

Section 15 – Overview of Guidelines for Data Exchange

Section 15 – Data Exchange

Version July, 2018

Ref: 15 Data Exchange v18.01.docx

Table of Contents

1	Introduction	4
2	Definitions and Terminology	4
3	Goals	5
4	Scope	6
5	Presentation of animal data exchange.....	6
5.1	Background.....	6
5.2	Objective of animal data Exchange	7
5.3	How to achieve the objectives	7
5.3.1	Content of ADE	7
5.3.2	ADE implementation	8
5.4	Business context	9

Tables

Table 1.	Definitions of Terms used in these guidelines.	4
----------	---	---

Figures

Figure 1.	Current situation for data exchange.....	7
Figure 2.	Animal data exchange architecture.	7
Figure 3.	Life cycle of the framework.....	8
Figure 4.	Background business context.	9

Change Summary

Date of Change	Nature of Change
August 2017	Reformatted using new template.
August 2017	Table of contents added.
August 2017	Heading numbers and heading text edited for clarity and removal of redundant text.
August 2017	Table and Figure numbers added. Cross reference to text added. List of tables and figures added to table of contents.
August 2017	Stopped Track changes and accepted all previous changes.
August 2017	Moved the file to the new template (v2017_08_29).
September 2017	Updated version to September. Corrected heading format error. Updated table of figures.

October 2017	Hyperlinks corrected and the date of the version updated.
July 2018	New version from ADE WG (1.8) replaces previous version in its entirety. Technical elements moved into Procedures (1, 2, 3, 4 and 5) and Data Dictionary moved to Appendix A of Section 15.

1 Introduction

Animal Recording involves the collection, storage and exchange of data. Over time the use of electronic systems has spread from initially, in the early 1970s, being used just for centralised data processing to now (in 2016) being used in all aspects of animal recording. The electronic systems for collecting, storing, transmitting, and processing data and information have evolved to be used widely by technicians, farmers, advisors and central systems. The evolution is continuing at a rapid pace worldwide. With this spread of electronic systems the need for standards to facilitate the ready transfer of data between systems has also grown very rapidly.

This part of the ICAR guidelines (chapter 15) is devoted to standards for facilitating data transfers and to processes for ensuring electronic systems comply with the ICAR guidelines.

2 Definitions and Terminology

Table 1 contains a list of important definitions for terms and abbreviations used in these guidelines.

Table 1. Definitions of Terms used in these guidelines.

Term	Definition
Data collection device	An electronic device which captures data. Examples include: an electronic data entry device (inc. keyboard, mouse, point and click, touch pad, touch screen), weigh scales, milk analyser, camera, electronic milk meter, electronic identification reader and a pedometer.
Decision information	Information that is provided by a device or computer system that is used in making a decision. The overall goal and purpose of animal recording is to help make better decisions.
Farm system	A computer system which collects, stores, and analyses animal data relating to a farm. The outputs of a farm system are used by the farmer to support a range of management decisions. Typically these were PC based systems located on the farm which have more recently started to evolve to cloud based systems.
Field technician	Any person who visits a farm to provide a specific animal related service. These include, for the purposes of this guideline, veterinarians, artificial insemination technicians, milk recording technicians, hoof trimmers, linear scorers, and animal classifiers.
Information system	A computer system, often with a relational database at its core, which collects, stores, and analyses animal data relating to many farms. These may be specific to certain services, for example registrations, movements, milk recording, Herd Book, artificial insemination, genetic evaluations (national or international), or they may cover multiple services for an entire industry within a defined geographic region.

Term	Definition
Milking equipment	Milking equipment is a set of hardware and software encompassing at least the following functions: a. Measuring milk quantity and characteristic, b. Filling bottles of milk sample and registering the links between the bottle identifier and that of the animals, c. Storing milking results, and d. Connecting, sending and receiving data from external information systems.
Operator	An operator may correspond to different persons: farmer, farm staff, or a technician from a service provider such as a milk recording organization.
REST	Representational State Transfer (REST) is an architectural style that defines a set of constraints to be used for creating web services.
Service provider	Organizations provide milking equipment owners with services whose purpose is to: a. have their data registered by a multipurpose data base, and b. to deliver aggregated information.
SOAP	(originally Simple Object Access Protocol) is a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks.
Technician system	A computer system used by a field technician providing animal services to multiple farms. Typically these systems collect, store, and analyse animal data relating to the farms serviced by one technician. The system provides information used by the technician to manage their work.
UNCEFACT	is the United Nations Centre for Trade Facilitation and Electronic Business. It was established as an intergovernmental body of the United Nations Economic Commission for Europe (UNECE) in 1996 and evolved from UNECE's long tradition of work in trade facilitation which began in 1957.
W3C	World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards.
WSDL	Web Services Description Language (WSDL) is an XML-based interface definition language that is used for describing the functionality offered by a web service.
XML	Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

3 Goals

The goals of these guidelines are:

- a. To provide the developers of data processing systems with standards and processes that they can use to ensure the devices and systems they develop can be readily and reliably linked with the full range of animal data information systems used by ICAR members.

- b. Harmonization of the definitions of exchanged data in order to be able to exchange information between heterogeneous information systems.
- c. Development of global consensual data dictionaries for livestock.
- d. Development and installation of standardized systems to support data exchange between information systems and farm equipment.

4 Scope

Animal data exchange consists of:

- a. Business requirements and technical specifications
- b. Data descriptions provided by files in compliance with the recommendations from W3C for XML schemas, UNCEFACT and ISO
- c. Interface specifications provided by files in compliance with the recommendations from W3C for web service description languages (WSDL)

This Guideline gives business requirements which are divided into three types:

- a. General business requirements which address any type of data exchange
- b. Specific business requirements which address a given type of data exchange (Procedure 2 to
- c. Data description (Appendx A)

For a given type of data exchange, specific business requirements detail:

- a. Purpose of data exchange
- b. Business context of data exchange
- c. Requirements to exchange data which includes exchanged message description

Data description encompasses:

- a. Service description
- b. Message composition
- c. Entity description
- d. Data item description
- e. Code set description

Technical implementation deals with the implementation of the same business requirements according different ways:

- a. Primarily by using W3C and UNCEFACT recommendations for data and interface definition (SOAP)
- b. Alternatively by using the existing ISO standards (17532 2007) for data exchange on farm with stationery equipment and REST technology (This has to be elaborated more in detail in a future version of the ICAR ADE specification).

5 Presentation of animal data exchange

5.1 Background

At the moment, data exchanges between equipment and external information systems do not exist or require a sort of middleware (see Figure 1) between the equipment and the

information system. It is based mainly on regional, manufacturer specific or outdated international standards (ISO ADED 1996).

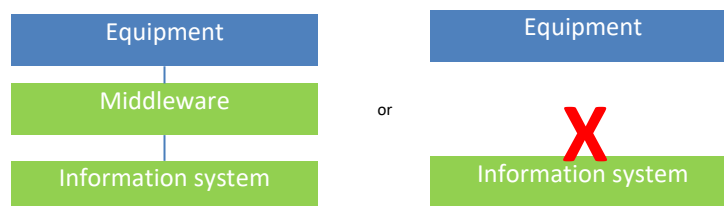


Figure 1. Current situation for data exchange.

This middleware is costly to implement, difficult to maintain. It requires manual operations and does not allow to reduce the delay between an event and its registration by an external information system and to exchange large amount of data.

Equipped with modern high-speed IT infrastructure today's animal recording data centres are capable of processing huge amount of data created by an exponentially growing number of on farm equipment, sensors and analytical methods. In order to get reliable access to this data a high level of automation and standardization of data exchange has to be organized.

A data exchange in real time of large amount of data would allow:

- a. Animal breeding on new traits
- b. Improve animal monitoring by the aggregation of data from different sources: equipment, analysis laboratories...
- c. Improve farm equipment calibration in order to get accurate measurements

5.2 Objective of animal data Exchange

The objective is to establish direct, permanent, reliable, easy to implement and to maintain and cost-effective exchanges of large amount of data both ways; between equipment and external information systems and from equipment to external information systems.

5.3 How to achieve the objectives

5.3.1 Content of ADE

This objective may be achieved by a framework consisting of:

- a. An architecture to support data exchange
- b. Standards for messages and data
- c. Tools to facilitate its implementation
- d. A reactive maintenance process

The architecture is based on a service oriented approach where the equipment is the client of the external information system which is a service provider (see Figure 2). This architecture may be implemented by using different information technologies which are specified by this document.

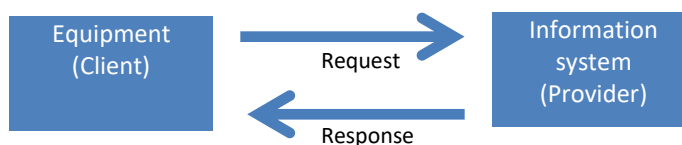


Figure 2. Animal data exchange architecture.

The standards encompass:

- a. Business requirements to use the service
- b. Business requirements to deliver the service
- c. Business requirements for exchanged data
- d. Semantic definition of exchange data as well as their cod set
- e. Syntax specifications for the exchanged data according W3C, UNCEFACT and ISO
- f. Interface specifications according W3C, UNCEFACT and ISO

The tools to facilitate the implementation encompass:

- a. A web site to download :
 - Files of xml data types (XSD files)
 - Files of web service specifications (WSDL files)
- b. A test plate form

5.3.2 ADE implementation

Figure 3 below describes the process.

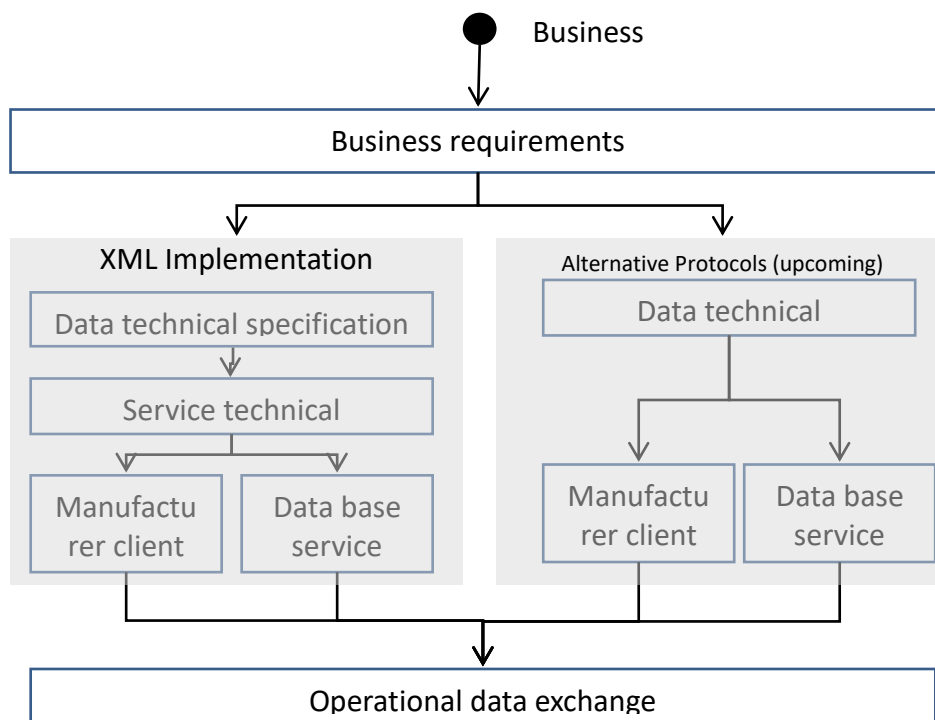


Figure 3. Life cycle of the framework.

The process is triggered by a business need.

Business requirements should be the result of collaboration between ICAR and manufacturers. They are independent from the techniques used to implement them.

Two types of implementations are considered:

- a. W3C and UNCEFACT standards, the primary one
- b. ISO (17532 2007) / REST, upcoming alternatives

The W3C implementation consists of:

- a. Data element technical description as “xsd” files based on UNCEFACT data types;

- b. Interface technical specifications as “wsdl” files
- c. From WSDL:
 - Client implementation by the manufacturer
 - Server implementation by the external information system

5.4 Business context

Figure 4 describes the business context.

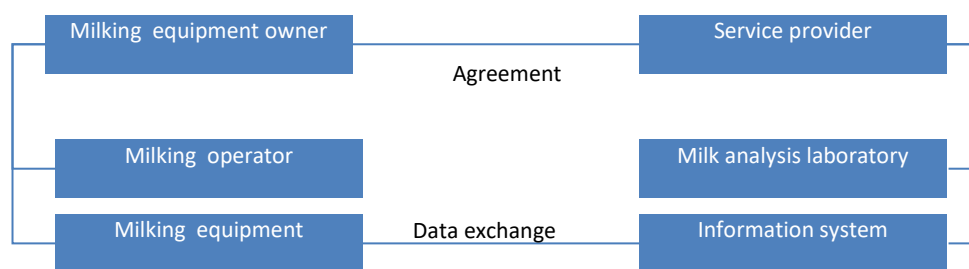


Figure 4. Background business context.

Service providers are organizations who provide milking equipment owners with services whose purpose is:

- a. to have their data registered by a multipurpose data base
- b. to deliver aggregated information

These organizations may be milk recording organizations, breeding organizations or other data base service providers.

The owner of the milking equipment and the service provider should agree on the conditions to use services and namely the parameters which are required for data exchange. Generally, this agreement is formalized by a contract.

Milking equipment is a set of hardware and software encompassing at least the following functions:

- a. Measuring milk quantity and characteristic
- b. Filling bottles of milk sample and registering the links between the bottle identifier and that of the animals
- c. Storing milking results
- d. Connecting, sending and receiving data from external information systems

Milking equipment may have several other functions which are not considered by that data exchange: animal monitoring, milking monitoring. However, characteristics not directly linked to milking but recorded during the milking process, e.g. weighing in the milking box, could be transported.

A milking system may correspond to different types of equipment: robot, electronic milking meters...

The milking equipment is operated under the supervision of a milking operator who is staff dependent from the owner and designated as ‘Operator’ throughout the rest of the document. A milking operator has the right and the capacity:

- a. to feed the milking equipment with parameters for data exchange and milk samples
- b. to install and to remove the bottles used for milk sample,
- c. to send milk samples to a milk analysis laboratory

A milking equipment operator may correspond to different actors: farmer, a staff dependent from a farmer, a technician from a service provider such as a milk recording organizations...

The service provider operates an information system which provides the services. That information system consists of server, data bases and software, which are not a part the milking equipment and which is connected to the milking equipment by a network. It has at least the following functions:

- a. Receiving and processing requests from the milking system
- b. Updating the data base of the information system according the requests of the milking system
- c. Providing the results of milk analytic laboratories which analyze the samples collected by the requesting milking equipment.

6 General Specifications

The general specifications are contained in Procedure 1 of these Section 15 ICAR Guidelines – [link here](#).

7 Specific Services

7.1 Milking

Specifications for milking services are contained in Procedure 2 of these Section 15 ICAR Guidelines – [link here](#).

7.2 Exchange of Animal Data

Specifications for exchange of animal data services are contained in Procedure 3 of these Section 15 ICAR Guidelines – [link here](#).

7.3 Exchange of Reproduction Events

Specifications for exchange of animal data services are contained in Procedure 4 of these Section 15 ICAR Guidelines – [link here](#).

7.4 Technical Services

Specifications for exchange of technical services are contained in Procedure 5 of these Section 15 ICAR Guidelines – [link here](#).

8 Data Dictionary

The data dictionary is contained in Appendix A to these Section 15 ICAR Guidelines – [link here](#).

9 References

1. Semantics for Smart Dairy Farming: a milk production registration standard – SDF June 2013
2. UN / UNCEFACT Modeling Methodology User Guide (CEFACT / TMG/No93)
3. UN / UNCEFACT Business Requirements Specifications Document Template (CEFACT/ICG/005)
4. ISO 11787: Electronic data interchange between information systems in agriculture – Agricultural data interchange syntax

5. ISO 11788: Electronic data interchange between information systems in agriculture – Agricultural data element dictionary –Part 1: General description –Part 2: Dairy farming
6. ISO 17532: Stationary equipment for agriculture –Data communications network for livestock farming
7. ISO 11784: Radio frequency identification of animals - code structure
8. ISO 3166 -1: Country code
9. ICAR Guidelines



THE GLOBAL STANDARD
FOR LIVESTOCK DATA

Procedure 1 of Section 15 of ICAR Guidelines - Methodology

Procedure 1 – Methodology

Version July, 2018

File Ref: 15 Procedure 1 Methodology v18.01.docx

Network. Guidelines. Certification.

Table of Contents

1	Introduction	4
2	Definitions and Terminology	4
3	Scope	4
4	General Data Transport Specifications	5
4.1	SOAP/XML Web Services	5
4.2	Alternative Transport Protocols (ADIS/ADED/IsoAgriNet, REST)	7
4.3	ICAR Types and Code Lists	7
4.4	External Elements	7
4.4.1	UNCEFACT Data types	7
4.4.2	UNCEFACT Code Lists	8
4.4.3	ISO Code Lists	9
4.4.4	IANA Code Lists	9
5	Access to the Service	9
5.1	Exchange Parameters	9
5.2	Data Transmission	10
5.3	Encryption	10
5.4	Compression	10
5.5	User Right Verification	11
6	Communication work flow	11
6.1	Data Processing	11
6.2	Request Specification	11
6.3	Response Specification	12
6.4	Best Praxis for Message-Id Creation	13
7	Dealing with Local Requirements	14
7.1	Local Code Lists	14
7.2	Local Additional Data	14
8	Common Components	14
8.1	ADE Exchanged Document Type	15
8.2	ADE Party Type	15
8.3	Standard Request Type	16
8.4	Standard Response Type	16
8.5	Error Type	17
8.6	Ticket Response Type	17
8.7	ZIP Message Type	18
8.8	Local Additional Data Type	18
8.9	Time Period Type	18
9	Naming Rules	19
10	References	19

Tables

Table 1. Definitions of Terms used in these guidelines.....	4
Table 2. ICAR ADE schema files	6
Table 3 XSD schema symbols	7
Table 4 UNCEFACT basic data types.....	8
Table 5 UNCEFACT Code Lists.....	8
Table 6 ISO code lists.....	9
Table 7 IANA Code Lists.....	9
Table 8 RequestProcessingStatusCodeType (base xsd:string)	12

Equations

No table of figures entries found.

Figures

Figure 1. Scope of procedure 1 of Section 15 of ICAR Guidelines.....	5
Figure 3 Example “GetHerdListRequest”	6
Figure 4 Request with a specific request.....	12
Figure 5 Response with a Specific Response.....	13
Figure 6 Specific Response Schema.....	13
Figure 7 Standard Response	13
Figure 8 ADEExchangedDocumentType	15
Figure 9 ADEPartyType	16
Figure 10 StandardRequestType	16
Figure 11 StandardResponseType	17
Figure 12 ErrorType	17
Figure 13 TicketResponseType	18
Figure 14 ZipMessageType	18
Figure 15 LocalAdditionalDataType.....	18
Figure 16 TimePeriodType	19

Summary of Changes

Date of Change	Nature of Change
July 2018	New procedure created from methodology chapter in Section 15 Overview. Replaced by new version from ADE-WG. Template applied. ADE WG feedback used to update last para in 4.2 to better reflect priorities of ADE WG.

1 Introduction

For electronic data exchange a lot of technologies exist and new technologies are continuously evolving. However, in order to establish standardized data exchange we have to concentrate on a defined set of accepted and mature technologies and describe the usage in detail.

This procedure, as part of Section 15 Guidelines, introduces the common methodology and technology used by the specific business processes described in procedures 2, 3 and 4:

- a. **General Data Transport Specifications**
describes the applied web service transport protocols and data protocols
- b. **Access to the Service**
describes some aspects to be taken into account when implementing and installing the services
- c. **Communication work flow**
covers the communication work flow aspects in a business process session like request, response and data processing.
- d. **Dealing with Local Requirements**
describes in detail how locally obligatory data elements which are not covered by the ICAR scope can be added to the interfaces
- e. **Common Components**
describes all the common protocol components which are reused in the specific business processes

2 Definitions and Terminology

Table 1 contains a list of important definitions for terms and abbreviations used in these guidelines.

Table 1. Definitions of Terms used in these guidelines.

Term	Definition
TCP/IP	Transmission Control Protocol
HTTP	Hyper Text Transfer Protocol
XML	Extensible Markup Language
XSD	XML Schema Definition
SOAP	Simple Object Access Protocol
WSDL	Web Service Description Language
REST	Representational State Transfer
ADIS	Agricultural Data Interchange Syntax
ADED	Agricultural Data Element Dictionary
JSON	JavaScript Object Notation
UNCEFACT	UN/CEFACT is the United Nations Centre for Trade Facilitation and Electronic Business
ISO	International Organization for Standardization
IANA	Internet Assigned Numbers Authority

3 Scope

gives a pictorial summary of the main elements of this procedure. The numbers in this figure refer to the heading numbers of this procedure.

Figure 1. Scope of procedure 1 of Section 15 of ICAR Guidelines.

4 General Data Transport Specifications

4.1 SOAP/XML Web Services

For the primary data transmission method the W3C standard SOAP/XML version 1.2 has been chosen. Detailed descriptions of this standard can be found here:

<http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>

Since a “top down” approach has been adapted, ICAR ADE provides WSDL files and a set of XSD files which make up a complete set of machine readable definition files from which SOAP web service implementations can be created. Most modern programming languages e.g. Java, .NET etc. provide tools which facilitate an easy setup process of interfaces classes to be used as a link between data transport layer and business process layer.

For each release of the ICAR ADE specifications a specific set of WSDL/XSD files can be downloaded from the ICAR ADE website – [link here](#). See Table 2. ICAR ADE schema files” for a description of the files. The files will be provided in a zip archive file e.g. “ADE_Schema20150309_1.8.zip” for version 1.8 available [here](#).

The WSDL/XSD files describe different aspects of the transport protocol:

- f. Service end points
- g. Input and output messages
- h. Basic data type definitions (XSD types, UNCEFACT types etc.)
- i. Complex data type definitions
(entity tags, item tags, compositions, attributes, usage of basic data types)
- j. Constraints (data type and value range restrictions)

Table 2. ICAR ADE schema files

File Name / File Name Schema	File Description
*.XSD	XML Schema Definition files, machine readable description of the ICAR ADE XML data structures
ICAR_*.XSD	Data element definitions created by the ICAR ADE group
ISO_*.XSD	Data element definitions derived from ISO standards
UNECE_*.XSD	XSD files provided by the UN/CEFACT project: (see http://www.unece.org/cefact/xml_schemas/index.html)
UnqualifiedDataType_13p0.XSD	XSD file which defines the unqualified UNCEFACT data types. These data types extend the standard XSD data types with additional attributes. In the ICAR and UNCEFACT definitions they are used instead of the standard XSD types.
CODE	XSD files defining code lists
wsMrAde.wsdl	ICAR ADE SOAP service description, (WSDL = Web Service Description Language) It defines a set of message pairs each consisting in a request and a response message.
ADE_v1p8.xsd	This file provides the entry point to the ICAR ADE message data structure definitions

Throughout this document messages and data definitions are illustrated using graphical presentations of XSD structures created by the commercial software XMLSPY from ALTOVA. See for example Figure 2 Example “GetHerdListRequest”

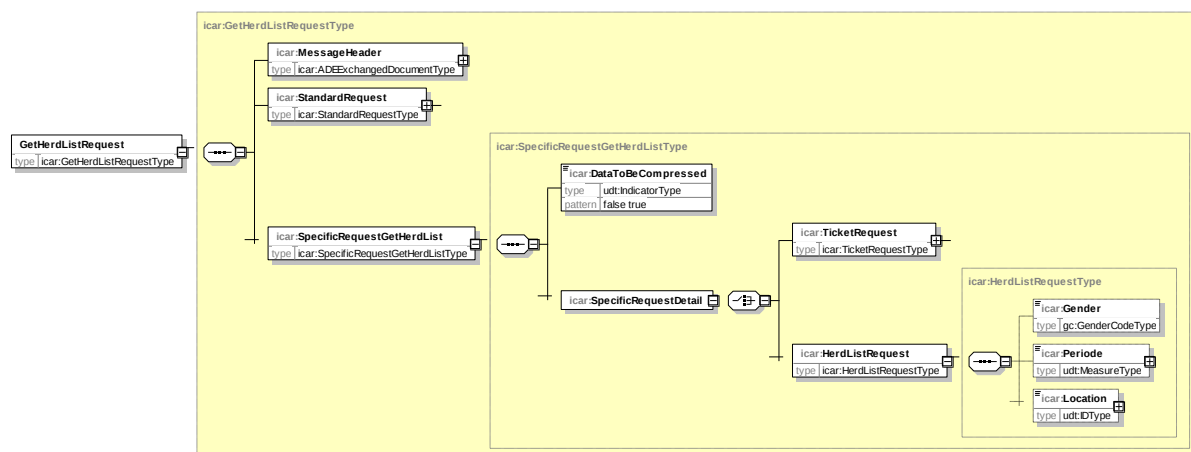
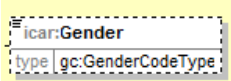
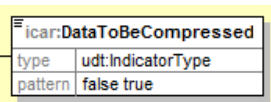
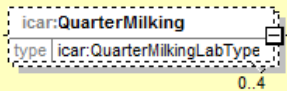


Figure 2 Example “GetHerdListRequest”

Below the symbols used to represent the XSD schemas in the following pages:

Table 3 XSD schema symbols

Description	Symbol
Optional tag	
Mandatory tag	
Sequence	
Choice	
Multiplicity	

4.2 Alternative Transport Protocols (ADIS/ADED/IsoAgriNet, REST)

Parallel to SOAP other international transport protocols exist. This is for example the ISO ADIS/ADED standard and its last extensions described in ISO 17532(2007). On the other hand the REST protocol, based on the HTTP transport protocol using XML, JSON and other data encoding technology, has gained a broader acceptance in the domain of internet data transfer during the recent years.

The ADE-WG is working on the development of a standardized REST-API description.

4.3 ICAR Types and Code Lists

Elements created within the process of the ICAR ADE specification can be identified by the prefix “**icar:**” within the WSDL/XSD files and descriptions below.

All data items are based on UNCEFACT data types (see section UNCEFACT Data Types)

A detailed description of the data elements is given below.

4.4 External Elements

It is the aim of the ICAR ADE standardization process to make as far as possible use of elements already defined within existing standardization frameworks as ISO, IANA and UNCEFACT.

4.4.1 UNCEFACT Data types

The ICAR ADE WSDL/XSD definitions make extended use of UNCEFACT basic types.

UNCEFACT basic types are types derived from basic XSD types. The concept consists mainly of adding further attributes and restrictions to the XSD types. They are defined in file UnqualifiedDataType_13po.xsd.

In the WSDL/XSD files and descriptions below those elements can be identified by the prefix “**udt:**”

Table 4 UNCEFACT basic data types” lists the UNCEFACT types currently used for the types of a data item.

Table 4 UNCEFACT basic data types

Name	Description	Comment	XSD Type
udt:CodeType	A character string that may be used to represent or replace the definitive value.	A code is referring to an enumeration	String
udt:DateType	A particular point in the progression of date	Representation as defined by transport protocol	String
udt:DateTimeType	A particular point in the progression of time	Representation as defined by transport protocol	String
udt:IDType	A character string used uniquely to establish the identity of, and distinguish, one instance of an object within an identification scheme from all other objects within the same scheme.		String
udt:MeasureType	A numeric value determined by measuring an object	Measures are specified with a unit attribute holding a measure unit, a resolution and when it is required a minimum and a maximum value.	Decimal
udt:NameType	A word that constitutes the distinctive designation of a person, place, thing, or concept		String
udt:BinaryObjectType	Base64 coded binary data	Used e.g. for zip compressed message	String
udt:TextType	A general text type		String
udt:IndicatorType	A Boolean indicator or true and false		Boolean

4.4.2 UNCEFACT Code Lists

UNCEFACT provides an extensive set of code lists. See Table 5 UNCEFACT Code Lists” for the code lists used by the ICAR ADE specifications.

Table 5 UNCEFACT Code Lists

Code List	Description
-----------	-------------

AgencyIdentificationCode	A list of agencies responsible for code list maintenance to be used as value of the attribute listAgencyID in the UNCEFACT data type udt:CodeType
CharacterSetEncodingCode	A list of character set encoding codes to be used as value of the attribute encodingCode in the UNCEFACT data type xsd:base64Binary
MeasurementUnitCommonCode	An extensive set of commonly used units to be used as value of the attribute unitCode in the UNCEFACT data type udt:MeasureType

4.4.3 ISO Code Lists

See Table 6 ISO code lists for the ISO code lists used by the ICAR ADE specifications.

Table 6 ISO code lists

Code List	Description
ISO3AlphaCurrencyCode	A list of ISO currency codes to be used as value of the attribute currencyID in the UNCEFACT data type udt:AmountType
ISOTwoletterCountryCode	A list of two letter ISO country codes to be used as value of the item Country

4.4.4 IANA Code Lists

See Table 7 IANA Code Lists for the IANA code lists used by the ICAR ADE specifications.

Table 7 IANA Code Lists

Code List	Description
CharacterSetCode	A list of IANA character set codes to be used as value of the attribute characterSetCode in the UNCEFACT data type xsd:base64Binary
MIMEMediaType	A list of IANA MIME media type codes to be used as value of the attribute mimeCode in the UNCEFACT data type xsd:base64Binary

5 Access to the Service

5.1 Exchange Parameters

The equipment should provide the operator with an interface to capture exchange parameters for a given service provider consisting of:

- a. Service provider URL
- b. Authentication information (user/password, token)
- c. Sender ID
- d. Sender name
- e. Sender country
- f. Recipient ID
- g. Recipient name
- h. Recipient country
- i. Type of location
- j. Type of identifier of the location
- k. Identifier of the location

- l. Name of the location
- m. Country of the location
- n. Type of primary animal identifier
- o. Type of secondary animal identifiers (e.g. on farm animal id, animal name)
- p. Other manufacturer or service provider specific parameters

For a given service provider the exchange parameters should be the same for all the consumed services.

5.2 Data Transmission

As the services are not permanently accessible, no request should be sent by the equipment before having checked if there is access to:

- a. The network
- b. The service provider server
- c. The service itself

Any new data should be registered by the information system as soon as possible.

No new data should be sent to the information system as long as previous data is still being processed. The service consumer has to wait for a processing result or a ticket in case of asynchronous processing.

No ticket should be sent to the information system before the time given by the information system.

Any message whose syntax is not in compliance with the syntax requirements should not be sent.

As soon as a data is included in a request and answered by a response it should be considered as sent to the information system.

Service providers and consumers should be prepared for updates and deletes. The service provider should also take into account that the sending party is declaring updates as a create operation not knowing that the data is already present in the service provider's data base. (e.g. in case of retransmission initiated by the consumer)

Preparing a request in compliance with business requirements should be performed separately from preparing a message whose syntax depends on the type of technical mapping (web service with SOAP at the moment).

5.3 Encryption

If possible encryption should be used in order to insure privacy and authenticity. The process of encryption itself is not described here. The encryption capabilities of the applied transport protocol should be used e.g. HTTPS for HTTP-Transport.

5.4 Compression

In order to minimize the transport delay of big messages, especially when coded in xml, compression should be applied. Some transport protocols offer compression features like transparent gzip

compression of http responses. However since http compression is only applicable to the response it is not usable for the compression of data loads in http requests.

In order to avoid this limitation a special XML based compression mechanism is described for ICAR data exchange. Specific messages can be defined as zip compressible offering the alternative of uncompressed or zip compressed Base64 binary encoded xml structures both for request and response. For details see chapter “Message Design” below.

5.5 User Right Verification

The right of a given user to use a given service is verified by either using an “**authentication token**” which prevents him from sending a user ID and a password at each request or by a **user name authenticated by a password**.

Authentication by user name and password should only be used with encrypted transport.

6 Communication work flow

6.1 Data Processing

Any service may be provided either in real time or in time differed.

A response should be sent to any request within several seconds.

Any request and any response should be stored by the equipment and the information system.

The requirements to update the data base of the information system differ from one service provider to another one.

Any request which is not in compliance with one the following conditions should not be processed:

- a. Compliance with the syntax
- b. Validity of the identification token
- c. Validity of the ticket if any

6.2 Request Specification

Any request should consist of three parts:

- a. A single **message header**
(see the entity MessageHeader of Type “ADEExchangedDocumentType” in ‘Data description’)
The sender of the request creates and provides his own unique MessageID (Item ADEExchangedDocumentType.Identifier).
See chapter “Best Praxis for MessageID Creation” below.
- b. A single **standard request**
(see the entity StandardRequest in ‘Data description’)
- c. A **specific request** which may consist of
a specific message for a given service
or the zip compressed base64 binary of the specific message
or a ticket.
A ticket is sent in order to retrieve the results of an asynchronous message processing task. (see diagrams below)

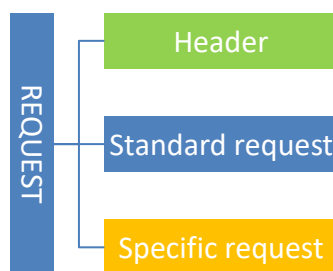


Figure 3 Request with a specific request

6.3 Response Specification

Any message response should consist of three parts:

- a. A single message header (see the entity MessageHeader of Type “ADEExchangedDocumentType” in ‘Data description’)

The sender of the response creates and provides his own unique Message id. (See chapter “Best Praxis for MessageID Creation” below)
- b. A single standard response (see the entity StandardResponse in ‘Data description’)

In the standard response the original MessageID from the sender is returned in the item RequestID.
The standard response contains the item RequestProcessingStatus which identifies the status of the request processing on the side of the service provider:

Table 8 RequestProcessingStatusCodeType (base xsd:string)

Key	Description
O	Processed without errors
P	Data accepted for asynchronous processing (client can retrieve the processing result later using provided ticket)
E	Processed and rejected due to errors
W	Processed with warnings

- c. A specific response which may consist of either a ticket or specific message for a given service or the zip compressed base64 binary of the specific message.

The existence and content of the specific response depends on the definition of the service and the content of the RequestProcessingStatus item.
In case of RequestProcessingStatus = ‘P’ the specific response holds a Ticket
If a specific response message is defined it can either be represented uncompressed or zip compressed of type Base64binary.
If RequestProcessingStatus = ‘P’, ‘E’ or ‘W’ there might be no specific response at all depending on the message definition.

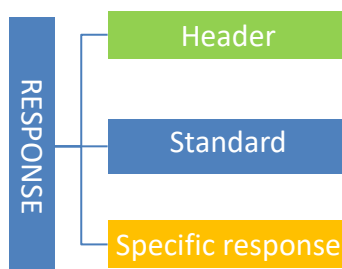


Figure 4 Response with a Specific Response

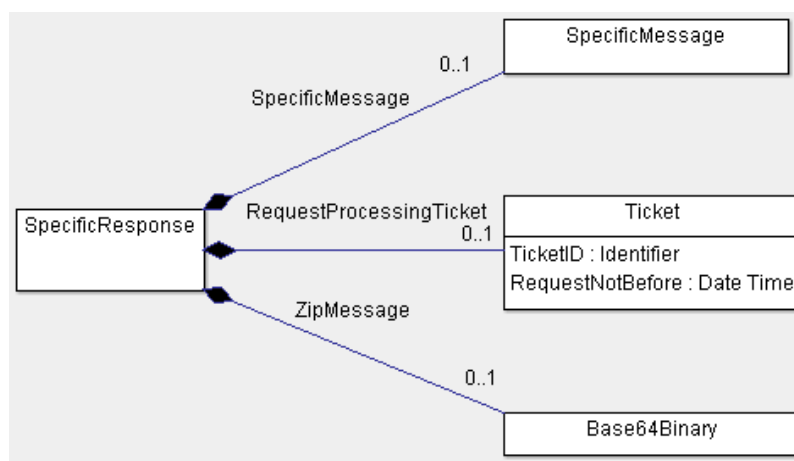


Figure 5 Specific Response Schema

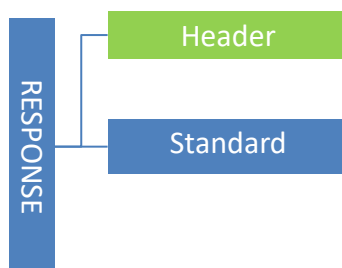


Figure 6 Standard Response

6.4 Best Praxis for Message-Id Creation

The message ID created by both service consumer and service provider should be globally unique in order to allow a simple identification of each message e.g. for tracing and debugging purposes.

The MessageID should be based on a chain of unique hardware identifier, a unique software identifier and an additional distinguishing component large enough to uniquely identify messages created at very fast rates. This dynamic component could be a timestamp with ms precision or a sequence. The unique hardware identifier could be the globally unique MAC address of the network interface. The unique software identifier should be unique on the system running the software.

There are tools in different programming languages which provide simple means of unique id creation e.g. Guid.NewGuid() in C#

7 Dealing with Local Requirements

The ICAR ADE standardization process cannot take into account all local requirements. In the first place it offers a common basis to build on.

In the first place “local” is a placeholder for national standards. However it is also possible to add local extensions based on an agreement between a service consumer and a service provider (e.g. for special projects, prototypes etc.).

In order to allow local extensions and to fill standardization gaps which cannot be covered by ICAR the ADE specifications provide two ways of local variations and extensions:

- a. **Local Code Lists:**
Local code lists for all items of type **udt:CodeType** for which ICAR is not yet able to provide a commonly accepted code set
- b. **Local Additional Data:**
An optional list of key value pairs in each specific entity to be used for locally required data elements

Local variation should be reduced to a minimum since it is contradictory to the goal of ICAR to establish global standards, leads to increased complexity, complicates software development and creates additional costs.

7.1 Local Code Lists

Local code lists must be provided for items of type **udt:CodeType** in the context of a local standardization process, e.g. breed codes.

The content cannot be checked with the SOAP/XML validation procedure but must be evaluated in an extra program step before filling the XML data structures and after extracting the data from the XML structures. It is the responsibility of the software developers to assure that the data exchange is in accordance to ICAR and to local specifications.

In order to help manufacturers with the maintenance and integration of local code lists these specifications define the service **GetLocalCodeList** which can be used to query local code lists from the local service providers. For details see section “Technical Services”.

7.2 Local Additional Data

Sometimes it is necessary for local implementations of the ICAR messages to add local data items to the specific messages which are not known or not covered by the ICAR specification process.

For those scenarios a simple and flexible extension (**LocalAdditionalData**) has been added to each specific entity.

It consists of an optional list of code/value pairs of type **udt:textType** which can be filled according to local definitions.

It is the responsibility of the local business process owners to document and assure the proper handling of this code/value pairs. The ICAR wsdl/xsd definitions only check the proper usage of the XML structure not the content.

For details see the wsdl/xsd files and the section “CommonComponents”-“LocalAdditionalDataType”.

8 Common Components

This part describes common components which are used in all the specific services described in annex B, C and D.

8.1 ADE Exchanged Document Type

The entity **ADEExchangedDocumentType** (Figure 7 ADEExchangedDocumentType) gives information describing general aspects of the message.

Used by data element: **MessageHeader**

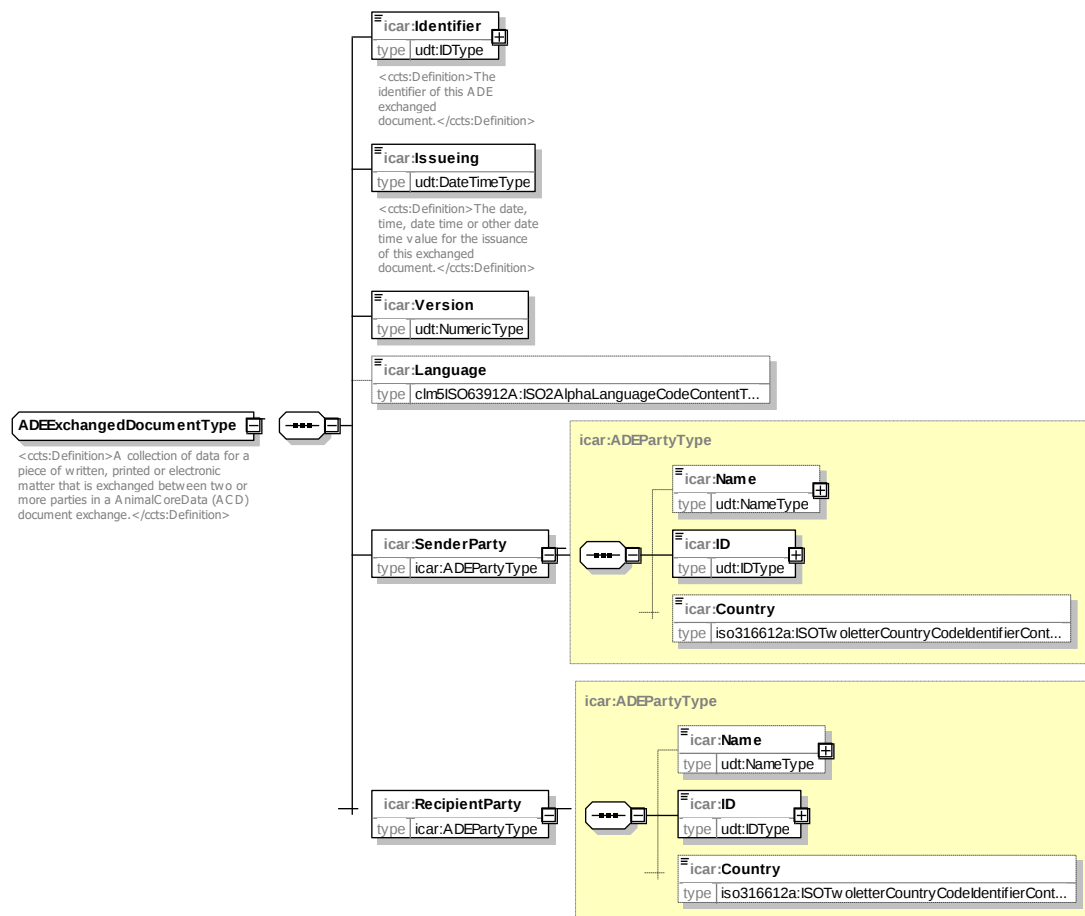


Figure 7 ADEExchangedDocumentType

- a. **Identifier:** Unique identifier of the message given by the sender
- b. **Issuing:** The date and the time where the message is issued.
- c. **Version:** Version number of the message
- d. **Language:** Language used for the message
- e. **LocalAdditionalData:** Optional list of key/value pairs used in a local context as described in section “Common Components – Local Adaptions”
- f. **SenderParty:** Organization or person responsible for the content of the message, not a server identifier.
- g. **RecipientParty:** Organization or person responsible for processing the message, not a server identifier.

8.2 ADE Party Type

The entity **ADEPartyType** (Figure 8 ADEPartyType) gives the name and the identifier of a party which may be either the sender or the recipient of a message.

Used by data elements: **SenderParty**, **RecipientParty**

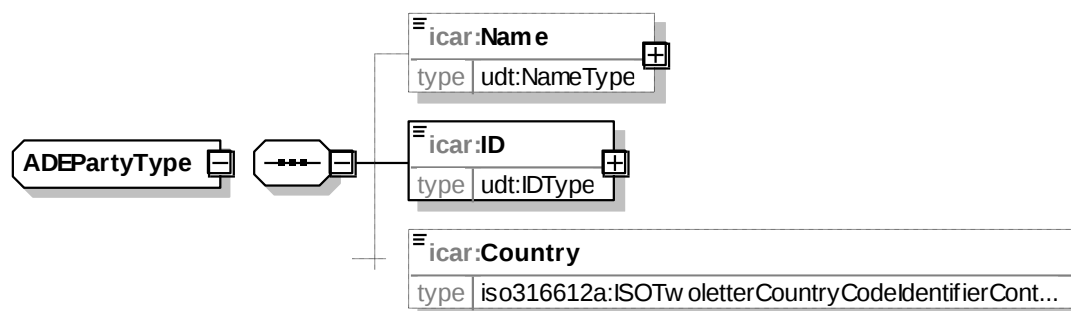


Figure 8 ADEPartyType

- **Name:** Identifier of the specified party
- **ID:** Identifier of the specified party
- **Country:** Country of the specified party in accordance to code list
ISOTwoletterCountryCodeIdentifierContent

8.3 Standard Request Type

The entity **StandardRequestType** (Figure 9 StandardRequestType) gives the standard content of a request.

Two different types of authentication can be used, token or user/password based.

Used by data element: **StandardRequest**

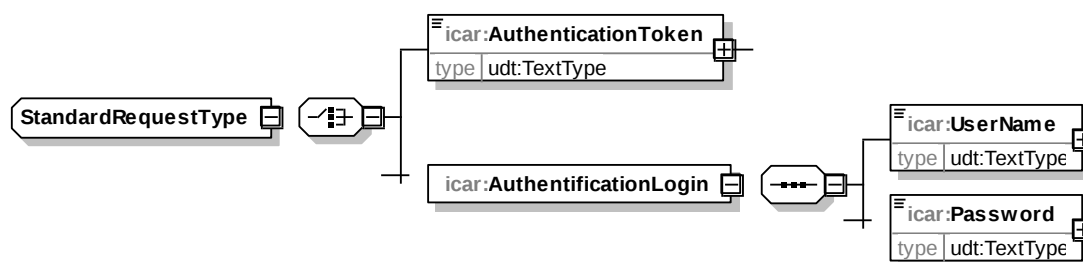


Figure 9 StandardRequestType

- a. **AuthenticationToken:** temporal code used for verification of the user's identity and of its right to use a service.
- b. **AuthenticationLogin:** Pair of username and password used for login
- c. **UserName:** User name used for authentication
- d. **Password:** Password used for authentication

8.4 Standard Response Type

The entity **StandardResponseType** (Figure 10 StandardResponseType) gives the standard content of a response.

Used by data element: **StandardResponse**

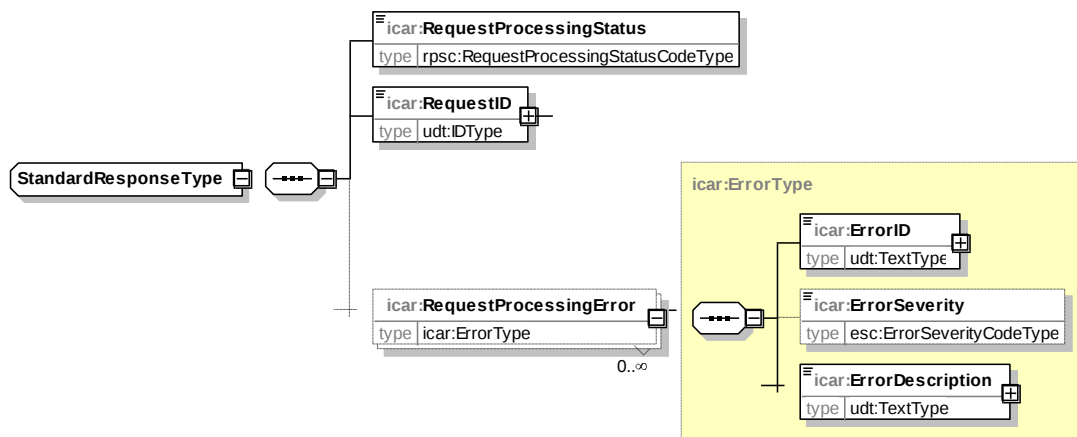


Figure 10 StandardResponseType

- a. **RequestProcessingStatus**: Status of request processing returned by server according to code list **RequestProcessingStatusCode**
- b. **RequestID**: Copy of the message identification in the **request's MessageHeader.Identifier** item
- c. **RequestProcessingError**: In case an error happens during the processing of the request an error description is provided here

8.5 Error Type

The entity **ErrorType** (Figure 11 ErrorType) gives the request processing errors.

Used by data element: **RequestProcessingError**

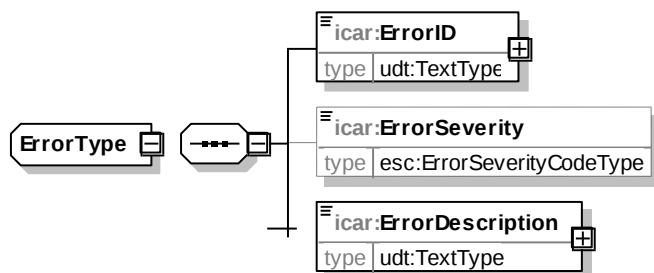


Figure 11 ErrorType

- a. **ErrorID**: Identifier the error, e.g. error text message, local error code etc.
- b. **ErrorSeverity**: Severity of the error according to code list **ErrorSeverityCode**
- c. **ErrorDescription**: Human readable description of the error as text

8.6 Ticket Response Type

The entity **TicketResponseType** (Figure 12TicketResponseType) conveys a ticket in order to retrieve a response processed in time differed (asynchronous processing mode).

It is used in **SpecificRequest** and **SpecificResponse** by data element **TicketResponse**.

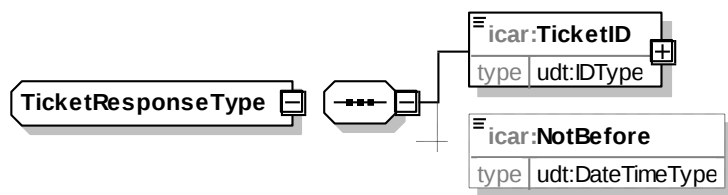


Figure 12 TicketResponseType

- a. **TicketID**: Identify the response to retrieve
- b. **NotBefore**: Date time of probable response availability

8.7 ZIP Message Type

ZipMessageType (Figure 13 ZipMessageType) is a wrapper type designed to transport the zip compressed version of the specific data part of a message. Contrary to http compression method which can only be used for the response part this method can be applied to request and response. It is defined as a choice for each specific request or response which transports data entities.

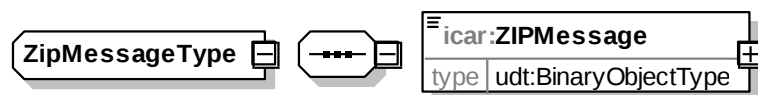


Figure 13 ZipMessageType

- a. **ZIPMessage**: zip compressed data

8.8 Local Additional Data Type

The entity **LocalAdditionalDataType** (Figure 14 LocalAdditionalDataType) is a container for code value pairs for local usage outside the ICAR specification scope.

It is used as an optional list element **LocalAdditionalData** in each specific entity.

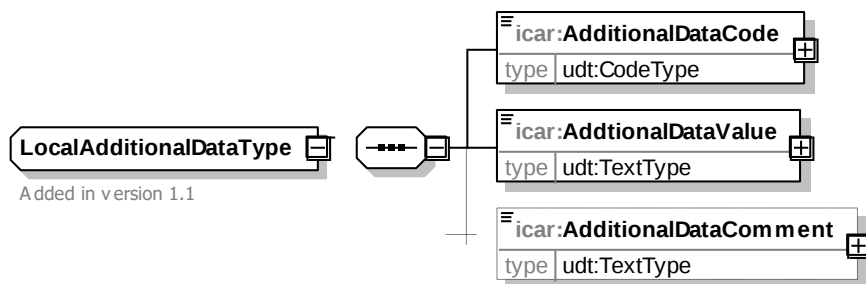


Figure 14 LocalAdditionalDataType

- a. **AdditionalDataCode**: Name of the code value pair
- b. **AdditionalDataValue**: Value of the code value pair
- c. **AdditionalDataComment**: Optional description of the local data

8.9 Time Period Type

The utility type **TimePeriodType** (Figure 15 TimePeriodType) is a wrapper type designed to transport the beginning and end of a time period.

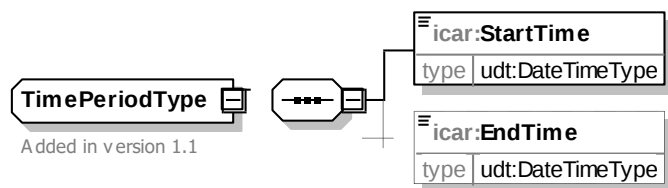


Figure 15 TimePeriodType

- a. **StartTime**: The start of the time period
- b. **EndTime**: The optional end of the time period

9 Naming Rules

Names of data elements, operations and messages use upper camel case.

Service name consist of:

- a. A prefix which gives the type of operation according the following:
 - Create: Insert data into a data base
 - Update: update of a data base
 - Get: retrieve data from data base
 - Delete: delete data from data base
- b. The name of the service

For example UpdateMilkingResult is the service which allows updating a data base from milking results exchange.

Message name consists of:

- a. The name of the service
- b. A suffix, gives the type of message:
 - Request
 - Response

For example, UpdateMilkingResultRequest is the request to trigger the service UpdateMilkingResult.

10 References

1. Semantics for Smart Dairy Farming: a milk production registration standard – SDF June 2013
2. UN / UNCEFACT Modeling Methodology User Guide (CEFACT / TMG/No93)
3. UN / UNCEFACT Business Requirements Specifications Document Template (CEFACT/ICG/005)
4. ISO 11787 : Electronic data interchange between information systems in agriculture – Agricultural data interchange syntax
5. ISO 11788 : Electronic data interchange between information systems in agriculture – Agricultural data element dictionary –Part 1: General description –Part 2: Dairy farming
6. ISO 17532 : Stationary equipment for agriculture –Data communications network for livestock farming
7. ISO 11784: Radio frequency identification of animals - code structure

8. ISO 3166 -1 : Country code
9. ICAR G



THE GLOBAL STANDARD
FOR LIVESTOCK DATA

Procedure 2 of Section 15 of ICAR Guidelines – Milking Data Services

Procedure 2 – Milking Data

Version July, 2018

File Ref: 15 Procedure 2 Milking Data v18.01.dotx

Network. Guidelines. Certification.

Table of Contents

1	Introduction	4
2	Definitions and Terminology	4
3	Scope	4
4	Common Components	5
4.1	MilkCharacteristicsType	5
4.2	MilkCharacteristicsLocalType	5
5	Exchange of Milking Event Data	6
5.1	Purpose	6
5.2	Request description	6
5.3	AnimalMilkingResults	6
5.4	Characteristics	7
5.5	AnimalMilkingSample	7
5.6	QuarterMilking	7
5.7	Response description	8
6	Exchange of Milk Analysis Result Data	9
6.1	Purpose	9
6.2	Request Description	9
6.3	Response Description	9
6.4	AnimalMilkingLabResult	10
6.5	Characteristics	11
6.6	AnimalMilkingSample	11
6.7	QuarterMilking	11

Tables

Table 1. Definitions of Terms used in these guidelines.....	4
---	---

Figures



Figure 1. Scope of Guideline.....	4
Figure 2 MilkCharacteristicsType	5
Figure 3 MilkCharacteristicsLocalType	5
Figure 4 UpdateMilkingResultRequest.....	6
Figure 5 AnimalMilkingSample.....	7
Figure 6 QuarterMilking	7
Figure 7 UpdateMilkingResultRsponse	8
Figure 8 GetMilkingLabResultsRequest.....	9
Figure 9 MilkingLabResultRequestType	9
Figure 10 GetMilkingLabResultsResults	9
Figure 11 QuarterMilking	11

Summary of Changes

Date of Change	Nature of Change
July 2018	New procedure created from methodology chapter in Section 15 Overview. Replaced by new version from ADE-WG. Template applied.

1 Introduction

The milking, milk sampling and milk analysis process is a fundamental part of dairy performance recording. Automatic, fast and reliable data exchange is a pre-requisite for effective monitoring of dairy herds.

The business process “Exchange of Milking Data” is designed to

- a. Exchange data linked to a milking event
- b. Exchange data linked to milk analysis

It provides several methods designed to help synchronizing the equipment data base and the farm management database without efforts:

- a. **UpdateMilkingResult:** Transfer details on milking events, milk sampling and on farm measurements
- b. **GetMilkingLabResults:** Request the results of a milk sample analysis done in a laboratory

The service descriptions follow the principals described in “Guideline 1 Methodology” - [link here](#) - and reuse the basic elements defined there.

The complete description of the codelists referenced here can be found in Appendix A ([link here](#)) data dictionary.

2 Definitions and Terminology

Table 1 contains a list of important definitions for terms and abbreviations used in these guidelines.

Table 1. Definitions of Terms used in these guidelines.

Term	Definition

3 Scope

Figure 1 gives a pictorial summary of the main elements of this guideline. The numbers in this figure refer to the heading numbers of this procedure.

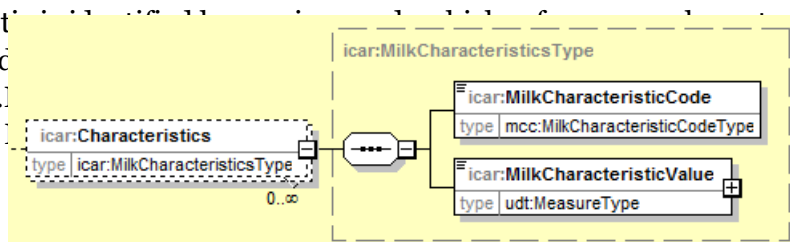
Figure 1. Scope of Guideline.

4 Common Components

4.1 MilkCharacteristicsType

The component Characteristics conveys a list of milking characteristics. A characteristic of type MilkCharacteristicsType is used here as a synonym for the results of a measurement or a classification applied to a milk sample, on farm or in a laboratory. It can also be used for recording parameters describing of the environmental situation or the status of the cows during the milking process, e.g. weight of the animal during milking.

The characteristic according to code section “Annex. characteristics.

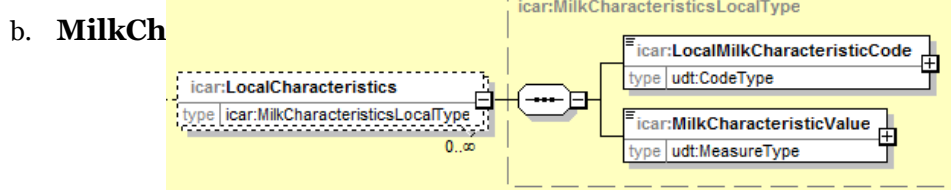


- a. **MilkCharacteristicCode:** The mandatory identifier of a milk characteristic.
- b. **MilkCharacteristicValue:** A milk characteristic measure value. The attribute “unitCode” must be set according to the table “Milk Characteristics Codes”

4.2 MilkCharacteristicsLocalType

Similar to MilkCharacteristicsType this container conveys a list of locally used milking characteristics which are not part of the official “ICAR Milk Characteristics Codes” list.

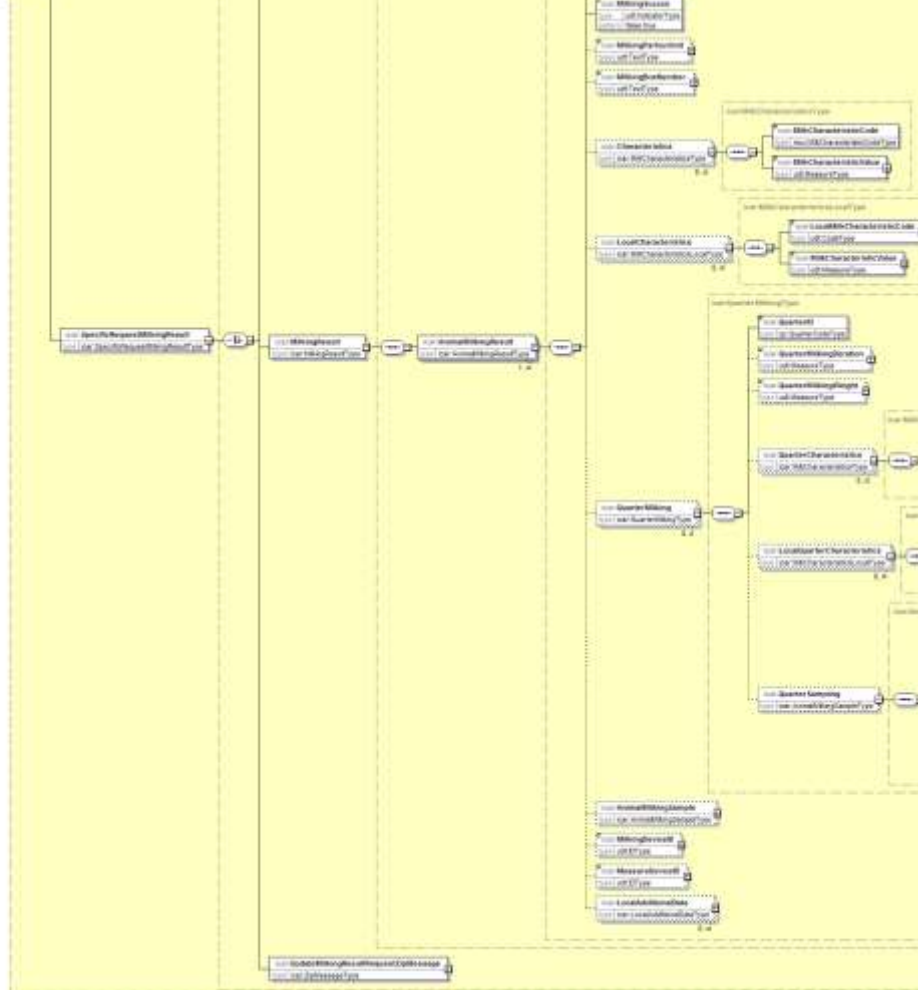
- a. **LocalMilkCharacteristicCode:** The mandatory identifier of a local milk character



5

5.1

5.2



system:

to

e general

provide.

5.3 AnimalMilkingResults

AnimalMilkingResult conveys the main feature of a given milking:

- AnimalIdentity:** One unique animal identifier used for data exchange holding id and label of the milked animal. The ID optionally references entity AnimalDetail transported by Service “UpdateAnimal” if required for further animal details.
- Location:** One milking location ID identifying the location where the animal has been milked. The ID optionally references an entity LiveStockLocation transported by Service UpdateLiveStockLocation if required for further location details.
- MilkingParlourUnit, MilkingBoxNumber:** optionally identify the location of milking more detailed
- MilkingStartingTime, MilkingDuration, MilkingVisitDuration, MilkingType, MilkingMilkWeight, MilkingSuccess:** Items describing the milking itself:
- Characteristics:** Milk characteristics (classification or measurements) which are conveyed by the entity **MilkCharacteristics**. In principal it is a list of key value pairs with keys identifying the type of characteristics and the values holding the result. The current list of characteristics key definitions is shown in table “Milk Characteristic Codes”
- LocalCharacteristics:** Locally defined milk characteristics (classification or measurements) which are conveyed by the entity LocalCharacteristics. It conveys a list of locally used milking characteristics which are not part of the official “ICAR Milk Characteristics Codes” list.

- g. **QuarterMilking**: Up to four optional entities quarter milking results, optionally including a quarter milk samples and quarter milk characteristics. This result may not be provided by every equipment.
- h. **AnimalMilkingSample** : One optional entity describing the details of a sample of milk taken for laboratory analyses
- i. **MilkingDeviceID**: One optional item holds an id of the device used for milking. The ID optionally references an entity Device transported by Service “UpdateDevice” if required for further device details.
- j. **MeasureDeviceID**: One optional item holds an id of the device used for measuring. The ID optionally references an entity Device transported by Service “UpdateDevice”

5.4 Char

The

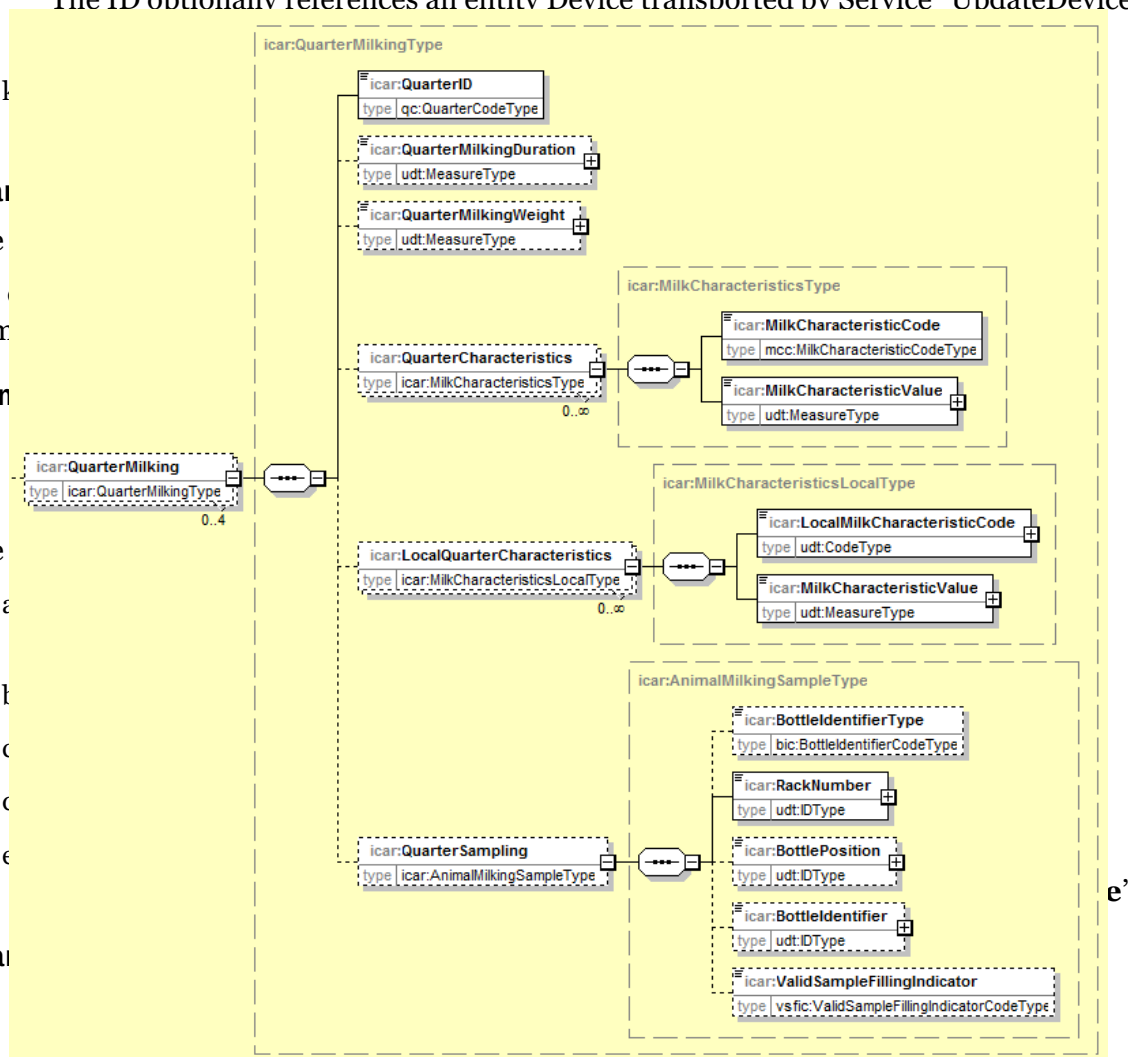
For

Com

5.5 Anin

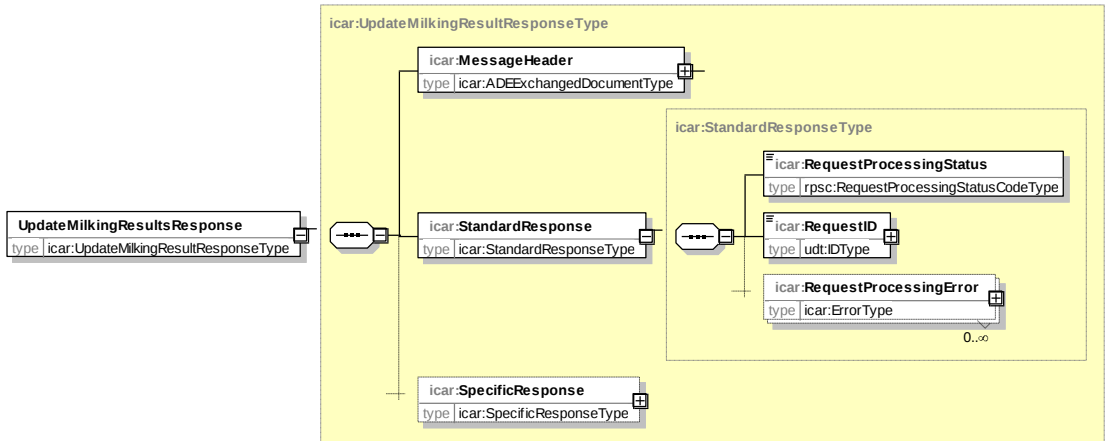
The

5.6 Qua



The entity QuarterMilking gives the milking result and optionally sampling and characteristic details for one of four given quarters.

- a. **QuarterID**: Identification of the quarter for which the results apply (1..4)
- b. **QuarterMilkingDuration**: Milking duration of the quarter, default unit seconds
- c. **QuarterMilkingWeight**: Milk weight of the quarter, default unit kg



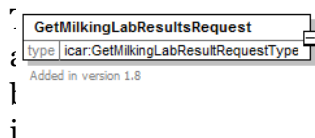
For
of

5.7 Res

6 Exchange of Milk Analysis Result Data

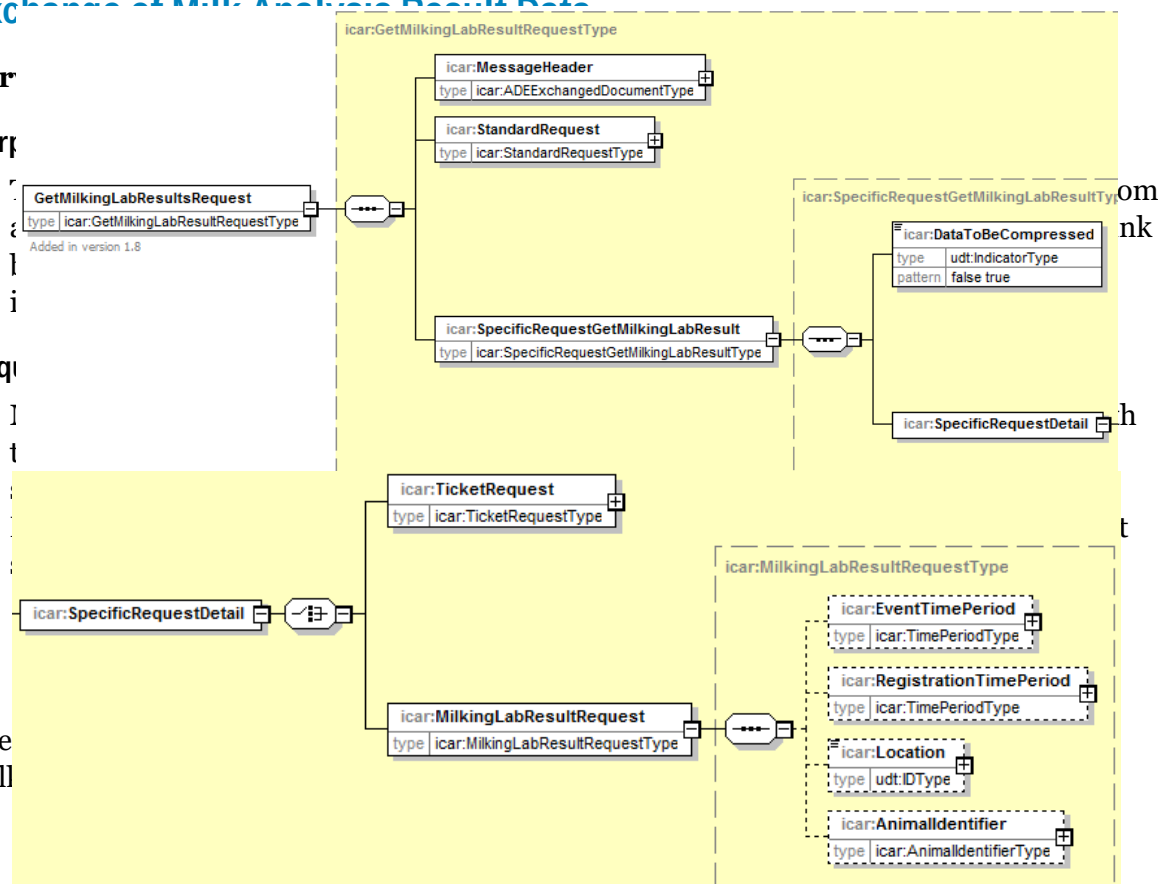
Service

6.1 Purpose



6.2 Requirements

The Milking



- Location:** The identification of the farm for which laboratory results are requested
- AnimalIdentifier:** The animal id for which laboratory results are requested
- EventTimePeriod:** Period of sampling moments at which the requested results have been sampled
- RegistrationTimePeriod:** Period of registration moments at which the requested results have been registered at the farm management system

It is the responsibility of the farm management system to define the requirements and the valid ranges for these parameters.

6.3 Response Description

Message GetMilkingLabResultsResults (see diagram below) is in compliance with the general specifications for results (see General specifications / Response specifications)

It is used to receive the results of a laboratory milk sample analysis stored by the service provider. Since the registration of a milk sample at a farm management system or a laboratory is done by the entity AnimalMilkingSample its returned data in entity AnimalMilkingLabResults is very similar to this entity. Instead of characteristics recorded on the equipment however characteristics based on a laboratory analysis are returned.

Figure 10 GetMilkingLabResultsResults

It conveys a list of locally used milking characteristics which are not part of the official “ICAR Milk Characteristics Codes” list.

- f. **QuarterMilking**: For analysis results of quarter specific samples. Up to four optional entities quarter milking results, optionally including a quarter milk sample and quarter milk analysis characteristics.
- g. **AnimalMilkingSample** : One optional entity describing the details of a sample of milk taken for laboratory analyses
- h. **MilkingDeviceID**: One optional item holds an id of the device used for milking. The ID optionally references an entity Device transported by Service “UpdateDevice” if required for further device details.
- i. **MeasureDeviceID**: One optional item holds an id of the device used for measuring. The ID optionally references an entity Device transported by Service “UpdateDevice” if required for further device details.
- j. **MeasureDeviceID**: One optional item holds an id of the device used for measuring. The ID optionally references an entity Device transported by Service “UpdateDevice” if required for further device details.
- k. **LabAnalyzerModelCode**: Specifies the model of the milk analyzing instrument
- l. **LabAnalysisDateTime**: Date and time of the analysis of this sample.
- m. **LocalAdditionalData**: Optional, unlimited number of locally used key/value pairs

6.5 Characteristics

The entity conveys the milk characteristics produced by a laboratory analysis of a given milking. The characteristics is identified by a unique code which references a characteristics definition in the table “Milk Characteristics Codes“, a basic set of ICAR defined characteristics. It can easily be extend by special laboratory specific definitions.

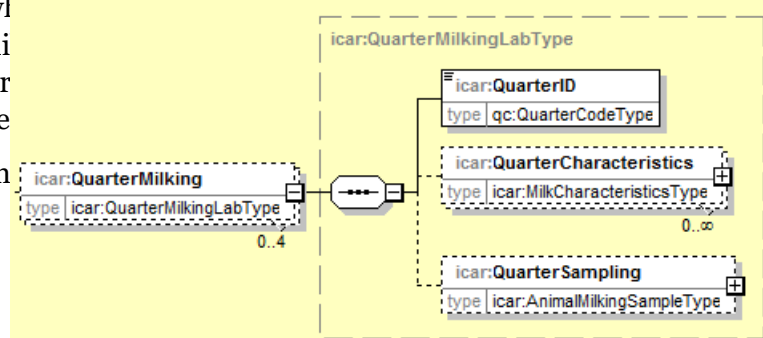
For details see the description of “MilkCharacteristicsType” in section “Common Components” above.

6.6 AnimalMilkingSample

The entity AnimalMilkingSample conveys details of the sampling of the milk on the equipment for which the data given in this helps the results provided by the farm manager.

For details see the description of “AnimalMilkingSampleType” in section “Common Components” above.

6.7 QuarterMilking



The entity **QuarterMilking** of type **QuarterMilkingLabType** gives the laboratory analysis result and sampling details for one of four given quarters.

- a. **QuarterID**: Identification of the quarter for which the results apply (1..4)

- b. **QuarterCharacteristics:** Milk characteristics of the quarter. For details see description of entity Characteristics above.
- c. **QuarterSampling :** One optional entity describing the details of a sample of milk taken from the quarter for laboratory analyses. For details see description of entity AnimalMilkingSample above



THE GLOBAL STANDARD
FOR LIVESTOCK DATA

Procedure 3 of Section 15 of ICAR Guidelines – Animal Data

Procedure 3 – Animal Data

Version July, 2018

File Ref: 15 Procedure 3 Animal Data v18.01.docx

Network. Guidelines. Certification.

Table of Contents

1	Introduction	5
2	Definitions and Terminology	5
3	Scope	5
4	Common Components	6
4.1	AnimalIdentityType.....	6
4.2	AnimalCoreDataSetType	6
4.3	ArrivalCoreDataSetType	9
4.4	DepartureCoreDataSetType.....	10
4.5	AnimalDescriptionType	11
4.6	MovementRequest	11
5	Exchange of Herd List	12
5.1	Service Description: GetHerdList.....	12
5.1.1	Purpose.....	12
5.1.2	Request description	12
5.1.3	Response description	13
5.2	Service Description: UpdateHerdList	14
5.2.1	Purpose.....	14
5.2.2	Request description	14
5.2.3	Response description	15
6	Exchange of Animal Core Information.....	16
6.1	Service Description: GetAnimal	16
6.1.1	Purpose.....	16
6.1.2	Request description	16
6.1.3	Response description	16
6.2	Service Description: UpdateAnimal	17
6.2.1	Purpose.....	17
6.2.2	Request description	17
6.2.3	Response description	19
7	Exchange of Animal Arrival in Herd	20
7.1	Service Description: GetArrivals.....	20
7.1.1	Purpose.....	20
7.1.2	Request description	20
7.1.3	Response description	21
7.2	Service Description: UpdateArrivals	22
7.2.1	Purpose.....	22
7.2.2	Request description	22
7.2.3	Response description	24
8	Exchange of Animal Departure from Herd.....	24
8.1	Service Description: GetDepartures	24
8.1.1	Purpose.....	24
8.1.2	Request description	24
8.1.3	Response description	25
8.2	Service Description: UpdateDepartures.....	27

8.2.1	Purpose.....	27
8.2.2	Request description	27
8.2.3	Response description	28

Tables

Table 1.	Definitions of Terms used in these guidelines.....	5
----------	--	---

Equations

No table of figures entries found.

Figures

Figure 1.	Scope of Guideline.	6
Figure 2	AnimalIdentityType	6
Figure 3	AnimalCoreDataType	7
Figure 4	ArrivalCoreDataSetType	9
Figure 5	DepartureCoreDataSetType	10
Figure 6	AnimalDescriptionType	11
Figure 7	MovementRequest.....	12
Figure 8	GetHerdListRequest.....	12
Figure 9	HerdListRequest.....	13
Figure 10	GetHerdListResponse	14
Figure 11	UpdateHerdListRequest	15
Figure 12	UpdateHerdListResponse	16
Figure 13	GetAnimalRequest	16
Figure 14	GetAnimalResponse	17
Figure 15	UpdateAnimalResponse	18
Figure 16	AnimalListMessageType.....	19
Figure 17	UpdateAnimalResponse	20
Figure 18	GetArrivalRequest	20
Figure 19	GetArrivalResponse	21
Figure 20	ArrivalDataSet.....	22
Figure 21	UpdateArrivalRequest	23
Figure 22	ArrivalDataSet	23
Figure 23	UpdateArrivalResponse	24
Figure 24	GetDepartureRequest	25
Figure 25	GetDepartureResponse	26
Figure 26	DepartureDataSet.....	26
Figure 27	UpdateDepartureRequest	27
Figure 28	DepartureDataSet.....	28
Figure 29	UpdateDepartureResponse.....	28

Summary of Changes

Date of Change	Nature of Change
July 2018	New procedure created from methodology chapter in Section 15 Overview. Replaced by new version from ADE-WG. Template applied.

1 Introduction

A carefully maintained list of animals in the herd is a prerequisite of reliable animal recording. The animal lists on the equipment and on the farm management system are likely to be different because of:

- a. Different input sources (manual maintenance by farmer, legal registration in the farm management system)
- b. Different availability of data

The business process “Exchange of Animal Data” is designed to

- c. initialize the database of the equipment or herd management system
- d. propagate new or changed data to or from the farm management system
- e. help detecting data conflicts earlier

It provides several methods designed to help synchronizing the equipment data base and the farm management database without efforts:

- a. **GetHerdList**: Initialize animal lists on the equipment or herd management system
- b. **GetAnimal**: Request animal descriptions from the service provider
- c. **UpdateAnimal**: Send animal description changes to the service provider
- d. **GetArrival**: Request animal arrival changes from the service provider
- e. **UpdateArrival**: Send animal arrival changes to the service provider
- f. **GetDeparture**: Request animal departure changes from the service provider
- g. **UpdateDeparture**: Send animal departure changes to the service provider

The service descriptions follow the principals described in “Guideline A Methodology” and reuse the basic elements defined there.

The complete description of the codelists referenced here can be found in the Annex document.

2 Definitions and Terminology

Table 1 contains a list of important definitions for terms and abbreviations used in these guidelines.

Table 1. Definitions of Terms used in these guidelines.

Term	Definition

3 Scope

Figure 1 gives a pictorial summary of the main elements of this procedure. The numbers in this figure refer to the heading numbers of this procedure.

Figure 1. Scope of Guideline.



4 Common Components

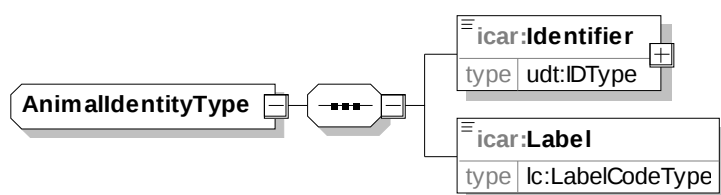
4.1 AnimalIdentityType

The entity gives official (official ear tag numbers etc.) and alternative animal identification used by farmers, equipment (farm number, necklace number, animal name etc.)

Only official identifiers should be used as primary identifier for data exchange. Alternative identifiers are used to detect changes of names or on farm numbers in addition to the AnimalID. At robot level farmers often work with on farm numbers or names which can be reused for other animals and cause errors in the identification of samples.

Alternative identification should be never used as the primary identifier for data exchange.

Figure 2 AnimalIdentityType

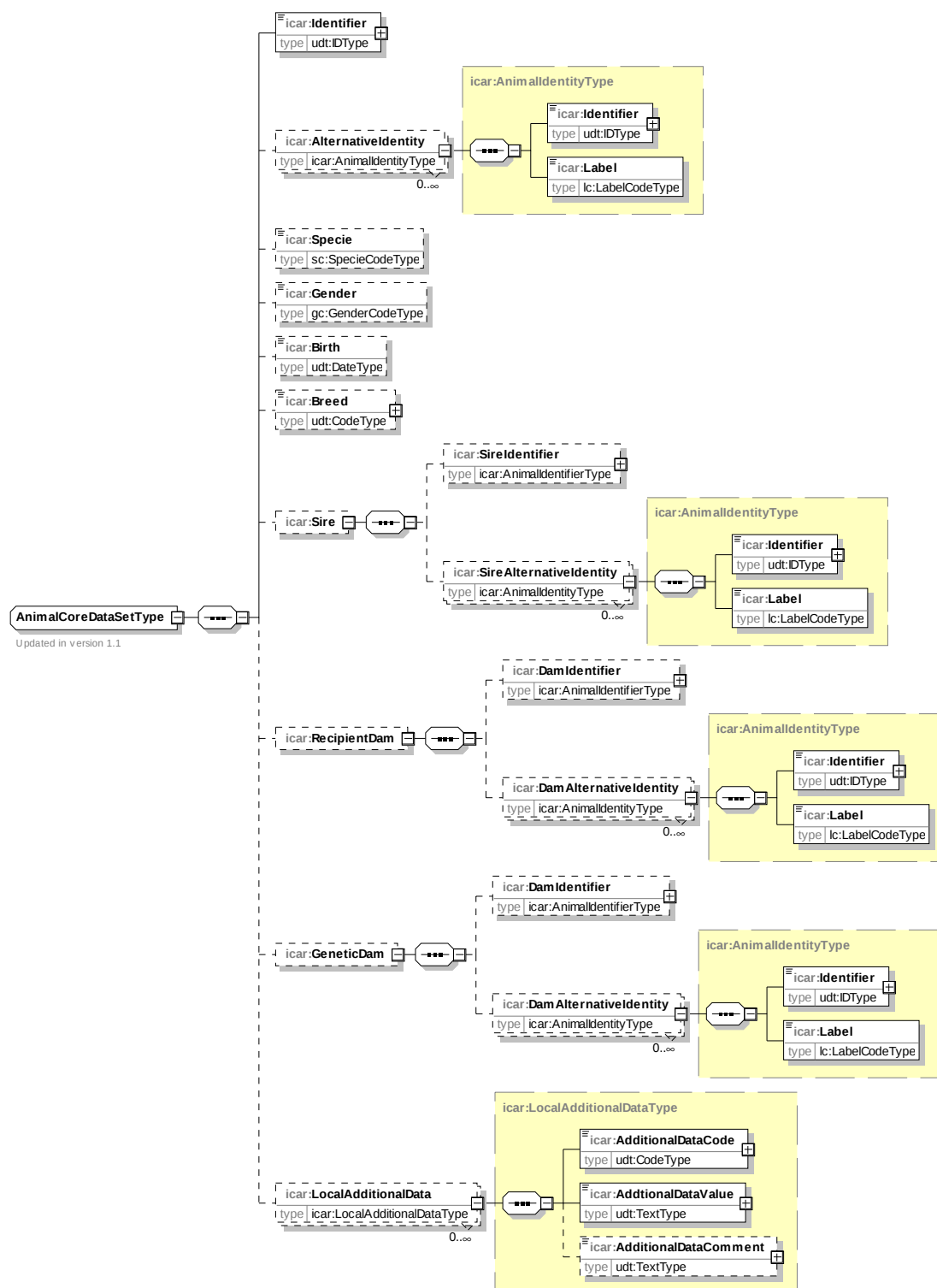


- **Identifier:** Alternative animal identifier
- **Label:** Type of alternative animal identifier according to code list **LabelCode**

4.2 AnimalCoreDataSetType

The entity gives the core characteristics of a given animal.

Figure 3 *AnimalCoreDataSetType*



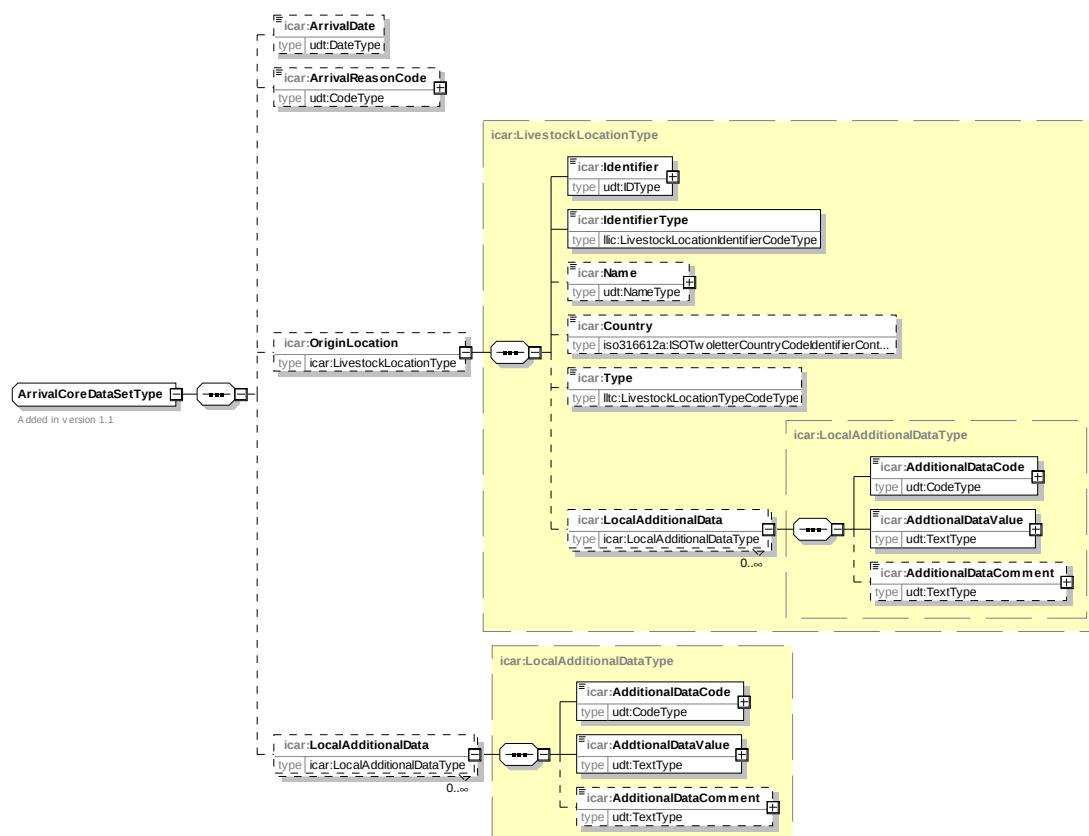
- a. **Identifier:** primary unique identifier of an animal
- b. **AlternativeIdentity:** secondary animal identifiers
- c. **Specie:** Specie of the animal
- d. **Gender:** Gender of the animal
- e. **Birth:** Birth date of the animal

- f. **Breed:** Local breed code of the animal
- g. **Sire:** Sire description entity
 - **SireIdentifier:** Animal identifier of the sire
 - **SireAlternativeIdentity:** Alternative animal identifiers of the sire
- h. **RecipientDam:** receiving dam description entity in case of embryo transfer
 - **RecipientDamIdentifier:** Animal identifier of the receiving dam
 - **RecipientDamAlternativeIdentity:** Alternative animal identifiers of the receiving dam
- i. **GeneticDam:** genetic dam description entity in case
 - **GeneticDamIdentifier:** Animal identifier of the genetic dam
 - **GeneticDamAlternativeIdentity:** Alternative animal identifiers of the genetic dam

4.3 ArrivalCoreDataSetType

The entity gives the core characteristics of an animal arrival to a location.

Figure 4 ArrivalCoreDataSetType

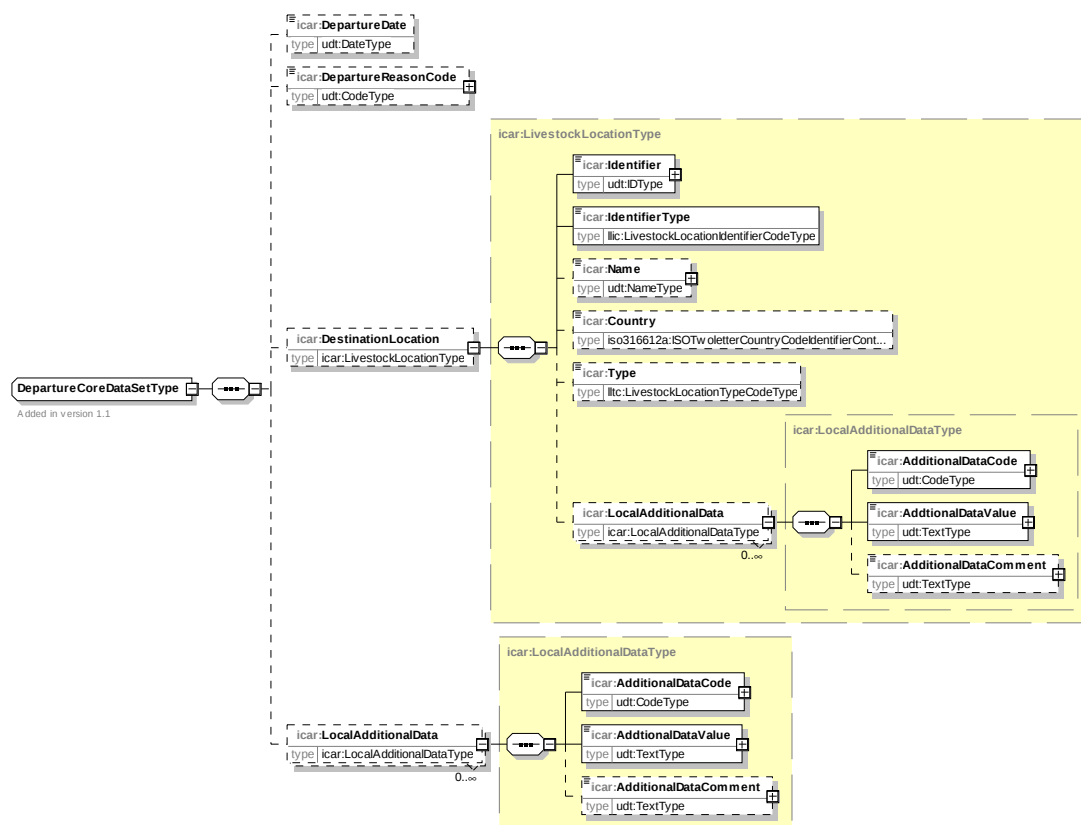


- a. **ArrivalDate**: Date of the arrival of the animal
- b. **ArrivalReasonCode**: local code specifying the reason of the arrival
- c. **OriginLocation**: The location the animal is arriving from
- d. **LocalAdditionalData**: List of locally defined code/value pairs

4.4 DepartureCoreDataSetType

The component gives the core characteristics of an animal departure from a location.

Figure 5 DepartureCoreDataSetType

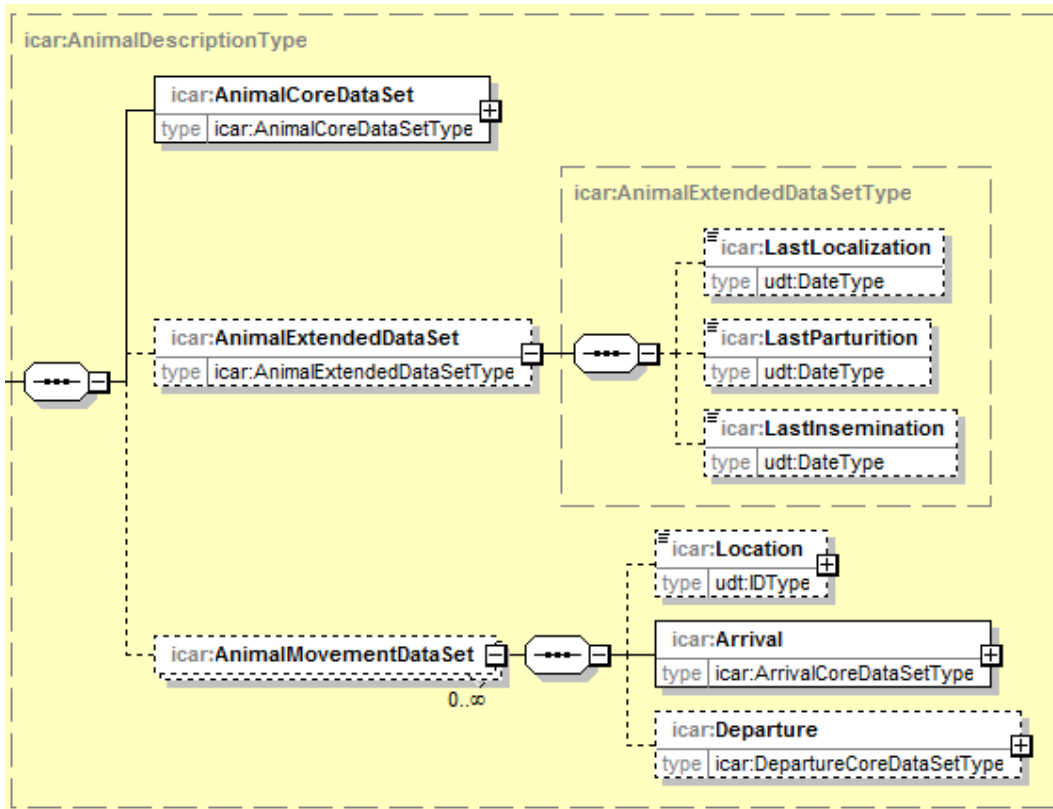


- DepartureDate**: Date of the departure of the animal
- DepartureReasonCode**: local code specifying the reason of the departure
- DestinationLocation**: The location the animal is departing to
- LocalAdditionalData**: List of locally defined code/value pairs

4.5 AnimalDescriptionType

The component conveys the description of an animal. Currently it comprises core data, extended data and movement data.

Figure 6 AnimalDescriptionType

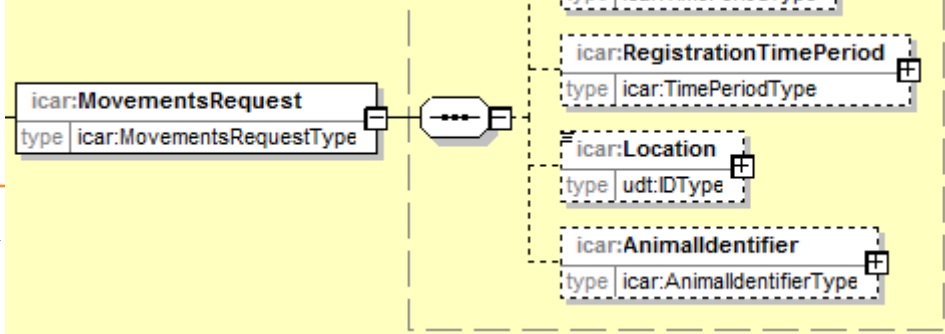


- a. **AnimalCoreDataSet**: Core description of an animal (for details see entity description in section “Common Components.AnimalCoreDataSetType” above)
- b. **AnimalExtendedDataSet**: Conveys the optional elements
 - **LastLocalization**: date of last arrival of the animal on the location
 - **LastParturition**: last recorded parturition of the animal
 - **LastInsemination**: last recorded insemination of the animal
- c. **AnimalMovementDataSet**: List of the animal’s movements
 - **Location**: A location on which the animal has been registered
 - **Arrival**: Conveys the arrival details of the animal on this location (see entity description in section “Common Components.ArrivalCoreDataSetType” above)
 - **Departure**: Conveys the departure details of the animal on this location (see entity description in section “Common Components.DepartureCoreDataSetType” above)

4.6 MovementRequest

The entity **MovementsRequest** (see diagram below) is designed to provide optional parameters which can be used to control the size of the result of a movement response.

It is e.g. used



- EventTimePeriod:** Limit the extend of the response to the given period of arrival event
(see entity description in section “Common entities”, ”TimePeriodType” above)
- RegistrationTimePeriod:** Limit the extend of the response to the given period of the registration of arrival events
(see entity description in section “Common entities”, ”TimePeriodType” above)
- Location:** If provided the extend of the response will be limited to the animals registered for this (sub)location. With the parameter missing the service provider has to deduce the location from the login credentials or can reject the request.
- AnimalIdentifier:** If provided the extend of the response will be limited to the arrivals of the given animal.

5 Exchange of Herd List

5.1 Service Description: GetHerdList

5.1.1 Purpose

To allow the owner of the equipment to retrieve the animal herd list including animal movement data and recent reproduction events like last calving and last insemination.

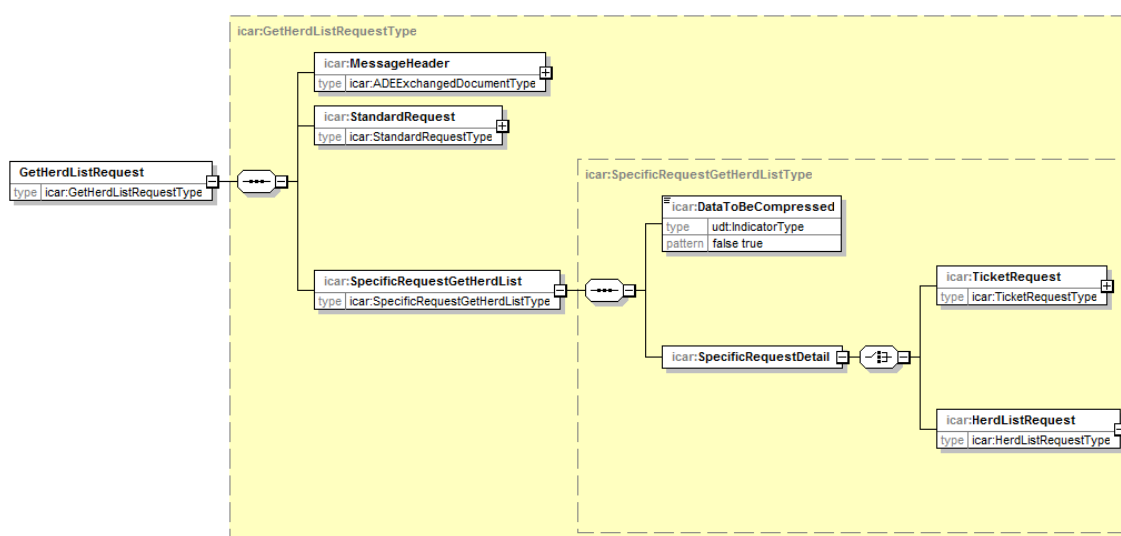
The result size can be narrowed by providing request parameters like location, gender and a period of time.

5.1.2 Request description

Message GetHerdListRequest (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

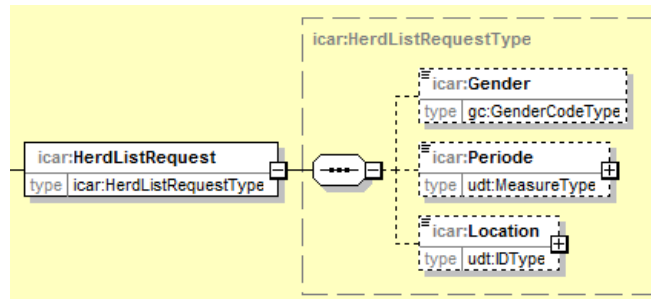
It is used to request the herd list for a given location stored in the farm management system.

Figure 8 GetHerdListRequest



The extend of the result can be controlled by providing optional parameters (see diagram below):

Figure 9 HerdListRequest

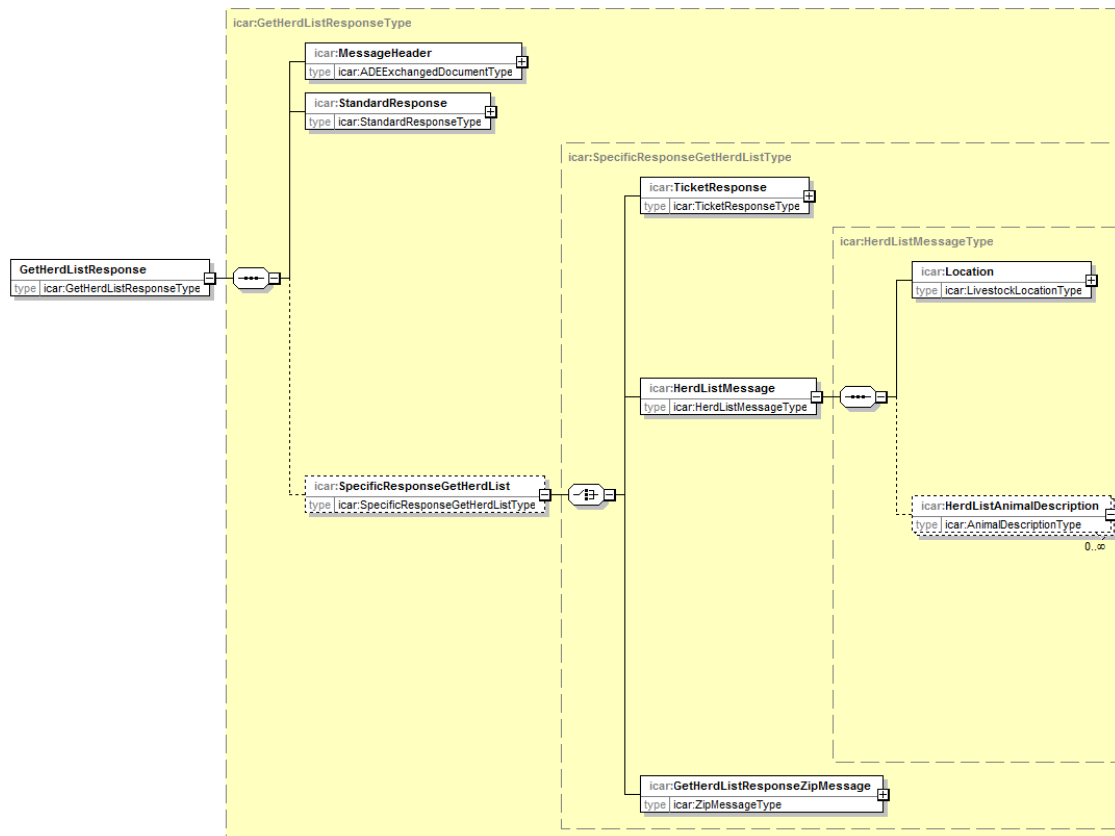


- a. **Gender:** restrict herd list results to male or female animals
- b. **Periode:** If provided the result will include also data from the herd's history within the period range to be interpreted as days. With the parameter missing or of value 0 the herd list result will convey only the animals currently in the herd
- c. **Location:** If provided the herd list result will be restricted to the animals registered for this (sub)location. With the parameter missing the service provider has to deduce the location from the login credentials or can reject the request.

5.1.3 Response description

Message **GetHerdListResponse** (see diagram below) is in compliance with the general specifications for results (see General specifications / Response specifications)

Figure 10 GetHerdListResponse



The result data container **HerdListMessage** conveys the **Location** descriptor and a list of animal descriptions in **HerdListAnimalDescription** of type **AnimalDescription** (for details see entity description in section “**Common Components.AnimalDescriptionType**” above).

5.2 Service Description: UpdateHerdList

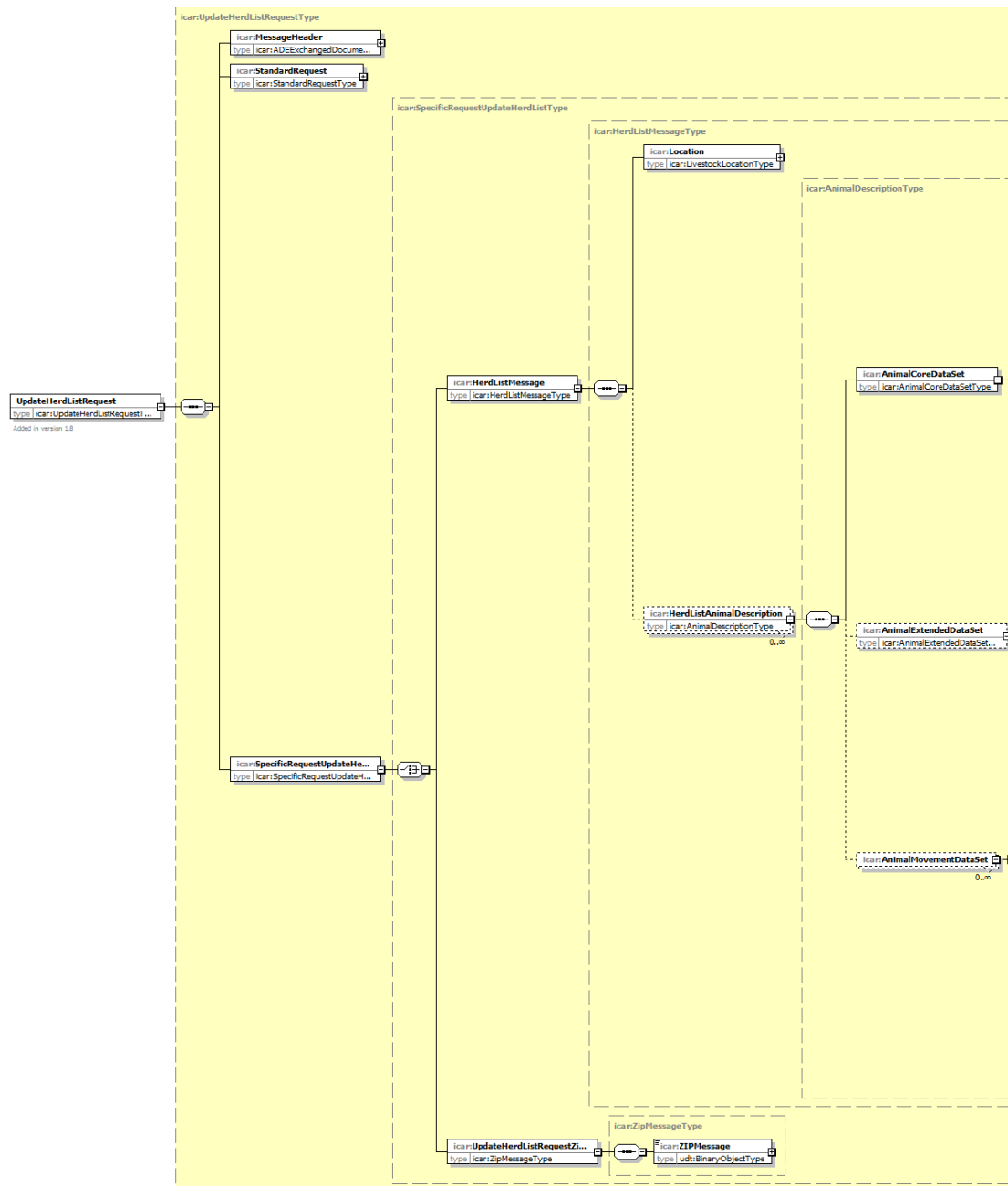
5.2.1 Purpose

To allow the owner of the equipment to send changes in the animal herd list including animal movement data and recent reproduction events like last calving and last insemination to the service provider.

5.2.2 Request description

Message **UpdateHerdListRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

Figure 11 UpdateHerdListRequest



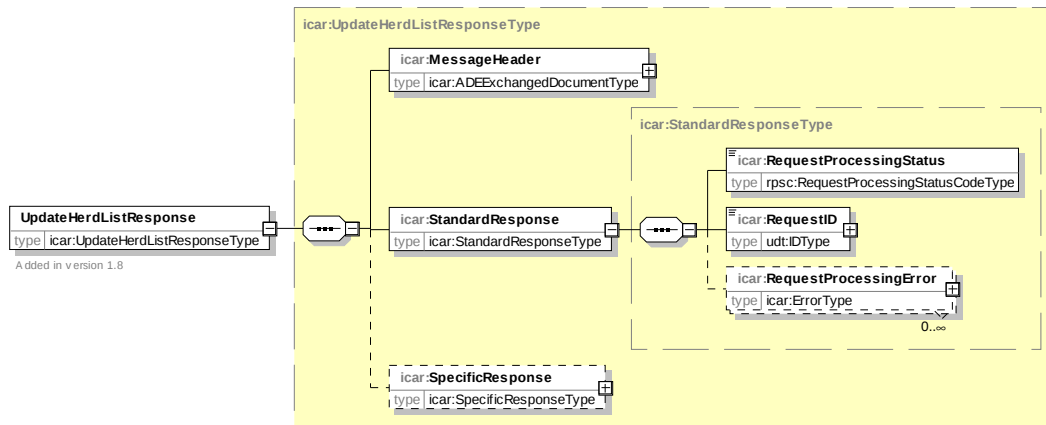
The container **HerdListMessage** conveys the **Location** descriptor and a list of animal descriptions to be processed at the service provider in **HerdListAnimalDescription** of type **AnimalDescription** (for details see entity description in section “**Common Components.AnimalDescriptionType**” above).

5.2.3 Response description

Message **UpdateHerdListResponse** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Response specifications)

It is used to receive the status of herd list data processing by the service provider.

Figure 12 UpdateHerdListResponse



6 Exchange of Animal Core Information

6.1 Service Description: GetAnimal

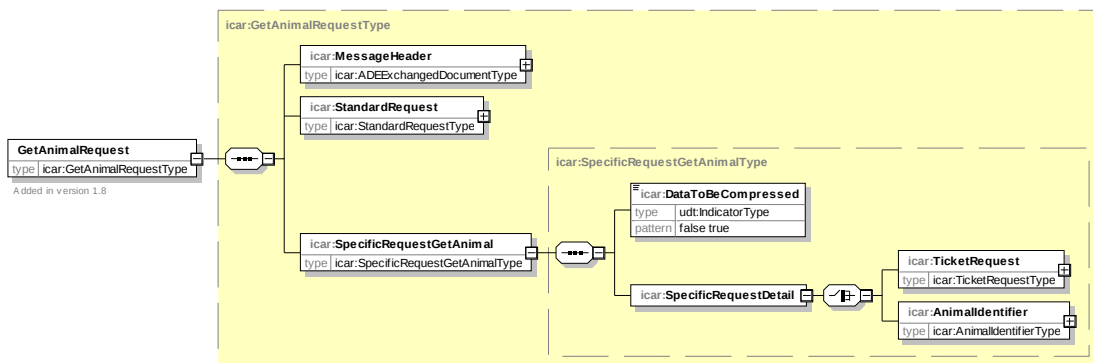
6.1.1 Purpose

The service **GetAnimal** allows the equipment to retrieve animal core information from the farm management system.

6.1.2 Request description

Message **GetAnimalRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications). It is used to request the core data describing an animal stored in the farm management system.

Figure 13 GetAnimalRequest



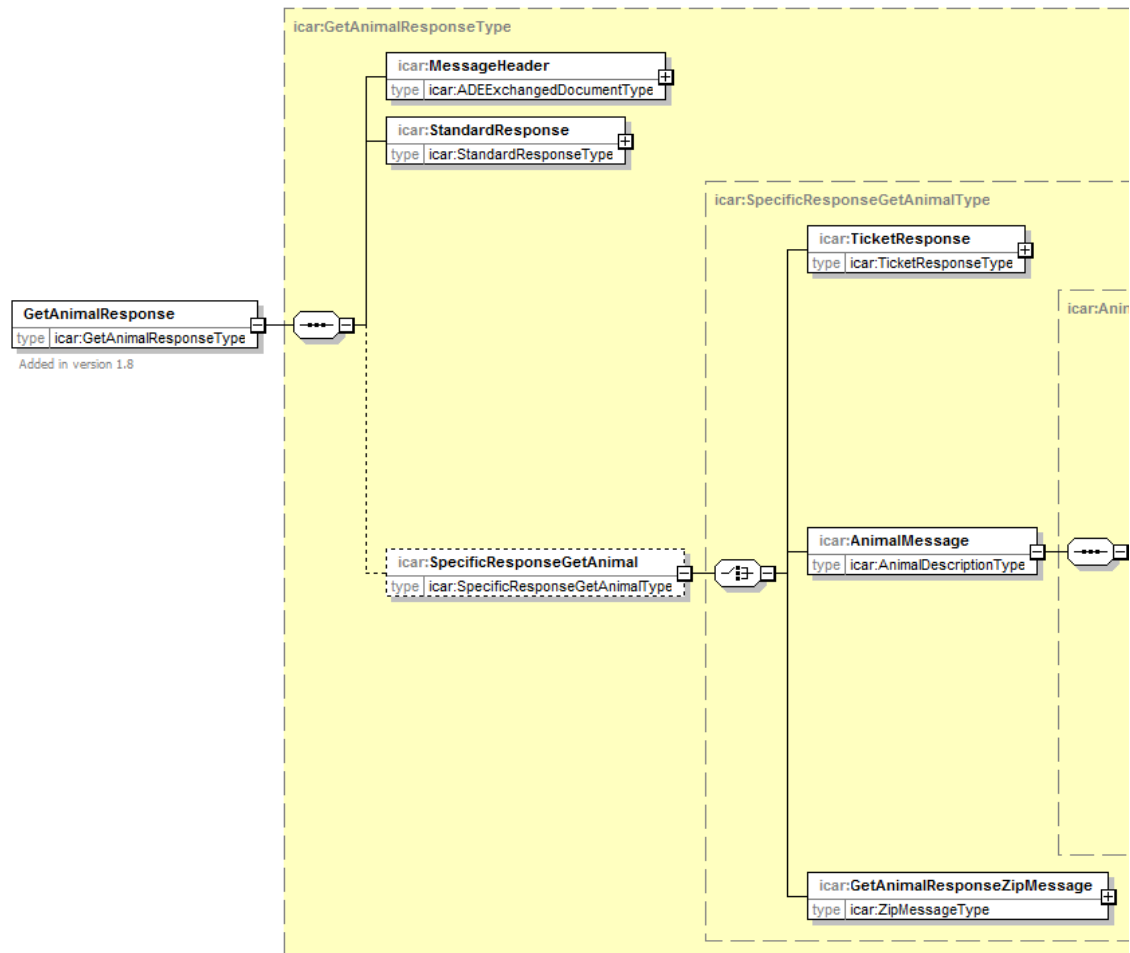
AnimalIdentifier: the identifier of the animal for which the data is requested

6.1.3 Response description

Message **GetAnimalResponse** (see diagram below) is in compliance with the general specifications for results (see General specifications / Response specifications).

It conveys the animal core data for an animal specified by the request parameter **AnimalIdentifier**.

Figure 14 GetAnimalResponse



AnimalMessage: conveys the animal descriptions for the requested animal using type **AnimalDescription** (for details see entity description in section “**Common Components.AnimalDescriptionType**” above).

6.2 Service Description: UpdateAnimal

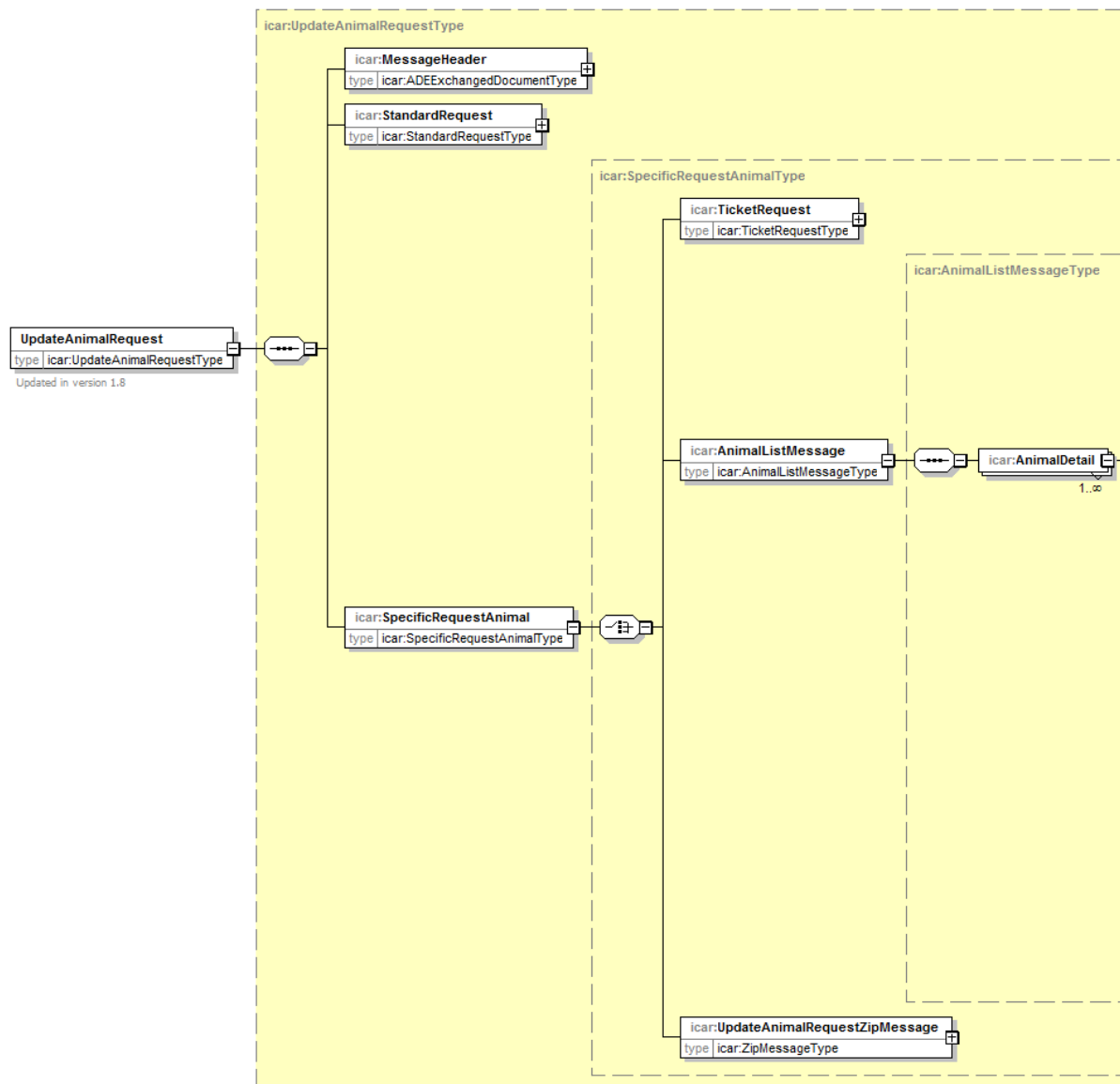
6.2.1 Purpose

The service **UpdateAnimal** allows the equipment to send information describing a list of animals to the farm management system.

6.2.2 Request description

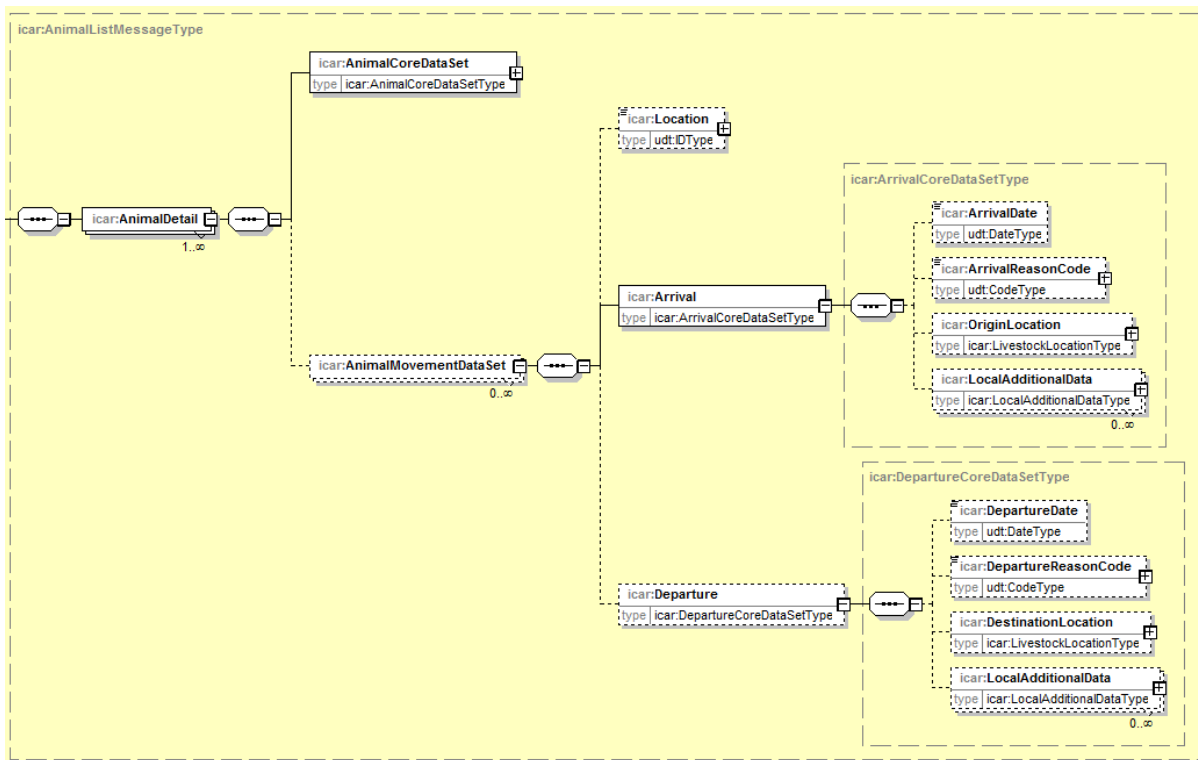
Message **UpdateAnimalRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

Figure 15 UpdateAnimalResponse



AnimalListMessage: A list of **AnimalData** components of type **AnimalListMessageType** conveying the details of animals currently managed by the equipment

Figure 16 *AnimalListMessageType*



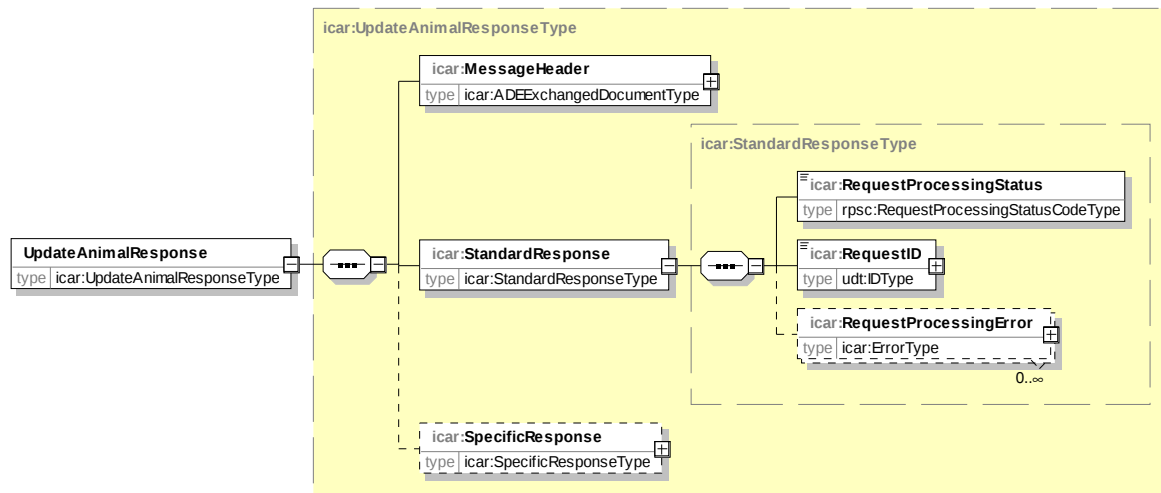
- a. **AnimalCoreDataSet:** Core description of an animal (for details see entity description in section “Common Components.AnimalCoreDataSetType” above)
- b. **AnimalMovementDataSet:** A list of animal movement information known to the equipment
- c. **Location:** The location of the animal
- d. **ArrivalCoreDataSet:** The details of the animal’s arrival in this location. (see entity description in section “Common Components.ArrivalCoreDataSetType” above)
- e. **DepartureCoreDataSet:** The details of the animal’s departure from this location. (see entity description in section “Common Components.DepartureCoreDataSetType” above)

6.2.3 Response description

Message **UpdateAnimalResponse** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Response specifications)

It is used to receive the status of animal data processing by the service provider.

Figure 17 UpdateAnimalResponse



7 Exchange of Animal Arrival in Herd

7.1 Service Description: GetArrivals

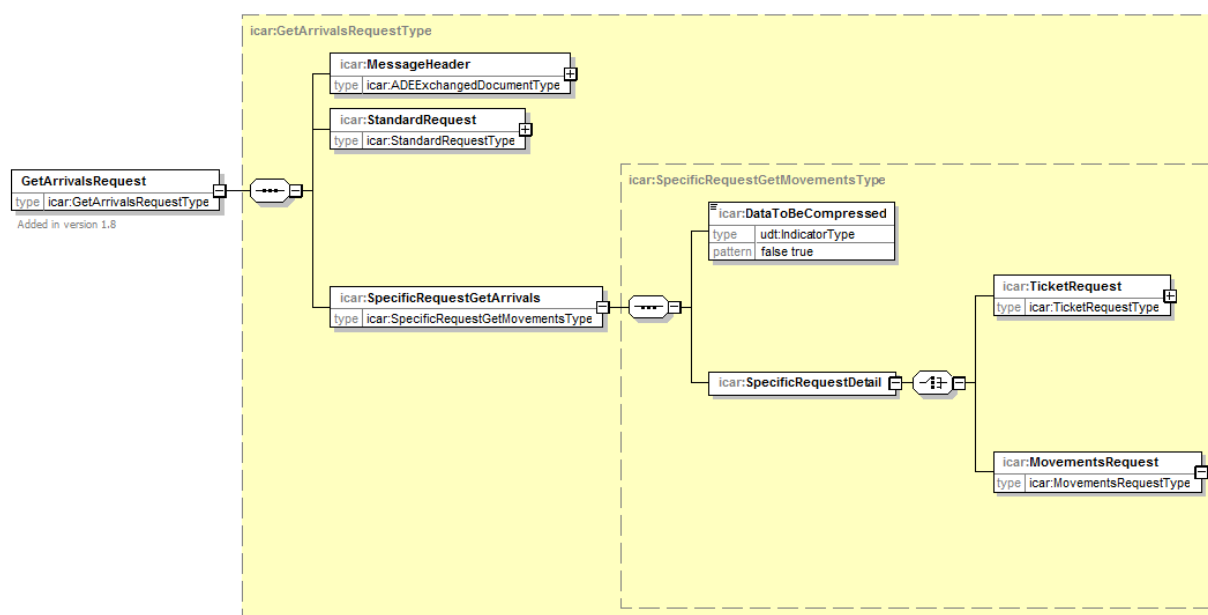
7.1.1 Purpose

The service **GetArrivals** allows the owner of the equipment to retrieve the arrivals of animals for a given location, animal and/or period.

7.1.2 Request description

Message **GetArrivalsRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications). It is used to request the list of animal arrivals stored in the farm management system.

Figure 18 GetArrivalRequest



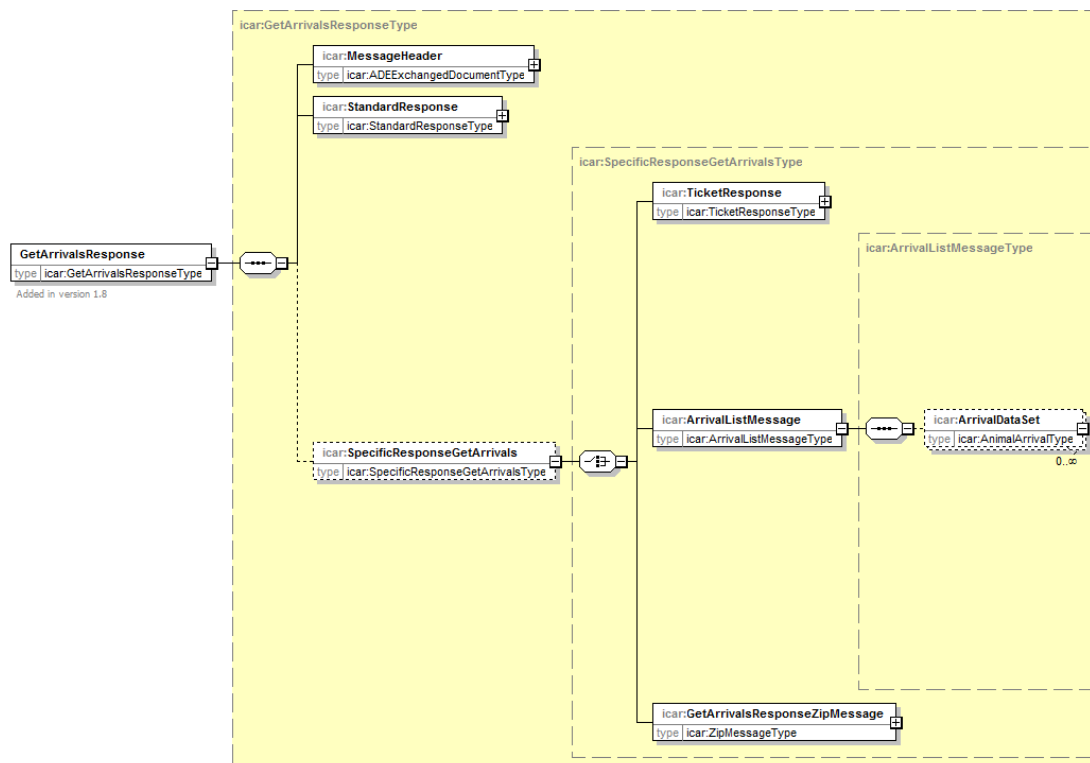
The size of the result can be controlled by providing optional parameters in the entity MovementsRequest (see description in section “Common Components” above).

7.1.3 Response description

Message **GetArrivalsResponse** (see diagram below) is in compliance with the general specifications for results (see General specifications / Response specifications).

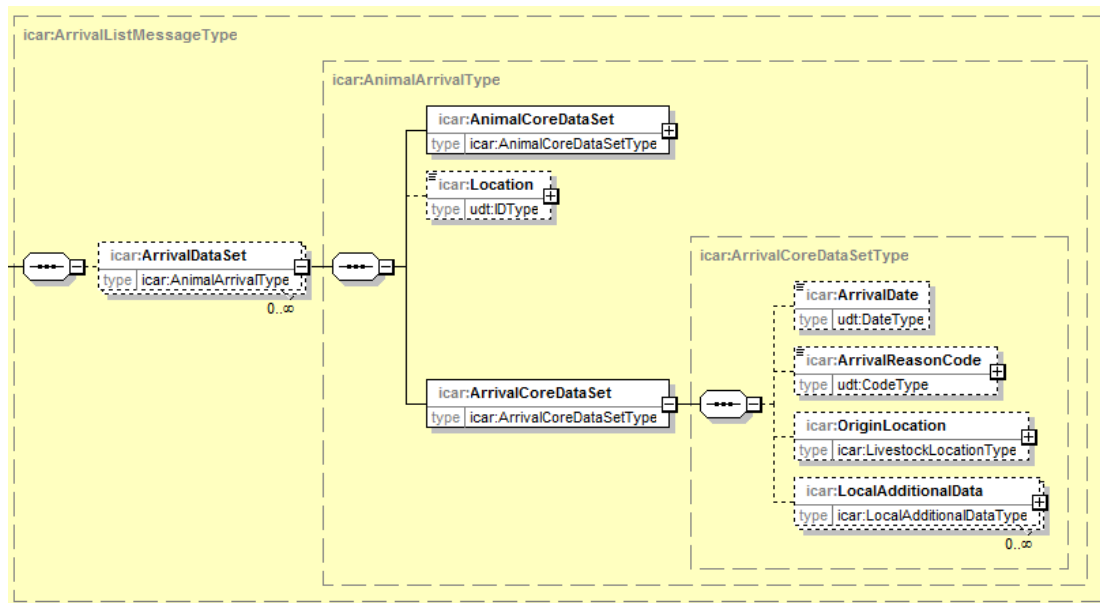
It provides a list of arrival data sets registered in the farm management system limited by the request parameters.

Figure 19 GetArrivalResponse



- a. **ArrivalListMessage**: List of animal arrival events conveyed as **ArrivalDataSetType**

Figure 20 ArrivalDataSet



- a. **AnimalCoreDataSet:** Core description of an animal (for details see entity description in section “Common Components.AnimalCoreDataSetType” above)
- b. **Location:** The location of arrival
- c. **ArrivalCoreDataSet:** It conveys the details of an animal arrival. (see entity description in section “Common Components.ArrivalCoreDataSetType” above)

7.2 Service Description: UpdateArrivals

7.2.1 Purpose

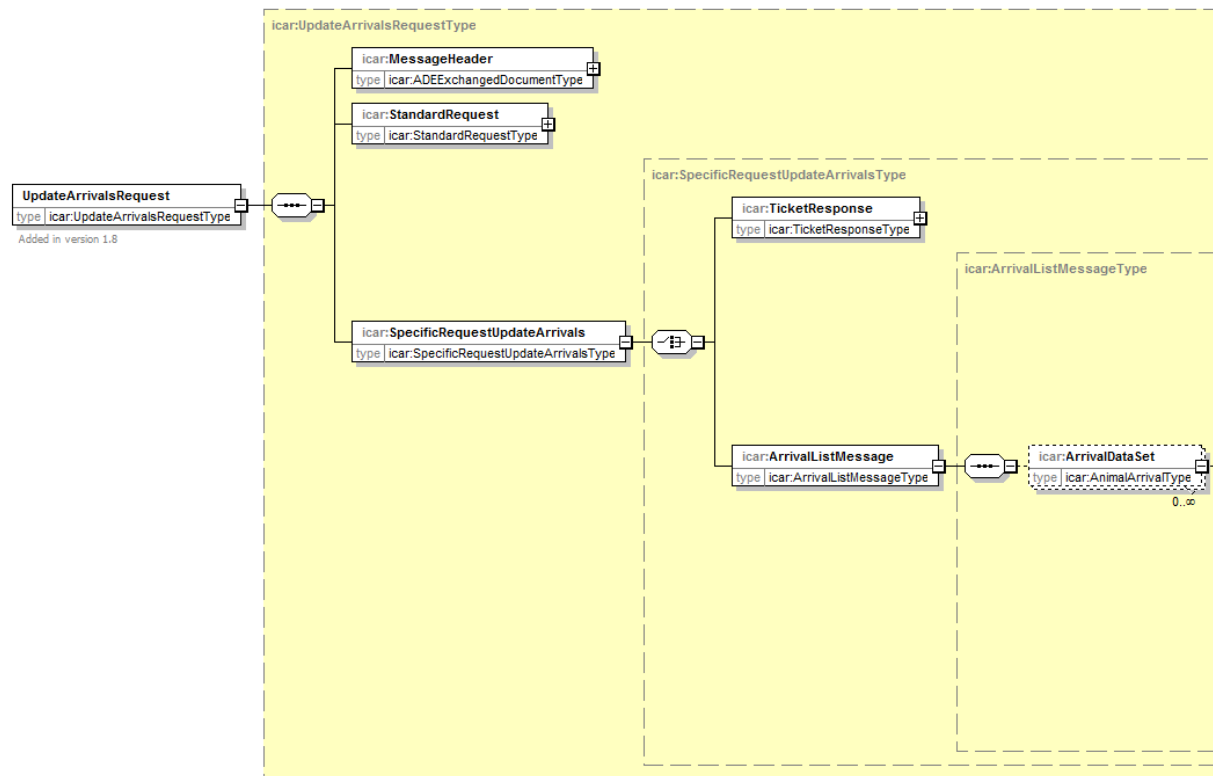
The service **UpdateArrivals** is designed for registering new or updated animal arrival events in the farm management system.

7.2.2 Request description

Message **UpdateArrivalsRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

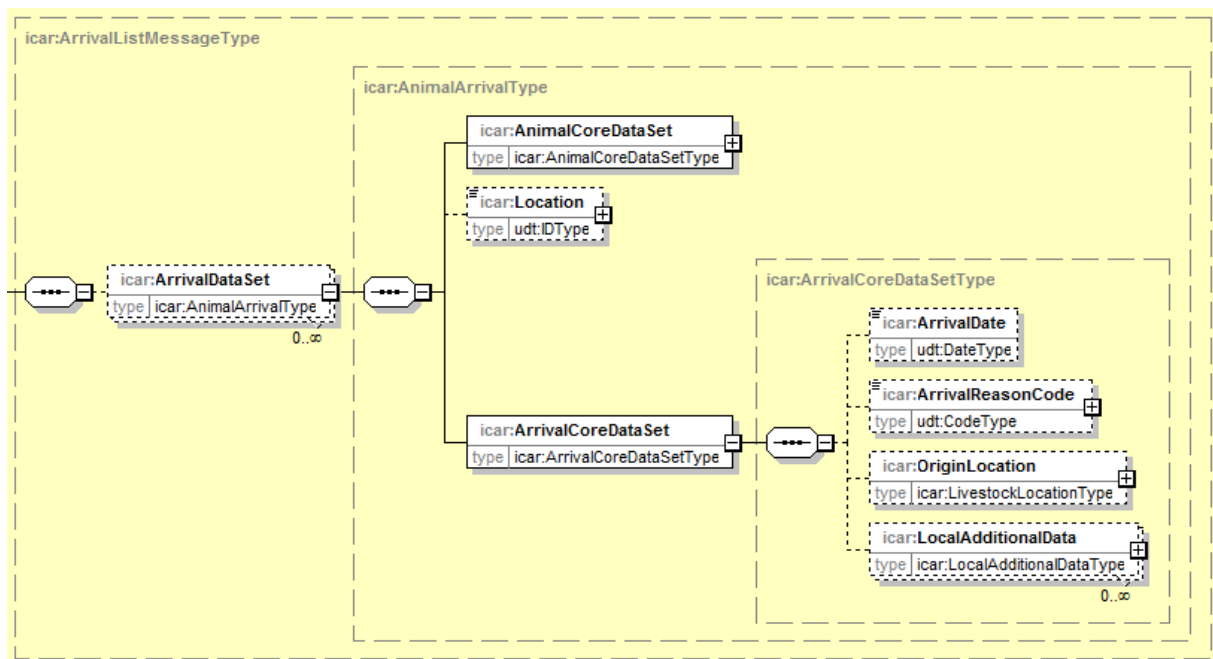
A list of arrival events is transferred to the service provider.

Figure 21 UpdateArrivalRequest



- a. **ArrivalListMessage**: List of animal arrival events conveyed as **ArrivalDataSetType**

Figure 22 ArrivalDataSet



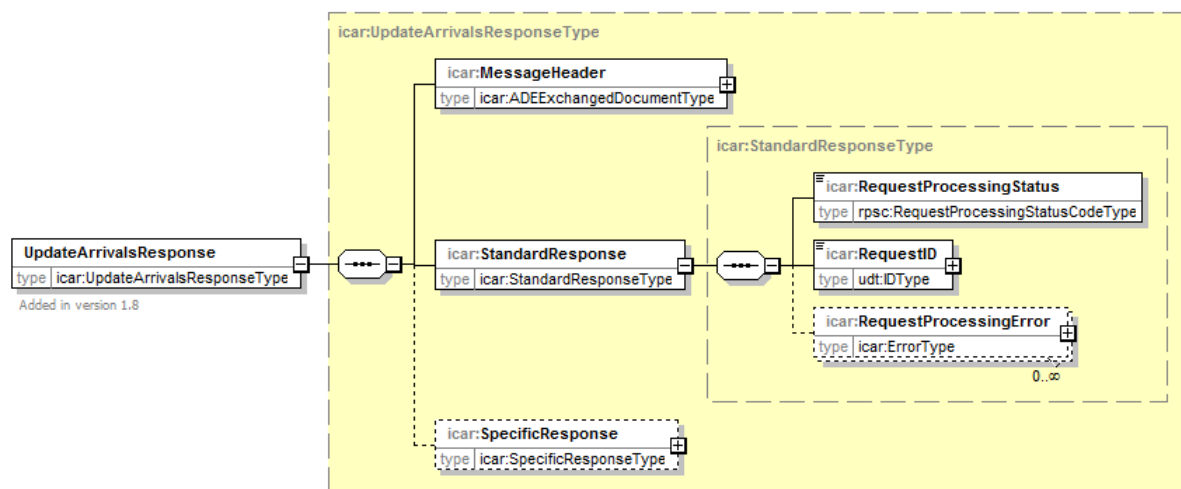
- a. **AnimalCoreDataSet:** Core description of an animal (for details see entity description in section “Common Components.AnimalCoreDataSetType” above)
- b. **Location:** The location of arrival
- c. **ArrivalCoreDataSet:** It conveys the details of an animal arrival. (see entity description in section “Common Components.ArrivalCoreDataSetType” above)

7.2.3 Response description

Message UpdateArrivalsResponse (see diagram below) is in compliance with the general specifications for requests (see General specifications / Response specifications)

It is used to receive the status of arrival events processing by the service provider.

Figure 23 UpdateArrivalResponse



8 Exchange of Animal Departure from Herd

8.1 Service Description: GetDepartures

8.1.1 Purpose

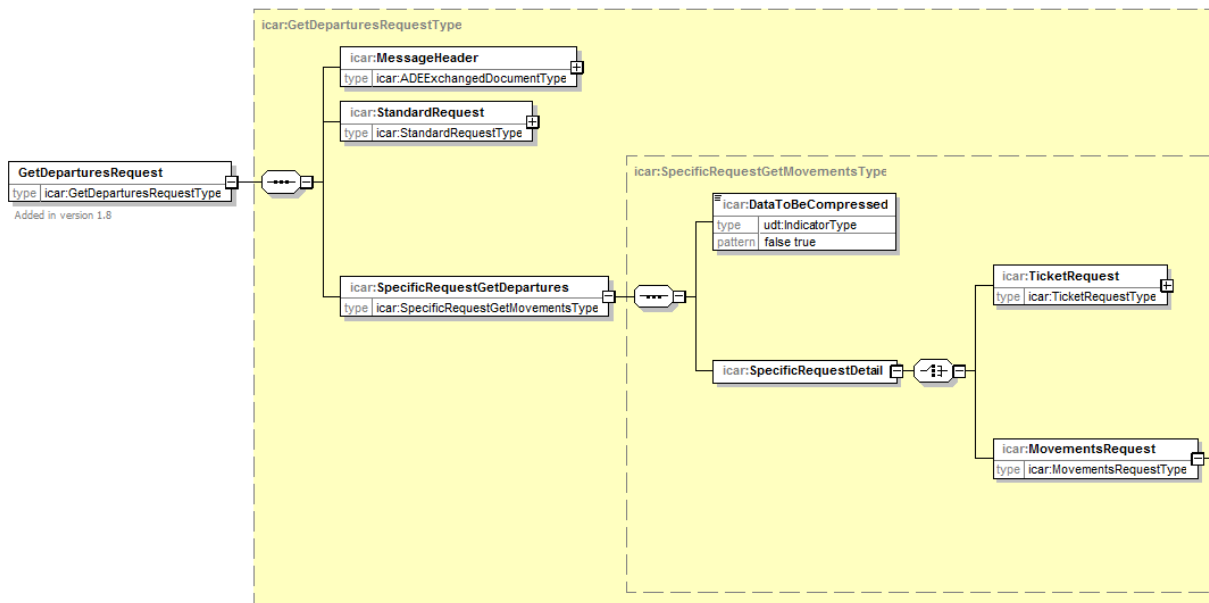
The service GetDepartures allows the owner of the equipment to retrieve the departures of animals for a given location, animal and/or period.

8.1.2 Request description

Message GetDepartureRequest (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

It is used to request the list of animal departures stored in the farm management system.

Figure 24 GetDepartureRequest



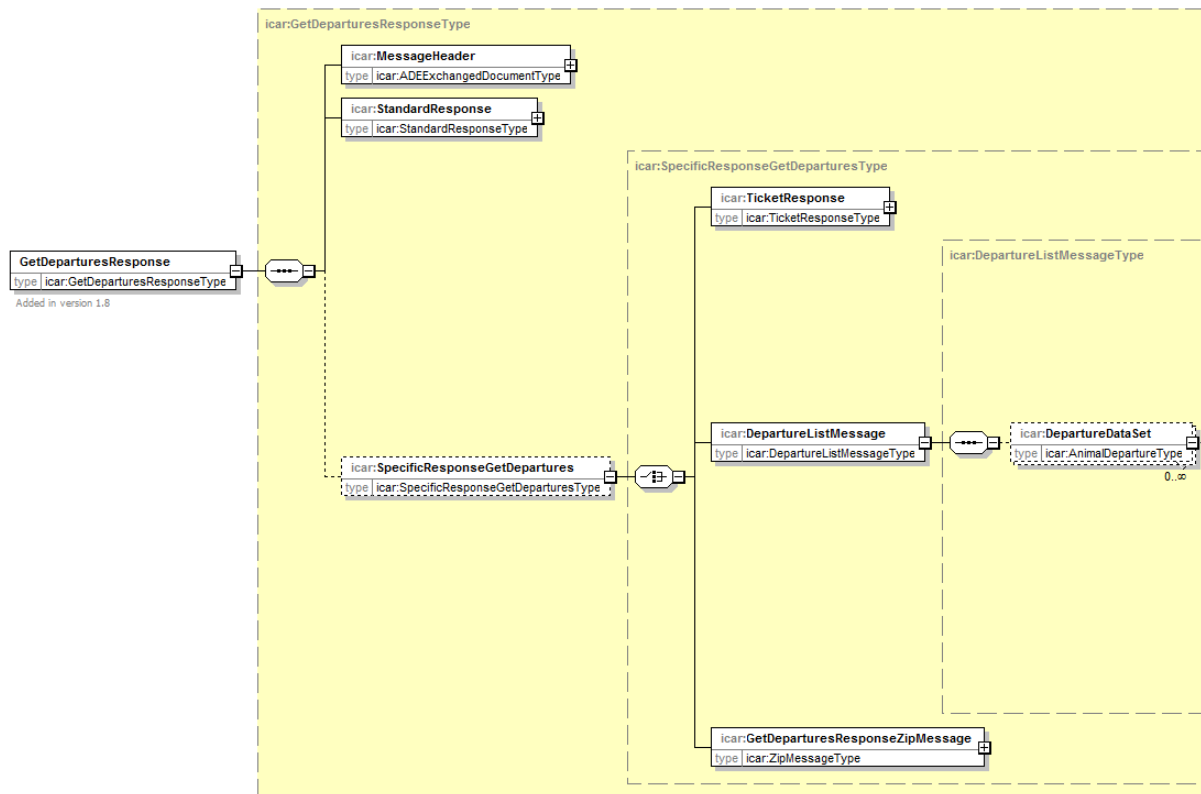
The size of the result can be controlled by providing optional parameters in the entity **MovementsRequest** (see description in section “Common Components” above).

8.1.3 Response description

Message **GetDepartureResponse** (see diagram below) is in compliance with the general specifications for results (see General specifications / Response specifications).

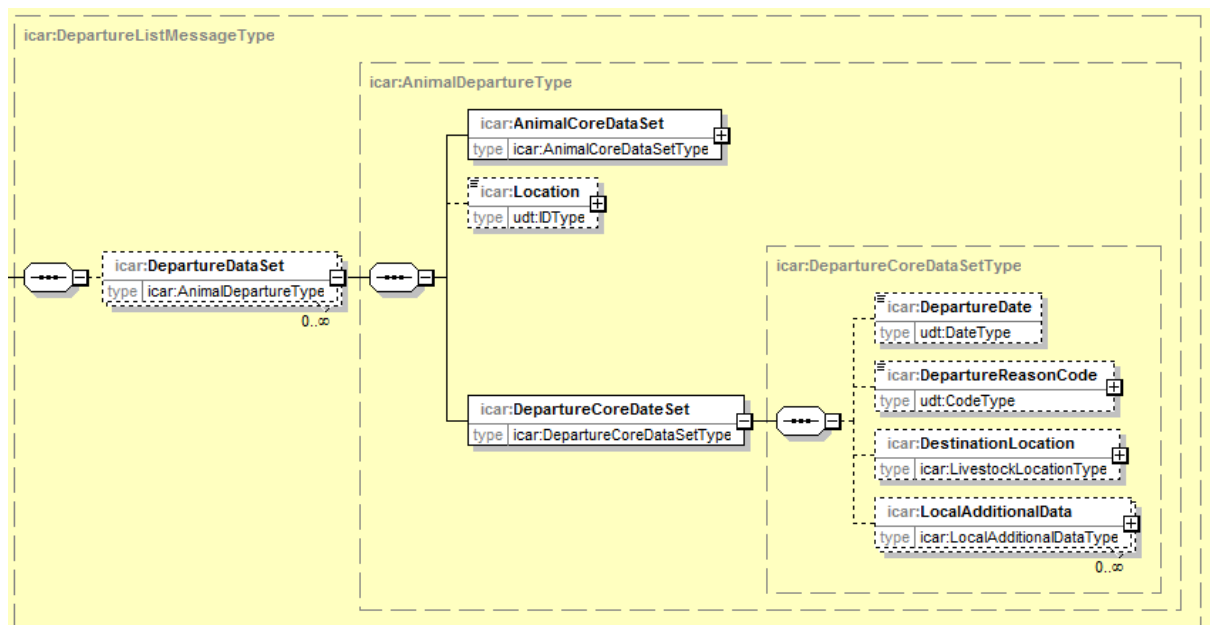
It provides a list of arrival data sets registered in the farm management system limited by the request parameters.

Figure 25 GetDepartureResponse



- a. **DepartureListMessage:** List of animal departure events conveyed as **DepartureDataSetType**

Figure 26 DepartureDataSet



- a. **AnimalCoreDataSet:** Core description of an animal (for details see entity description in section “Common Components.AnimalCoreDataSetType” above)
- b. **Location:** The location of arrival

DepartureCoreDataSet: It conveys the details of an animal arrival. (see entity description in section “Common Components.DepartureCoreDataSetType” above).

8.2 Service Description: UpdateDepartures

8.2.1 Purpose

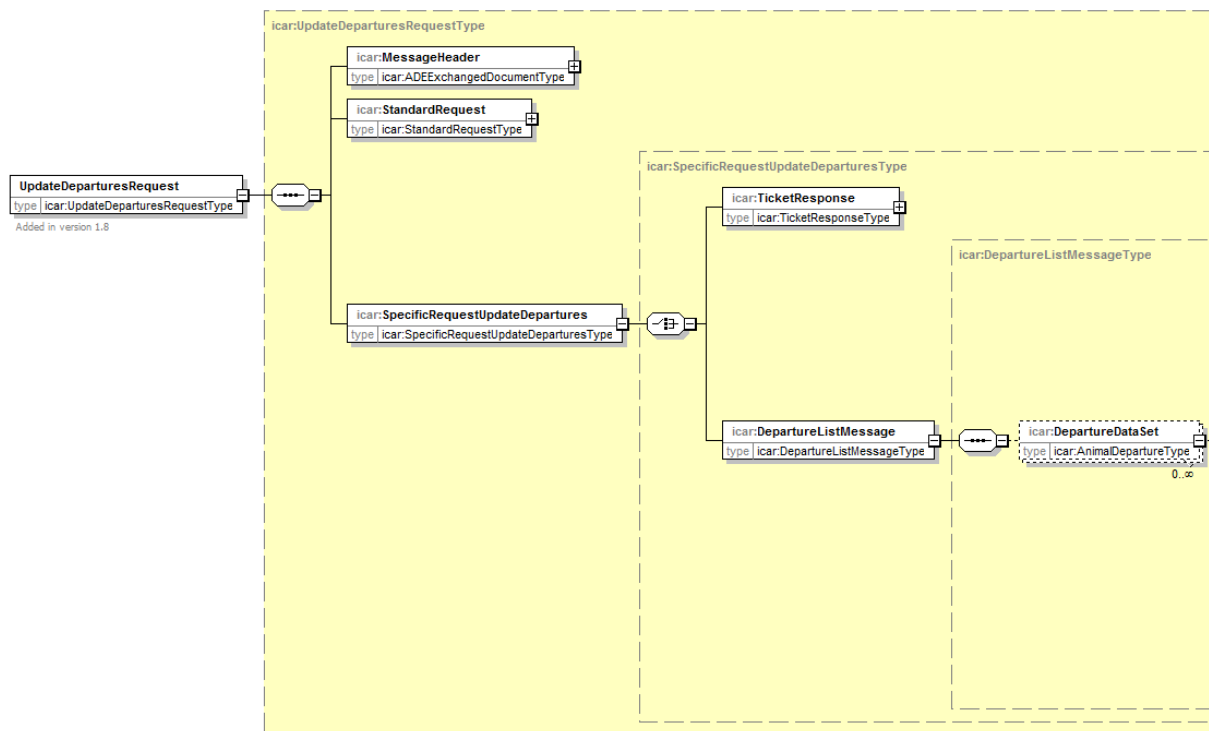
The service **UpdateDepartures** is designed for registering new or updated animal departure events in the farm management system.

8.2.2 Request description

Message **UpdateDeparturesRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

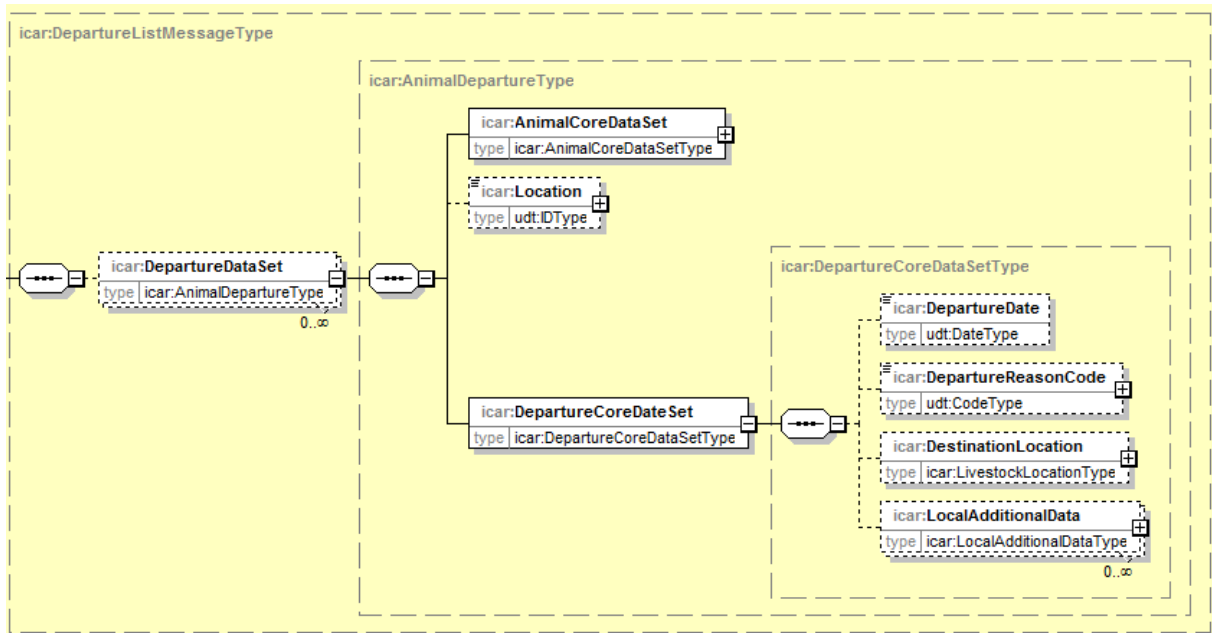
A list of departure events is transferred to the service provider.

Figure 27 UpdateDepartureRequest



- a. **DepartureListMessage:** List of animal arrival events conveyed as **DepartureDataSetType**

Figure 28 *DepartureDataSet*



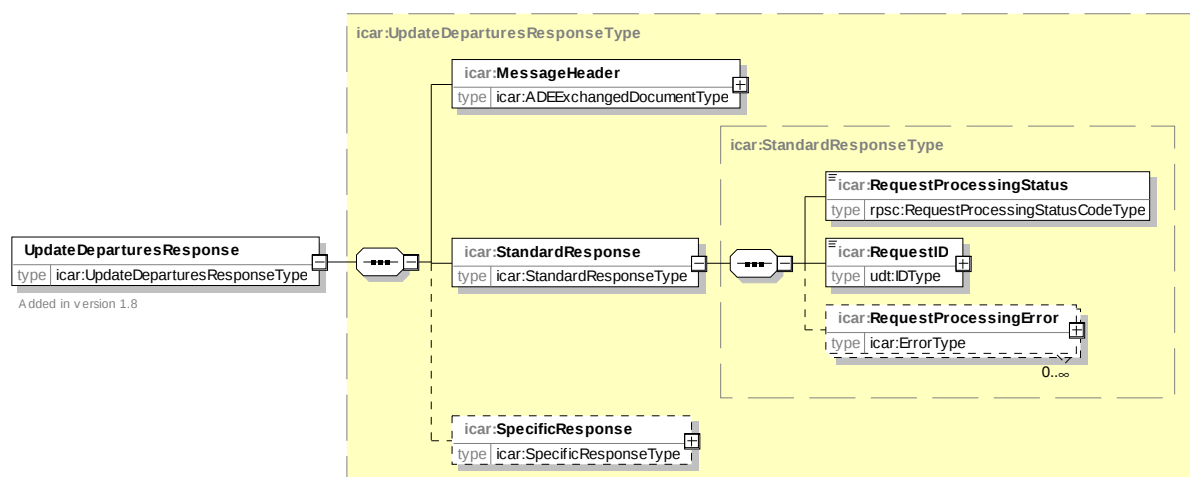
- AnimalCoreDataSet:** Core description of an animal (for details see entity description in section “Common Components.AnimalCoreDataSetType” above)
- Location:** The location of arrival
- DepartureCoreDataSet:** It conveys the details of an animal arrival. (see entity description in section “Common Components.DepartureCoreDataSetType” above)

8.2.3 Response description

Message *UpdateDeparturesResponse* (see diagram below) is in compliance with the general specifications for requests (see General specifications / Response specifications)

It is used to receive the status of departure events processing by the service provider.

Figure 29 *UpdateDepartureResponse*





THE GLOBAL STANDARD
FOR LIVESTOCK DATA

Procedure 4 of Section 15 of ICAR Guidelines – Reproduction Data Services

Procedure 4 – Reproduction Data

Version July, 2018

File Ref: 15 Procedure 4 Reproduction Data v18.01.docx

Network. Guidelines. Certification.

Table of Contents

1	Introduction	4
2	Definitions and Terminology	4
3	Scope	4
4	Exchange of Reproduction Events	5
4.1	Common Components	5
4.1.1	ReproductionEventMessage.....	5
4.2	Specific Event Component Description.....	6
4.2.1	Heat	6
4.2.2	ArtificialInsemination.....	6
4.2.3	NaturalService	7
4.2.4	PregnancyCheck	8
4.2.5	Specific component description: Abortion.....	8
4.2.6	Specific component description: Parturition	8
4.2.7	DryOff	9
4.3	Service Description: GetReproductionEvents.....	10
4.3.1	Purpose.....	10
4.3.2	Request description	10
4.3.3	Response description	11
4.4	Service Description: UpdateReproductionEvents	11
4.4.1	Purpose.....	11
4.4.2	Request description	12
4.4.3	Response description	12

Tables

Table 1. Definitions of Terms used in these guidelines.....	4
---	---

Equations

No table of figures entries found.

Figures

Figure 1. Scope of Guideline.	4
Figure 2 ReproductionEventMessage	5
Figure 3 Heat component	6
Figure 4 Artificial insemination component.....	7
Figure 5 Natural service component	8
Figure 6 Pregnancy check component	8
Figure 7 Parturition component	9
Figure 8 Overall description of GetReproductionEventRequest	10

Figure 9 Detailed description of ReproductionEventsRequest.....10
Figure 10 Overall description of GetReproductionEventResponse 11
Figure 11 Overall description of UpdateReproductionEventRequest 12

Summary of Changes

Date of Change	Nature of Change
July 2018	New procedure created from methodology chapter in Section 15 Overview. Replaced by new version from ADE-WG. Template applied.

1 Introduction

Monitoring and controlling reproduction is a key issue for livestock farming. It is based on the registration and the analysis of the following events:

- a. Heat
- b. Artificial insemination
- c. Natural service
- d. Pregnancy check
- e. Abortion
- f. Parturition
- g. Dry off

Farmers need:

- h. to retrieve any types of events to allow the AMS to control precisely milking
- i. to retrieve heats and parturitions which may be captured by reproduction monitoring devices
- j. to send part or all the reproduction events they have captured manually on the farm
- k. Precise reproduction monitoring requires to transmit the hour in addition to the date of the event.

The service descriptions follow the principals described in “Guideline A Methodology” and reuse the basic elements defined there.

The complete description of the codelists referenced here can be found in the Annex document.

2 Definitions and Terminology

Table 1 contains a list of important definitions for terms and abbreviations used in these guidelines.

Table 1. Definitions of Terms used in these guidelines.

Term	Definition
CBD	Central Database

3 Scope

Figure 1 gives a pictorial summary of the main elements of this guideline. The numbers in this figure refer to the section numbers of this guideline.

Figure 1. Scope of Guideline.

4. Exchange of Reproduction Events

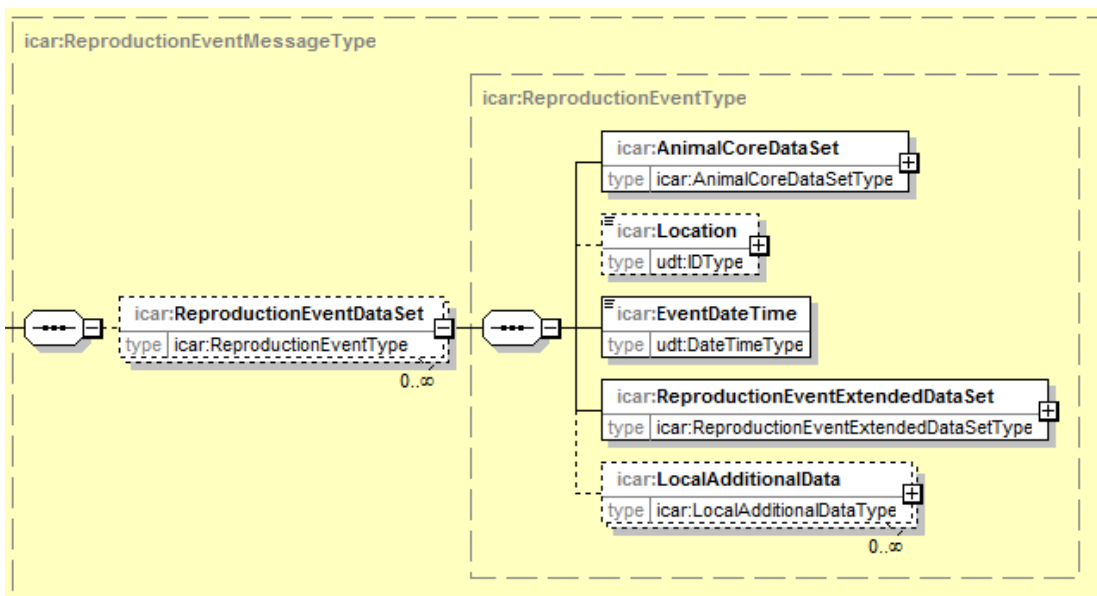
4 Exchange of Reproduction Events

4.1 Common Components

4.1.1 ReproductionEventMessage

A **ReproductionEventMessage** (see diagram below) may be transmitted by the on-farm device to be recorded by the CBD or retrieved from CBD.

Figure 2 ReproductionEventMessage



The component includes:

- a. **AnimalCoreDataSet**: mandatory; (see General specifications / Common component)
- b. **Location**: mandatory; (see General specifications / Common component)
- c. **EventDateTime**: mandatory; date time when the event occurred.
- d. **ReproductionEventExtendedDataSet**: mandatory, it must be one the following components which are described below:
 - e. Heat
 - f. ArtificialInsemination
 - g. PregnancyCheck
 - h. Abortion
 - i. Parturition
 - j. DryOff
- k. **LocalAdditionnalData**: (see General specifications / Common component)

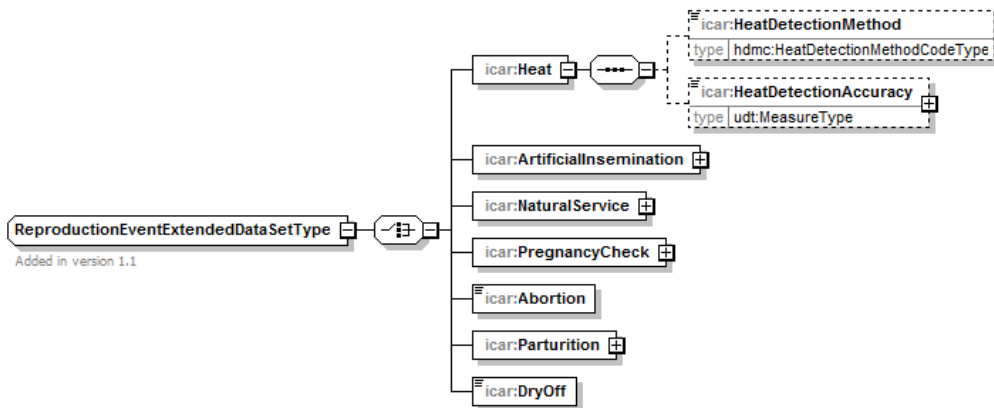
4.2 Specific Event Component Description

4.2.1 Heat

The component includes:

- a. **Method**: mandatory; Chemical, Visual, Spectral Pedometers and Other; see ADE code list.
- b. **Accuracy**: optional.

Figure 3 Heat component

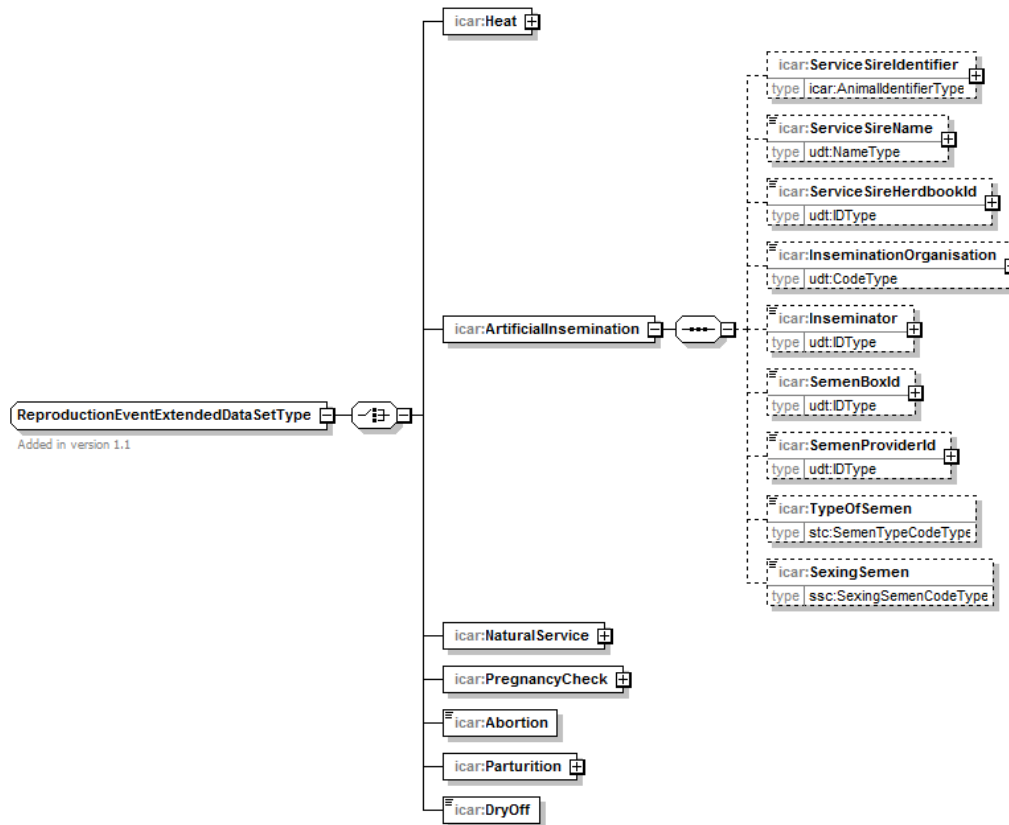


4.2.2 ArtificialInsemination

The component includes:

- a. **ServiceSireIdentifier**: mandatory; official identifier of the sire used to produce the semen.
- b. **SireName**: optional
- c. **SireHerdBookID**: optional; only if different from ServiceSireIdentifier
- d. **InseminationOrganisation**: optional; only when the artificial insemination is performed by a service provider; locally defined.
- e. **Inseminator**: optional; only when the artificial insemination is performed by a service provider; locally defined.
- f. **SemenBox**: optional; the number of the box where the straws of semen are stored; locally defined.
- g. **SemenProvider**: optional; the identifier of body which produced the straws: locally defined.
- h. **TypeOfSemen**: optional; Fresh or Frozen; see ADE code list.
- i. **SexingSemen**: optional; sexing male, sexing female; see ADE code list.

Figure 4 Artificial insemination component

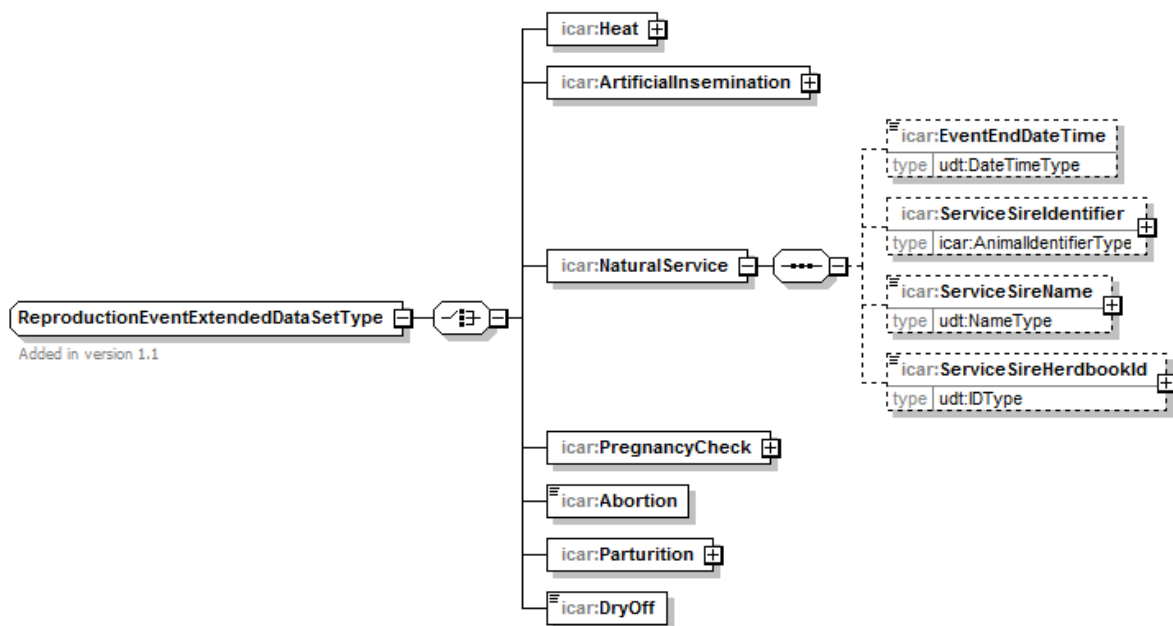


4.2.3 NaturalService

The component includes:

- a. **EventEndDateTime**: optional; only in case of mating period.
- b. **ServiceSireIdentifier**: mandatory; official identifier of the sire used to produce the semen.
- c. **SireName**: optional
- d. **SireHerdBookID**: optional; only if different from ServiceSireIdentifier

Figure 5 Natural service component

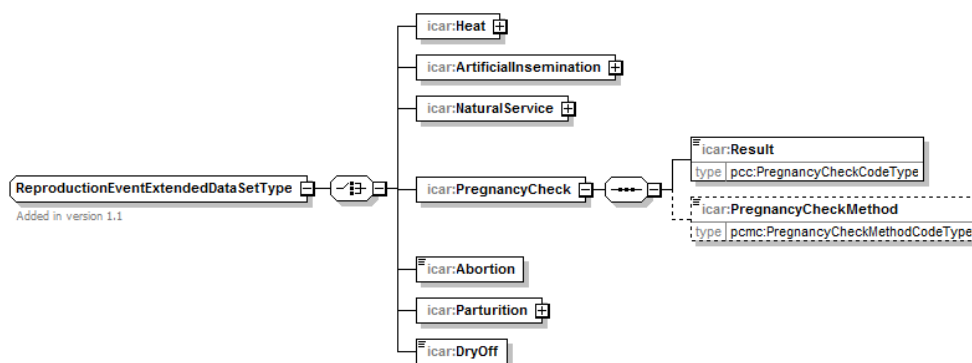


4.2.4 PregnancyCheck

It includes.

- a. **Result**: mandatory, see ADE code list.
- b. **PregnancyCheckMethod**: mandatory; Palpation, Echography, Other; see ADE code list

Figure 6 Pregnancy check component



4.2.5 Specific component description: Abortion

To be used in case of loss of embryos. No specific information.

4.2.6 Specific component description: Parturition

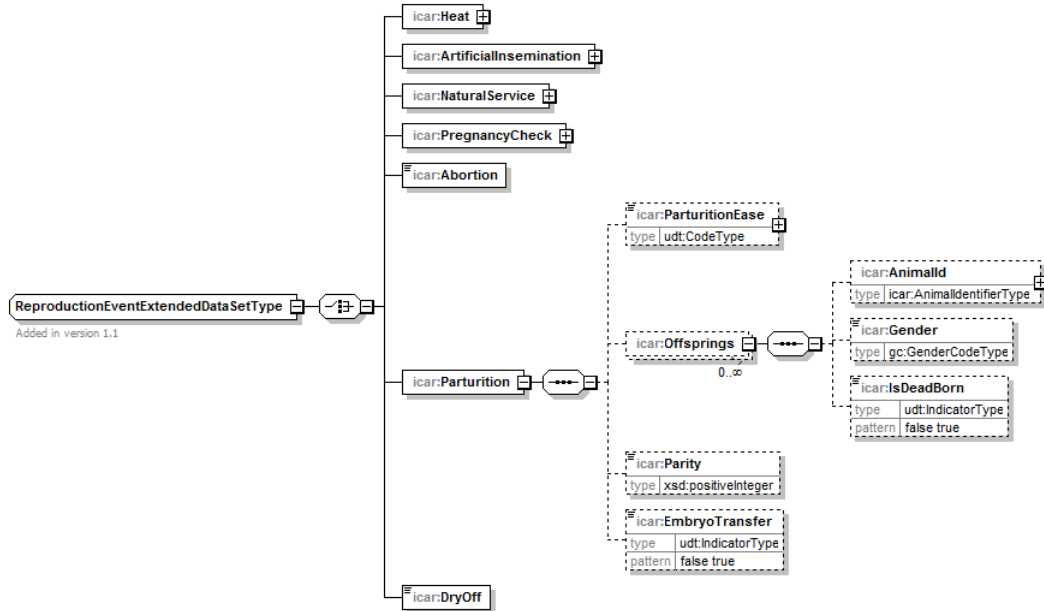
To be used when the result of the gestation is not embryos whatever live or dead.

The parturition includes:

- a. **Parturition ease**: optional; a local code list should be provided.
- b. **Offspring**: optional; the list of animals further to that parturition.

- c. **Parity**: optional; number of parturitions for that female since its birth.
- d. **EmbryoTransfer**: yes or no.

Figure 7 Parturition component



4.2.7 DryOff

No specific information.

4.3 Service Description: GetReproductionEvents

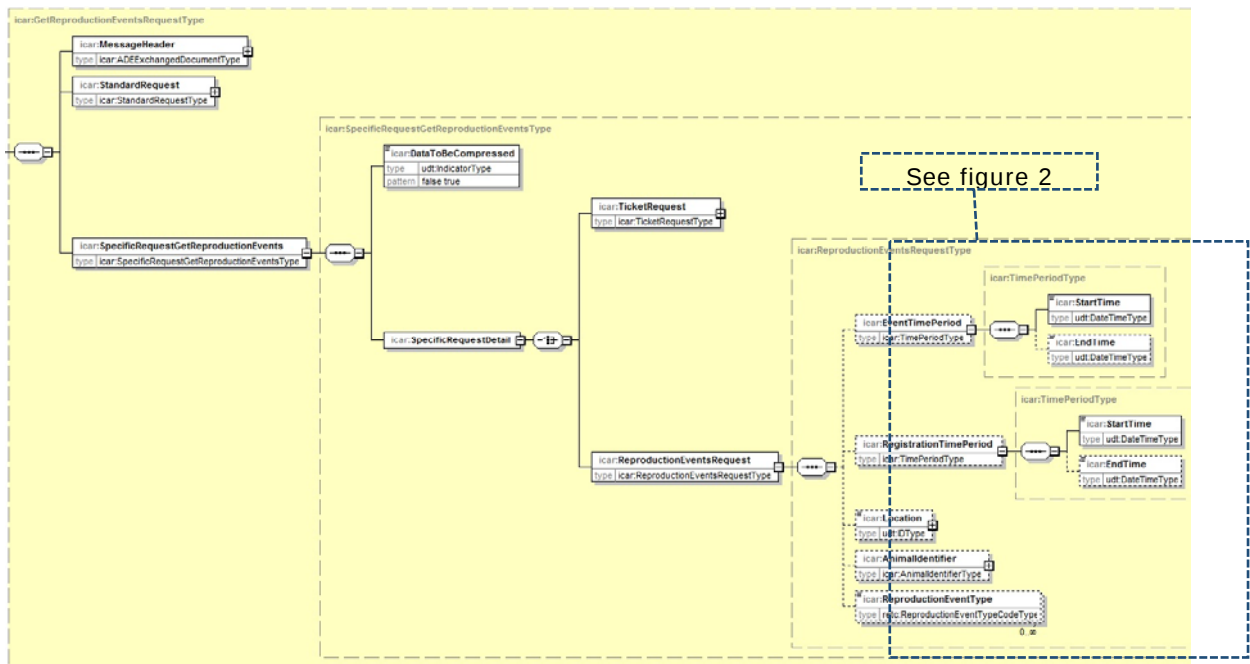
4.3.1 Purpose

To retrieve reproduction events.

4.3.2 Request description

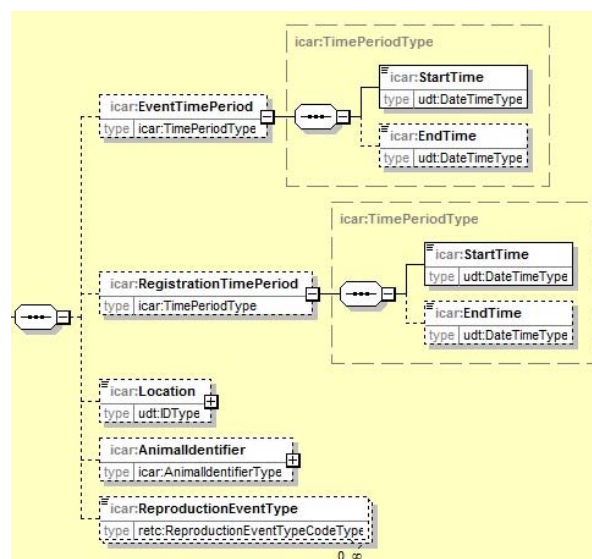
GetReproductionEventRequest (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications)

Figure 8 Overall description of GetReproductionEventRequest



The component ReproductionEventsRequest (see diagram below) contains the parameters of the request.

Figure 9 Detailed description of ReproductionEventsRequest



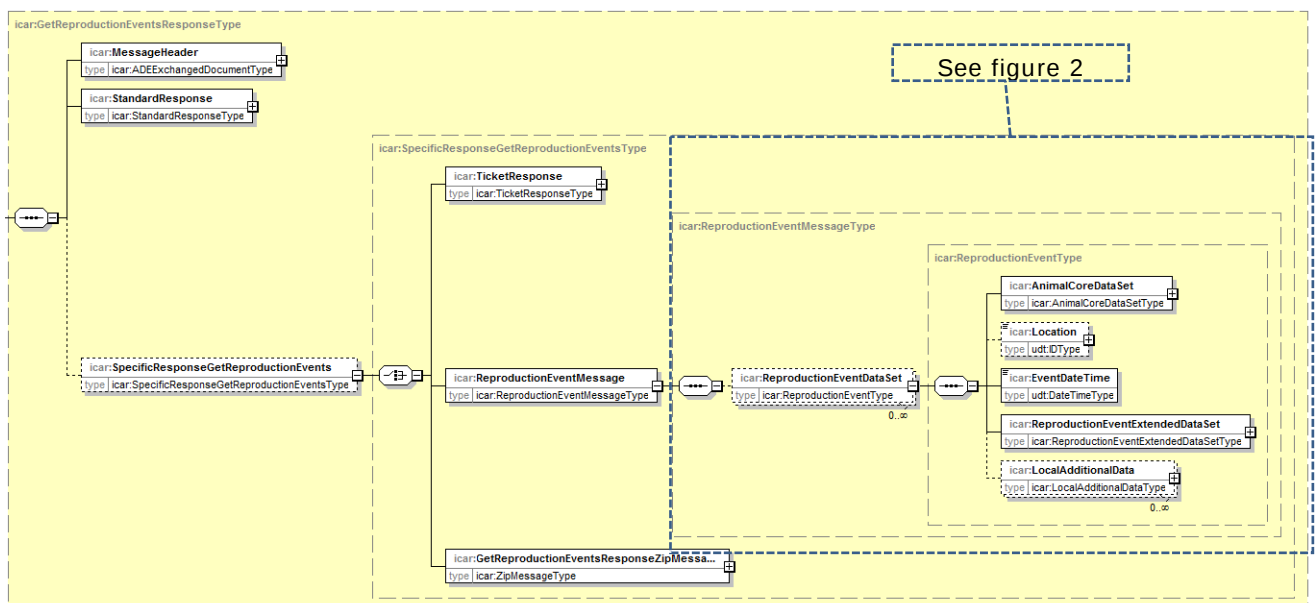
The parameters include:

- e. **A time period:** optional; if specified, only events which occurred during the specified time period are retrieved; otherwise all the events are retrieved whatever the date. A time period may be :
 - **EventTimePeriod:** the dates which are considered to select the events are the dates when the events occurred. Although event dates are recorded by all the servers, it is not the most appropriate method to retrieve data to update the equipment.
 - **RegistrationTimePeriod:** the dates which are considered to select the events are the dates when the events was registered by the server. Although registration dates are not recorded by all the servers, it is the best practice to retrieve data to update the equipment.
- f. **Location:** mandatory; to retrieve the events of a specified location.
- g. **AnimalIdentifier:** optional; if specified, only events of the specified animal are retrieved.
- h. **ReproductionEventType:** optional; a list of event types; if specified, only events whose types have been specified are retrieved.

4.3.3 Response description

GetReproductionEventResponse (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications)

Figure 10 Overall description of GetReproductionEventResponse



The response contains a list of **ReproductionEventMessage**.

There is one **ReproductionEventMessage** per event
(see **Specific Event Component Description**)

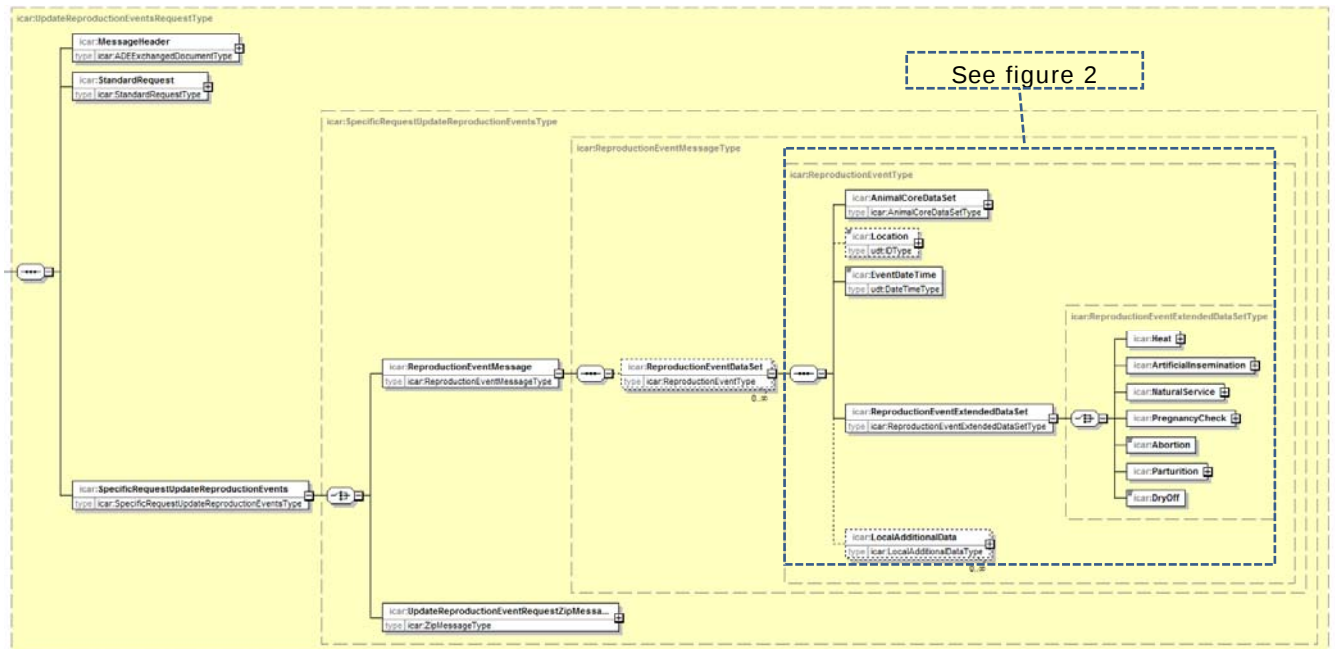
4.4 Service Description: UpdateReproductionEvents

4.4.1 Purpose

To have reproduction events registered by the CDB.

UpdateReproductionEventsRequest (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications)

Figure 11 Overall description of UpdateReproductionEventRequest



The request contains a set of `ReproductionEventMessage` (see `Specific` component description) to be registered.

4.4.3 Response description

UpdateReproductionEventsResponse is in compliance with the general specifications for responses (see General specifications / Request specifications)



THE GLOBAL STANDARD
FOR LIVESTOCK DATA

Procedure 5 of Section 15 of ICAR Guidelines – Technical Services

Procedure 5 – Technical

Version July, 2018

File Ref: 15 Procedure 5 Technical Data v18.01.docx

Network. Guidelines. Certification.

Table of Contents

1	Introduction	4
2	Definitions and Terminology	4
3	Scope	4
4	Service GetLocalCodeList	5
4.1	Purpose	5
4.2	Request description.....	5
4.3	Response description	6
5	Service UpdateDevice	8
5.1	Purpose	8
5.2	Request description.....	8
5.3	Response description	9
6	Service UpdateLivestockLocation	10
6.1	Purpose	10
6.2	Request description.....	10
6.3	Response description	12

Tables

Table 1. Definitions of Terms used in these guidelines.....	4
---	---

Equations

No table of figures entries found.

Figures

Figure 1. Scope of Guideline.	4
Figure 2 GetLocalCodeListRequest	5
Figure 3 SpecificRequestGetLocalCodeList	5
Figure 4 GetLocalCodeListResponse.....	6
Figure 5 LocalCodeList	6
Figure 6 UpdateDeviceRequest	8
Figure 7 DeviceListMessage.....	9
Figure 8 UpdateDeviceResponse.....	10
Figure 9 UpdateLivestockLocationRequest	11
Figure 10 LivestockLocationListMessage	11
Figure 11 UpdateLivestockLocationResponse.....	12

Summary of Changes

Date of Change	Nature of Change
July 2018	New procedure created from methodology chapter in Section 15 Overview. Replaced by new version from ADE-WG. Template applied.

1 Introduction

Technical services are used to support the correct functionality of data exchange software.

In order to minimize the data transfer size often just a reference is transferred in the messages described above e.g. the location identifier or the device identifier. The describing details of the referenced object have to be transferred in the context of creation or changes by using specific technical services.

Furthermore the equipment must be enabled to retrieve the current content of locally used code lists in order to be able to react to changes.

A number of services has been designed to meet these requirements:

- a. **UpdateDevice:** transfer details of a device
- b. **UpdateLocation:** transfer details of a location or sub location
- c. **GetLocalCodeList:** retrieve local code lists from local service providers

2 Definitions and Terminology

Table 1 contains a list of important definitions for terms and abbreviations used in these guidelines.

Table 1. Definitions of Terms used in these guidelines.

Term	Definition

3 Scope

Figure 1 gives a pictorial summary of the main elements of this guideline. The numbers in this figure refer to the section numbers of this guideline.

Figure 1. Scope of Guideline.



4 Service GetLocalCodeList

4.1 Purpose

The service **GetLocalCodeList** is designed to retrieve local code lists from local service providers. It is the responsibility of

- d. the service provider to provide the correct set of codes as well as a description
- e. the equipment manufacturer to retrieve the local code lists and maintain a correct mapping to internal code lists
- f. the equipment manufacturer to make sure that local code lists are kept up-to-date on the operational equipment

4.2 Request description

Message **GetLocalCodeListRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

It is used to request the list of local codes for all code list or for a specified code list name from the local service provider.

Figure 2 GetLocalCodeListRequest

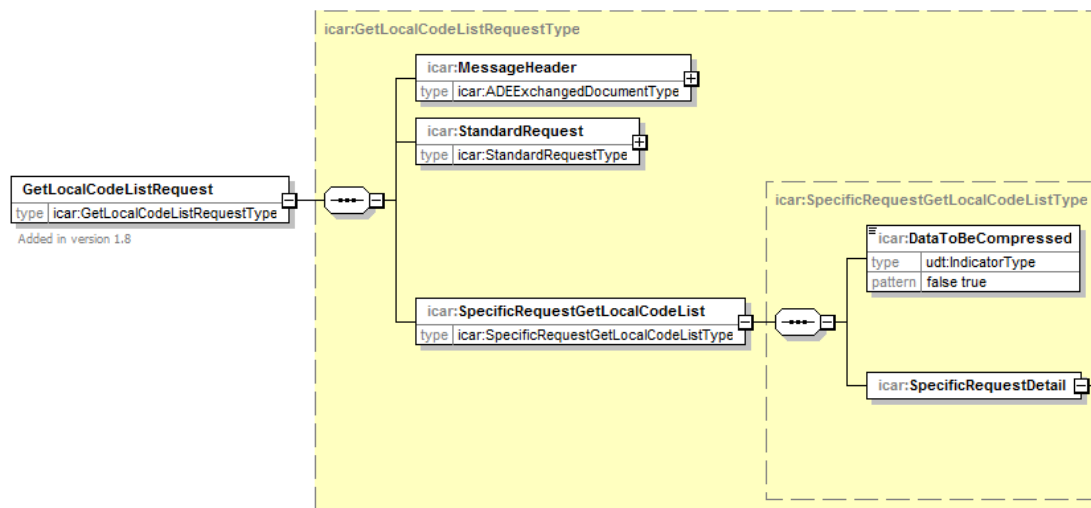
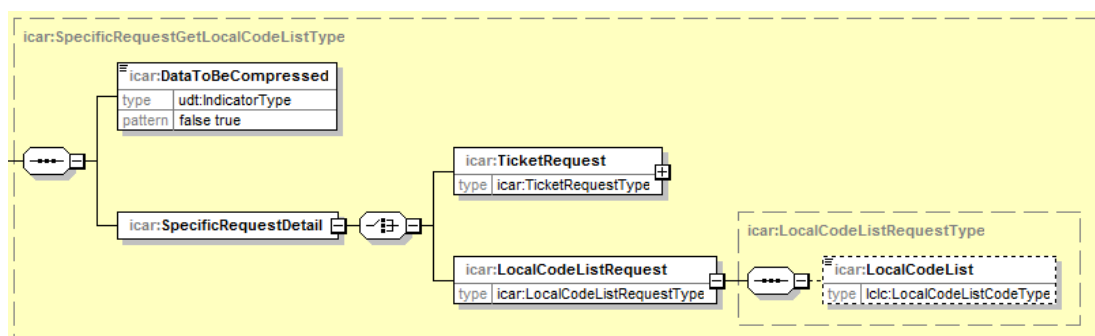


Figure 3 SpecificRequestGetLocalCodeList



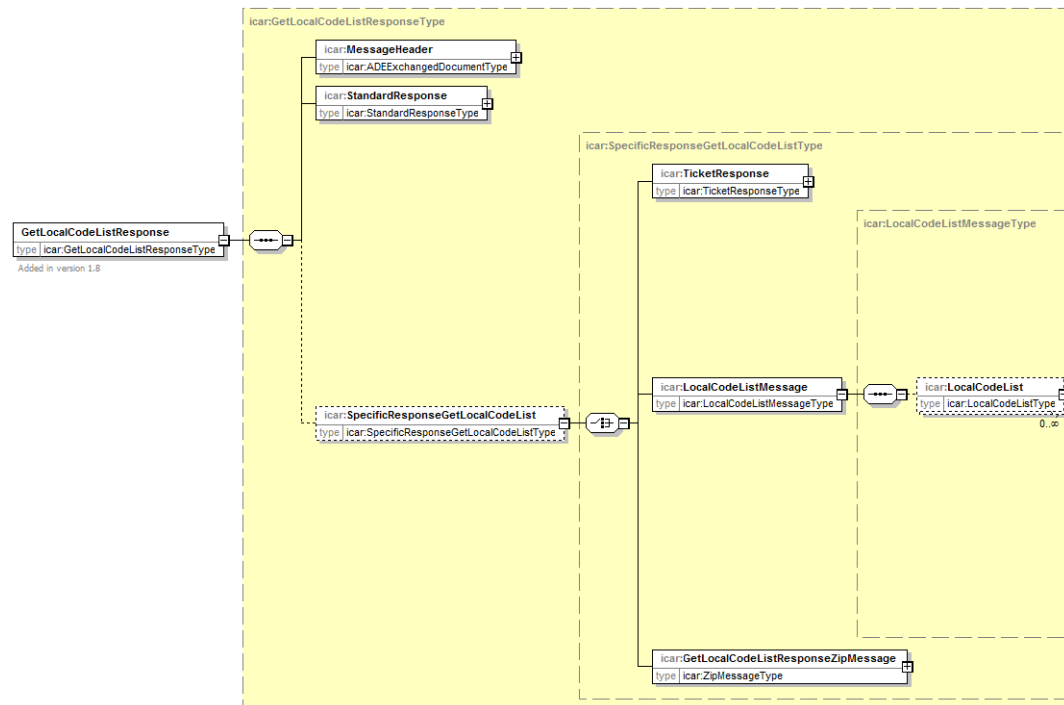
- a. **LocalCodeList**: Conveys the code of the LocalCodeListCode according to the code list **LocalCodeListCodeType** for which the local code list is requested. If missing the content of all local code lists will be returned.

4.3 Response description

Message **GetLocalCodeListResponse** (see diagram below) is in compliance with the general specifications for results (see General specifications / Response specifications).

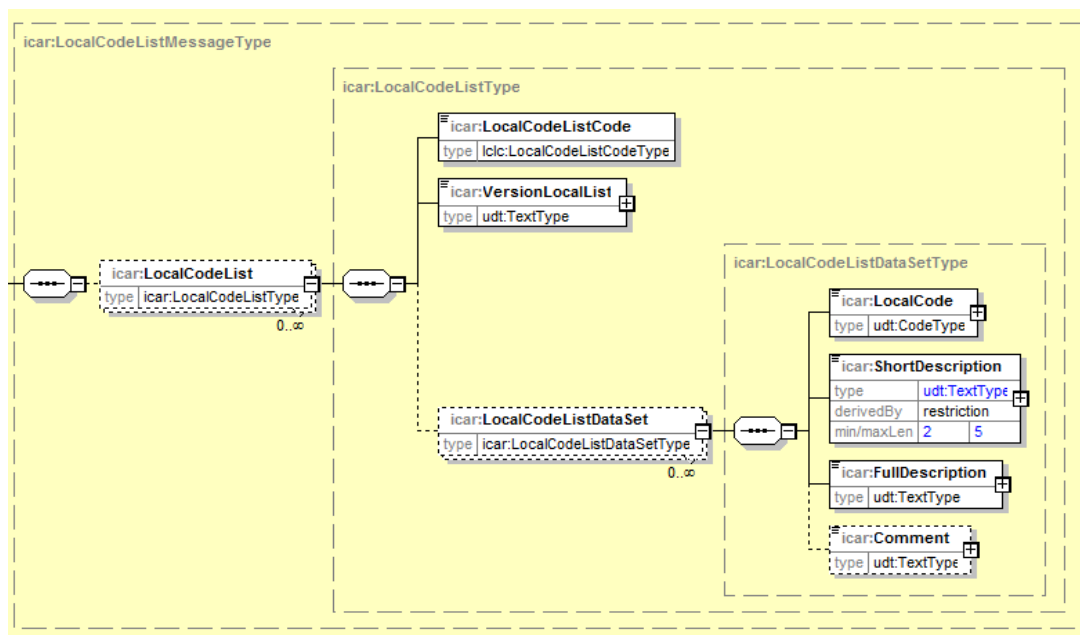
It provides the content of one or all local code lists depending on the request parameter **LocalCodeList**.

Figure 4 GetLocalCodeListResponse



The component **LocalCodeListMessage** conveys a list of **LocalCodeLists** with each of it representing the content of a single local code list.

Figure 5 LocalCodeList



- a. **LocalCodeListCode**: The identifier of a local code list according to the ICAR code list **LocalCodeListCodeType** (see section “Code Lists”)

- b. **VersionLocalList:** Version identifier of the provided local code list in order to detect changes and assure compatibility
- c. **LocalCodeListDataSet:** the list of code details of the code list
 - **LocalCode:** the unique code of a code list record
 - **ShortDescription:** Abbreviation of the code description e.g. to be displayed as a small label
 - **FullDescription:** Full code description e.g. to be displayed in wider display components
 - **Description:** Description of a code e.g. used by developers for a better understanding

5 Service UpdateDevice

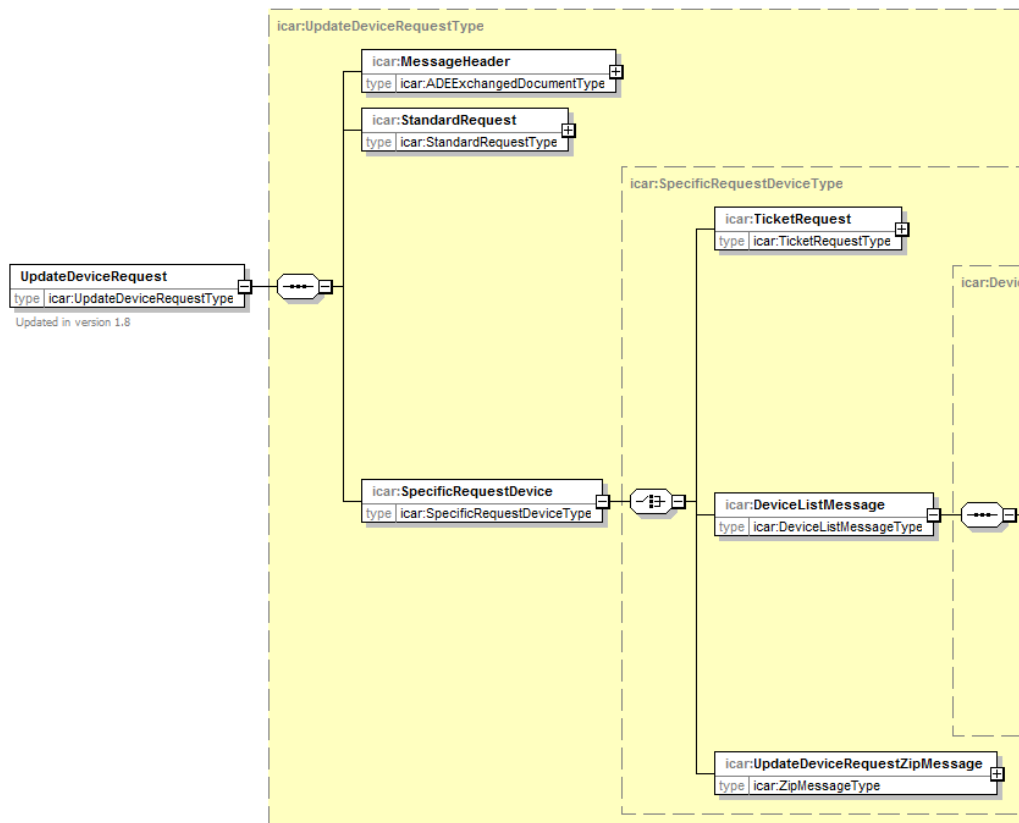
5.1 Purpose

The service **UpdateDevice** is designed to inform the farm management system about details of a milking or analysis device referenced by device id in the services described above. The details might be useful for correct interpretation of the data produced on this device. Updates should be transmitted as soon as devices are referenced or relevant information of referenced devices has changed.

5.2 Request description

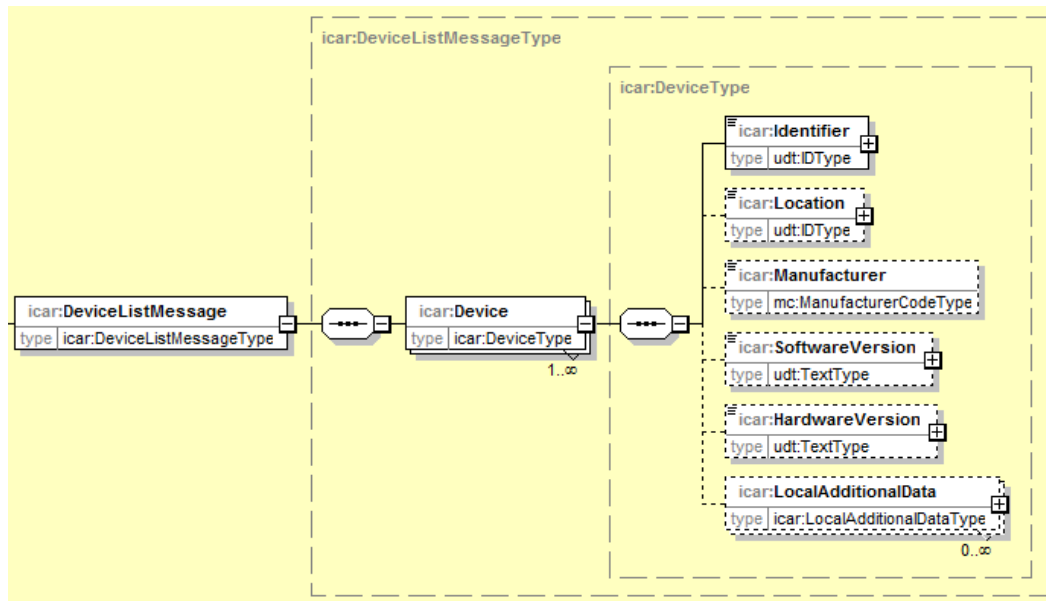
Message **UpdateDeviceRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

Figure 6 UpdateDeviceRequest



DeviceListMessage: A list of components of type **DeviceType** describing devices referenced by device id in the messages described above.

Figure 7 DeviceListMessage



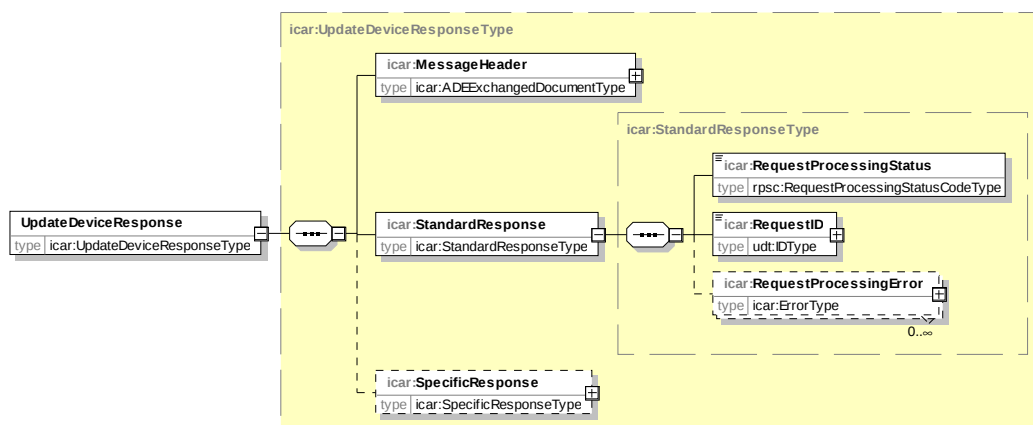
- a. **Device:** conveys the identifier and the characteristics of a device used to get a set of given results
- b. **Identifier:** unique identifier of the device
 Best practice proposals on the creation of unique device identifiers:
 - The device could be identified by the MAC address of its network interface which is globally unique
 - In case two and more devices share one network interface, a local unique postfix should be added to the MAC address
- c. **Location:** current location of the device
- d. **Manufacturer:** manufacturer of the device according to code list **ManufacturerCode**
- e. **SoftwareVersion:** current version of the device software
- f. **HardwareVersion:** current version of the device hardware
- g. **LocalAdditionalData:** List of locally defined code/value pairs

5.3 Response description

Message **UpdateDeviceResponse** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Response specifications)

It is used to receive the status of device detail processing by the service provider.

Figure 8 UpdateDeviceResponse



6 Service UpdateLivestockLocation

6.1 Purpose

The service **UpdateLivestockLocation** is designed to inform the farm management system about details of a locations revered by Location id in the services described above. Updates should be transmitted as soon as locations are referenced or relevant information of referenced locations has changed.

A livestock location is an identified place where an information system can localize the animals.

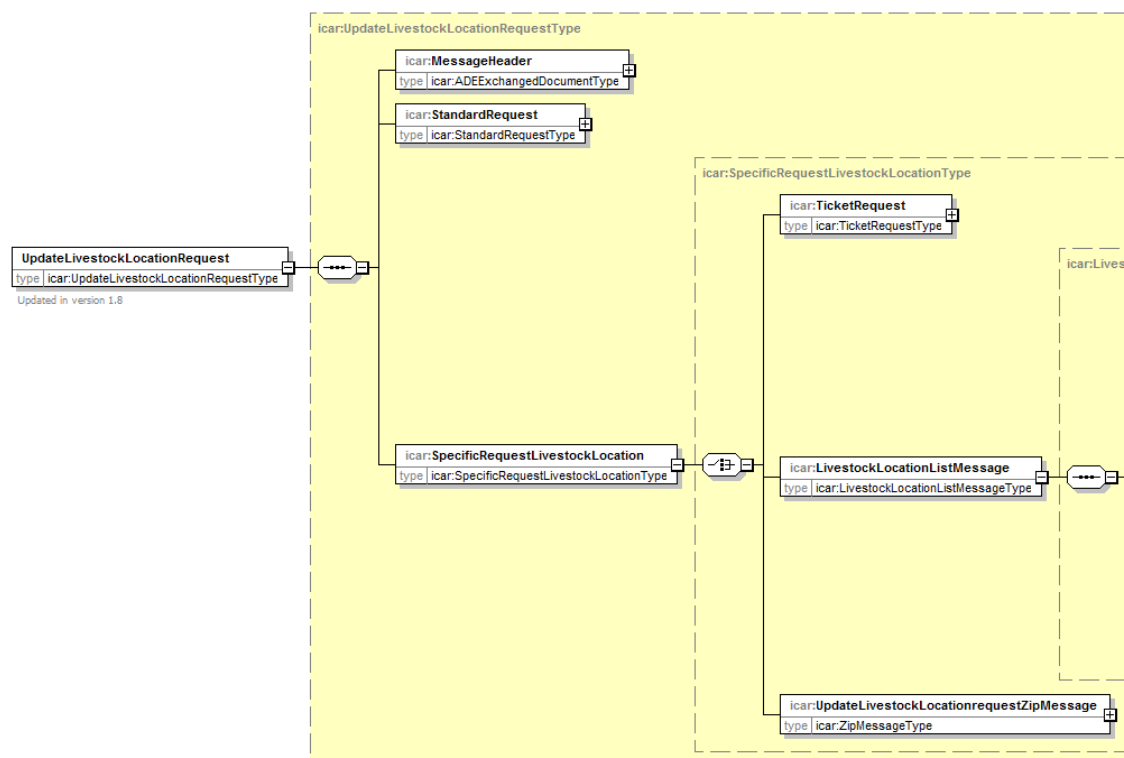
It is not possible to define a concept which may be usable everywhere. Different types of livestock location exist: farm, barn, holding, premise... Moreover for given type of location, a farm for example, different identifiers may exist: that given by the milk recording organization, that given by the government... For the European Union, in most cases, the holding identifier as defined by the EU regulation for animal traceability is used. The type of livestock location as well as its identifier depends on the business context. Hence before starting data exchange, the sender and the recipient should agree on the

- a. The type of location
- b. The type of identifier

6.2 Request description

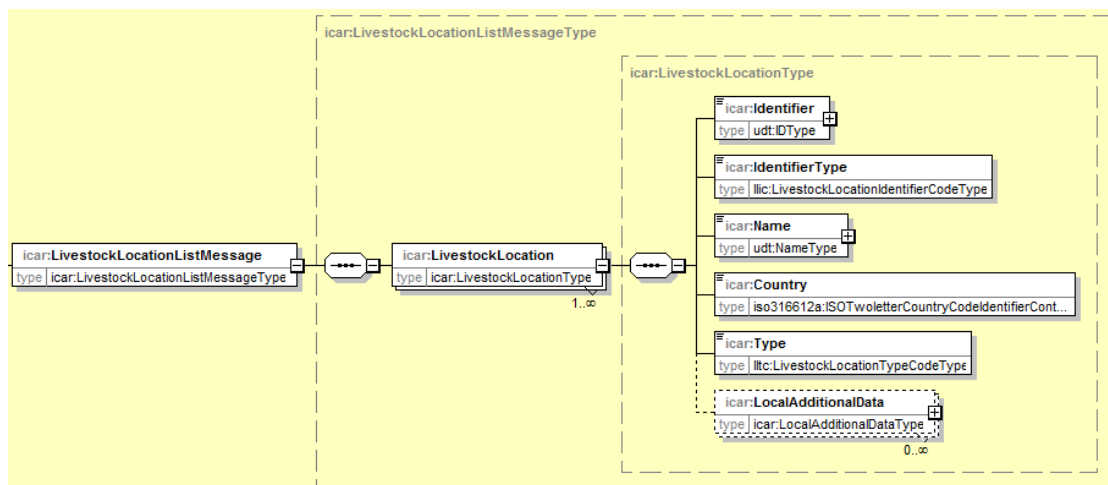
Message **UpdateLivestockLocationRequest** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Request specifications).

Figure 9 UpdateLivestockLocationRequest



LivestockLocationListMessage: A list of components of type **LivestockLocationType** describing locations referenced by **Location id** in the messages described above.

Figure 10 LivestockLocationListMessage



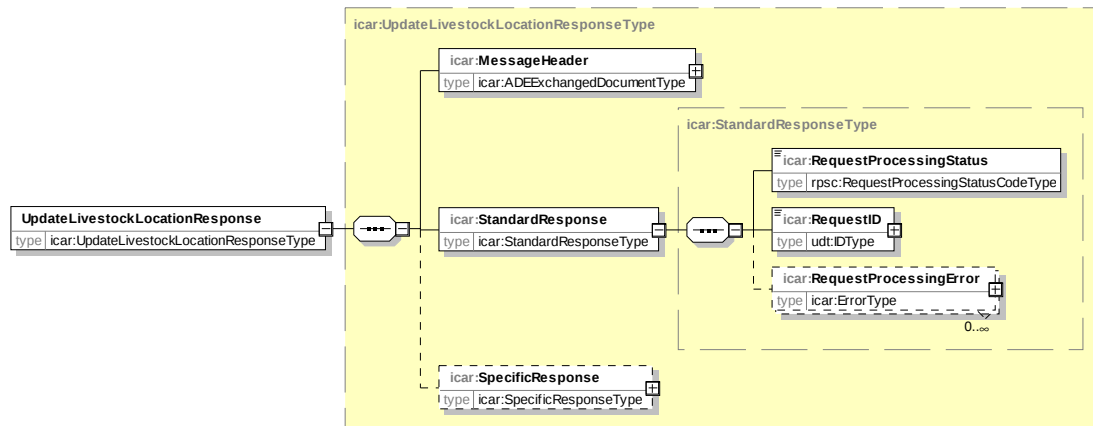
- a. **Identifier:** Identifier of a livestock location according to the locally defined schema of location id creation
- b. **IdentifierType:** Type of location identifier according to code list **LivestockLocationIdentifierCode** and locally permitted identifier schemas
- c. **Name:** The name of the location
- d. **Country:** The country the location is belonging to according to code list **ISOTwoletterCountryCode**
- e. **LocalAdditionalData:** List of locally defined code/value pairs

6.3 Response description

Message **UpdateLivestockLocationResponse** (see diagram below) is in compliance with the general specifications for requests (see General specifications / Response specifications)

It is used to receive the status of location detail processing by the service provider.

Figure 11 UpdateLivestockLocationResponse





THE GLOBAL STANDARD
FOR LIVESTOCK DATA

Appendix A of Section 15 of ICAR Guidelines – Data Dictionary

Data Dictionary

Version July, 2018

File Ref: 15 Appendix A Data Dictionary v18.01.docx

Table of Contents (Table of contents)

1	Introduction	4
2	Definitions and Terminology	4
3	Scope	4
4	Code Lists	4
4.1	IANA_CharacterSetCode	4
4.2	IANA_MIMEMedia	5
4.3	ICAR_BottleIdentifierCode	5
4.4	ICAR_ErrorSeverityCode	5
4.5	ICAR_GenderCode	5
4.6	ICAR_HeatDetectionMethodCode	5
4.7	ICAR_LabAnalyzerCode	5
4.8	ICAR_LabelCode	6
4.9	ICAR_LivestockLocationIdentifierCode	6
4.10	ICAR_LivestockLocationTypeCode	6
4.11	ICAR_LocalCodeListCode	6
4.12	ICAR_ManufacturerCode	7
4.13	ICAR_MilkCharacteristicCode	7
4.14	ICAR_MilkingTypeCodeType	8
4.15	ICAR_PregnancyCheckCode	8
4.16	ICAR_PregnancyCheckMethodCode	8
4.17	ICAR_QuarterCode	8
4.18	ICAR_ReproductionEventTypeCode	9
4.19	ICAR_RequestProcessingStatusCode	9
4.20	ICAR_SemenTypeCode	9
4.21	ICAR_SexingSemenCode	9
4.22	ICAR_SpecieCode	10
4.23	ICAR_ValidSampleFillingIndicatorCode	10
4.24	ISO_ISO3AlphaCurrencyCode_20120831	10
4.25	ISO_ISOTwoletterCountryCode	10
4.26	UNECE_AgencyIdentificationCode	10
4.27	UNECE_CharacterSetEncodingCode	10
4.28	UNECE_MeasurementUnitCommonCode	10
5	ICAR Milk Characteristics Codes	10

Tables

Table 1. Definitions of Terms used in these guidelines. 4

Figures

Figure 1. Example of xsd file contents..... 4

Summary of Changes

Date of Change	Nature of Change
July 2018	Data Dictionary moved from Section 15 Overview into this Appendix A and updated with new version from ADE-WG.

1 Introduction

For standardized electronic data exchange a data dictionary is a basic requirement. It provides a complete listing, definition and description of all elements in use.

The xsd files are from now on regarded as the primary data dictionary. Descriptions of the elements have to be maintained inside the xsd files using special xsd:document elements:

Example:

```
<xsd:element name="GetAnimalRequest" type="icar:GetAnimalRequestType">  
  <xsd:annotation>  
    <xsd:documentation>Message GetAnimalRequest is used to request the core  
data describing an animal stored in the farm management  
system</xsd:documentation>  
  </xsd:annotation>  
</xsd:element>
```

Figure 1. Example of xsd file contents.

A human readable hyperlink document will be created from the xds files using the export features of tools like e.g. XMLSpy. The dictionary document will be published on the ICAR ADE website together with the specifications and the wsdl/xsd files.

2 Definitions and Terminology

Table 1 contains a list of important definitions for terms and abbreviations used in these guidelines.

Table 1. Definitions of Terms used in these guidelines.

Term	Definition
wsdl	Web Services Description Language (WSDL / 'wɪz dəl/) is an XML-based interface definition language that is used for describing the functionality offered by a web service.
XMLSpy	XMLSpy is a proprietary XML editor and integrated development environment (IDE) developed by Altova.
xsd	XSD (XML Schema Definition), a recommendation of the World Wide Web Consortium (W3C), specifies how to formally describe the elements in an Extensible Markup Language (XML) document.

3 Scope

This appendix contains code lists and milking characteristic codes.

4 Code Lists

4.1 IANA_CharacterSetCode

For a full code list see file IANA_CharacterSetCode_20130108.xsd

4.2 IANA_MIMEMedia

In this context only code “application/zip” is used.

For a full code list see file IANA_MIMEMediaType_20130103.xsd

4.3 ICAR_BottleIdentifierCode

The type of bottle identifier method.

Code	Definition
BRC	Barcode
RFD	RFID

4.4 ICAR_ErrorSeverityCode

The error severity code.

Code	Definition
0	Warning
1	Data error
2	Fatal error

4.5 ICAR_GenderCode

The gender code.

Code	Definition
F	Female
M	Male

4.6 ICAR_HeatDetectionMethodCode

The heat detection method.

Code	Definition
C	Chemical
V	Visual
P	Pedometer
O	Other

4.7 ICAR_LabAnalyzerCode

The lab analyzer model.

Code	Definition
11	Foss FT6000
12	Foss FT+
21	Delta FTIR automatic
22	Delta FTIR 600

31	Bentley FTS
----	-------------

4.8 ICAR_LabelCode

The type of animal identifier.

Code	Definition
EVN	EU legal visual identifier
ISM	Animal ISO RFID identifier starting with a manufacturer code (ISO 11 784)
ISL	Animal ISO RFID legal identifier starting with a country code (ISO 11 784)
FAN	Farm animal name
HBN	Herd Book name
FNB	Farm number
NKN	Necklace number

4.9 ICAR_LivestockLocationIdentifierCode

The type of livestock location identifier.

Code	Definition
LEH	Legal Location Identifier
MRO	Location ID provided by a milk recording organization
HBO	Location ID provided by a herd book organization
OLI	Other local location identifier type

4.10 ICAR_LivestockLocationTypeCode

The type of location identifier.

Code	Definition
LEH	Legal European holding
	TODO: No clear idea of the usage at the moment. Item will be removed in the final version if nobody gives good reasons and provides a useful list of codes during the review period.

4.11 ICAR_LocalCodeListCode

This code list conveys a list of items of type `udt:CodeType` for which a local code list must be provided.

It also provides a list of code list names for locally allowed values in item **AdditionalDataCode** of entity **LocalAdditionalDataType**. These code lists can be identified by the prefix **LocalAdditional** + the name of the entity wherein the **LocalAdditionalData** entity is used.

Code	Definition
ArrivalReason	Arrival Reason for an animal
Breed	Breed for an animal
DepartureReason	Departure Reason for an animal
InseminationOrganisation	Insemination Organisation
LocalAdditionalAnimalCoreData	Local code for AnimalCoreData
LocalAdditionalAnimalMilkingResultData	Local code for AnimalMilkingResultData
LocalAnimalMilkingLabResult	Local code for AnimalMilkingLabResult
LocalAdditionalArrivalData	Local code for ArrivalData
LocalAdditionalDepartureData	Local code for DepartureData
LocalAdditionalDeviceData	Local code for DeviceData
LocalAdditionalLiveStockLocationData	Local code for LiveStockLocationData
LocalMilkCharacteristic	Codes for locally used milk characteristic
ParturitionEase	Ease level for a parturition

4.12 ICAR_ManufacturerCode

The code for the manufacturer of a device.

Code	Definition
LL	Lely
DL	Delaval
FW	Fullwood
GE	GEA
BM	Boumatic
AF	Afikim

4.13 ICAR_MilkCharacteristicCode

A list of ICAR approved milk characteristics.

Code	Definition
SCC	Somatic cell count
FAT	Fat
PROTEIN	Protein
LAC	Lactose
UREA	Urea
BLOOD	Blood
ACTETONE	Acetone

BHB	Beta hydroxybutyrate
LDH	Lactate dehydrogenase
PRO	Progesteron
AVGCOND	Average conductivity value of the milk at 25 ° C
MAXCOND	Maximum conductivity value of the milk at 25 ° C
AVGFLWR	AverageFlowRate
MAXFLWR	MaxFlowRate
WEIGHT	Weight of animal

4.14 ICAR_MilkingTypeCodeType

A list of codes representing the types of milking.

Code	Definition
1	Official milk result supplied by milk recording organisation
2	Measure by ICAR approved equipment
3	Measure of not approved equipment

4.15 ICAR_PregnancyCheckCode

A list of codes representing the result types of a pregnancy check.

Code	Definition
U	Unknown
E	Empty
P	Pregnant

4.16 ICAR_PregnancyCheckMethodCode

A list of codes representing the methods of a pregnancy check.

Code	Definition
E	Echography
P	Palpation
O	Other

4.17 ICAR_QuarterCode

A list of codes representing the position of an udder quarter.

Code	Definition
LF	Left front
RF	Right front
LR	Left rear

RR	Right rear
----	------------

4.18 ICAR_ReproductionEventTypeCode

A list of codes representing the types of reproduction event.

Code	Definition
CAL	Calving
PGC	Pregnancy Check
INS	Artificial insemination, Embryo transfer, Assisted mating
PEM	Period mating
DRO	Dry off
HET	Heat
ABT	Abortion

4.19 ICAR_RequestProcessingStatusCode

A list of codes representing the possible return status of a request processing.

Code	Definition
O	Processed without errors
P	Data accepted for asynchronous processing (client can retrieve the processing result later using provided ticket)
E	Processed and rejected due to errors
W	Processed with warnings

4.20 ICAR_SemenTypeCode

A list of codes representing the semen types.

Code	Definition
F	Fresh
Z	Frozen

4.21 ICAR_SexingSemenCode

A list of codes representing the sexed semen type.

Code	Definition
Y	Sexing semen : Male or Female
F	Sexing semen : Female
M	Sexing semen : Male
N	No sexing semen

4.22 ICAR_SpecieCode

The code for the animal species.

Code	Definition
BUF	Buffalo
OVI	Ovine
BOV	Bovine
GOT	Goat

4.23 ICAR_ValidSampleFillingIndicatorCode

A list of codes representing the milk sample filling indicator.

Code	Definition
0	successful milking (> 80% of expected milk)
1	incomplete (< 20 % of expected milk) or interrupt milking
2	milking complete, measurement not complete (between 20 and 80 %)

4.24 ISO_ISO3AlphaCurrencyCode_20120831

A list of two letter codes representing the currencies according to ISO.

For details see file ISO_ISO3AlphaCurrencyCode_20120831.xsd

4.25 ISO_ISOTwoletterCountryCode

A list of two letter codes representing the countries according to ISO.

For details see file ISO_ISOTwoletterCountryCode_SecondEdition2006VI-13.xsd

4.26 UNECE_AgencyIdentificationCode

A list of codes representing the measurement unit defining agencies according to UNECE.

For details see file UNECE_AgencyIdentificationCode_D12A.xsd

4.27 UNECE_CharacterSetEncodingCode

A list of codes representing character set encodings according to UNECE.

For details see file UNECE_CharacterSetEncodingCode_40106.xsd

4.28 UNECE_MeasurementUnitCommonCode

A list of codes representing measurement units according to UNECE.

For details see file UNECE_MeasurementUnitCommonCode_8.xsd

5 ICAR Milk Characteristics Codes

Measurement of milk characteristics or characteristics related to milking like “weighing during milking” is a quickly evolving domain. For that reason the preference of the ICAR ADE standardization body tended toward defining a flexible container conveying milk characteristics as an open list of key value pairs with the identifying key being the characteristics code.

For details see section 5.1 “Milking.Common Components.MilkCharacteristicsType”

Code	Description	Unit	TYPE	DIGITS DECIMAL PLACES	PATTE RN	MI N	MAX	UnitCode UNCEFACT	Annotation UNCEFACT
SCC	Somatic cell count	x1000 cells/ml	NR	4	9990	0	9999	NCL	number of cells
FAT	Fat	percentage	NR	4.2	90.99	0	15.00	VP	percent volume
PROTEIN	Protein	percentage	NR	3.2	0.99	0	9.99	VP	percent volume
LAC	Lactose	percentage	NR	3.2	0.99	0	9.99	VP	percent volume
UREA	Urea	ppm, mg/l	NR	3	990	0	999	M1	milligram per litre
BLOOD	Blood	boolean	NR	1	0 1	0	1	undefined	implied
ACTETONE	Acetone	mmol/l	NR	4.3	0.999	0	9.999	M33	milli mole per liter
BHB	Beta hydroxybutyrate	mmol/l	NR	4.3	0.999	0	9.999	M33	milli mole per liter
LDH	Lactate dehydrogenase	IU/l	NR	4.3	0.999	0	9.999	undefined	implied
PRO	Progesteron	mmol/l	NR	4.3	0.999	0	9.999	M33	milli mole per liter
AVGCOND	Average conductivity value of the milk at 25 ° C	mS/cm	NR	2.1	0.9	0	9.9	H61	millisiemens per centimetre
MAXCOND	Maximum conductivity value of the milk at 25 ° C	mS/cm	NR	2.1	0.9	0	9.9	H61	millisiemens per centimetre
AVGFLWR	AverageFlowRate	Kg/min	NR	3.2	0.99	0	9.99	F31	kilogram per minute
MAXFLWR	MaxFlowRate	Kg/min	NR	3.2	0.99	0	9.99	F31	kilogram per minute
WEIGHT	Weight of animal	Kg	NR	4	9990	0	1500	KGM	kilogram