eosc | FAIR-EASE
Building Interoperable Earth Science & Environmental Services

eosc | Blue-Cloud2026
A federated European FAIR and Open Research Ecosystem for oceans, seas, coastal and inland waters

FAIR Data Discovery and Access

@OSFAIR 2023 Madrid, 27 September 2023

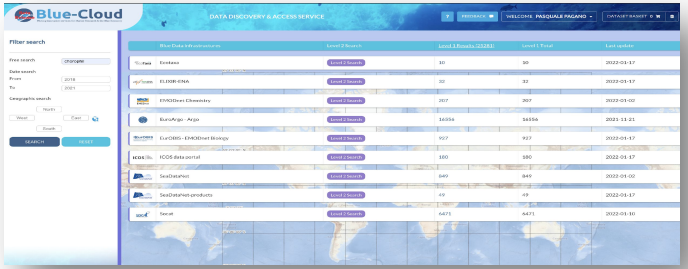
Peter Thijsse - MARIS (Blue-Cloud and FAIR-EASE)

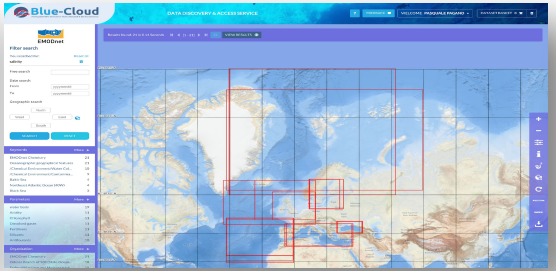
Content

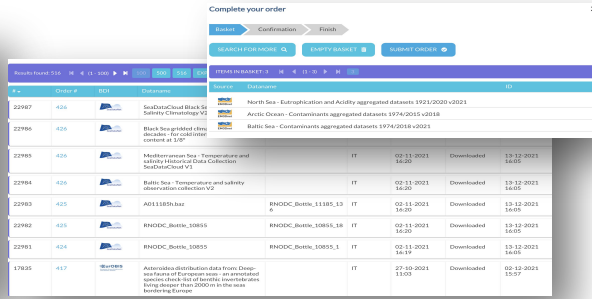
- Introduction
- The challenge of data discovery and access in VRE development
- FAIR data
- FAIR software
- Important step to support VRE developments: FAIR services

Introduction to the challenge

- Many different data providers, distributed in location, services, standards
- Important to offer human and machine users a data discovery and access mechanism with clear guidance on characteristics
- Data lake/space/workspace/…





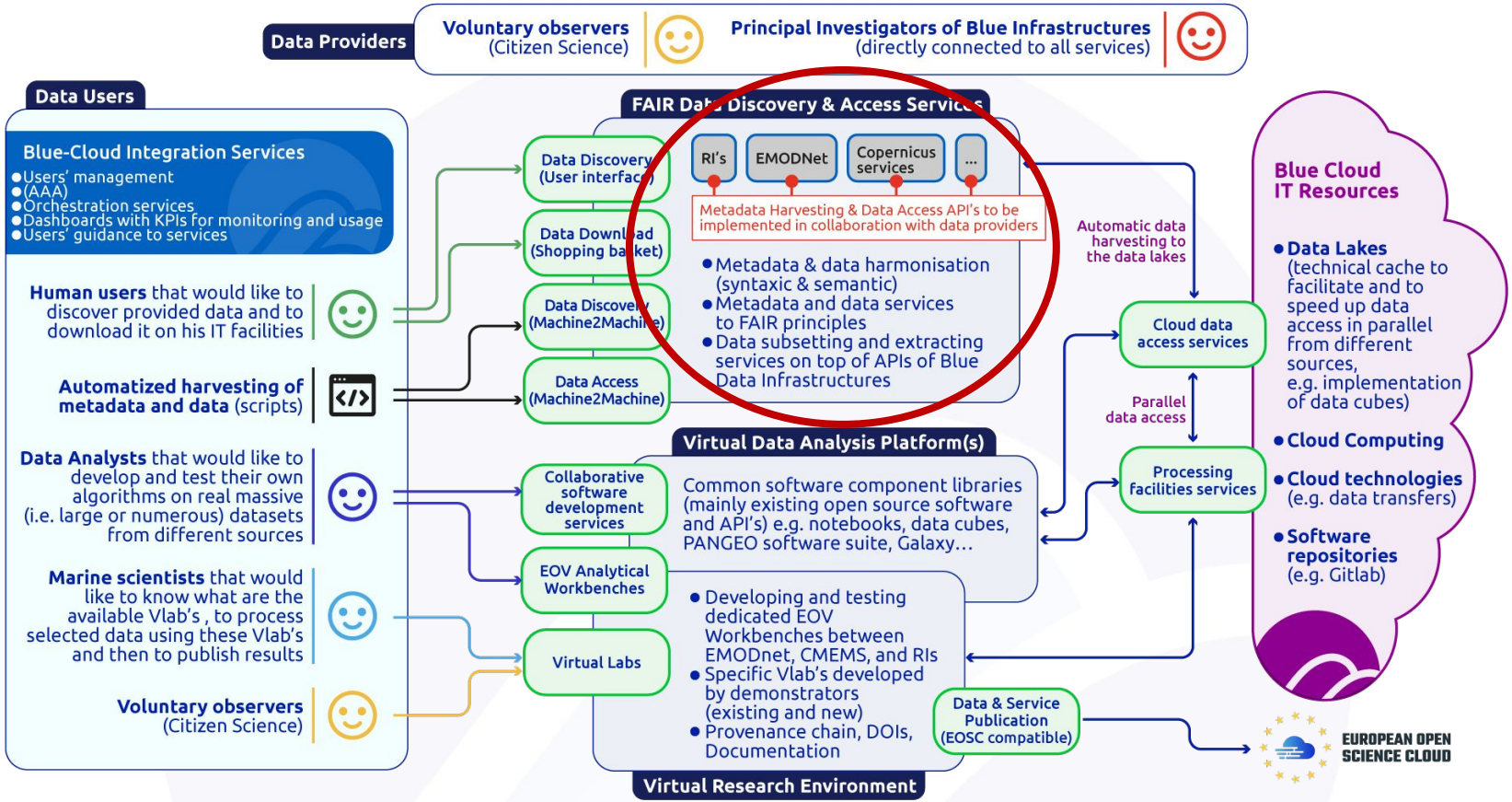


Compose and submit shopping request at the granule level

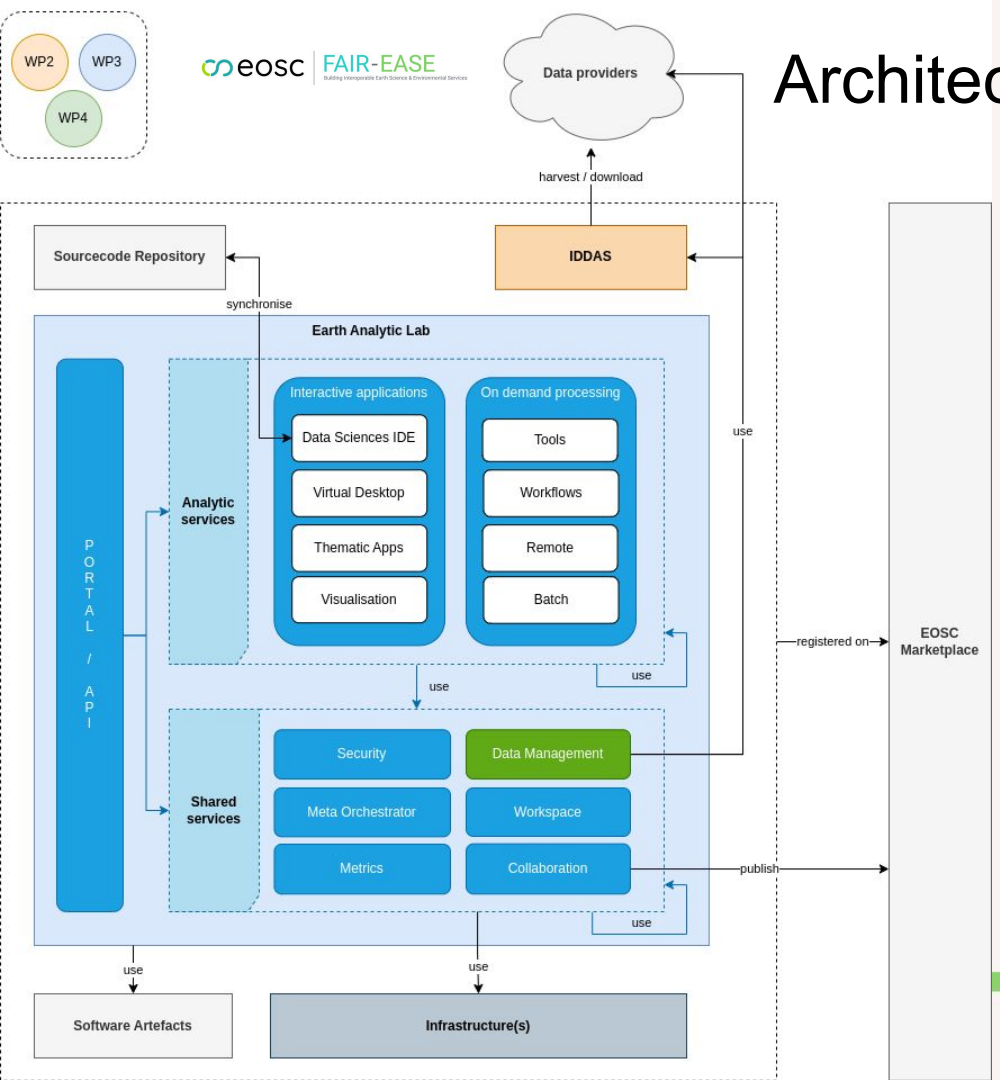
Retrieve the datasets by downloading from the Dashboard

Push datasets to the Blue-Cloud VRE Data Pool

Architecture Blue-Cloud 2026 services core



Architecture FAIR-EASE (EAL focus)



Infrastructure(s)

- Storage : workspace, reference data, scratch (temp)
- Processing resources : CPU, GPU, memory, local disk
- Job/service orchestrator : deploy services and/or submit jobs
- Monitoring : user usage

Data

- IDDAS - Assets catalogue
- Data Management - Assets selector
- Data Providers - data access/subsets services

Software and scripts

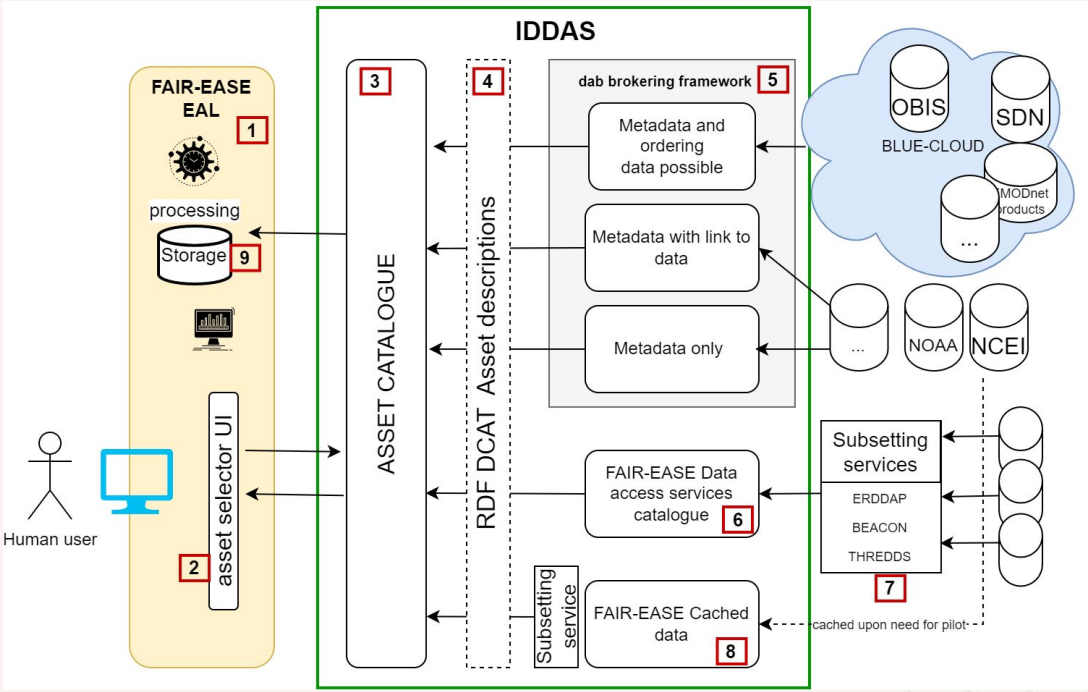
- Sourcecode repository : git, …
- Software artefacts :
  - Packages (python, R, julia, …, conda)
  - Container images (docker, apptainer/singularity)

EOSC Marketplace

- Register FAIR-EASE services
- Publish users output : data, tools/services, workflows/scripts, documentation, …

IDDAS in development

1. Earth Analytical Lab
2. **Asset selector**
3. **Asset catalogue**
4. **Asset descriptions**
5. **DAB brokering framework**
6. **Data access services catalogue**
7. **Sub-setting services**
8. **Cached data**
9. Data Storage



Central role of data access

- Similar challenges in BC2026 and FAIR-EASE, and also in EOSC
- VRE systems are highly dependent on
  - FAIR data: for findability and (re-)use of distributed data
  - FAIR software: e.g. for processing
  - and well described interoperable (FAIR?) services => to access the data, without required human contact/interpretation
- Only then we are supporting the full “data lake/data space” concept

Important points/differences

- DDAS key position in both architectures
- Level 1 metadata level access via DAB (CNR)
- Level 2 data access level to distributed services via metadata to data access services.
- Important: Metadata and data harmonisation
  - metadata model mapping
  - vocabulary mapping (parameters, units, etc)
- For each data access service a specific “conversion” has to be implemented
  - how to search datasets
  - how to order
  - how to move from metadata to the data file request
  - difficult and human intervention needed.
- Difference between BC and FE:
  - BC only marine Blue Data Infrastructures, all on board of consortium (so able to upgrade services, implement agreed solutions)
  - FE is multidisciplinary, and most infrastructures not on board as partner
  - FE aims to include also direct data access (subsetting services) as part of the IDDAS

FAIR data and software

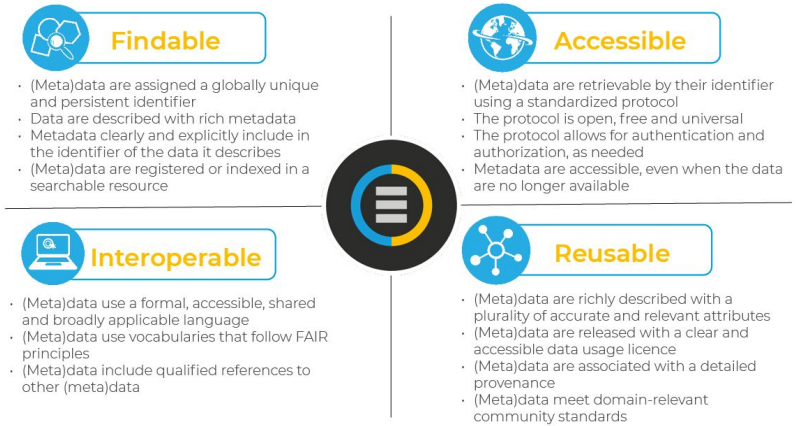
FAIR data solutions (e.g. in ENVRI-FAIR) using FIP approach:

- improved machine2machine services for metadata and data access
- Upgraded metadata model for enhanced FAIRness (e.g. quality info)
- Expanded vocabularies to support provenance (Re-usability)

FAIR software examples:

- Software as a research object
- Publication in Zenodo/Github with sufficient metadata
- version management
- Clear license in metadata
- Software meets community standards

Metrics: FIP, F-UJI



FAIR data principles - source: CCDC

F: Software, and its associated metadata, is easy for both humans and machines to find.	**I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.**
F1. Software is assigned a globally unique and persistent identifier.	I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.
F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.	I2. Software includes qualified references to other objects.
F1.2. Different versions of the software are assigned distinct identifiers.	**R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).**
F2. Software is described with rich metadata.	R1. Software is described with a plurality of accurate and relevant attributes.
F3. Metadata clearly and explicitly include the identifier of the software they describe.	R1.1. Software is given a clear and accessible license.
F4. Metadata are FAIR, searchable and indexable.	R1.2. Software is associated with detailed provenance.
A: Software, and its metadata, is retrievable via standardised protocols.	R2. Software includes qualified references to other software.
A1. Software is retrievable by its identifier using a standardised communications protocol.	R3. Software meets domain-relevant community standards.
A1.1. The protocol is open, free, and universally implementable.	
A1.2. The protocol allows for an authentication and authorization procedure, where necessary.	
A2. Metadata are accessible, even when the software is no longer available.	

FAIR software principles - source: M. Barker et al. Nature 2022

One step further: FAIR services?

FAIR principles for **data access services** → increase findability, accessibility and interoperability of data access services (machine-2-machine)

This can be achieved by describing the services in a standardized manner, such that information is made available on:
1. what the service does, what is offers
2. how it works, how to make requests
3. how to access it (authentication?)
4. input/output

A starting point will be **research on currently available ontologies for describing services**
- Several standardized vocabularies and ontologies are available
  - Particularly in the context of the Semantic Web and Linked Data
- These vocabularies help provide structured and machine-readable descriptions of services, making them more discoverable and interoperable

Some of the commonly used models and vocabularies for describing services (but these are in our opinion not yet complete):
- OWL-S
- OpenAPI Specification (formerly Swagger) => most promising candidate, when published as RDF. Needs additional attributes and semantics
- Dublin Core Metadata Initiative (DCMI)
- DCAT Class
- Hydra
- ESIP
- ODIS
- schema.org

Way forward

- In FAIR-EASE a working group will focus on best possible solution for describing services for m2m access
- Solutions will be documented and tested as prototype
- Starting point what already exists, building on top of that
- Close contact with BC2026, and possibly other initiatives

- Looking for examples in other domains, RDA WG, other infastructures?
  - Please contact us when interested to share views and experiences

=> Let’s discuss!

Time for questions and discussion.

contact: peter@maris.nl