| | |
|---|---|
| Project Title | FAIR EArth Sciences & Environment Services |
| Project Acronym | FAIR-EASE |
| Grant Agreement No. | 1010587 |
| Start Date of Project | 1/09/2022 |
| Duration of Project | 36 Months |
| Project Website | fairease.eu |

# D4.1 Landscaping exercise: the (meta)data, software, and cloud needs for the data lake

| | |
|---|---|
| Work Package | **WP 4, Interoperability, Integration & Supporting Services** |
| Lead Author (Org) | **Marc Portier (VLIZ)** |
| Contributing Author(s) (Org) | **Charles Troupin (ULiège), Clément Weber (pokapok), David Sarramia (CNRS), Damian Smyth (MI), Erwan Bodéré (Ifremer), Frederic Leclercq (VLIZ), Gwenaelle Moncoiffe (BODC), Ioulia Santi (EMBRC), Joanna Golley (VLIZ), Katrina Exter (VLIZ), Peter Thijsse (Maris), Reiner Schlitzer (AWI), Tjerk Krijger (Maris), Vincent Breton (CNRS), Simona Simoncelli (INGV)** |
| Due Date | **31-03-2023** |
| Date | **11-04-2023 / Edited final version 24-05-2023** |
| Version | **V1.1** |

Dissemination Level

| | |
|---|---|
| X | PU: Public |
| | PP: Restricted to other programme participants (including the Commission) |
| | RE: Restricted to a group specified by the consortium (including the Commission) |
| | CO: Confidential, only for members of the consortium (including the Commission) |

Funded by the European Union

# Versioning and contribution history

| Version | Date | Author | Notes |
|---|---|---|---|
| 0.0 | 12-02-2023 | Marc Portier (VLIZ) | Brainstorm and exploration scratchpad |
| 0.1 | 22-02-2023 | Marc Portier (VLIZ) | TOC and main questions to address |
| 0.2 | 24-02-2023 | Marc Portier (VLIZ) | Completed Unified View,<br>Applied Template,<br>Ready for internal Review |
| 0.3 | 27-03-2023 | Marc Portier (VLIZ) | Completed the "fit to the pilots" and the concluding "draft work plan"<br>Ready for Final Review |
| 0.4 | 31-03-2023 | Marc Portier (VLIZ) | Incorporated feedback from internal review and various contributing authors. |
| 1.0 | 11-04-2023 | Valentin Jay (Neovia), Corentin Lefevre (Neovia) | Final edition for submission |
| 1.1 | 24-05-2023 | Corentin Lefevre (Neovia) | Edited final version to complete contributing authors |

# Table of Contents

# List of Figures



# List of Tables

# Terminology

| Terminology/Acronym | Description |
|---|---|
| AAAI | Authentication, Authorisation and Accounting Infrastructure |
| BGC | Bio-Geo-Chemical |
| BON | Biodiversity Observation Network |
| CTD | Conductivity, temperature, and depth |
| DL | Data Lake |
| DCAT | Data Catalogue Vocabulary (W3C recommendation) |
| DIVAnd | Data Interpolating Variational Analysis in n dimensions (https://github.com/gher-uliege/DIVAnd.jl) |
| EBI | European Bioinformatics Institute |
| EMBRC | European Marine Biological Resource Centre |
| EMO-BON | European Marine Omics Biodiversity Observation Network |
| ENA | European Nucleotide Archive |
| EOSC | European Open Science Cloud |
| FAIR | Findable; Accessible; Interoperable; Reusable |
| LDES | Linked Data Event Stream |
| OAI-PMH | Open Archive Initiative - Protocol for Metadata Harvesting |
| ODV | Ocean Data View (https://odv.awi.de/) |
| OGC | Open Geospatial Consortium |
| RDF | Resource Description Framework |
| RO-Crate | Research Object Crate |
| SOURCE | Sea Observations Utility for Reprocessing, Calibration and Evaluation (https://zenodo.org/record/6319836) |
| SPARQL | SPARQL Protocol And RDF Query Language |
| UML | Unified Modeling Language |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| VRE | Virtual Research Environment |
| W3C | World Wide Web Consortium |
| WP | Work Package |

# Executive Summary

This deliverable has been projected as the long-awaited and definitive description of the Data Lake, that mythical answer to all dwelling technical questions in the whole FAIR-EASE project. The approach presented in this document tries to address that expectation, yet allows itself to do so without conforming to any assumptions derived from common existing definitions of a Data Lake. Instead, it applies high-level systems engineering on the base requirements as gathered in D5.1 and presented during the one-on-one pilot interviews, to conclude in a General Architecture plan that replaces the "Central Data Lake" scenario with a "Many Lakes" one.

Overall, this deliverable is intended to provide a framework for planning the rest of the work and should not be evaluated solely based on how accurately it describes the end result when the project finishes. It is as much a management tool as an actual architecture document, and we hope that it will serve as a useful guide for all stakeholders involved in the FAIR-EASE project.

Funded by the European Union

# 1 Introduction

## 1.1 Process Fit

As a result of the project's one-on-one sessions[1] with the five pilots, we have gained valuable insights into their inputs, expertise and scope. This information has been synthesised in Deliverable D5.1, which provides a factual analysis of the current and desired states of the individual pilots.
Additionally Deliverable D1.2 will be released in the same time frame. It will lay out a project plan applying "Development Cycles" to the identified "Work Zones" (i.e. the main blocks) in this document.
In parallel, Deliverable D2.1 is also underway, focusing on a service description of the various data infrastructures used in the pilots.

This document, Deliverable D4.1, takes a critical step in the FAIR-EASE project development plan to address the interoperability between these pilots. This requires the development of a technical architecture that provides a unified view of the project's supporting infrastructure, and feeds a plan allowing all development work, whether targeted at one pilot or the overall architecture, to be integrated seamlessly.
Deliverable D4.2 will follow up on this document and translate the architectural outline from this document into practical implementation tasks for a number of selected datasets per pilot.

## 1.2 Pilot Interviews

During our one-on-one sessions with the five pilots, we identified two main coarse-level observations that have informed the technical architecture and overall plan outlined in this deliverable. The first observation was the high diversity and lack of uniformity between the pilots in terms of their technology stacks and maturity levels. This observation is neutral and not pejorative. In fact, the lesser-established parts of some pilots allow for a higher level of flexibility and the opportunity to align with and test a more unified view coming from the FAIR-EASE project work. The second observation was a diverse mix of expectations on the data lake (DL), which is the main focus of this deliverable. The expressed expectations for the DL were as follows: it should provide access to datasets, enable sharing of datasets for community and other tools/pipelines, provide download management for large datasets, manage metadata, provide accessibility to the Earth Analytic Lab (EAL), manage user authentication seamlessly, and be agnostic to the location and format of the datafiles.

Although discovery of datasets is a recurring core element of any interoperability effort, it is interesting to note that it did not emerge as a practical need for any of the pilots[2]. These pilots operate in a context where all experts around the table already know the data they want to

[1] These interviews got colloquially labelled as the "one-on-one sessions". They consisted mainly of an open dialogue between representatives of the technical WPs and domain-expert associated from one pilot at a time.

[2] Note: The only exceptions to this statement are (i) the "Marine Omics" pilot, which expressed an eagerness to discover and integrate other sources of available CTD data and (ii) the clear desire in the "Coastal Dynamics" pilot to have newly produced derived datasets (by applying the DIVAnd and SOURCE modules) get to be discoverable in the platform automatically.

work with, and the required datasets have been selected and identified upfront. Nevertheless, discovery remains an essential part of any interoperability exercise to enable access to additional data and services, and support additional users and use cases beyond the Pilots, and we will ensure that our DL supports the discovery of datasets.

On a communication level, the pilot interviews also highlighted the added value of open and direct dialogue. Despite the fact that the various pilots had different goals, ambitions, and intentions, we were able to find alignment between them. These interviews also provided an opportunity for representatives from the technical work packages to enter the conversation as one entity which enabled them to discuss freely with each other and exchange ideas. This open dialogue was quite revealing and helped us to progress towards our goal of interoperability. We will continue to encourage and facilitate these kinds of conversations among the pilots. In a similar fashion we hope that this deliverable will provide an opportunity for the pilots to learn from each other.

## 1.3 Scope and Rationale

This document aims to define an architecture for the FAIR-EASE project as a whole, and finally shed some light on the mythical data lake. It provides a high-level view of the project's features and functions, as well as guidance on how to plan and build these. While some may expect this document to provide a comprehensive, final definition, it is important to acknowledge that we are still in the early stages of the project, and many details and aspects have not been uncovered.

Additionally, the project's diverse pilot activities have highlighted a variety of use cases and requirements for the data lake. To address these needs and accommodate this diversity, we introduce the idea of a "many data lakes" scenario. Rather than focusing on a single, centralised data lake, the architecture is designed to prioritise the functionality that is needed, and to enable pilot-specific optimal decisions on where these are going to be placed. We embrace the principle of "location agnosticism" for the data lake, and propose a list of features and functions that can add value regardless of where those functional components are deployed.

## 1.4 Usage Notes

Throughout the process of developing this deliverable, we have encountered some common reactions and concerns which we would like to address here.

### 1.4.1 High-Level Systems Engineering:

The approach we have taken in this deliverable is based on high-level systems engineering, which may be new and challenging for some stakeholders. Rather than listing and considering every possible detail, we have taken a step back and adopted a broader view that classifies and abstracts many details. This approach does not dismiss the importance of details; rather, it recognises that details are subject to change, and values a generic view that will likely prove to be more durable over time. Additionally, this approach separates concerns and confines details and expertise to a local area rather than spreading them across the board. Ultimately,

the goal is to get the big blocks of the technical architecture and the contracts between them right.

### 1.4.2 Lack of Solidified Contracts:

We acknowledge that this document does not yet provide those clear and concrete formal contracts[3]. Instead, we have provided hints and suggestions towards them. This is due to the short timeframe we had for developing this deliverable. Achieving the level of solidity we need will require a broad validation of the expressed ideas by all partners and further reviewing and tuning in the coming months. However, having this deliverable published in its current state will enable, rather than hinder, that process. It serves as an incentivising invitation to willing participants to collaborate on and contribute to fleshing out the open areas.

### 1.4.3 Required Technical Background:

Finally, while much of this approach would be considered common practice in large-scale computer science projects, it does require a certain level of technical background to fully understand. We encourage all involved to embrace learning from the different areas of expertise that are joined together in the FAIR-EASE project and optimally profit from the opportunity through investing "just enough" into understanding the benefits from each other's expertise.

## 1.5 Readers Guide

Chapter 2 proposes a general architecture for the FAIR-EASE project with a high-level "Unified View" of interactions between the packages that are required to conduct the pilot's research. These packages are then elaborated upon by highlighting and augmenting the relevant concerns.

Chapter 3 revisits that central piece of the document via the various pilots, using those as a further explanation of the approach as well as inviting all stakeholders to review the suggestions in a context most familiar to them.

Chapter 4 looks ahead and proposes several actions and work topics for the FAIR-EASE project to be included in the DevCycles approach that was introduced recently as the work-planning tool.

[3] We consider the actual definition of those to be the subject of future collaborations within the FAIR-EASE project. For the [[Data Provider]] and [[Data Access]] we already introduce some early sketches and suggestions in chapter 9.

# 2 Unified View

## 2.1 Proposed General Architecture

The FAIR-EASE project's technical board has developed a top-level UML diagram[4] to capture and group various concerns into distinct packages with clear interactions and contracts between them.

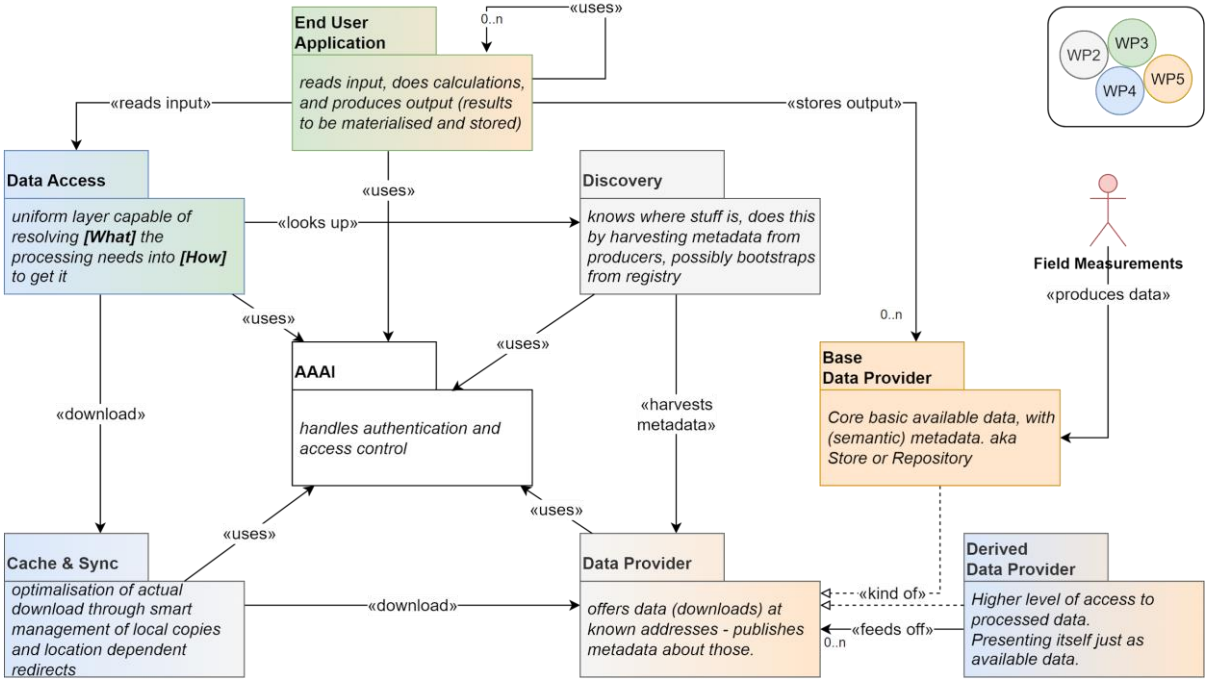


**Figure 1 – UML Package diagram of the Top Level Concerns in FAIR-EASE.**

The architecture starts with the common need to provide [[End-User Applications]], composed of reusable modules. These applications encompass a variety of use cases and will be further elaborated in subsequent sections.

To access any required (and assumed FAIR-published) data, the diagram channels all needs through a dedicated [[Data Access]] block. This layer aims to abstract the implementation details of discovering and retrieving data and allow the application layer to describe the data it needs. For convenience, separate libraries will be implemented for each platform that deliver the data in a readily accessible format tuned for that platform (e.g., as Python pandas dataframes).

[4] The diagram presented here is a UML package diagram, which is commonly used to group related elements together into packages. In this diagram, we use the packages to represent different blocks of functionality in the FAIR-EASE architecture. We have also applied colour-coding to the packages to indicate their association with different work packages. The [[double square brackets]] in the text refer to the labels of the packages in the diagram. More information on UML package diagrams is widely available. See section 8.1 Process Fit for the steps taken leading up to this result.

To perform its job, the [[Data Access]] layer relies on a [[Discovery]] block that aids in finding candidate sources of data. This block will harvest and index information about all known data providers in the FAIR-EASE constellation, which are described and analysed in D2.1. The harvesting process will be bootstrapped from a central registry that contains a managed list of starting points. The [[Discovery]] block's effectiveness will be evaluated based on its ability to precisely narrow down the needs of the data access layer to the smallest complete set of downloads it needs to perform. The level of detail the [[Discovery]] block can obtain from [[Data Providers]] will affect this process.

[[Data Providers]] will expose the data they make available through a standard interface that identifies all available download points of datasets. For each download point, an elaborate information sheet will provide hooks to match the [[Discovery]] results with the nuances and details expected from the requests. This architecture enables reorganising and completing the original datasets on offer from the [[Base Data Provisioning]] through various techniques like indexing, subsetting, aggregation, and additional processing. The additional derived datasets will be exposed to the discovery block through the same [[Data Provider]] interface.

Two additional supporting blocks cover two distinct concerns to complete this picture. The first, labelled [[Cache and Sync]], covers solutions to support the data access layer in the optimal retrieval of the discovered datasets. The second, labelled [[AAAI]], covers the concern of validating and enforcing proper authentication and authorisation to access certain datasets.

## 2.2 Details on the packages

The suggestions we express below are there to help clarify the separation of concerns at the heart of this exercise. Throughout this part we will introduce variants of the top-level diagram presented earlier that highlight and augment each concern.

### 2.2.1 End User Application

The [[End User Application]] package, which contains multiple components as seen in Figure 2, serves as a platform for end users, such as data scientists, to analyse, visualise, and interact with the available data. The [[End User Application]] block contains a wide range of commonly-applied processing techniques to support unbounded data exploration, enabling the discovery of new connections and planned attempts to prove or disprove hypotheses or address new research questions.

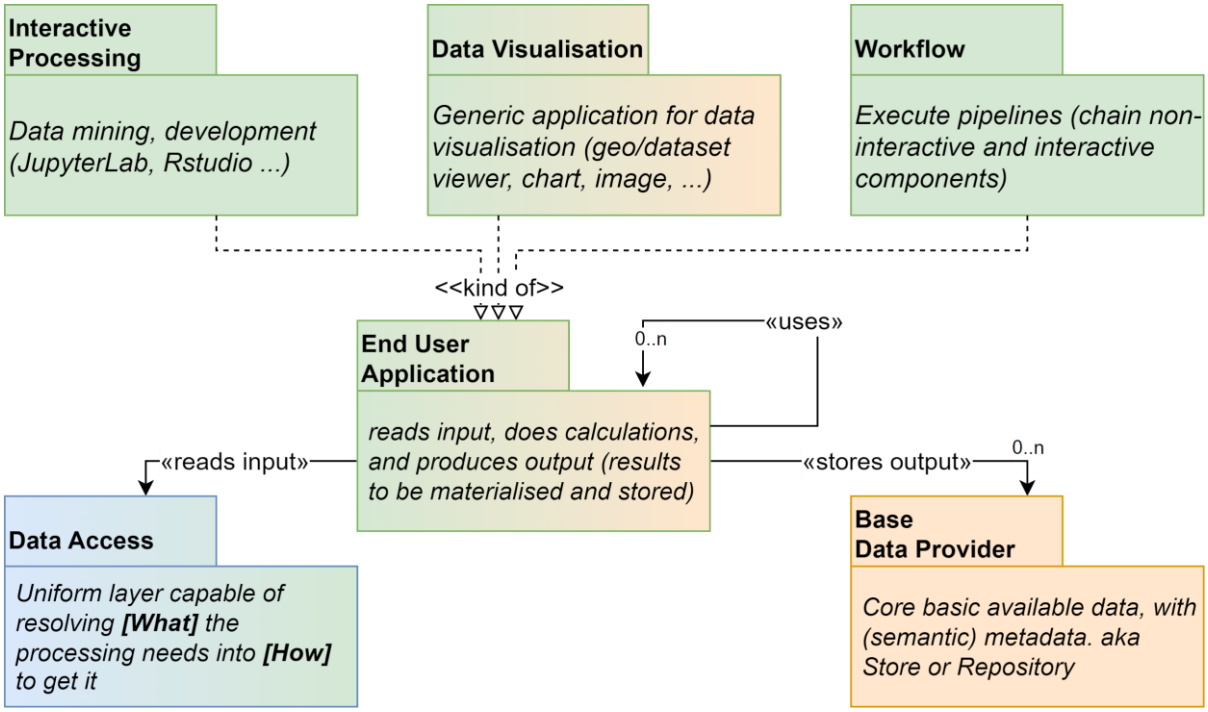


**Figure 2 – UML Package diagram concentrating on the [[End User Application]].**

To ensure independent and durable reuse of the components in this package, there is a cautious limit on the freedom these components should take. This is expressed in the diagram by the blocks that flank the [[End User Application]] package: the [[Base Data Provider]] and [[Data Access]] blocks. The core claim we want to make is that the end-user application components should never take up the concerns of these companion blocks. In other words, the components in the [[End User Application]] package should not be tied up with either the implementation details of how to get to the needed data ([[Data Access]] concern) or with keeping up the service of providing the published end results of their calculations ([[Base Data Provider]] concern).
Guarding these boundaries is crucial to ensure the durable usefulness of these applications.

The work in this area is going to be mainly a joint effort between WP4 (designing the [[Data Access]] interface, WP3 (supporting expertise and infrastructure) and WP5 pilots (achieving their own expressed end goals for developing or upgrading existing applications).

### 2.2.2 Data Access & Data Reformatting

We propose to have a separate module called [[Data Reformatting]] to handle platform-specific data structures and reformatting. This will allow for a higher level of reuse in the data access parts that can remain platform independent and support new formats in the future. By isolating this functionality, we can also relieve the [[Data Providers]] from catering to all possible client-requested formats, as the final reformatting can be handled as a client-side concern.

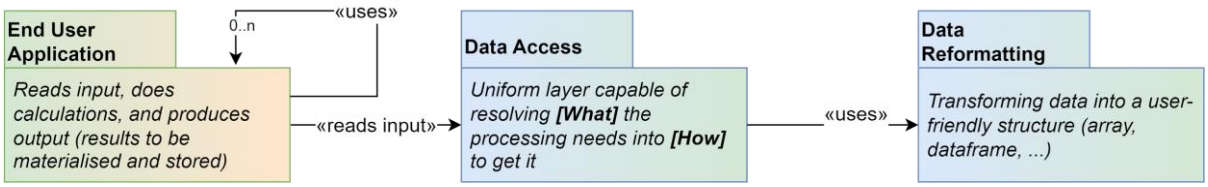


**Figure 3 – UML Package diagram concentrating on the [[Data Access]].**

The main objective of the [[Data Access]] block is to extract the complexities of data discovery, query languages, combinations, joins, merges, and reformatting from the actual data processing in the end-user application. We propose a conceptual "NamedQuery" approach, which is a virtual source of resulting datasets that accepts described parameter values. These definitions should guarantee the semantic outcome of the rows and columns in the resulting dataset they represent.

The "NamedQuery" approach formally describes the functional contract of providing the described results with columns as specified and containing rows matching the provided parameters. The approach allows for diverse and competing implementations to provide the actual code and strategy for knowing and understanding these "NamedQuery" definitions and executing them based on the available data architecture.

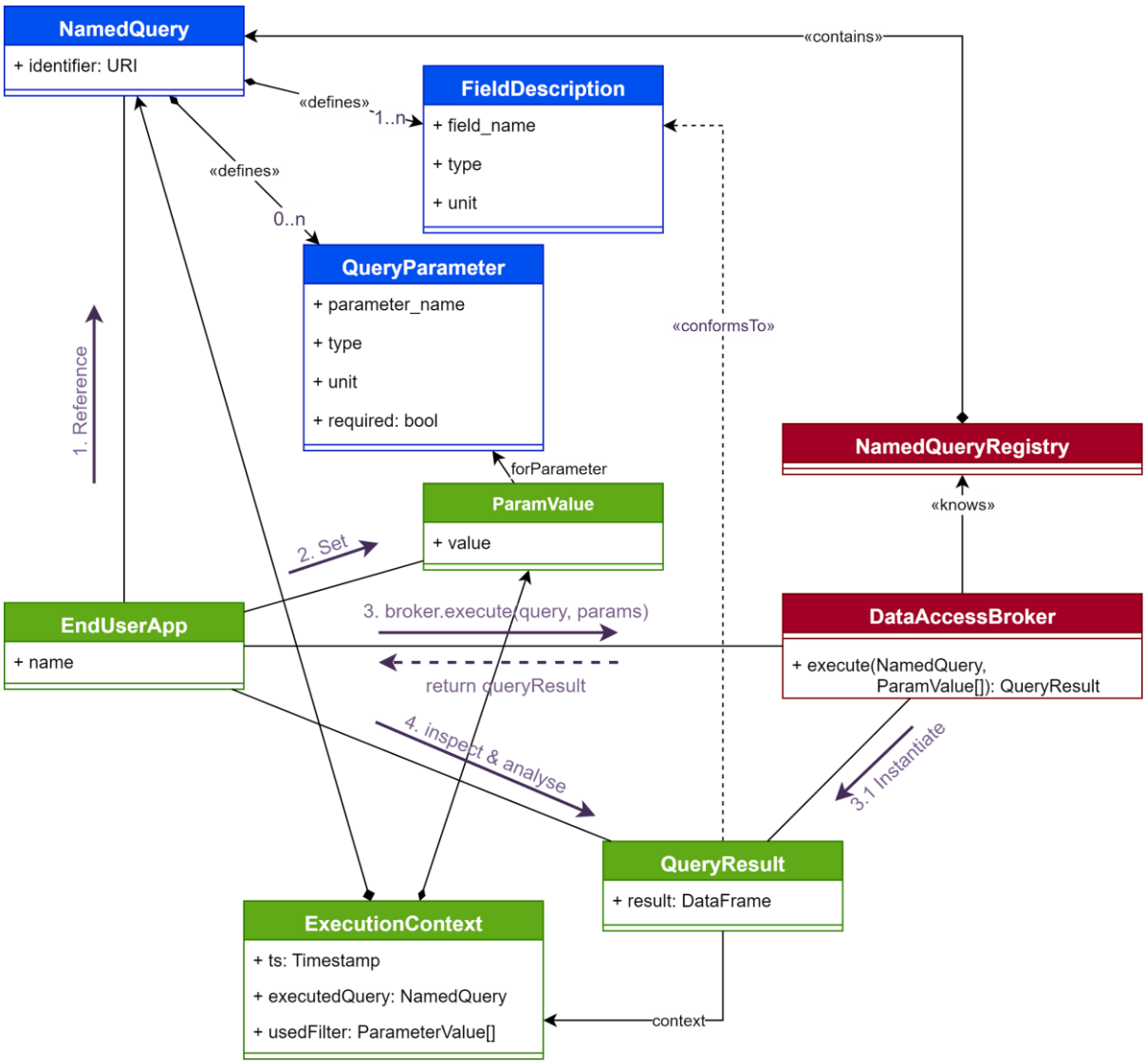


**Figure 4 – Mixed UML Class/Communication Diagram for the [[Data Access]].**

The application of standard software patterns should effectively decouple the [[End User Application]] from these alternative implementations. This decoupling separates the life cycles of the [[Data Providers]] and the [[End User Applications]], ensuring both of their lifetimes to be extended beyond merely their combined use.

In a Python context, this is an example of the usage of the [[Data Access]] module[5]:

```
from udal.specification import
     UDALConnection, UDALProvider, UDALBroker, UDALQuery, UDALResult
from fairease.udal import FE_UDALProviderImpl # FAIR-EASE approach
import pandas as pd

# choose an implementation
```

[5] The prefix "udal" in this example stands for "uniform data access layer".

```
provider: UDALProvider = FE_UDALProviderImpl()

# connect
auth_kwargs: dict = dict(...) # optional authentication settings
connection: UDALConnection =
    UDALConnection("http://example.org/udal-service-provider", **auth_kwargs)
broker: UDALBroker = provider.connect(connection)

# setup the question
query: UDALQuery = broker.namedQuery("http://example.org/23498769")
params: dict = dict(param_name="param_value", …, …)
query.validateParams(params)  # assert fitting data types / formats

# execute and process the response
answer: UDALResult = broker.execute(query, params)

answer.data: pd:DataFrame     # access the data in dataframe format
answer.describe # access the semantics of rows and cols in the dataframe
answer.prov     # access the sources involved in producing the response
```

Table 1 - Code Listing - usage example of the projected access library

Note that the actual implementation of the DataAccessBroker and NamedQuery classes will depend on the specific platform and data architecture being used. This approach allows for flexible and reusable data access and reformatting functionality that can adapt to changing requirements and technology.

To give one example of such adaptability we acknowledge that the above example simply proclaims the target return-type of the results to be a Python pandas dataframe. In reality this is surely too strong an assumption: not only do other table-like memory structures need to play a role, but depending on the case a multitude of various conceptual result-formats should be considered: N-dimensional structures, knowledge-graphs, tree models, images or even existing interactive services that otherwise bypass the model we propose here. The actual result-structure and format too will need to be subject to either negotiation or dynamic introspection.

In this view the NamedQuery contracts are purely conceptual, requiring only a uniform identifier (URI). However, following the common "follow your noise" approach in Semantic Web Technology, we would welcome representations describing these NamedQueries in variants for both humans (text/html) and machines (text/turtle) to support dereferencing. The development and agreement of the required vocabularies to capture these formal descriptions is however out of scope of this document, and even considered as not essential for the FAIR-EASE project as a whole.

This [[Data Access]] implementation suggests an interaction level between the [[End User Applications]] and towards the data-providers. This makes it a field of important collaboration between experts from all WPs.

### 2.2.3 Discovery, Registry, Cache & Sync

It now becomes essential that the [[Data Access]] block can answer two key questions: What are the source pieces of information needed to produce the query result, and what is the most effective way to actually obtain that information?

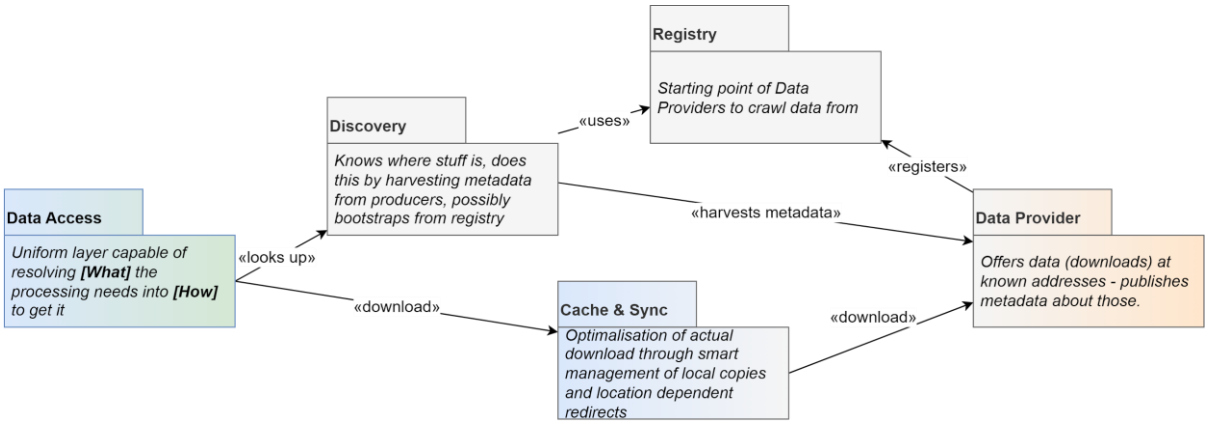


**Figure 5 – UML Package diagram focussing on [[Discovery]] and [[Cache & Sync]]**

The [[Discovery]] block is responsible for answering the first question. By harvesting known [[Data Providers]] (which can be kickstarted from a [[Registry]]), this system can build a "map of the full data landscape" - an overview of the datasets that are available and what they contain. The ability to inspect and interrogate this knowledge will depend on the amount, accuracy, and depth of metadata that is provided about each dataset.

Using the [[Discovery]] block within the [[Data Access]] block is optional, but it is recommended for open and interoperable reuse of data and for increased serendipity in data science. However, it may be bypassed in the short term or as a first iteration, particularly if pilots already know which datasets to work with. In such cases, quick-hack or mock implementations of the [[Data Access]] block can be hardcoded with the pointers to necessary datasets.

Beyond this opportunistic quick start, the ambition within FAIR-EASE should be to tip the scale from "quickly achievable" towards "highly desirable", and explore the possibilities of smarter ways of discovering the relevant and trimmed datasets. Particularly, by exploring this within the scope of specific use cases or pilots we could be exploring richer dataset information models (i.e. more elaborate dataset descriptions) that enable this.

The second question is about the optimal way to access or download the identified information. To address this concern, we introduce the [[Cache & Sync]] block. This system is responsible for optimally introducing and maintaining available copies of the provided datasets to speed up local processing. This will require a mix of approaches that are tuned to

the specific case or type of data involved. However, all approaches should follow a common procedure of resolving the identifier (URI) of the dataset into an actual location (URL) from which it can be downloaded.

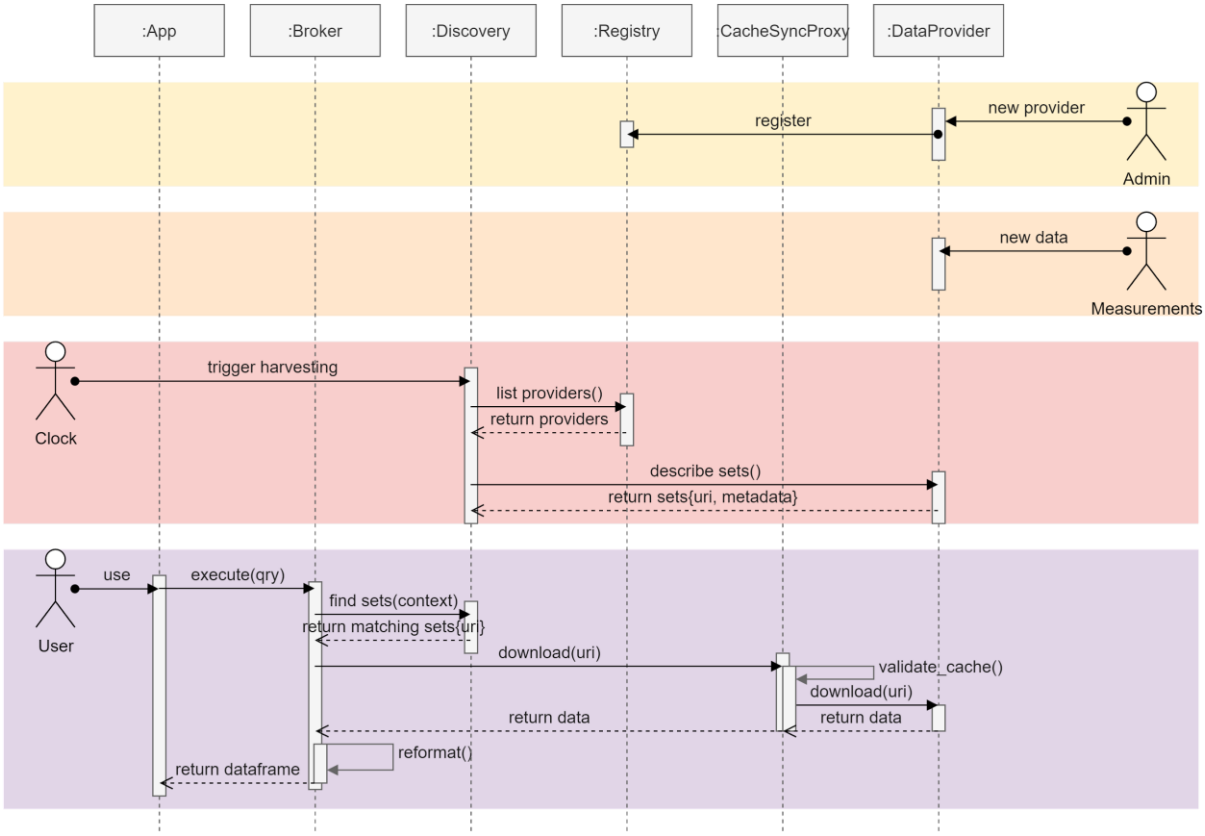


**Figure 6 – UML Sequence Diagram of the phases to get data from [[Data Provider]] to [[End User Application]].**

## 2.2.4 Data Providers

Data providers are an essential component of the FAIR-EASE architecture, as they are responsible for providing the raw data that users will access and analyse. To ensure interoperability and reusability, data providers must comply with the architectural contract specified by the FAIR-EASE architecture.

Our proposed contract includes the following requirements:

- All provided datasets must have a clear identifier, which should be a URI. This URI can be resolved to a URL for downloading the data.
- [[Data providers]] must produce a listing of the available datasets with their identifiers. This listing is the one that will be harvested and should use standard formats like DCAT and include an elaborate "Dataset Information Model"[6] that allows for the needed cleverness in [[Discovery]]

[6] People generally refer to this kind of "description of data / datasets" with the term "metadata". This document tries to avoid that term for three reasons: (i) we want to avoid (possibly unhelpful) prior assumptions associated to the term (ii) the distinction between data and metadata is often a very arbitrary one and (iii) under this

- The Data Providers should allow for effective harvesting by chunking this information stream, ordered by last modification date (descendingly), into separate change-blocks. This resulting "change-feed" or "stream of changes" should be encoded using standard techniques like LDES or OAI-PMH.

To be very clear, this means that the [[Discovery]] (or any DL for that matter) will not harvest or store the data from the [[Data Providers]], rather the latter will be responsible for providing data with endpoints (agnostic to location).

We also acknowledge that clearly not all existing catalogues are going to be fitting this approach simply because we would like them too. Within this reality the FAIR-EASE project will clearly need to balance available resources and pragmatically choose to either simply ignore and work around the misfits, leaving instructions to improve them for increase FAIRness or some future planned inclusion; retrofit or wrap existing systems to close the gaps in an opportunistic way; or even replace or rebuild them with a clear view on a long term strategic vision.

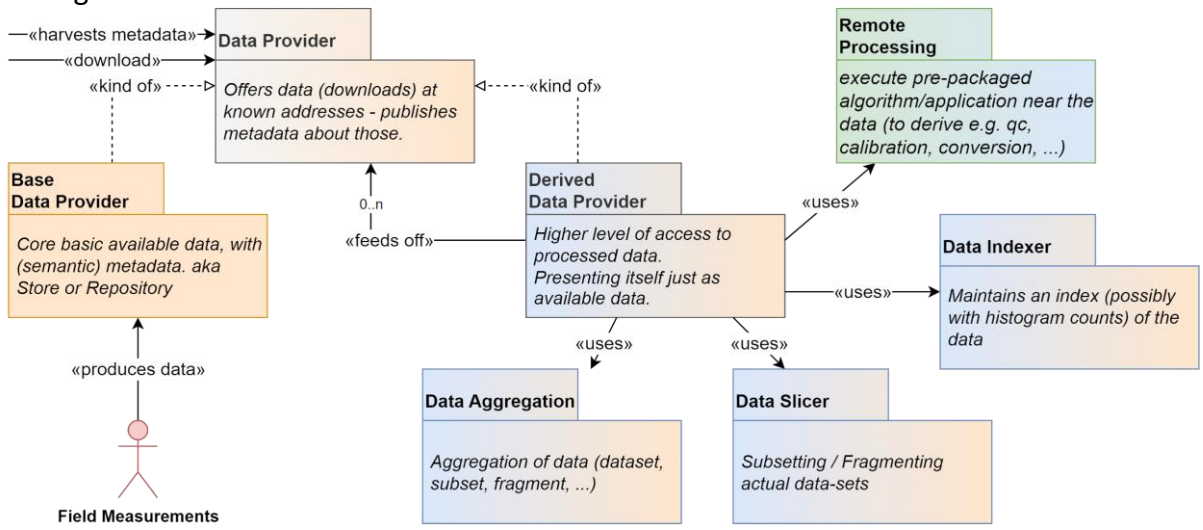


**Figure 7 – UML Package Diagram focussing on [[Data Provider]]**

This approach covers the typical case of the [[Base Data Provider]]. However, we also allow [[Derived Data Providers]] to reconfigure and complete the available base data using a mix of techniques like indexing, slicing, aggregating, and processing to produce new and tuned (i.e. derived) datasets that can cover specific data requests.

To perform their tasks these [[Derived Data Providers]] will feed off existing [[Data Providers]], and do this by performing a similar kind of additional harvesting. This means: they will tune in to the change-feed of these to keep in sync with needed updates, or even become aware of new ones to include. Additionally, unlike the pure [[Discovery]] service, these will effectively also download the actual data in order to perform the processing on it and produce the derived data offer.

umbrella people often either assume or neglect "semantics". This wording hopefully opens the space for a clear and fresh thinking about the what and how of information to include.

To ensure consistency, we propose that derived datasets be made available through the exact same interface that the [[Base Data Provider]] is using. This includes producing a feed or listing of available datasets, having download URLs and offering the linked "data information model" resources describing them. Furthermore, to communicate and describe the existence of a whole range of dynamic URIs that will produce the described data response, possibly on the fly, we suggest using techniques like URITemplates (RFC 6570) and Hydra-CG.

By complying with the FAIR-EASE architectural contract, data providers can ensure that their datasets are discoverable, accessible, interoperable, and reusable by the wider community. This not only facilitates data sharing and collaboration but also promotes a culture of open science and research.

## 2.3 Piecing together the Data Lake

We hope the above explanation shows how the expected functions of the Data Lake are not the superpower attributes of one single entity in the architecture. Instead, its added value emerges from the collaborative effort between a whole range of components;

- not one lake, but many [[Derived Data Providers]] organise post-processed variants of the datasets coming from the [[Basic Data Providers]] taking the opportunity to do that both effectively close to the sources and with an expert knowledge of their content;
- not one central service, but many instances of a common [[Data Access]] layer coordinate the actual lookup, merge and perform the reformatting to produce the dataset needed by the [[End User Application]].

Furthermore, we hope to have presented an initial coarse level separation of concerns that captures the do's and don'ts of these parties to ensure their independent evolution and ensure a reuse of each of them that goes beyond the strict combination described above.

# 3 Landscaping the pilots - fit to unified view

To provide a more concrete understanding of the proposed architecture we want to present how it applies to a selection of topics to be covered by the various pilots. We understand that abstract concepts can sometimes be difficult to follow, so we hope that by presenting these translations to the domain of the individual pilots, we can help bridge that gap and provide a more relatable version of the architecture.

It's important to note that this exercise is intended to explain the unified view and is not meant to be a comprehensive coverage of all the details, but rather a selection of noteworthy elements that emerged from our one-on-one interviews. We recognise that different stakeholders may have different levels of technical expertise and experience, and we aim to provide a shared level of understanding that fits their experience and view.

For each pilot we will follow a recurring outline consisting of an introduction to the selected focus, a UML diagram that reflects those, and explanatory text to further clarify the diagram.

## 3.1 Coastal Dynamics Observatory (Pilot 5.1.1 )

Within the Coastal Dynamics Observatory pilot the aim is to deploy and extend the existing webODV architecture as well as the DIVAnd and the SOURCE tools. Together these cover an important part of the technology stack used in the context of analysing the coastal marine environment near river estuaries. They also put a specific mix of demands towards the DL approach this document is formulating. From that mix we highlight these specific elements:

- The webODV architecture provides a performant data visualisation and analysis solution for end users that tightly integrates with known and locally available community data-sets. It combines (i) a browser based client-module that translates actual end-user interaction into fine grained interaction requests with (ii) a websocket server that handles these in a way that effectively reduces the amount of data transfer by producing those tuned subsets out of the data to satisfy the interactive need. The core requirement to enable this behaviour is fast and local access to the actual files composing the datasets.
- The modules DIVAnd and SOURCE are serving very different goals, but can both be seen as techniques to calculate derived datasets that augment the existing collections with interpolation sets resp. model verification feedback. To optimise their processing, both require access to subsets of the original datasets. Additionally the natural expectation for both is that the produced derived datasets become discoverable (and available) for further usage.

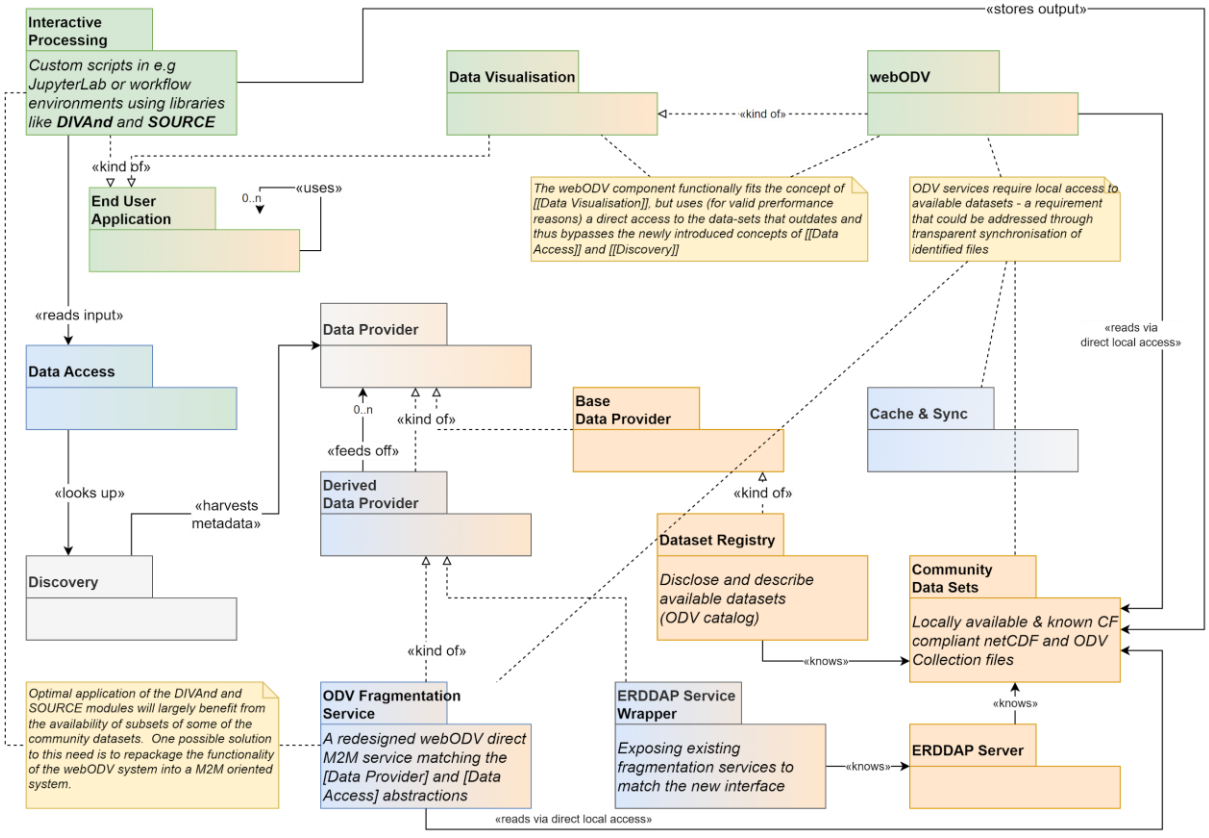


**Figure 8 – Fitting the Coastal Dynamics Pilot to the Unified View**

The above UML diagram suggests how these particular requirements can be fitted to the abstractions provided in the unified view.

- The extra datasets that come out of the interactive processing applications (driven by DIVAnd and SOURCE), should be made available to some internal [[Dataset Registry]]. This is essentially a catalogue of available datasets that not only can learn about new available additions, but also adheres to the [[Data Provider]] contract to make them harvestable by the [[Discovery]].
- The need for subset variants of the base files is often addressed by the community by deploying ERDDAP servers. These also can be made to fit the [[Data Provider]] contract by providing some 'wrapping' service that is capable of interpreting and converting the available metadata and or configuration on these services to practically expose the various subset URLs they provide and the meaning of the parameters they accept.
- Interestingly however, during the discussion of this new approach with experts from the pilot, we also identified that the current webODV solution is in fact internally performing a very effective and tuned variant of precisely this subsetting activity. From this observation came the consideration to provide the needed subsetting through a smart reorganisation of the webODV architecture. Its goal is to provide this core functionality not to cater for end-user interaction but rather to satisfy stateless machine-2-machine requests.
- Addressing the need for fast local access to the data files is something that fits the concern identified in the [[Cache and Sync]] block.

The above observations obviously leave out a lot of important details to address to ensure a balance between desirable extra interoperability and reuse and practical performance. Here is just an initial set of aspects that will require further investigation:

- A new and extended dataset-description scheme will be needed to cover the inclusion of subsetting information, covering how various subset-URLs are connected to each other as well as to their base set, allowing optimal discovery and selection of the best fit for any specific case.
- An investigation of content and completeness of ERDDAP catalogue metadata to achieve these goals.
- Concerning the caching of subsets that could be based on real-value parameters it will be useful to foresee a reasonable 'discretisation' of available subset-boundaries. The net effect of such an approach will be to have a server side decision on available 'buckets' comparable to "tiling" in a geo-based context. Making this work in practice will require a combination of techniques: canonization of the URLs even if they carry multiple parameters, a further expansion of the schema to describe these aspects towards [[Discovery]] and [[Cache]], some service-request metadata that effectively negotiates the applied boundaries in the response, and finally practical testing and tuning of the optimal settings.

## 3.2 Earth Critical Zones Observatory (Pilot 5.1.2)

For this pilot, we observed the importance of capturing regions of interest in the end-user application (portal) to drive the pre-calculation of required geospatial and temporal data-slices, which are consumed in visualisations and calculations within the portal to produce indicators for those regions. The heavy use of GIS-specific access and service protocols in this context serves as a reminder that our [[Data Access]] block cannot be viewed as solely serving semantic data-structures to be freely interpreted by any processing client. Instead, result delivery needs to describe and negotiate responses that remain opaque and can only be tunnelled to dedicated client modules that process or visualise its content.

Another significant observation from this pilot is the specific optimization of download to navigate build-up indexes to find the subset ranges actually needed. Figure 9 shows how the ECZ pilot fits into the unified view.

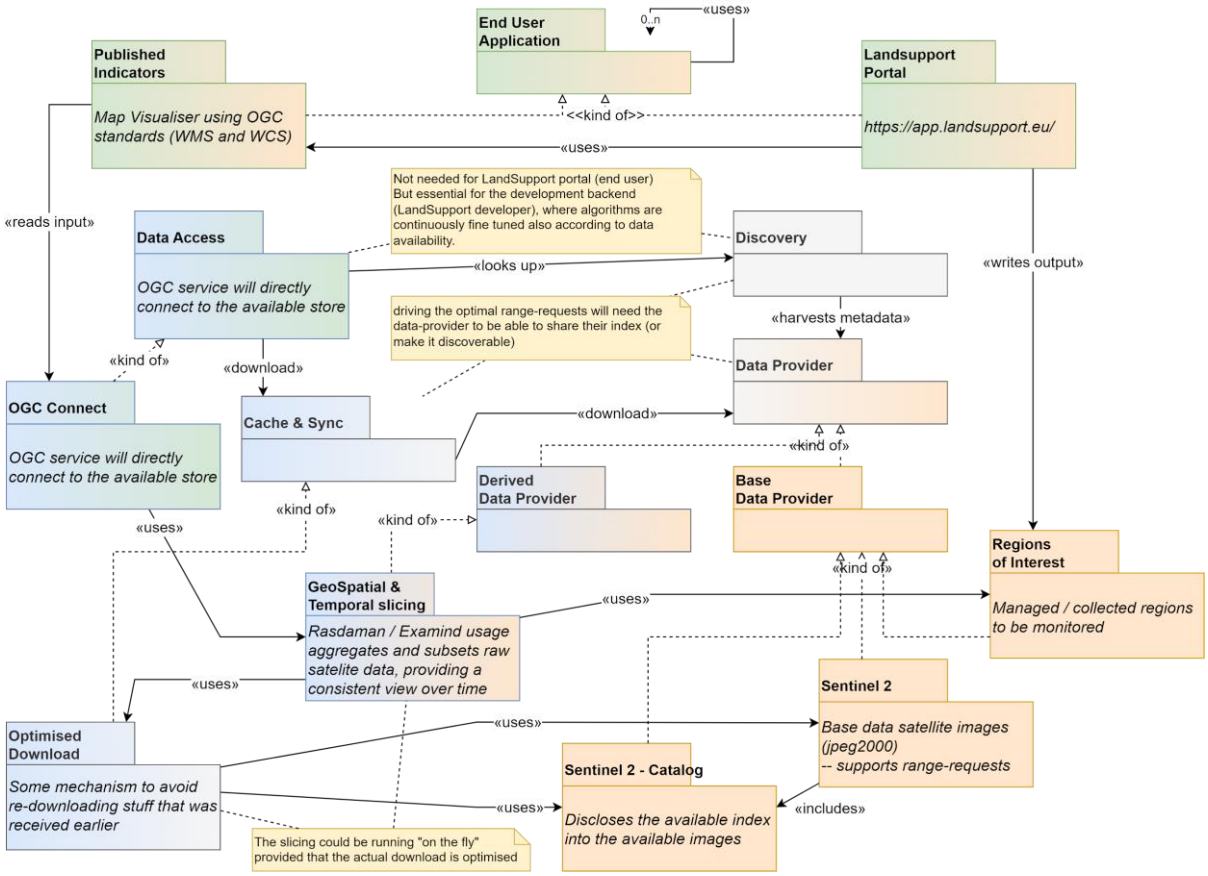


**Figure 9 – Fitting the ECZ Pilot to the Unified View**

The tension between leveraging available protocol stacks that efficiently handle opaque and dedicated formats end-to-end and true open and semantic data disclosure to enable new data pathways, innovative reuse, and interoperability into new contexts is a classic force to balance in open and FAIR-data projects. The reality of this pilot highlights that this challenge grows with data sizes, limiting the solution space to options that all feel suboptimal.

One approach is to tunnel opaque datasets between a limited number of endpoints that know how to deal with them. The other approach is the more difficult task of disclosing and processing semantic details in the entire chain, unifying further the common handling of all data being exchanged. However, a positive interpretation of this situation recognises the opportunity to evolve from one approach to the other by adhering to the introduced design contracts. Gradually decoupling the tight relations between current data formats, the protocols to serve them, and the platforms that consume them can be achieved through the separation of concerns.

Another important aspect highlighted by this pilot is the extension of the data-catalogue scheme to cater for exposing new subsets and the custom indexes built to navigate them. This requires range-request-information to complete the strict addressing (URL and parameters) information that we have considered thus far. Without this extension, the full benefits of the approach will not be realised.

In addition, the pilot experts expressed concerns about the licensing of the tool used to produce the datacube and indexes. The lack of readily available open alternatives is indicative of the tight chains and interconnected components that currently characterise the ecosystem around these data formats. Introducing decoupling interfaces will require effort, but the current costs and limitations make it a necessary step towards a more open and interoperable data ecosystem.

## 3.3 Volcano Space Observatory (Pilot 5.1.3)

Unsurprisingly, this next pilot, which uses similar datasets that are equally large, specifically formatted, and condensed, reaffirms many of the observations made in the previous section. The following highlights from this pilot further explain and complete our unified view, as captured in the diagram below:

- The end-user portal will use a list of volcanoes in the world active in the Holocene period. Essentially a guide to process the required subsets and indices. Note that these 'areas' are in fact a combination of location and time of interest, the latter possibly retrofitting to a recently occurred eruption event that missed our attention.
- The production of interferograms that show and quantify soil displacement.
- Enabling cross-referencing between the atmospheric and solid-earth datasets.

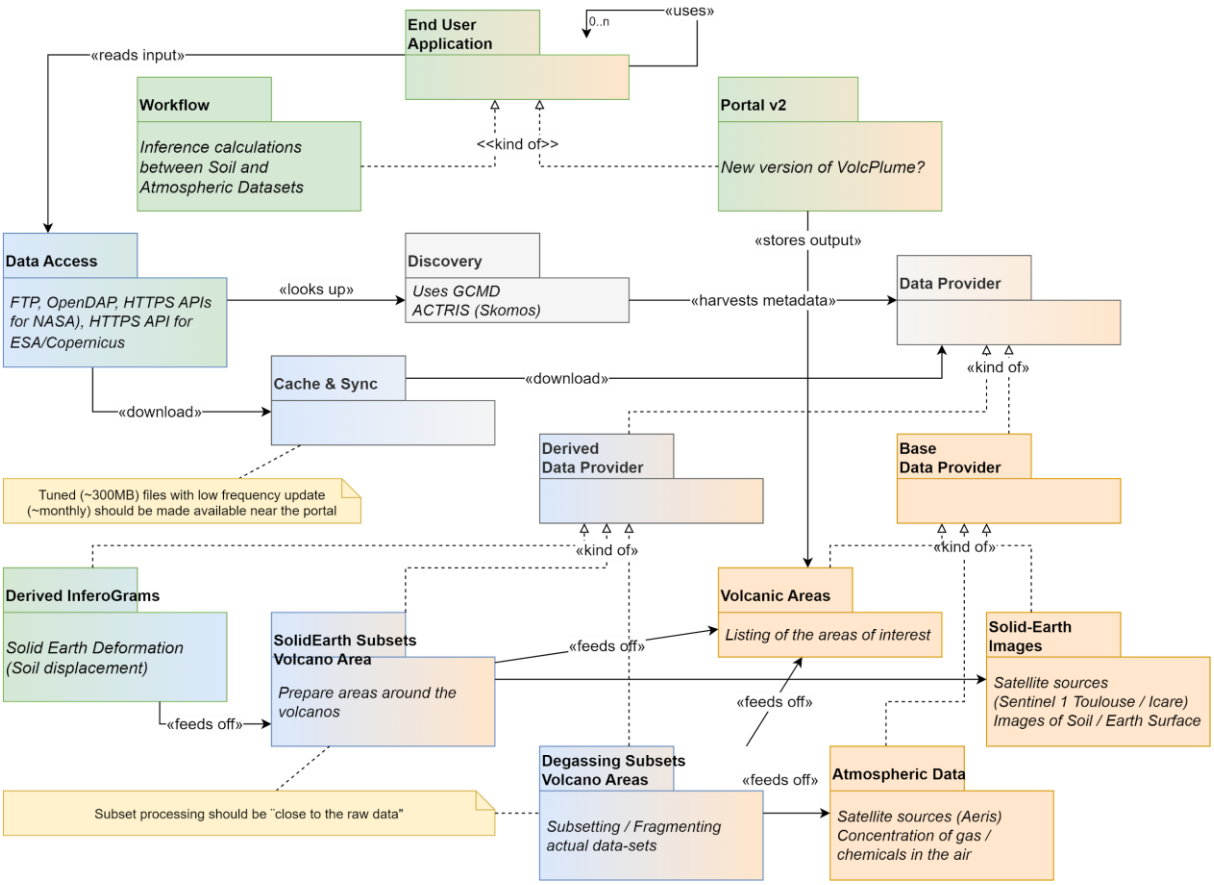


**Figure 10 – Fitting the Volcano Space Pilot to the Unified View**

In this case, again, the many existing "data-access" protocols and techniques will push us to tunnel these opaque sets in an opportunistic way. Fitting this flexibility into the contracts from the [[Discovery]] and [[Data Access]] blocks guarantees a running start rather than a preemptive hurdle by being able to work off existing systems. As explained earlier, the introduced abstractions can be simple wrapper implementations that allow for independent evolution and rewrite as we go along and identify priorities.

In addition to the clear opportunity of aligning partial solutions between these two last pilots, a bonus is expected from a tuned solution for the [[Cache & Sync]] concern. Practical engineering choices will ensure local and performant access to raw data within this pilot. The remaining challenge will be to address the intended 'location agnostic' feature of a truly interoperable data-space.

## 3.4 Ocean Bio-Geo-Chemical Observatory (Pilot 5.2.1)

The purpose of this pilot is to establish a shared platform for data scientists to qualify, calibrate, and validate BGC (biogeochemical) data collected by sensors on various platforms, including Argo. This process results in revised, quality-annotated versions of the raw datasets that include a traceable provenance trail. An interesting feature of this solution is the calculation of subsets from remote-sensing datasets based on the free-floating trajectories of the in-situ platforms.

Funded by the European Union

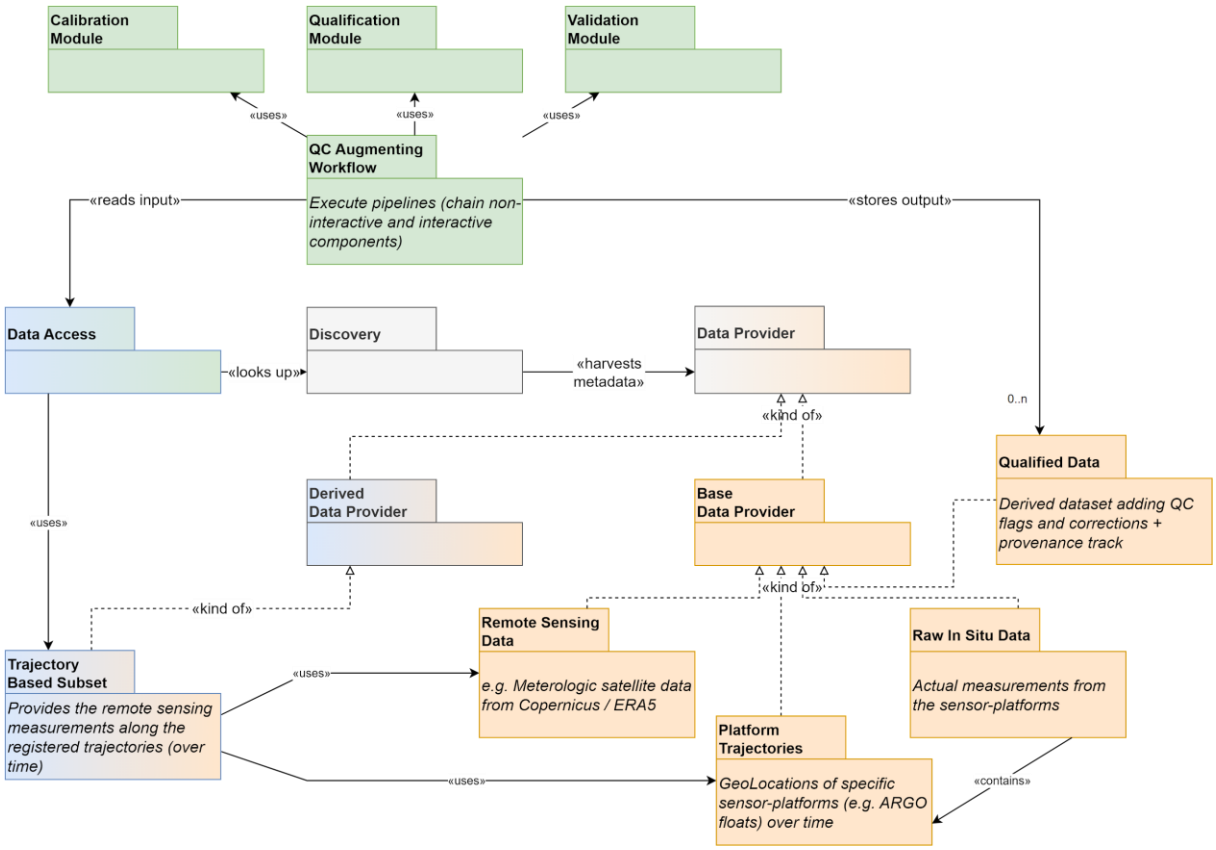


**Figure 11 – Fitting the BGC Pilot to the Unified View**

As with the previous pilots, existing data-access protocols and techniques will be leveraged to overcome the challenges posed by the diverse data sources and formats involved. However in this case an exposed data description that supports custom calculations will surely be required.

## 3.5 Marine Omics Observatory (Pilot 5.3.1)

The aim of this pilot is to manage and semantically share the data from the EMO BON project of EMBRC, which includes: log sheets from regular samplings at different locations (raw complementary data), DNA sequences from the samples (raw data), and results of bioinformatics analysis (data products). The ambition is to provide an open data explorer and two special demonstrator services, one focusing on "bioprospecting" (building on functional analyses of expressed genes in the samples) and one on "ecological strategies" (based on detecting species' guild differentiation among environments).

The selected highlights from this pilot include the intention to include published CTD (conductivity, temperature, and depth) data from the FE platform as an alternative and completion to the data captured during the sampling, adoption of specific knowledge graph technologies and data exposure schemes (RO-Crate, DCAT, LDES), and the introduction of a triple-store and SPARQL-endpoint.

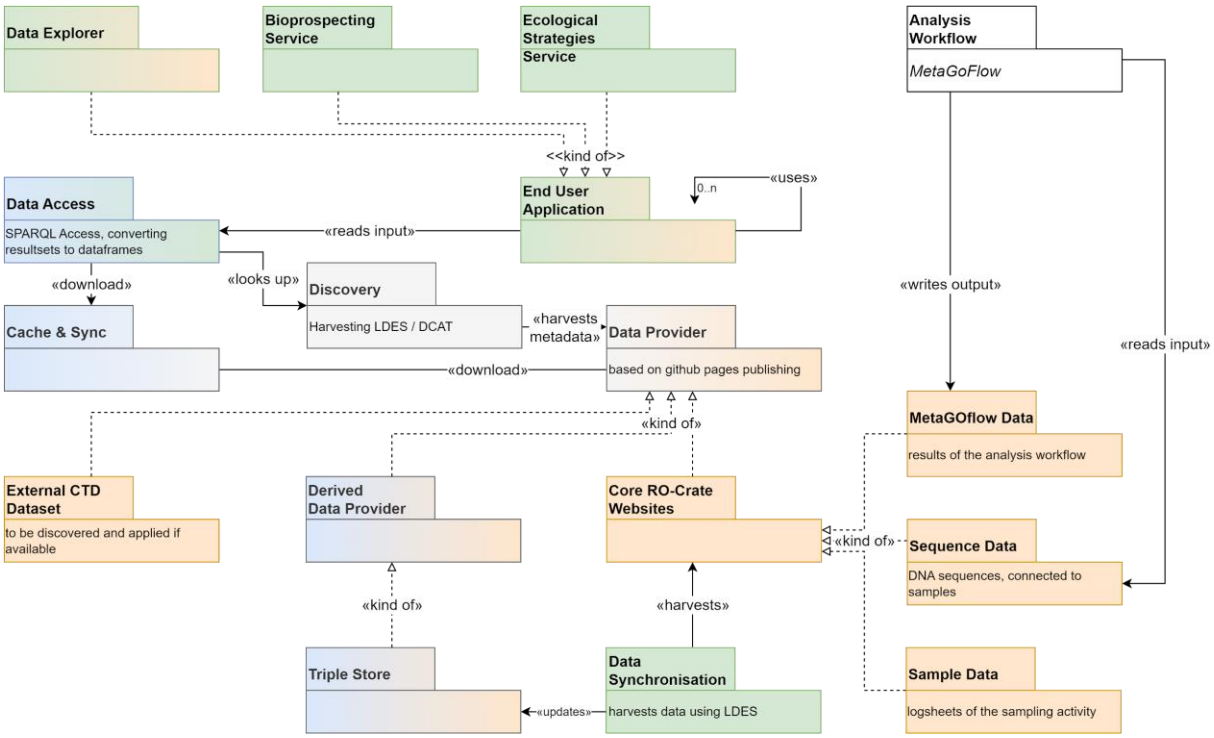


**Figure 12 – Fitting the Marine Omics Pilot to the Unified View**

In this setup, various partners collaborate on datasets using online collaboration tools such as Google Spreadsheets and Git, which are automatically converted into RO-Crate-compliant mini-data-websites through custom workflow-actions on the GitHub platform. Semantic uplifting is applied to make the data available in a ready RDF serialisation format, and a change feed in LDES format is provided through tracking the Git changes of those resulting files.

This allows for optimal harvesting from these core publication zones into an aggregating triple store, which plays the role of a data-lake local to this pilot. To fit the unified model, it too must expose available answers through the abstraction of the [[Data Provider]] interface to make them discoverable and available for the [[Data Access]] needed in the applications.

Regarding the applications, it is important to note that the bioinformatics analysis workflow will not run on the FAIR-EASE platform, but the other applications will fit into the environment provided by WP3.

# 4 Drafting a Working Plan

Looking ahead, we propose several actions and work topics for the FAIR-EASE project. The DevCycles, introduced recently as the work-planning tool, will provide the platform to discuss and address these.

Firstly, we anticipate that the dataset focus planned for the follow-up deliverable D4.2 will take the first step towards gathering requirements to define the necessary metadata schema. This schema will be harvested from the [[Data Provider]] implementations to feed the [[Discovery]], enabling it to point the [[Data Access]] to the best resource to be fetched and optionally support the [[Cache & Sync]] in its workings to have recent fast access local copies around. Collaborating with WP2 to define and plan this metadata schema is instrumental to the success of this architecture.

Secondly, we hope to advance the work within the various pilots by implementing the systems engineering and strategic separation of concerns laid out in this document. We believe that this approach will not only achieve future interoperability but also provide short-term benefits for the pilots within their own domains. Collaboration and ongoing dialogue will be crucial in evolving the architecture.

Finally, we propose to collaborate on the design and initial development of the universal data-access library described in the [[Data Access]] block in collaboration with WP3. We recognise the challenges of covering the many different historical formats and protocols used in the various pilots. Moreover, the revaluation of client-based processing implicit in this approach requires a mental shift and cautious guarding. Pushing this processing back closer to the edge of the network while keeping a separation from the application logic will be instrumental in retaining the opportunity for flexible relocation.

In summary, these proposed actions and work topics will help move the FAIR-EASE project towards its goal of enabling the efficient sharing and management of data across different scientific domains.