EOSC SYNERGY

FAIR EVA - Webinar

April 2022

Fernando Aguilar

EOSC-SYNERGY receives funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 857647

EOSC-SYNERGY in a nutshell

https://eosc-synergy.eu/

Promote EOSC High Quality Services

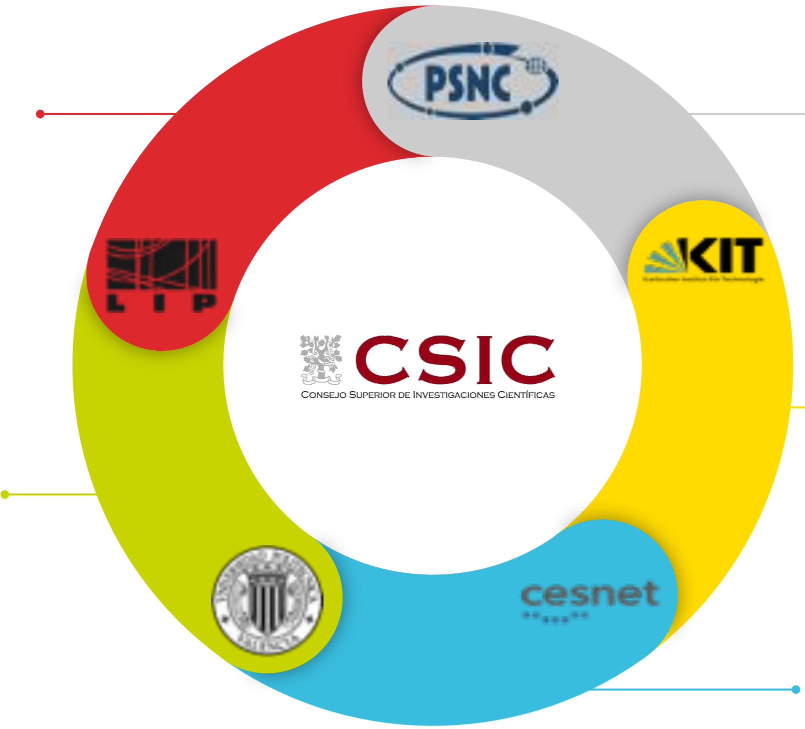
Software quality as a service, **FAIRness evaluation** and quality certification badges

Thematic Services Integration

10 thematic services addressing 4 scientific areas (Earth Observation, Environment, Biomedicine and Astrophysics)

22 partners in 10 countries

(ES, PT, FR, UK, DE, NL, CZ, SK, PL and BR)



Skills development

Environment for tutorials with a dedicated MOOC platform, courses methodology and a Hackaton as a service platform

Capacity Expansion at the Infrastructure level

Integration of services and resources from the RIs of the consortium partners

Alignment at the Policy Level

Collaboration with regional projects on landscaping activities, gap analysis and contribution to EOSC policies

Overview of the FAIR Principles

- Findable
- Accessible
- Interoperable
- Reusable

Refer to three types of entities: data (digital object), metadata (information about that digital object), and infrastructure.

FAIR Indicators

RESEARCH DATA ALLIANCE

Findable

- F1: (Meta) data are assigned globally unique and persistent identifiers
- F2: Data are described with rich metadata
- F3: Metadata clearly and explicitly include the identifier of the data they describe
- F4: (Meta)data are registered or indexed in a searchable resource

Accessible

- ❑ A1: (Meta)data are retrievable by their identifier using a standardised communication protocol
- ❑ A1.1: The protocol is open, free and universally implementable
- ❑ A1.2: The protocol allows for an authentication and authorisation where necessary
- ❑ A2: Metadata should be accessible even when the data is no longer available

Interoperable

- ❑ I1: (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation
- ❑ I2: (Meta)data use vocabularies that follow the FAIR principles
- ❑ I3: (Meta)data include qualified references to other (meta)data

Reusable

- R1: (Meta)data are richly described with a plurality of accurate and relevant attributes
- R1.1: (Meta)data are released with a clear and accessible data usage license
- R1.2: (Meta)data are associated with detailed provenance
- R1.3: (Meta)data meet domain-relevant community standards

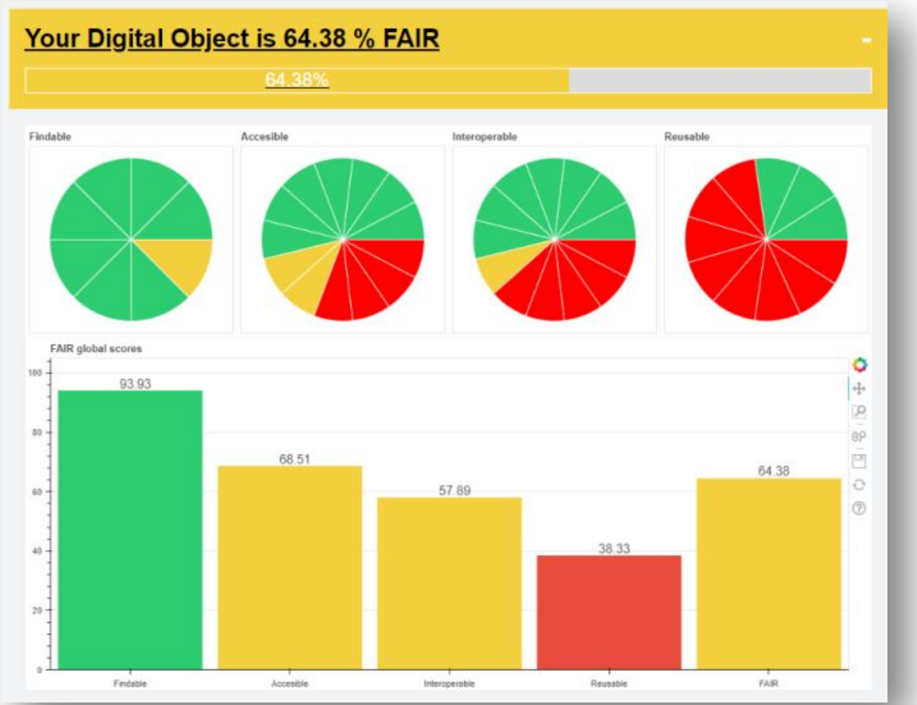
FAIR EVA - Evaluator, Validator & Advisor

1. FAIR indicators technical implementation.
    a. Starting from RDA
2. Modular, Scalable, Flexible
    a. Generic implementation OAI-PMH
    b. Plugins
3. Not only evaluate, but also validate and **advise**
4. Target: data producers, repository/data portal admins, funders.
5. Python. API + Web Interface
6. Stand-alone - Docker

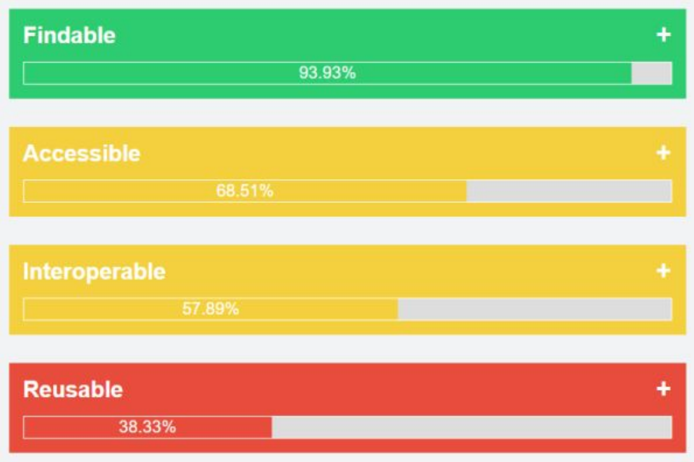
FAIR Assessment tools - FAIR EVA

- FAIR EVA functionality
- Comply with FAIR Data principles:
  1. **Data**: use a proper format
  2. **Metadata**: community standard. Machine-actionable (JSON, XML, RDF…)
  3. **PIDs**: Persistent Identifier (e.g. DOI). Provided by an accepted authority.
  4. **Repository/Data service**: indexed and machine-actionable.
- Integration: Different types of repositories/data portals

FAIR EVA


Your Digital Object is 64.38 % FAIR

64.38%


Findable +

93.93%

Accessible +

68.51%

Interoperable +

57.89%

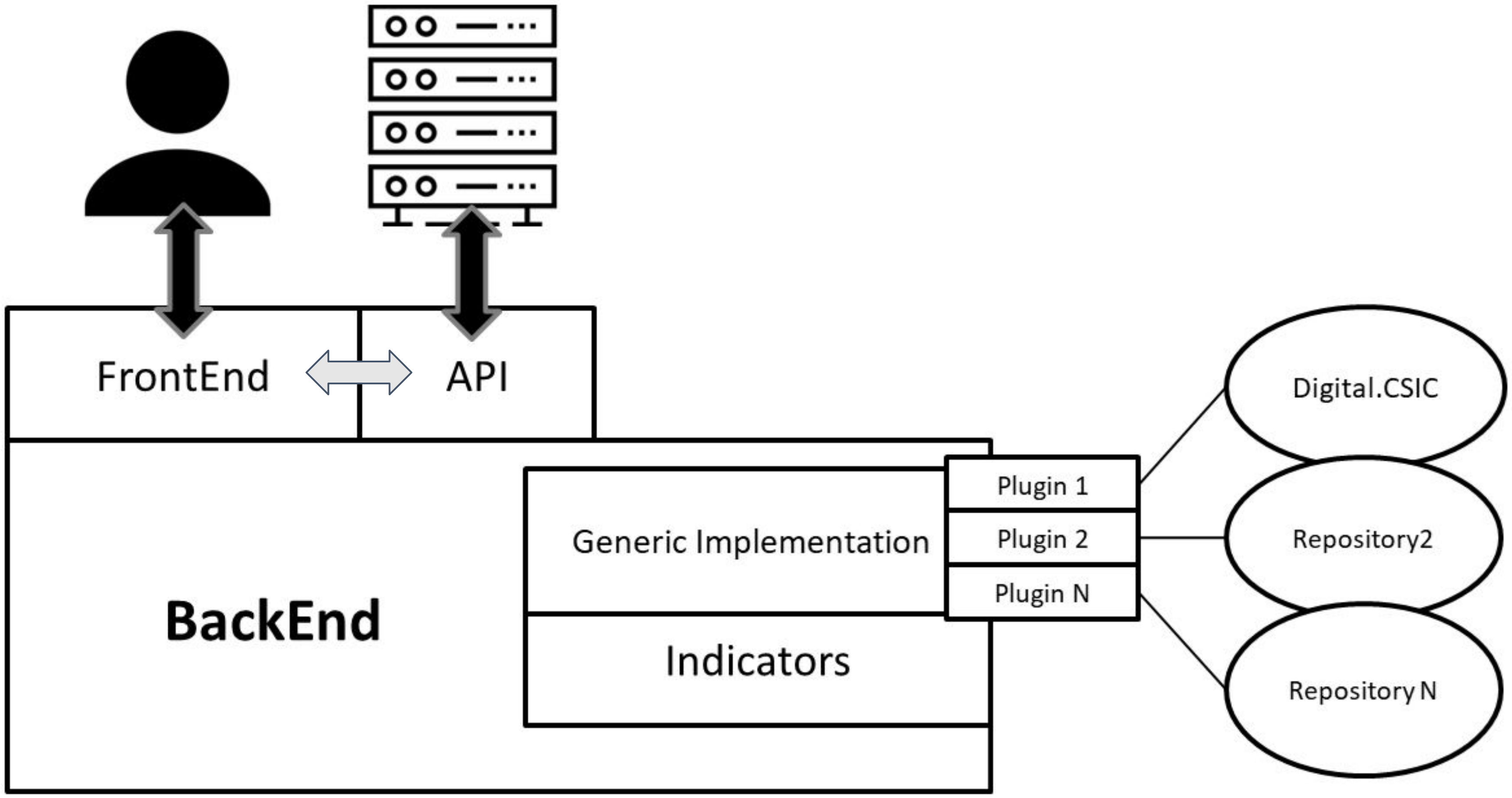
Reusable +

38.33%

Technical Implementation examples

Indicator Code	EN_title	EN_tech
RDA-F1-01M	Metadata is identified by a persistent identifier	Search within a predefined list of potential metadata terms to identify the metadata (dc.identifier.uri and dc.identifier.doi) if any information is available.
RDA-A1-04M	Metadata is accessed through standardised protocol	Checks the protocol to access metadata (OAI-PMH, API..)
RDA-I1-01M	Metadata uses knowledge representation expressed in standardised format	Checks metadata terms including controlled vocabulary information. So far, it checks GEONAMES, Library of Congress Subject Headings and ORCID, but it's being extended.
RDA-R1-01M	Plurality of accurate and relevant attributes are provided to allow reuse	Depending on the metadata schema used, checks that at least the mandatory terms are filled (75%) and the number of terms are high (25%)

FAIR EVA - Architecture



Launch application

Docker deployment

```
docker pull ferag/fair_eva:latest

docker run --name=fair_eva -dit --network host ferag/fair_eva:latest
```

Launch application

Stand-alone mode

```
git clone https://github.com/EOSC-synergy/FAIR_eva.git
```

```
cd ./FAIR_eva
```

```
pip3 install -r requirements.txt
```

```
cp config.ini.template config.ini
```

```
/FAIR_eva/fair.py &
/FAIR_eva/web.py &
```

Demo

localhost:5000

OAI-PMH generic implementation

DOI: `10.5281/zenodo.23176`

OAI-PMH Endpoint: https://zenodo.org/oai2d

Configuration

The config.ini file contains all the configuration parameters. They are distributed in different sections. To customize your FAIR EVA deployment. It will work by default, but this is what you can edit:

```
[local]

# Defines if your service is deployed close to the data service and if it is only configured to work with that service

# only_local = true if it will run only for your service

only_local = false
  repo = oai-pmh
```

Configuration

The repositories or data portals that implements a plugin can be listed in the service. You can configure those that you wants to appear in the list. Every repository should be shown with the display name equal python Class name. The Generic class is "Evaluator"

```
[Repositories]

oai-pmh = 'Evaluator'

digital_csic = 'Digital.CSIC'

example_plugin = Example_Plugin
```

Configuration - Metadata schemas

```
[oai-pmh]

# Metadata terms to find the resource identifier
identifier_term = ['identifier']


# Metadata terms to check richness (generic). These terms should be included [term, qualifier]. None means no qualifier
terms_quali_generic = [['contributor',None],
                       ['date', None],
                       ['description', None],
                       ['identifier', None],
                       ['publisher', None],
                       ['rights', None],
                       ['title', None],
                       ['subject', None]]
```

Configuration - Metadata schemas

```
# Metadata terms to check richness (disciplinar). These terms should be included [term, qualifier]

terms_quali_disciplinar = [['contributor', None],

                           ['date', None],

                           ['description', None],

                           ['identifier', None],

                           ['publisher', None],

                           ['rights', None],

                           ['title', None],

                           ['subject', None]]



# Metadata terms that defines accessibility

terms_access = [['access', ''], ['rights', '']]
```

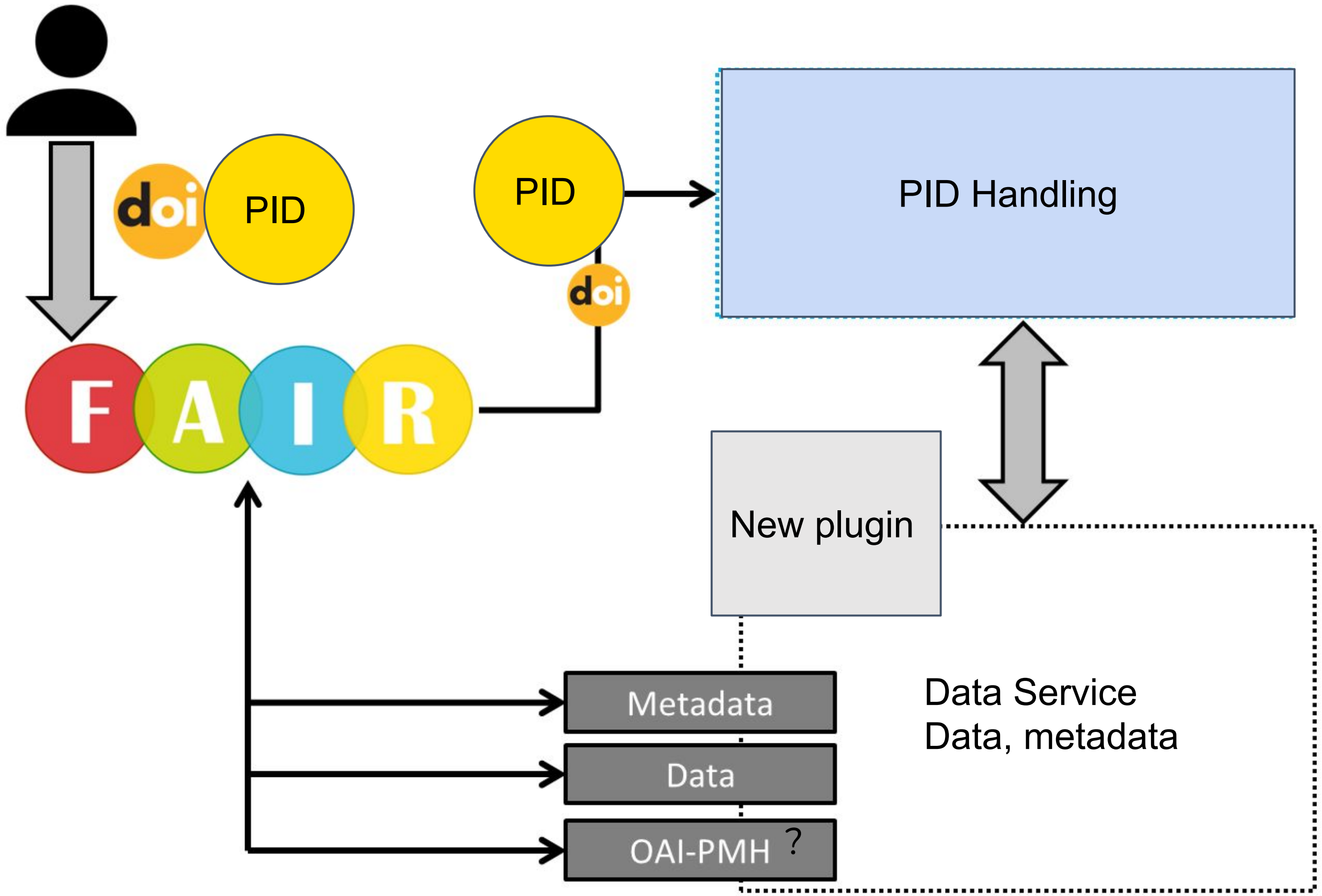
Configuration - Metadata schemas

```
# Metadata terms wich includes controlled vocabularies. More controlled vocabularies can be implemented in plugins

terms_cv = [['coverage', 'spatial'], ['subject', 'lcsh']]


# List of data formats that are standard for the community

supported_data_formats = [".txt", ".pdf", ".csv", ".nc", ".doc", ".xls", ".zip", ".rar", ".tar", ".png", ".jpg"]


# Metadata terms that defines links or relation with authors, contributors (preferebly in ORCID format)

terms_qualified_references = ['contributor']


# Metadata terms that defines links or relation with other resources, (preferebly in ORCID format, URIs or persistent identifiers)

terms_relations = ['relation']


# Metadata terms that defines the license type

terms_license = [['license', '', '']]
```

Demo

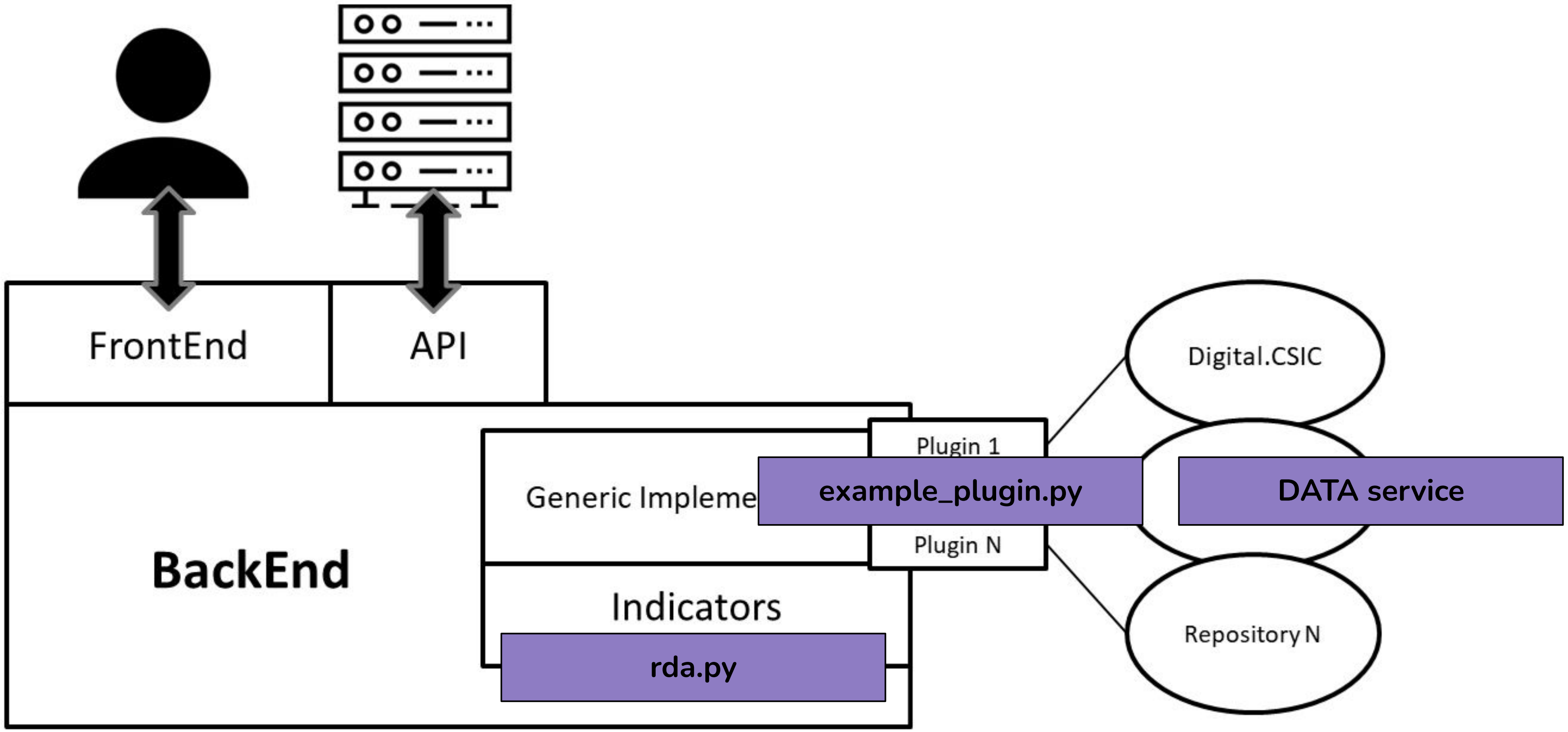
Changing metadata term to check [license - rights]

FAIR EVA - New plugin

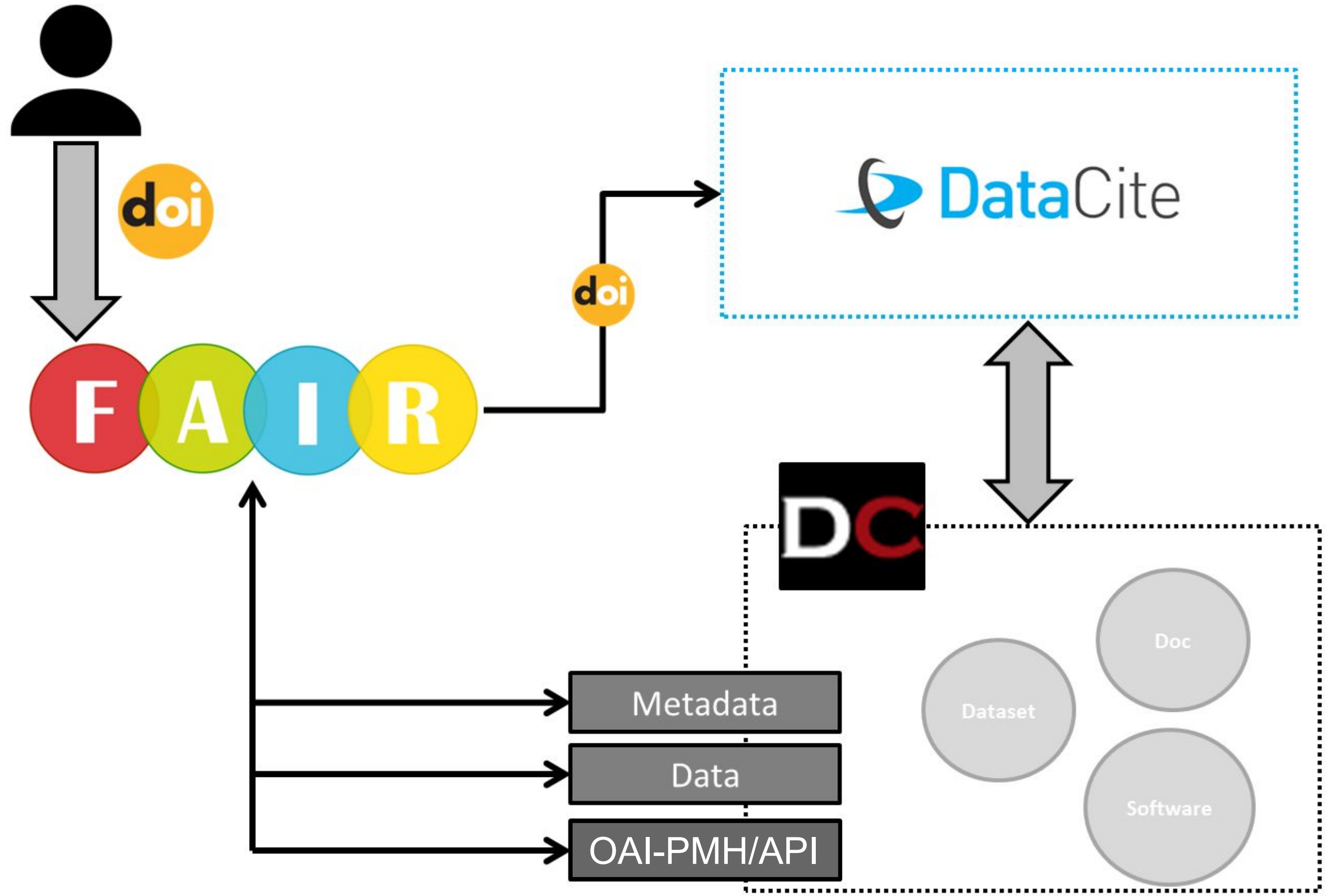


- How metadata can be gathered?
- Persistent Identifier minted?
- How data is gathered?
- Any standard protocol for metadata/data harvesting?

Files to edit



FAIR EVA - DIGITAL.CSIC plugin flow



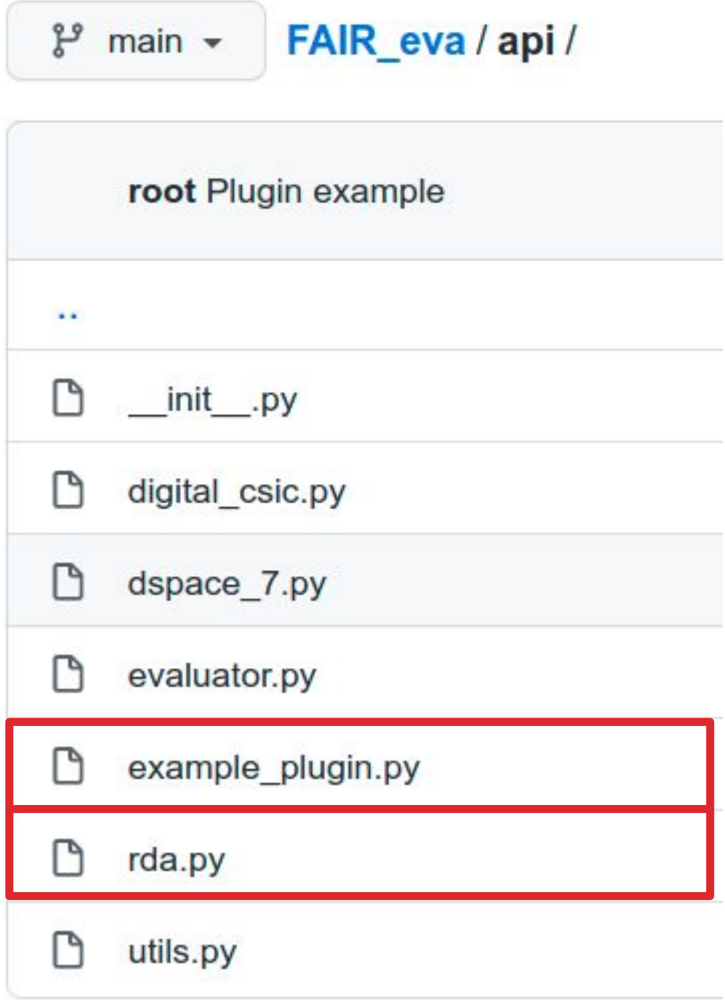
FAIR EVA - Digital.CSIC plugin

Demo: Configuring…

PID: 10261/256232, **10261/153475**

http://digital.csic.es/dspace-oai/request

Easy way



main

FAIR_eva / api /

root Plugin example

..

__init__.py

digital_csic.py

dspace_7.py

evaluator.py

example_plugin.py

rda.py

utils.py

API exploitation

Demo Jupyter

Next steps

- Improve plugin system. Automatic
- Plugin validation
- Web Interface
- Any feedback is welcome
- Links:
  - DOI: 10.20350/digitalCSIC/14559
  - GitHub repository: https://github.com/EOSC-synergy/FAIR_eva

Thank you!

Fernando Aguilar (CSIC)

EOSC-synergy receives funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 857647.