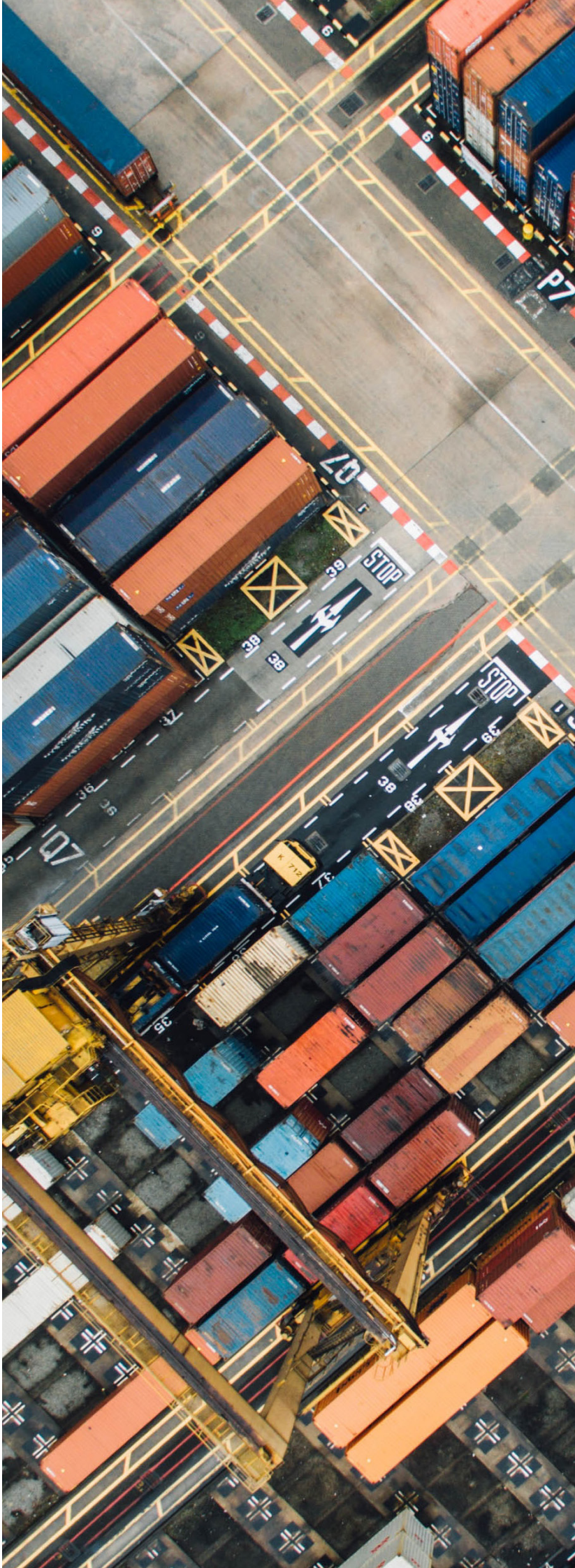
COA
UDM FOR REEFER CONTAINER DATA EXCHANGE FORMATS

Development by the Container Owners Association of a Unified Data Model (UDM) to provide a single, global definition of data related to reefer containers – a 'Reefer Data Classification Schema'

Version 1



CONTAINER OWNERS ASSOCIATION

COA
UDM FOR REEFER CONTAINER DATA EXCHANGE FORMATS 2021

CONTENTS

Published by:
Container Owners Association
Suite 3, Charter House, 26 Claremont Road,
Surbiton KT6 4QU United Kingdom

secretary@containerownersassociation.org
www.containerownersassociation.com

General Information
The Container Owners Association launched the **COA Reefer Telematics Data Work Group** in Rotterdam in November 2018, to develop a Standard API for Reefer Data.

This Document provides a Universal Data Model (UDM) to provide a single, global definition of data related to reefer containers – a 'Reefer Data Classification Schema'.

Participation
The following companies and organisations participated in the Work Group and have contributed to the information and data that it contains.

OEMs

- Carrier Transicold
- Daikin Refrigeration
- MCI – Maersk Container Industry
- Thermo King

Telematics System Suppliers

- Emerson
- Globe Tracker
- Nexxiot
- Orbcomm
- Traxens
- BoxPlus
- ArrowSpot

Secretariat

- Container Owners Association

Project Development and Consultation Process
The development of the UDM was undertaken through a series of of Project Meetings, taking place from May 2019 to March 2021, at which time a Final Draft was prepared for industry consultation. The consultation process was launched via a COA Webinar on 9 March 2021, when industry stakeholders were invited to review the document and submit feedback.

Following the consultation period, the Work Group reviewed industry feedback and made relevant amendments to the UDM.

Updated Version and Periodic Review Process
This updated Version of the UDM is posted on the COA website, with open access. After this first release, the COA Telematics Work Group will review further feedback and comments every six months. Feedback can be sent to: secretary@containerownersassociation.org

The UDM may be updated, based on any feedback received, on an annual basis (reactively). Otherwise, it will be reviewed and updated every 3 years (proactively).

Decisions on review and updates will be carried out in the presence of the COA Secretariat. Updated versions will be numbered and updated to the COA website – www.containerownersassociation.com

Published by:
Container Owners Association
Suite 3, Charter House, 26 Claremont Road,
Surbiton KT6 4QU
United Kingdom

E: secretary@containerownersassociation.org
W: www.containerownersassociation.com

INTRODUCTION

This document introduces a standard, Unified Data Model (UDM) for the representation and exchange of refrigerated container and associated telematics data. The purpose of the data model is to enable the exchange of information on refrigerated containers in a uniform way independently on the model and manufacturer. Data exchange by using the standard data model introduced in this document will provide refrigerated container operator with mixed fleets the convenience of accessing relevant data without the complexity of accessing them with disparate formats and means.

The UDM will be implemented via standard compliant web services or application program interfaces (API) that will be published by holders of data in computer servers. A consumer of data will access information about fleet of refrigerated container equipment and their associated telematics data by calling APIs compliant with the UDM. The UDM will support any compliant fleet, container types, and telematic equipment.

Consumers of data including container operators will access UDM compliant APIs by sending HTTPS GET requests to the server of a data holder. The server will respond by returning equipment data structured in a standard schema using the definitions and classifiers defined in this document.

The goal of creating this standard is to benefit container operators to access their own fleet's refrigerated container data consistently and uniformly. The use of the UDM will enable users or their assigned third-party application developers to build new useful data driven applications. On the other hand, unauthorized aggregated access of data across fleets by third parties will be prevented as UDM compliant APIs will have provisions to be access controlled and secure.

- This document has been produced by experts within the field of container telematics.
- This document does not describe how a telematic device on a refrigerated container sends data to a backend endpoint (server)
- This document does not describe how consumers intend to use the data after having accessed them via API

1.1 SCOPE

Most data from a refrigerated container is collected and elaborated by its controller. A telematics data-logging and reporting device connected to the controller reads and stores controller data as well as acquires telemetry data via its own sensors. The device transmits all data to remote cloud servers typically owned by the device provider. Data in a telematics provider's server may be acquired by third-party client applications or via the internet via API.

The UDM standard specifications in this document apply to the API for server to application data transmission. They include the instructions and commands used to request data from the server and the format of the responses from the server which contain refrigerated container data. In particular, the UDM provides standard data attributes definitions for temperature conditions as wells as machinery operational state and telemetry.

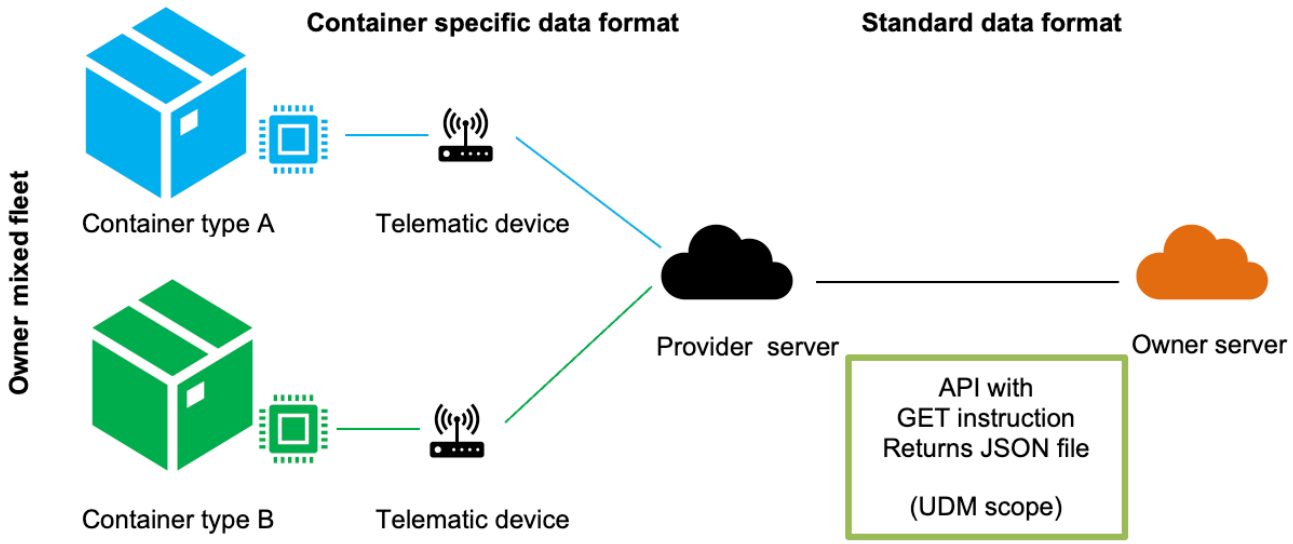


Figure 1 – Mixed fleet data flow and UDM scope

1.2 TERMS, DEFINITIONS AND ABBREVIATED TERMS

Backend - a part of a computer system or application, typically responsible for storing and manipulating data collected by multiple devices. Synonyms used in this document: Server, Cloud.

Container - in this particular document only, "container" refers to a refrigerated container (reefer) and is a shipping container used in intermodal transportation of temperature-sensitive cargo, which requires temperature regulation and control through cooling process. Synonyms used in this document: Reefer, Refrigeration unit, Refrigerated container.

Controller - a reefer container electronic component, which acquires information from sensors and operates the refrigerated container through commands. Information on measurements and state in a controller can be extracted electronically via communication channels by an external device.

Device - an IoT device which connects wirelessly to a network and has an ability to transmit data. Types of devices vary based on installation location (external/internal), life cycle (single trip/permanent), etc. In this particular document only, a "device" is part of the equipment of a container. A device part of a refrigerated container is connected to the controller.

Hardware - a collection of physical components of a particular device, which can include the hard case- the "box", battery, electronic components, semiconductors, embedded sensors, connectivity elements.

Intermodal transportation - a process involving two or more different modes of transport in conveying goods.

Optional - data element or minimum set of data entries, which follow the UDM standard but is not required to be available in a UDM compliant data transaction.

Public - an item/element/object dedicated to public use, reuse and redistribution by any user of a standard with no existing legal restrictions on access or usage. Public objects have standard definition.

Private - an item/element/object is made not available to the general public and dedicated to concrete authorised users only. The definition of private objects cannot be fully standardized or deviate from the standard.

Required - data element or minimum set of data entries, common across all refrigeration equipment manufacturers and telematics providers, which is always available in a UDM compliant data transaction.

Source - a data origin. In reference to this document a source is the refrigerated container controller.

UDM - acronym for Unified Data Model

2. USE CASES

The main objective of the Reefer Data Classification UDM is to enable visibility and access of information and data about refrigerated containers in a uniform and standardized way. This has benefits for those who produce and consume data in terms of visibility across different refrigeration unit types and development of simpler, reusable application software. Customers (carriers and shippers) can have data visibility across their fleets in a uniform way.

This objective is met by introducing a standard data classification which defines data availability and regulates data access. As described in Section 2.3, the UDM introduces two main categories of data, Public and Private data. Identified Public data are given standard identifiers which allow uniform access by any data consumers. Furthermore, Public data are divided into Required and Optional to guarantee that a minimum (Required) set of data is always available to a data consumer for any refrigeration unit whose data comply with the UDM standard.

Note: the use cases below have been defined to support the technical definition of the UDM for data exchange regardless of who owns the data and has the right to data access, which is not in scope of this standard.

2.1 ACTORS

The following have been identified as actors making use of the Reefer Data Classification UDM.

Container operator - End users who utilize the data from a refrigerated container to feed their digital container operations and commercial services.

Data holder - Entities that acquire refrigerated container data from units via telematics devices and store them into a cloud system they are in control of.

Data processor - Entities that use refrigerated container data and elaborates them to provide added value information to their customers.

Reefer maker – Entities that manufacture container refrigeration units including its electronic controllers. Therefore, reefer makers own the electronic and measurement equipment of the refrigeration unit where data are first collected when a unit is in operation.

Public data admin – An entity whose responsibility is to govern, maintain and update the list of public data and its standard representation.

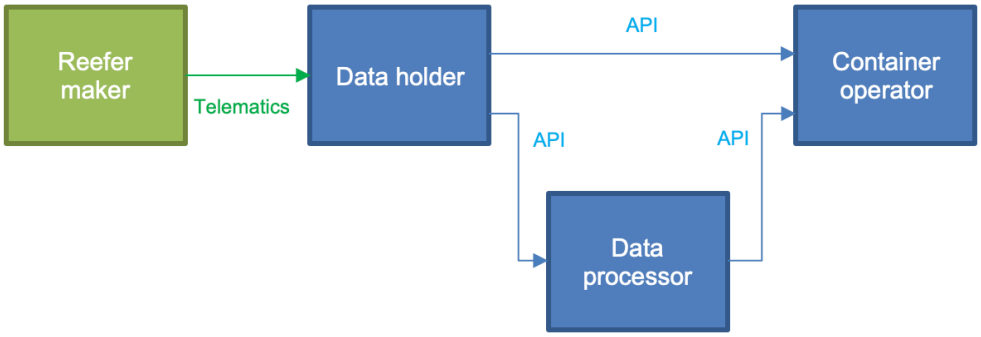
Actors generate, transform and make use of data as they flow from a refrigeration unit controller, to a gateway, to a first collection cloud environment and then to the end consumer of data, i.e. the container operator.

More details on the actors' roles are described in the following sections.

2.2 USE CASES

2.2.1 Summary

Within this standard, data transfer (flow) from actor to actor via transactions that are supported by UDM compliant APIs, except for data from Reefer Maker to Data Holder, which occurs via telematics data transfer via a device. The diagram below illustrates these data flows.



Access to refrigerated container data is needed by the actors as follows:

Case #	As a [persona]	I want to	so that
1	Container operator	Log and elaborate standardized public data for my refrigerated container fleet in my information system	I can track conditions of my refrigerated container fleet (which may be made by units of different manufacturers) with guaranteed access to uniform information like location, box temperature, etc (Public information).
2	Container operator	Log and elaborate standardized private data for my refrigerated container fleet in my own information system	I can uniformly access additional information about my refrigerated container fleet if available.
3	Data holder	Provide access to public refrigerated container data I acquired via telematics via API if asked by a Container operator.	I can do so in a uniform way without having to issue custom API formats for specific customers.
4	Data holder	Provide access to private refrigerated container data I acquired via telematics via API to specific customers (Data Processors or Container operator)	I can do so in a uniform way without having to issue custom APIs for specific customers
5	Data processor	Log and elaborate public and (if available) private data on behalf of a Carrier	I can provide a Container operator with additional valuable fleet information. Use of standardized data allows me to provide information more uniformly and efficiently and extend the UDM for nonstandard information.
6	Reefer maker	Ensure that data generated by a refrigeration unit I make are consistent with the Unified Data Model.	I can provide my customers with reefer information that is consistent and retrievable in the most efficient way.
7	Public data admin	Govern the Public Required and Public Optional data rapidly and efficiently.	I can ensure the UDM model is always up to date with latest data requirements fulfilling the Container operator needs.

A more detailed description of actors and typical workflows is described in the next sections.

2.2.2 Reefer maker
Entities that manufacture container refrigeration units including its electronic controllers. Therefore, reefer makers own the electronic and measurement equipment of the refrigeration unit where data are first collected when a unit is in operation.

i. A reefer maker typically manufactures or sources the refrigeration unit controller which collects data associated with the reefer operation.
ii. A reefer maker makes data from the controller accessible to a data collection (telematics) device usually by means of a proprietary transmission protocol.
iii. A reefer maker complying with the Reefer Container UDM standard will make Public Required data available in the transmission protocol.
iv. Because the UDM standard data are made available in the controller protocol, it will be possible to transfer them to the Data holder backend so it can generate UDM compliant APIs

2.2.3 Data holder
Entities that acquire refrigerated container data from units via telematics devices and store them into a cloud system they are in control of.

i. A data holder is typically responsible for the hardware and software of the device installed on the Container Operators' equipment
ii. A Data holder is typically responsible for enabling the devices to interface with the equipment Controller.
iii. The Data holder is responsible for acquiring the rights from the Reefer maker to interface with their equipment's controllers.
iv. The protocol licence defines which properties are available to that particular data holder. It will contain at least all Public Required data if the reefer maker complies with the Reefer Container UDM model. It may also include Public Optional and Private data.

2.2.4 Data processor
Entities that use refrigerated container data and elaborates them to provide added value information to their customers (container operators).

i. A Data processor will acquire reefer data from a Data holder to analyse and transform it into derived data.
ii. Data processor and Data holder may exchange reefer data through API (cloud to cloud).
iii. If the Data holder complies with the Reefer Data UDM, the API will conform to the UDM and contain at least Public Required data.

2.2.5 Container Operator
Container operators are in the business of transporting refrigerated containers from point A to point B or use them for fixed storage cooling. Container operators require remote access to reefer data to support their operations.

i. A container operator owns or leases in refrigerated containers which may be manufactured by various reefer makers.
ii. A container operator will have telematic devices installed on its refrigerated containers.
iii. The container operator will acquire services for data access from data holders and/or data processors via API.
iv. If compliant with the UDM standard, the API will conform to the UDM and contain at least Public Required data.

3 UNIFIED DATA MODEL (UDM)

The Unified Data Model (UDM) is a single, "global" definition of data related to refrigerated containers. The UDM can thus also be defined as a 'reefer data classification schema'.

All models of Reefer containers and controllers have common physical and logical properties, e.g. the return temperature, or the container id. Other properties are unique to a specific reefer container model and its configuration, for example the measurement of CO2 concentration in the box, or its alarm classification. The UDM model includes a standard definition and naming of all the common and unique relevant properties describing the operation of a refrigerated container.

For each property the UDM defines a set of attributes structured in a standard schema. Attributes include the identifier, the standard name, the description, and value within a range of admissible values.

As anticipated in Section 2, the UDM properties can be Public or Private. Public properties may be Required or Optional.

3.1 UDM STATES

In order to define the utilization and availability of properties in the UDM standard, each property is assigned a state. While the state of a property may change, a property may only have one state at a time. States defined under the UDM are "Public", "In Review", "Private" or "Obsolete".

3.1.1 Public
Public properties are common across all refrigeration equipment models and made available by manufacturers and telematics providers to their customers. UDM Public properties are accessible via standard UDM compliant API no matter what reefer model, controller and device the data is acquired from.

A Public property is assigned a unique identifier (pXXX). The Public property identifiers are publicly visible including any meta data and descriptions.

A Public property value may be "Blank" or "Not available" if that property cannot be extracted from a Controller type.

A Public property can be "Required" or "Optional".

- A Public Required property has a unique public identifier. The property and its value are always provided in a transaction when extracted by UDM compliant API.
- A Public Optional property has a unique public identifier. The property and its value may or may not be provided in a transaction when extracted by UDM compliant API.

3.1.2 In Review
A property state is In Review before is made available by UDM subscribers. Only the Public data admin and the Review committee has visibility of the property requiring revision.

The Review committee must include representatives from Reefer makers and Data holders:

- As Reefer makers enable access to refrigerated container properties via connection to the controller, they need to verify that the property is measurable, can be extracted by the controller, and can be included in the communication protocol to the device. In addition, they need to ensure that the property is properly modelled in the UDM.
- As Data Holders enable transfer of properties from the device to the back end, they need to ensure that devices are capable to handle the information associated with the property and APIs are created to support the standard property.

The review of a property may be initiated by Container Operators, Reefer Makers and Data holders. A property may be reviewed for addition to the Public list, deletion and modification.

At the end of the revision process a property "In Review" may be promoted to Public, deemed Private or may be dispositioned as Obsolete.

3.1.3 Private

Private properties are unique to the specific refrigeration units and the UDM standard does not specify its ids, attributes or admissible values. However, Private properties may be modelled consistently with the UDM so that an API including Private and Public data extraction has a uniform structure.

A Data holder can create his own Private properties and their definition with the UDM structure. There is no limit to how many Private properties can be defined, and no approval is needed by the Review committee. The Data holder will define the identifier, description and admissible values and will share API descriptions with selected data consumers as needed.

As identifiers must always be unique, each Data holder participating in the UDM will have a predefined key (issued by the UDM review committee) which must be included in each Private property identifier. This is to avoid that non unique identifiers are defined.

The structure of the data models is extensible with additional attributes as needed. It is however recommended that standard attributes are used as much as possible to enable consistency.

Examples:

A. Data Holder defines Private properties for measurements from custom sensors directly connected to the telematics device.
B. Data Holder extends a Public property with additional attributes or additional entry values for specific operating modes of the refrigerated container. For example, a “Chilled” mode, or “FunctionTest” mode available in a specific refrigerated container model.

The creation of such Private extensions needs to be carefully evaluated by the individual Data holders as it may lead to unwanted duplications, conflicts and compromise of data integrity.

4 STANDARD DATA

4.1 GENERAL

The UDM model aims to simplify and standardize the format of APIs used for transaction of refrigerated container data.

When transferring data from server A to server B, a transaction between A and B must contain data as specified in this section and conform to the required format described in Section . Moreover, a refrigerated container controller which supports the UDM standard must provide data included in the UDM tables.

UDM is encoded into JSON and exchanged via REST APIs between parties in a data transaction. By using the UDM model, each party must encode/decode the data from/into the internal (proprietary) data structures.

4.2 UDM PROPERTIES

The set of measurement data listed in 4.2.1, 4.2.2 use the common schema to provide a snapshot view of a fleet or an individual piece of equipment. A snapshot is a single point in time.

- Focus is on devices that retrieve data from reefer controller. It is the intention to extend the model in the future to support other sources, such as external sensors, gensets and dry containers.
- Furthermore it will be extensible to include support for genset, dry etc.

4.2.1 Measurement data
The measurement data elements listed below describing a unit of equipment.

4.2.1.1 Meta-data
Required:
- Key
- Name
- Description
- Unit
- EnumSet
- Type

To be considered:
- Compact serialization Max Range
- Compact serialization Min Range
- No of decimals
- Allow Manual Entry

5 STANDARD FORMAT

5.1 EQUIPMENT TYPE IDENTIFICATION

To decide the communication protocol and to correctly interpret data read from the controller on a specific reefer container, it is necessary to precisely determine details about the equipment. Each reefer maker provides a range of reefer models, controller models, options and versions. A specific reefer container will represent a certain combination of these details.

Historically this was referred to as the controller type; however, each reefer maker has a unique way to identify controllers and associated systems and options which continue to change, meaning that it is not practical to create a standard list of controller identifiers.

Therefore, the UDM standard defines a set of properties that together determine the equipment. Each property is of type Enum and relates to a certain Enum Set. Each reefer maker will maintain separate parts of these Enum Sets.

The relevant properties are:

Key	Name	Type	Description
pMK	Maker	Enum	All controller manufacturers: Daikin, Carrier, MCI, TK. See section 5.4.2.1 for all Maker details.
pCM	Controller Model	Enum	Each Maker is assigned a pCMxx range for maintaining controller model types.
pUM	Unit Model	Enum	Each Maker is assigned a pUMxx range for maintaining Unit Model.

The relevant properties and Enum Sets are, to the extend possible, defined in the following sections.

5.1.1 Example
A complete equipment identification would include data like this:

- Maker: “SuperContainer”
- Controller Model: “Standard”
- Unit Model: ?
- Options: CA

5.2 DATE AND TIME FORMATS

All date and time stamps in an electronic interchange are to be formatted as ISO 8601 “date and time” that includes the year, month, day, hour, minutes and seconds. It does not include fractional seconds. If the information available to the server does not include minutes and seconds, the minutes and seconds are set to zero.

It is expressed as universal coordinated time (UTC) with the use of the UTC indicator (“Z”).

Date and Time are expressed as: YYYY-MM-DDThh:mm:ssZ.

5.3 REEFER CONTROLLER PROPERTIES

5.3.1 Public, Required

The listed device properties below will have "Public Required" status and as such must be supported by all device types.

Property Name	Property Description	Key	Unit	Enum set	Type
Container ID	The equipment id of the controller – 4 bytes as 8 hexadecimal characters	P1			String
Source Power	Indicates if the source is connected to power	P2		PowerStates (see section 5.5.6)	Enum
Setpoint	The temperature setpoint	P3	CelsiusDegrees		Float
Supply	The supply temperature	P4	CelsiusDegrees		Float
Return	The return temperature	P5	CelsiusDegrees		Float
Ambient	The ambient temperature	P6	CelsiusDegrees		Float
Relative Humidity	Relative Humidity inside the container.	P7	Percentage		Float
CO2 reading	The CO2 level	P8	Percentage		Float
CO2 setpoint	The CO2 setpoint	P9	Percentage		Float
O2 reading	The O2 level	P10	Percentage		Float
O2 setpoint	The O2 setpoint	P11	Percentage		Float
Controller operating state	The controller operating state (controller specific)	P12		OperatingModes (see section 5.5.9)	Enum
Maker	The manufacturer of the container.	P13		Makers (see section 5.5.1)	Enum
Controller Model	The Controller Model	P14		ControllerModels (see section 5.5.2)	Enum
Controller time	Controller / data logger date and time	P15			Date Time
Supply 1	The supply sensor 1 temperature	P16	CelsiusDegrees		Float
Supply 2	The supply sensor 2 temperature	P17	CelsiusDegrees		Float
CA mode	Controlled Atmosphere Mode	P18		CAmodes (see section 5.5.10)	Enum
USDA 1	The temperature of USDA Probe 1.	P19	CelsiusDegrees		Float
USDA 2	The temperature of USDA Probe 2.	P20	CelsiusDegrees		Float
USDA 3	The temperature of USDA Probe 3.	P21	CelsiusDegrees		Float
Cargo	The cargo temperature probe sensor.	P22	CelsiusDegrees		Float

5.3.2 Public, Optional

The listed device properties below will have "Public Optional" status and as are not required to be

Property Name	Property Description	Key	Unit	Enum set	Type
Humidity setpoint	The humidity setpoint	P100	Percentage		Float
Active Alarm count	The number of active alarms.	P101			Integer
Current PhaseA	The phaseA current or Average current	P102	Amps		Float
Current PhaseB	The phaseB current	P103	Amps		Float
Current PhaseC	The phaseC current	P104	Amps		Float
Controller serial no.	The serial number of the controller.	P105			String
Source SW	The software version running in the source.	P106			String
Defrosting	True if defrost is in progress	P107			Boolean
Discharge pressure	The discharge pressure	P108	BarRelative		Float
Discharge temperature	The discharge temperature	P109	CelsiusDegrees		Float
Efficiency setting	The efficiency setting (controller specific)	P110		EfficiencyModes (see section 5.5.8)	Enum
Evaporator temperature	The evaporator temperature	P111	CelsiusDegrees		Float
Line frequency	The mains power frequency	P112	Hz		Float
Last Defrost	The end time of the last defrost period	P113			DateTime
Line voltage	The mains power voltage	P114	Volts		Float
Return 1	The return probe 1 temperature	P115	CelsiusDegrees		Float
Return 2	The return probe 2 temperature	P116	CelsiusDegrees		Float
Suction pressure	The suction pressure	P117	BarRelative		Float
Suction temperature	The suction temperature	P118	CelsiusDegrees		Float
Ventilation reading	The ventilation	P119	m3perH		Float
Source hardware revision	Source hardware revision	P120			String
Unit Model	The Unit Model	P121		UnitModels (see section 5.5.3)	Enum
Equipment Options	The possible option available on the container	P122		EquipmentOptions (see section 5.5.4)	List of Enums

5.4 DEVICE PROPERTIES

5.4.1 Public, Required

The listed device properties below will have “Public Required” status and as such must be supported by all device types.

Property Name	Property Description	Key	Unit	Enum set	Type
Device Time	UTC time automatically set by a wireless device or similar.	1	UTC		DateTime
Device name	The model type name of the device.	2		See appendices.	Enum
Received	The time the recording was received by the Data Holder cloud system	3	UTC		DateTime
Device Power	Indicates if the device is connected to power	4		PowerStates	Enum

5.4.2 Public, Optional

The listed device properties below will have “Public Optional” status and as are not required to be

Property Name	Property Description	Key	Unit	Enum set	Type
Device brand	The brand name of the device.	5		See 5.5.5	Enum

5.5 ENUMS

An enum type is a special data type that enables for a variable to be a set of predefined constants. The variable must be equal to one of the values that have been predefined for it. Common examples include compass directions (values of NORTH, SOUTH, EAST, and WEST) and the days of the week.

Each enum type is described and assigned a single unique key found in the previous section.

5.5.1 Controller Makers

This Enum Set identifies all manufacturers of Source equipment – for now, reefer container controllers.

Key	Range	Maker Name	Description
100	100-199	Thermo King	All refrigerated container equipment provided by Thermo King
200	200-299	Carrier	All refrigerated container equipment provided by Carrier Transicold
300	300-399	Daikin	All refrigerated container equipment provided by Daikin
400	400-499	MCI	All refrigerated container equipment provided by Maersk Container Industry (MCI)
...			...

5.5.2 Controller Model

All Controller Models from all manufacturers are defined in the following Enum Set. For utilization it is very useful if all controller models are defined within a single enum.

- COA assumes that there will “never” be more than 9 makers on the market
- COA assumes that the maker will ever have more than 99 models.

Controller Models from Maker N in the Makers Enum Set will have Key numbers N00 to N99. That maker will keep this Key range updated. Details for each Controller Model can be found in the respective appendixes.

5.5.3 Unit Model

All Unit Models from all manufacturers are defined in the following Enum Set.

Unit Models from Maker N in the Makers Enum Set will have Key numbers N00 to N99. That maker will keep this Key range updated.

5.5.4 Equipment Options

All Equipment Options from all manufacturers are defined in the following Enum Set.

Equipment Options that are considered general to all or multiple makers have Key numbers 0 to 99.

Separate Equipment Options from Maker N in the Makers Enum Set will have Key numbers N00 to N99. That maker will keep this Key range updated.

5.5.5 Device Brands

A Device is a component on the container communicating with the controller with the purpose of transferring data between the container and a site or cloud application.

Devices are standardized and easily replaced or moved within the network.

A device will always have a unique ID within the Device Type. Details for each Device Type can be found in the respective appendixes.

Key	Name
1	Emerson
2	Globetracker
3	Orbcomm
4	Nexxiot
5	Traxens
6	BoxPlus
7	ArrowSpot
8	MCI
9	Carrier

5.5.6 Power State

Key	Name	Description
1	Inactive	Inactive
2	OffPower	OffPower
3	OnPower	OnPower
4	Unknown	Unknown

Example

```
{
      “EnumSet”: “PowerStates”,
      “Key”: “2”,
      “Name”: “OffPower”,
      “Description”: “OffPower”
},{
      “EnumSet”: “PowerStates”,
      “Key”: “3”,
      “Name”: “OnPower”,
      “Description”: “OnPower”
}
```

5.5.7 Humidity Settings

Key	Name	Description
1	Off	Humidity control Off
2	On	Humidity control On
3	Unknown	Unknown

Example

```
{
      “EnumSet”: “HumiditySettings”,
      “Key”: “1”,
      “Name”: “Off”,
      “Description”: “Humidity control Off”
},{
      “EnumSet”: “HumiditySettings”,
      “Key”: “2”,
      “Name”: “On”,
      “Description”: “ Humidity control On”
}
```

5.5.8 Efficiency Settings

Key	Name	Description
1	Off	Off
2	Standard	Standard
3	Full	Full
4	Custom	Custom
5	Quest	Quest
6	QuestII	Quest II
7	EconomyMode	Economy Mode

Example

```
{
      “EnumSet”: “EfficiencyModes”,
      “Key”: “1”,
      “Name”: “Off”,
      “Description”: “Off”
},{
      “EnumSet”: “EfficiencyModes”,
      “Key”: “2”,
      “Name”: “Standard”,
      “Description”: “Standard”
}
```

5.5.9 Operating Modes

Key	Name	Description
1	Reefer Controlled Atmosphere	Reefer Controlled Atmosphere
2	Defrost	Defrost
3	Economy	Economy
4	Full cool	Full cool
5	Modulation	Modulation
6	Pre-trip inspection (PTI)	Pre-trip inspection (PTI)
7	Idle	Idle
8	Heating	Heating
9	Cool reduced	Cool reduced
10	Start up	Start up
11	Shut down	Shut down
12	Cooling	Cooling
13	Alarm	Alarm

Example

```
{
      " EnumSet ": "OperatingModes",
      "Key": "12",
      "Name": "Cooling",
      "Description": "Cooling"
},{
      " EnumSet ": "OperatingModes",
      "Key": "4",
      "Name": " Full cool",
      "Description": " Full cool"
}
```

5.5.10 Controlled Atmosphere Mode

The controlled Atmosphere Mode provides a snapshot of the CA mode setting.

Key	Name	Description
1	On	CA mode is On
2	Off	CA mode is Off
3	No_Unit	CA not installed

Example

```
{
      "EnumSet": "CAmodes",
      "Key": "3",
      "Name": "No_Unit",
      "Description": "No_Unit"
},{
      "EnumSet": "CAmodes",
      "Key": "2",
      "Name": "Off",
      "Description": "Off"
}
```

6 UNIFIED MODELS

6.1 UNIFIED PROPERTIES

The UnifiedProperties model is a model used to describe values across solutions and API's. In the essence the UnifiedProperties is a well defined list of properties with a key ("p1-pXXX"), name, description, unit, type, min/max etc.

The property list is maintained as described by the governance model.

Small sample of randomly picked Properties:

Request URL

```
https://              .com/api/core2/UnifiedModel?type=Properties
```

Response content

```
Date: Tue, 08 Dec 2020 13:29:48 GMT
Content-Length: 13018
Content-Type: application/json; charset=utf-8

[{
  "Key": "p5",
  "Name": "Device name",
  "Description": "The name of the device",
  "Type": "String"
}, {
  "Key": "p6",
  "Name": "Controller time",
  "Description": "Controller / data logger date and time",
  "Type": "DateTime",
  "Unit": "UTC"
}, {
  "Key": "p7",
  "Name": "Controller id",
  "Description": "The equipment id of the controller - 4 bytes as 8 hexadecimal characters",
  "Type": "String"
}, {
  "Key": "p10",
  "Name": "Setpoint",
  "Description": "The temperature setpoint",
  "Type": "Float",
  "Unit": "CelciusDegrees"
}, {
  "Key": "p11",
  "Name": "Supply",
  "Description": "The supply temperature",
  "Type": "Float",
  "Unit": "CelciusDegrees"
```

6.2 UNIFIED ALARMS

Just like the UnifiedProperties, the Unified alarm model is a model used to combine alarms from different controllers to avoid duplicates that basically means the same thing.

The proposed standard will be limited to sharing of the unique alarm codes as provided by the manufacturers. In the model the unique alarm code (“AlarmNo”) will be provided as well as the manufacturer information.

Small sample of randomly picked Alarms:

Request URL

```
https://                .com/api/core2/UnifiedModel?type=Alarms
```

Response content

```
Date: Tue, 08 Dec 2020 17:06:28 GMT
Content-Length: 70124
Content-Type: application/json; charset=utf-8

[{
  "Key": "a0",
  "AlarmNo": "0",
  "Description": "Power up",
  "Manufacturer": "ISO"
}, {
  "Key": "a1",
  "AlarmNo": "0",
  "Description": "Supply air sensor open circuit",
  "Manufacturer": "Thermoking"
}, {
  "Key": "a2",
  "AlarmNo": "1",
  "Description": "Sensor failure",
  "Manufacturer": "ISO"
}, {
  "Key": "a3",
  "AlarmNo": "1",
  "Description": "Supply air sensor short circuit",
  "Manufacturer": "Thermoking"
}, {
  "Key": "a4",
  "AlarmNo": "10",
  "Description": "Low compressor oil pressure",
  "Manufacturer": "ISO"
}, {
  "Key": "a5",
  "AlarmNo": "10",
  "Description": "Heater current too high",
  "Manufacturer": "Thermoking"
}, {
```

7 MESSAGE FORMAT

7.1 EXAMPLES OF MESSAGE PROTOCOL

7.1.1 JSON data to REST API

```
{
  “DeviceId”: “123456789123456”, // Device unique ID
  “DeviceType”: 5,               // Emerson RMM-W
  “SourceId”: “ABCD1234567”,     // ContainerId of the Reefer [optional]
  “SourceType”: 1,               // Container/Reefer [optional]
  “Logged”: “2019-03-14T07:23:00Z”, // Timestamp for the logging, following ISO8601
  “Properties”: {
      “p10”: 9.2,  // Setpoint
      “p58”: 56.1116263,  // Latitude
      “p59”: 10.1569933  // Longitude
      “p145”: “2.0.0.0”,  // Device software version
},
  “Alarms”: {
      “a14”: “2019-03-14T07:23:00Z”  // Value is when the alarm first appeared
}
```

7.1.2 Example Query Source Latest Data

The SourceLatestData resource retrieves latest available source data and adheres to UDM standards to represent data.

Request URL

```
https://              .com/api/query2/sourcelatestdata?sourceId=SIMT0000047&sourceType=Container&deviceType=RmmW
```

Response content

```
Date: Tue, 08 Dec 2020 18:02:09 GMT
Content-Length: 3430
Content-Type: application/json; charset=utf-8

{
  "SourceId": "SIMT0000047",
  "SourceType": 1,
  "DeviceId": "000071413000004",
  "DeviceType": 5,
  "Logged": "2020-12-08T17:54:00",
  "Properties": {
    "p145": {
      "Value": "99.99.1.40",
      "Logged": "2020-12-08T17:54:00"
    },
    "p144": {
      "Value": "DIDAmdm000071413000004",
      "Logged": "2020-12-08T17:54:00"
    },
    "p136": {
      "Value": "Emerson TS Location",
      "Logged": "2020-12-08T17:54:00"
    },
    "p16": {
      "Value": 1,
      "Logged": "2020-12-08T17:54:00"
    },
    "p17": {
      "Value": "2020-12-08T17:54:00",
      "Logged": "2020-12-08T17:54:00"
    },
    "p10": {
      "Value": 0.0,
      "Logged": "2020-12-08T17:54:00"
    },
    ,,
    "Alarms": {
      "a85": "2020-10-29T04:39:55"
    },
```

7.1.3 Example Push API

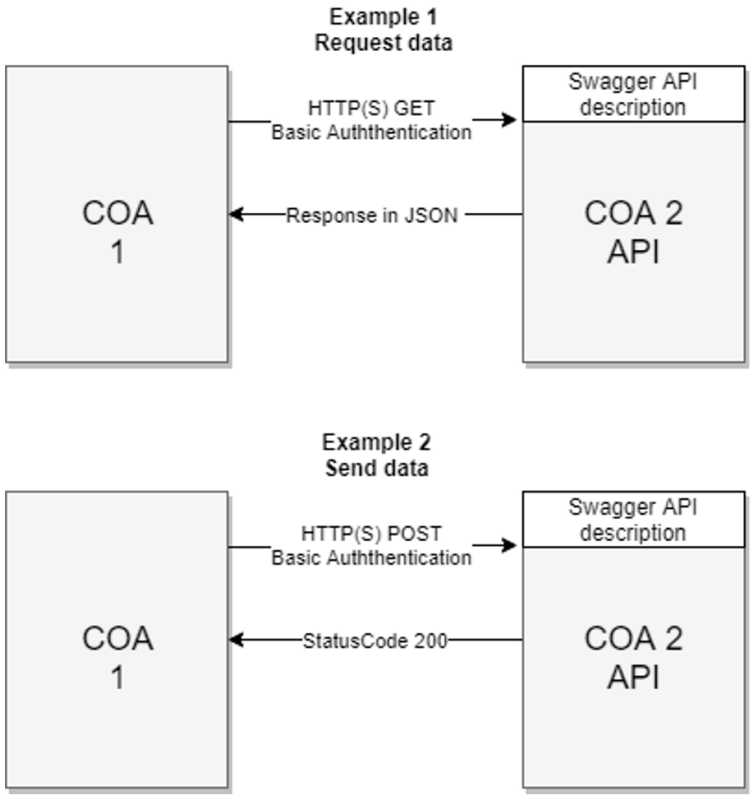
The service allows push data to an API end point configured in the system. The push data adheres to UDM standard. The sample snippet of push data:


```
{“SourceId”:”SIMT0000047”,”SourceType”:1,”DeviceId”:”000071413000004”,”DeviceType”:5,”Logged”:”2020-12-08T18:12:00”,”Properties”:{“p145”:”3.1.1_201045”,”p144”:”DIDAmdm000071413000004”,”p136”:”EmersonTSLocation”,”p61”:”209.0”,”p16”:0,”p17”:”2020-12-08T18:12:00”,”p10”:5.0,”p11”:5.8,”p83”:5.7,”p84”:5.8,”p12”:5.3,”p75”:5.3,”p76”:5.4,”p14”:92.0,”p24”:0.0,”p65”:0.0,”p49”:5.6,”p13”:14.0,”p40”:false,”p60”:”2020-12-07T19:49:00”,”p85”:”2020-10-21T02:00:00”,”p56”:”310170203637201”,”p7”:”82928292”,”p38”:”Thermo King MP-4000”,”p37”:”C:5168,DC:5168”,”p36”:”04990727”,”p33”:0.0,”p125”:true,”p122”:true,”p69”:5,”p67”:1,”p66”:”2020-11-14T11:38:00”,”p68”:”2020-11-14T11:35:00”,”p45”:1,”p19”:0.8,”p20”:1.4,”p21”:0.8,”p51”:60.0,”p63”:465.3,”p81”:-1.0,”p82”:0.0,”p43”:-1.0,”p44”:-196.8,”p78”:false,”p124”:3,”p22”:false},”Alarms”:{“a85”: “2020-10-29T04:39:55”}}
```


7.1.4 Description of the API

Examples using Basic authentication as proposed by NEXXIOT.

API description in swagger would be beneficial to require a less strict standardization around api routes and payloads. Both POST and GET payloads for a single device/source, could follow the model proposed in 3.1.1.



8 DOCUMENT REVISION RECORD A

Rev.	Date	Author	Brief description of change	Pages affected
0v1	081019	COA	Initial document	All
0v1	091019	COA	Cleaning tracked changes	All
0v2	291019	NP	Updated 2.1 (properties) to include enum example Updated 2.2 (alarms) per agreement, Updated 3.1.1 (JSON example) Updated 3.1.2 API description + examples New section 4.0 (Definition of properties) New sub Section 5.2.1,	
0v3	211119	BA	Categorized properties into Suggested & Required	
0v4	230120	NP	Complete re-org of document. New sections added and revised.	All
0v7	150320	NP	Added “1.2 Scope”, “3.0 Requirements”,	
0v8	080920	NP	New: 1.0 Introduction, Response formats, Changed: 1.1 Scope,	
1v0	240920	NP	Cleaned document for external sharing.	All
1v1	091220	NP	Revised section 6.5 and added new example to section 8	
1v5	270121	NP	Removed all reference to particular OEM’s. Added reference to appendixes.	

APPENDIX

CONTROLLER TYPE DEFINITIONS

DEVICE TYPE DEFINITIONS

CONTROLLER TYPES DEFINITIONS
THERMO KING

Definitions for controller types as per Section 5.4.2 of the UDM standard:

Key	Name	Type	Description
pMK	Maker	Enum	All controller manufacturers: Daikin, Carrier, MCI, TK.
pCM	Controller Model	Enum	Same as Maker. Each Maker is assigned a pCMxx range for maintaining controller model types.
pUM	Unit Model	Enum	Same as Maker. Each Maker is assigned a pUMxx range for maintaining Unit Model.

Maker

Key	Controller Type Id	Description
100	Thermo King	All refrigerated container machinery produced by Thermo King

Controller Model
Reefer maker: Thermo King

Key	Controller Model Id	Description
101	MP-3000	Magnum unit
102	MP-3000 SuperFreezer	Magnum unit with deep freeze function
103	MP-4000	Magnum+ unit
104	MP-4000 SuperFreezer	Magnum+ unit with deep freeze function
105	MP-4000 CFF	

Unit Model
Reefer maker: Thermo King

Key	Controller_Unit_Model_ID	Description
101	MP3000	Magnum unit
102	MP3000 SuperFreezer	Magnum unit with deep freeze function
103	MP4000	Magnum+ unit
104	MP-4000 SuperFreezer	Magnum+ unit with deep freeze function
105	MP-4000 CFF	

CONTROLLER TYPE DEFINITIONS
CARRIER

Definitions for controller types as per Section 5.4.2 of the UDM standard:

Key	Name	Type	Description
pMK	Maker	Enum	All controller manufacturers: Daikin, Carrier, MCI, TK.
pCM	Controller Model	Enum	Same as Maker. Each Maker is assigned a pCMxx range for maintaining controller model types.
pUM	Unit Model	Enum	Same as Maker. Each Maker is assigned a pUMxx range for maintaining Unit Model.

Maker

Key	Controller Type Id	Description
p2	Carrier	All refrigerated container machinery produced by Carrier Transicold

Controller Model
Reefer maker: Carrier

Key	Controller Model Id	Description
201	ML2i	ML2i controller
202	ML3	ML3 controller
203	ML5	ML5 controller

Unit Model
Reefer maker: Carrier

Key	Controller_Unit_Model_ID	Description
201	ThinLINE	ThinLINE reciprocating compressor unit
202	EliteLINE	EliteLINE scroll compressor unit
203	PrimeLINE	PrimleLINE scroll compressor unit
204	PrimeLINE Edge	PrimleLINE Edge scroll compressor unit
205	NaturaLINE	NaturaLINE CO2 natural refrigerant unit

CONTROLLER TYPE DEFINITIONS
DAIKIN

Definitions for controller types as per Section 5.4.2 of the UDM standard:

Key	Name	Type	Description
pMK	Maker	Enum	All controller manufacturers: Daikin, Carrier, MCI, TK.
pCM	Controller Model	Enum	Same as Maker. Each Maker is assigned a pCMxx range for maintaining controller model types.
pUM	Unit Model	Enum	Same as Maker. Each Maker is assigned a pUMxx range for maintaining Unit Model.

Maker

Key	Controller Type Id	Description
300	[Daikin]	All refrigerated machinery produced by Daikin

Controller Model - Reefer maker: Daikin

Key	Controller Model Id	Description
301	DECOS IIIc	DecosIIIc controller
302	DECOS IIId	DecosIIId controller
303	DECOS IIId-1	DecosIIId-1 controller
304	DECOS IIIe	DecosIIIe controller
305	DECOS IIIf	DecosIIIf controller
306	DECOS IIIg	DecosIIIg controller
307	DECOS IIIh	DecosIIIh controller
308	DECOS IIIj	DecosIIIj controller
309	DECOS V	DecosV controller
310	DECOS Va	DecosVa controller

Unit Model - Reefer maker: Daikin

Key	Controller_Unit_Model_ID	Description
301	LXE10E-A, -1	LXE10E – scroll compressor unit with DECOS IIIc, IIId or D-1 controller
302	LXE10E100A, 100B	LXE10E – scroll compressor unit with DECOS IIIe controller
303	LXE10E100F	LXE10E – scroll compressor unit with DECOS IIIf controller
304	LXE10E100G	LXE10E – scroll compressor unit with DECOS IIIg controller
305	LXE10E100H	LXE10E – scroll compressor unit with DECOS IIIh controller
306	LXE10E100J	LXE10E – scroll compressor unit with DECOS IIIj controller
307	LX10F10A	ZESTIA - inverter compressor With DECOS V controller
308	LX10F11B	ZESTIA - inverter compressor With DECOS Va controller

CONTROLLER TYPE DEFINITIONS MAERSK CONTAINER INDUSTRY

Definitions for controller types as per Section 5.4.2 of the UDM standard:

Key	Name	Type	Description
pMK	Maker	Enum	All controller manufacturers: Daikin, Carrier, MCI, TK.
pCM	Controller Model	Enum	Same as Maker. Each Maker is assigned a pCMxx range for maintaining controller model types.
pUM	Unit Model	Enum	Same as Maker. Each Maker is assigned a pUMxx range for maintaining Unit Model.

Maker

Key	Controller Type Id	Description
P3	MCI	All refrigerated container machinery produced by Maersk Container Industry

Controller Model
Reefer maker: Maersk Container Industry

Key	Controller Model Id	Description
P301	RCCU 6	RCCU 6 Controller (CIM6)
P302	RCCU 5	RCCU 5 Controller (CIM5)

Unit Model
Reefer maker: Maersk Container Industry

Key	Controller_Unit_Model_ID	Description
	Star Cool	Star Cool refrigerated unit

DEVICE DEFINITIONS
ARROWSPOT

Definitions for devices types as per Section 5.5.4 of the UDM standard:

Maker

Key	Name	Description
700	Arrowspot	Telematics equipment produced by Arrowspot.

Device
Device maker: Arrowspot

Key	Device Model ID	Description
701	ArrowTrack	ArrowTrack 3G-433 Model: ARS-AD0001

DEVICE DEFINITIONS
BOXPLUS

Definitions for devices types as per Section 5.5.4 of the UDM standard:

Maker

Key	Name	Description
600	BoxPlus	Telematics equipment produced by BoxPlus

Device
Device maker: Boxplus

Key	Device Model ID	Description
601	IBOX BPRDC203	
602	BPRDC/204	
603	BPRDC300	

DEVICE DEFINITIONS
CARRIER

Definitions for devices types as per Section 5.5.4 of the UDM standard:

Maker

Key	Name	Description
	Carrier	All devices produced by Carrier

Device Model
Reefer maker: Carrier

Key	Device Model	Description
	TripLINK	TripLINK connectivity telematics device provided by Carrier

DEVICE DEFINITIONS
EMERSON

Definitions for devices types as per Section 5.5.4 of the UDM standard:

Maker

Key	Name	Description
100	Emerson	All telematics equipment produced by Emerson.

Device
Device maker: Emerson

Key	Controller Model ID	Description
101	RMM	ISO10368 compliant device.
102	RMM+	ISO10368 and cellular device.
103	RMM-W	Cellular device.
104	REFCON Portable Modem	Portable Bluetooth device.

DEVICE DEFINITIONS
GLOBE TRACKER

Definitions for devices types as per Section 5.5.4 of the UDM standard:

Maker

Key	Name	Description
	GlobeTracker	Gen 2.0 and 2.5 Global Cellular, LoRa, LoRaWan, BLE

Device
Device maker: Globe Tracker

Key	Device Model ID	Description
	Gen 2.0, 2.5	Reefer monitoring on Cellular, LoRa, LoraWAN and BLE
		Capable of communicating to LoRa, LoRaWAN, Cellular and BLE gateways on land and sea.

DEVICE DEFINITIONS
MAERSK CONTAINER INDUSTRY A/S

Definitions for devices types as per Section 5.5.4 of the UDM standard:

Maker

Key	Name	Description
	Maersk Container Industry	All devices produced by MCI

Device Model
Reefer maker: Maersk Container Industry

Key	Device Model	Description
	Sekstant Gateway	Sekstant Gateway provided by Maersk Container Industry

DEVICE DEFINITIONS
ORBCOMM

Definitions for devices types as per Section 5.5.4 of the UDM standard:

Maker

Key	Name	Description

Device
Device maker: Orbcomm

Key	Device Model	Description