14 DS2 Edge to Cloud Connector Module (E2C)

14.1 DS2 Edge to Cloud Connector Module (E2C)

Owner(s): SWAG
DoA Task: T6.2
Tier: T2
Nature: Optional
Result: K6.2

This task will enable edge-to-cloud connectivity through applications and devices capable of collecting, processing data and interconnecting this data with the cloud infrastructure of T6.1. It will be receiving, storing, and processing enormous amounts of data and management tools needed to extract insights and make data-driven decisions based on data. This task will implement the DLC ecosystem that ensures data security and privacy and implement appropriate measures such as encryption, access control and anonymization. All relevant adapters, interfaces and UIs are developed within this task that allow the use, maintenance, and full control of the ecosystem. This task will establish concepts of open data that permit public data availability and accessibility for use and reuse without restrictions. It will further ensure that individuals and organizations will stay in control of their data, allowing them to control their own data and decide how and when it is shared. Once the federated IDT environment is operational through T6.3 amongst other tasks, it will be necessary to monitor it continuously at execution time (aka T4.2). Based on a data quality lifecycle, tools will be developed to define and constantly monitor data quality and establish a framework for automatic checks, e.g. of loss detection.

14.1.1 Introduction

Purpose: Dataspaces allow for data to be shared between data providers and data consumers. This includes data coming from sensors and devices at a high rate. This module is used to establish a secure edge-to-cloud connectivity for data providers to cloud-based IoT platforms like Azure IoT or Cumulocity IoT or AWS IoT via a MQTT bridge. The data providers will decide which data to share and with whom. In addition, the data quality can be monitored as well.

Description: The DS2 Edge-to-Cloud Connector module (DS2 E2C) will use Software AG's open-source product thin-edge.io as background. The underlying communication relies on MQTT, a standards-based messaging protocol used for machine-to-machine communication. The data is locally collected on the edge device, mainly IoT-based but not exclusively, and published to the MQTT broker supplied by this module. Depending on the specific cloud platform used by the data provider or data consumer, appropriate rules will get applied to map the generic format to the specific cloud format. The data will constantly be monitored for loss of data quality which will be reported. Using the MQTT-bridge functionality the data and quality information gets mapped to the chosen cloud platform. The communication is encrypted via SSL/TSL using local certificates.

14.1.2 Where this component fits

14.1.2.1 Big Picture

This module solves the challenge of IoT device enablement in industrial IoT or more precisely to integrate (IoT devices) with cloud and IoT platforms in a reliable and secure way. With thin-edge.io the module provides re-usable and modular components, which are not bound to a specific IoT platform, domain model or vendor.

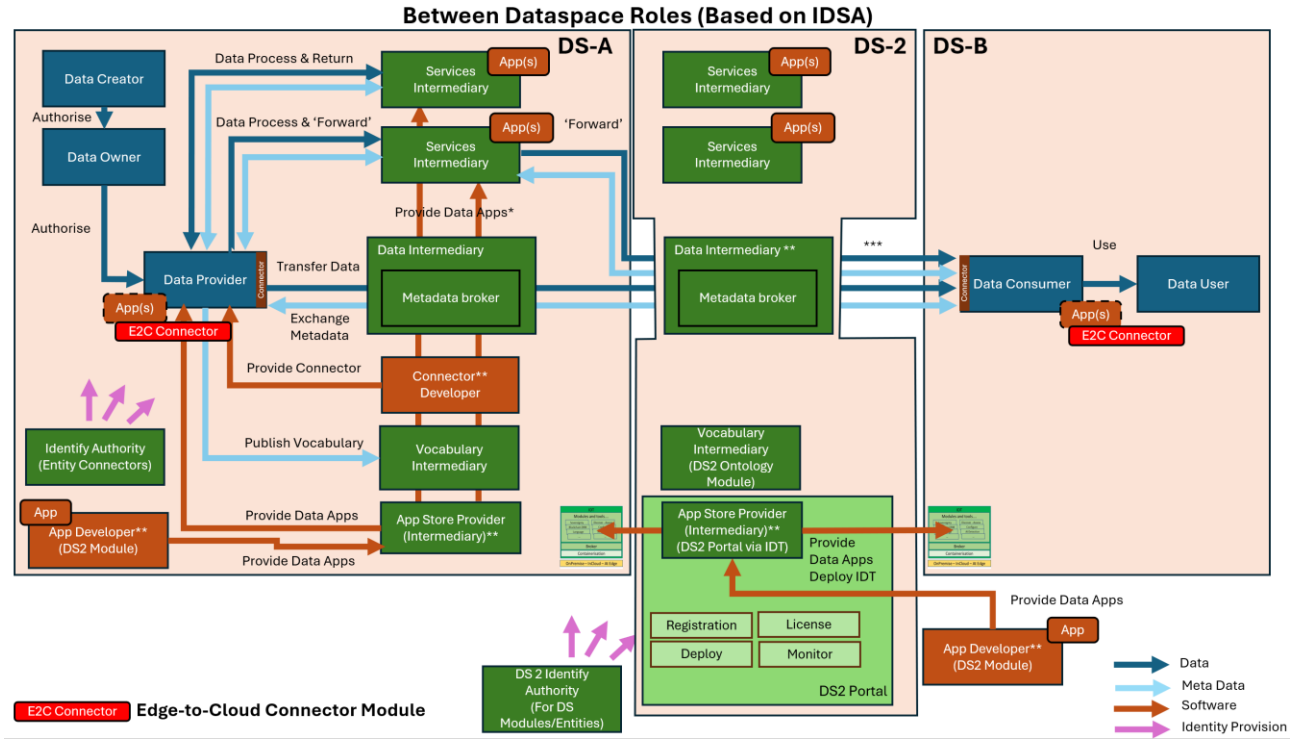


Figure 1: E2C Connector Module in the DS2 Architecture

Where	Status
Within a single Dataspace for use between participants in that Dataspace only	Yes
Deployed and used by a single participant to enable the participant in either an In-Data space or Inter- Data space scenario	Yes. The cloud instance is configurable.
Across Dataspaces without Service Intermediary	Yes. The data transfer is directly between provider and consumer.
Across Dataspace with Intermediary	No
Other Comments	N/A

14.1.2.2 Within a single Dataspace (where applicable)

The mostly likely use case is that a data provider transfers data from their own edge device(s) to their own cloud instance which can then be shared with data consumer within the same dataspace.

14.1.2.3 Deployed and used by a single participant (where applicable)

E2C connector module can also be configured so that the cloud instance of another participant, either In-Dataspace or Inter-Dataspace, is used instead. The communication is encrypted via SSL/TSL.

14.1.2.4 Across Dataspaces without Intermediary(where applicable)

The Edge-to-Cloud (E2C-Connector) is configured so that data is directly and securely transferred from the edge to the cloud, without any Service Intermediary.

14.1.2.5 Across Dataspace Dataspaces Intermediary (where applicable)

No

14.1.3 Component Definition

The figure below represents the actors, internal structure, primary sub-components, primary DS2 module interfaces, and primary other interfaces of the module. The architecture of the Edge-to-Cloud Connector Module is shown in Figure 2, data is collected via a MQTT broker and can be sent to different cloud environments such as Cumulocity, Azure or AWS via adapters.

Thin-edge.io is supporting any Linux distribution, in most cases even with the appropriate package manager. In DS2, thin-edge.io will mainly be used to connect on-premises edge devices to cloud environments of the Data Providers.

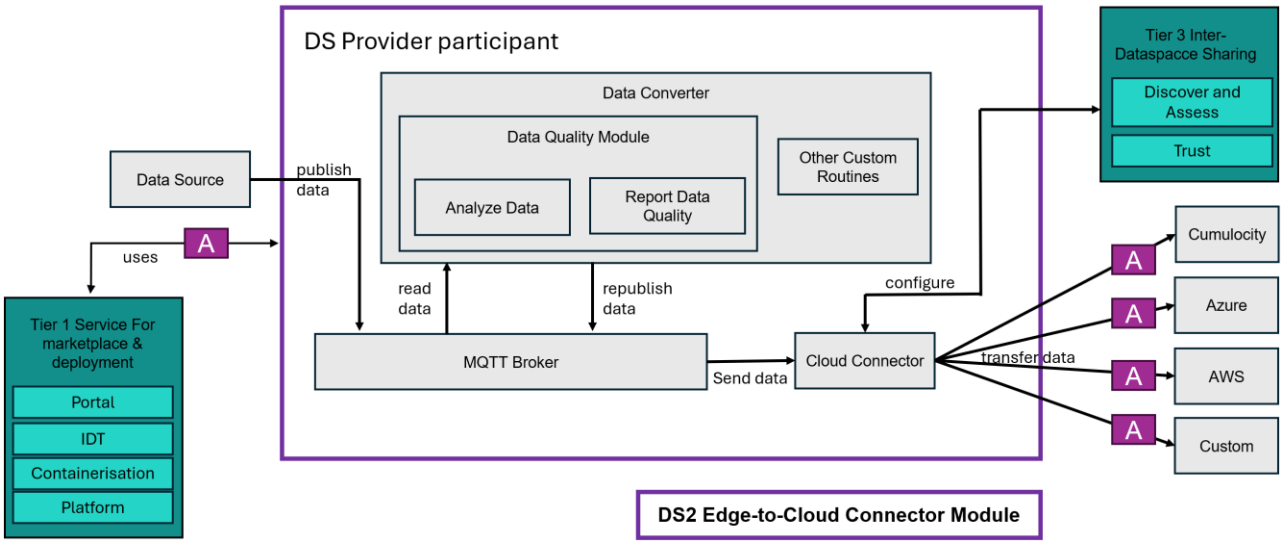


Figure 2: Schema for the Module

This module has the following subcomponent and other functions:

Edge-to-Cloud Connector Module – Core:

- **MQTT Broker:** This is the central message bus of the module, based on the open-source Mosquitto implementation.
- **Data Converter:** It reads data, which was published under a certain broker topic, and transforms it to the specific format for the configured cloud platform The data converter can use a pipeline of custom routines for specific tasks before. The transformed data is re-published to the MQTT broker under a different topic. It contains further subcomponents for:
  - **Data Quality:** The data read by the Data Converter component is constantly analysed by the:
    - **Analyze Data component** of this module to check if the data quality goals are met.
    - **Report Data Quality** component. The information is collected and sent back to the MQTT broker via the Report Data Quality component.
  - **Other Custom Routines:** Users can write their own routines to perform tasks on their data published to the MQTT broker before it is sent to the cloud platform via the cloud connector. This could be e.g. calculate moving averages, many examples are available in the thin-edge.io plugins directory.
- **Cloud Connector:** Is a message bus where data can be published under a topic and data consumer can subscribe to these topics

External Components Used:

- **Data Source:** The data provider configures and selects where the data to be offered and shared comes from, most likely an edge device
- **Tier 1 Service Stack for Marketplace and Development:** The module uses the marketplace especially for the configuration of the desired cloud adapter
- **Tier 3 Inter-Dataspace Sharing:** The module uses the Discover and Trust components for the configuration of the desired cloud adapter
- **Cumulocity, Azure, AWS, Custom:** These components represent the possible cloud adapters that can be configured. Besides connecting to the commercial clouds

Cumulocity, Azure and AWS, one can write a custom adapter for any IoT platform having a MQTT connector, e.g. the open-source ThingsBoards platform.

14.1.4 Technical Foundations and Background

thin-edge.io, the basis of the E2C Connector Module, is an open-source development toolbox designed for rapid development of IoT agents. It is based on a versatile set of ready-to-use software components that can be easily combined with application-specific extensions into smart, secure, robust, and efficient IoT agents which integrate cloud services, edge computing and operational technologies. A typical agent uses as a foundation the building blocks provided by thin-edge.io for telemetry data processing and cloud connectivity as well as for device monitoring, configuration and updates. In combination with these blocks, the agent designer can provide application-specific extensions, which cooperate with thin-edge.io over MQTT and HTTP along a JSON API, to address any hardware, protocol or use-case specificity. As such, thin-edge.io is good choice for implementing smart-equipment that collect in-situ real-time data, perform analytics on the edge, forward key data to the cloud, and need to be secured, configured and updated at scale.

Subcomponent/Component	Owner	License
thin-edge.io	SWAG	Open Source

14.1.5 Interaction of the Component

The following table specifies the primary input/output controls/data to blocks, which are not part of the module

With Module/Feature	Receives From/Gives To	What
Tier 3: Trust	Receives From	Participant information for the cloud platform to connect to
Tier 3: Discover	Receives From	Configuration information for the cloud platform to connect to
Data Source	Receives From	Data and information to be published to the cloud platform
Cumulocity	Gives To	Data in the correct cloud format, possibly enriched, corrected and transformed
Azure	Gives To	Data in the correct cloud format, possibly enriched, corrected and transformed
AWS	Gives To	Data in the correct cloud format, possibly enriched, corrected and transformed
Other	Gives To	Data in the correct cloud format, possibly enriched, corrected and transformed

14.1.6 Technical Risks

Risk	Description	Contingency Plan
Connection issue	The connection to the cloud environments might be unstable.	Monitor the connection and raise an alarm if lost
Unknown Data Type or data types	Data type mismatch, so it comes to a discrepancy between what is expected and what is sent	Work with generic data (JSON) and monitor the data schema

which do not fit		

14.1.7 Security

Security Issue	Description	Need
N/A	No special security issues are anticipated	

14.1.8 Data Governance

Data Governance Issue	Description	Need
Need data examples.	GDPR rules need to be obeyed.	A mechanism is needed so approved (or open) data samples can be shown

14.1.9 Requirements and Functionality

This module will be used in the following usecases:

City Scape ✓

Green Deal ✓

Agriculture ✓

Inter-Sector TBD

Their requirements and functions/extensions to achieve them relative to this module, specifically extracted from the use case are as per the table below noting that in many cases further discussion might need to take placed between pilot partner and module partner to determine if a fit or the scope of the precise fit.

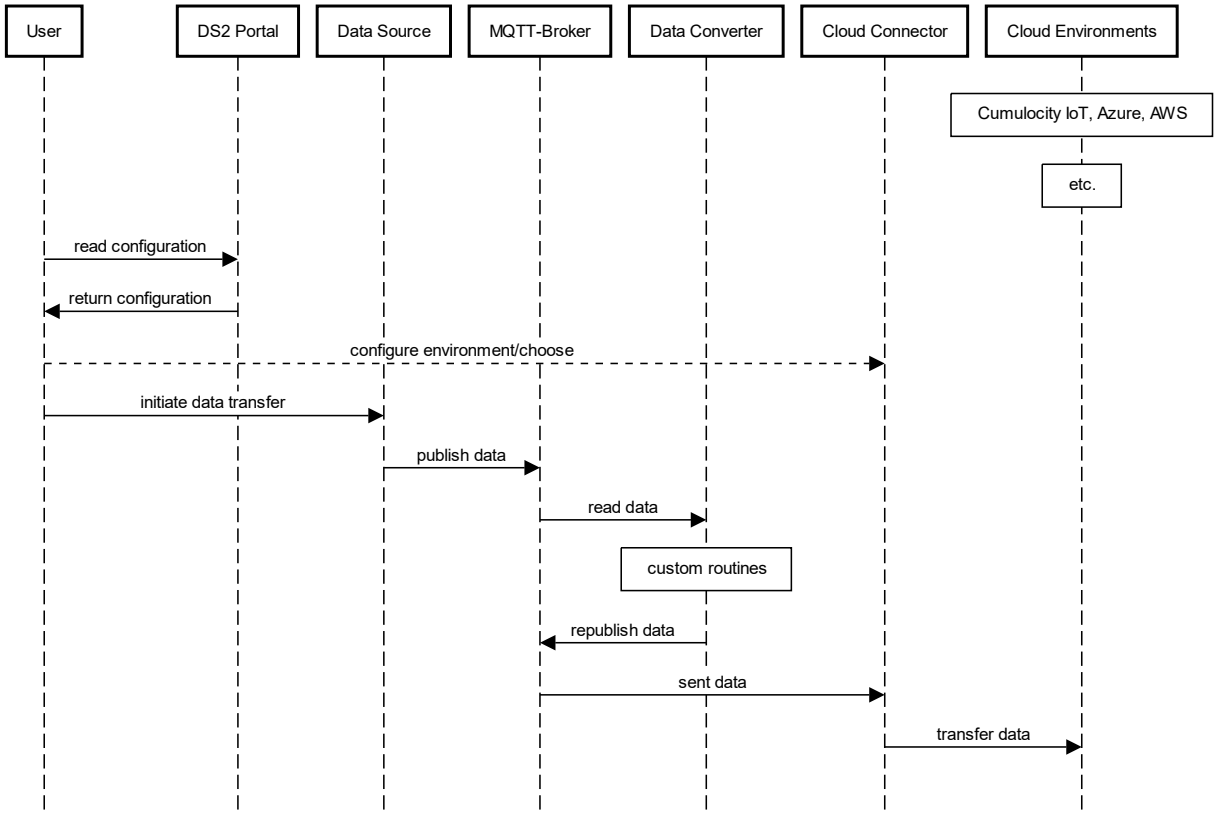
WHERE	WHAT	WHY	Run/Design Time	Priority
	Use Case 1: City Scape			
Section 1.1.7(All Scenarios)	N/A		R & D	C
	Use Case 2: Green Deal			
Section 1.1.7(All Scenarios)			R & D	C
Section 1.1.5(All Scenarios)	*SENSEDGE IOT (Farmers)*	Measurement from Agricultural IoT devices (weather station, soil sensor) which are setup	R & D	C

		at different farmers in the region. *Need to understand, whether we need to integrate with the sensors itself or get API*		
Annex 1 (All Scenarios)	Title: Sensedge IoT no additional information compared to section 1.1.5	Measurement from Agricultural IoT devices (weather station, soil sensor) which are setup at different farmers in the region. (looks identical to 1.1.5) Need to clarify if to integrate with the sensors? (which sensors) or should use an API	R & D	C
	Use Case 3: Agriculture			
Section 2.1 (UC3.3)	field-based camera nodes at the edge originating from DigiAgro DS.	To be further clarified	R & D	C
Section 1.1.4 (All Scenarios)	Data Integration (edge devices and sensors) we need to understand what edge devices are and what sensors or do we get API to integrate with the sensors?	To be further clarified	R & D	C

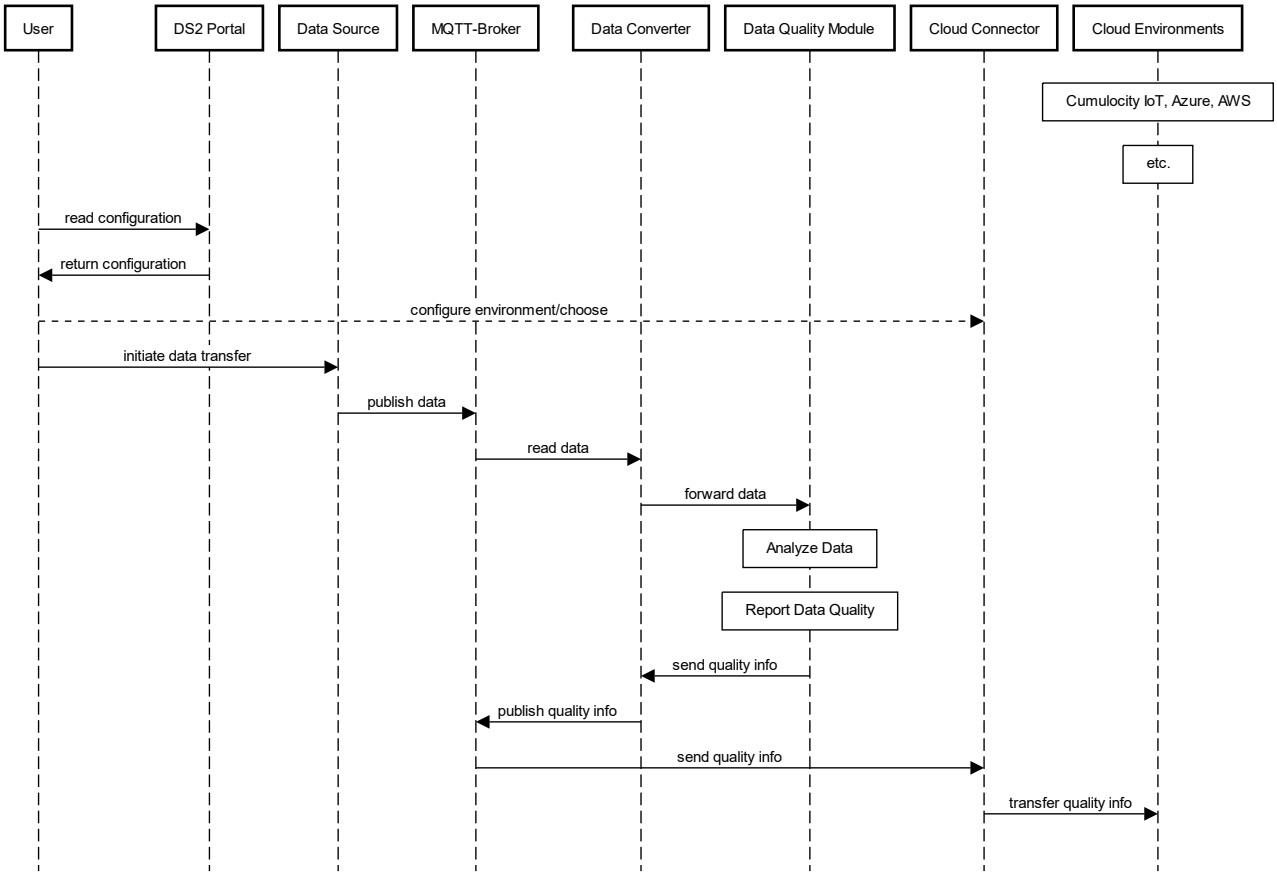
14.1.10 Workflows

The following sub-sections describe the sequence diagrams of the Module

14.1.10.1 Read/Write Data in the cloud from the edge and vice versa



14.1.10.2 Analyse and report Data Quality


User
DS2 Portal
Data Source
MQTT-Broker
Data Converter
Data Quality Module
Cloud Connector
Cloud Environments
Cumulocity IoT, Azure, AWS
etc.
read configuration
return configuration
configure environment/choose
initiate data transfer
publish data
read data
forward data
Analyze Data
Report Data Quality
send quality info
publish quality info
send quality info
transfer quality info

14.1.11 Role, Resourcing, and Milestones

Sub-component	Main Activity	M18	M24	M30	M36
Edge-to-Cloud Connector	Basic Configuration and Deployment				
Edge-to-Cloud Connector	Connect Sample Data Source and Cloud Platform				
Edge-to-Cloud Connector	Write module configuration to DS2 portal				
Data Quality Module	Write module code and integrate the module in the Data Converter				
Edge-to-Cloud Connector	Write custom routines for different scenarios				
Edge-to-Cloud Connector	Support to write custom routines for pilots				
Edge-to-Cloud Connector	Configure Cloud connector for in-dataspace platforms of data consumer				
Edge-to-Cloud Connector	Configure Cloud connector for inter-dataspace platforms of data consumer				
Data Quality Module	Adapt data quality detection to the pilots				
Edge-to-Cloud Connector	Final integration of pilots				
Data Quality Module	Final integration and testing of module				
Edge-to-Cloud Connector	Documentation and final testing				
DS2 componentization, final integration	final integration				
Table Total/DOA Task Total/Resilience	**Comments:**				

14.1.12 Open Issues

The following table summarise open issues/uncertainties that need to be resolved during the next stages or implementation.

Issue	Description	Next Steps	Lead or Related Component
Possible overlap with DIGI's T6.1?	Result of the discussion with DIGI. There is no overlap, as the data in the multi cloud project will be taken from data bases and shares the data with the cloud. The quality of the data is assured, but Software AG gets the data from sensors, which requires a quality check of the data.	Revised text to clarify this	DIGI, SWAG