8 DS2 Orchestration Module (ORC)

8.1 DS2 Orchestration Module (ORC)

Owner(s): ICE
DOA Task: T4.1
Tier: Tier 2
Nature: Optional
Result: K4.1

This task will create a federation mechanism to enable different data spaces to interoperate. This task will orchestrate the lifecycle of data from data collection to data exchange, to data disposal/deletion across a federation of distributed data stores. The data lifecycle will include the establishment of a data contract between the data sources, the establishment of trust between entities in the data flow, and the adherence to data sovereignty and security requirements in the resulting federated data set. Other challenges will be addressed, such as accountability for use of purpose, the propagation of new domain-specific data restrictions (such as policy changes) across the federation, and methods for non-repudiable lineage across the lifecycle. Additionally, topics to reduce the latency involved in the transfer of huge amounts of data, such as caching, and data relocation will be investigated. This task will examine current, emerging technologies in this field, such as work being led by European Data Spaces and from the GAIA-X project as a basis for extension

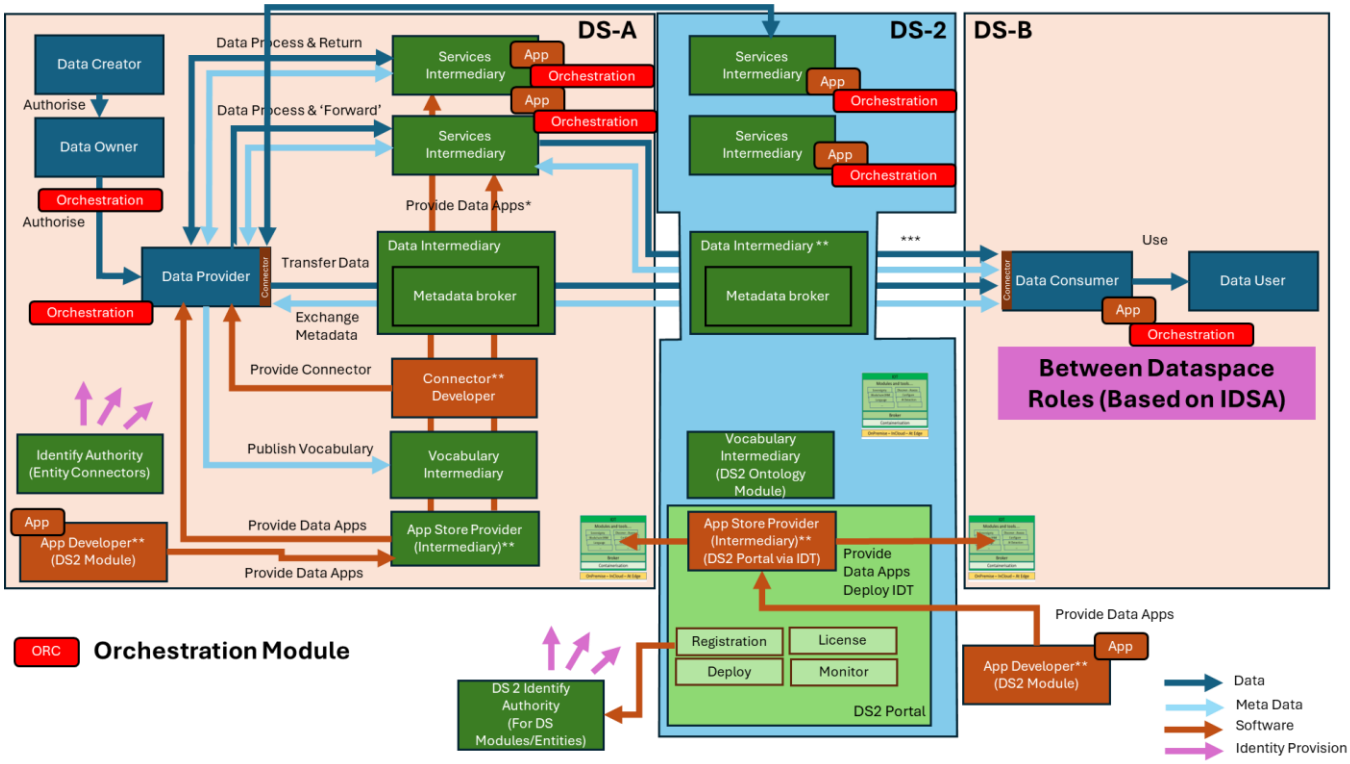
8.1.1 Introduction

Purpose: To design and then orchestrate at runtime In-Dataspace, Inter-Dataspace, internal, and third-party services which facilitate common data-orientated operations such as transformation of data, checks on data, data updates etc. The orchestrator contains a flexible GUI to design workflows and decision points on these services, and run time component to implement the workflow

Description: Services are added to and then selected from a Service catalog from a participant's local service catalog (In-Dataspace deployment), the DS2 service intermediary catalog (Inner-Dataspace), or other available catalog/service knowledge. These services can be graphically linked together to form a workflow and where decision pathways, decision points, and other operators can be deployed to determine the workflow. Error and exit points should be predetermined with defaults ensuring that failures and error conditions allow flows to be closed automatically. One class of operator is for user defined forms for human input but most often the flows contain backend services. In the context of DS2 the operators will, if necessary, be expanded based on novel usecase peculiarities as will the forms designer. Primarily the design interface is orientated around service interconnectivity, but this will be augmented with a data pre-viewer to help interconnect and understand the results of interconnecting data-orientated services. The orchestrator will be available as a module and for interparticipant service orchestration will extend the connectors.

8.1.2 Where this component fits

8.1.2.1 Big Picture



Where	Status
Within a single Dataspace for use between participants in that Dataspace only	Could. This is not piloted/validate in this project but in essence an in-Dataspace version would work similar to the two Across Dataspace scenarios without the need for more complex identity checking
Deployed and used by a single participant to enable the participant in either an In-Data space or Inter- Data space scenario	Yes: NB can still orchestrate external services (eg non-dataspace in other dataspace participants) if user requires and at users (probably necessary) choice can implement DS2 trust mechanisms for service action
Across Dataspaces without Service Intermediary	Yes :Would then mean orchestration modules "liaise" and interoperate with each other
Across Dataspace with Intermediary	Yes. Potentially could be performed by a Service Intermediary [not covered in DS2
Other Comments	Choice dependent on user needs and they are not mutually exclusive

8.1.2.2 Within a single Dataspace (where applicable

N/A

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

To move data from provider to consumer invariably involves a set of processes. From a Consumer perspective this could involve acquiring via the data in the first place, to navigate it through further data-orientated or movement processes (eg transformation,

fusion, filtering), to placing it in their own data store. At the Provider side the processes are similar but in reverse. In this scenario the orchestrator will facilitate this at either consumer or provider independently since it assists a participant in efficiently creating such a process flow connecting data and other services, as necessary. This thus helps enable their part in the dataspace.

8.1.2.4 Across Dataspaces without Service Intermediary (where applicable)

Whilst individual data provider/consumer orchestrations can add value, there may be further opportunities to link process engines already established at either end. For example, a provider may have a set of processes which then fill a staging database and then when those processes are done trigger a orchestration engine at the consumer end to begin a process which extract the data and populates their own database. This allows control by both parties yet still allow for sharing processing operations. The processes (services) need to be selected from each participants' service catalogs so the participant/services need to be known in advance. Note that the process between participants in the same dataspace or differing dataspaces is the same but will be reliant on the use of the DS2 IDT connector.

8.1.2.5 Across Dataspace with Dataspace Intermediary (where applicable)

A dataspace service intermediary can be used host the service catalog of consumer and provider which allows orchestration to be made without especially knowing the participant in advance. The process will operate similar to the use of a metadata broker.

8.1.3 Component Definition

The figure below represents the actors, internal structure, primary sub-components, primary DS2 module interfaces, and primary other interfaces of the module.

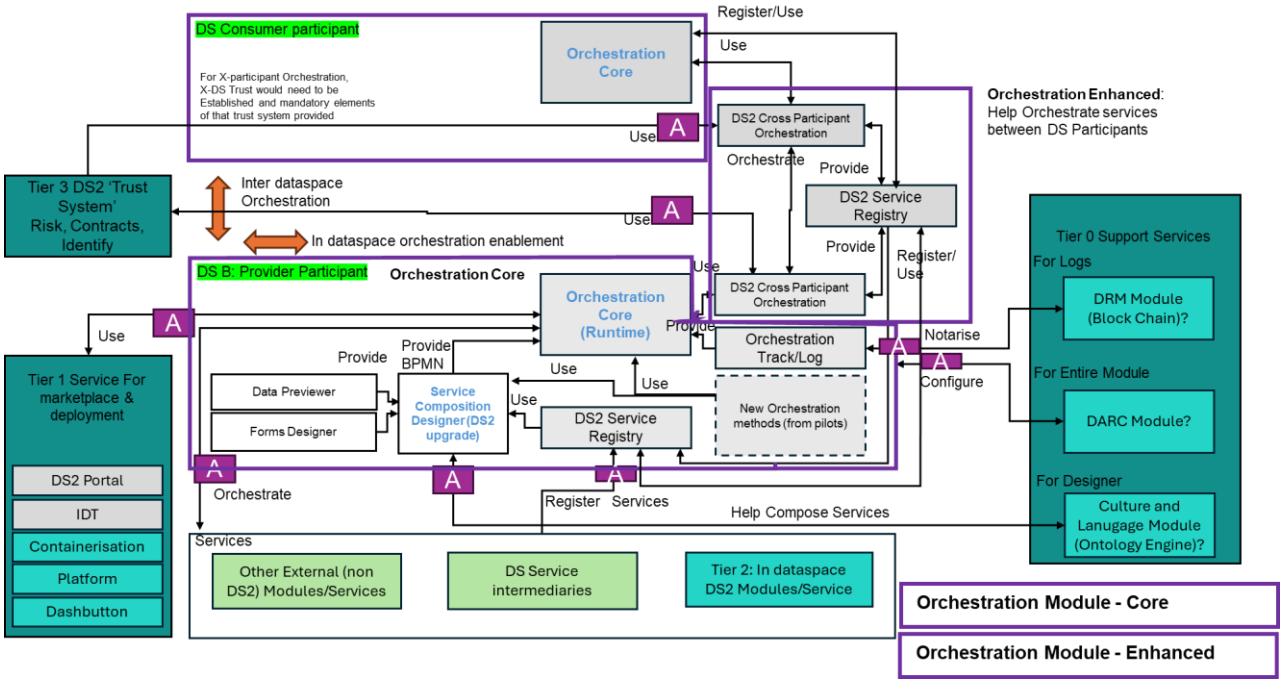


Figure 1: Schema for the Module

This module has the following subcomponent and other functions:

Orchestration Module – Core:

- **Orchestration Core:** This is the runtime heart of orchestration which conducts a process (workflow) triggering other services and via a BPMN module from the Service Composition designer repository. For the tier 1 standard connections (Portal etc) it can be perceived as the entry point. If new orchestration design methods are needed it will use them. Runtime events are connected to the logging components and for inter-participant dataflows it will interface with the DS2 Cross Participant Orchestration subcomponent. This is currently ICE background and will see little development except for a DS2 compliant UI.
- **DS2 Service Registry:**
  - **In-participant:** This is a local registry of all services which a participant may potentially use in a workflow, composed together in the designer, and executed in the runtime. Registration can be automatic in the case of IDT installed services. It will also expose services in the inter-participant DS2 registry. The submodule exists now but will be rebuilt in the context of DS2 and IDT.
  - **Between Participant:** 95% the same functionality but can function similarly to a metadata broker to host services from multiple participants which can be shared in a controlled way to the In-participant registry to allow participant-participant service interactions
- **Service Composition Designer (DS2 Upgrade)**: This is the main UI for the Orchestration Designer based on existing ICE background. It allows a user to select or drag various elements from a toolbox (services/APIs from the DS2 Service Registry, methods), which can be placed on a canvas where they can then begin to start designing their orchestration by dragging and connecting various elements together. The saved BPMN2.0 notation model will then be used by the runtime orchestration core. The DS2 upgrade will be mainly for UI and inclusion of New Data Previewer and Forms designer blocks.
- **Forms designer:** Many orchestrations have a need for user input and whilst some might come from other systems this can be complex when only limited information. The forms designer will allow the easy inclusion of simple form in any Service composition and also ensure that it respect the data flow as well as service needs.
- **Data Previewer:** This is also a new subcomponent which will be rendered via the Service composition designer. Currently services are connected but when designing it is useful to know at design time what might be the inputs and the expect result. In a data orientated project this is especially useful, and this utility will allow some rendering of data to help show flow operations between building blocks before they are deployed.
- **New Orchestration methods (from pilots):** Many methods – eg choice boxes, selections are already implemented in the orchestrator, but it is possible that the pilot might suggest further ones that could be interesting to implement – although at this stage of the analysis it seems there is not. The new methods will be exposed in the orchestrator runtime & designer.
- **Orchestration track/log:** Currently this is rudimentary and especially in the trustworthy context of dataspace an major overhaul is necessary to extract more granular logging information at runtime.
- **Services** and **API:** These are the services that can be orchestrated, and the API block is the interface to:

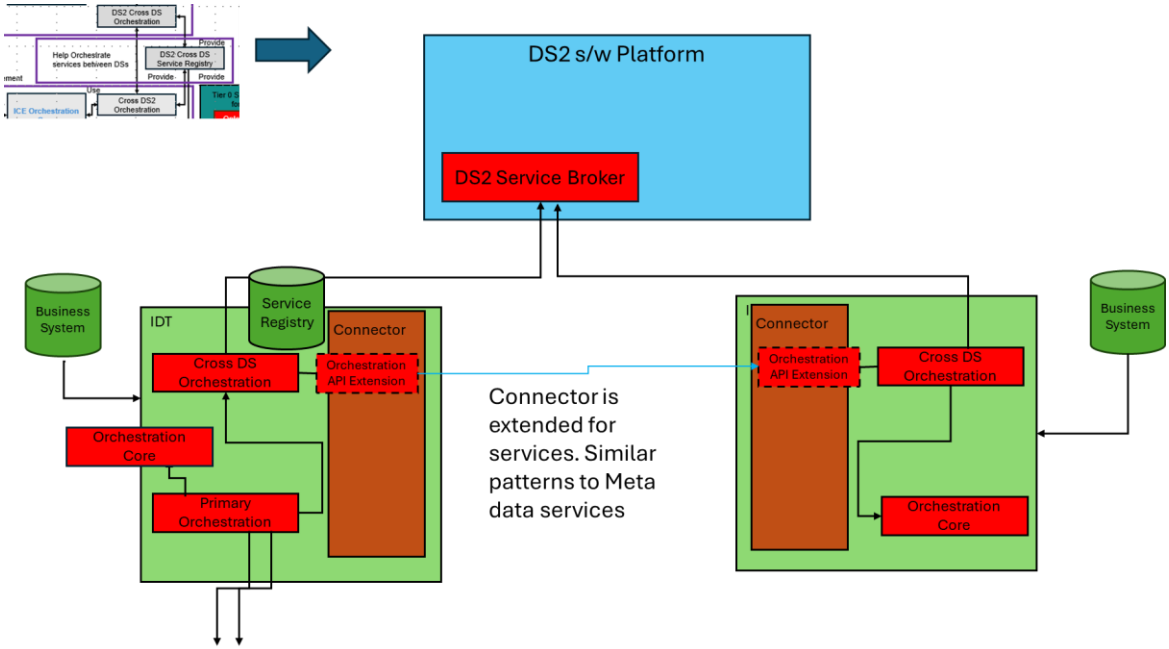
- Other External (non DS2) Modules/Services:
- DS Service Intermediaries:
- Tier 2 In dataspace DS Modules/Service:

- **Tier 0 Support Service Stack:**
  - **DRM** and **API**: For further exploration, but if room to implement and a match of requirements to feature the blockchain part of the DRM module to enhance logging
  - **DARC** & **API**: As with DRM but in this case to use DARC to help configuration of the module
  - **Culture and Language Module** and **API**: As with DRM but in this case to use this modules ontology engine to help auto-link services
- **Tier 1 Service Stack for Marketplace and deployment** and **API:** The full stack will be implemented as generically described elsewhere in this document. Exceptions: The Platform will only be needed for inter-participant service orchestrations if used

Inter-Participant orientated:

- **DS2 Cross Participant Orchestration:** This is a new runtime module which will act as a bridge between the orchestration within each participant through interconnections to the Inter-Participant Service Registry and the Orchestration core at each participant
- **DS2 Service Registry: As described above**
- **Tier 3 Trust Stack and API:** For interparticipant service the module will use relevant parts of the DS2 trust stack – see diagram below

The component for cross participant flows will fit into the IDT/Connector architecture as follows:



8.1.4 Technical Foundations and Background

The foundations of the Orchestration Designer make use of an open-source BPMN modellers framework BPMN-IO, which is a rendering toolkit and web modeller for BPMN. It allows easy creation of BPMN2.0 diagrams using a web-based modelling library, which

can be extended to add the functionality for DS2. The toolbox inside the Process List Manager offers DS2 modules (specifically those that can be run as a service) that are available from the portal. It contains all the standard BPMN modelling elements, such as Parallel or Exclusive Gateways, Processes, and many others that are needed to create a BPMN diagram. The BPMN2.0 Rendering Service can render the diagram whereas the Orchestration Designer can save, retrieve, and annotate a BPMN process as well as deploy it for execution. The execution functionalities are made possible by accessing the public interfaces from the called services. Other relevant ones are Web technologies such as Angular2, NodeJS, JavaScript ES6 and HTML5. The participant-participant connector expects to upgrade the EDC Connector used as a basis within IDT.

Subcomponent/Component	Owner	License
Wasp	ICE	Apache 2.0

8.1.5 Interaction of the Component

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

T	Receives From/ Gives To	What
Portal	Gives To	When published on the portal information (technical, how tos etc etc) will be provided according to the general model
Tier 3	Receives From	Authenticity of participant information in participant-participant scenario
Tier 3	Gives To	Participant identity
Services	Receives From	Service End Point information of Other external Modules/Services. DS2 Service Intermediaries, DS2 Modules/Services
Services	Gives To	Run time triggering of end point APIs
DRM	Gives To	Log/Control information for notarisation
DRM	Receives From	Log/Control information from notarisation
DARC	Receives From	Configuration information
DARC	Receives From	Interactive requests for configuration information and information of how to auto configure
Culture and Language	Gives To	Name and descriptive information from services and their bindings
Culture and Language	Receives From	Ontological alternative suggestions of the given information

8.1.6 Technical Risks

Risk	Description	Contingency Plan
Changes to Data	Should the data change in form or function then the data model will need to be updated accordingly	Make sure the component is aware of changes to data. It will aim to raise a warning if the data has changed in format or schema

8.1.7 Security

Security Issue	Description	Need
Inter-participant trust	Potentially services could be intercepted	Strong relationship with identity system since the service requests may operate outside a connector
In-Dataspace	Where the orchestrator is deployed solely in at a participant there is no specific DS Security risk	N/A

8.1.8 Data Governance

Data Governance Issue	Description	Need
Data Previewer	By-design the data previewer subcomponent will show samples of the data (eg before and after transformation) to help compose workflows.	A mechanism is needed so approved (or open) data samples can be shown
Handling of business process	Business processes may represent confidential information	Secure data transfer of the business process information and of the users instantiating the orchestration
Handling of personal data	This component is not set up to deal with the monitoring of personal data	User/Provider should ensure personal data transferred is transferred according to relevant regulations

8.1.9 Requirements and Functionality

This module will be used in the following usecases:

City Scape ✓
Green Deal ✓
Agriculture ✓
Inter-Sector TBD

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

WHERE	WHAT	WHY	Run/Design Time	Priority
	Use Case 1: City Scape			
Section 2.2 UC1.1	N/A		R & D	M
Section 2.2 UC1.1	N/A		R & D	M
Section 2.2 UC1.1	N/A		R & D	M

Section 2.2 UC1.1	"Sharing and gathering data from multiple sources and sectors"	To orchestrate the data from the sources to one location. To orchestrate the DS2 different modules including the DS2 connector for the data share.	R & D	M
	Use Case 2: Green Deal			
Section 2.2 UC2.1	"Relevant data sources to be obtained from both data spaces within the use case. "	To orchestrate the data from the sources to one location. To orchestrate the DS2 different modules including the DS2 connector for the data share.	R & D	M
Section 2.2 UC2.2	"Relevant data sources to be obtained from both data spaces within the use case. "	To orchestrate the data from the sources to one location. To orchestrate the DS2 different modules including the DS2 connector for the data share.	R & D	M
Section 2.2 UC2.3	N/A		R & D	M
Section 2.2 UC2.4	N/A		R & D	M
Section 2.2 UC2.5	N/A		R & D	M
	Use Case 3: Agriculture			
Section 1.1.4	"Data Integration and Accessibility"	"Facilitating seamless integration" between the DS2 different modules including the DS2 connector for the data share.	R & D	M
Section 2.2 UC3.1	The section 2.1 is quite generic but in general involves the movement of data between different parties and processes	To achieve efficiency to orchestrate the DS2 different modules including the DS2 connector for the data share.	R & D	M
Section 2.2 UC3.2	The engagement of orchestration is similar to UC3.1 except the domain is different – from fruit sourcing to forecasting/crop management	To achieve efficiency to orchestrate the DS2 different modules including the DS2 connector for the data share.	R & D	M
Section 2.2 UC3.3	This is also the same except it is based on Crop Productivity	To achieve efficiency to orchestrate the DS2 different modules including the DS2 connector for the data share.	R & D	M

8.1.10 Workflows

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

8.1.10.1 Register New Service (Local and Intermediary)

This feature provides the capability to register a new Service API in the local service registry to then be able to use it in a new flow. This also allows to create a new remote service registry and synchronize the local and remote services by updating the local one with the remote services, and publishing the local services to the remote registry. The figures below shows the sequence diagram of this feature.

The main steps/functionalities are as follows:

- Fill in New Service Details
- Crete and Store New Service
- Fill in New Remote Registry
- Create and Store New Registry
- Sync Local and Remote Registries

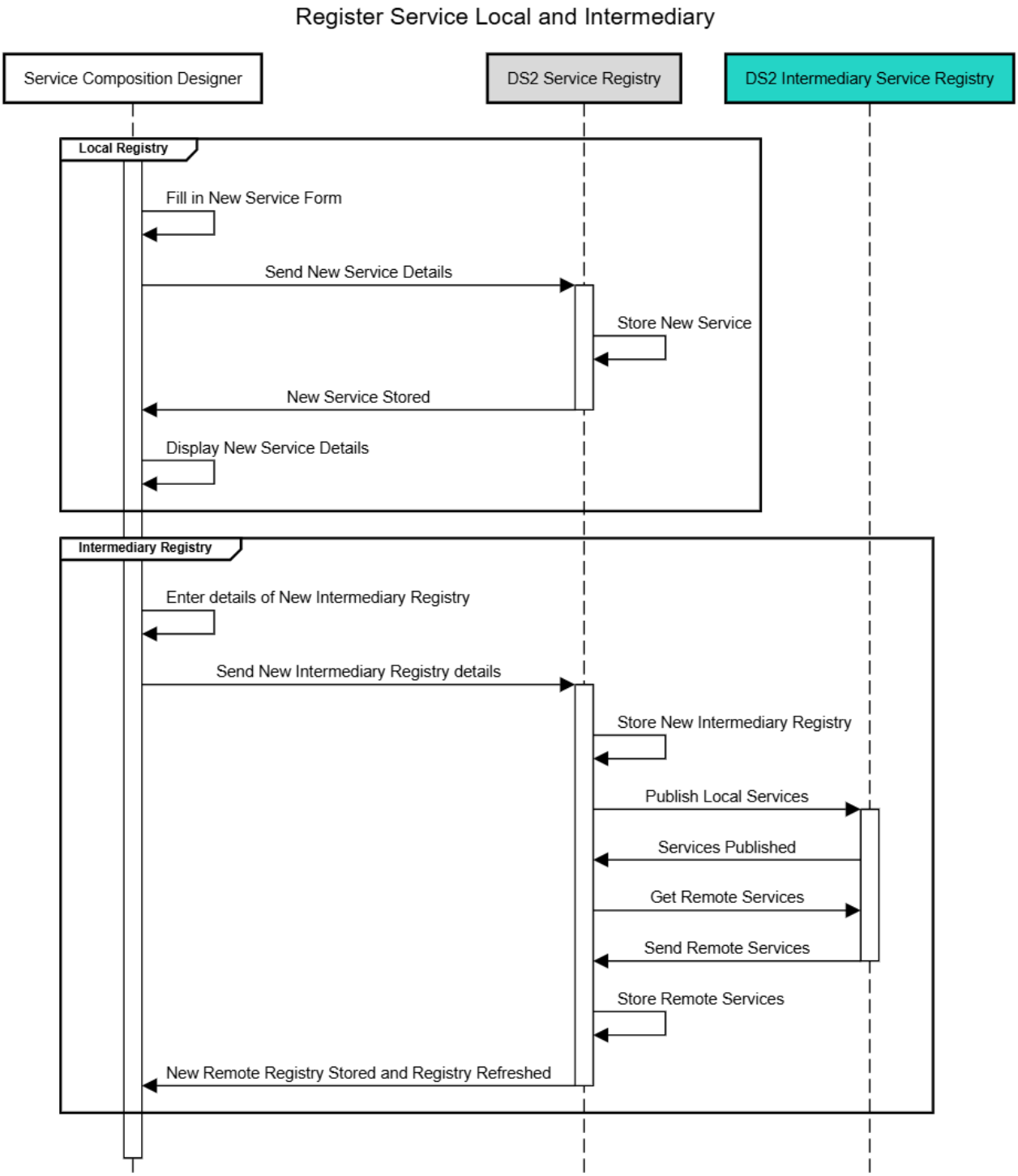


Figure 2: Register New Service Local and Intermediary

8.1.10.2 Design and Run New Flow (Process)

This feature provides the capability to create a new Flow or BPMN Process by composing a number of services from the Local or Remote Service Registry. This allows to use also the Form Designer and the Data Previsualisation tools. Figure 1 shows the sequence diagram of this feature.

The main steps/functionalities are as follows:

- Get List of Services

- Design Flow
- Design Form
- Run a Data Preview
- Store Flow
- Run Flow

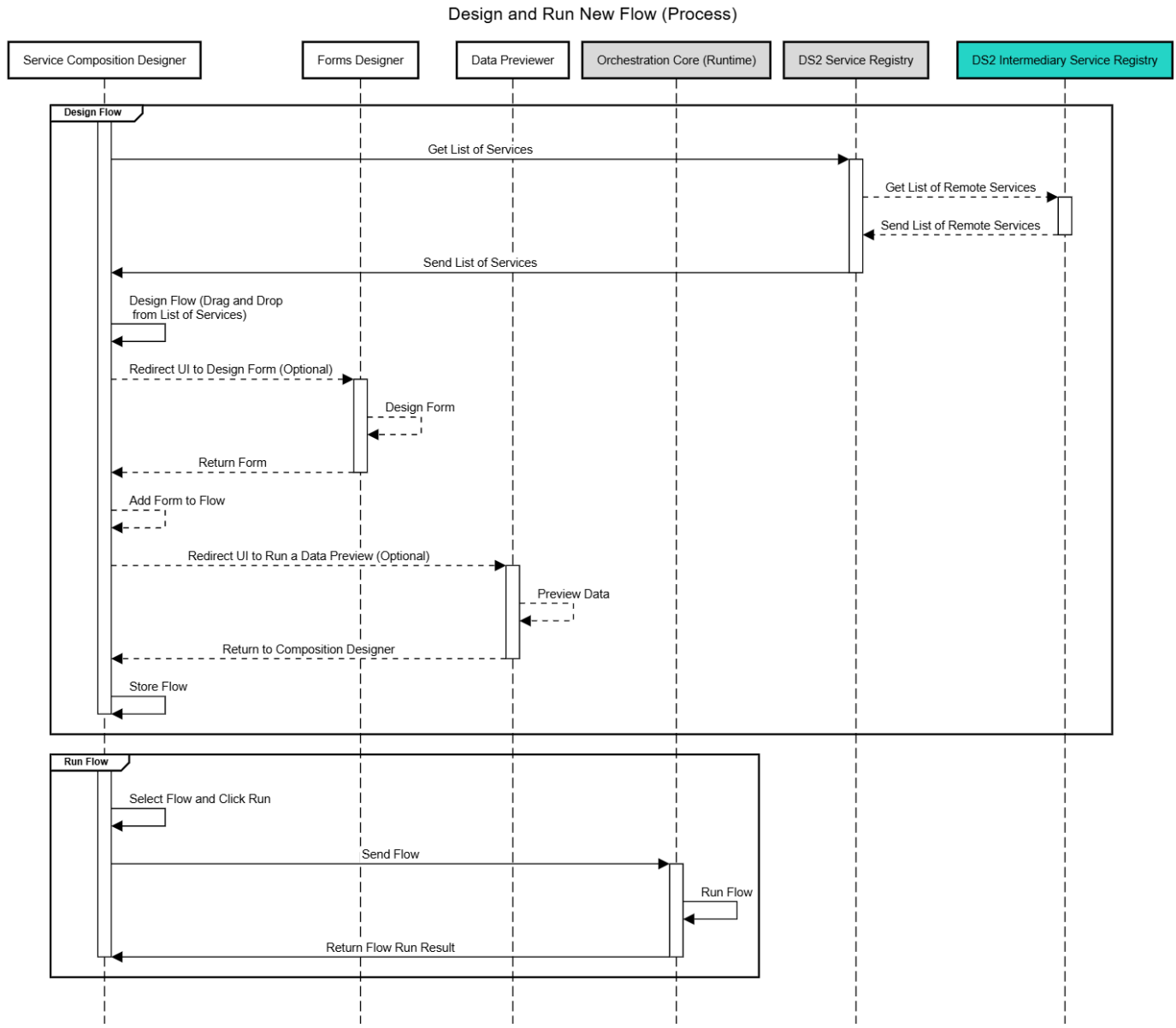


Figure: Design and Run Flow (Process)

8.1.10.3 Interaction with the IDT Connector

Regarding the interaction with the Connector the flow is exactly the same as the previous ones. The Connector API will be pre-registered in the Orchestration local service registry and there will be some pre-defined flows to use it. In order to design and run new flows, the sequence diagram in Section 8.1.10.2 can be used.

8.1.10.4 Participant to Participant Interaction

Regarding the interaction between participants, this will depend on the services exposed in the Intermediary Service Registry. A participant will be able to design a flow using services

from the other participant. For this, again, the sequence diagram in Section 8.1.10.2 can be used.

8.1.11 Role, Resourcing, and Milestones

Sub-component	Main Activity	M18	M24	M30	M36
Orchestration Core (Runtime)	Largely upgrade to look feel				
DS2 Service Registry	Ensuring support of modules and data orientated services				
Service Composition Designer (DS2 Upgrade):	Upgrade for new types of services (modules)				
Forms designer upgrade	Upgrade to respect form approaches perceived in data vs service orientated environment				
Data Previewer	Full Module development				
New Orchestration methods orientated to data flow (from pilots)	Indeterminate until user flows are know				
Orchestration track/log	Significant upgrade to also match data space orientated log expectations				
Blockchain & Blockchain API:	Understanding and implement DS2 block chain module				
DARC & DACR API:	X – Indeterminate – depends on selection of test module				
Ontology Engine & Ontology API:	Further time to explore if a working ontology module could advance ICE orchestration				
Dashbutton	Integration				
Tier 3 Trust Stack & Tier 3 API:	This will be key and new work and new understanding on how to innovate/develop connectors for cross-participant service orchestration				
DS2 Registry Service:	Creation of a communal registry record				
DS2 Cross DS Orchestration	Cross party orchestration by connecting orchestrators at each participant and by utilising IDT				
Table Total/DOA Task Total/Resilience	**Comments:** Blockchain, Ontology are sacrificial. Anticipated no new methods are needed				

8.1.12 Open Issues

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

Issue	Description	Next Steps	Lead or Related Component
Support module interfaces	Final design and at least some Implemented of block chain/DARC/Ontology modules. These are not key/blockers but if opportunity could be useful for the module	The API of the modules will be registered in the orchestration service directory so they can be used as a service.	WP3: Block Chain & WP5: DARC and Ontology
Connector	From theory→research→design→implementation of service upgrade to chosen connector	Further connector research	None
Usecase	Precision is necessary on the flows to see if any additional forms, service methods are need	After discussion with use cases, no additional forms or methods are necessary at the moment.	ICE and all usecases