MQTT. The communication consists mainly of payloads in Google Protocol Buffer format.

Endpoints without a constant internet connectivity are not able to send real time data. Sending recorded data after establishing an internet connection will be a valid use case for endpoints not having internet connectivity during the main working process of a machine / -s.

5.3 Telemetry platform and virtual CUs

A telemetry platform is a cloud software solution, that provides access to multiple so called virtual CUs. Virtual CUs are comparable to real CUs, but in difference, they do not implement the agrirouter protocol. Virtual CUs implement a proprietary protocol to connect to the telemetry platform only.

The telemetry platform is than connected to the agrirouter providing the data of selected virtual CUs to the connected agrirouter account. The onboarding process of a telemetry platform differs from the onboarding process of a CU, as the user might not have to enter a TAN in the CU. In the agrirouter UI, the telemetry platform and each virtual CU are displayed as endpoints.

A list of compatible CUs can be found here: https://my-agrirouter.com/en/marketplace/telemetry-units/. Please note, that there might be additional telemetry systems in development of the compatibility.

5.4 Farming software

Farming software can be a cloud application or a software released on a single device like PC or Smartphone. Farming software mostly consumes data to provide extra services for farmers or contractor. FMIS (Farm Management and Information Systems) are a common example for farming software.

A list of compatible CUs can be found here: https://my-agrirouter.com/en/marketplace/apps/. Please note, that there might be additional software in development of the compatibility.

6 General agrirouter integration process

6.1 Terms and short description

This chapter will introduce and declare multiple new terms. The following text gives a short overview of what you will learn in this chapter and how the terms connect to each other. After reading the whole chapter, the following sentences should make good sense to you.

Companies and developers can create software, that integrate an interface to the agrirouter.

This software is called application.

The short term for application is app.

The providers of applications are app providers.

Apps need to be certified to be onboarded to the agrirouter.

The new terms will be described in more detail in the following paragraphs.

6.2 General

To maximize the quality of the agrirouter ecosystem, there are several requirements additional to the general development of the application itself. This chapter shall describe all necessary steps additional to the development of the application to integrate this app with the agrirouter.

6.3 Overview of the process

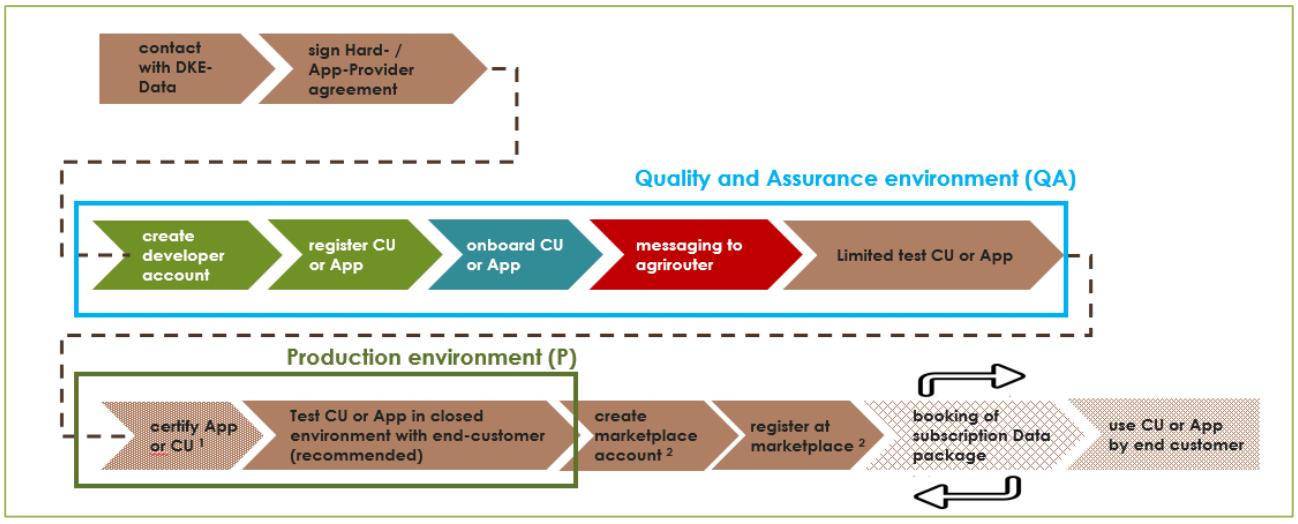

Figure 4 General agrirouter integration process

After the first contact with the company DKE-Data and signing the App- and Hardware Provider agreement, an agrirouter developer account can be created in the agrirouter quality assurance (QA) environment.

In the developer account, a new farming software, communication unit or telemetry platform can be added. The app provider has to create a first Certification version in the UI, that can be approved by DKE.

Afterwards, it is possible to register and onboard endpoints of this software as well as any other software endpoint. It is important to know, that the quality assurance environment is not made for real life productive data exchange, but only for test of the connectivity of a software. The number of endpoints, that can be created on the quality assurance environment is limited.

Once the connectivity to agrirouter is successfully developed and tested, the developer sends a request to agrirouter support team for starting a certification process.

DKE will provide a list of a trusted partners for agrirouter certification on the https://my-agrirouter.com/support/ website.

If the agrirouter certification of the application is successful, it is recommended to do live test on the agrirouter production environment with real accounts of selected farmers. The developer can add those accounts to his test environment in the developers' user interface.

To release the new application to public, the app provider can create an application description in the agrirouter marketplace. See https://my-agrirouter.com/marketplace/apps/

6.4 Register Application

Applications must be certified before they can be provided to the public.

After creating an agrirouter developer account in quality assurance environment, it is possible to register a communication unit, application or telemetry platform.

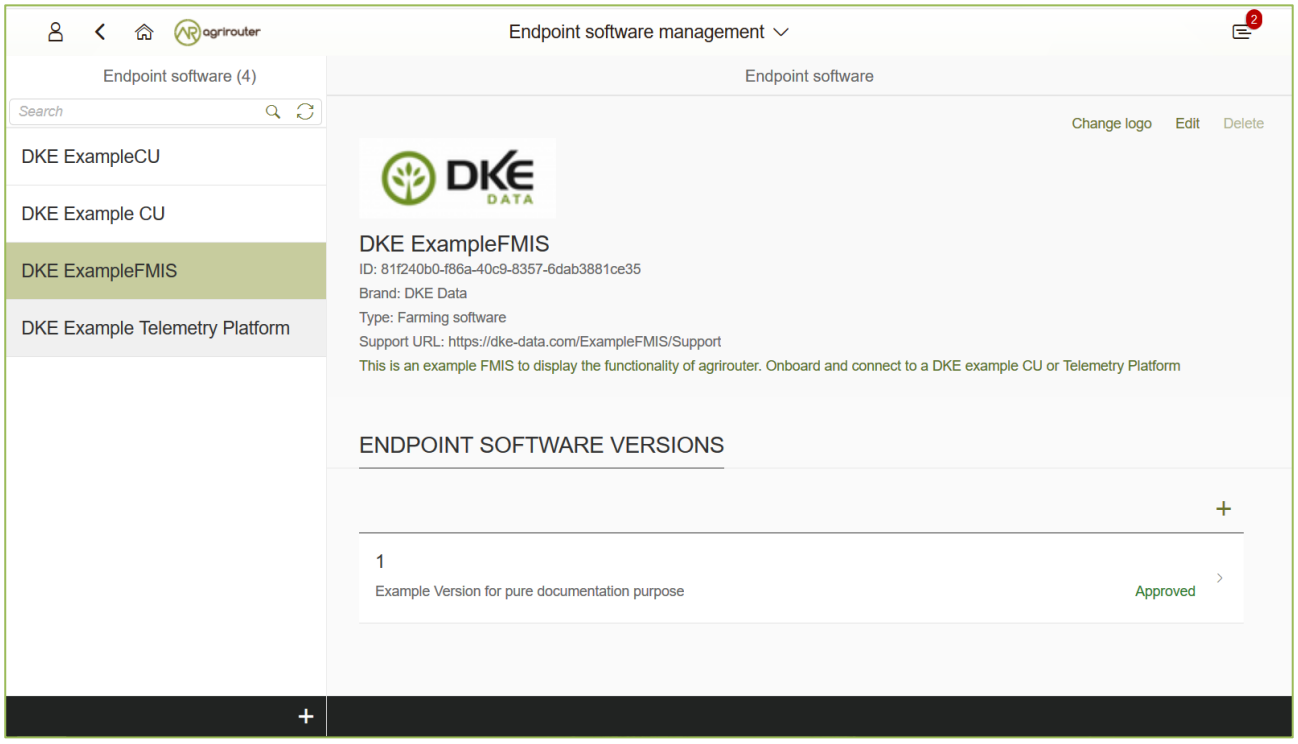

Figure 5 agrirouter endpoint software management

6.4.1 Mixed Applications

It's not always obvious, to which category a new application belongs. To simplify the decision, DKE advices the following decision tree:

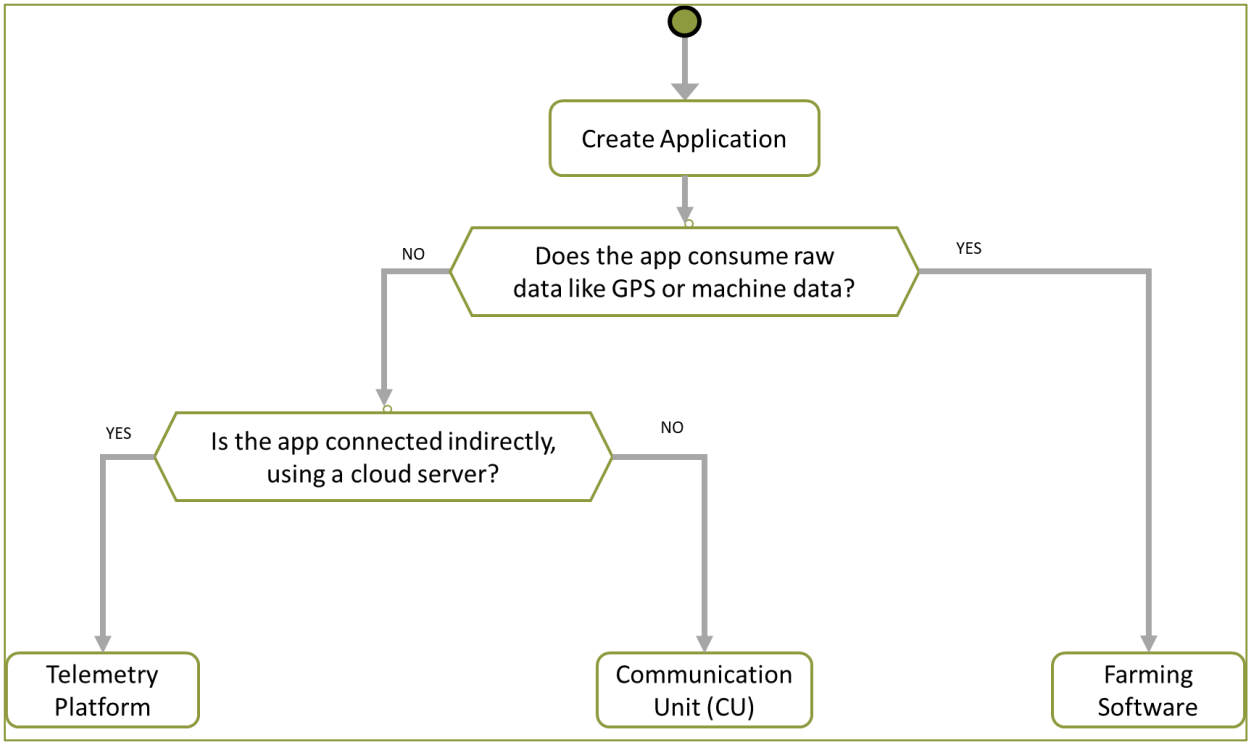

Figure 6 Decision Tree: What type of application do I create?

The following information should be available to create a communication unit, application or telemetry platform:

#	Description	Remarks
1	Name	Short name of application
2	Brand	Your company or product brand
3	Type	farming software, communication unit (CU) or telemetry platform
4	Support URL	Your company or product support URL for end user
5	Description	Descriptive text, including main features, supported message in human readable format, required DDIs
6	Provider or product logo	The logo will be visible to end users and should clearly identify your company or product. The logo must comply with the following rules: • Allowed formats: .png, .svg or .gif • Maximum size: 512KB • Must have a transparent background

7	Certification version	Purpose of this field is to differentiate multiple versions of the application; for each new version with new capabilities, a new certification version must be created. **Remark:** an Application can also be the software part of a CU. see Glossary. Version Number of the certification: (Integer (!) version Id) this is a version that requires a DKE certification (typically because the capabilities have changed compared to previous version) **Remark:** This is not the Software version (like 1.3.12), but just an internal identifier for the DKE certification.
8	Technical message types:	Technical message types the endpoint can send and receive.

All required steps are available in Endpoint Software Management (see Figure 11 and Figure 12) in the section agrirouter for developers (1) in the launch pad:

- List of existing applications (2)
- Registration of new application (3)
- Details on application (4)
- Application versions (5)
- Creation of new application version (6)

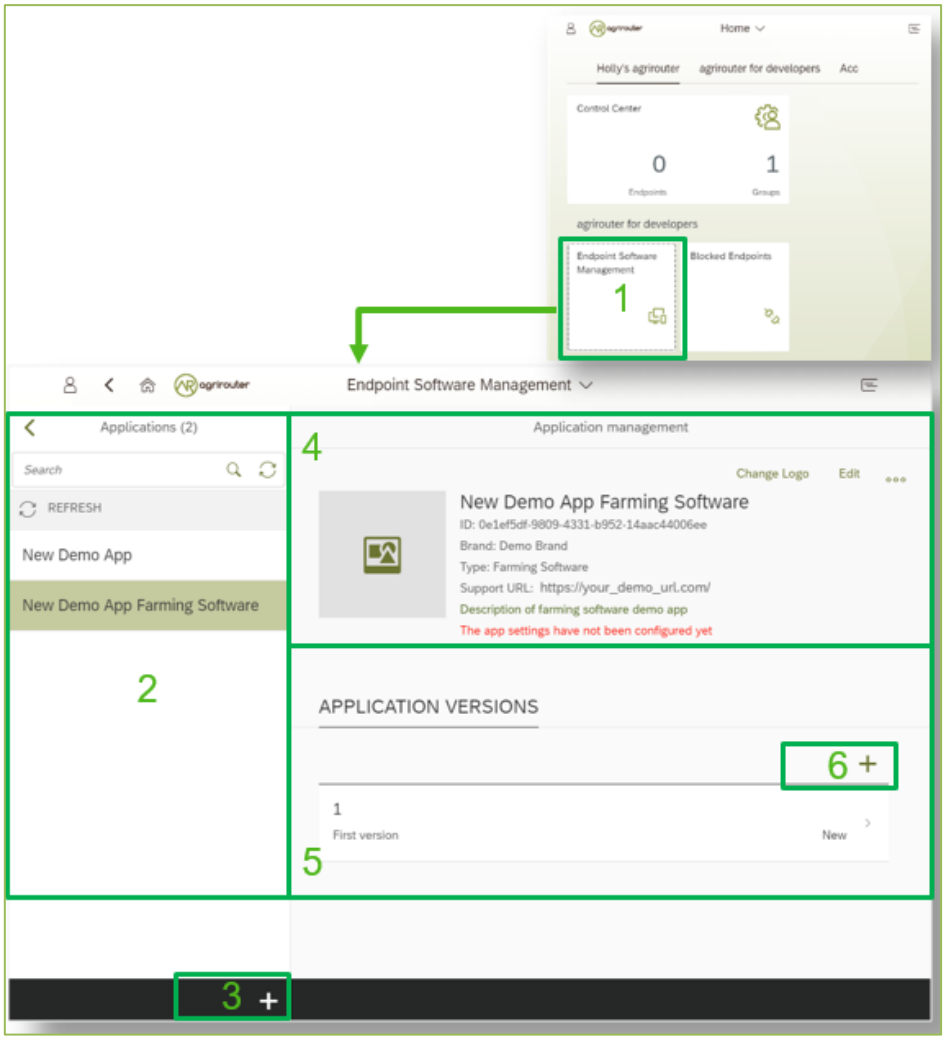

Figure 7 agrirouter application registration

When a new version has been created, the supported capabilities for this version can be defined. With the button “New”, the technical message types supported by this application version can be selected. Finally, the direction (send/receive) of the capabilities must be specified before the version can be submitted.

After submitting, DKE will receive a notification and set the status to *Approved for Testing* or *Approved*, so that instances of this endpoint can be onboarded. Please also send an email to support@my-agrirouter.com, once you created a new software version that shall be approved for testing.

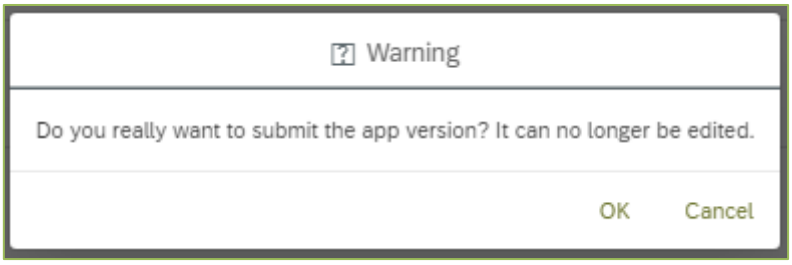

Figure 8 agrirouter warning

Once the request has been submitted by the application developer (the status of the certification request changes to *submitted*),

- it can no longer be changed by the application developer
- the agrirouter shall create a unique certification version ID
- both IDs shall be displayed in the detail view of the certification request

6.5 Application certification

Every application (communication unit, farming software or telemetry platform) has to be certified by the DKE or a trusted partner of the DKE. During this certification process every application gets a unique certification identifier. This identifier ensures that the certification can only be applied to the single version of the application.

The certification process shall disclose name, version information and supported technical message types, in order to allow the agrirouter to check whether the application supports a certain technical message type and to properly display the name of the application as an endpoint in an end-user's account after registration of an instance of the endpoint.

The certification process also ensures, that there will be a unique ID for the application which the application can send to the agrirouter in order to allow the agrirouter to look up its capabilities.

The certification process allows the agrirouter to check the state of the application with a given ID. The agrirouter needs that to make sure, that only certified apps communicate with the productive platform.

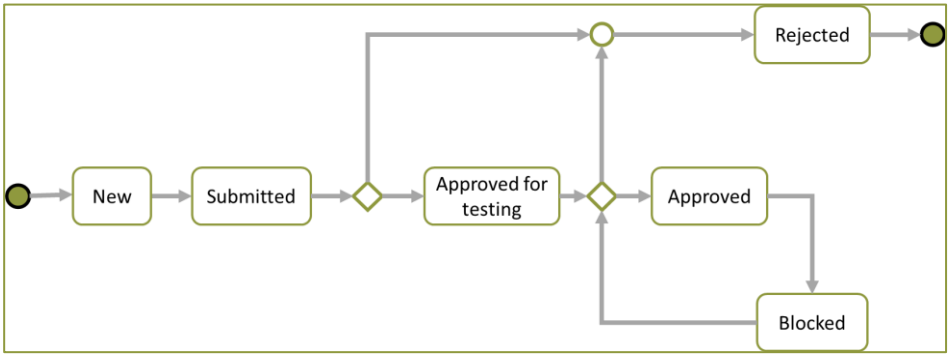

Figure 9 Workflow for application certification; status flow

This certification needs to be done initially once for each application before instances of the application may be registered within end-user accounts.

After the initial certification, each time changes to the capabilities are made in subsequent releases, a new certification is required for the new version of the application.

The app provider has to create a new version in the agrirouter user interface. This new version has the initial status *New*.

Once he selected all the desired capabilities for the new version, he can submit this version and the status changes to *submitted.*

DKE initially decides, if this version can be certified and sets the status to *Approved for testing*.

An application developer and his listed test customers can now onboard instances of the new application version.

After testing, the app provider can certify his app to set it to *approved*. Now everyone can onboard instances of the app.

If the app provider decides, that an app version is outdated, he can set the status for this app to *Blocked* and instances of this app will no longer be able to send messages to or receive messages from the agrirouter. If the test fails, the app cannot be approved for testing or an app is finally blocked, the status will be set to *Rejected*.

Important note:

For communication it is indispensable that this certification id is transmitted with the onboarding request and the capabilities message. The developer is obliged to store this certification id as well as the application id securely in every delivered application or application instance. The storage has to be encrypted.

The agrirouter will not provide functionality to validate, that applications correctly handle the exchanged data content. Certification partners only certify the capabilities of an application for implementing the agrirouter communication protocol, meeting some requirements. DKE and the agrirouter support team is not allowed to take a look inside the message body.

6.6 Approval of the application certification

For all certification requests in status *submitted*, agrirouter administrators are able to change the status:

- either to rejected, making further processing impossible, or
- it can be set to *Approved for Testing* if the certification request was successful.

In status *Approved for Testing*, application developers can register farmers' accounts as test accounts. If a certification version is in status *Approved for Testing*, instances with this certification version can be registered in these test accounts and an external test can be started. The developers account automatically is an end user account, that is a test account. This means, that a developer can onboard an endpoint in his account.

Using other accounts is not possible, because the certification version is not yet approved by DKE for public use. When the certification request is in status *Approved for Testing*, the agrirouter administrator can either finally set the status of the request to *Approved* in case of successful testing or *rejected* if testing was not successful.

7 agrirouter EcoSystem

The agrirouter is a central data exchange platform, but to understand its full power, it's important to understand, that the data provided from the agrirouter comes from the Applications attached to the agrirouter. The following figure lists different Applications, that can be attached to one end user account.

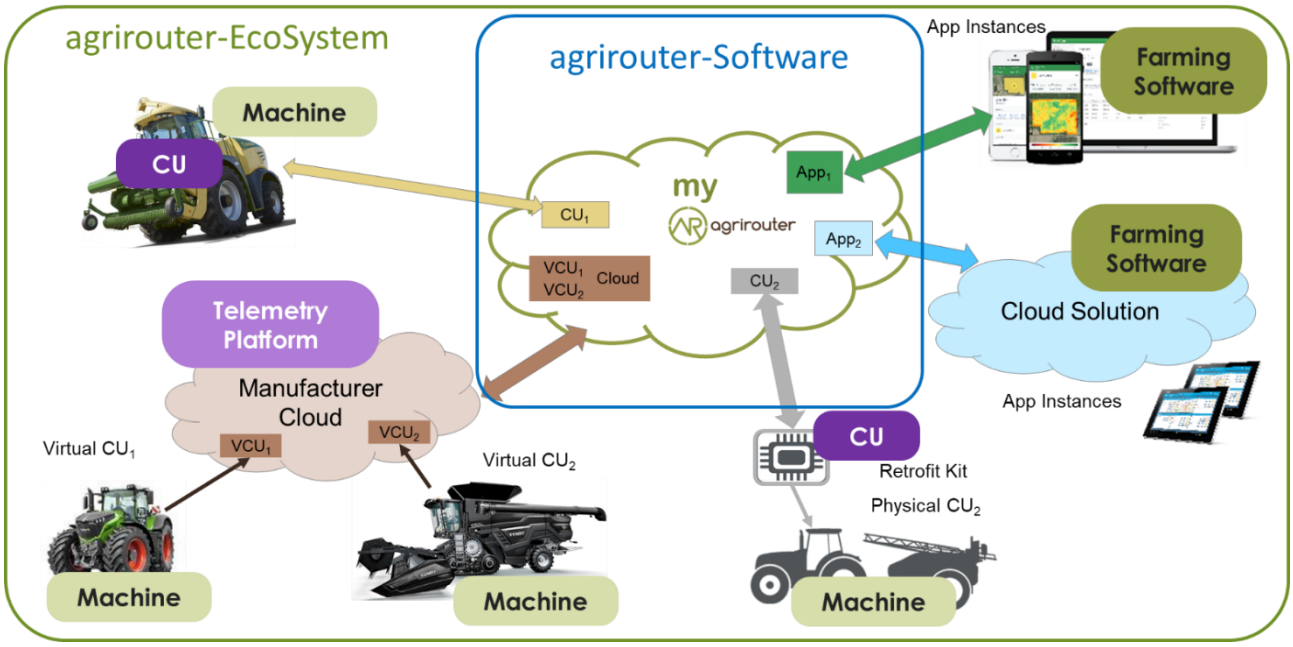

Figure 10 The agrirouter ecosystem of members

7.1 Terms and short description

This chapter will introduce and declare multiple new terms. The following text gives a short overview of what you will learn in this chapter and how the terms connect to each other. After reading the whole chapter, the following sentences should make good sense to you.

Applications can be CUs, Telemetry Platforms or Farming Software.
Every end user can onboard one or more application instances of any application to his account.
Application instances could be a CU or an account or installation of a farming software or telemetry platform.
Telemetry platforms can onboard additional application instances; these are called virtual CUs.
Each application instance will create an endpoint in the agrirouter account of the end user.
App Instances communicate with their endpoints at the agrirouter.
One application instance (e.g. a CU) could be onboarded in multiple end users accounts.
The application instance will than have multiple endpoints in the agrirouter; one per account.
CUs and Virtual CUs can report attached machines.
Each machine will receive its own endpoint in the agrirouter, once it was reported by a virtual CU or a CU.
Every agrirouter account can be connected to another users account, if both users agree.
The other users account will than become an endpoint within the end users agrirouter account.
Messages sent to this endpoint will become messages to the public address of the other users account.
An endpoint of the other users account cannot directly be addressed from the end users account.

Machines of the other users' accounts will become endpoints in the end users account.

The new terms will be described in more detail in the following paragraphs.

7.2 Hierarchical Overview of endpoints

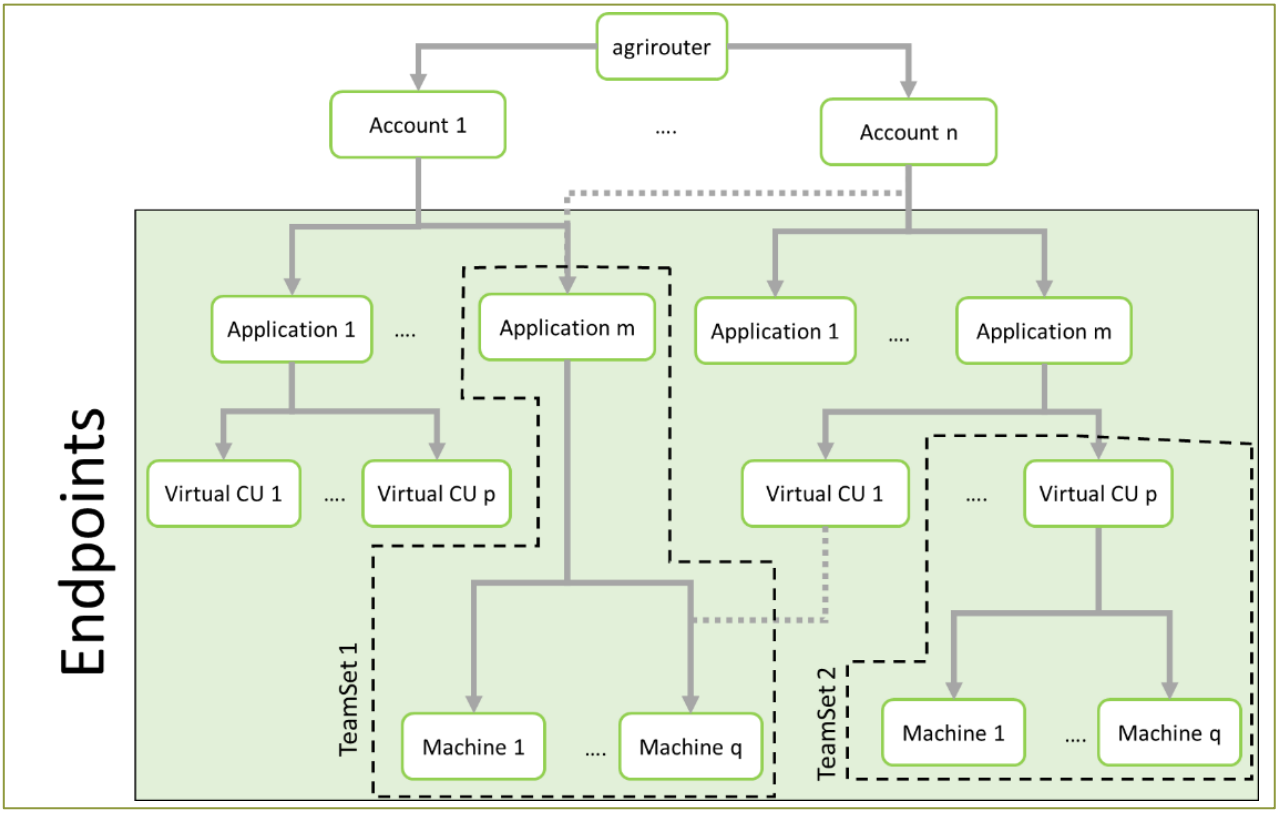

Figure 11 A hierarchical overview of agrirouter ECO System

There is 1 agrirouter solution, with multiple accounts, one for each end user. An end user equals a Farm, a Contractor, etc.

Every Account can onboard multiple Application Instances.

Applications can be onboarded to multiple accounts (dotted gray lines).

Applications can be CUs, Farming Software or Telemetry Platforms.

Applications of type "Telemetry platform" can onboard multiple virtual CUs.

Machines are connected to CUs or Virtual CUs. Machines can be attached to different CUs or Virtual CUs.

A teamset is one (virtual) CU with zero to n attached machines.

If there are 2 CUs installed on a real-world machine, a machine can be found in multiple teamsets and therefore even in multiple accounts.

Applications, Virtual CUs and Machines are endpoints, that can be source or target of a message.

7.3 Machines

You might wonder, why machines are not listed as applications. The answer is easy: Machines cannot be directly connected to the agrirouter. Machines are always connected through an application like a CU or a virtual CU.

(Virtual) CUs, that are connected to the ISOBUS can provide the device description of connected machines and send live telemetry data. If a CU provides the device description of connected machines, these machines can be addressed by agrirouter messages. A farmer could for example send an initial taskset to his seeder and the taskset is delivered to whichever CU reports to agrirouter, that it is connected to this seeder.

7.4 Communication addresses: Endpoints

An endpoint is an addressable communication address for an application instance connected to the agrirouter. One application instance can be part of multiple agrirouter accounts or there can be multiple instances of the same application in one agrirouter account. An example for multiple instances of the same application in one account are multiple CUs onboarded to one account.

The address of an endpoint is used by its corresponding app instance to communicate with the agrirouter and by other app instances within the same account to address this app instance.

7.5 Connected end user accounts

It is possible to connect 2 agrirouter accounts with each other using the email address of the end users and setting up a connection using the graphical user interface of agrirouter. Each connected agrirouter account gets its own endpoint in the partners agrirouter account and vice versa.

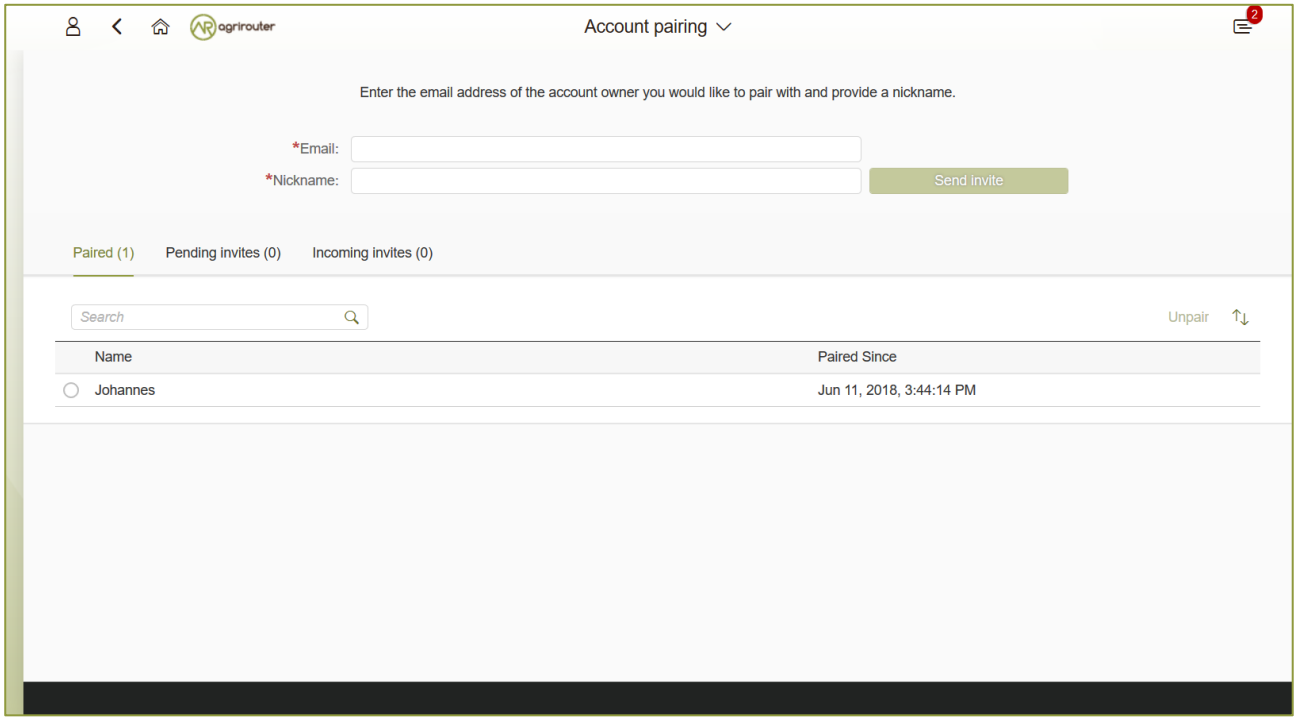

Figure 12 List of paired accounts

Important note:

It is not possible to address an endpoint inside another agrirouter account, neither is it possible to list the endpoints of this account.

7.6 Teamset

A teamset is a set of connected machines which work and move together and are connected to the same communication unit. The machines in the teamset are typically connected physically and informationally (for example via ISOBUS).

A (virtual) CU is responsible for the agrirouter communication of one teamset. It sends descriptions of the machines in the teamset whenever the teamset changes or when the descriptions of at least one of the machines changes (for example because of a reconfiguration or the CU connects to another machine). This way the agrirouter knows about the machines themselves, and about which machine is connected to which communication unit.

Each CU only sends one teamset, every teamset can only be part of one CU. If multiple CUs are on the same network (e.g. a terminal in the tractor + a telemetry box on the baler), there will be multiple teamsets in agrirouter including the same machines and sending the same data. Apps are then responsible for filtering duplicated data. If there are no machines connected, the teamset of a CU will just be empty.

8 Major development technologies

To perform an agrirouter integration, you will need knowledge to the following technologies:

- Required:
    - REST; at least for onboarding, perhaps for communication (see https://wikipedia.org/en/REST)
    - Protobuf: for message packaging (see https://developers.google.com/protocol-buffers/)
    - Certificates, signatures and encrypted communication, e.g. using HTTPs
- Optional
    - secured MQTT for the communication (see https://mqtt.org)
    - ISO11783 for telemetry data (see Integration Guide Part 2)

9 Concept of Message exchange

Important note:

This chapter only handles the message exchange on the agrirouter in a general way. This chapter does not handle network layers or TCP-IP protocols. The communication of an application with its endpoint at the agrirouter is described in 10 Application Interaction with agrirouter

9.1 Terms and short description

This chapter will introduce and declare multiple new terms. The following text gives a short overview of what you will learn in this chapter and how the terms are connected to each other. After reading the whole chapter, the following sentences should make good sense to you.

agrirouter provides an inbox, an outbox and a feed unique for every endpoint.

Message forwarding:

Every App Instance can send messages to its inbox at the agrirouter.

Each message has a technical message type(TMT) and a list of recipient addresses.

Instead of or in addition to the recipients list, a message can also be published.

If a message is published, agrirouter adds all endpoints to the recipients list, that are subscribed for this TMT.

agrirouter forwards the messages to the feed of all relevant endpoints.

Each Application Instance can subscribe its endpoint for technical message types within its account.

Routings:

Messages are only delivered, if there is a routing for that.

Routings are used to control the message flow and disallow wrong message flow.

Routings are created by the end user.

Each routing consists of a sender, a receiver and a list of information types and categories.

Information types are lists of technical message types, used to simplify the routings creation.

Categories are lists of DDIs, used to simplify the routings creation for telemetry data.

Categories are used to filter telemetry messages, only forwarding allowed Categories DDIs values.

Grouping:

For simplification, endpoints can be grouped into endpoint groups.

Endpoint groups are only relevant for routings creation in the user interface of an end user.

Inter-account communication:

The connected account of another user will be a single endpoint in the end users agrirouter account.

Endpoints of another users account are not directly addressable by an endpoint of the end users account.

For connected accounts, messages are only delivered, if routings are created in both users' accounts.

The new terms will be described in more detail in the following paragraphs.

9.2 Overview

The main purpose of the agrirouter platform is to exchange messages between endpoints.

The agrirouter provides a mechanism to receive and forward messages according to a combination of app subscriptions, public or direct addressed messaging and routings configured by the end-user.

In general, there are 3 steps:

1. An application instance sends a message to its endpoints inbox
2. agrirouter forwards (“routes”) this message to all relevant and permitted endpoints storage (“feed”)
3. Each application instance can call for available messages in its endpoints feed and pull them from the outbox

The following graph shows the way of a message through the agrirouter:

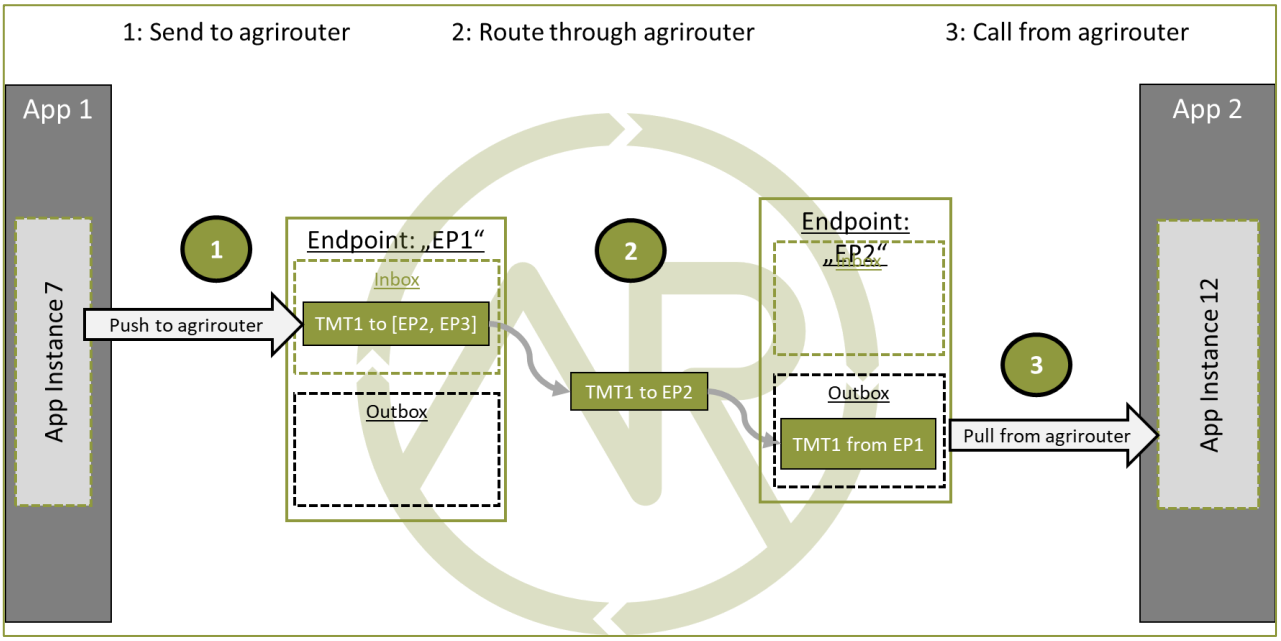

Figure 13 Way of a message through agrirouter

Every endpoint can send messages to specific endpoints and/or publish them.
Every end user can provide routings to allow this message delivery. Per default, no routings are set and therefore, the delivery is disallowed. This can be changed easily, see .9.10.2 Setup of routings.

If messages are sent published, they are only delivered to those allowed endpoints that subscribed for this message type. If messages are sent directly and a recipient are mentioned in the address lists, they are delivered, if the endpoint supports this message type and if there is a routing allowing the delivery.

9.3 Message addressing

Messages can be directly addressed to one or more endpoints. They can also be published to send the message to all subscribed endpoints. It is remarked in the message header, if the message is directly sent, published, or if both techniques shall be used. agrirouter will take care of not delivering a message twice to the same endpoint.

As a machine has its own endpoint in the agrirouter account, it is possible to address messages to a machine.

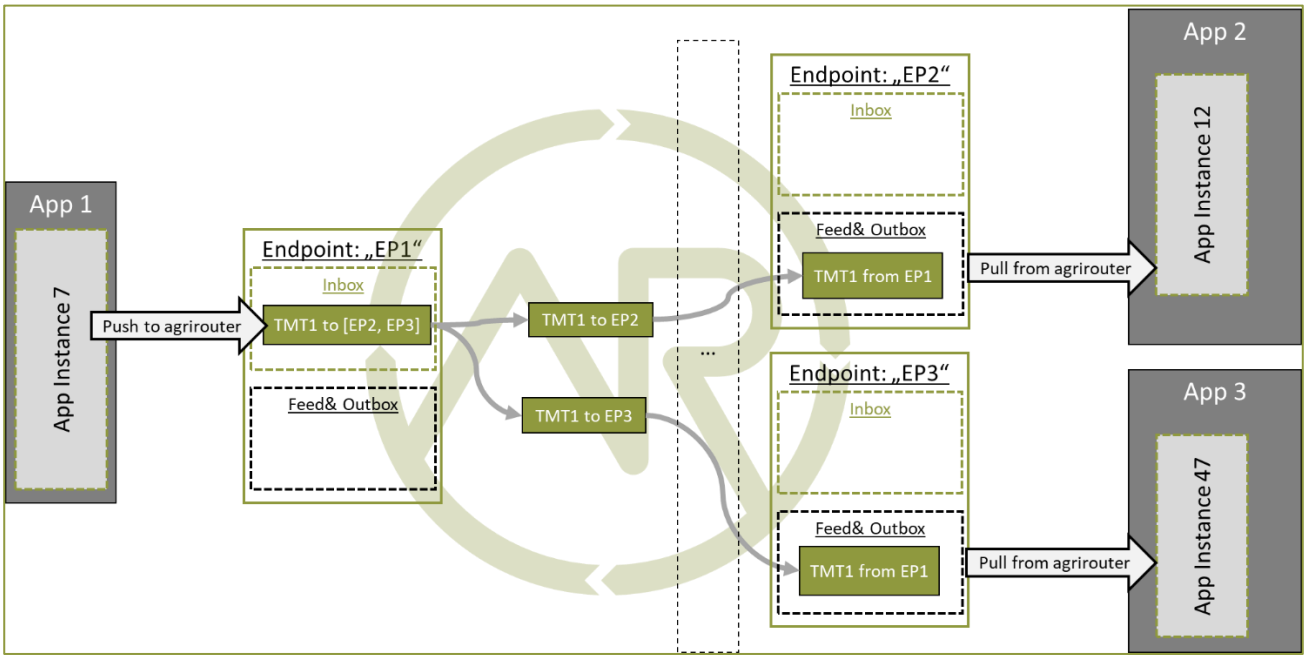

Figure 14 Message addressing to multiple endpoints; simplified

Important note:

To deliver messages, that are addressed to a machine, agrirouter needs to know, which applications currently is connected to that machine. Therefore, delivery of messages to a machine can only work, if the connected machine properly reports its teamset including the EFDI Device Description. See later following chapter 9.11.3 Technical Message Types

Currently, this is an optional feature of Virtual CU or CUs and not required for a certification.

9.4 In- and Outbox and Feed

After onboarding, every endpoint can send commands to the agrirouter inbox. agrirouter forwards and potentially duplicates the included messages to store them in the feed of the relevant target endpoint(s).

The feed is the time limited storage for messages in each endpoint.
All messages in agrirouter Connectivity-Platform have a retention time of maximum 4 weeks in total before they are deleted.
The end-user will be informed about a possible data loss after 3 weeks, so that he has time to react.

Every registered endpoint can request the list of messages available in his feed and request specific messages. agrirouter forwards those requested messages to the outbox, delivery to the endpoint depends on the protocol.

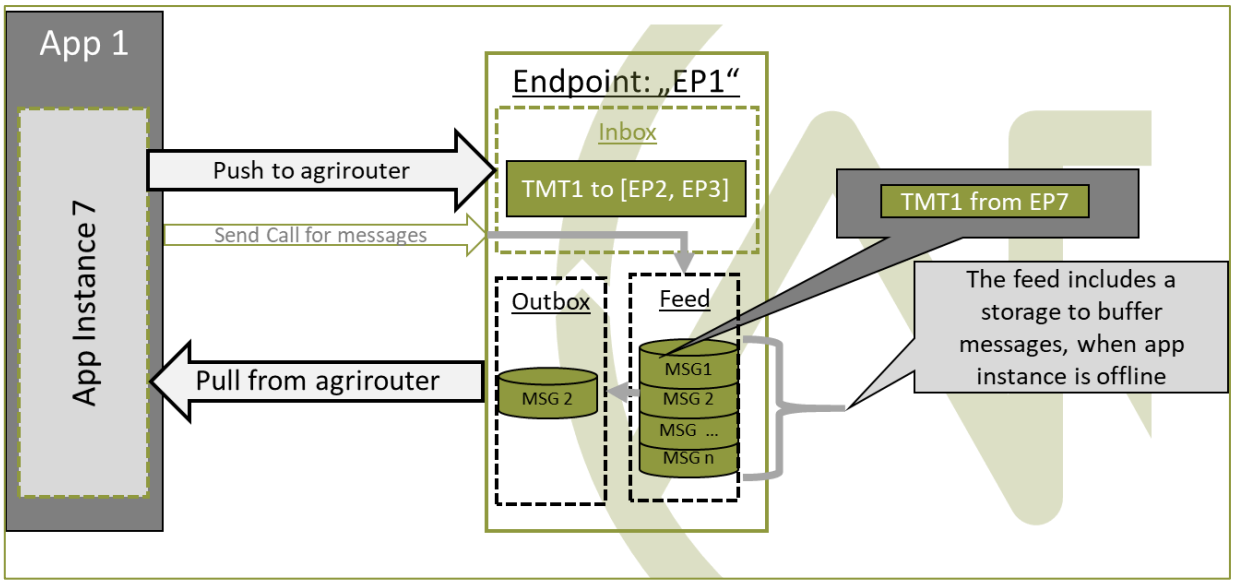

Figure 15 In- and outbox of an agrirouter endpoint; simplified

Example:

An App instance sends a message to the inbox of its endpoint.
The message is addressed to another app instances endpoint.
agrirouter forwards this message to the feed of the addressed endpoint, if there is a routing.
The addressed App Instance is than able to request agrirouter to pull this message from its endpoints feed to its endpoints outbox.
From there, it can be pulled by the app instance.

9.5 Message Header and Message Payload

Every message or command sent to agrirouter consists of a Message header and a payload.

9.5.1 Message Header

The header includes information on how to handle the message. When sending a message, it includes the recipient list and which technical message type is encoded in the payload. It also includes the creation timestamp and a unique ID. When receiving a message, it includes the source, creation date, technical message type and a unique ID

9.5.2 Payload

The payload includes the encoded raw data of the message content. Its structure and content differs depending on the technical message type.

9.6 Information types and Technical message types

The technical message type describes the type (e.g. the format) of the content of an agrirouter message. Every endpoint capable of sending such a message declares, that it is able to create a valid message of such type. Every endpoint capable of receiving such a message declares, that it can interpret that message.

Information types are an abstraction of the technical message types to simplify the setting up of routings.
One technical message type can be part of multiple Information types.

Each technical message type must be assigned to an information type, which represents its meaning and purpose. The technical message types Bitmap, PNG and JPEG for example are summarized as Information Type “Image”. This means, that endpoints can send Bitmaps, PNGs and JPEGs, if a routing for Images exists.

Technical message types are defined by groups outside the agrirouter project. If message types are missing, DKE will add a useful standardized message type. DKE however will not standardize any formats.
An information type is a group of technical message types.

The agrirouter message payload has a technical message type, which is indicated on the envelope. The technical message type of the payload could be:

Information Type	Technical Message Type
TaskData message Type	*iso:11783:-10:taskdata:zip* *iso:11783:-10:device_description:protobuf*
EFDI Message Type	*iso:11783:-10:time_log:protobuf* *iso:11783:-10:device_description:protobuf*
Image Message Type	*img :001:jpg* *img:001:png* *img:001:bmp* *001 : Any* *002-999 : Reserved for future assignment*
Video Message Type	vid:*001*:avi vid:*001*:mp4 vid:*001*:wmv *001 : Any* *002-999 : Reserved for future assignment*
Manufacturer Specific Message Type	Specific message formats only known by a manufacturer Message formats in development

The definition of a new technical message type is possible, see 13.5.2 Message formats.

9.7 Capabilities

Each endpoint has to describe, which technical message types it can send and/or receive. The listing of those technical message types is called capabilities. Each endpoint has to provide its capabilities when starting to communicate with the

agrirouter. The capabilities, an endpoint generally supports (e.g. with all optional features enabled) is required for the Certification process; see 6.5 Application certification.

Remark

Some applications are offered with several optional features, that are only available to the end user under specific circumstances, e.g. if he buys a special package. an app instance, that has such optional features should always send those capabilities to the agrirouter, that it really supports under its current configuration. Otherwise, this might lead to data exchange problems.

9.8 Subscriptions

As a sender of a message (e.g. an EFDI Telemetry message) does not always know the relevant endpoints, he can send the message as a public message. Every other endpoint can subscribe to any message type that is part of its capabilities.

Before sending the Subscription message, the endpoint has to send the Capabilities message

The steps for this – in general – are:

1. Endpoint EP3 subscribes for a technical message type(TMT)
2. App Instance 1 of App 1 sends a message of this TMT to its EP1 Inbox
3. agrirouter forwards the message to the feed of Endpoint EP3
4. This message is forwarded to any endpoint that subscribed for that list
5. As EP 3 is subscribed, the message is forwarded to the outbox of EP3
6. The App instance 47 can now pull this message from its endpoints feed

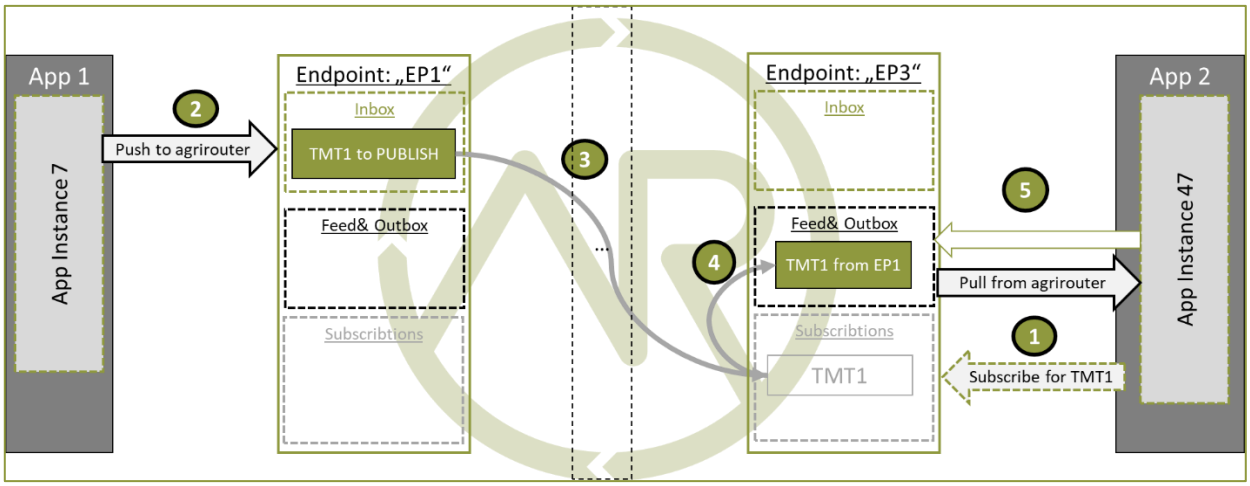

Figure 16 Subscribing for a message; simplified

9.9 Exchange with other user accounts

Messages can be sent to other users' accounts, if this users account is connected with the agrirouter account of the sending end point. A connected account is represented as an endpoint in the users agrirouter account.
A message, that is addressed to this endpoint will be published in the connected account. An app instance, that wants to receive messages from a different account has to subscribe its endpoint to the desired technical message type. The process is visible in the following graph:

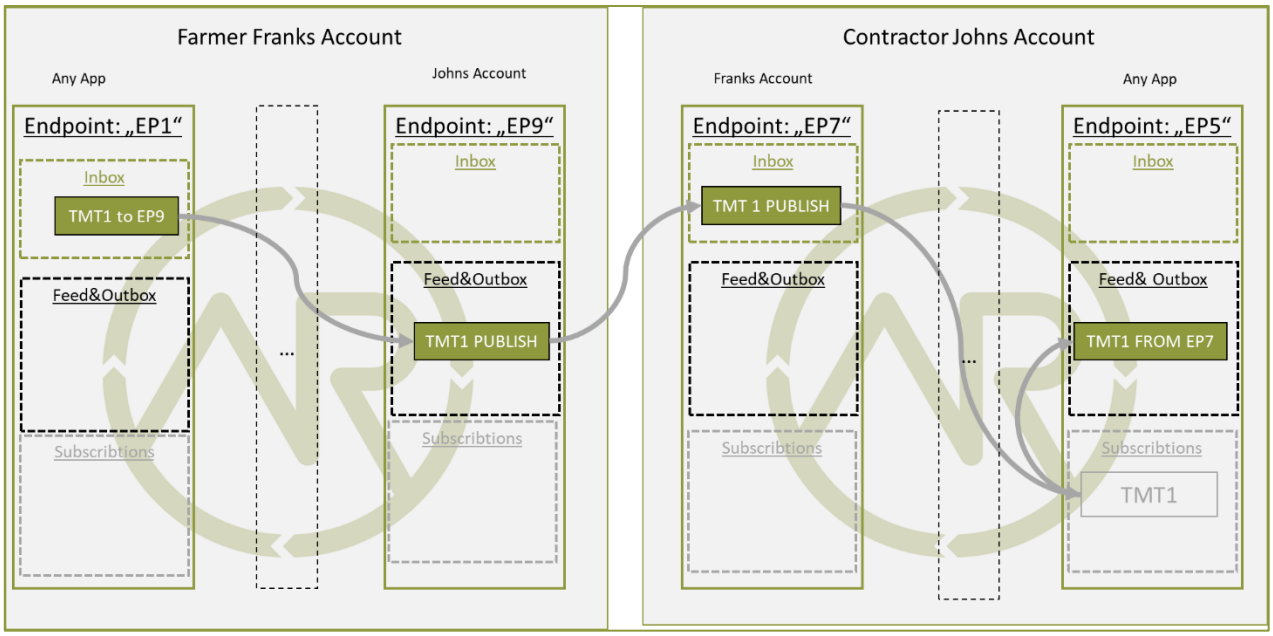

Figure 17 Sending messages into another account

Short description:

App Instance 1 sends a message of TMT1 to the Endpoint EP9. As there is a routing, agrirouter will forward that message. agrirouter recognizes, that EP9 in real is the connection to Contractor Johns account. Therefore, it now handles the message like an incoming message, published in EP 7 of Contractor Johns account. agrirouter looks for endpoints, that are:

- Part of Contractor Johns account
- have an existing Routing with TMT1 and EP7
- Are subscribed for the TMT.

The message will be forwarded to any endpoint matching these criteria.

9.10 Routings

9.10.1 General

To control the message flow between endpoints and to avoid sensitive data being forwarded to the wrong endpoints, end users can setup routings. Only routings, that are allowed by these routings will be performed by the agrirouter.

A routing consists of:

- Sender
- Information Type as abstraction of technical message types
- Receiver

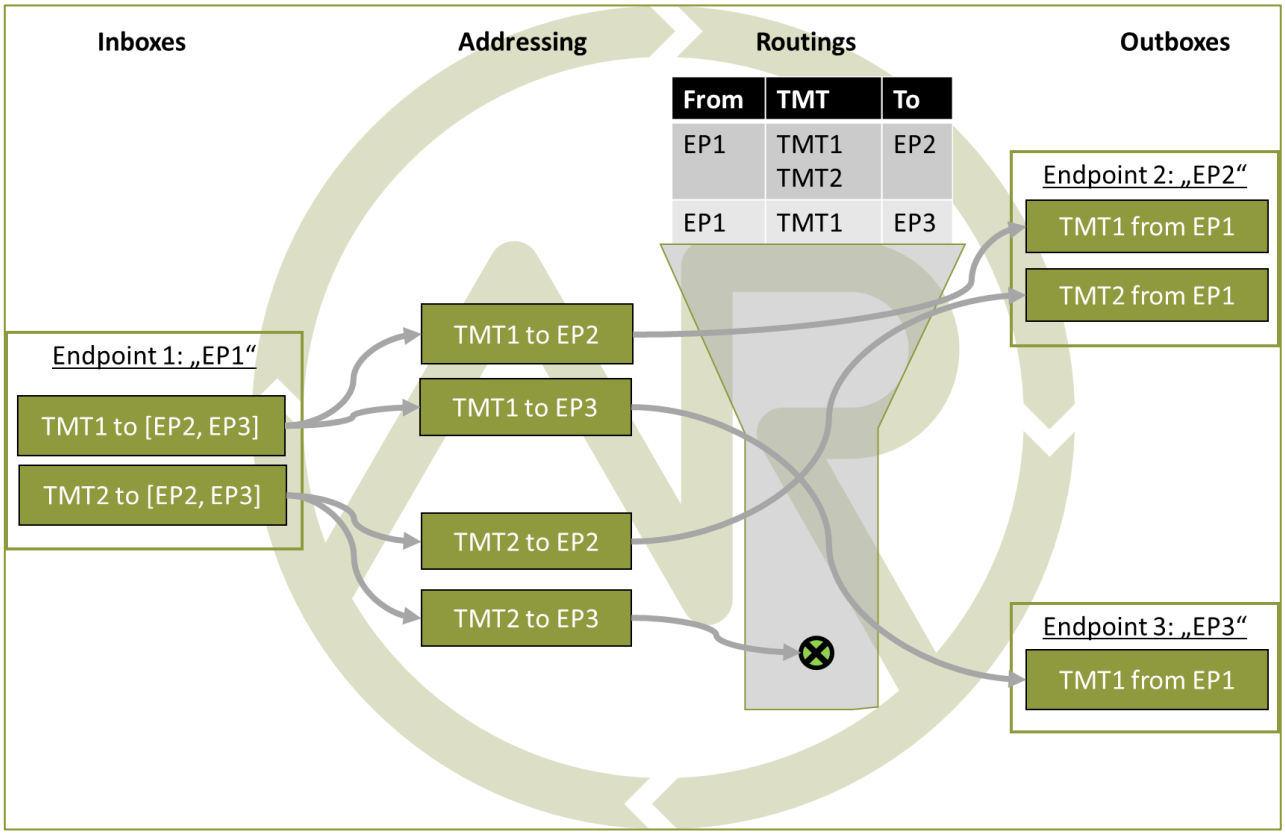

Figure 18 Message addressing and routing through the agrirouter

Short Description:

EP 1 sends 2 messages, each addressed to EP2 and EP3. while both messages of TMT1 receive the destination, because there is a routing given in agrirouter, Only the TMT2 message for EP2 arrives at its destination, because there is no routing given between EP1 and EP3 for TMT2.

9.10.2 Setup of routings

Routings are created by the end user in his agrirouter account.

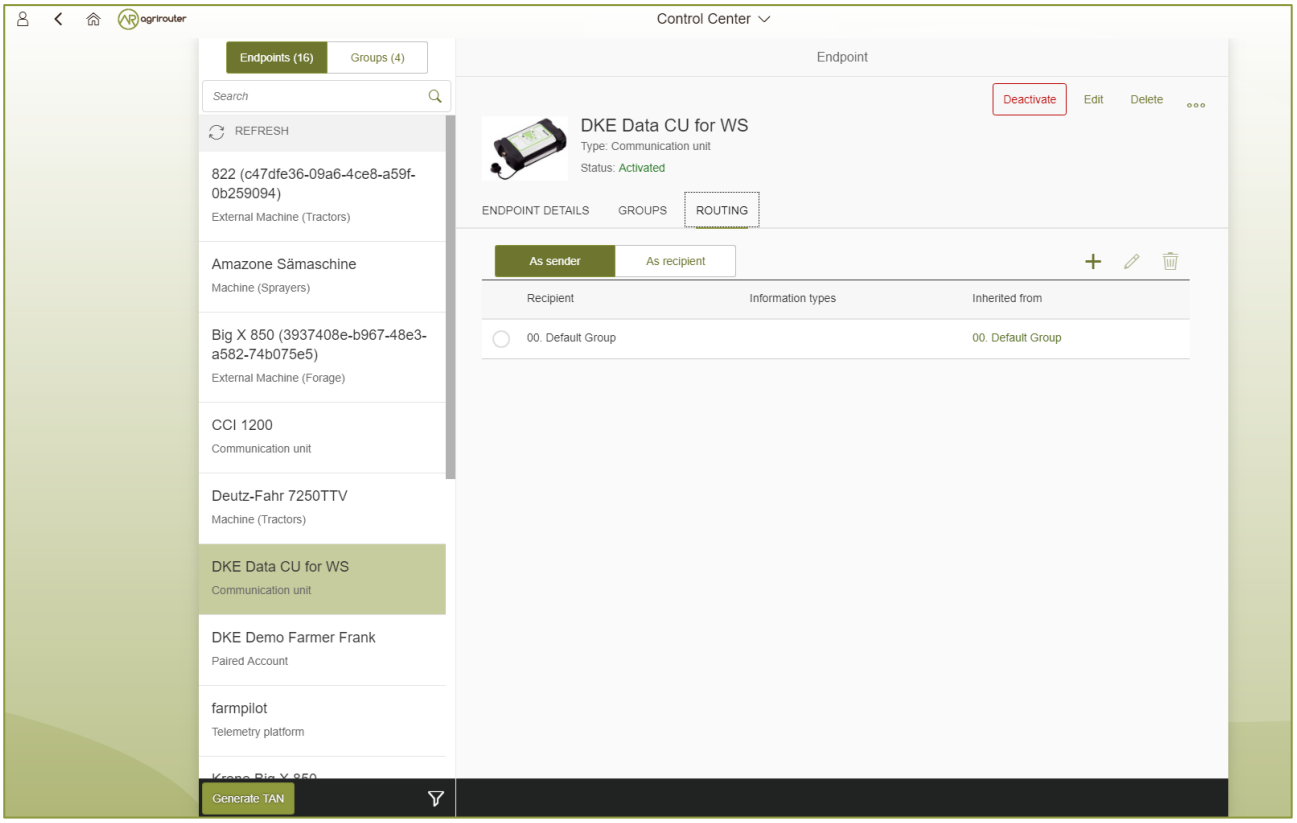

Figure 19 Graphical user interface for message routing

9.10.3 Endpoint Groups

To simplify the creation of routings, endpoints can be grouped in the end user's user interface. This grouping is only done on the visual layer, there is no real endpoint group, that can be addressed.

9.10.4 The default group

Sometimes, new machines are added to the ecosystem, because a (Virtual) CU is connected to a new machine. If this happens, a new endpoint for that machine is created in the end users agrirouter account. As the end user is not always logged in to setup routings, he can just setup routings for this – not deletable – default group. Every new endpoint is automatically assigned to the default group. any endpoint can be removed from that group, the default group itself however cannot be deleted.

9.11 Routing of telemetry data

For most information types, agrirouter does not read but only route the messages from the inbox of the sending endpoint to the feed(s) of the target endpoint(s). For telemetric data, this is different. The agrirouter does still **not** read the values, but it can setup routings based on the purpose of the values context.

9.11.1 DDIs

Telematic sensor values are assigned to DDIs, a standardized list of possible sensor and task information. A full list can be found here: https://dictionary.isobus.net.

The TimeLogs message includes a list of DDI numbers and their corresponding current values. The agrirouter can filter timelog messages for DDIs, that are part of a routing. The DDI list provides a number area for proprietary DDIs, that can have a specific definition for each manufacturer.

9.11.2 Categories

As the list of standardized DDIs is quite long (more than 530 by May 2018), agrirouter provides DDI categories, to simplify the routings setup for end users. Each category stands for multiple DDIs. One DDI can only be in one category.
An end user sets up routings using Categories.

The current List of Categories includes the following Categories:

No.	Name	Description
0	**GPS Geo Position**	GPS Geo-Position (North and East Coordinates) where the telemetry data was measured or logged.
1	**Guidance and Geo Data**	Data related to geographical and guidance information
2	**Application Data**	Data related what is applied to the field (e.g. fertilizer, seeds, plant protection, dry matter, …)
3	**Crop and Yield Data**	Properties of harvested material
4	**General Work Data**	Task and Lifetime Counter or average values (Counters that are not relevant for Application and or yield)
5	**Fuel and Exhaust Fluid Consumption Data**	Data related what a machine consumpt of fuel and Exhaust Fluid (Energy overall)
6	**Process Data**	Data related to the main working process of the machine
7	**Machine Data**	Data related to the machine characteristics (not process relevant)
8	**Environment Data**	Data related to the Environment (weather data)
9	**Basic Data**	Fundamental values that are relevant for the whole system
10	**Proprietary Data**	Manufacturer specific data (not part of the standard)

9.11.3 Technical Message Types

9.11.3.1 EFDI Device Descriptions

The message iso:11783:-10:device_description:protobuf is used to understand, which machine is attached to which CU or Virtual CU and which machine provides which DDI. EFDI Device Descriptions are protobuf encoded, the data structure however is mainly equal to the form of an ISO11783-10 TaskData.

9.11.3.2 EFDI Time Logs

The Message iso:11783:-10:time_log:protobuf includes a list of live telemetry data. For those live telemetry data, agrirouter is able to select only those DDIs to be forwarded to an endpoint, that were selected by the end user when creating routings. EFDI Time Logs are as well comparable to ISO11783 TimeLogs.

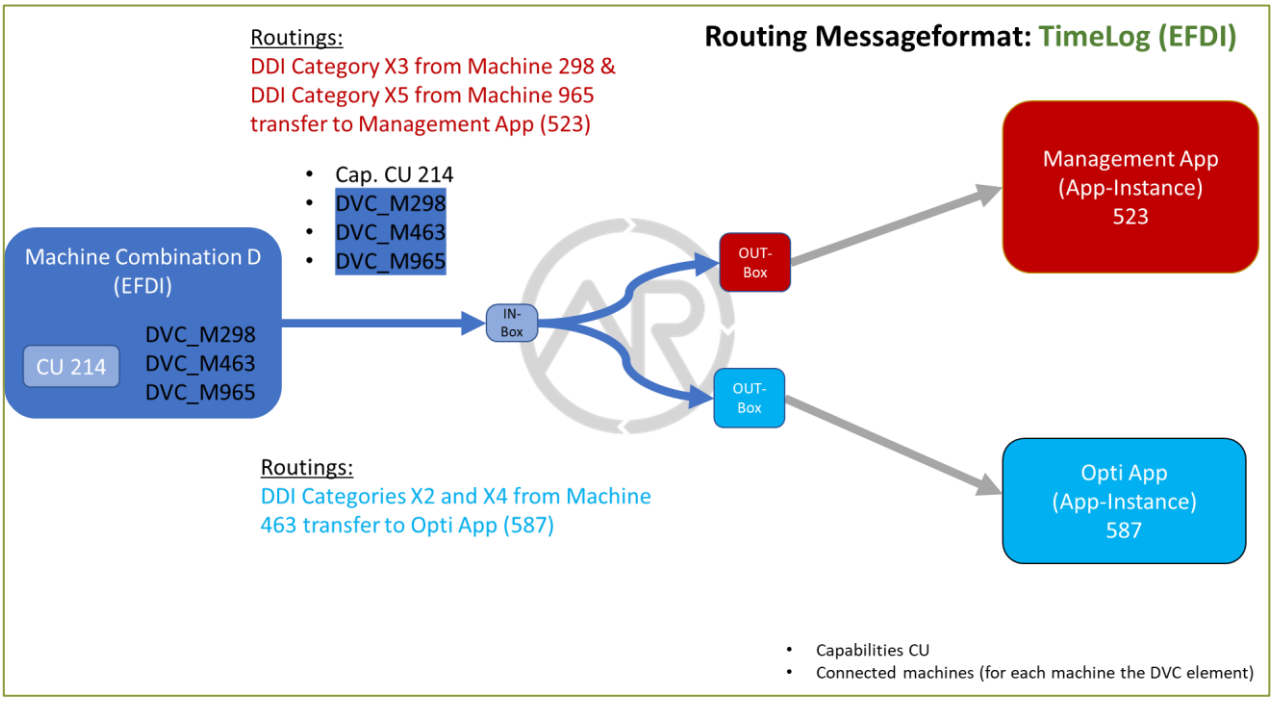

Figure 20 Message delivery due to routings, subscription and public address

Important note:

The format for telemetry data; EFDI is defined by a subgroup of the AEF. The documentation of this format is currently intellectual property of the AEF. For further information on those message types, please refer to https://aef-online.org . DKE is currently not able to provide the documentation for EFDI.

9.12 Condensed Messaging Workflow

The following flow charts show message flow in different complexity

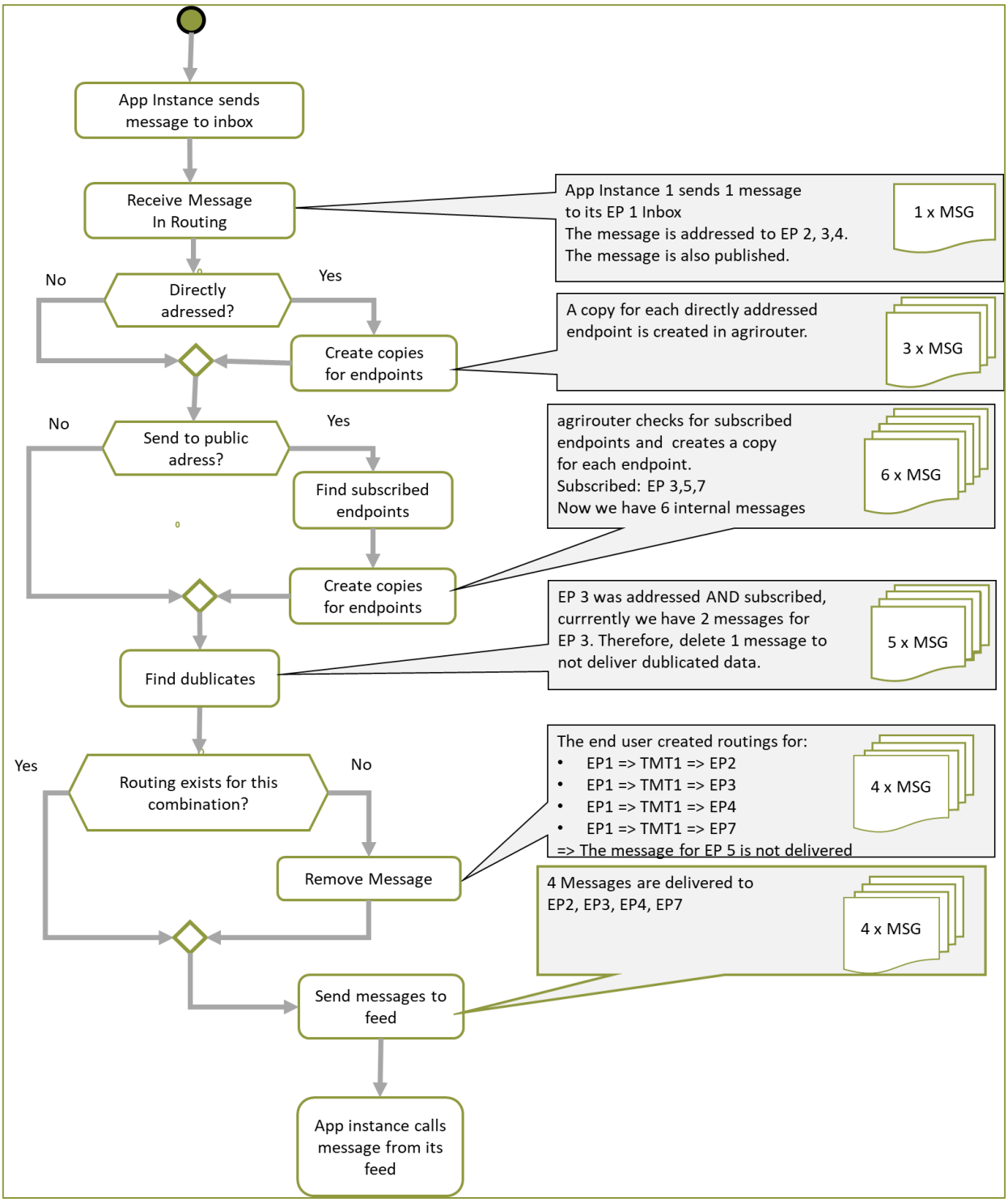

Figure 21 Condensed workflow of message routing

10 Application Interaction with agrirouter

10.1 Terms and short description

This chapter will introduce and declare multiple new terms. The following text gives a short overview of what you will learn in this chapter and how the terms connect to each other. After reading the whole chapter, the following sentences should make good sense to you.

App Providers can use an authorization process, to assign endpoints and users of application instances.
Any App Instance has to perform onboarding to create an endpoint in an agrirouter account.
For onboarding, the app instance has to provide a TAN for assignment to the end users account.
The authorization process can be used to receive a TAN.
A TAN can alternatively be provided by the user interface of the agrirouter for CUs.
Telemetry Platforms can onboard their own Virtual CUs.

After onboarding, each app instance can communicate with its endpoint using REST or MQTT.
App instances using REST send requests and receive responses from their inbox or outbox.
App instances using MQTT send requests and receive responses from their inbox or outbox.
Using the desired protocol, App Instances send commands and messages to their inbox.
The response for a request to the inbox buffer will be the information, that the command or message is being processed.
App Instances using the REST protocol will have to poll for a result of this processing at the outbox.
App instances using the MQTT protocol will only receive the result without polling or confirmation.
The result of the reception of the command or message will be the response of the inbox.
Messages, that are no commands for the agrirouter will be forwarded to addressing and routing.
Commands will be processed by the agrirouter.
If a command has a result, this result will be placed in the outbox.
An app instance uses commands to call for information.
App Instances call for messages from their feed by sending a command to their inbox.
The agrirouter will than forward the desired messages from the endpoints feed to its outbox.
App Instances can call for a filtered header list of available messages.
A message containing a list of message headers will then be delivered to the outbox.
An app instance can call for a list of endpoints, that can receive a specific technical message format.
A list of endpoints will then be delivered to the outbox.

The new terms will be described in more detail in the following paragraphs.

10.2 General Overview

A communication with agrirouter requires several, partly parallel actions to take. The following Figure shows one possible workflow.

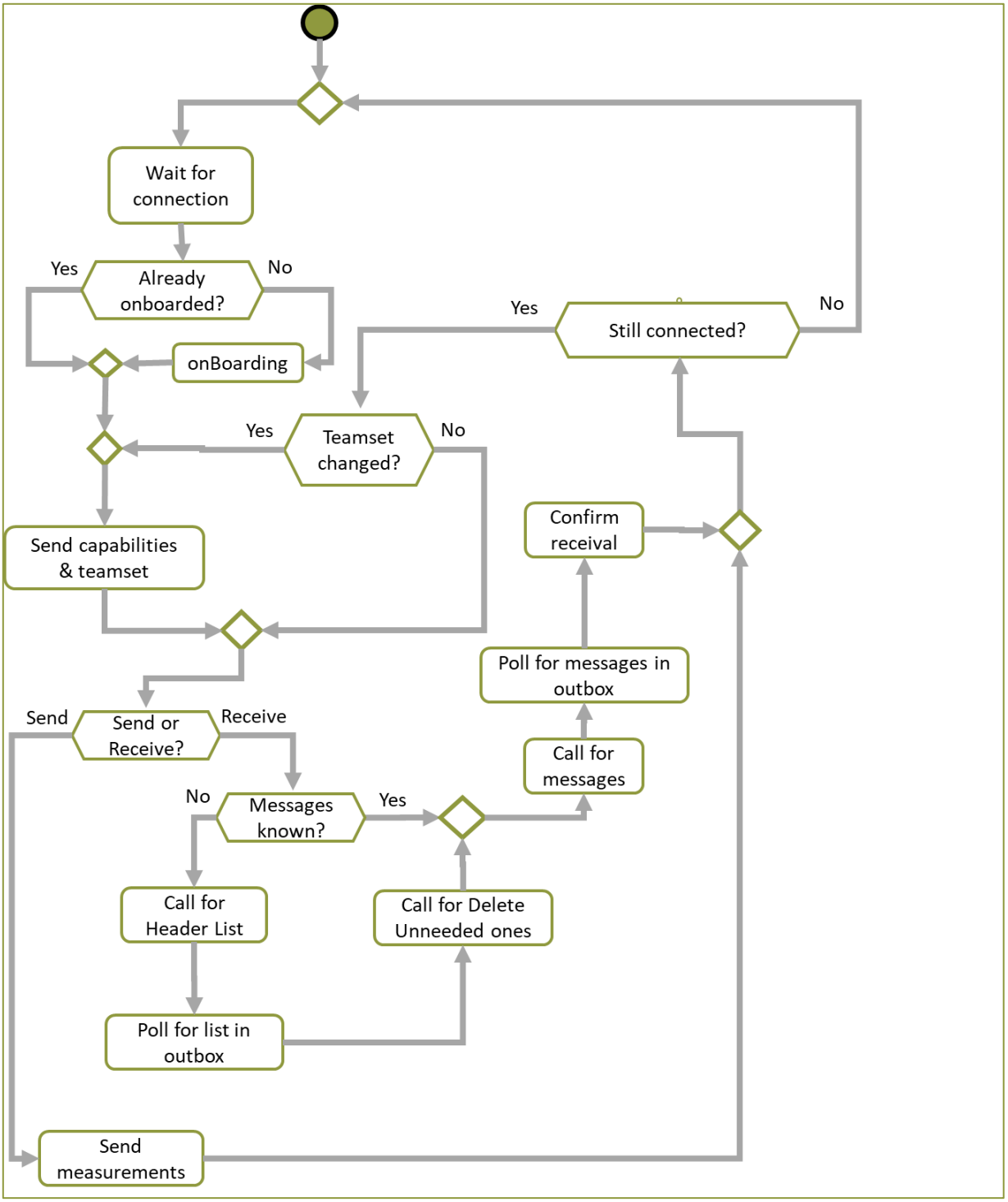

Figure 22 General messaging workflow

10.2.1 Communication Protocols

The agrirouter offers two different communication protocols for the common communication:

- REST via HTTP(S)
- MQTT

The commands exchanged with the agrirouter are protobuf encoded (in both directions). MQTT clients MUST send commands as base64 encoded strings embedded into a JSON frame structure.

REST based clients have the choice to transfer the commands in the same format as required for MQTT (base64 encoded protobuf nested in JSON structure). For REST, it is also possible to send native protobuf.

The advantage of using MQTT is that polling for message delivery (see following chapters) will not be needed. REST however has to be used anyway for the registration of the endpoint. Therefore, applications have to be able to do REST communication anyway.

The internal Google Protocol Buffer (protobuf) definitions for the communication with the agrirouter are available in GitHub repository under following URL: https://github.com/DKE-Dataagrirouter Endpoints

This chapter gives an overview about the endpoints including their base requirement and definition. This chapter gives a high-level overview about the agrirouter Connectivity-Platform architecture, an application can interact with.

10.2.2 Overview of the architecture

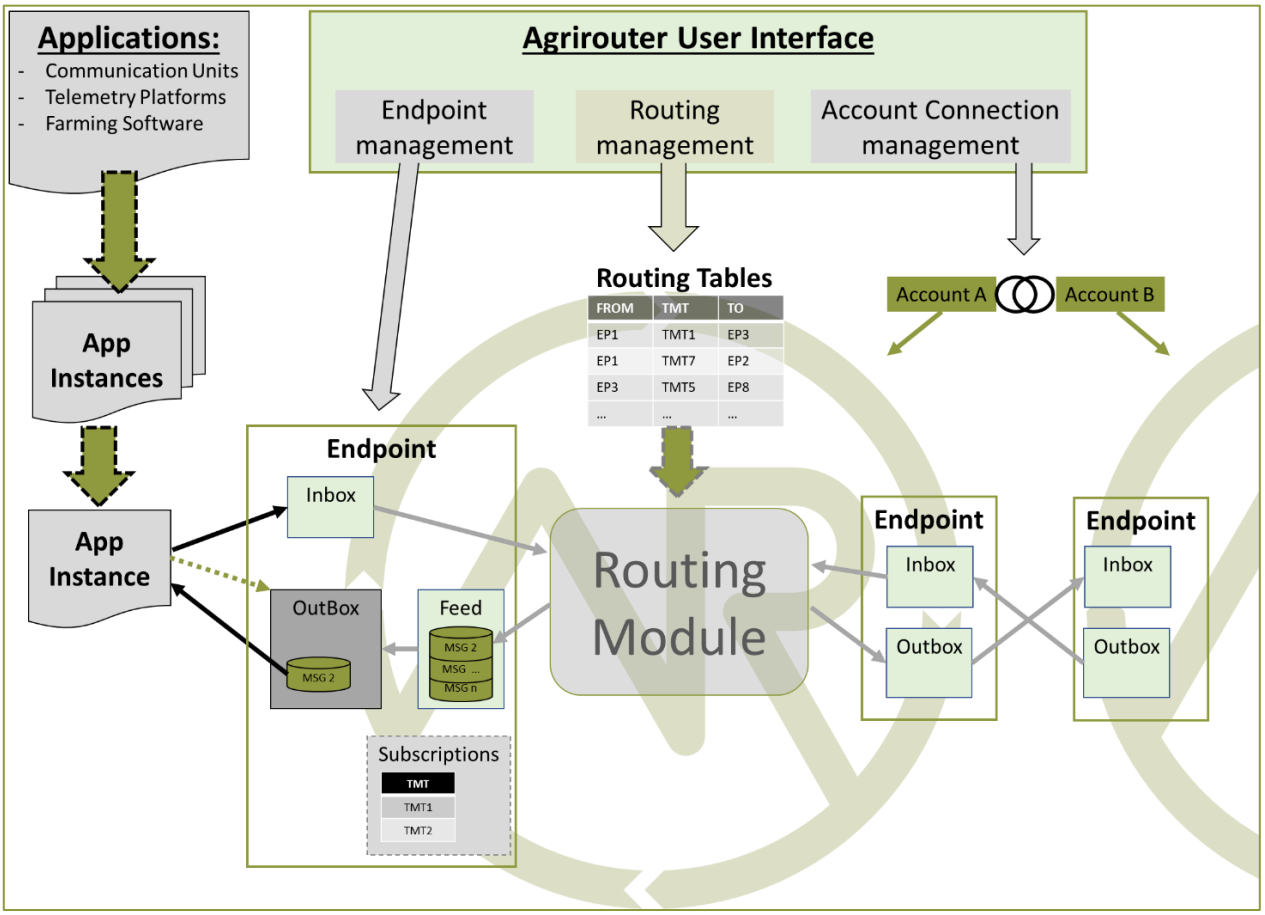

Figure 23 agrirouter Connectivity-Platform IT architecture

10.3 Authorization process for endpoints

Important note:

App providers, whose software receives data have to pay for these data. The authorization process is used by application providers to assign each endpoint with a known user of their software.
The authorization process is major for farming software and telemetry platforms, for CUs it is an optional function.

For CUs, this requires additional infrastructure, DKE advices to use the TAN exchange process.

10.3.1 Account authorization for farming software

As application providers of farming software have to pay for the consumption of raw data, the application provider should make sure, that only such agrirouter accounts can onboard one of his applications, that he can assign to one of his users. Otherwise, fake accounts could consume data on his costs.

10.3.1.1 Authorization Process Overview

The authorization process works as follows:

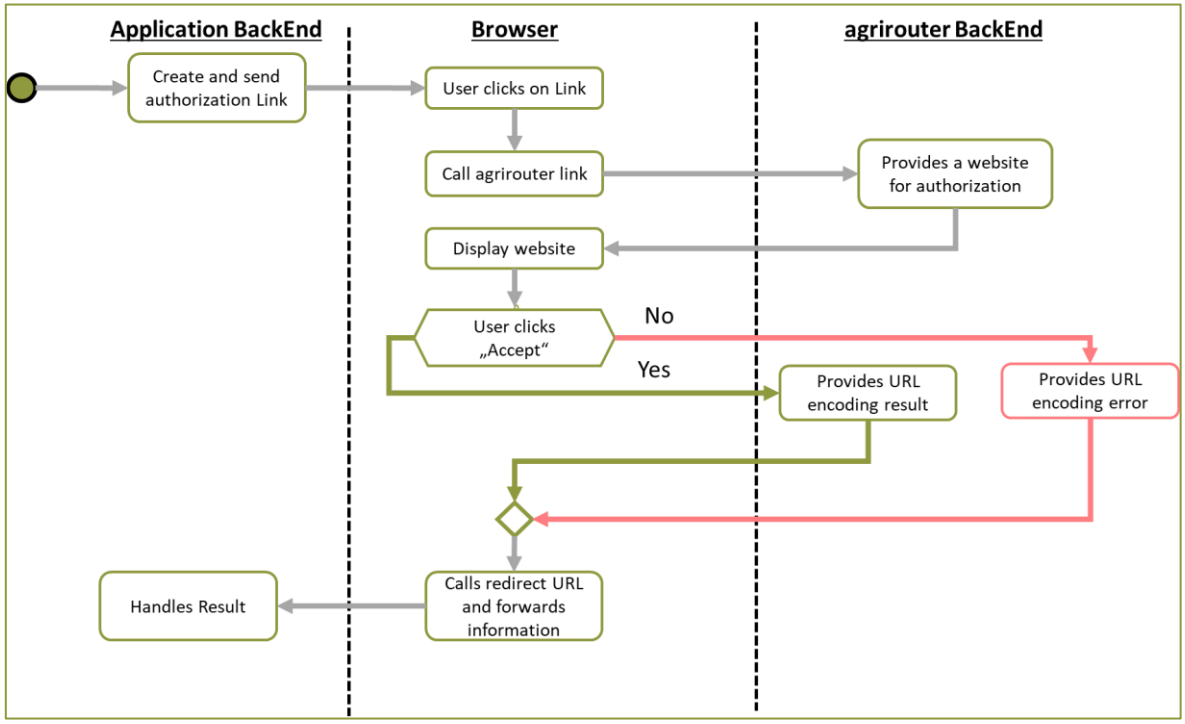

Figure 24 Process of app authorization

To better understand, what happens here, try the following:
1. Call https://httpbin.org/get in your browser. You'll get a JSON view of the get request
2. Call https://httpbin.org/get?Param1=Value1&Param2=Value2 in your browser. You'll get a view of the get request
   ⇨ https://httpbin.org simply echoes the request that is send to the page. That's important to understand

10.3.1.2 Generating an authorization Link

To provide a link for authorization, create a link like this:

{{agrirouter-url}}/application/{{applicationID}}/authorize?{{response-type}}&{{state}}&{{redirectURL}}

Remark: The detailed request is described in Integration Guide Part 2 (see Purpose of further documents

10.3.1.3 Perform authorization

When the user clicks on the link, the agrirouter website is called. If the user is currently not logged in, he has to log in. After logging in, he is delivered a website to authorize the connection between agrirouter and the application provider:

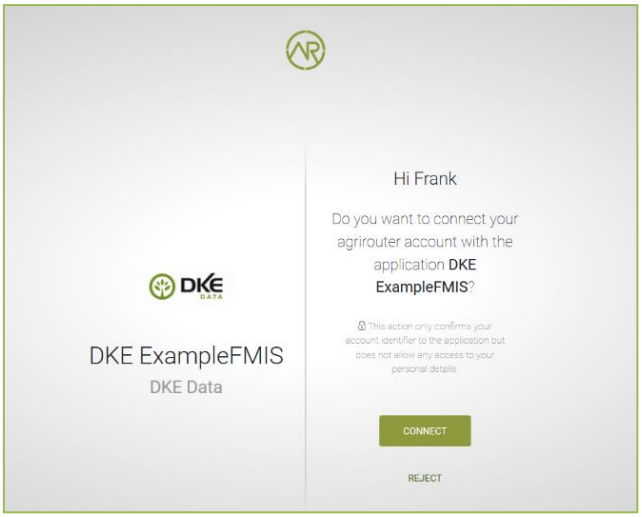

Figure 25 Application authorization screen

10.3.1.4 Analyses result

agrirouter sends an HTTP 301 redirect to the browser, encoding the authorization result in the get queue.


```
{
  - args: {
        signature:
        "H45ZxgsWJZzYZgyBCrxnUAKpdDKuJy0BwXYztenNZI73+Rxb/pFhtc6x7YUAgMAQE1qXV0mM9hrXpXwEhdqr2l+NRAEY9
        state: "b862103e-7689-4a7a-9cab-f60c98448215",
        token: "eyJhY2NvdW50IjoiNTA5ZTY1YWEtODRkZi00YzRkLWJmOTgtYzMyOGYwODQ2NGM1IiwicmVnY29kZSI6ImQ2ND
    },
  - headers: {
        Accept: "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
        Accept-Encoding: "gzip, deflate",
        Accept-Language: "en,de-DE;q=0.9,de;q=0.8,en-US;q=0.7",
        Cache-Control: "max-age=0",
        Connection: "close",
        Host: "httpbin.org",
        Upgrade-Insecure-Requests: "1",
        User-Agent: "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
    },
    origin: "88.130.2.214",
    url: "http://httpbin.org/get?state=b862103e-7689-4a7a-9cab-
    f60c98448215&token=eyJhY2NvdW50IjoiNTA5ZTY1YWEtODRkZi00YzRkLWJmOTgtYzMyOGYwODQ2NGM1IiwicmVnY29kZSI
}
```


Figure 26 Example of an authorization result

The details of this answer are described in Part 2 of the integration Guide.

10.3.2 Authorization for CUs and non-cloud-software

To perform authorization for software, that is not provided as a cloud solution, a small cloud onboarding service could be created to handle the onboarding communication:

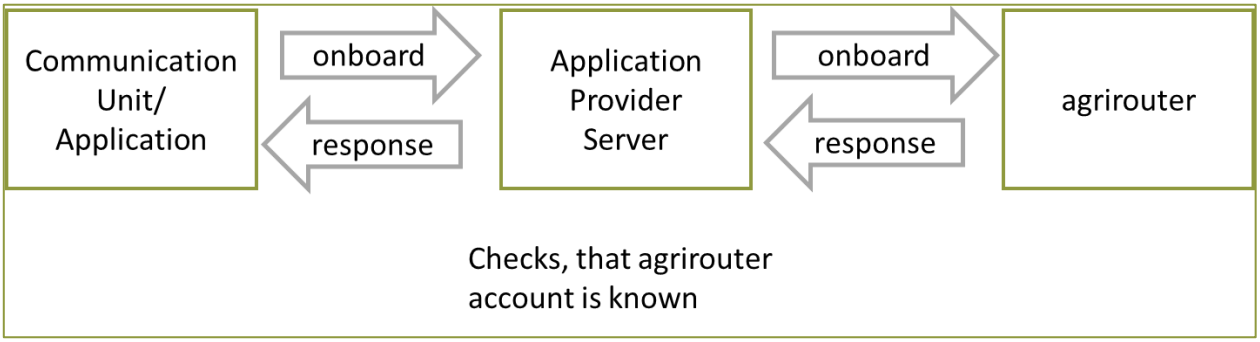

Figure 27 architecture for CU and non-cloud application authorization

This process will be described more precisely in Integration Guide Part 2.

10.4 Onboarding Process

10.4.1 General work flow

10.4.1.1 For simple CU onboarding

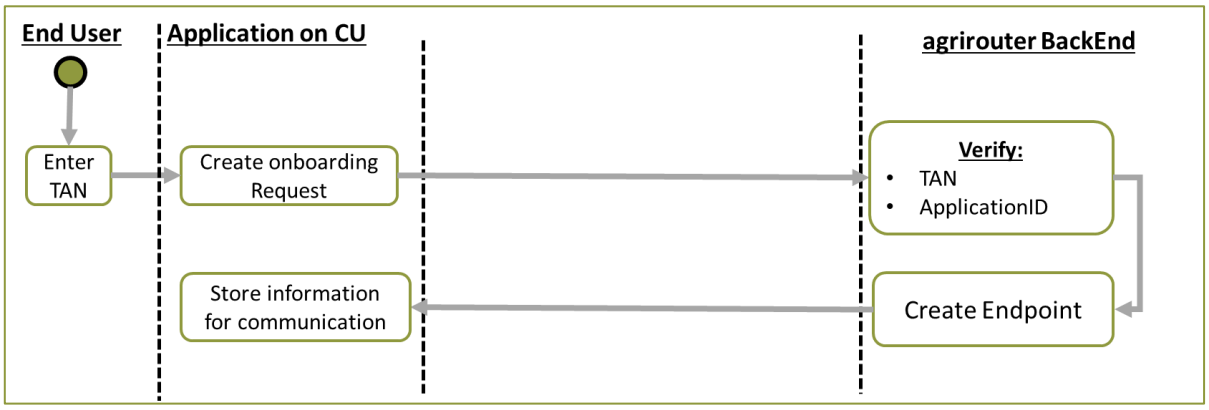

Figure 28 Process of simple CU onboarding

10.4.1.2 For authenticated CU onboarding

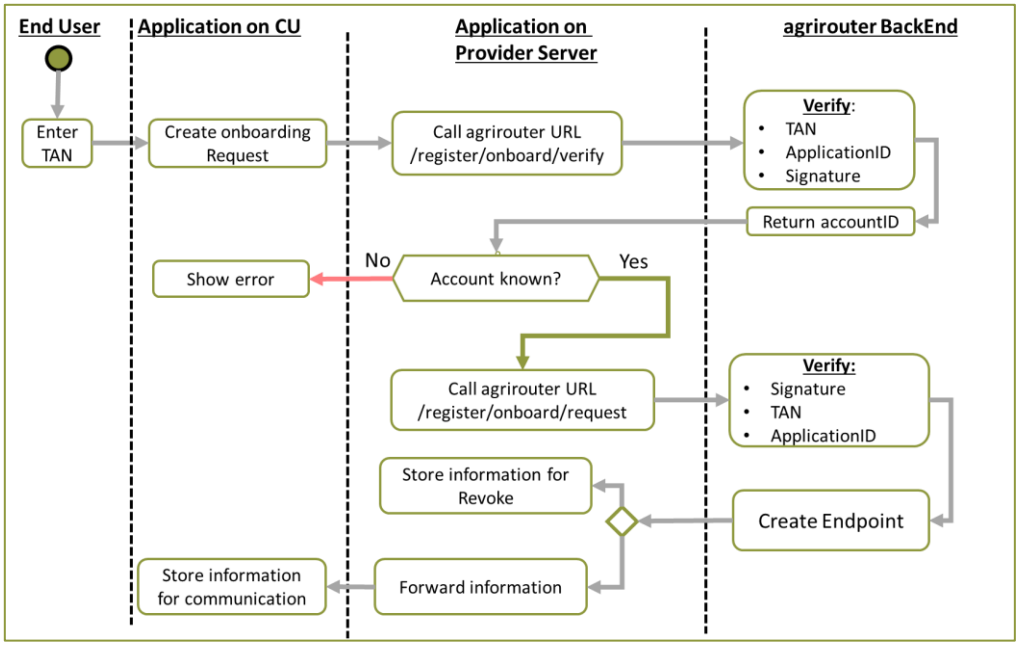

Figure 29 General workflow for onboarding with user interaction and authorization

10.4.1.3 For Farming Software or Telemetry Platform onboarding

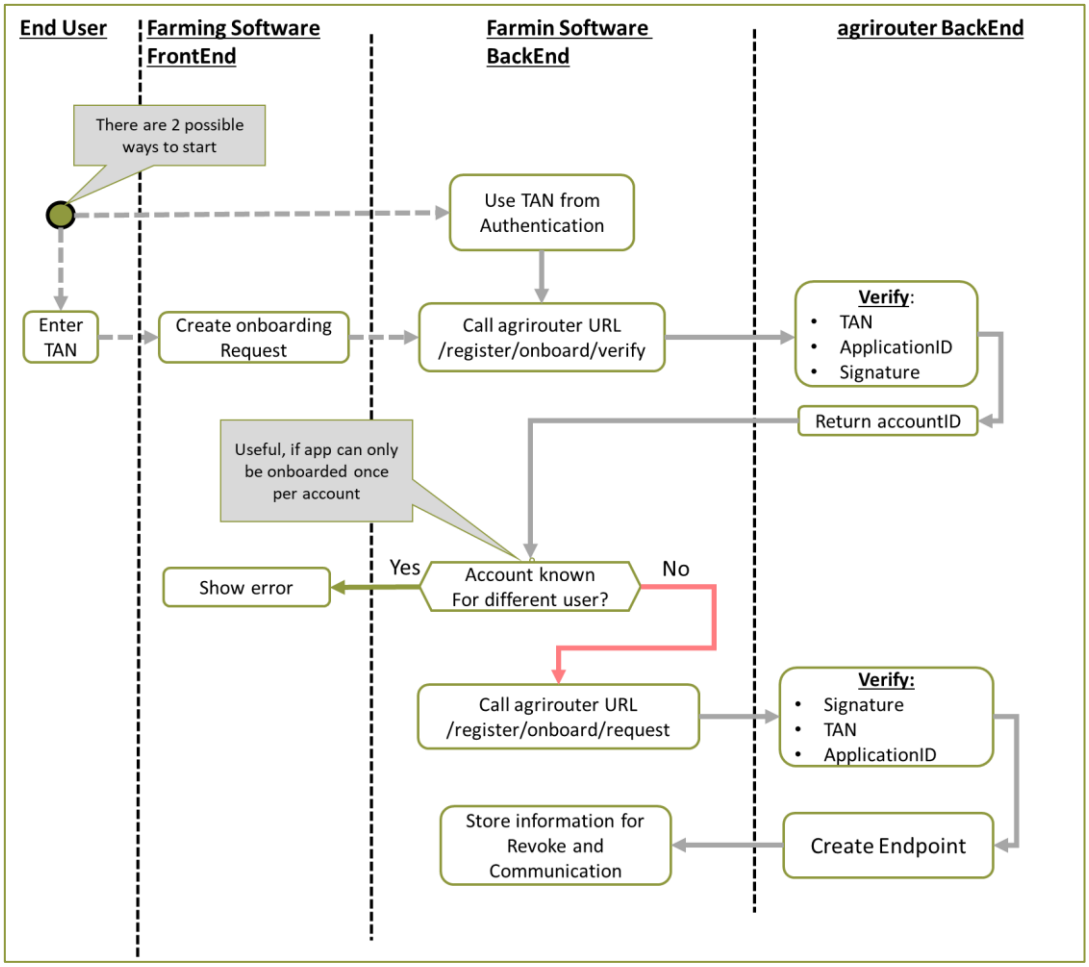

Figure 30 Onboarding process of a Farming Software or Telemetry platform

10.4.2 Creating a registration code (TAN)

10.4.2.1 For CUs

The TAN for a new CU can be created by the end user clicking “Generate TAN”(1) in agrirouters’ control center. He has to select the desired CU(2) and gets a 10-digit code consisting of letters and numbers(3). A CU needs an interface to input this registration code.

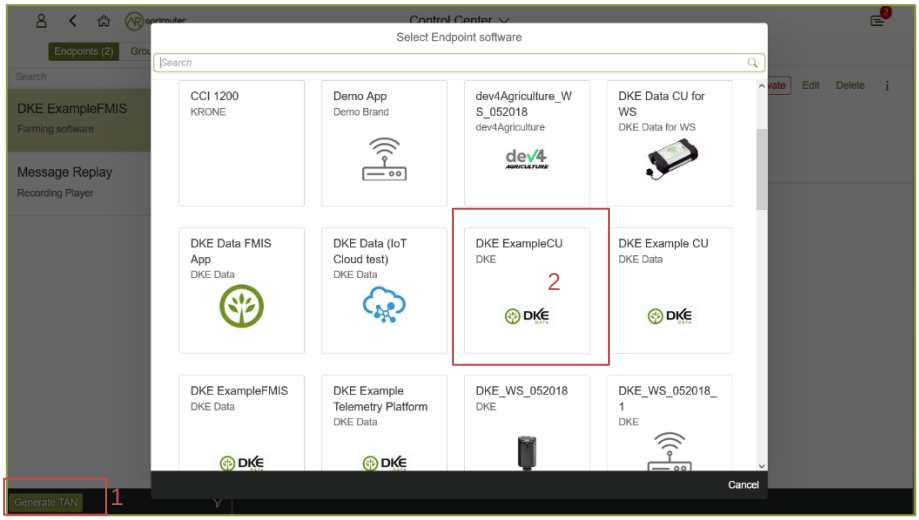

Figure 31 Requesting a registration code in agrirouter UI

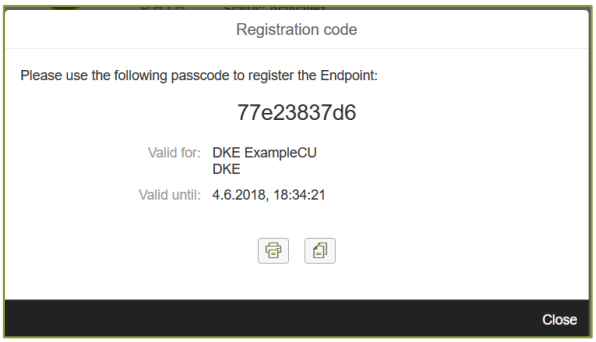

Figure 32 Registration code for a CU

10.4.2.2 For applications

Applications can either receive a TAN like CUs or as part of the authorization process, see 10.3 Authorization process for endpoints

10.4.2.3 For virtual endpoints of telemetry platforms

Virtual CUs can be onboarded by their telemetry platform, a TAN generation is not required.

10.4.3 Onboarding Request

To onboard a new endpoint, the endpoint has to send an onboarding request providing the TAN to agrirouter.
As a result, the agrirouter will return a JSON object including the endpoint IDs and the certificates required for the further communication with agrirouter. The onboarding request shall only be done once per Application Instance. The used unique app instance ID however should be stored, as it is required for reonboarding.

Remark: There is no MQTT onboarding mechanism, so onboarding always has to be done using REST.

An example for an onboarding request can be found in the postman collection, which is part of Integration Guide Part 2.

10.5 Reonboarding

Whenever an app instance receives an update of its software, it possibly needs to reonboard to agrirouter to receive its certificates and the endpoint IDs again. Reonboarding equals the onboarding request. Important is, that the same app instance id is used again, so that agrirouter recognizes the reonboarding and the routings and group settings of the endpoint persist. Otherwise, it would create a new endpoint.

10.6 Communication and Message exchange

10.6.1 The onion principle of commands and messages

agrirouter is a platform, that is mostly used to transport messages **through** it and not towards it.
Therefore, the content is encapsulated
in messages, which are encapsulated into
the commands to the agrirouter, which are encapsulated into
the protocol layer of REST or MQTT.

With except to the EFDI telemetry messages (DeviceDescription and timelogs), agrirouter doesn't analyse the messages inside the agrirouter command. It just checks the message type and the addressing to determine the recipients based on the routings and subscriptions.

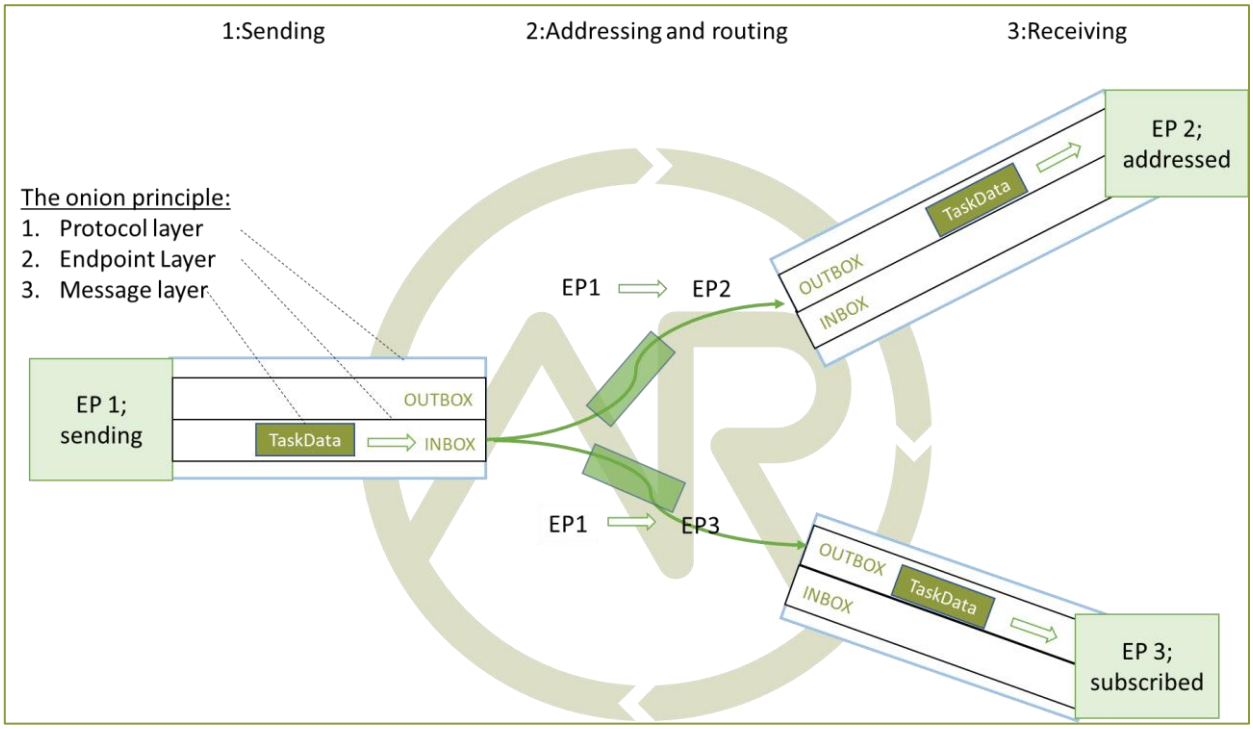

Figure 33 The onion principal for a non-telemetry message

An exception are the EFDI messages.
The device descriptions are needed by the agrirouter to

- determine the relevant CU when sending a Message directly to a machine
- filter for DDIs that are allowed to be sent to specific endpoints

The timelog telemetry data is analyzed, so that a filtering for value categories like fuel consumption is possible.

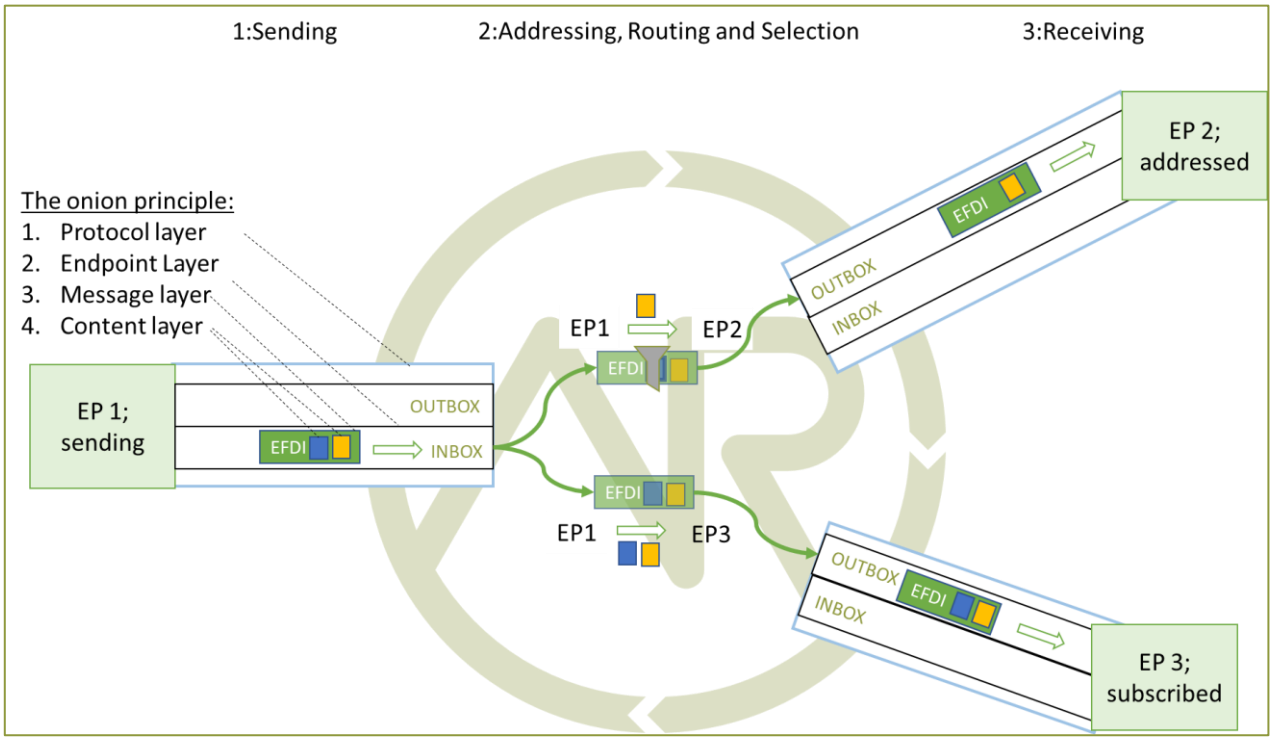

Figure 34 The onion principle for a telemetry message

10.6.2 Sending messages

An endpoint can send messages to other endpoints via the agrirouter. Therefore, it has to create commands to call at the agrirouter. To send a message to agrirouter, the application has to encode the involved file format,

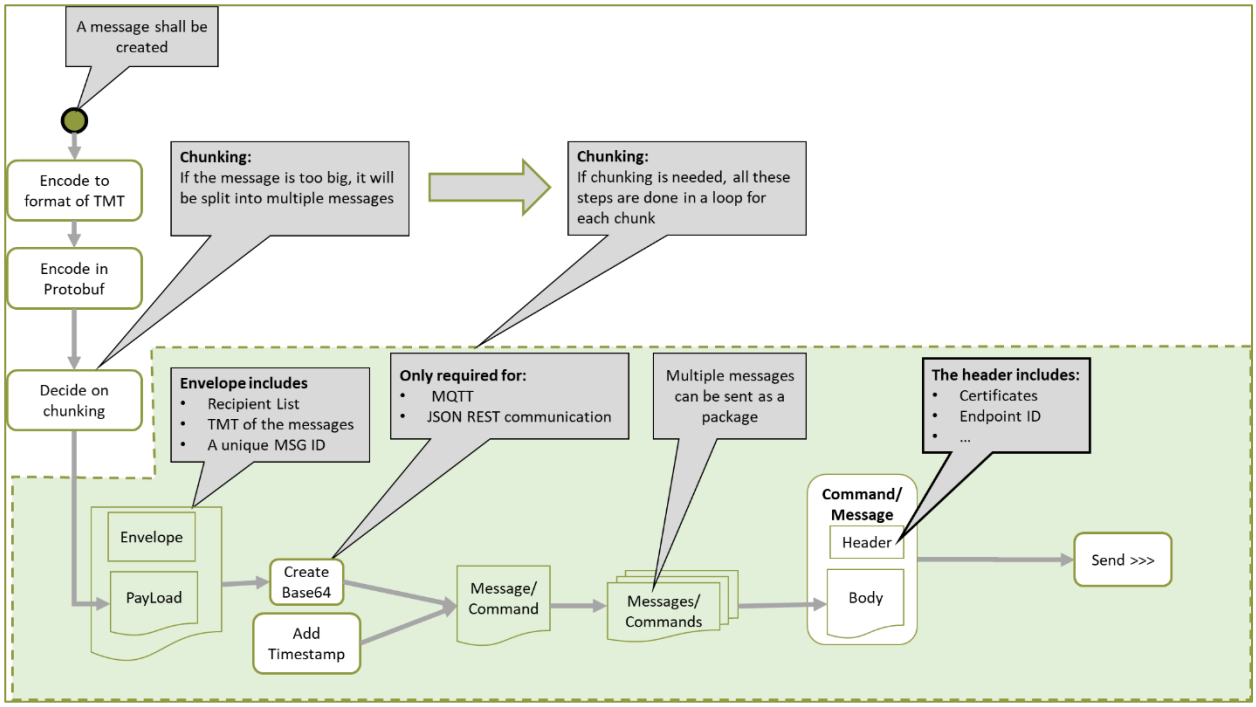

Figure 35 Required Data to create a message

The message is answered by the agrirouter with a HTTP status, that might be 200 (OK). In that case, the endpoint can poll for a confirmation at the outbox before sending the next message.

agrirouter commands like the request for messages are Protobuf-Encoded.

10.7 Requesting messages

10.7.1 General

agrirouter stores messages of an endpoint in its feed. This feed is part of the message module. Requesting a message from the agrirouter equals sending a command to the inbox, requesting agrirouter to forward the message to the outbox. The endpoint can than receive this forwarded message through the selected protocol. For HTTP REST, the endpoint has to poll at the address of the outbox. If the endpoint uses the MQTT protocol, it will receive the message from the agrirouter.

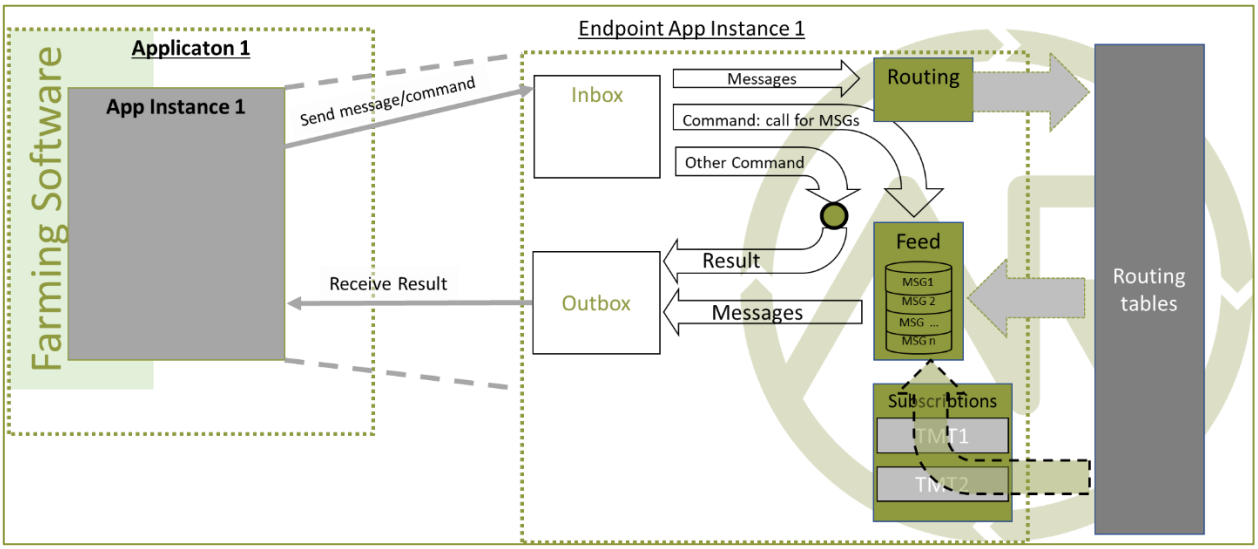

Figure 36 Handling of messages and commands inside agrirouter endpoint; Simplified

10.7.2 Requesting available messages

To receive a list of all buffered messages from the feed of the endpoint, the application has to request the list of available messages. agrirouter will than forward a message header list to the outbox. The app instance can request each message by its unique ID afterwards. It's up to the developer to decide on a useful order of requesting the buffered messages. Ann application could for example use the principles of *last in first out* or *first in first out* or priories specific technical message formats. It would also be possible for timelogs to request every n-th message and requests the messages in between afterwards, so that the app could create a graph, that is getting more and more precise (like a preview becoming a real view).

10.7.3 Relevant Message formats

The message formats relevant to control the agrirouter are available on the GitHub: https://github.com/DKE-Data/agrirouter-api-protobuf-definitions . A full documentation of those messages can be found in the Integration Guide Part 2.
More relevant message formats can be found in the resources List of this document.

10.7.4 Resumable transfer for binary content

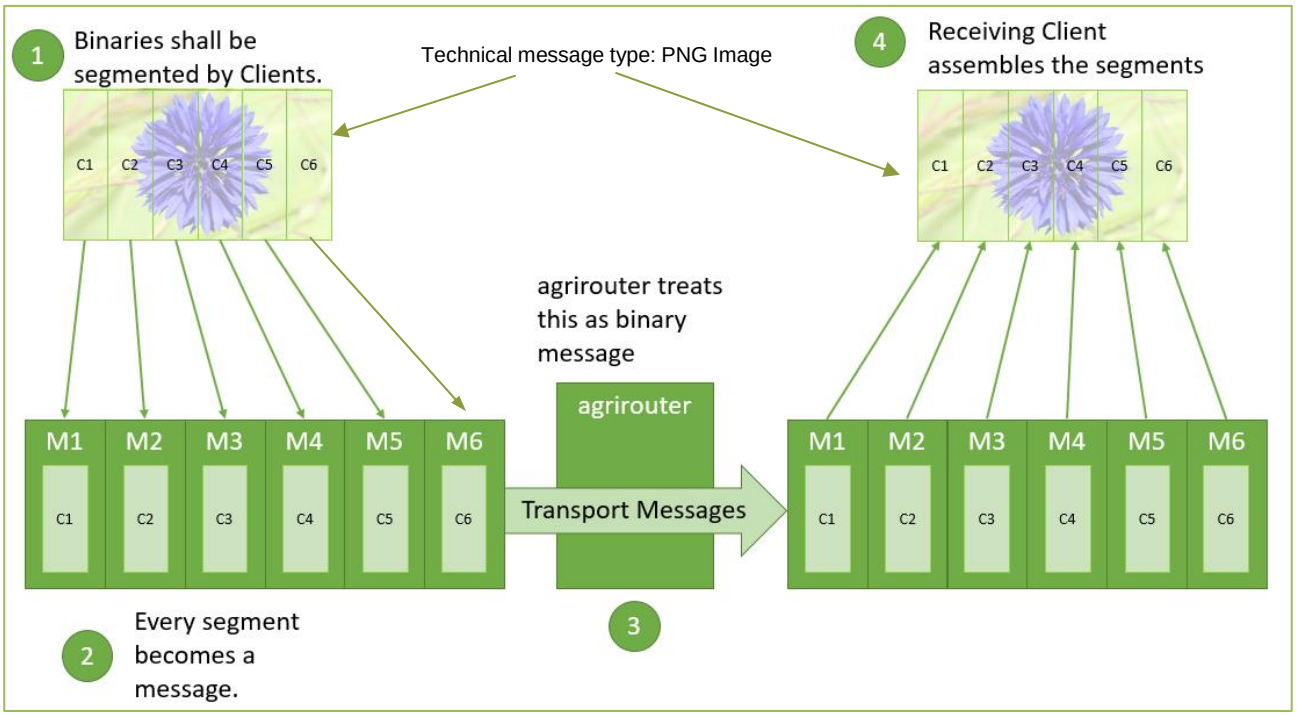

Figure 37 Transfer for Binary Content

Regarding large data, endpoints need to implement a concept for chunked data transfer to ensure a resumable transfer of (binary) content. The endpoint is in charge of the segmentation, this is not done by the agrirouter. The agrirouter is in charge of the delivery and order of the messages.

If the app instance intends to send binary content to its endpoint, the app instance segments the binary content into smaller chunks of a determined size of at maximum 1 MB. Afterwards the client envelopes these chunks into the agrirouter message protocol and sends those chunks each as a single message to the agrirouter. The client has to ensure that the binary relevant header information in the agrirouter message header is filled correctly. The agrirouter transports the messages without changing the content to the feed of the receiver and the client receives those chunks message by message and then can reassemble the binary content again.

Important note:

The segmentation of message does not apply to the telemetry related data as it is described in EFDI.
The max chunk size is capped at 1MB. This size might be decreased during the project for performance reasons.

10.8 Telemetry platform integrations

To simplify and optimize the connection of telemetry platforms, there are several additional functionalities for such platforms. A telemetry platform can onboard virtual CUs itself, so that it is not necessary to enter a TAN in the terminal of a virtual CU

10.8.1 Integration Concept

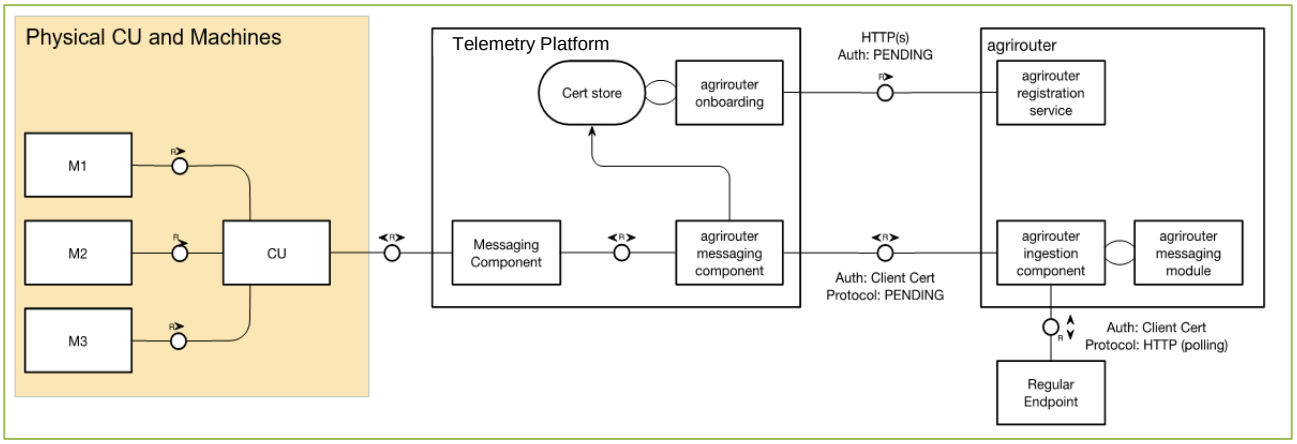

Figure 38 Telemetry platform Integration Concept Overview

10.8.2 Device Onboarding

A certified telemetry platform (having received a certificate by DKE) can onboard new virtual CUs in an automated way. The agrirouter provides a designated API for automated virtual CU onboarding.

10.8.3 Virtual CU and Messaging

Regarding messaging, there is no difference between a real-world CU and a virtual CU onboarded by a certified telemetry platform. The connecting telemetry platform has to implement the same communication pattern as it applies to all other endpoints categories.

Important note:

There will be a trusted Relationship between the agrirouter and certified telemetry platform. The communication between the telemetry platform and the agrirouter can be realized using HTTP(S) based REST communication or MQTT messaging over a designated messaging component accepting both protocols. As mentioned before the payload has to be Google Protocol Buffer encoded, the communication pattern will be asynchronous most of the time.

10.8.4 Levels of the Cloud Solution

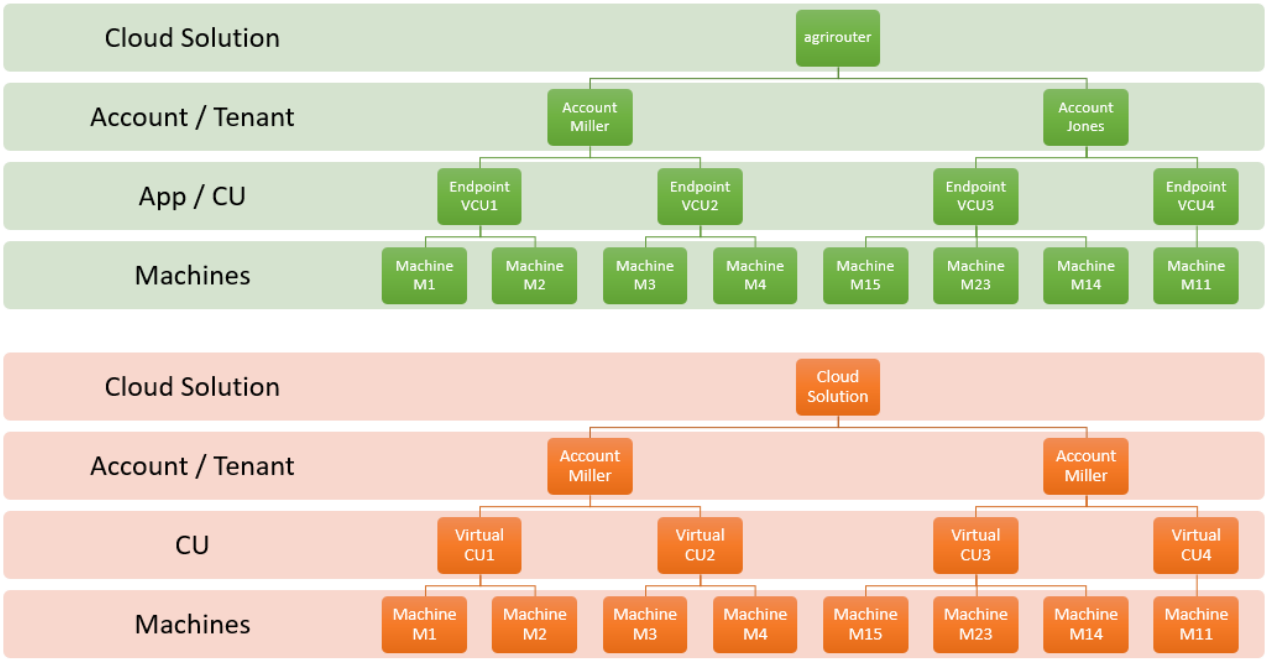

Figure 39 Comparing agrirouter hierarchy with a possible cloud platform hierarchy

11 Certification process

11.1 General

Every Application needs to be certified to be onboardable to the agrirouter. The certification makes sure, that the communication protocol of the agrirouter was successfully implemented and the application is able to communicate with the agrirouter.
A certification is needed any time, the app changes its capabilities.

11.2 Out of focus

The certification will not check, if an application “understands” the data sent over agrirouter, it will only check, if the application can send and receive data marked as such Technical message types, that were selected in the Certification Version.

11.3 Certification institutes

DKE and the agrirouter support team will release a list of certified partners, that are able and allowed to do the certification process. Please refer to https://my-agrirouter.com/support/ download area.

11.4 Tests

11.4.1 For Farming Software and Telemetry platforms

The test for farming software and telemetry platforms will check the following workflows:

- Onboarding of a Farming Software or Telemetry Platform
- Capabilities
- Receive data
    - Respecting capabilities
- Send data
    - Respecting capabilities

11.4.2 For Communication Units

The test for communication units will check the following workflows:

- Onboarding of Communication Units
- Capabilities
- Receive data
    - Respecting capabilities
- Send data
    - Respecting capabilities

12 Managing applications as app provider

The developer account offers several functionalities to manage the app providers apps.

12.1 Blocking specific Endpoints

To avoid misusage of data packages, an app provider can block a specific endpoint in the agrirouter UI. This endpoint is not able to communicate with agrirouter anymore, until the blocking is removed.

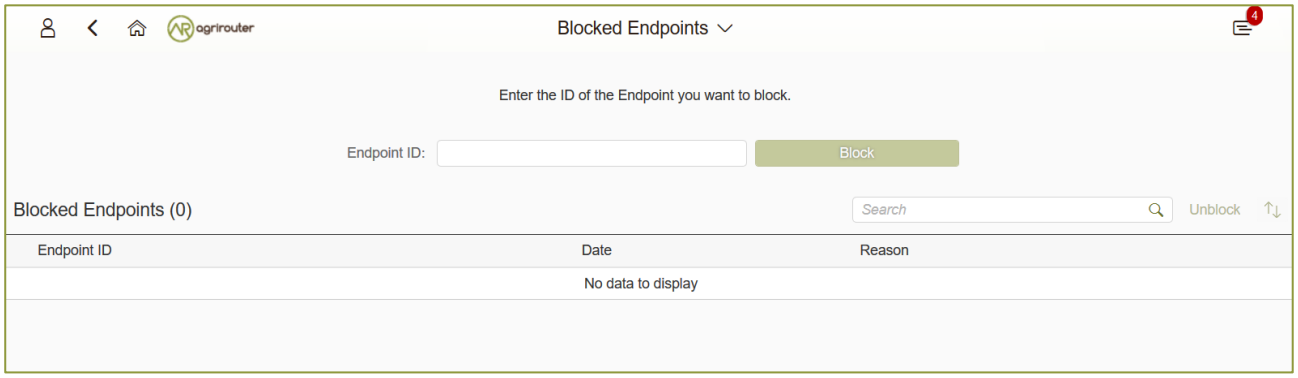

Figure 40 UI to manage blocked endpoints in developer account

12.2 Blocking several or all Instances of Certified Application

The agrirouter provides a mechanism to prevent all or specific instances of an application to send and receive commands. To achieve a blocking of all instances, the agrirouter administrator has the option to change the certification status of an application version from *Approved* to *Blocked*.

Every app provider can block specific instances of an application in the list of blocked endpoints. This can be used to disallow endpoints to communicate with the agrirouter, e.g. to avoid the app from creating further traffic.

When instances of blocked applications communicate with the agrirouter in order to send or retrieve commands, or to invoke other agrirouter functionalities, the agrirouter will respond with an error message, indicating that the instance is not allowed to use agrirouter functionality because of different reasons (payment not successful, certificate not valid, etc.).

The *Blocking* status can be changed by the agrirouter administrator to

- *either Approved* to allow all application instances to communicate again,
- *or to Rejected* as final status.

13 Service and Support

Agrirouter and agrirouter support team provide several tools and contact possibilities. A contact field and further information can be found here: https://my-agrirouter.com/support/ https://my-agrirouter.com/support. To get in contact with agrirouter support team, you can also send an email to support@my-agrirouter.com

13.1 Endpoint recording

Every agrirouter account includes a functionality to record communication on that account and specific for every endpoint. To record data, the user can start and stop the recorder in the agrirouter UI.

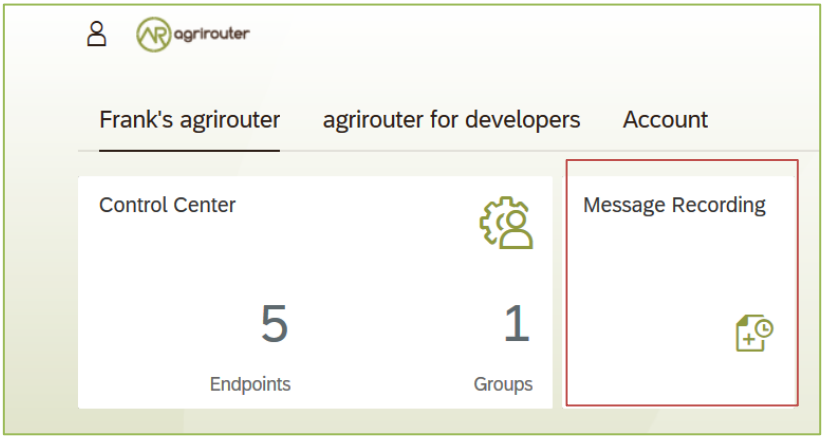

Figure 41 Endpoint Recording main view

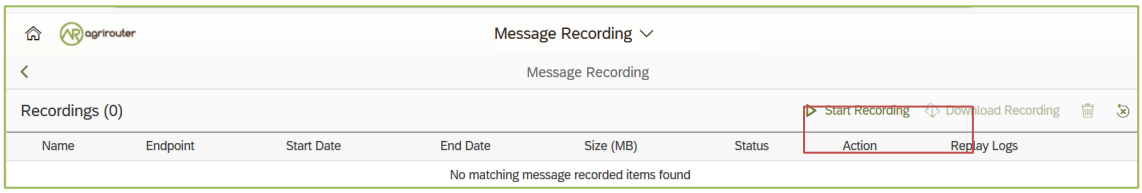

Figure 42 Endpoint Recording Overview

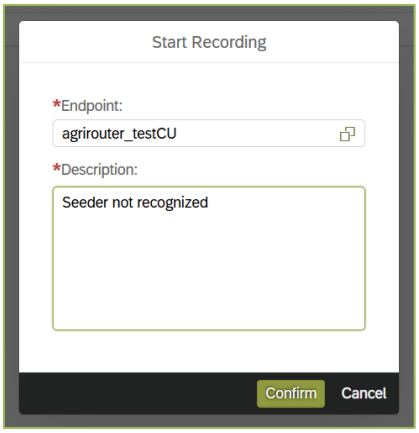

Figure 43 Start Recording of messages & commands

After recording started, the recording is displayed in the list of active records. Multiple records at the same time are possible to record multiple endpoints.

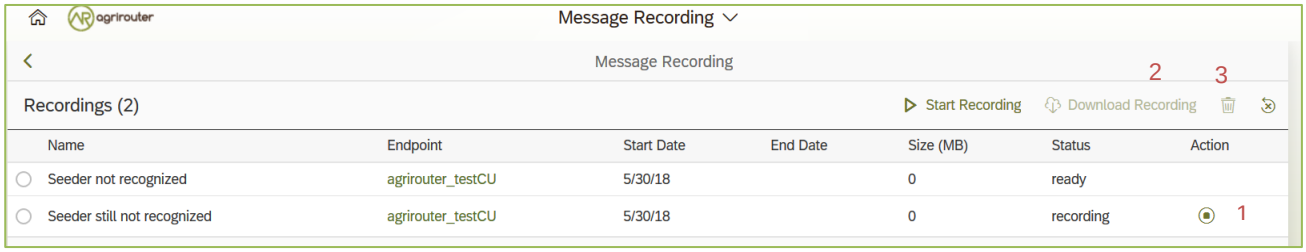

Figure 44 Recording overview

Recording can be stopped (1), downloaded (2) or deleted (3).

Important note:

Every account may store 10 recordings with a maximum size of 10 MB in a sum.

13.2 Replay of records

In a developer account, records can be uploaded (1) additionally to the recording functionalities, that are available in both account types; end user and developer.

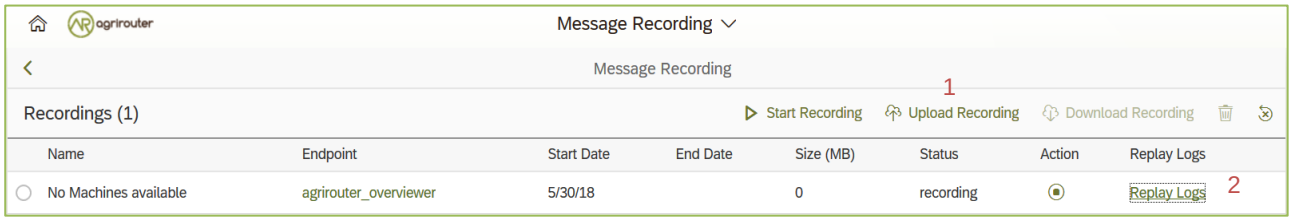

Figure 45 Endpoint recording in a developer account

As a developer, you are able to replay recorded commands (2); replacing the endpoint ID with any other endpoint ID. This will be described in more detail in Integration Guide Part 2.

13.3 Getting in contact with the agrirouter support team as a developer

The agrirouter support team provides support for developers for their integration with the agrirouter. To ask a question, simply send an email to support@my-agrirouter.com. Please understand, that agrirouter support team only provides agrirouter specific answers, no answers specialized for your coding language, environment or used frameworks. You might however be lucky to get such a specific answer, if agrirouter support team has one, so feel free to mention this information as well.

13.3.1 Required information

Helping with your request is easier, if you provide all information necessary to check the problem. Please check, which of the following topics fits your problem; it could even fit multiple problems

13.3.2 For endpoint related incidents

Please submit the following information with your request:

- Account Id
- Endpoint Id
- Endpoint Software Id
- Endpoint Software Version Id

(all of the above can be found in the endpoint details by clicking the 'Info' button next to the delete button)

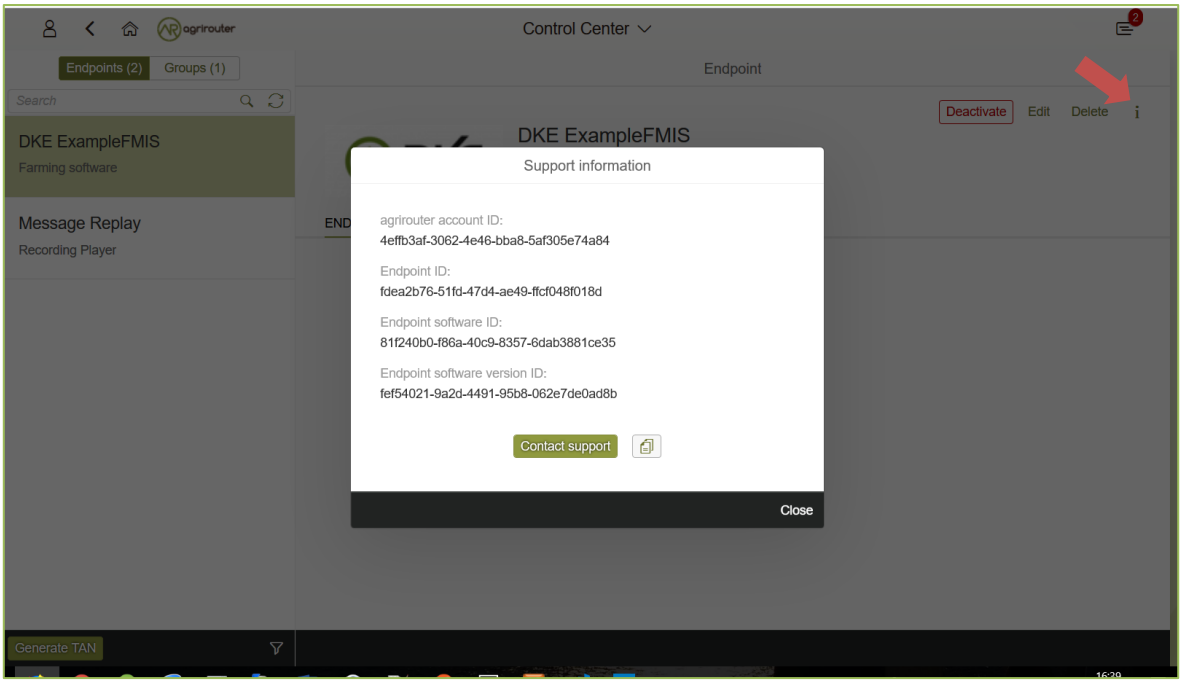

Figure 46 Relevant information for agrirouter support

13.3.3 For commands related incidents

Please provide the commands sent from and received by the endpoint, if available. The communication protocol can be provided as log file for example. On agrirouter Side, you can record commands, see 13.1Endpoint recording

13.3.4 For account related incidents

Please provide the account Id. See screenshot for the endpoint relevant information above.

13.3.5 For UI related incidents

Please provide the following information to report issues with the graphical user interface of the agrirouter

- which browser and version you are using to reproduce the issue,
- the localization settings (browser language, number and date formatting etc.)
- if issue occurs on a mobile device, on which device this happens (e. g. iPhone 7, iPad Mini, Galaxy S7 etc.)
- a screenshot of the issue

13.4 Workshops and Trainings

DKE and other companies might offer trainings or workshops for onboarding and development of software for the agrirouter. To see a list of these workshops and trainings, see http://www.dke-data.com/category/news/. If there are no offers available and you feel like you might need one, don't hesitate to send a request for training to info@dke-data.com.

13.5 Feature Requests

13.5.1 General

Having a good idea to improve agrirouter? Don't hesitate to send us an email to support@my-agrirouter.com . Any valid feature request will be discussed, and you'll receive feedback.

13.5.2 Message formats

If a message format is missing, please send us an email to support@my-agrirouter.com. DKE will check, if there is a common need for this. Please provide the following information with any request:

- **Name:** What's the name of the new format?
- **Type:** What message type would the new format fit the best? E.g. PNG would fit to a message type "Images"
- **Scope:** What is this message type used for? If possible, please provide a use case.
- **Documentation:** Is the format already documented? Please provide information on the required standard.
- **Ability of participation:** If DKE recognizes, that the format is not yet standardized or that there are multiple possible formats, it's very likely, that DKE will request you and other requesters to agree on a common format. Please state, that/if you are wanting to participate in such a group.
- **Timeline:** Please provide a timepoint, for which you would need the new format

agrirouter support team will get back to you on your request, please be informed, that this could take a while. For test purposes, you can however start your development with a proprietary format.

14 my-agrirouter.com Marketplace

Under the URL www.my-agrirouter.com it is possible for the application or hardware provider to present their own apps, functions, modules or hardware solutions.

This information must be created before going live under http://tptest.my-agrirouter.com/tptest/ in content management test environment according DKE specifications. DKE quality management reviewed, approved and released this information for go live.

After certification process is done, application or hardware provider can create partner account for my-agrirouter.com and can place their solution at my-agrirouter.com marketplace (https://my-agrirouter.com/en/marketplace/apps/).

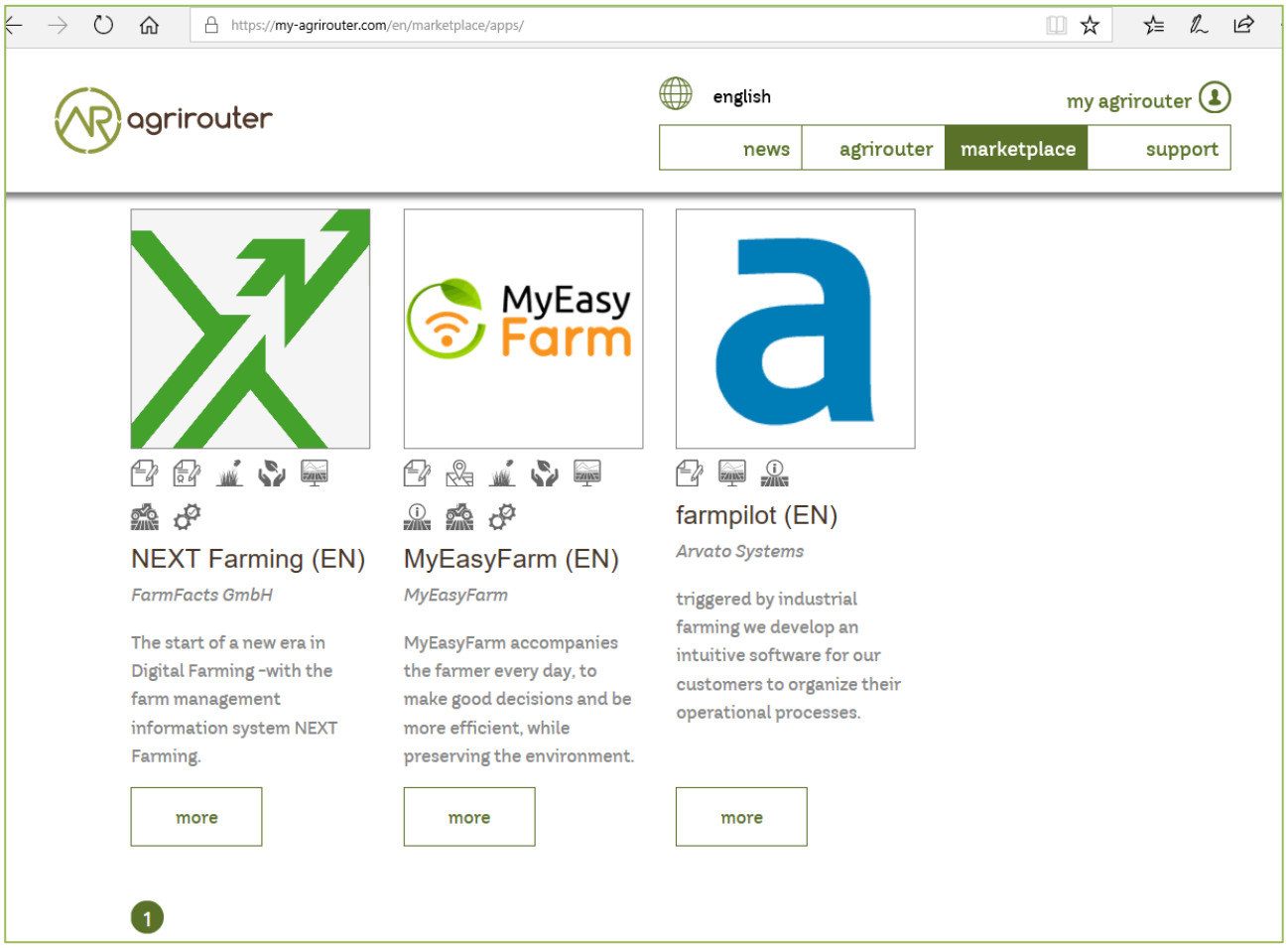

Figure 47 my-agrirouter.com Marketplace

DKE will publish a link for creating an partner account for my-agrirouter.com marketplace. After certification process is finalized the application or hardware provider is responsible for correctness and for keeping their solution information on the marketplace up to date.

Some examples of DKE partners are under link https://my-agrirouter.com/en/marketplace/apps/ available.

The following content is required from **application provider**:

#	Description	Quantity	Remarks
1	App / Module Name	1	-
2	Software Version	1	-
3	Release Date / Last Update	1	-
4	Reference to App (in case of a module)	1	-
5	Download link to App Provider	1	-
6	Keywords	10	-
7	Short description in 3 basic languages DE, EN, FR	< 150 characters	-
8	Long description in 3 basic languages DE, EN, FR	< 500 characters	-
8	App Provider name		
10	App Provider homepage		
11	Company Information	1	Address, Location, Support Information
12	Privacy Policy / Terms of use Link	1	
13	App Category	1 < n	Select box: 1. Documentation 2. Cross-Compliance documentation 3. Calculator 4. Application Maps 5. Prescription 6. Fertilization 7. Plant Protection 8. Farm Management and Information System (FMIS) 9. Product information (fertilizer, plant protection, seed, ...) 10. Machine Optimization 11. Process Optimization
14	Operating System / Platform	1 < n	Select box: 1. Native Android 2. Native iOS 3. Native Windows 4. Native Windows mobile 5. Native Mac 6. Web applications
15	Message format (receive)	1 < n	Select box: 1. Task-Data (TaskData) 2. Telemetry Data (EFDI) 3. Image 4. Video 5. Manufacture specific data formats
16	Message format (send)	1 < n	Select box: 1. Task-Data (TaskData) 2. Telemetry Data (EFDI) 3. Image 4. Video 5. Manufacture specific data formats

17	Country selection	1 < n	
18	Languages	1 < n	EN, DE, FR
19	App Screenshots	3 < n < 6	Max. 1400 width pixel Resolution in .jpg or .png
20	App Provider Logo	1	Min. 500x500, max. 1000x1000 pixel Resolution in .jpg or .png

The following content is required from **hardware provider**:

#	Description	Quantity	Remarks
1	Hardware Name	1	-
2	Hardware Version	1	-
3	Release Date / Last Update	1	-
4	Link to Hardware Provider	1	-
5	Keywords	10	-
6	Short description in 3 basic languages DE, EN, FR	< 150 characters	-
7	Long description in 3 basic languages DE, EN, FR	< 500 characters	-
8	Hardware Provider name		
8	Hardware Provider homepage		
10	Company Information	1	Address, Location, Support Information
11	Privacy Policy / Terms of use Link	1	
12	Compatible with manufacturer	1 < n	Select box: 1. AGCO 2. AMAZONE 3. EXEL Industries 4. GRIMME 5. HORSCH 6. KRONE 7. KUHN 8. LEMKEN 9. PÖTTINGER 10. RAUCH 11. SDF 12. CLAAS 13. John Deere 14. Kubota 15. CNH Industrial
13	Construction year	1 < n	Select box: from 2000 to 2020 (one-year steps)
14	Model Type	1	Manufacturer model type (depending of 12 and 13 selection criteria) …
15	Supported interfaces	1 < n	Select box: 1. ISOBUS-INCAB 2. Signal Socket 3. CAN-BUS 4. None

16	Power supply	1 < n	Select box: 1. 12V 2. 24V 3. Battery 4. 220V
17	GPS position	1	Yes / No / External
18	Mobile communication	1	Yes / No / External
19	SIM card from hardware provider	1	Yes / No
20	WIFI communication	1	Yes / No
21	Hotspot functionality	1	Yes / No
22	Input possibility for entering the agrirouter TAN number		Select box: 1. ISOBUS-Terminal 2. Mobile Device (Tablet, Smartphone) 3. Own display 4. Website 5. Other
23	Possibility to select different End user profiles	1	Yes / No (Telemetry box can be used with several agrirouter accounts / only one end user profile can be active at a time)
24	Can transmit Machine information	1	Yes / No
25	Message format (receive)	1 < n	Select box: 1. Task Data (TaskData) 2. Telemetry Data (EFDI) 3. Image 4. Video 5. Manufacture specific data formats
26	Message format (send)	1 < n	Select box: 1. Task Data (TaskData) 2. Telemetry Data (EFDI) 3. Image 4. Video 5. Manufacture specific data formats
26	Country selection	1 < n	
27	Languages	1 < n	EN, DE, FR
28	Product Photos / Screenshots	1 < n < 6	Max. 1400 width pixel Resolution in .jpg or .png
29	Hardware-Provider Logo	1	Min. 500x500, max. 1000x1000 pixel Resolution in .jpg or .png

15 Annex A (Glossary)

#	Concept	Description
1.	Application	In the agrirouter context, an application is a product which can connect to the agrirouter. The applications are not running on the agrirouter, but they connect to the agrirouter over the network.
2.	Application Instance	An application instance is the communication partner for an endpoint in an agrirouter account. An application instance represents a single software product or user account in a cloud software.
3.	Communication Unit (CU)	An application acts as a communication unit when • Its main purpose is to provide data collected by sensor systems like GPS or machine sensors • It manages the agrirouter communication for these machines. It sends machine data to agrirouter, and receives and processes messages addressed to machines
4.	agrirouter Account	An agrirouter account is a logical space on the agrirouter, that is used by one person or company. (e.g. a farm or contractor)
5.	agrirouter Message Format	An agrirouter command consist of a agrirouter command header (also called envelope) and the content (sometimes also called payload). The header contains fields such as the sender ID, the list of recipients, and the technical message type.
6.	agrirouter routings	agrirouter users can maintain routings to specify what data may go from which sender/source to which recipient. Without a corresponding routing, no data is transferred.
7.	Capabilities	Capabilities are a list of technical message types, an app instance supports to send or receive
8.	Command	A command is sent to the agrirouter from any endpoint. It consists of a header and an agrirouter control message (e.g. a TeamSet Message). The most common type of command includes a message of a specific message type, that shall be forwarded to one or more endpoints
9.	Directly Connected Endpoints	A directly connected endpoint is any application, that communicates with its own endpoint at the agrirouter.
10.	EFDI	EFDI (Extended FMIS Data Interchange) is the working title of an upcoming standard of the Agricultural Industry Electronics Foundation (AEF). It defines message formats for network communication between machines and FMIS as well as between different FMIS, including the transfer of live telemetry data from machines. The information that are transferred is based on ISOBUS concepts. It uses, for example, ISOBUS DDIs, and even the message structures are based on ISOXML elements. As the technical serialization format, EFDI does not use XML, but Google protocol buffers. The agrirouter uses EFDI messages as the payload format for some message types, for example for live telemetry data and device descriptions. At the time of this writing, EFDI is still in an early stage. The working group is just finalizing the internal definition of the first message types (live telemetry data, device descriptions).
11.	Endpoint	In the agrirouter context, an endpoint is an addressable entity in an agrirouter account, which is the communication entry point for a single app instance or machine.
12.	Endpoint Descriptions	The endpoint description contains detail information about an endpoint. Important part of this information is the list of the technical message types, the endpoint supports.
13.	Farm Management Information System (FMIS)	An application providing farm management functionality. An FMIS may, for example receive data from machines, send settings and tasks to machines, and exchange data with other applications.
14.	Information Type	A summarize of different Technical message types.

15.	ISOBUS, ISO 11783, taskdata	Wikipedia: “ISO 11783, known as Tractors and machinery for agriculture and forestry -- Serial control and communications data network (commonly referred to as "ISO Bus" or "ISOBUS") is a communication protocol for the agriculture industry based on the SAE J1939 protocol (which includes CANbus) .” (source: https://en.wikipedia.org/wiki/ISO_11783) ISO 11783-10 describes the file-based and batch-oriented interchange format between machines and FMIS, using XML (ISOMXL) and binaries with XML header data. This is called a taskdata file set, even though it can contain many other data (fields, products, crop, worker, and many more) and need not even contain a task. The ISOBUS standard specifies only the data but not how it is transferred between machines and FMIS.
16.	ISOBUS DDI	A DDI (literally: Data Dictionary Identifier) represents a device or process parameter in the ISOBUS norm. Over 500 DDIs are defined in ISO 11783-11, see http://dictionary.isobus.net/isobus/ . DDIs are used, for example, in device descriptions to describe properties, the supported settings, and the provided data of a machine. DDIs are also used to identify the data records in telemetry messages. agrirouter users can filter telemetry data for DDI Categories, that abstract all DDIs into 10 categories. A list of DDIs and their Category assignment can be found here: A filtering for specific DDIs could not be provided, as this would have been too complex for the end user.
17.	Live Telemetry Messages	A live telemetry message contains data points with data from the machines in one teamset. Each data point at least contains the time when it was logged. Additionally, each data point can have a geo-position. Each data point may contain many log entries. Each log entry contains the value of a specific parameter (DDI) of a specific component or function (device element) of one of the machines in the teamset. The user-defined agrirouter routings define which parameters of which machine may go to which recipient. In addition, recipients can subscribe for certain parameters they want to get (sender-independent). Recipients will receive those data, when they are sent to the public address. The agrirouter uses the routings and the subscriptions to determine the recipients, and the information each of them gets. To each recipient it delivers a filtered version of the messages, which contains only the allowed and subscribed parameters of the allowed machines.
18.	Machines	Machines are agricultural machinery, in the sense of a ISO11783 device. Machines are tractors, implements such as sprayers, or self-propelled machines like combine harvesters. A machine is described with machine description, which is conceptually based on ISO11783-10 device description. Machines are the sources of the data records in live telemetry messages which applications send via agrirouter. From an abstract view, machines are just sensor networks providing sensor data.

19.	Machine Endpoint	A machine endpoint represents one real-world machine in the context of an agrirouter account. The same real-world machine can have endpoints in several agrirouter accounts, but not more than one in the same account. A message can be addressed to a machine as the recipient. This tells the agrirouter to deliver the message to that CU, to which the machine is connected. If the addressed machine is currently not connected, the agrirouter puts the message in the machine's feed and delivers it as soon as some CU reports that the machine is now connected. Since machines can be attached to different CUs at different times (connected to different tractor, for example), it is not known in advance which CU that will be. The message itself will be received and processed by the CU, but addressing it to the machine makes sure that it goes to the right one.
20.	Message	A message is an information or perhaps a request, that is sent from an endpoint to any other endpoint. A message is a possible payload of a command
21.	Teamset	A teamset is a set of connected machines which work and move together and are connected to the same communication unit.
22.	Virtual Communication Units	A virtual communication unit is the equivalent of a communication unit for situations where the teamsets are not directly connected to the agrirouter. Instead they are connected to an external telemetry-enabled cloud service, which itself is connected to the agrirouter. Such a telemetry-enabled cloud application has its own mechanisms for connecting farming machines. For each farm, many machines are connected to the external cloud service, grouped in many teamsets. The cloud application, which makes these machines known on the agrirouter, also reports one virtual communication unit for each teamset.

16 Annex B (Shortings)

As agrirouter uses multiple, sometimes quite long special terms, these terms are shortened in some of the graphics. Following a list of shorting's and the corresponding full term

Shorting	Full Term
[]	Shorting for a list
…	additional steps; not shown here
App	Application
ar	agrirouter
CAP	Capabilities
CU	Communication Unit
DDI	Data Dictionary Identifier; see ISOBUS DDI
DVC	Device Description/ Device
EP	Endpoint
IT	Information Type
MSG	Message
TMT	Technical Message Type
VCU	Virtual CU; Virtual Communication Unit

17 Resources

Following resources can help developers to integrate their applications to the agrirouter.

Name	URL	Description
Agrirouter main page	https://my-agrirouter.com	The official website of the agrirouter
GitHub	https://github.com/dke-data/	Official GitHub with libraries and resources
DDI Dictionary	https://dictionary.isobus.net/	The official list of standardized Sensor values
AEF	https://aef-online.org .	The official website of the Agricultural Electronics Foundation. This group defines EFDI and the standards of ISOXML.