PROGRESS AND ON WEB

DELIVERABLE D6.1 - FEDERATED IDT PLATFORM - PHASE I

PROJECT ACRONYM:	DS2
PROJECT TITLE:	DataSpace, DataShare 2.0
GA NUMBER NO.	101135967
WEBSITE:	www.dataspace2.eu
DUE DATE OF DELIVERABLE:	2025-06-30
SUBMISSION DATE:	2025-06-27
LEAD BENEFICIARY:	ICE
LEAD AUTHORS:	Stuart Campbell (ICE)(Editor). Module/Additional activities content: Elias Dakos (ATC), Noel Tomas (ICE), Victor Anaya (ICE), Carlos Sanchez (INDRA), Soumya Kanti Datta (DIGI), Jens Schimmelpfennig (SWAG)
REVIEWERS:	Eliot Salant (IBM), Ilias Syrigos (UTH)
TYPE:	OTHER
DISSEMINATION LEVEL:	PUBLIC

DISCLAIMER

STATEMENT OF ORIGINALITY

This Deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation, or both.

DOCUMENT HISTORY

VERSION	DATE	DESCRIPTION	NAME	ORG
V0.1	2025-05-19	Table of contents	S.Campbell	ICE
V0.2	2025-05-26	WPL Sections Initial	S.Campbell	ICE
V0.3-0.5	2025-05-29	Module Added	Elias Dakos, Noel Tomas, Victor Anaya, Carlos Sanchez, Soumya Kanti Datta, Jens Schimmelpfennig	Various
V0.6-0.6	2025-05-29	WPL Final sections and editing	S.Campbell	ICE
V0.8	2025-06-07	Comments by Reviewer 1 Processed	S.Campbell	ICE
V0.9	2025-06-13	Comments by Reviewer 2 Processed	S.Campbell	ICE
V1.0	2025-06-20	Final Document	S.Campbell	ICE

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This deliverable covers the WP6 activity until M18 and gives an overview of the first prototype of the different WP6 solutions or applications and current status as well as partner activity. These solutions or applications are related to the tasks T6.1, T6.2, T6.3 and T6.4 of the WP and the objectives:

- O6: To establish a federated, interconnected design and runtime ecosystem through an Intersector DataSpace Toolkit (IDT) supporting the IDT Broker, WP3-5 modules/methodology, containerisation, and quality control:
    - O6.1: Multicloud sharing environment, on a basis of infrastructure and containerisation as well as providing a data-as-a-service space for experimentation and piloting
    - O6.2: Edge to Cloud Data Enablement, Control, and Data Quality Detection for the design and runtime access of data and modules to allow control, security as well as execution time data content monitoring
    - O6.3: Producing the IDT Broker which enables the Federated Environment for Intersector data cooperation
    - O6.4: Delivering IDT Integrated Environment which packages all modules and systems of WP3-6 in a holistic containerised system deployable InCloud/OnPremise and provide out-of-the box connection to other IDTs

The formal WP deliverables are all software (Type: OTHER) which are not uploadable to EU Sygma, so the software, possibly source code, documentation, detailed progress tracking are PRIMARILY on the DS2 GitHub which will evolve and grow in content over time – see https://ds2-eu.github.io/documentation/. The online documentation includes module overview, architecture, components descriptions, example screen shots, information on installation and use and other matters. Within this document, which accompanies the formal software/information on the DS2 GitHub, an introduction to the WP is provided, followed by a brief description of the different modules developed. Then, there is a summary of the activity and progress for each module including links to the DS2 repository. In addition to this, some WPs have other non-module and/or non-software activities which are also reported where applicable. This is then followed by a description of KPI's, Primary Risks, and Primary Issues for the WP, followed by a conclusion section.

There are few significant risks and issues, and all software is on track or exceeding expectations.

1 INTRODUCTION

The WP6 suite of software provides the core integration and interoperability features for the other modules of WP6, as well as those in WP3, 4, and 5.

This includes modules for the below which are documented in Section 3:

- DS2 Portal PORTAL
- DS2 IDT Broker IDT
- DS2 Containerisation Module CONT
- DS2 Identity Module IDM
- DS2 Data Marketplace Module DMK
- DS2 MultiCloud Module MCL
- DS2 Edge to Cloud Module E2C
- DS2 Data Inspection Module DINS
- DS2 Data Share Controller DSHARE
- DS2 Security Module SEC
- Policy Creation Module PCR
- Data Visualisation Module DVM

In addition, additional activities are also documented in Section 4:

- Experimentation Facility [EXP]
- DS2 Dash Button [DASH]
- DS2 Platform [PLATFORM]
- DS2 Connector [DS2CON]

Some of the narrative texts in section 3 refer to other modules in this WP, or other WPs, so corresponding documents in other WPs may need to be referred to – namely, D3.2, D4.1, D5.1, D6.1. Similarly, modules may refer to additional activities, particularly [DASH, PLATFORM, and DS2CON] which appear later in section 4 of D6.1.

The technical partners involved are ICE, i4RI, INDRA, ATC, SWAG, DIGI, VTT. INTU also had resources in the WP, but these have been consolidated in WP5 since the tasks were overlapping. All DS2 users have minor resources in the WP and participated intermittently/as necessary in meetings and/or email exchanges.

The formal WP deliverables are all software (Type: OTHER), so the software, possibly source code, documentation, and detailed progress tracking are primarily on the DS2 GitHub which will evolve and grow in content over time - https://ds2-eu.github.io/documentation/.

1.1 Document Structure

- Section 1: Introduction: Introduces the deliverable – this section
- Section 2: Module Description: High-level reminder of modules connected with the work package
- Section 3: Software progress: Module, by module, progress
- Section 4: Additional Activities: Progress of any non-software activities
- Section 5: KPI, Risks, and Primary Issues: Main factors affecting progress and Status
- Section 6: Conclusion: Conclusion and next steps
- Annex A: DS2 Online Documentation and Development Process/Tracking

1.2 Glossary and Abbreviations

A definition of common terms related to DS2 as well as a list of abbreviations, is available at https://www.dataspace2.eu/results/glossary

1.3 External Annexes and Supporting Documents

External Documents:

- DS2 D2.2 Requirements, baselines, KPIs, Architecture & Specifications
- DS2 Communication, Dissemination, and Exploitation report (M18)
- DS2 Risk identification spreadsheet

2 MODULE DESCRIPTION AND OVERALL STATUS

2.1 Modules

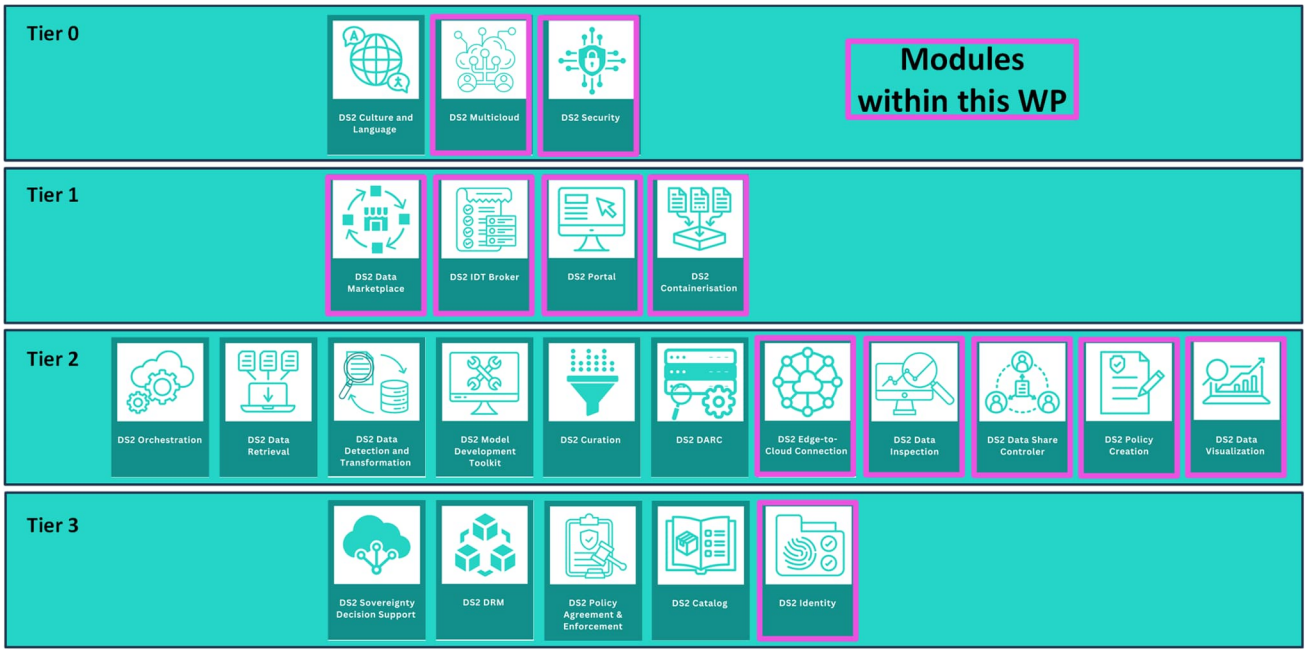
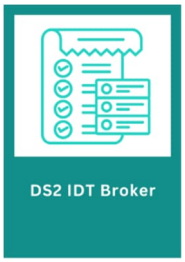
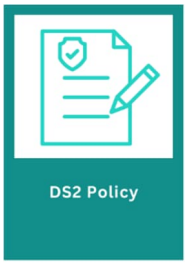


Figure 1. WP6 Modules and Tiers

This WP has the following modules:

Module	Purpose
Tier 0	
SEC 	The DS2 E2C security Module (SEC) covers data security, data protection, and privacy with a focus on securing the edge-to-cloud data enablement and ensuring data quality and privacy. This involves implementing secure communication protocols, robust authentication mechanisms, encryption, anonymisation, and continuous monitoring of events in the DS2 ecosystem. The DS2 architecture is designed to manage large volumes of data, facilitating data-driven decision-making whilst maintaining stringent data security and data privacy standards. The proposed component works in conjunction with all DS2 architecture components.
Tier 1	
DMK 	The Data Marketplace Module (DMK) will provide a marketplace for data and data models. It will allow the registration of data from a catalog, record all transactions, and communicate transactions to any external system if required (e.g. Data Rights Management Module, Clearing House). Data will not be stored in the Data Marketplace Module. It will provide the capability to support not only datasets but also algorithms.
IDT 	The Intersector Dataspace Toolkit (IDT) is the core enabler of DS2 whose purpose is to be deployed in front of participants' data source/spaces and network connected to any other IDT-enabled data source. As such its aim is to run all DS2 Modules, including the DS2 Connector, the core Module for Inter-Dataspace communication and data transfer, and the Containerisation Module for DS2 Module deployment. IDT contains the core Kubernetes runtime to run all containerised Modules and a series of additional open-source software for Module management.
PORTAL 	To provide a user and developer friendly portal allowing dataspace participants to register and select DS2 Modules which can then be packaged into an IDT environment and then subsequently deployed by participants enabling both In-Data Space and Inter-Data Space operations. As such, it includes functionality for developers to include Modules, users to find those Modules, to trigger the packaging through links with the containerisation Module, as well as supporting functionality for dataspace support, dataspace resources, registration and identity management, and administration. It also provides support for the Data Marketplace.
CONT 	To allow easy and automated packaging and deployment of Modules on the IDT Kubernetes runtime subcomponent environment. The containerisation Module leverages on custom Helm Chart descriptors to automatically convert them into full Kubernetes Helm Charts representing the Module, based on standard base templates located in the DS2 Portal Marketplace. The Helm Charts are then deployed on the IDT Module.

Tier 2	
E2C 	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.
DINS 	The Data Inspector Module (DINS) facilitates the configuration and deployment of processes for real-time data analysis, ensuring data quality and compliance with thresholds set by the parties involved. It performs several key functions: generating notifications based on the values of the exchanged data, executing reactions such as sending requests and notifications to external tools, and integrating with models developed to enhance its capabilities. It is a complement to the Data Share Controller which focuses on control information with both Modules using the Data Interceptor.
DHARE 	To provide a user-oriented view of control plane information related to a specific exchange of data, to monitor its status and to potentially limit or block it. It will access data through a Data Interceptor component which it shares with the DS2 Data Inspection component (DINS) which operates more at the data level. It can be seen as an In-Dataspace enablement Module. Its role is especially important in an Inter-DS environment to provide extra monitoring and control of the data exchanges when partners are less known.
PCR 	The DS2 Policy Creation Module (PCR) serves as an essential tool for Data Space authorities, facilitating the creation and comparison of dataspace policies through an intuitive user interface (UI), which can then be matched against other dataspace policies. This module leverages the Open Digital Rights Language (ODRL) to offer a sophisticated yet simplified approach to policy generation.
DVM 	The Data Visualisation Module (DVM) provides users with configurable dashboard functionalities to combine, visualise and analyse data from various sources. It can be used by data creators, data owners, data providers and data consumers to better understand and communicate raw data and to showcase the results of other modules (such as DDT).

Tier 3	
IDM 	The DS2 Identity Module (IDM) is a 'system' since it relies on a central DS2 Identity Module deployment, IDTs which are pre-configured to communicate with this Module, and access to/from individual dataspace Identity Providers. It aims to provide a practical framework for the creation and validation of participant identities for inter-Dataspace activities based on the existence of current dataspaces and their own individually selected Identity authorities. This is linked to the IDT Connector and allows for a federated approach of the connectors whilst relieving participants from connector interoperability, outside dataspace change and maintenance issues, and minimising or eliminating the changes to their existing environment.

The module fit is as follows:

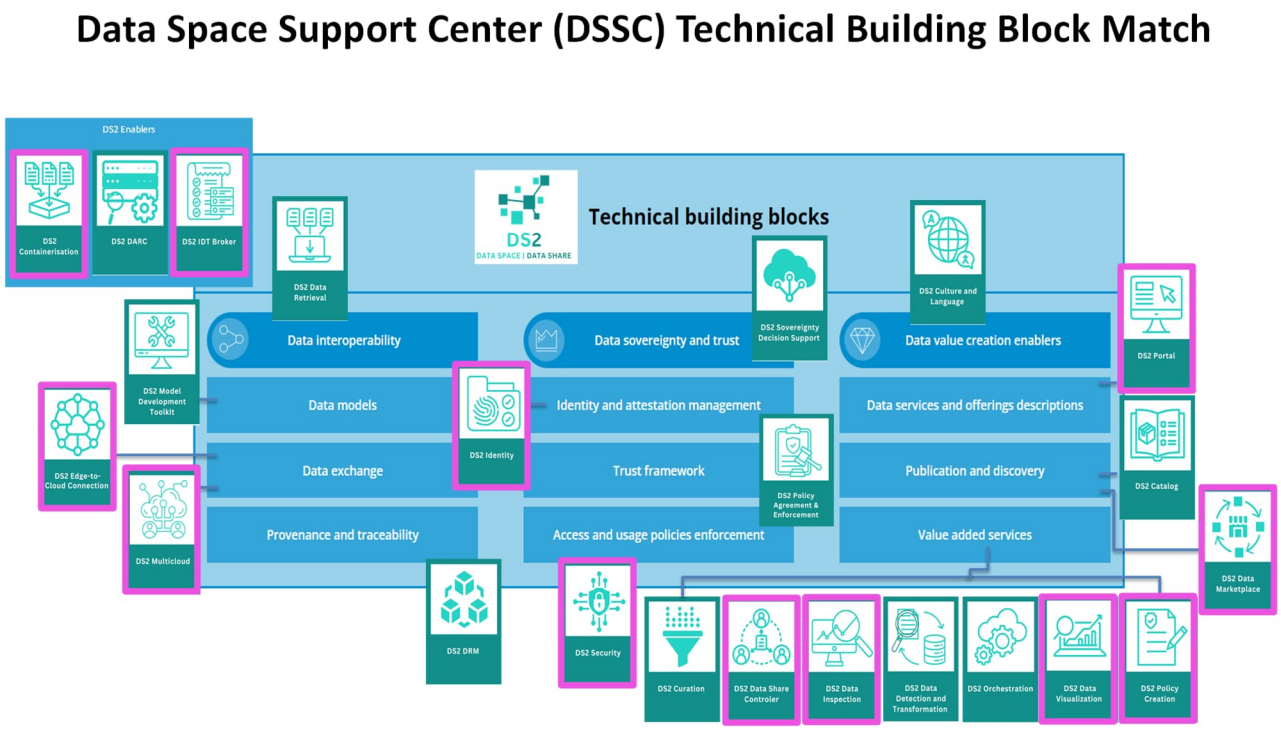


Figure 2. Module Fit

2.2 Status

In terms of the percentage tracking / status columns, these are a high-level best-estimates given by the developers based on the detailed per-module tracking activity for features express in Section 3 and according to the methodology identified in Annex A of the D6.1 deliverable.

Module	Task	Partners	License Type	Software Status @ M18	Estimated Completion Month	How to Install "vs 0.1" at M18	How to Configure "vs 0.1"	Marketing Video	PORTAL Helm Chart	Marketplace Entry
ORC	T4.1	ICE	Opensource, Apache 2.0	40%	M24: 70% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
PORTAL and [MARKETPLACE]	T6.4	ICE	Opensource, Apache 2.0	80% PORTAL 60% [Marketplace]%	M24: 65% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
IDT	T6.3	I4RI/ICE	Opensource, Apache 2.0	50%	M24: 75% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
CONT	T6.4	ICE	Opensource, Apache 2.0	50%	M24: 80% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30

IDM	T6.4	ICE	Opensource, Apache 3.0	40%	M24: 70% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
DMK	T6.1	INDRA	Opensource, Apache 2.0	10%	M24: 50% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
MCL	T6.1	DIGI	Opensource, Apache 2.0	40%	M24: 75%, sufficient for 1st validation M30: 100%	On Github	By education session	M30	M24	M30
E2C	T6.2	SWAG	Opensource, Apache 2.0	40%	M24: 65% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
DINS	T6.2	INDRA	Opensource, Apache 2.0	60%	M24: 90% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
DSHARE	T6.2	i4RI	Opensource, Apache 2.0	75%	M24: 85% - Sufficient for 1st Validation M30: 100%	On Github	By education session	M30	M24	M30
SEC	T6.2	DIGI	Opensource, Apache 2.0	80%	M24: 100%	On Github	By education session	M30	M24	M30
PCR	T6.3	ATC	Opensource, TBD, if Apache 2.0	0% (due to a late start – additional module not in the DOA)	M24: 80% - Sufficient for 1st Validation M30: 100%	Not available due to late addition to DS2 software blocks	By education session	M30	M24	M30
DVM	T6.2	SWAG	Opensource, Apache 2.0	0% (new module, pending Amendment and EU approval of transition from SWAG to BBS)	M24: 75% - Sufficient for 1st Validation M30: 100%	Pending amendment	By education session	M30	M24	M30

2.3 Major Changes/Deviations impacting this WP

- Partner change SWAG and Blue Bridge. As per the periodic report, SWAG will leave the project, and its activity will be taken over Bluebridge solutions UG (pending Amendment and EU approval). Since the same personnel will be involved, and the IPR is open source, no issues are foreseen.
- New Module PCR – Policy Creation Module. As the DS2 nature and knowledge evolved, around 2025-03 it became clear that this additional module could benefit the project and ATC had resources/volunteered to do this. This module is documented but development will not start until M19

- New Module DVM – Data Visualisation module. Several use cases and some modules had needs for data visualization, but this was also not part of the original proposal. bluebridgesolutions UG noted they could spin-off activity from E2C and create an independent module to add to the DS2 module stack. This discussion took place in 2025-05 and architecting work has just commenced
- INTU resources consolidated in WP5. INTU also had resources in the WP6, but these have been consolidated in WP5 since the tasks were overlapping so from a management perspective, and bearing in mind WP6 is overloaded with modules, this was seen as the most appropriate approach

3 SOFTWARE PROGRESS

The software, documentation, and progress for the modules developed is located at the DS2 GitHub repository accessible by the links below. These links give the current module documentation and at the top further links to the software and module progress. By plan, at this interim state of the project, the modules are still under development, and documentation is limited to essential information. The framework for documentation and progress monitoring is identified in Annex A of this WP6 deliverable "DS2 D6.1 - FEDERATED IDT PLATFORM - PHASE I". Progress is monitored through two-weekly sprints with detailed highlights. In terms of the documentation, it will be improved with How-Tos, API definitions, etc. overtime.

Module	Link
PORTAL	https://ds2-eu.github.io/documentation/modules/PORTAL/
IDT	https://ds2-eu.github.io/documentation/modules/IDT/
CONT	https://ds2-eu.github.io/documentation/modules/CONT/
IDM	https://ds2-eu.github.io/documentation/modules/IDM/
DMK	https://ds2-eu.github.io/documentation/modules/DMK/
MCL	https://ds2-eu.github.io/documentation/modules/MCL/
E2C	https://ds2-eu.github.io/documentation/modules/E2C/
DINS	https://ds2-eu.github.io/documentation/modules/DINS/
DSHARE	https://ds2-eu.github.io/documentation/modules/DSHARE/
SEC	https://ds2-eu.github.io/documentation/modules/SEC/
PCR	https://ds2-eu.github.io/documentation/modules/PCR/
DVM	https://ds2-eu.github.io/documentation/modules/DVM/

The rest of this major section identifies for each module:

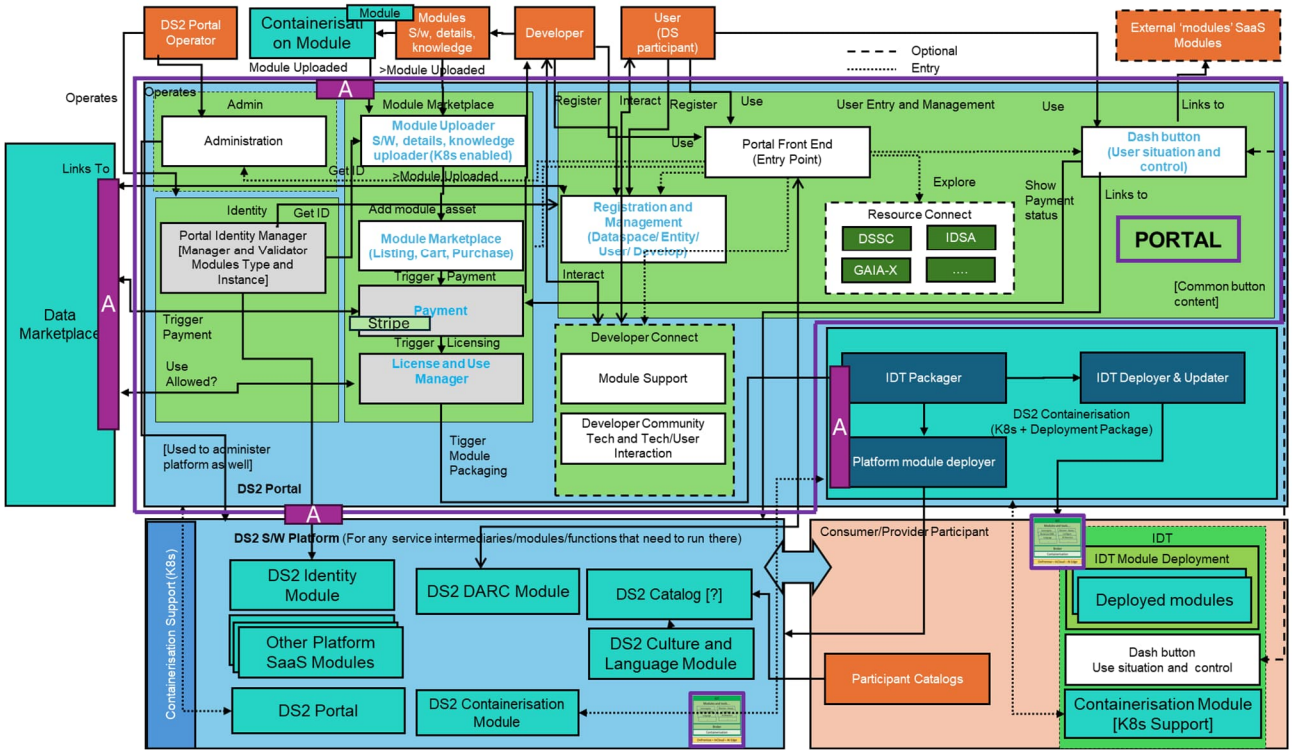
- A reminder of the architectural diagram
- Summary table of primary feature progress based on the functionalities and status of completion at M18 and which were defined for each module in the D2.2 Architecture annexes. Progress was measured via bi-weekly highlight/sprints/meetings. Highlight 1 represented 2025-02-04 and highlight 9 is the last two weeks of June 2025. Post review, the feature lists will be updated/enhanced. Also indicated in the table (on the right) is an expectation of when each feature will be completed (100%) at milestones M18, 24, 30, and 36 noting that to ensure a suitable validation plan all software should be completed by M30. Since at M18 some feature implementations will still be in-progress, then the percentages may be less than 100%. Details and highlight narratives are at the above links
- Activity progress of the partner(s) involved in terms of features implemented noting that feature descriptions are at the above links
- A use case validation plan, with an example of the PORTAL below. "Default Install" (BLUE) means that the module is either used or installed by IDT by default, whereas "Specific to Case" (GREEN) implies its use is specific to a particular use case, for example, UC1.2 – these cases were described in D2.2 and

D7.1. A BLUE with no usecase mention implies that the module will be needed/used whereas if a case is listed it means it will be needed/used AND that formal validation will take place in that pilot

	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
PORTAL		UC2.2	UC1.1, UC1.2

3.1 Module: PORTAL

3.1.1 Architecture Diagram



3.1.2 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	- -	M18	M24	M30	M36
Portal Front End	50	65	70	70	75	80	90	95	98	100		50	75	100	100
Registration and Management	20	30	40	60	60	60	60	60	60	60		0	75	100	100
Dash Button	75	75	75	80	90	95	95	98	100	100		0	0	100	100
Resource Connect	0	0	0	0	0	0	0	0	0	0		0	0	100	100
Developer Connect	0	0	0	0	0	0	0	0	0	0		0	0	100	100
Module Uploader	50	60	60	60	60	60	60	60	60	60		0	50	100	100
Payment	0	10	10	15	15	15	15	20	20	20		0	0	100	100
License and Use Manager	0	25	30	35	40	40	40	40	40	40		0	0	100	100
Administration	0	0	0	0	0	0	0	0	0	0		0	0	50	100
Bug Fixing and followup	0	0	0	0	0	0	0	0	0	0		0	0	50	100

3.1.3 Activity

The main activity in the first months of development has been related to define the initial set of documents for the software, mainly the architecture document and requirements: deliverable 2.2. Development was also started, providing an initial version for the DS2 Central Portal where users can register organisations including integration with identity management system.

The Portal has been split into two Portals: The DS2 Global one, and a lighter version of it, which will be used for the Local IDT Portal deployed by participants.

The Global Portal is the main one where organisations can be registered and the main work was focused on several enhancements for DS2 of existing Background, such as the user profile and organisation and personal information, existing organisation validation. User management was added to allow creating additional users to the organisation. The main dashboard page was improved with an on-boarding tour so when first logging in to the Portal the user receives some guided instructions. Behind the scenes, a new way of selecting which modules are displayed on the Local and Global Portal was also added in the Portal API.

As part of the Portal, a first version of the Marketplace where module owners can publish their modules and dataspace participants can acquire them has been provided. The Marketplace was fully customized to meet the DS2 style guide and integrated with the DS2 Dash Button ([DASH]). Additions were implemented

such as support for new types of applications (modules), and several UI improvements. Then, a new integration with Stripe payment system was started.

Regarding the Dash Button, significant effort has been spent with an initial version already released for integration with all the modules. It has been developed as a web component for a) Integration to the single sign on functionality across all modules and b) Holistic navigation between all modules. Work has also been performed on how to link it to the organisation profile and ways of displaying the available links from modules and core modules.

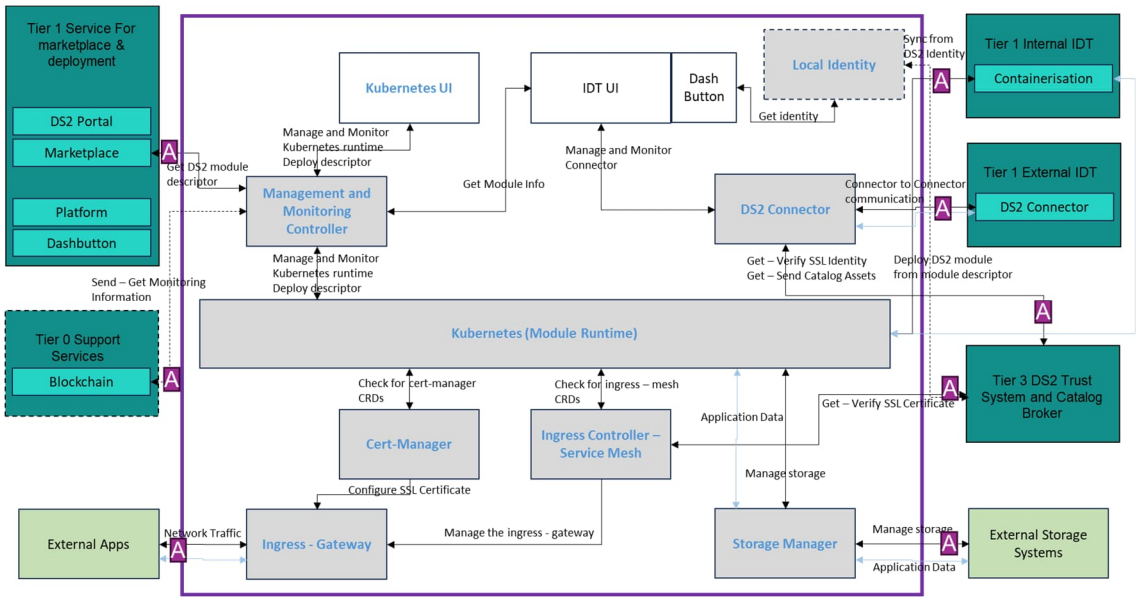
The next steps for the Portal are to continue improving the UI and adding new integrations such as the CLM for extra organisation metadata. Also, seamless integration with the IDM module for Dataspace registration is planned. Regarding the Marketplace, work has taken place to support the new module acquisition process - especially the IDT orientated installation.

3.1.4 Use Case Validation Plan

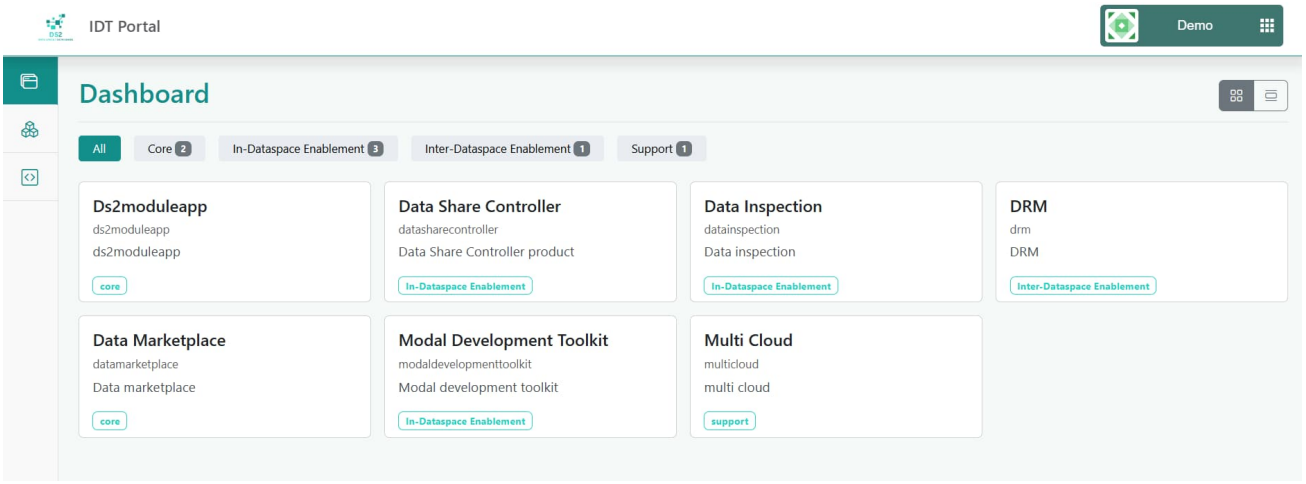
	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
PORTAL		UC2.2	UC1.1, UC1.2

3.2 Module: IDT - IDT Broker

3.2.1 Architecture Diagram



3.2.2 Sample Interface



3.2.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	- -	M18	M24	M30	M36
Kubernetes (Module Runtime)	50	50	50	60	60	60	70	90	100	100		100	100	100	100
Kubernetes UI	50	50	50	60	70	70	70	80	90	100		100	100	100	100
Management and Monitoring Controller	50	50	50	60	60	60	60	70	80	80		100	100	100	100
Ingress Controller - Service Mesh	50	50	50	50	50	50	60	90	100	100		100	100	100	100
Cert Manager	90	90	90	90	90	90	90	100	100	100		100	100	100	100
Ingress - Gateway	90	90	90	90	90	90	90	100	100	100		100	100	100	100
Storage Manager	25	25	25	25	25	25	25	25	25	25		100	100	100	100
DS2 Connector	10	20	20	20	20	20	20	20	20	20		0	0	100	100
Dash Button and Local Identity	50	75	85	90	90	90	90	90	95	100		0	100	100	100
Tier 0: Blockchain and Blockchain API	0	0	0	0	0	0	0	0	0	0		0	0	100	100
Tier 3: Integration with identity system and catalog	0	0	0	0	0	0	0	0	5	10		0	0	100	100

3.2.4 Activity

The main activity in the first months of development was to define the initial set of documents for the software, mainly the architecture and requirements deliverable D2.2. RTD activity also started, first researching on Dataspace connectors, and particularly the EDC Connector to add it to the IDT module.

Another significant activity was to release the initial version of the core Kubernetes-based platform of the IDT with some of the main components, and a script to deploy it, updating versions from previous Background. Several GUIs are being developed for the IDT: IDT Local Portal (which is an extension of the DS2 Portal) and the Connector UI (which is a user interface to interact with the IDT DS2 Connector).

The initial version of the IDT Local Portal was developed to include several enhancements from the Global Portal such as the Grid and List display and initial integration with the Marketplace to display the list of acquired modules. The Dash Button has already been fully integrated already in the Local Portal and connected to the Global Portal identity system.

A test EDC Connector UI was developed to interact with the EDC Connector, based on the existing EDC Data Dashboard which in turn allows the setting of, for example, EDC parameters and Catalog meta data entry. This Data Dashboard was updated to support the latest EDC Connector version and the different APIs. The UI quickly evolved to be the Connector interface in the IDT that supports creation of Data Assets, Policies, Contract Definitions, etc. The EDC Connector UI has also been fully integrated with the Dash Button. As part of the Connector UI work, an initial draft version of the EDC Connector has been integrated.

In addition, some tests were performed regarding using Rancher, a tool included in the IDT to manage the Kubernetes cluster, to be the Containerisation UI for module installation.

Work has been initiated lately to develop a more user-friendly IDT installer, since now it is a bash script, like a Windows installer.

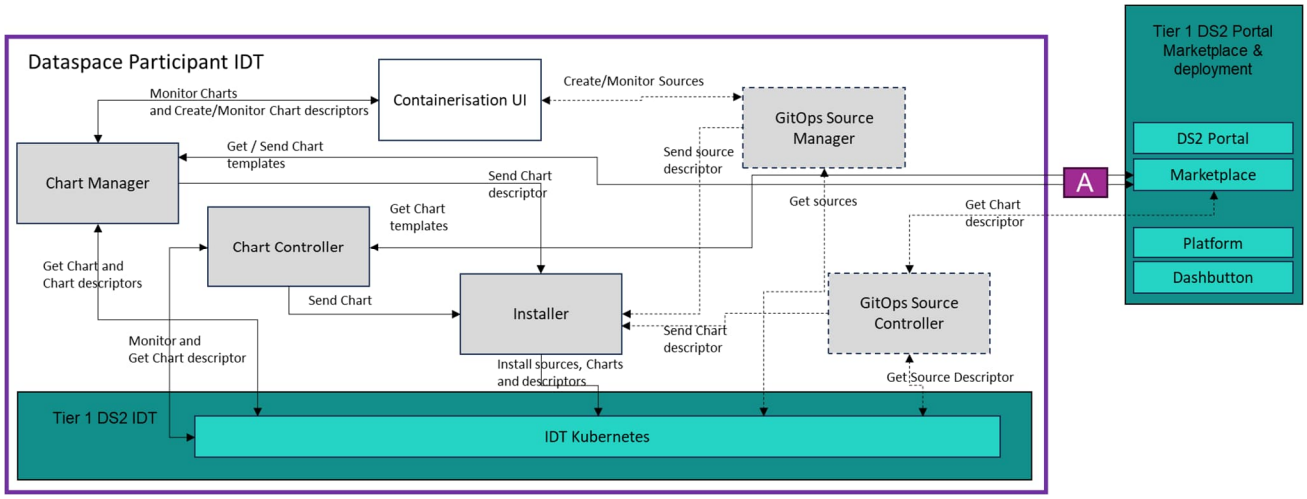
The immediate next steps are to complete the IDT core platform and continue evolving and enhancing the Local Portal and the EDC Connector. Then to complete the new installer and start integrating the Containerisation module in the installation phase. In addition, it is planned to start working towards the Connector development and integration with the IDM and CAT modules.

3.2.5 Use Case Validation Plan

	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
IDT		**UC2.2**	**UC1.1, UC1.2**

3.3 Module: CONT - Containerisation Module

3.3.1 Architecture Diagram



3.3.2 Sample Interface

The Containerisation UI is planned for M24; hence no UI is available now.

3.3.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	- -	M18	M24	M30	M36
Chart Controller	25	30	40	60	80	80	90	90	95	100		100	100	100	100
Chart Manager	25	25	25	25	40	40	40	50	60	60		100	100	100	100
Installer	50	50	60	75	80	80	80	80	90	100		100	100	100	100
Containerisation UI	0	0	0	0	0	0	0	20	25	30		0	100	100	100
GitOps Source Controller	25	30	40	60	80	80	80	80	90	100		0	0	100	100
GitOps Source Manager	0	0	0	0	0	0	0	0	0	0		0	0	100	100
Dash Button	0	0	0	0	0	0	0	0	10	50		0	100	100	100

3.3.4 Activity

The main activity in the first months of development was related to define the initial set of documents for the software, mainly the architecture document and requirements: deliverable D2.2. Development was also started, performing some initial research on some open-source tools such as Flux and The GitOps Toolkit to automate the deployment of Helm Charts based on predefined platform templates and application descriptor.

After the initial research, the original idea of fully developing from scratch the Chart Manager, Controller and Installer and define new CRDs (Custom Resource Definitions) for Kubernetes was changed, since a new and better way was found using the already provided CRDs from the GitOps Toolkit tool (https://fluxcd.io/flux/components/). Consequently, the module was re-designed to take advantage of this new idea of re-using an existing open-source tool. Different tests were performed to validate this innovative approach - basically creating sample Base Charts and module Helm Charts and installing them on the IDT.

Work has also been performed regarding using platform-based configuration for modules, meaning that modules will be able to use such configurations during installation time.

An initial installer was also created for the Chart Controller, Installer and GitOps Source Controller so the plan was slightly changed to prioritize the Source Controller, which was needed for the Installer, over the Chart Manager. Work has also started to design the first Base Charts and template Helm Charts for module installation.

In addition, even though the UI is not planned until M24, some research has also been performed to try to provide an initial version of that UI. Together with the IDT, work was enacted to test Rancher as the UI for installation.

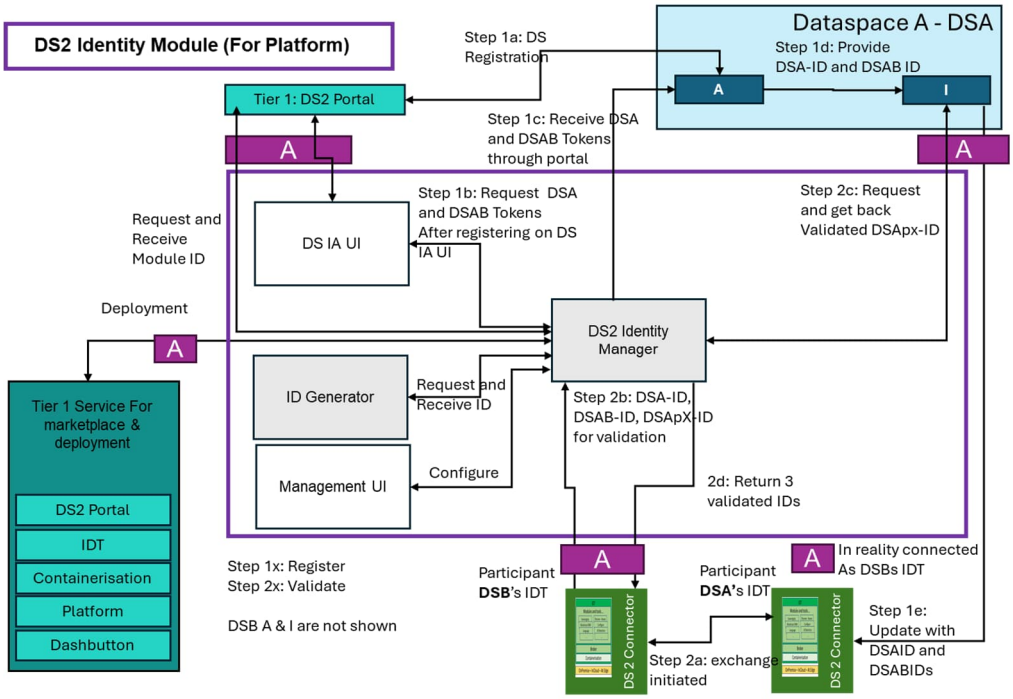
The next immediate steps are finishing the installer and integrating it together with the IDT installer and complete the first version of Base Charts and template module Helm Chart. Following that, complete development of the Chart Manager and start working with the Containerisation UI.

3.3.5 Use Case Validation Plan

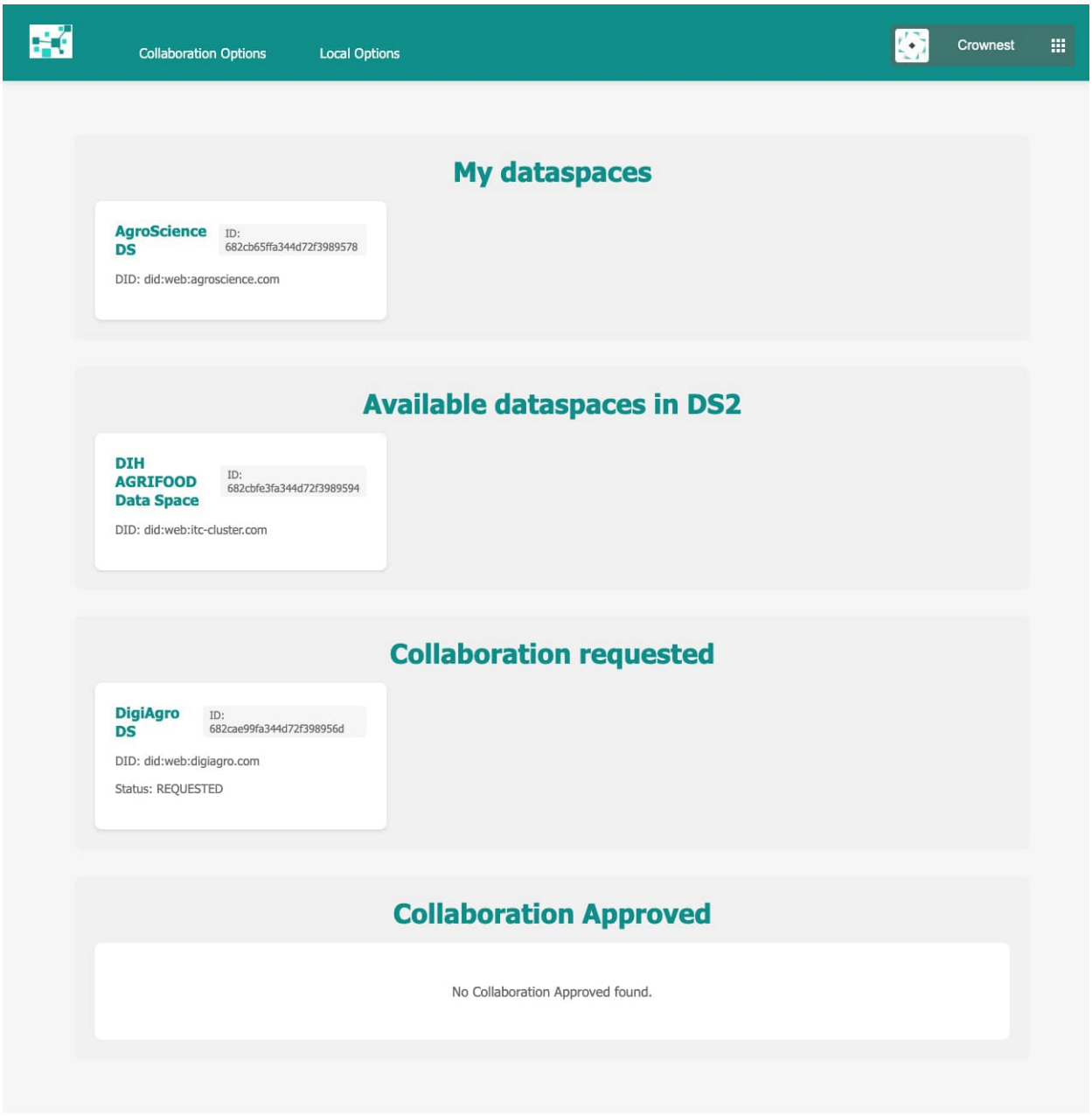
	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
CONT		UC2.2	

3.4 Module: IDM - Identity Module

3.4.1 Architecture Diagram



3.4.2 Sample Interface



3.4.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	- -	M18	M24	M30	M36
DS2 Identity Manager	10	20	25	35	45	70	75	80	85	90	- -	50	75	100	100
ID Generator	0	5	15	25	50	70	90	90	90	90	- -	0	0	100	100
DS IA UI	20	20	30	50	60	65	85	87	89	90	- -	100	100	100	100
Tier 1: DS2 Portal and API	0	0	10	40	50	60	70	70	70	70	- -	0	0	100	100
Management UI	0	0	0	0	25	40	50	50	60	60	- -	0	50	100	100
Tier 1: Services for marketplace & deployment and API	0	0	0	10	20	30	30	30	30	30	- -	0	0	100	100
IDT APIs	0	0	0	0	0	0	15	15	15	20	- -	0	100	100	100
DSx IA APIs	0	0	5	15	30	40	60	60	60	60	- -	0	0	100	100

3.4.4 Activity

The work started by analysing the initial set of documents for the software, mainly the architecture document and requirements: Deliverable 2.2 to clarify the scope of the component and the main functionality to be provided.

Development then started performing some initial research on Dataspace technology and identity systems in dataspaces. Extensive research has been performed on the EDC Framework related to identity management and lately in Decentralized Identities and Verifiable Credentials technology.

The main purpose of the work was to consider the primary existing identity managers in existing dataspaces and the mechanisms that DS2 should consider, reusing the identity credentials of the different companies and the actual mechanisms that can be used to verify the membership of the companies in their dataspaces.

Development quickly progressed to wallet visualisation functionality – wallets are used to store credentials. Verifiable Credentials of the different companies is one of the core elements to be considered and its integration with the EDC Identity Hub was successfully achieved in this period.

Once the basic mechanisms to host credentials for a given company were developed, the basic functionality to generate those Verifiable Credentials was provided. DS2 has provided its own implementation of the credential's issuer, capable of generating signed credentials against the dataspace agreement and which can be forwarded to specific companies.

This period has been also intensive in the development of the backend and frontend of the DS2 IDM portal, the set of UIs that allow the registration of datasets in DS2, the request for collaboration between dataspaces, the search of dataspaces already registered for the establishment of new agreements, the provision of actions

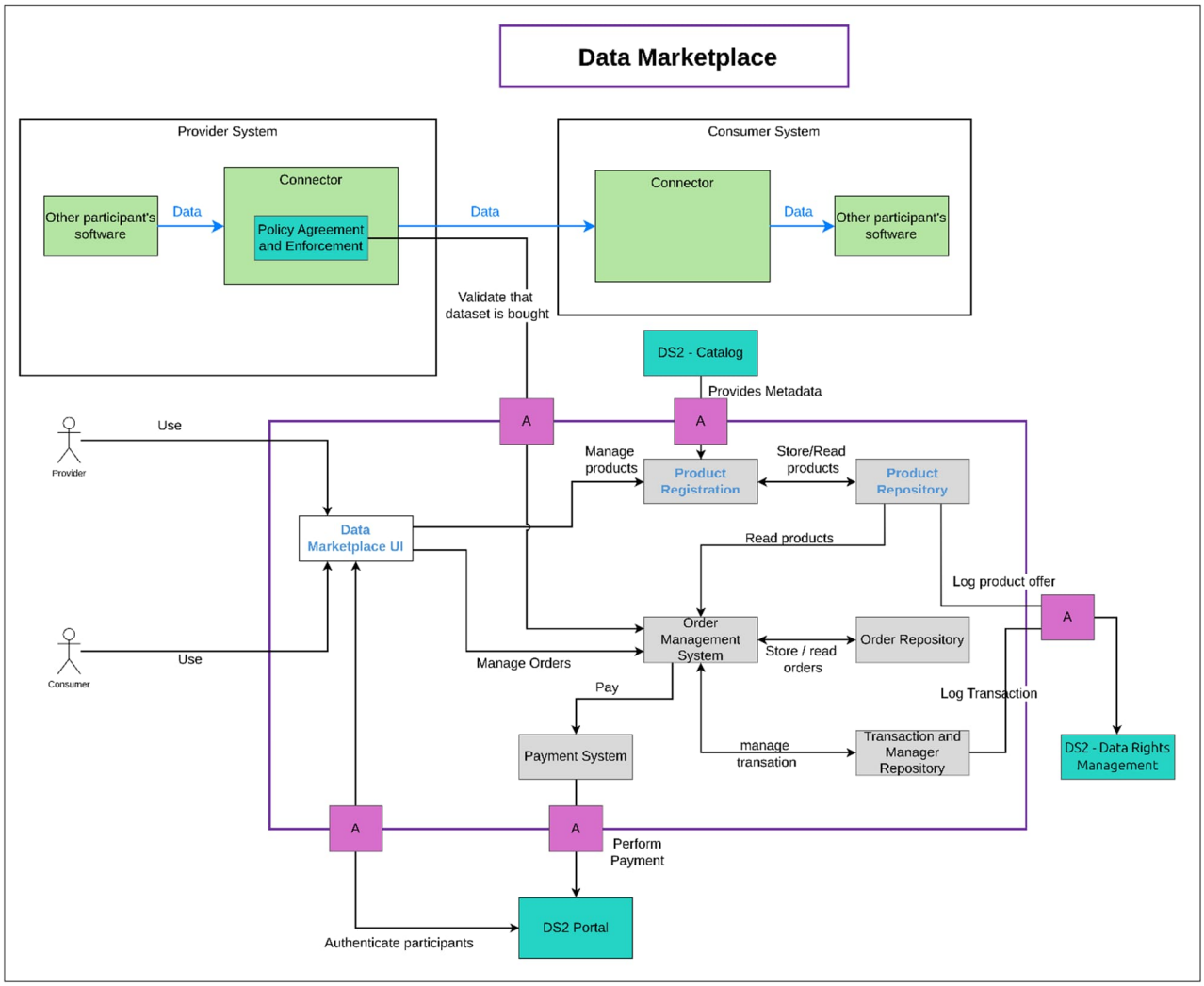
to accept or reject agreements, the visualisation of dataspaces policies, and the appropriate mechanisms to register companies registered in the DS2 portal as members of DS2.

3.4.5 Use Case Validation Plan

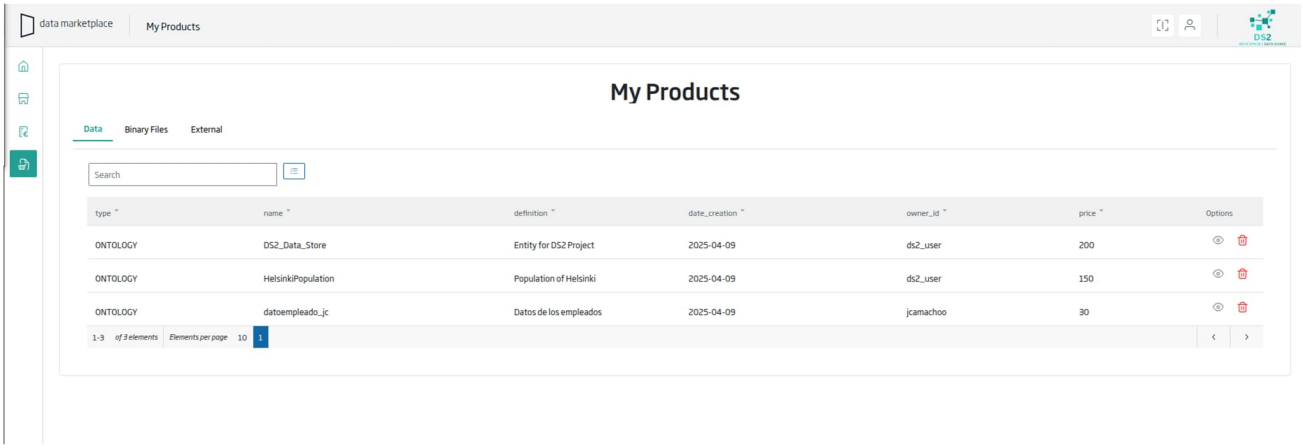
	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
IDM	UC3.1, UC3.2, UC3.3	UC2.1	UC1.1, UC1.2

3.1 Module: DMK - Data Marketplace Module

3.1.1 Architecture Diagram



3.1.2 Sample Interface



3.1.3 Primary Feature Progress in %

Component	0	1	2	3	4	5	6	7	8	9	--	M18	M24	M30	M36
Data Marketplace UI	0	0	15	20	20	20	20	20	20	20	--	20	100	100	100
Product Registration	0	10	10	20	20	20	20	20	20	20	--	20	100	100	100
Product Repository	0	10	10	20	30	30	30	30	30	30	--	30	100	100	100
Order Management System	0	0	0	5	20	20	20	20	25	25	--	25	100	100	100
Order Repository	0	0	0	0	0	0	0	0	0	0	--	0	100	100	100
Payment System	0	0	0	0	0	0	0	0	0	0	--	0	100	100	100
Transaction Repository	0	0	0	0	0	0	0	0	0	0	--	0	100	100	100
DS2 Portal	0	0	0	0	0	0	0	0	0	0	--	0	25	100	100
Data Right Management	0	0	0	0	0	0	0	0	0	0	--	0	25	100	100
DS2 Catalog	0	0	0	0	0	0	0	0	0	0	--	0	50	100	100
Policy Agreement and Enforcement	0	0	0	0	0	0	0	0	0	0	--	0	50	100	100
Bug Fixing and Maintenance	0	0	0	0	0	0	0	0	0	0	--	0	0	0	100

3.1.4 Activity

The data marketplace module allows the acquisition of datasets which are on its marketplace. The first feature release for this module was planned for M24 to allow focus on other INDRA modules. As a result, there are no major developments to put forward at M18. Nevertheless, development has already begun, and some non-functional requirements have been implemented.

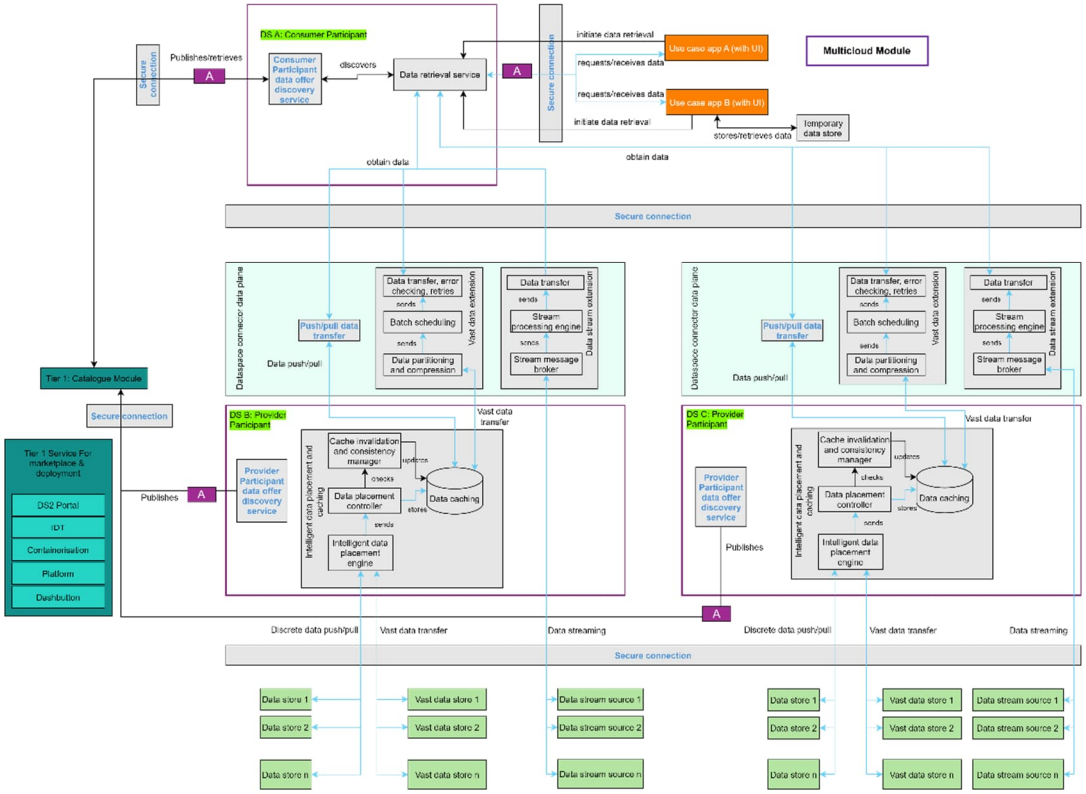
Two main advances can be highlighted: The initial development of the web application's UI foundation and the first version of the data model to support core operations. On the UI side, integration with Keycloak for authentication has been completed, along with support for internationalization. Regarding the data model, an initial alpha version is already in place, which will continue to evolve over the coming months.

3.1.5 Use Case Validation Plan

	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
DMK	UC3.1, UC3.2	UC2.1	

3.2 Module: MCL – Multi-cloud

3.2.1 Architecture Diagram



3.2.2 Interface

Per plan, the MCL module user interface is a future development for reporting period 2.

3.2.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	--	M18	M24	M30	M36
Stream message broker	100	100	100	100	100	100	100	100	100	100	--	100	100	100	100
Stream processing engine and data transfer	100	100	100	100	100	100	100	100	100	100	--	100	100	100	100
Data retrieval service	100	100	100	100	100	100	100	100	100	100	--	100	100	100	100
Data space discovery module	10	30	30	50	50	100	100	100	100	100	--	100	100	100	100
Push/pull data transfer	0	75	90	100	100	100	100	100	100	100	--	100	100	100	100
Intelligent data placement engine	0	0	0	0	0	0	0	10	15	20	--	20	80	100	100
Data placement controller	0	0	0	0	0	0	0	10	10	20	--	20	80	100	100
Data caching	0	0	0	0	0	0	15	25	25	25	--	30	80	100	100
Cache invalidation and consistency manager	0	0	0	0	0	0	0	10	10	10	--	10	50	100	100
Data partitioning and compression	0	0	0	0	0	0	0	0	0	0	--	0	100	100	100
Batch scheduling	0	0	0	0	0	0	0	0	0	0	--	0	100	100	100
Data transfer, error checking, and retries	0	0	0	0	0	0	0	0	0	0	--	0	100	100	100
Integration and bug-fixes	0	0	0	0	0	0	0	0	0	0	--	0	50	100	100

3.2.4 Activity

The activities conducted for the MCL module began with the implementation of data streaming and processing functionalities which will be later integrated into the data streaming extension of the data space connector. The stream message broker, stream processing engine and data transfer, and data retrieval service were all established at operational capacity from the outset, ensuring that the system could reliably manage real-time data flows and retrieval tasks. Parallel to these efforts, the data space discovery module was incrementally developed, reaching deployment at test servers after a series of enhancements, which enabled dynamic identification and management of available data spaces within the system.

Push/pull data transfer capabilities were introduced and rapidly advanced to completion, allowing for discreet data transfer. As the foundational streaming and transfer mechanisms stabilised, attention shifted towards more advanced features. The intelligent data placement engine and data placement controller were initiated, with desk research conducted and an early-stage implementation planning produced. Desk research on data

caching was also progressively developed in parallel to the above covering the cache invalidation and consistency manager. The idea behind it is to ensure data integrity and coherence across distributed caches.

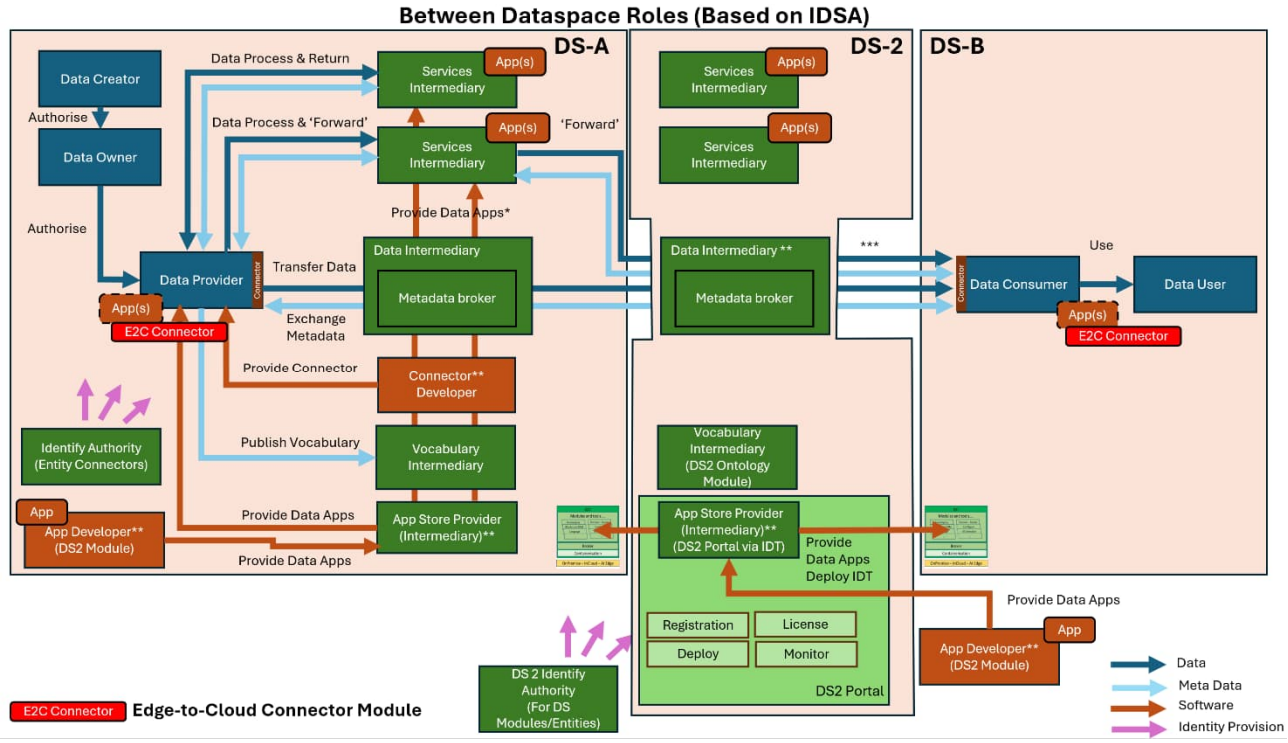
Other advanced functionalities, such as data partitioning and compression, batch scheduling, and robust mechanisms for data transfer error checking and retries, have not been started yet. Along with that, during RP2, continuous integration and bug-fixing activities will be performed alongside feature development to maintain system stability and address issues as they arise.

3.2.5 Use Case Validation Plan

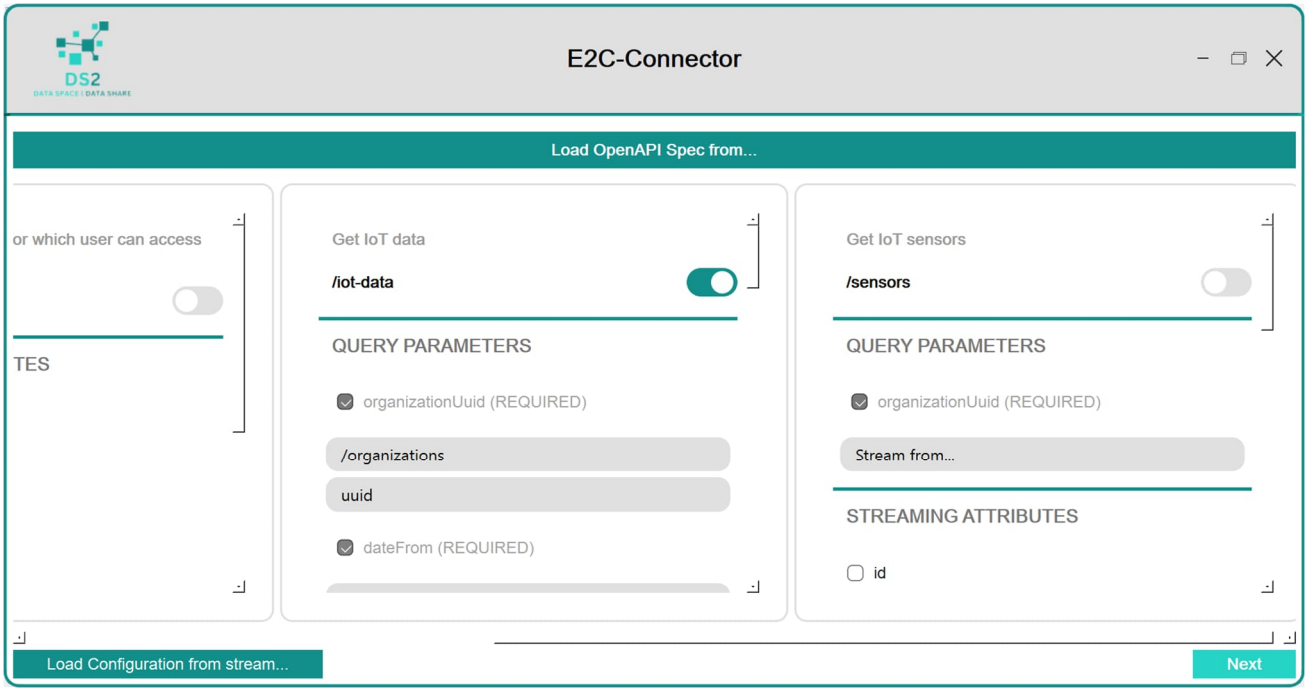
	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
MCL		UC2.2	

3.3 Module: E2C - Edge to Cloud Module

3.3.1 Architecture Diagram



3.3.2 Sample Interface



3.3.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	-	M18	M24	M30	M36
Configuration UI	20	25	30	40	40	40	40	40	40	40	-	40	60	100	100
UI-to-Service Connector	0	20	30	30	30	30	30	30	30	30	-	30	60	100	100
Spec Interpreter	75	75	80	80	80	80	80	80	80	80	-	80	90	100	100
Data Value Selector	70	70	70	70	70	70	70	70	70	70	-	70	90	100	100
Quality Assurance	30	30	30	35	35	35	40	40	40	40	-	40	75	100	100
DateQuery Language	20	20	20	20	20	20	20	20	20	20	-	20	75	100	100
Custom-Value Interpreter	20	20	20	20	20	20	20	20	20	20	-	20	75	100	100
Dynamic Parameter Mapper	50	50	50	50	50	50	50	50	50	50	-	50	80	100	100
Streaming service	30	35	35	35	35	35	40	40	40	40	-	40	75	100	100
Cloud mapper	50	50	50	50	50	70	70	70	70	70	-	70	90	100	100
Containerization	20	20	25	25	25	25	25	25	25	25	-	25	75	100	100
Bug Fixing and Maintenance	0	0	0	0	0	0	0	0	0	0	-	0	25	100	100

3.3.4 Activity

At the beginning of the project, the team started designing and developing the E2C module with the intent to provide connectivity of edge devices and existing API endpoints to cloud platforms, noting that these edge devices have limited processing capabilities and usually limited bandwidth. E2C would then make the data, that either resides on local devices or existing API endpoints, available for future data exchange in dataspaces using a specifically designed high-availability technology. Here, IoT platforms are a natural choice. Many providers of IoT platforms supply their own software for transferring data to their platforms. To prevent a lock-in effect for a specific vendor or proprietary products, the E2C module is designed to allow for a multitude of target platforms such as Cumulocity IoT, Azure IoT Hub, etc. Additionally, its connectivity is built on top of the open-source product “thin-edge.io” which was designed to be extensible, so new target platforms can be interconnected with little effort to suit all potential use cases.

At first, Cumulocity IoT, Software AG’s IoT platform was selected as cloud platform, so development and testing began by setting up and hosting an instance of Cumulocity for DS2. By closely collaborating with DS2 use case partners, requirements for streaming data from their systems were collected. As a first application scenario, the Green Deal use case was selected. Here, an API endpoint was already described using OpenAPI specification in which all capabilities and data types of a service are defined in a standardised way.

E2C consists of two components, a streaming service and a user interface which can both be deployed as separate docker containers. The streaming service carries out the upload of data to cloud platform instances,

while the user interface is used to configure the streaming services. Here, E2C was built to interpret OpenAPI specifications to allow users to directly select all relevant attributes of API endpoints that are to be streamed to one of the supported IoT platforms by a streaming service. The user interface was built according to the DS2 design style provided to the consortium and allows to define data queries, time windows and data mappings.

With Cumulocity IoT being a proprietary cloud platform, the team began exploring open-source alternatives that can be used as target platform. Not only does this simplify the use of an IoT platform for use case partners by not having to manage licensing, but it also serves as a proof-of-concept for the extensibility of the streaming service's upload mechanism to target platforms. The team chose to support the open-source IoT platform ThingsBoard (https://thingsboard.io), set up and hosted an instance of ThingsBoard and extended the streaming service interface respectively. As a result, in the user interface, users can now choose to stream data either to instances of Cumulocity IoT or ThingsBoard.

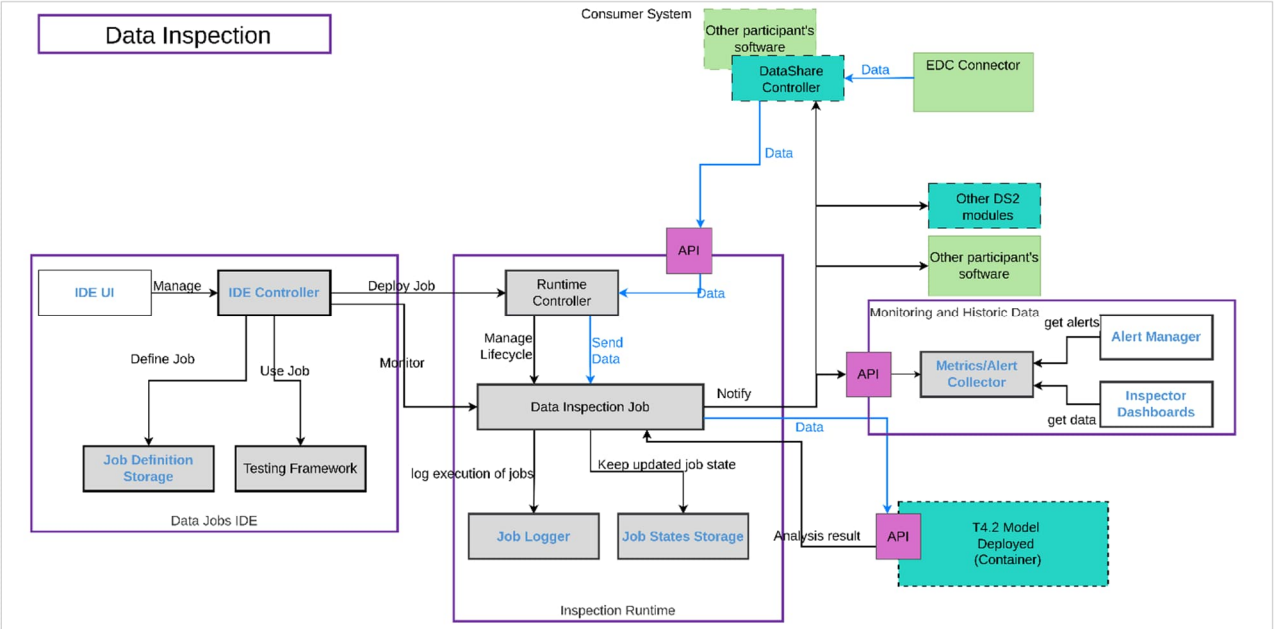
In preparation for the Darmstadt plenary meeting, a demonstrator was designed based on Green Deal use case data. In the weeks following the meeting, maintenance was conducted for the streaming service's northbound interface, and work concerning an easier replication of streaming services started. The team also began extending the user interface to make it easier to configure multiple instances of streaming services, allowing users to manage uploading data from more than one data source in a more convenient manner.

3.3.5 Use Case Validation Plan

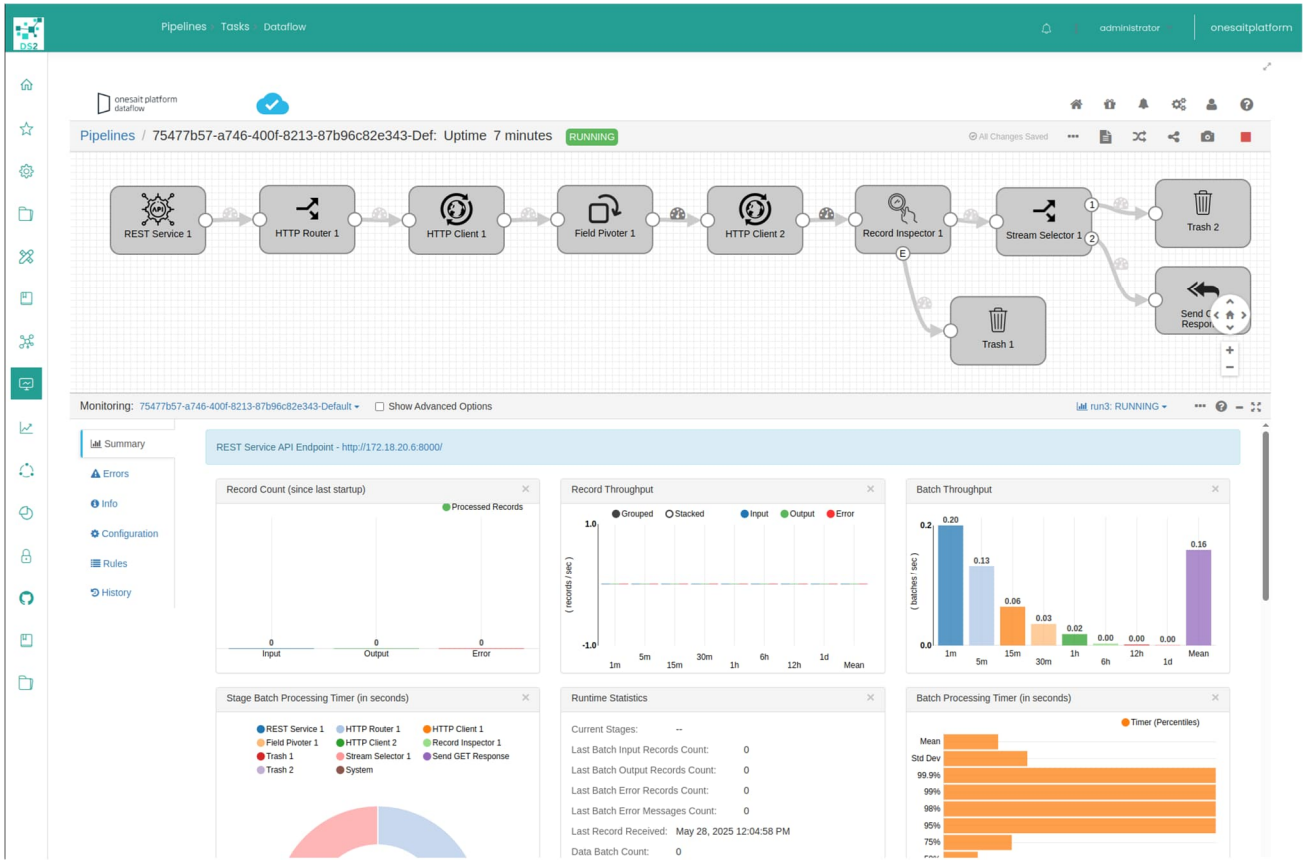
	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
E2C	UC3.1, UC3.2, UC3.3	UC2.1	

3.4 Module: DINS - Data Inspection Module

3.4.1 Architecture Diagram



3.4.2 Sample Interface



3.4.3 Primary Feature Progress in %

Component	0	1	2	3	4	5	6	7	8	9	- -	M18	M24	M30	M36
IDE UI	10	20	25	50	55	65	70	80	90	100	- -	100	100	100	100
IDE Controller	10	20	50	60	60	70	75	85	95	100	- -	100	100	100	100
Job Definition Storage	25	30	70	70	75	75	75	80	90	90	- -	90	100	100	100
Testing Framework	50	50	50	50	50	50	50	50	50	50	- -	50	100	100	100
Runtime Controller	10	10	30	60	70	75	80	85	95	100	- -	100	100	100	100
Data Inspection Job	0	15	15	15	30	40	50	70	90	100	- -	100	100	100	100
Job Logger	0	0	0	0	0	0	0	0	0	0	- -	0	100	100	100
Job States Storage	0	0	0	0	0	0	0	0	0	0	- -	0	100	100	100
Metrics/Alert Collector	0	0	0	0	0	0	0	0	0	0	- -	0	50	100	100
Alert Manager	0	0	0	0	0	0	0	0	0	0	- -	0	50	100	100
Dashboard	0	0	0	0	0	0	0	0	0	0	- -	0	50	100	100
T4.2 Model Deployed	0	0	0	0	0	0	0	0	0	0	- -	0	100	100	100
DataShare Controller	0	0	0	0	0	0	0	0	0	0	- -	0	50	100	100

3.4.4 Activity

Substantial progress has been achieved in developing the DINS module. The development of the module is more advanced than initially expected, but some adjustments have been made in terms of feature prioritization. The Testing Framework, initially planned for M18, has been delayed. This decision was made to accelerate the development of other features that are needed by the PAE module. As a result, several components are now almost complete before their original M24 schedule. The Testing Framework has been rescheduled for M24.

After the rescheduling, the focus has been the separation of runtime and development engines for data pipelines; therefore, the IDE and Runtime Controllers will be ready by M18. This architectural refinement supports the dynamic creation of data pipelines from templates. Capability, that will be used by PAE module

to enforce policies. The new architecture required modifications to the internal data model, which have already been applied.

The authorization mechanisms controlling access to pipeline instances have also been updated. As stated in the module description, DINS is an evolution of a previous background that it is part of INDRA's Onesait Platform. A revised authorization mechanism has been implemented, driven by integration needs with the Onesait Platform's security layer. This is a key step in aligning the DINS module with the access control strategies.

From a user interface perspective, several improvements have been introduced to enhance interaction with the pipeline engine components. A redesigned control panel for managing pipelines is now almost complete, and UI support for job and task management has been implemented. Jobs and Task represent the data pipeline at runtime. This includes the addition of visual consistency through the DS2 theme, and the integration of AngularJS-based IDE components into the Onesait Platform user interface.

A new stage in the standard library of components to define data pipelines has been developed to allow users to define and manage inspection rules more intuitively, providing greater flexibility in pipeline composition.

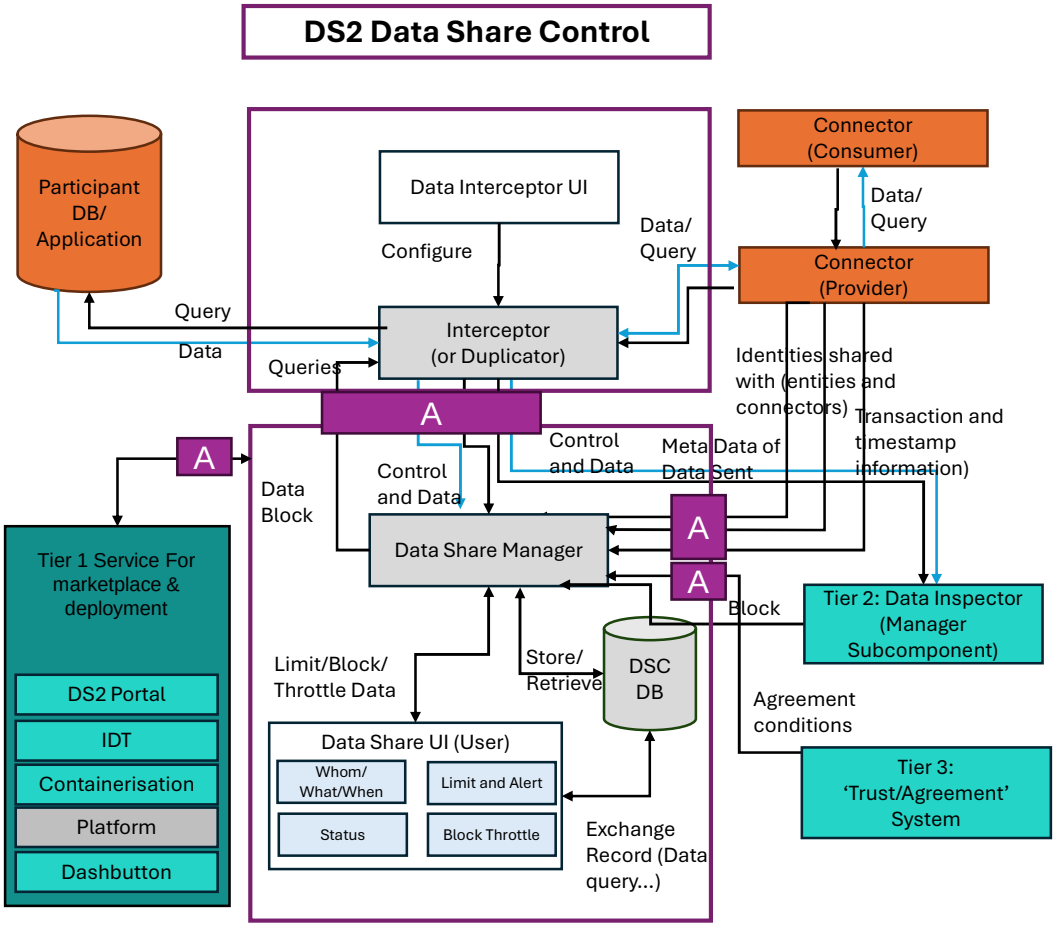
Simultaneously, a modernization effort is being done to bring the software stack up to date. That includes the migration from Java 8 to Java 21 and the upgrade of Jetty from version 9 to 12. Additionally, the update of outdated libraries was necessary to ensure compatibility and security. These updates have necessitated the replacement of Swagger 1 with OpenAPI 3 for API documentation, aligning with standards and improving maintainability. As part of this process, the API layer itself has been refactored to better support the workflow for creating and managing versions, jobs, and tasks within the platform. New endpoints and features are being validated.

3.4.5 Use Case Validation Plan

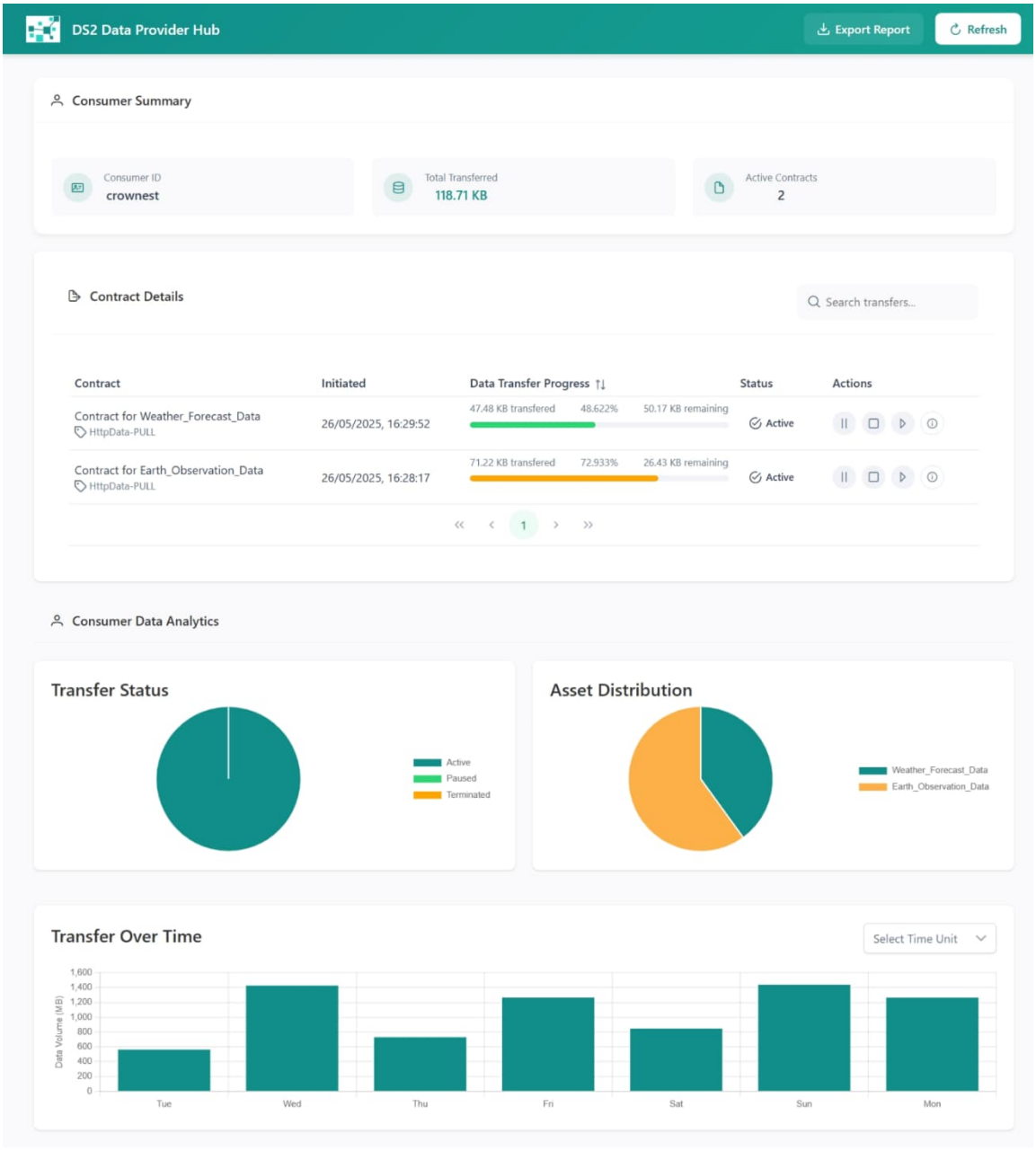
	KEY:	**Default install, UC = Validation**	**Specific To Case UC = Validation**
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
DINS		UC2.5	UC1.1, UC1.2, UC1.4

3.5 Module: DSHARE - Data Share Controller

3.5.1 Architecture Diagram



3.5.2 Sample Interface



3.5.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	--	M18	M24	M30	M36
Data Share Manager	15	20	30	60	70	75	80	90	90	95	--	50	75	100	100
Data Share UI	30	30	60	60	80	80	85	90	90	95	--	75	100	100	100
DSC DB	0	0	0	0	50	60	70	95	95	95	--	100	100	100	100
Tier 1: Service Stack	0	0	0	0	30	50	70	70	70	72	--	0	0	100	100
Tier 2: Data Inspector Manager	0	0	0	0	0	0	0	0	0	0	--	0	0	100	100
Tier 3: Trust Environment	0	0	0	0	0	10	10	10	10	10	--	0	100	100	100
Interceptor	10	10	80	80	80	80	80	80	80	82	--	60	100	100	100
Interceptor UI	0	0	0	0	0	0	0	0	0	0	--	0	100	100	100
Connector	0	0	0	0	0	15	20	50	50	50	--	50	75	100	100

3.5.4 Activity

The primary activity in the first months of development has been related to define the initial set of documents for the software, mainly the architecture document and requirements: Deliverable 2.2. Development started performing some initial research on Dataspace technology capabilities for monitoring data exchanged between connectors and alternatives approaches such as proxies or Kubernetes injection mechanisms. The final decision was to focus the development of the data sharing component as an extension of the Eclipse Dataspace Connector (EDC). The work has centered around on creating an interceptor-based solution to track and measure data exchanges between connectors, with particular emphasis on the Provider's data plane public API endpoint. This development aims to establish monitoring and potentially control data transfer volumes within the dataspace.

The interface enables real-time tracking of data exchange, highlighting monitoring and blocking functionalities. Additionally, a dedicated testing server was set up to simulate HTTP data transfer requests, allowing for improved validation and debugging. Several features were integrated, including weekly data charts, alerts for transfer limits, and support for handling parallel data requests. These improvements provide a more comprehensive view of data flow within the dataspace and help refine monitoring capabilities.

ICE developed new APIs to track transfer sizes, contract IDs, and user identification tokens, enabling a more granular view of data exchanges. These extensions allow for real-time monitoring of transfer metrics, including per-user contract counts and transfer sizes, which can be visualised in a dashboard. Additionally, PostgreSQL integration was improved to store and retrieve these metrics efficiently.

After laying out the background elements of the component, the development diverted into a complete frontend interface that displays all consumers associated with a provider and visualizes their active contracts using previously implemented monitoring APIs. In parallel, we began deeper research into the Consumer Hub, focusing on the internal data transfer flows. This research led to the conceptualization of new APIs that allow providers and consumers to register and manage multiple data planes, enabling scalable multi-user support and dynamic transfer handling.

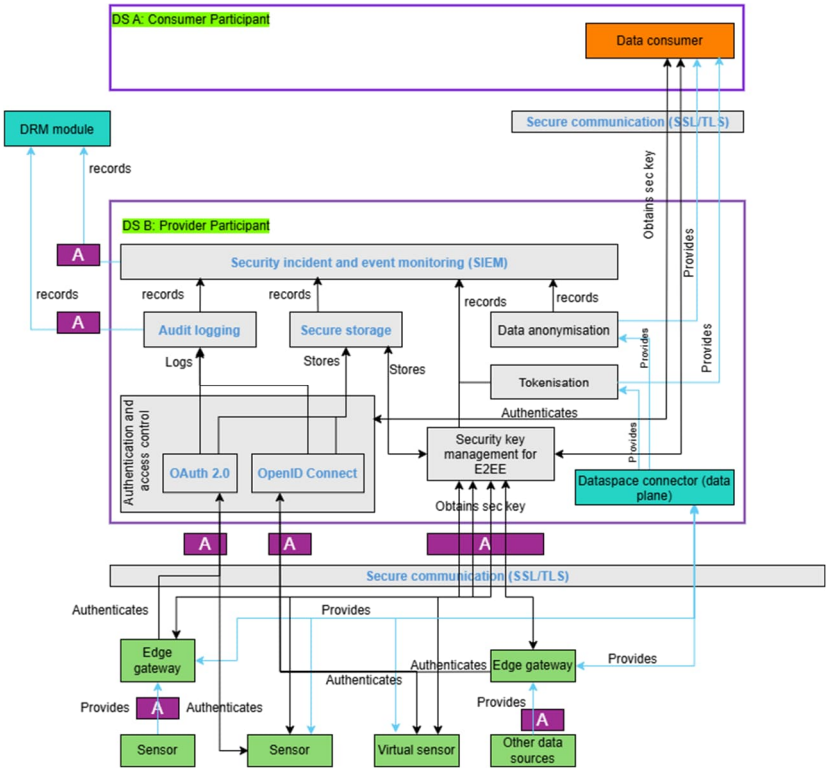
The latest work conducted in this iteration is focused on redesigned key elements for clarity, implemented contract management controls, and to introduce analytics for consumption pattern monitoring. The new consumer ranking system based on data usage provides valuable insights for resource allocation. These targeted developments significantly improve system usability, control mechanisms, and data-driven decision-making capabilities.

3.5.5 Use Case Validation Plan

	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
DSHARE	UC3.1, UC3.2	UC2.1	UC1.1

3.6 Module: SEC – E2C Security

3.6.1 Architecture Diagram



3.6.2 Interface

The SEC module components operate in background, handling tasks like authentication, tokenisation without direct user interaction. Since its primary function is to enable secure interactions and safeguard data rather than facilitate user workflows, this module does not require a user interface.

3.6.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	- -	M18	M24	M30	M36
Authentication	100	100	100	100	100	100	100	100	100	100	- -	100	100	100	100
Access control	100	100	100	100	100	100	100	100	100	100	- -	100	100	100	100
Audit logging and secure storage	100	100	100	100	100	100	100	100	100	100	- -	100	100	100	100
Tokenisation	100	100	100	100	100	100	100	100	100	100	- -	100	100	100	100
Data anonymisation	0	30	80	85	85	95	100	100	100	100	- -	100	100	100	100
SIEM	0	0	0	10	10	15	20	30	30	50	- -	50	100	100	100
Security key management	0	0	0	0	0	0	0	0	0	0	- -	0	100	100	100

3.6.4 Activity

The development activities carried out for the SEC module focused initially on establishing a robust foundation of core security features. Authentication, access control, audit logging with secure storage, and tokenisation were implemented at full functionality from the very beginning, ensuring that the system had strong identity and access management, comprehensive logging capabilities (at the blockchain of DRM module), and secure token-based operations in place early on. Data anonymisation was developed then, starting with basic implementations that were continuously refined and enhanced over time until reaching maturity. This gradual improvement reflects careful attention to privacy requirements and compliance with data protection requirements of the use cases, ensuring that sensitive information is effectively anonymised within the DS2 ecosystem. Security Information and Event Management (SIEM) capabilities were introduced after the core security mechanisms were stabilised in parallel to the data anonymisation part. Initial SIEM functionality was integrated and then incrementally enhanced to provide effective monitoring, detection, and response features tailored to the platform's security needs. This part is still ongoing.

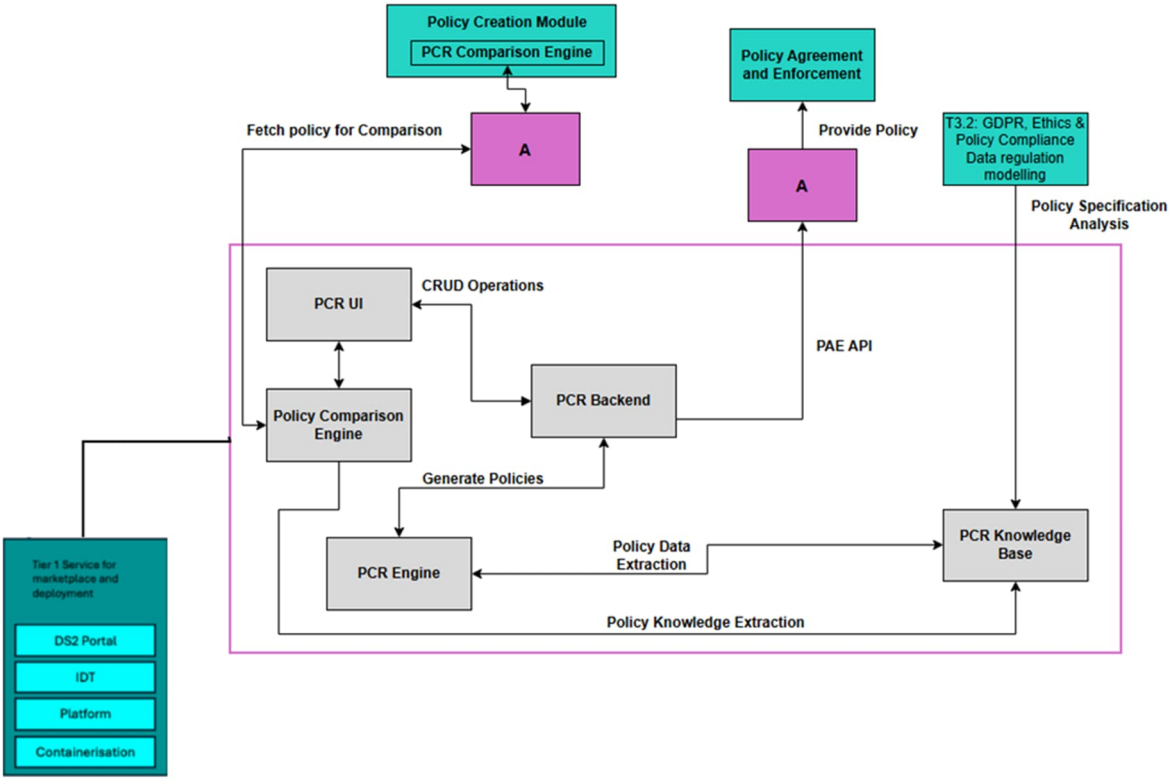
The immediate next tasks for the SEC module are to complete the SIEM deployment to its full capacity. Upcoming work will aim to expand SIEM capabilities to provide comprehensive security monitoring, real-time threat detection, and automated incident response. This will require integration with log sources across the platform, fine-tuning alerting mechanisms, and ensuring that SIEM outputs are actionable and support compliance requirements. For security key management which has not yet been initiated, future development will prioritise deployment of a robust key management system. Integration with other modules of DS2 will be a parallel focus. Collaborative testing and iterative refinement will be essential to ensure that the security architecture remains cohesive as new features are introduced and integrated.

3.6.5 Use Case Validation Plan

	KEY:	Default install, UC = Validation	Specific To Case UC = Validation
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
SEC		UC2.1	

3.7 Module: PCR - Policy Creation Module

3.7.1 Architecture Diagram



3.7.2 Sample Interface

As mentioned elsewhere, the need for an additional module - Policy Creation (PCR)- was identified by the consortium around 2025-03 and although the module is documented in this section of the deliverable, software development will not start before M19. Therefore, a sample interface is currently unavailable at this point but will be made available in the next iteration of this deliverable (M30).

That said, the Policy Creation (PCR) module 's User Interface will be designed to align closely with the DS2 framework, prioritizing simplicity, clarity, and ease of use and by leveraging Digital Rights Language (ODRL) will offer a sophisticated yet simplified approach to policy generation. The layout will be intuitive, allowing users to interact with the system efficiently without requiring extensive training or technical knowledge.

Once logged in through the Dash Button of DS2, users will be able to:

- Create new policies through guided forms or structured input fields.
- Edit existing policies with clear visual indicators of editable elements.

- Download policies in standard formats for offline use or sharing.
- Apply policies directly to the Policy Agreement and Enforcement (PAE) module through integrated options.
- Compare policies side-by-side using a built-in comparison view, highlighting key differences for transparency and integrity of information between the negotiation parties (DS2 participants).

Overall, the Policy Creation (PCR) User Interface will be lightweight and responsive, built with usability in mind to support both expert users and newcomers within the DS2 environment.

3.7.3 Primary Feature Progress in %

Functionality	0	1	2	3	4	5	6	7	8	9	--	M18	M24	M30	M36
PCR UI	-	-	-	-	-	-	-	-	-	-	-	-	70%	100%	
PCR Engine	-	-	-	-	-	-	-	-	-	-	-	-	60%	100%	
PCR Comparison Engine	-	-	-	-	-	-	-	-	-	-	-	-	70%	100%	
PCR Knowledge Base	-	-	-	-	-	-	-	-	-	-	-	-	70%	100%	

3.7.4 Activity

The Policy Creation (PCR) module's initial architecture has been created and the integration and interfacing flow with the DS2 portal and DS2 modules such as Policy Agreement and Enforcement (PAE) has been identified with technical discussions underway to enable Dataspace-to-Dataspace policy comparison using defined APIs.

Furthermore, a discussion at policy level is also underway with WP3 to align on policy structure and inputs. The Policy Creation (PCR) will not rely upon hardcoded policies that will limit the use of the module but will be fully configurable and flexible allowing users to create, edit, compare any policy they wish to agree upon as part of their data-sharing agreement. Specifically, the Policy Creation (PCR) module will be use a library of standard policy clauses rather than predefined rules, with WP3 provide these clauses to support implementation and assist definition of milestones.

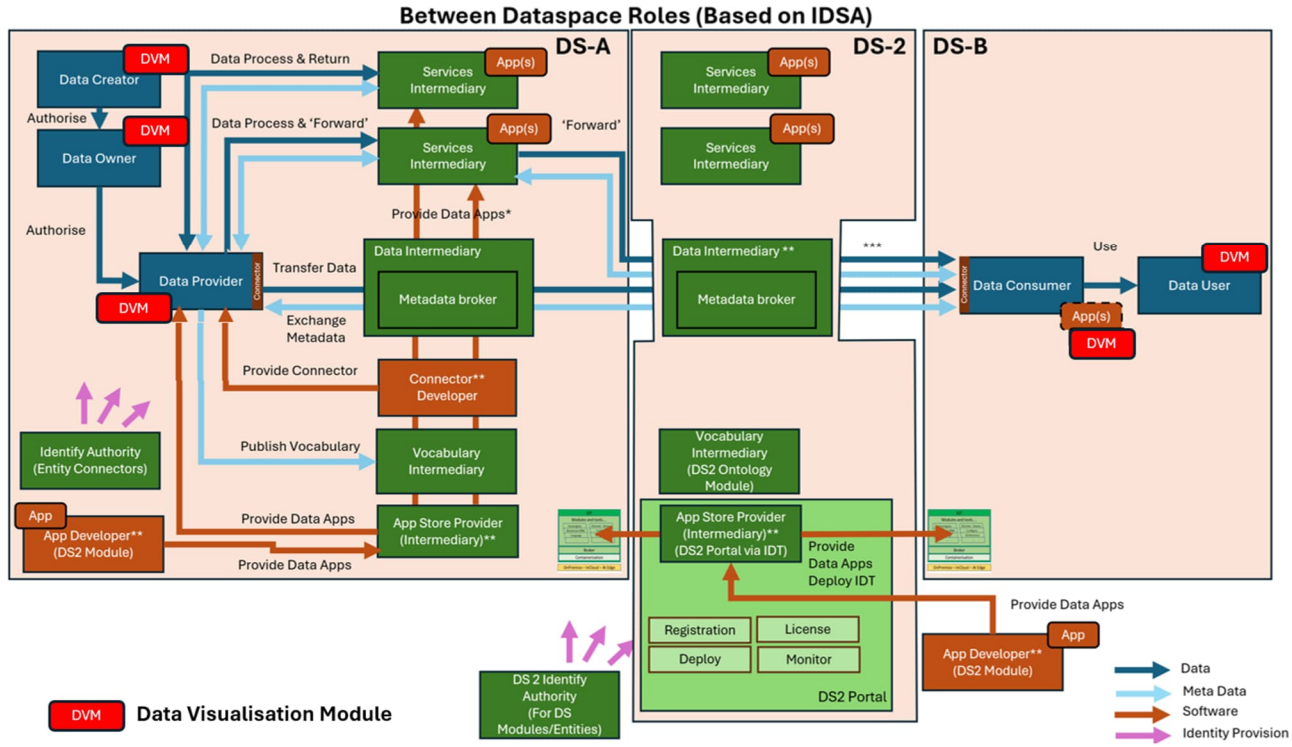
As part of the preparation process of software development, ATC has been investigating the technological landscape to identify those technologies that would be the best fit for PCR. For this matter, an ongoing discussion is taking place with the UPCAST project (https://www.upcast-project.eu/) to investigate the possibility of adopting and fine-tuning an existing policy creation and negotiation tool and explore its integration into the DS2 framework.

3.7.5 Use Case Validation Plan

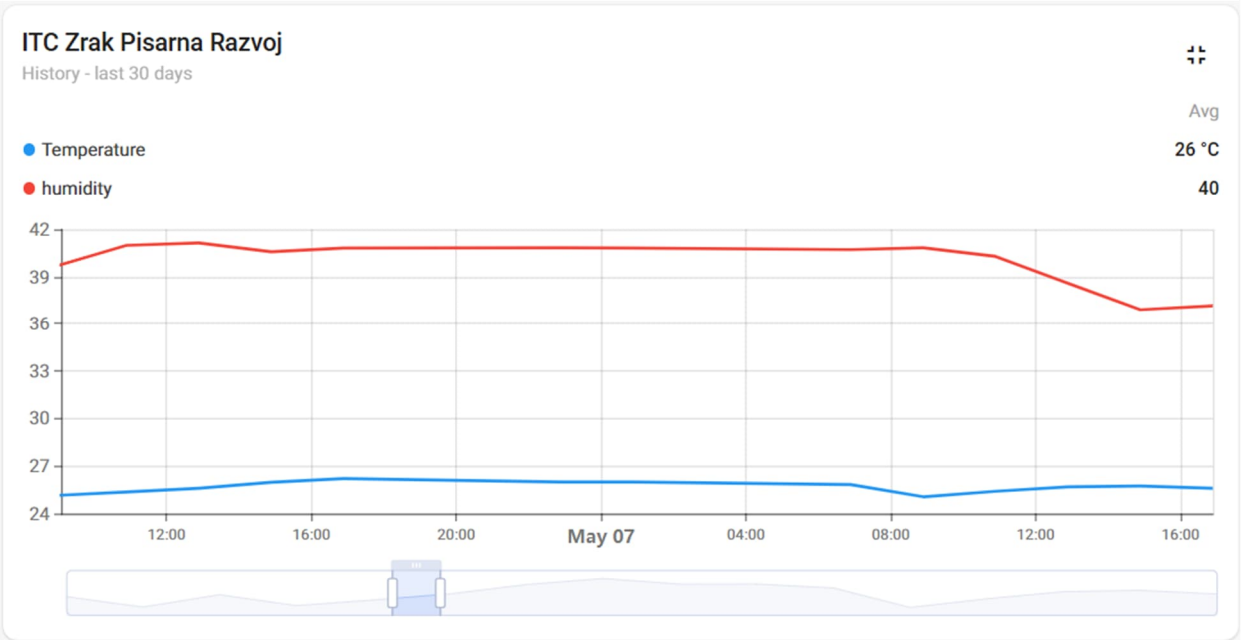
	KEY:	**Default install, UC = Validation**	**Specific To Case UC = Validation**
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
PCR		UC2.1	

3.8 Module: DVM - Data Visualisation Module

3.8.1 Architecture Diagram



3.8.2 Sample Interface



3.8.3 Primary Feature Progress in %

DVM is a new module and will be planned and implemented by Bluebridge Solutions starting July (pending Amendment and EU approval of company transition from SWAG to Bluebridge Solutions, see 2.3), hence concept and feature planning is still in progress.

3.8.4 Activity

A general idea of this module was discussed in Darmstadt plenary meeting in 05-2025: The bluebridgesolutions team proposed to separate and generalise data visualization functionalities developed by SWAG for E2C and DDT into a new module that could also be used by other partners for data visualization (without the need to use E2C or DDT).This will significantly enhance the presentation and communication capabilities in the project and facilitate a better understanding of raw data by means of graphical analysis.

3.8.5 Use Case Validation Plan

	KEY:	**Default install, UC = Validation**	**Specific To Case UC = Validation**
Module ID	**Precision Agriculture**	**Green Deal**	**City Scape**
DVM	UC3.1, UC3.2	UC2.1	

4 ADDITIONAL ACTIVITIES

Additional non-module activities are presented below.

4.1 Experimentation Facility – [EXP]

DS2 is committed to provide an experimental data space whose infrastructure can be used as a baseline for developing customized data space solutions for specific application domains or use cases where appropriate to DS2 technical parties and users. This allows the testing of business concepts and building of proof of concepts before investing in real-life solutions. It is implemented via VTT's Data Space Innovation Lab (DSIL).

4.1.1 Architecture Diagram

The architecture diagram below illustrates the overall structure of DSIL highlighting its alignment with established data space reference principles and tailored for use within the DS2 project context.

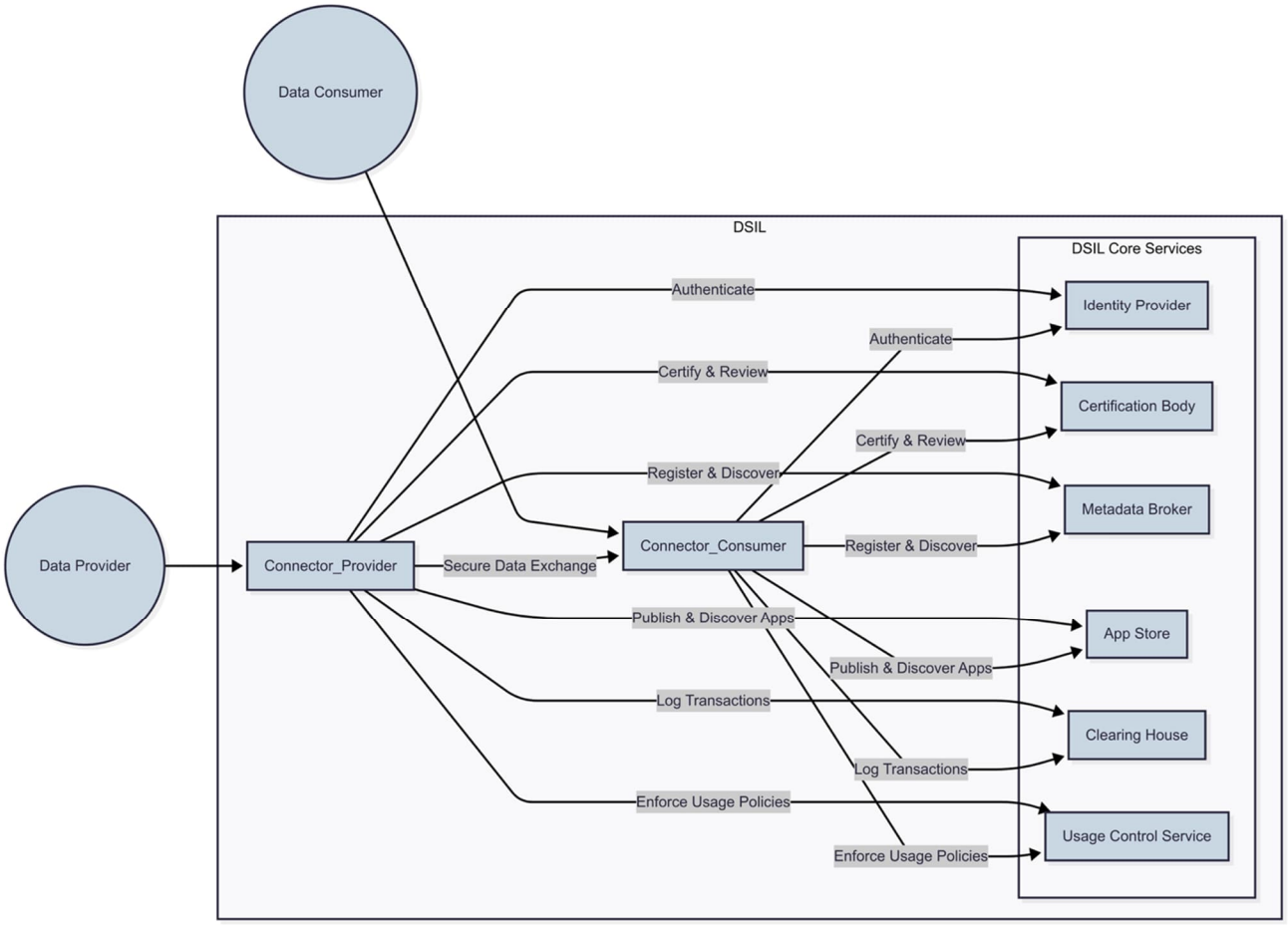
Key actors include:

- Data Provider: Supplies datasets into the DSIL ecosystem through a dedicated Provider Connector, ensuring secure entry and interaction within the data space environment.
- Data Consumer: Retrieves or consumes datasets using a separate Consumer Connector, leveraging the established DSIL interfaces and infrastructure.
- Connector Provider and Consumer: Facilitate secure and standardized data exchanges between providers and consumers. These connectors manage interactions such as authentication, transaction logging, certification, policy enforcement, and service discovery.
- DSIL Core Services (hosted centrally at VTT):
    - Identity Provider: Manages authentication of participants, verifying identity and permissions.

- Certification Body: Reviews and certifies connectors and participants, ensuring adherence to standards and interoperability.
- Metadata Broker: Provides registration and discovery services, enabling efficient identification and usage of available data offerings.
- App Store: Enables discovery and publication of interoperable applications and services.
- Clearing House: Logs and audits transactions, offering traceability and transparency.
- Usage Control Service: Enforces defined data usage policies, ensuring compliance with terms and conditions of data sharing agreements.

Interactions within this ecosystem follow clear and predefined pathways, ensuring secure and efficient data sharing aligned with both technical requirements and governance policies.

This architecture effectively supports the core objectives of DSIL, providing an experimental and scalable environment for robust data space functionalities, aligned with the broader goals of the DS2 project.



4.1.2 Sample Interface

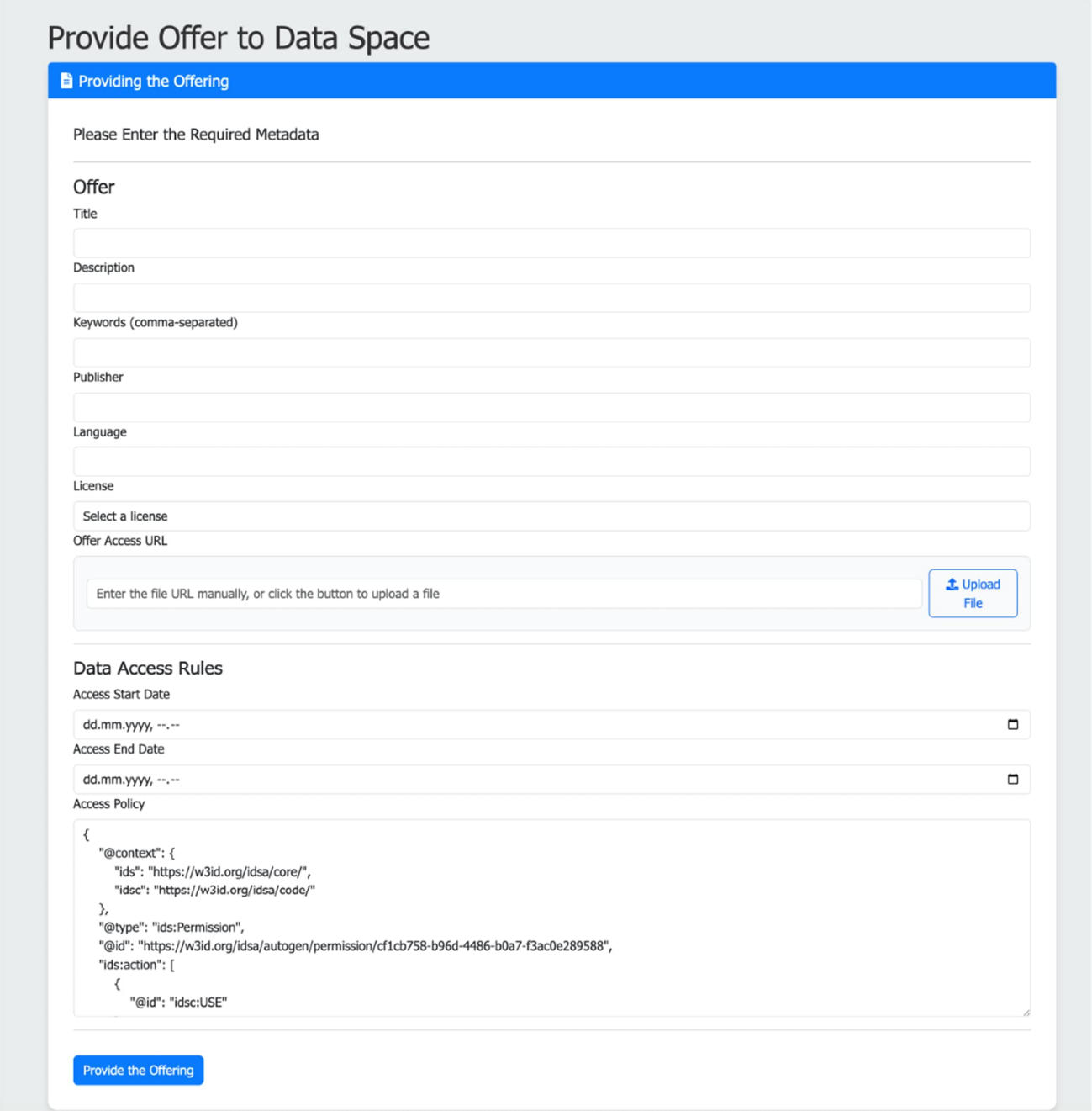
Provider Interface

The provided screenshot demonstrates the DSIL Data Provider Interface, designed to simplify the process of defining and publishing data offerings within the DSIL environment. This intuitive web application, fully Dockerized and seamlessly integrated with the Data Space Connector, enables providers to:

- Easily input essential metadata such as title, description, keywords, publisher, language, and licensing.

- Specify data access URLs, supporting static file uploads or endpoint definitions for internet-facing web applications.
- Clearly define detailed access policies and restrictions via standardized JSON structures to enforce precise data governance and compliance requirements.

This interface ensures a straightforward workflow, guiding data providers clearly through the essential steps required to share datasets securely and effectively within the DSIL ecosystem.

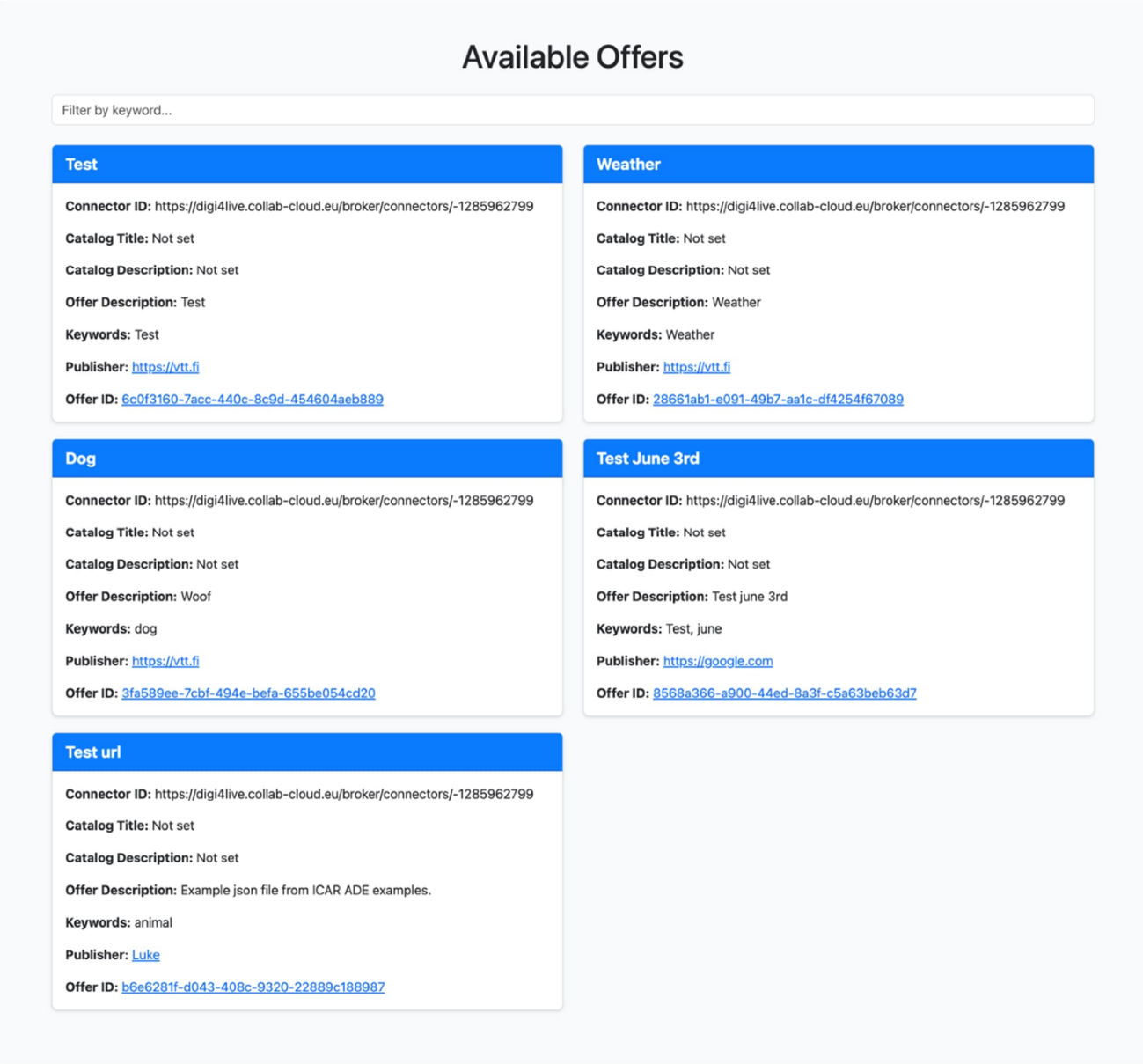


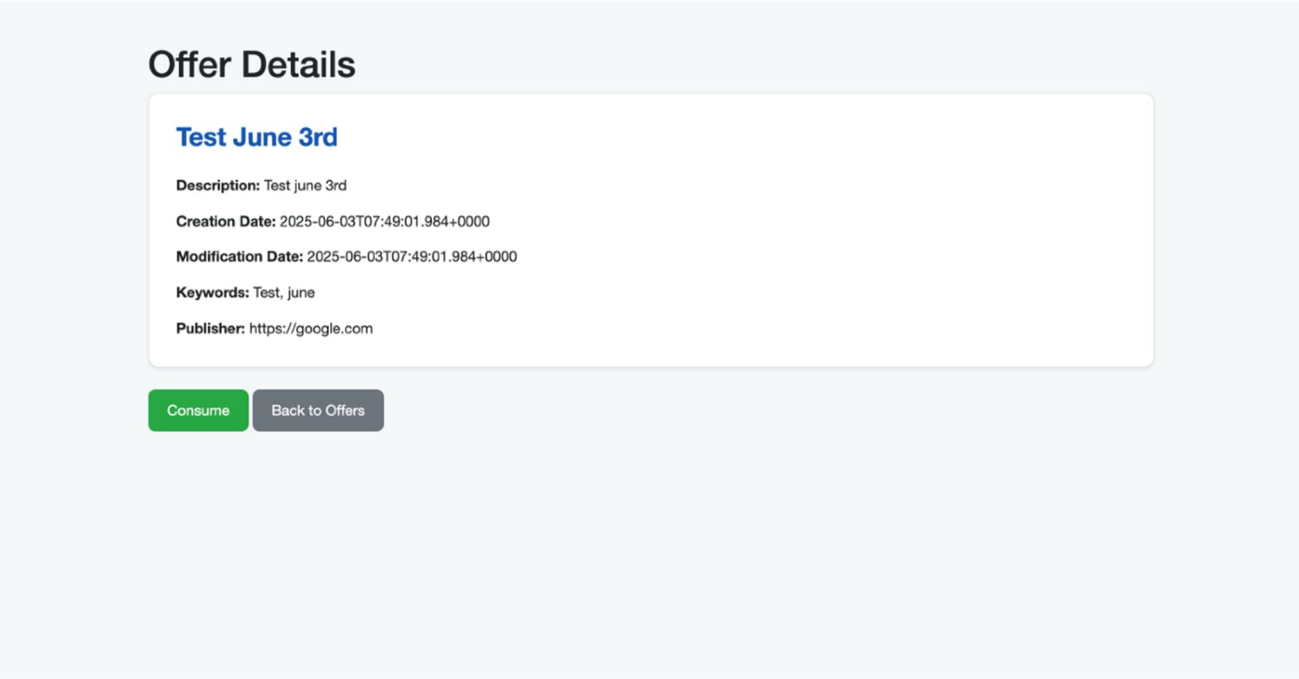
Consumer Interface

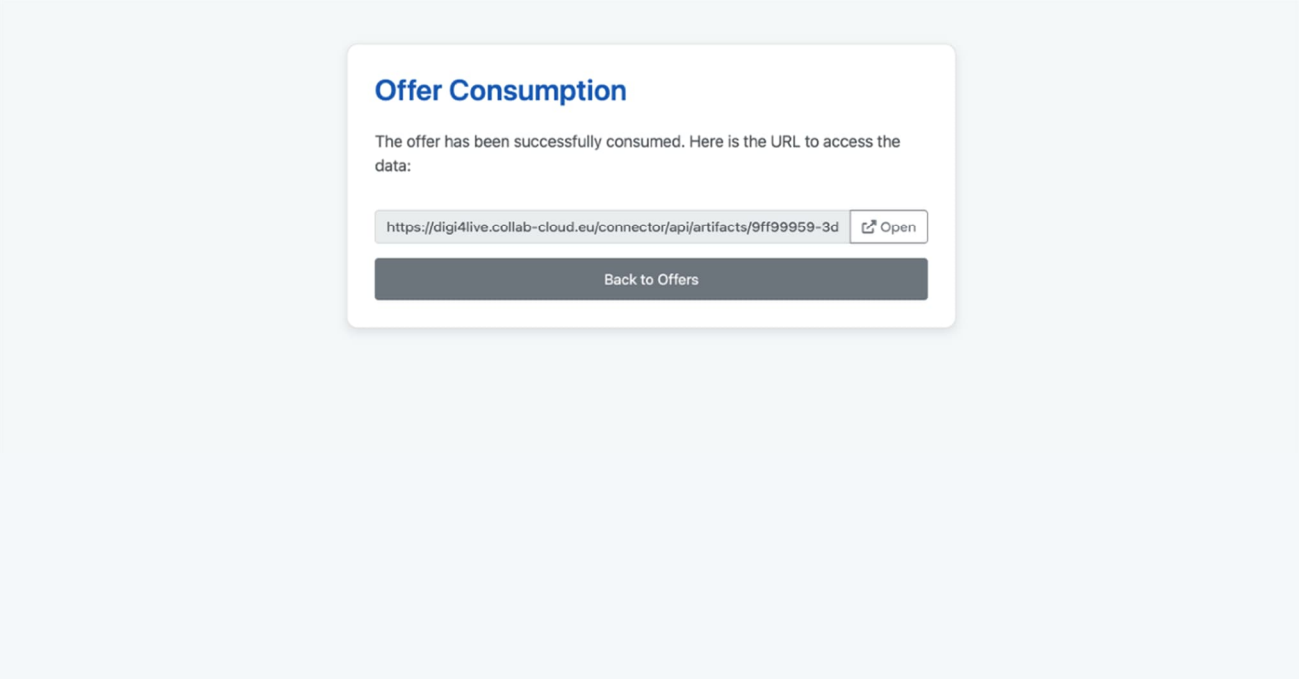
The DSIL Data Consumer Interface provides a streamlined experience for consumers to discover, evaluate, negotiate, and consume available data offerings. Key functionalities clearly demonstrated in the provided screenshots include:

- A clear overview of available data offers, detailing critical metadata such as description, publisher, and keywords.
- Advanced filtering capabilities for efficient discovery and selection of relevant datasets.
- Detailed view of selected offerings, enabling consumers to review additional context and initiate the consumption process through negotiation workflows.

Both interfaces are specifically tailored for clarity, ease of use, and seamless integration with DSIL's backend connector infrastructure, aligning explicitly with the broader goals and technical requirements of the DS2 project.







4.1.3 Primary Feature Progress in %

The following table summarizes the current completion status of key DSIL functionalities as of the M18 reporting milestone, providing a clear overview of progress towards the established project targets:

FUNCTIONALITY	COMPLETION (%)	STATUS
Use Case Definition	100%	Completed
Use Case Alignment	100%	Completed

Data Space Connector	70%	Well-advanced
Data Space Provider Connector User Interface	70%	Well-advanced
Data Space Search	70%	Well-advanced
Data Space Consumer Connector User Interface	50%	Moderate progress
Data Space Deployment on Bare Metal	0%	Not yet started

4.1.4 Activity

The [EXP] is currently in Phase 1 of a structured four-phase development roadmap designed to comprehensively explore and validate all core aspects of data spaces—including providing, consuming, and overarching management functionalities. This initial phase has primarily focused on establishing fundamental components necessary for the secure and effective sharing of data, including both provider and consumer interfaces, as well as the foundational Data Space Connector infrastructure.

At this stage, DSIL has successfully implemented fully Dockerized, web-based user interfaces integrated directly into the Data Space Connector. The Provider Interface facilitates the definition and publication of data offerings through straightforward metadata entry, policy definitions, and support for static file uploads or endpoint integrations. Concurrently, the Consumer Interface provides users with the ability to discover, evaluate, and initiate negotiations for data offerings, laying the groundwork for efficient and intuitive data interactions within the data space ecosystem.

Notable progress has been achieved in the core functionalities, with both the provider and consumer connector interfaces, as well as the search capabilities, reaching approximately 70% completion. The foundational Data Space Connector, integral to secure data exchange and interoperability, also reflects a similar advancement at 70%. Additionally, DSIL has thoroughly completed essential groundwork tasks, including defining and aligning use cases with key stakeholders (both at 100% completion).

Despite these advancements, several challenges have emerged, particularly around maintaining stable integrations with external components such as the IDSA Broker and associated authentication services. Specifically, recurring issues with token validity (DAT tokens) and certificates have necessitated dedicated troubleshooting and proactive management to maintain operational reliability.

Moving forward, Phase 2 will extend the current capabilities by distinctly separating provider and consumer connectors, enhancing clarity in roles and responsibilities within data exchange workflows. Phase 3 will introduce advanced registration management capabilities, streamlining participant onboarding and connector management. Finally, Phase 4 will focus on the full integration and deployment of user interfaces and connectors onto production-level bare-metal infrastructures, offering scalable, secure, and robust operational scenarios.

Ultimately, the completion of these sequential phases ensures a rigorous exploration and validation of each critical aspect within the DSIL environment, aligning fully with the project's overarching goals and firmly positioning both VTT and the DS2 project as leading entities in practical data space deployments.

4.2 DS2 Dash Button – [DASH]

As explained in the IDT architecture document, the Dash Button is a web component that needs to be integrated in all DS2 modules and provide a holistic experience with two primary features:

- Single Sign On: The same authentication and authorisation system is used for both the central and local IDTs and when an organisation is registered in the platform, the same credentials are used to log in the local IDT. This has been a design decision for simplicity but further integrations such as local organisation authentication systems can be provided. In addition, integrating the Dash Button in all module frontends, provides single sign on capability since a user logs in once and can navigate between modules with the same login. To accomplish this, the Dash Button connects to the central authentication system, retrieves a JWT token and then stores it in a local cookie, which the modules can use to authenticate and authorise the organisation. Later, this token can also be used for remote secure API calls.
- Module Navigation: This provides navigation between the different modules, by connecting to the Marketplace and the Global Portal API and retrieving the list of modules to be displayed. This allows a participant to navigate between the different modules installed in a local IDT and/or navigate to the Global Portal and other intermediary services.

The figure below shows the Dash Button with demo user logged in and a list of Core Links in the Global Portal.

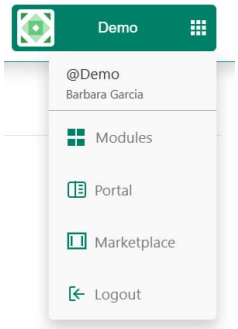


Figure 3. Dashbutton (Top Level)

The next figure shows the Dash Button integrated in the IDT Local Portal and the pop-up dialog for the list of modules and core modules.

A summary of the overall progress and work performed regarding the Dash Button is described in the Portal section.

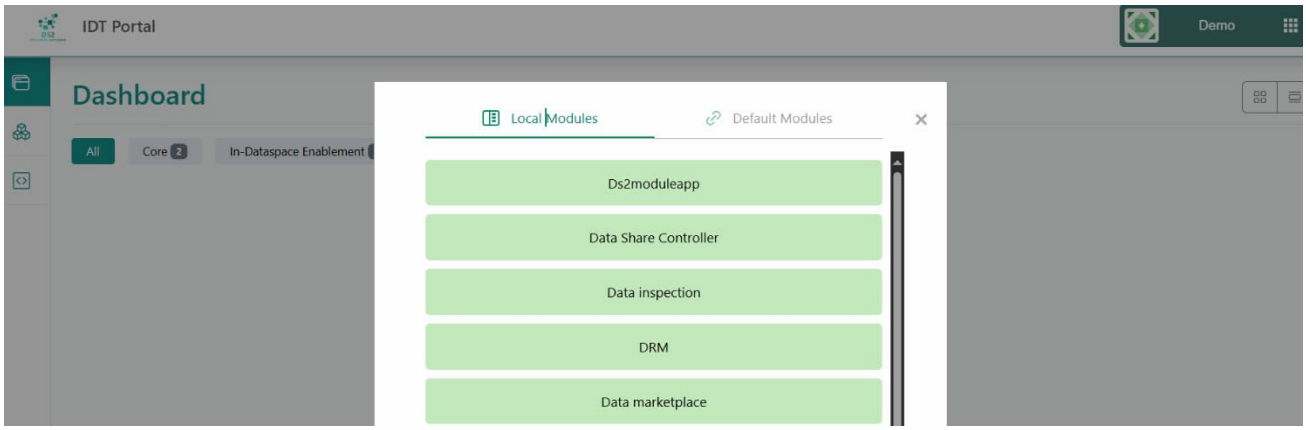


Figure 4. Dash Button (Module Selection)

4.3 DS2 Platform – [PLATFORM]

The DS2 DOA specified that partner i4RI had 54K Euros allocated to them to an outsourced cloud infrastructure, roughly calculated as 36 months * 1500E of project operation. This would be a centralised resource of multiple Virtual Machines. The initial intent was to use this for:

- Setting common systems and methods to benefit developments such as:
  - Software development (Gitlab, documentation portal)– e.g. GitLab
  - Sonar based Quality control
  - Technically validating modules and the systems before user deployment
  - Test and Implementation Servers
- Project operations (management, Owncloud)

Thus, partner i4RI and ICE (also as TM) performed an analysis of DOA/Project needs as well as polling of technical and user partners for their individual needs. For technical partners, this included an estimate of the number of VMs/Resource etc.

It was also noted that:

- In terms of project operation, VTT would use their established Teams/SharePoint facilities, so resources would not be needed for this
- Up until M18 there was little need for usage of resources, except to just establish them, since most development would still be ongoing at an individual module level and that users were only setting up post M18
- Development organisations would normally have similar capacity as part of their normal operations and several committed to this in the proposal submission Part A
- Cloud Server provision was envisioned on a case-by-case basis
- Once working/tested, modules will be deployed on the PORTAL/CONT/IDT environment and thus these modules will probably place the most-heavy demand these central resources (which are in turn developed by ICE/i4RI)
- There is not an unlimited number of resources so expectations will need to be managed in a framework
- The external cost of hosting is always perceived as inexpensive, but with the number of modules and the mandatory need to support the central modules, resources might become limited
- Nearly all requesting partners simply need a Unix environment, but INTU requires a Windows VM – incurring more costs
- What would be provided individually to partners is VMs on a central hosted resource – and not individual servers
- Whilst it was originally expected in the call that use cases would bring into the project their existing DS environments, there would be a definite advantage to both technical and user partners if there were test environment ‘replicating’ use case environments and the modules deployed
- At an operational level, especially considering validation, most modules would be deployed at participants’ locations and thus validation resource demands would be local and not rely on the DS2 cloud environment

With this in mind, it was decided to support, in priority (highest to lowest) order,

- Project:
  - First served is “unserved Project needs “(DOA) e.g. Documentation repository
  - To get an initial block booking (e.g. 50 VMs) and add further VMs individually if needed and DS2 has financial capacity.

- Pay-per-use was not considered as it is complex to administer
- Central needs – the "DS2 Platform" (PORTAL/CONT/IDT/IDM)
  - This can use any excess VMs (up to 25 VM/50 CPU limit) for platform components
  - These modules will have a "TEST" and "PRODUCTION" set of VMs (Potential CDI pipelines)
- "Across Dataspaces with intermediary" if needed (eg SDS, DMK CAT)
- Modules/Technical Partners who stated they would like resources:
  - For modules that asked, to be provided with a VM - A VM of Standard specs for "TEST" VM
  - Tested modules deployed on integrated production environment
- Use Case Test Environment – 6 VMs Environment per use case (If needed but could be limited – see comment above on having existing Dataspaces)
- Given sufficient resource/financial capacity, more VMs could be acquired ensuring a) This framework is still adhered to b) The need for additional resource be re-evaluated once VMs are in real use
- Changes to allocations (extensions/restrictions) would be managed and determined by i4RI in conjunction with Technical Manager

A DS2 Standard VM is defined initially as:

- 4 CPU
- 16 GB memory
- Storage 50 GB
- Ubuntu 22.04
- 1 VM = 1 module by default
- Every partner allocated VMs will need to manage their own VMs with i4RI providing support if possible

Expected Initial Allocation:

Party	ICE/i4RI	DIGI	ATC	INTU	INDRA	Greek, A griculture	Romanian, Cityscape	Slovenian, Green Deal	TOTAL
Documentation	1								1
Quality	1								1
Specific non central Modules	0								0
Integration Server	1								1
Platform 1	1*PORTAL, 1*IDT 1*CONT, 1*IDM								4
Demo/UC4 environment	2*3 VM, 2 DSs each with consumer, provider, central***								6
Module		1 MCL [1 SEC]	1 DRM	1 CLM	Eg PAE, DINS, and DMK				6
User Environment						Not requested but provisioned	Same as UC4	Same as UC4	18
OS	Linux – Don't care	Ubuntu Jammy (LTS only)	?	Ubuntu Windows			Ubuntu		
Spec	Each Standard	Each Standard	Each Standard	Non standard –			No Comment	Each Standard	

Figure 5. Initial VM Allocation Plan

During the start of 2025, i4RI specified the final needs and researched potential providers (Ascens, AWS, Digital Ocean, Azure) also taking advantage of knowledge gained from project ZDZW where a similar exercise was made by a different beneficiary. This ensured that the EU rules for such procurements were respected. The result was that the same hosting provider as ZDZW was selected, aka Ascens, and a contract ("Cloud Data Center 25 VMs" – 849€ per month) put in place on 2025-04-02. Note we have started with 25 VM since we

can dynamically grow this based on real use. ICE/i4RI then explored the initial VM management. The contract then had to be extended due to an access issue ("VP Web 25" – 150€ per month) albeit with functionality and a price still better than other possibilities. Then, once access and use had been fully tested, developers/users were approached in a systematic way with guidance on how to use the VMs. As of month 18, 6 VMs are in use by Partners INDRA, DIGI, ATC, ATIT, i4RI, ICE but this is expected to ramp up from M18-24 as the modules become more stable/complete and the use cases are set up.

Current contract:

- Supplier: Ascens (Telefonica) – Spain
- Current VMs: 25
- Total Price pm: 999€

Current use (sample):

Figure 6. Example VM use

4.4 DS2 Connector – [DS2 CON]

As described in deliverable D2.2, one of the key decisions to support Dataspace interoperability is to use a dedicated Dataspace connector for DS2 and add it to the IDT by default, since Connectors are the main basis for Provider-to-Consumer data exchange. All participants will use the DS2 Connector for any DS2 related transaction and it will be installed automatically with the IDT.

DS2 decided to select the Eclipse EDC Connector which appears the most stable (although still experimental) and supported by the open-source community in the Dataspace landscape. However, it should be noted that 'stable' is a relative concept and the evolving nature of EDC and dataspaces in general has provided some challenges.

The figure below shows the typical scenario for Dataspace data exchange between two connectors.

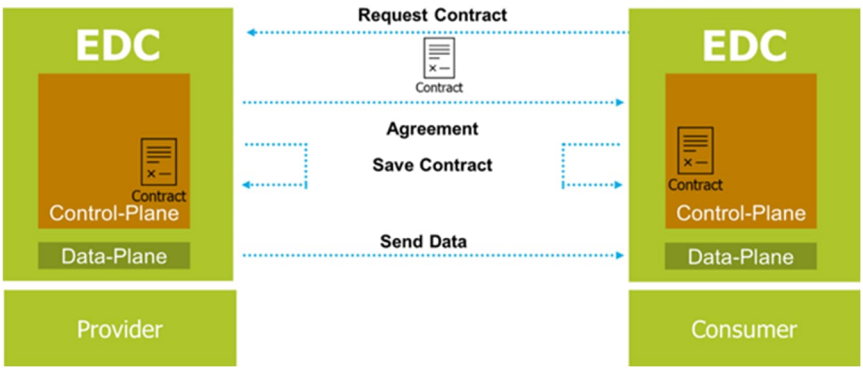


Figure 7. EDC

The EDC Connector is architected so that extensions can be added to create the "final" Connector. In this regard, DS2 and its partners are working towards creating this first version of the Connector by selecting a set of core extensions required and developing the necessary new extensions that each module needs. Several modules have already identified possible extensions and some of them have already started developing them:

Partner	Extension Description
IDT	IDT will package the Connector and all the required extra components (control and data plane, database, etc.) as an embedded module in the IDT
DRM	Log contract agreements in the Fabric ledger and query transaction history from the Fabric ledger
IDM	Support DID and Verifiable Credentials – add new credential types which need to be implemented as extensions
PAE	Rules/Policies validations, obtaining additional data for validation, and implementing actions for enforcement
DINS	Extension to inspect the data shared by the connector
DSHARE	Extensions to control the data exchange and obtain data related stats
MCL	Extensions for streaming data and vast amounts of data

During the first months of development, the EDC Connector was widely tested in the scope of the project by running multiple tutorials from the documentation including the Minimum Viable Dataspace which simulates data exchanges within a dataspace by running two Connectors and the different components required. Meetings were scheduled to discuss the Connector, how to use it, how to extend it, etc. including a development workshop led by INDRA.

So far, the main tests from modules involving the Connector have been using sample Connectors from the EDC repository to test integration and validate different functionalities. Work has also started in defining how the DS2 project will manage Connector development. As said, the Connector is part of the IDT module but will be extended by several partners. To facilitate collaborative development, a dedicated repository for the Connector has been created. The next steps will be creating the first initial version of the DS2 Connector, by selecting the main existing extensions required. Then each module's extensions will be added as pull requests in the Connector repository and newer versions of the DS2 Connector will be released, which will then be added to new releases of the IDT module.

These module extensions will potentially be published back to the open-source EDC community.

5 KPI, RISKS, AND PRIMARY ISSUES

KPI Status:

KPI ID	Description	Status
KPI6.1	Customized dataspace solutions are enabled, and experimentation and proof-of-concept are demonstrated using multiple cloud providers	3+ cloud providers in tests. In progress.
KPI 6.2	Demonstration of connectivity for edge devices from all DS2 use cases to the DS2 ecosystem. Privacy-preserving & access control measures in place	1 edge device/use case part of demonstration. Achieved.
KPI 6.3	Allowing users to pick-and-mix final DS2 tools with both existing and 3rd party tools.	Tools, aka DS2 modules, will be independent allowing them to be deployed where relevant. This is being implemented in the DS2 by ICE/i4R in the PORTAL and via deployment in IDT
KPI 6.4	User trials driven by DS2 use cases. Users confirm functionality and benefits of complete IDT solution.	Not started yet. This KPI is only related to RP2.

Primary Risks (Those in DS2 Risk Register whose likelihood/impact are both > Low):

- "Insecure setup or any vulnerability in the data transfer process can lead to unauthorized access, data breaches, or leakage of sensitive information". Related to SEC component". The project has not reached a state where this risk is relevant.
- "Data Life Cycle (DLC) not finalized in time" which is related to the E2C component. ATC to provide further information on this cycle
- "Complexity of containerisation (largely to tech partners)". Related to the CONT component. Focus now is on individual module development vs deployment/IDT thus the packaging of modules for deployment using the CONT module is more likely to occur after M18.

Primary Issues:

- Moving state of the external E2C connector and DS technology in general
- Lack of DS standards

6 CONCLUSION

In this deliverable, the modules being developed in the scope of the work package and their current status have been presented. In addition, any other significant activities undertaken have been specified.

The main content for the solutions is located in the DS2 GitHub site, whilst the deliverable document contains links to that online documentation and additional information.

In addition, a summary of the WP KPI's current status and a table of Risks detected during the first phase of development have been reported.

Overall, the work package is on track and development is in progress, and all modules are expected to be 100 % completed for the next iteration of the deliverable even achieving extra enhancements. Also, all KPIs are expected to be met and any risks which may materialize will be mitigated.

Within the original DOA the list of modules for WP6 was less than DS2 proposed in the architecture document D2.2, and at this point even more modules are being developed: ATC (PRC Module) and new (expected) partner bluebridgesolutions (DVM) have decided to develop new standalone modules.

The modules are all in a good state considering the planning, but at M18 this state by plan is still initial. The validation use cases of WP7 have been set up in parallel and initial deployment of their facilities and some modules will take place to-plan M19-M24. Formal validation cycles will be from M25-M30 and M31-M36 in an iterative process between users and developers. Many modules are already adhering to the standardized interfaces of DS in terms of relevant UIs, deployment of the Dash Button etc.

Since modules are still in development, then to-plan the documentation is limited but a framework has been made and top-level content added on the DS2 documentation site which will later be linked to the PORTAL/Marketplace.

A hybrid agile-process has been adopted and in the software phase a two weekly sprint paradigm has been adopted with corresponding formal highlight updates and meetings to ensure progress. This process is only documented in the annex of WP6 (this deliverable) but applies to WP3, 4, 5, and 6.

The next step is to continue improving all modules according to both generic and use case parameters. Documentation must be further developed at the user level ("How-To's") and at the technical level (API specifications). Partners must add all modules to the PORTAL/MARKETPLACE adhering to DS2 containerisation strategy and the IDT deployment mechanisms. These marketplace entries need to have commercial and technical data.

In summary, WP6 is on track and expects to exceed its objectives.

Annex A DS2 Online Documentation and Development Process/Tracking

Note: This annex is relevant to, and referred to by, all RTD M18 deliverables – ie D3.2, D4.1, D5.1, and D6.1.

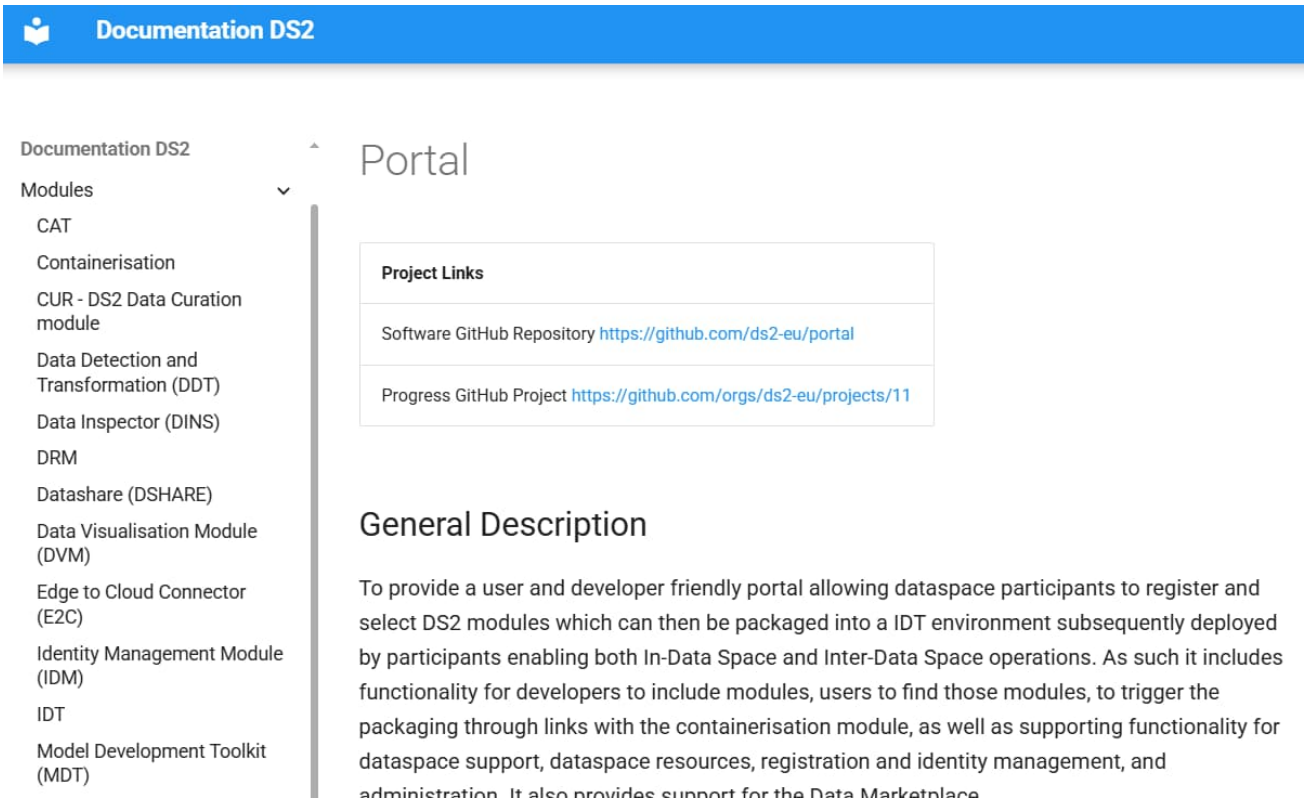
Online Documentation

As stated in Section 3, even at M18 of the DS2, the project has provided support material for every module on the DS2 online documentation repository in a templated approach. This will improve over time but is pertinent now to ensure that users can start to understand, install, and use in the validations from M24 onwards. At this stage, invariably module education and interaction sessions with users will be necessary. Ultimately, this documentation will also be linked to the module assets on the PORTAL's Marketplace allowing any DS2-interested party to explore the details of the modules before 'purchase'.

In summary, the document template contains:

Section	Nature
Software GitHub Repository	This is the link to the latest developed software for the module.
Progress GitHub Project	This is the development progress of the module. See Process/Tracking section below. Note that in the final 'public' version this link will be removed.
General Description	A short contextual description also used on the marketplace itself.
Architecture	A diagram showing how the module fits into the DS-DS environment. Plus, an architecture figure that represents the actors, internal structure, primary sub-components, primary DS2 module interfaces, and primary other interfaces of the module.
Component definition	A description of each building block within the above architecture for each module.
Screen shots	An animated cycle of screen shot illustrating the module for those that have UIs.
Commercial	Table with the organisation, license nature (Open Source, Commercial ...) and the license.
Top features	Business or marketing-oriented features which make the module appealing.
How to install	How to install instructions. Note that as the PORTAL/Marketplace and IDT/CONT are not fully developed, nor the modules, then this is a temporary process which will be partially replaced/simplified by IDT.
How to use	Very preliminary instructions on how to take advantage of the module. Note that given the interim nature of the development these will be enlarged during the next phase of the project.
Other information	Any other pertinent information.
Open API	API definitions/parameters – these will be included when at the next stage of the project.
Additional link	Any links to other related material, video links etc.

A partial extract of an example template (PORTAL) is as shown below:



Development Process/Tracking

The approach specified in the DOA, and so far followed by the development process, is a hybrid-agile approach, i.e. between Waterfall and Agile, influenced mainly from the previous project experience for exceptional software delivery in the ZDMP and ZDZW projects. DS2 is a large project involving many different organizations from many different countries that need to be coordinated and integrated into several development teams to produce the expected multiple software Modules with the required efficiency and quality. To this end, in the early stages of the project, a waterfall-like approach was taken, where a set of requirements were collected from the different use cases then initial Module architectures produced based on the DOA, Technical Partners research and innovation aims, and user “wants” etc – these were summarised in D2.2.. After completing this first Waterfall phase, since 2025-Q1 a more Agile approach, has been used which is described in the next sections including source code management, progress tracking, software testing, pipelines, and Software and accompaniments Delivery.

Source Code Management (SCM)

DS2 uses GitHub as the main software to provide the means for SCM using Git. GitHub is a leading platform for source code management and collaboration, empowering developers, and teams to work together on projects of any scale. It has an intuitive interface, robust features, and seamless integration with Git, GitHub has become an indispensable tool for modern software development workflows.

At its core, GitHub hosts Git repositories, allowing developers to store, version, and manage their code with ease. Whether working on open-source projects, private initiatives, or enterprise applications, GitHub provides a centralized hub for organizing and sharing code, enabling collaboration across teams and communities. GitHub SaaS offering is the preferred choice for DS2 since it provides all the necessary tools for SCM in the scope of the project in their free tier, removing the need to self-host and self-administer, and providing the necessary storage capacity for DS2 source code when it is open source.

GitHub follows a typical structure of organization, teams, and users. DS2 will structure GitHub for both organisations and teams. In addition, the GitHub repository is the place where users can store their code, files, each revision. Repositories can have multiple collaborators, such as individual users or teams and can be private or public. In DS2 a repository will be created per Module with the Module owner as lead.

Beyond use for in-project SCM, Git/Hub will also be used by the the PORTAL/Marketplace so that once Modules are containerised and their details/licenses made available on the marketplace; the downloadable modules are stored on Git.

Progress Tracking

To track the development progress during the project, a combination of GitHub Issues and GitHub Projects are being used. GitHub's issue tracking system is for project management with GitHub Issues being items that can be created in a repository to plan, discuss and track work. Teams can use GitHub Issues to report bugs, suggest new features, or discuss enhancements, creating a centralized repository for tracking tasks and priorities. By integrating seamlessly with code repositories, GitHub Issues streamline communication and help teams stay organized throughout the development lifecycle.

GitHub Projects is an adaptable, flexible tool for planning and tracking work on GitHub. A project is an adaptable spreadsheet, task-board, and road map that integrates with issues on GitHub to help plan and track work effectively. Users can create and customize multiple views by filtering, sorting, grouping issues, visualize work with configurable charts, and add custom fields to track metadata specific to the team. Rather than enforcing a specific methodology, a project provides flexible features that can be customized to project needs and processes.

Usually, issues are created according to the work to be done in a development project, then in order to manage and better visualize those issues, GitHub Projects are used as an Agile main board, typically in a SCRUM or Kanban way.

In DS2, since the consortium is comprised of multiple different organizations which, in most cases, already have their own development processes in place, a slightly different hybrid approach has been chosen:

- Each organization and development team can use their own systems and tools for development and issue tracking although a central one is available
- The “Top Ten” Functionalities are created as issues in the DS2 GitHub per Module repository which should at least relate to/items the steps in the Module architecture templates pointed to in Section 3 of each RTD deliverable. Experience of EU R&D projects has shown that tracking at a deeper level is not very effective due to the research nature of the project
- A GitHub Projects project is created per Module, with the “Top Ten” Functionalities / issues listed to keep track of the overall progress of each application from the management team
- The project is updated in iterations of two weeks (sprints in SCRUM terminology), updating the status of the issues in the form of % of completion, and adding a special Highlight issue per iteration, with a summary of the work performed in the iteration

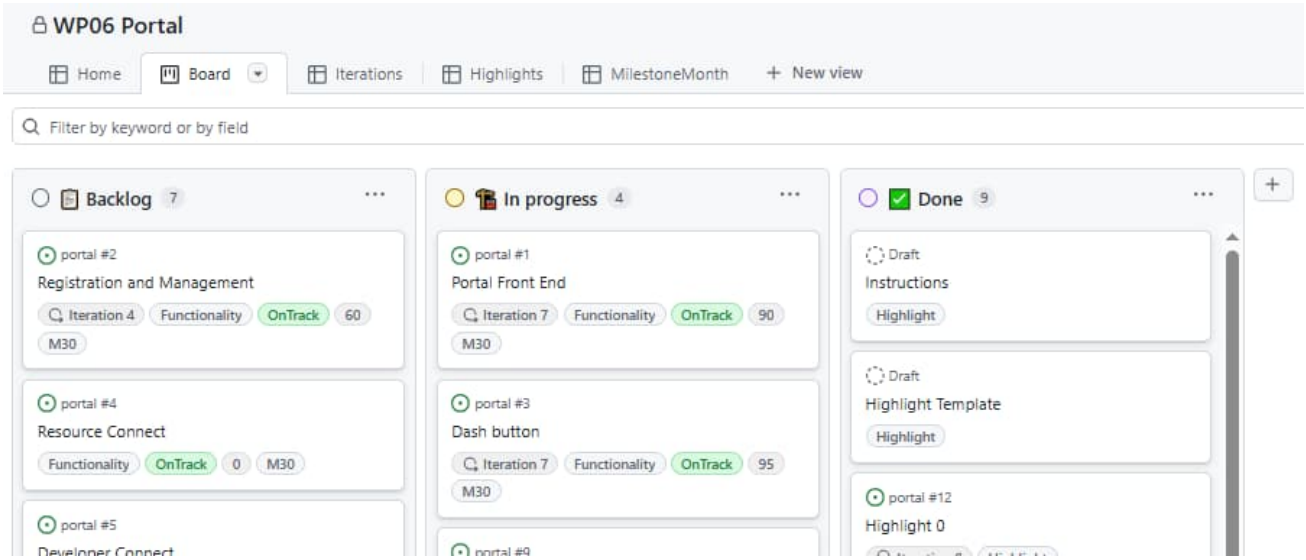


Figure 8. Kanban

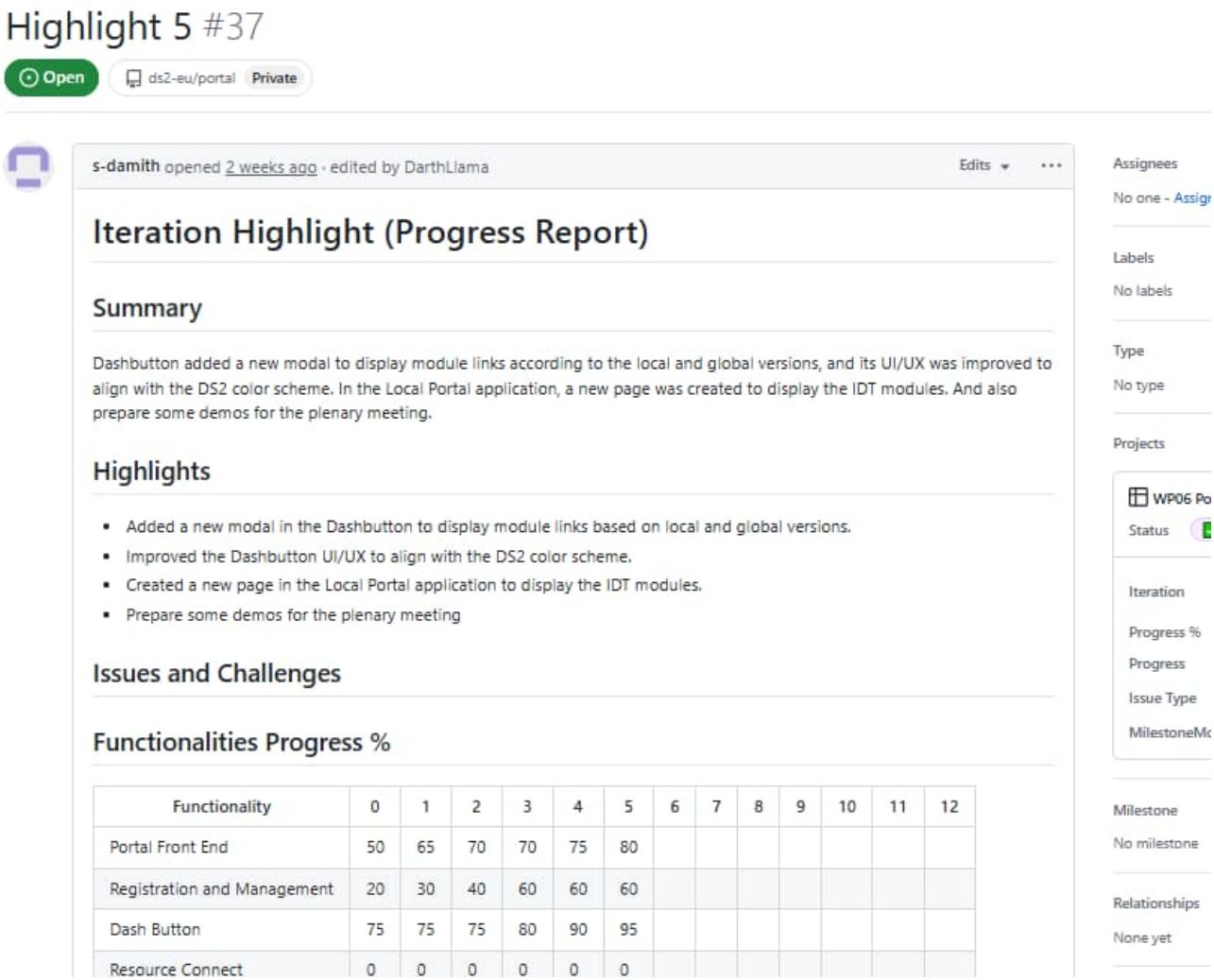


Figure 9. Example Highlight

As can be seen in the figure above, there are three main columns in the project, Backlog, InProgress and Done. The Backlog column is where all Functionality issues are initially listed. The workflow is as follows:

- The module team, linked technical parties, TM, use case partners may insert issues on the board at any point – for examples users during the validations
- Module Development teams work in a set of Functionalities during an iteration
- The corresponding issues are moved to the InProgress column and marked as belonging to that iteration
- When the iteration is completed, the % of completion for each issue is updated. If an issue is 100 % completed, it is moved to the Done column. If not, it is either kept in the InProgress column if the team is going to keep working in this Functionality during the next iteration, thus changing the iteration label to the corresponding one, or moved back to the Backlog if the team is not going to work in this Functionality during the iteration. In that case, the iteration label is removed.
- In addition, the Highlight issue is created with the summary of what has been done in the iteration and the corresponding iteration label as well as listing the % complete expectations at M18. M24, M30, M36 which related to project milestones

This workflow gives freedom and flexibility to the development teams, in the sense that they can keep using their own development control systems but also gives the technical managers the ability to monitor the progress in a straightforward way. Of course, if development teams are willing to fully use the DS2 GitHub Issues and Projects for full Issue tracking of all the work, DS2 partners can also directly see the progress/resolutions.

Software Testing

As part of the DevOps methodology and the CI/CD pipelines, among all the various stages, one of the key stages is the automation of software testing as a step in the CI/CD pipeline. Software testing is a critical phase in the software development lifecycle, aimed at evaluating a software product's functionality, performance, and security to ensure it meets the specified requirements and works as intended. Its primary purpose is to identify defects, reduce errors, and improve overall software quality, ensuring a reliable user experience.

Software testing can be categorized into two major types: manual and automated. Manual testing involves human testers executing test cases while automated testing, on the other hand, uses scripts or tools to execute tests automatically, making it highly efficient for large projects or when repetitive tests are required. Equally, there are several different types of software testing, each with its own focus and methodology, such as Unit Testing, Integration Testing, Functional, Performance, Security, etc.

Other extended form of ‘testing’ include: i) Deployment which is described below; ‘ii) Validation’ which is a user’s analysis of the software delivered against their initial requirements and expectations accompanied necessary feedback for where-possible implementation and adaptation by developers. This will be addressed by WP7. The scheduling of the testing also needs to be conformant with the project phases especially considering deployment and validation as shown in Figure.

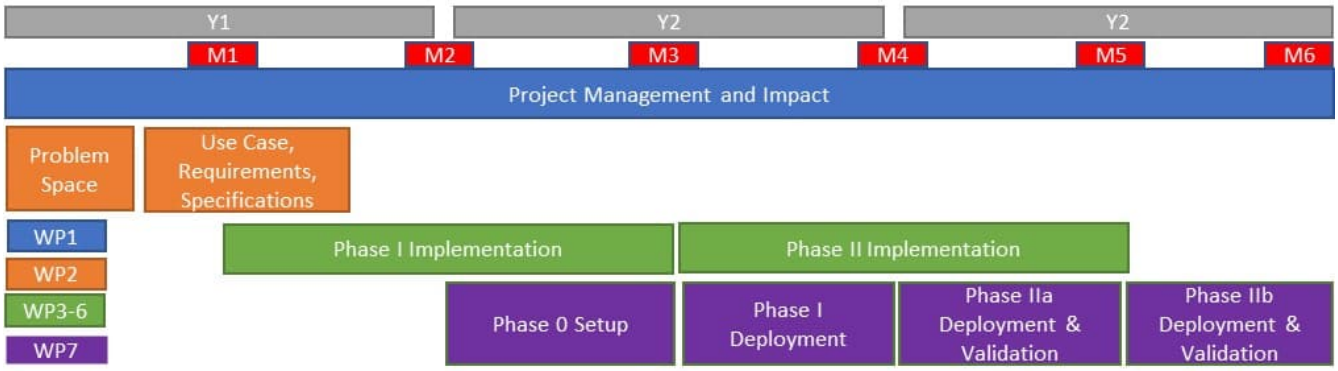


Figure 10. Timeline

In the scope of the project and as part of the automation and CI/CD pipelines, the main type that will be covered in WP3-6 initially will be Unit Testing, and later Security, Deployment Testing and Validation.

- Unit Testing: Is when the smallest part of software, such as individual functions or methods, is tested in isolation. Unit tests are usually written by developers to ensure that each piece of code works correctly by itself. There is a large number of different frameworks in different languages that support automated unit testing. Those tests need to be included as part of the code and will be run automatically as a step in the CI/CD pipelines.
- Security Testing: This will be covered during the vulnerability scanning phase both at the source code and docker image level.
- Deployment Testing: Once the portal is sufficiently active around M24, and Modules are at a sufficient state of development; initial deployment activities need to take place such that by M30 90% of deployment activity is completed including: Producing Helm charts, ensure successful packaging, uploading to PORTAL's marketplace, successful packaging by CONT/IDT, "user" deployment itself and important sufficient documentation to be user deployable. The remainder 10% represents the provision of accompanying information for the portal with must be definitive by M33; for example, the Module marketing profile (price model, images, descriptions) and full usage documentation.
- Validation: Formal user validation on M25-30, M31-M36

Further frameworks and automation may be explored for Functional testing that can be used to validate that the software performs its intended functions as per requirements. These tests check if specific features work, such as logging in, making a transaction, or sending data.

In addition, Integration Tests, which focus on the interaction between different Modules or components of the software, will be manually tested by the corresponding Module developers.

As a summary, these are the different tests initially that Module owners will be mandated to be covered as part of the DevOps and CI/CD procedures of the project:

Test Type	Automation	Who	Where	When
Unit	Yes	Module owners include Unit Tests as part of the source code	Automated as a step prior to build in CI/CD pipeline	Prior to main builds in CI/CD pipeline

Integration	No	Module owners run integration tests periodically to test necessary interaction between relevant different Modules	N/A	Ad hoc based in joint Modules integration plan. It is mandatory to have perform this well before M30
Functional	Not initially	Module owners will run Functional Tests manually. Further research will be performed in order to use automation frameworks such as Selenium	If automated, as part of the CI/CD pipeline	3-6 monthly times around major project points, meetings, and milestones
Security	Yes	All Module owners + DevOps manager will include vulnerability scanning	As part of the CI/CD pipelines	On any release to users during the validation phases. Additional release testing is encouraged
Validation	No	Use Case owners interacting with developers	At Use cases	See D7.1
Deployment	No	Module Owners with PORTAL/CONT/IDT developers	At developers	M24, M30 (90%), M36 (100%)

Software and Accompaniments Delivery

For DS2 users and EU purposes the DS2 Modules are delivered in different phases, synchronous with the high-level Gannt above, and dependent on the phase the software needs to be accompanied by additional material as mentioned in the table below. However, by the end of the project, all should be in place to form a solid basis for post-project exploitation. It is noted that as a RIA project, the material mentioned in the table is more "representative and useful" rather than "final and definitive". This information will be provided to/via the DS2 on-line documentation web page using GitHub and GitHub Pages to publish it. The release of the documentation will be automated using GitHub and Material for MkDocs, allowing for continuous updates by Module partners to reflect the latest version of their Modules in the documentation.

Type	How/ Where	M18 Review	M24	M30	M36 Review
Software	GitHub	To EU/Users– 25%. Installation of some parts with help in their set up for Phase I Validation, Clearly conformant to DS2 visual identity even if only initial elements	To Users– 55% Installation of significant parts with help in their set up for Phase IIa Validation.	To Users– 90% Fully useable and validate-able Module for final Phase IIb Validation. Must at this stage be deployable through PORTAL/CONT/IDT. UI is fully DS2 visual identity complaint where feasible	To EU/Users– 100% Bug fixing, user feedback solutions – NOT development completion
How to Install	GitHub on-line documentation	To EU/Users - Suitable re current software state and NOT considering via DS2 PORTAL Deployment	To Users - Suitable re current software state and NOT considering via DS2 PORTAL Deployment	To Users - Suitable re current (ie Final) software state and MSUT BE via DS2 PORTAL Deployment	To EU/Users. M30 Final version+ plus fine tuning due for bug fixing/feedback
How to Configure	GitHub on-line documentation	Education session with users (nothing for EU despite M18)	To User Suitable re current state	To User Suitable re current state and should include information on other linked Modules that may need to be configured	To EU/Users. M30 Final version+ plus fine tuning due for bug fixing/feedback
"Marketing Video"	For PORTAL	No	No	No	2-3 Minute Intro Video suitable for PORTAL
Portal Helm Chart	For PORTAL	No	No	Yes, must be included	Yes, must be included
Portal Marketplace descriptor	For PORTAL	No	No	Yes, must include significant information	Yes, must include Final information