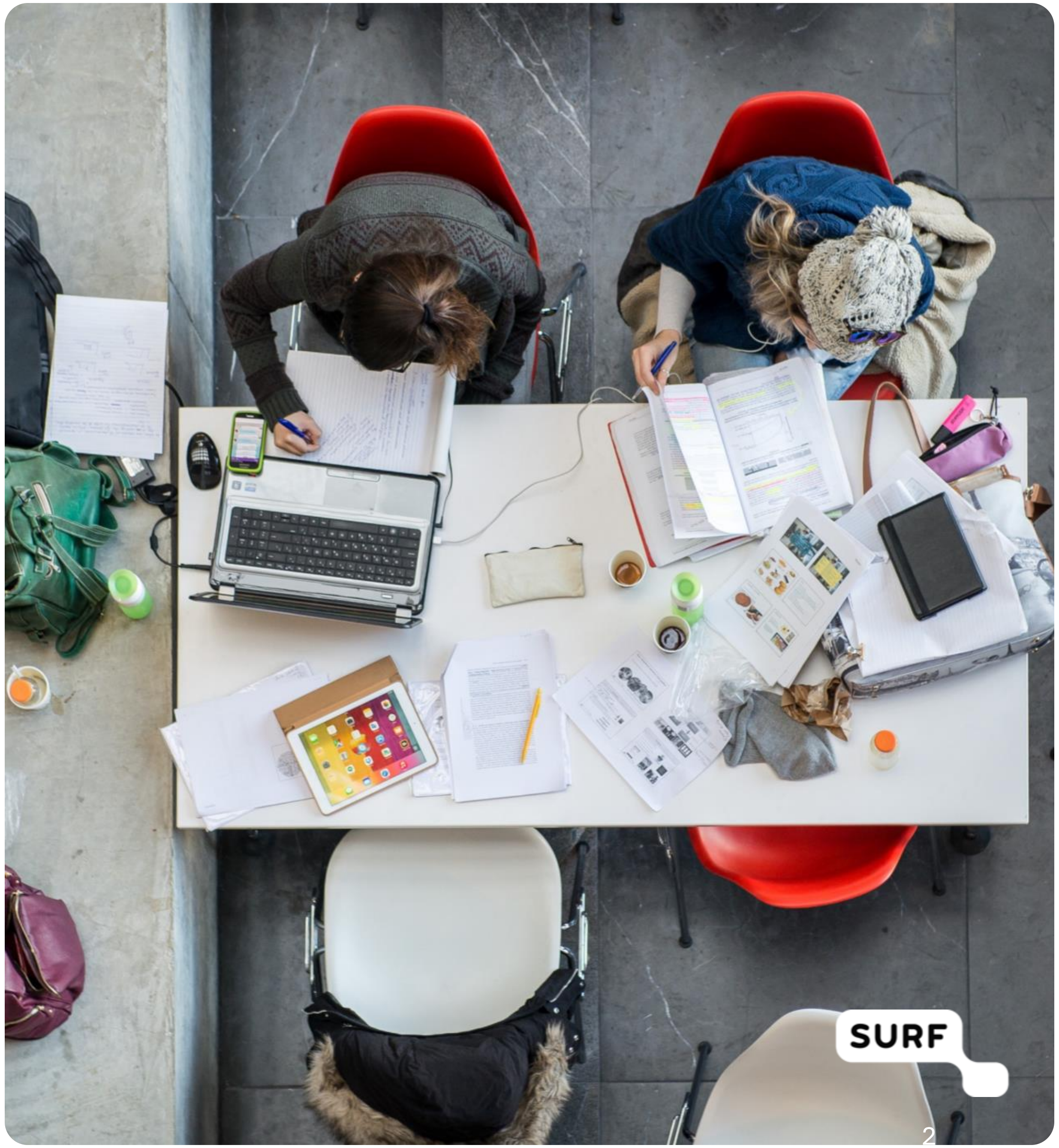
A data mesh for research data management

Claudio Cacciari

16 June, 2025

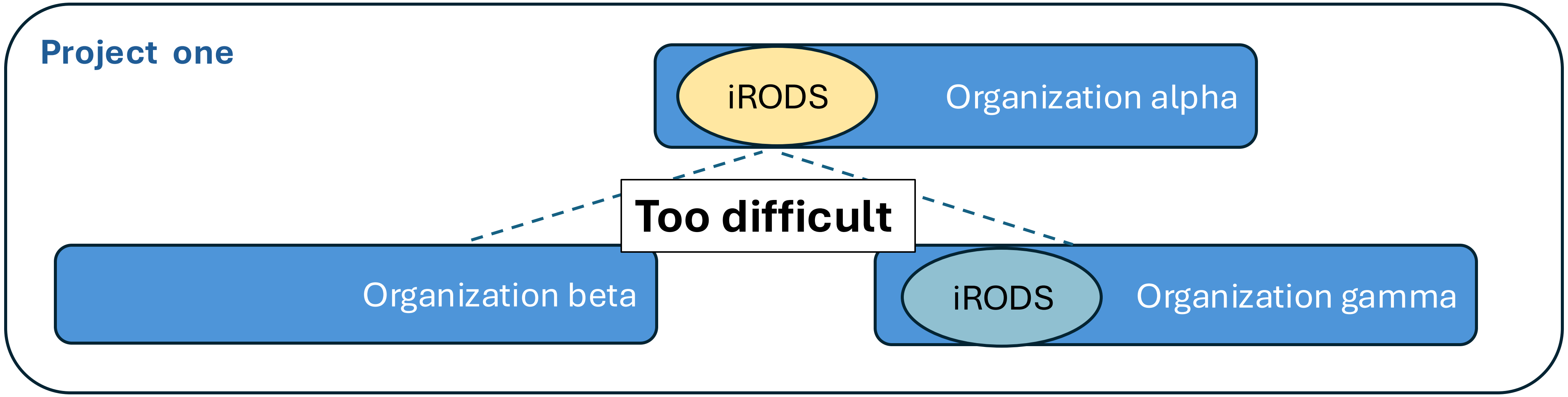
SURF is the collaborative organisation for IT in Dutch education and research

Within the SURF cooperative, education and research institutions work together to make full use of the opportunities offered by digitalisation, **with the aim of ensuring that the education and research communities have access to the best IT facilities for top-level research and talent development**



One data platform, multiple organizations

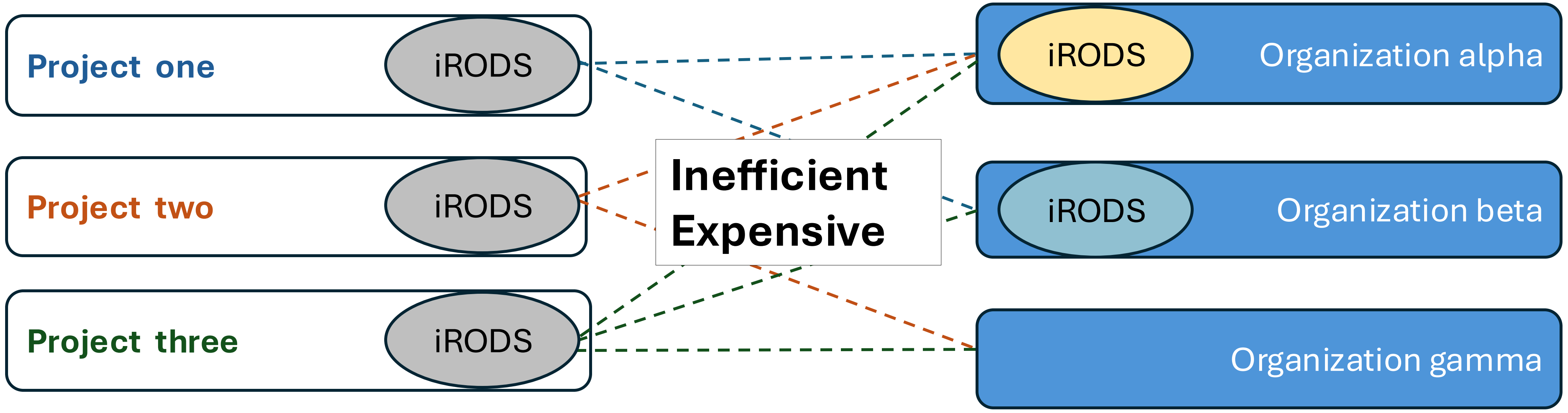
- We have seen iRODS adopted successfully in various organizations.
- When we look at research projects that encompass multiple organizations, or multiple units within big, distributed organizations, we notice that there are more difficulties, both of human and technical nature.



One data platform, multiple organizations

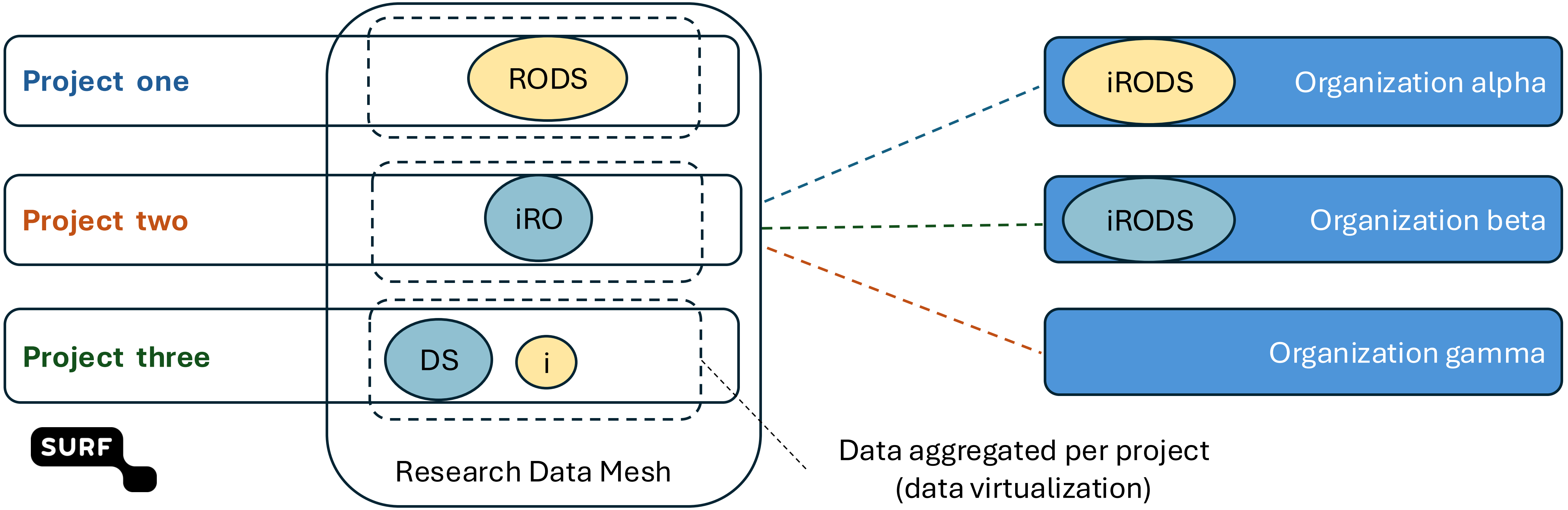
- The workaround is to just deploying a new dedicated data platform, but that means higher costs and bigger fragmentation among organizations and within each single organization itself.

- How can we make the creation of project or community-oriented data infrastructures easier? How can we make them benefit from existing iRODS and YODA services already well established within partner organizations?



Research Data Mesh

- A Data Mesh is a decentralized data architecture that enables the extraction of large-scale analytical data.
- We propose here to adapt the Data Mesh concept for the field of research data, where its scope can be re-defined in this way: to create a decentralized data architecture that enables the exchange of large-scale FAIR data.



Data Mesh principles

A Data Mesh relies on four principles:

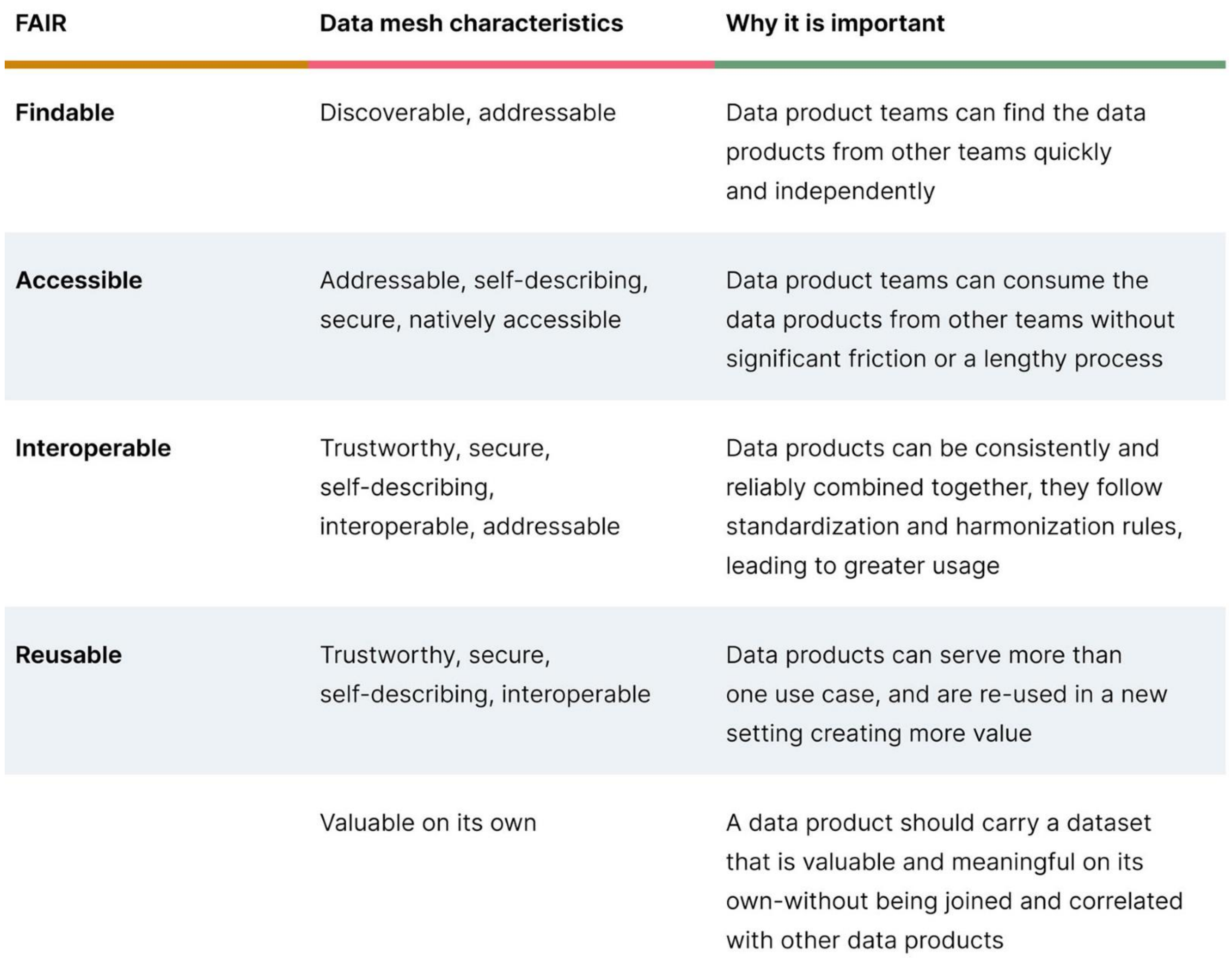
1. domain-oriented decentralized data ownership and architecture,
2. data as a product,
3. self-serve data infrastructure as a platform, and
4. federated computational governance

Domain-oriented decentralized data ownership and architecture

- This principle help us to address the problem of the fragmentation of data infrastructures among different technologies and services.
- The idea is to support a different way to provide and consume the data, aggregating them per research domain, community, project.

Data product

- This principle guarantees the quality of the data
- In the business field, a node in the mesh is a data product. In the research field, we may have similar scenarios, but we have certainly also cases where the Data Mesh node is a set of data products, aggregated by project or scientific domain.



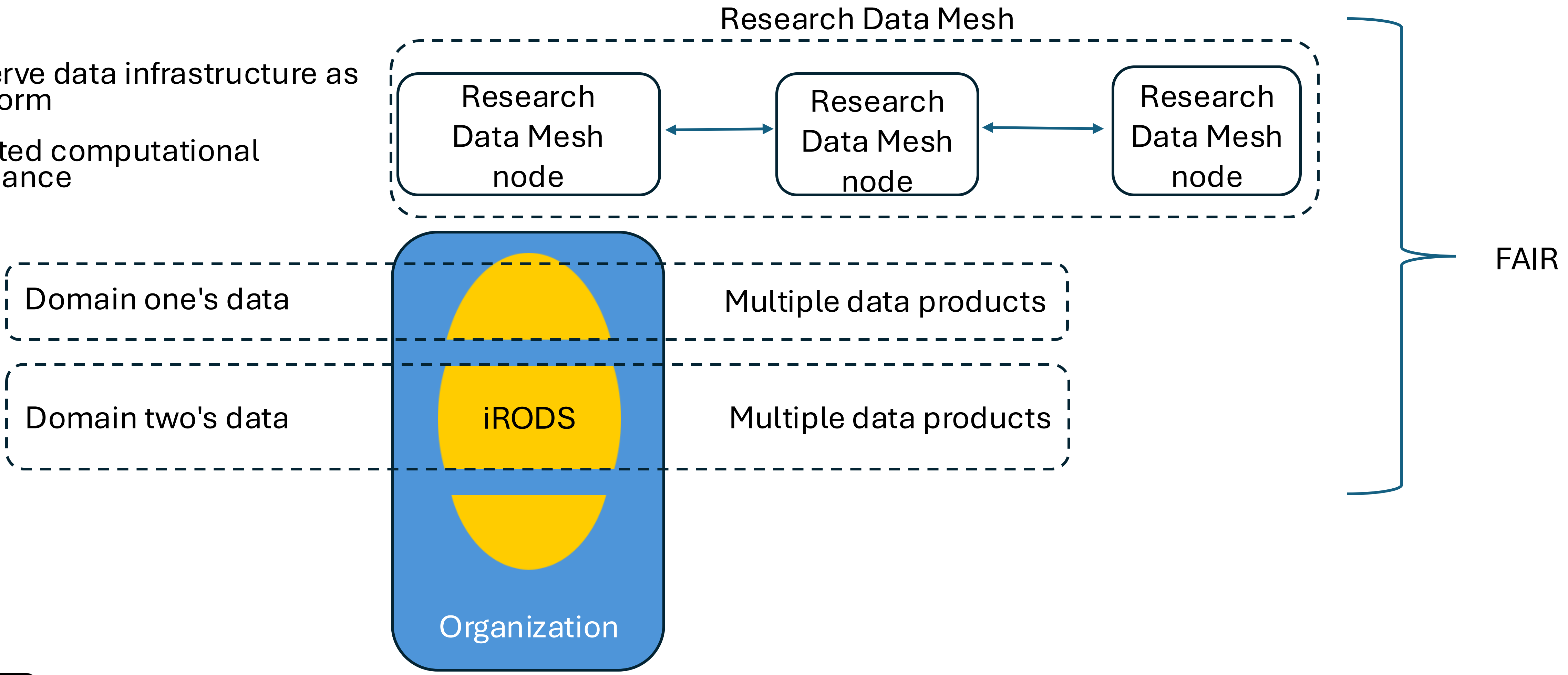
FAIR	Data mesh characteristics	Why it is important
Findable	Discoverable, addressable	Data product teams can find the data products from other teams quickly and independently
Accessible	Addressable, self-describing, secure, natively accessible	Data product teams can consume the data products from other teams without significant friction or a lengthy process
Interoperable	Trustworthy, secure, self-describing, interoperable, addressable	Data products can be consistently and reliably combined together, they follow standardization and harmonization rules, leading to greater usage
Reusable	Trustworthy, secure, self-describing, interoperable	Data products can serve more than one use case, and are re-used in a new setting creating more value
	Valuable on its own	A data product should carry a dataset that is valuable and meaningful on its own-without being joined and correlated with other data products

https://www.thoughtworks.com/insights/blog/data-strategy/data-mesh--helping-life-sciences-organizations-get-more-from-the

All together

self-serve data infrastructure as a platform

federated computational governance



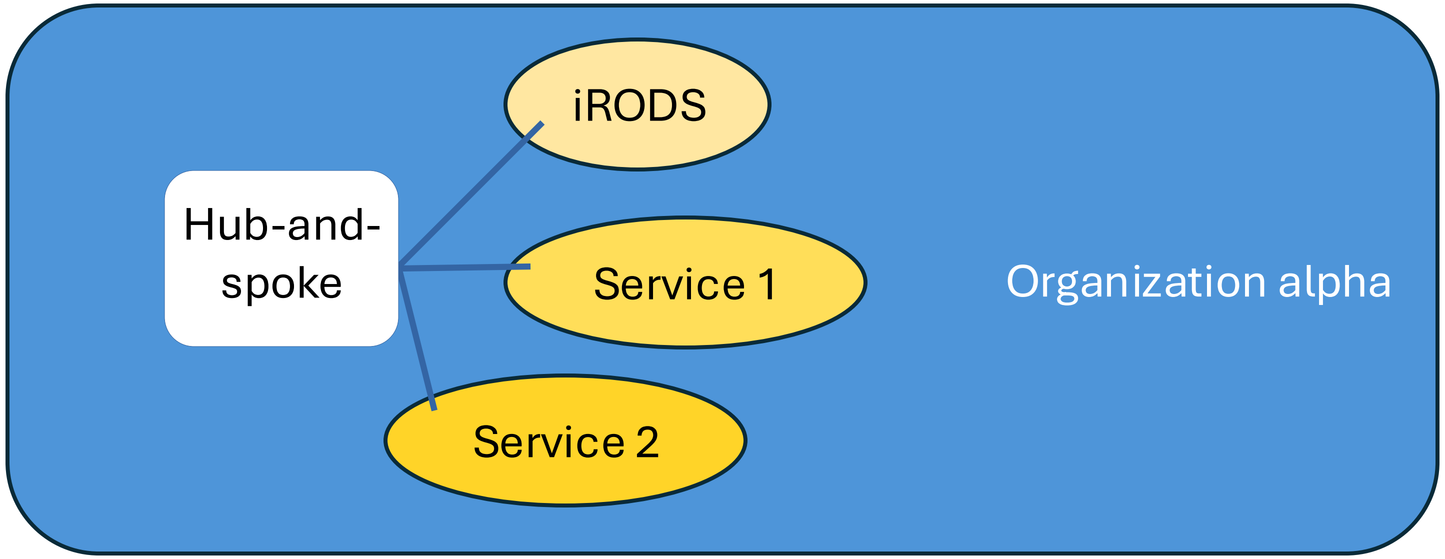
iRODS as a Research Data Mesh node

- iRODS and YODA can be seen as nodes of a Research Data Mesh.
- Then we could offer a solution to the initial problem of the creation of project or community-oriented data infrastructures, building a mesh where the services and the data of the different partner organizations are shared in a FAIR way.
- The federated governance approach would require an organizational effort to define common rules, but minimal, just enough to define the “borders” of the mesh’s space, leaving each partner in control of its own data.

SURF

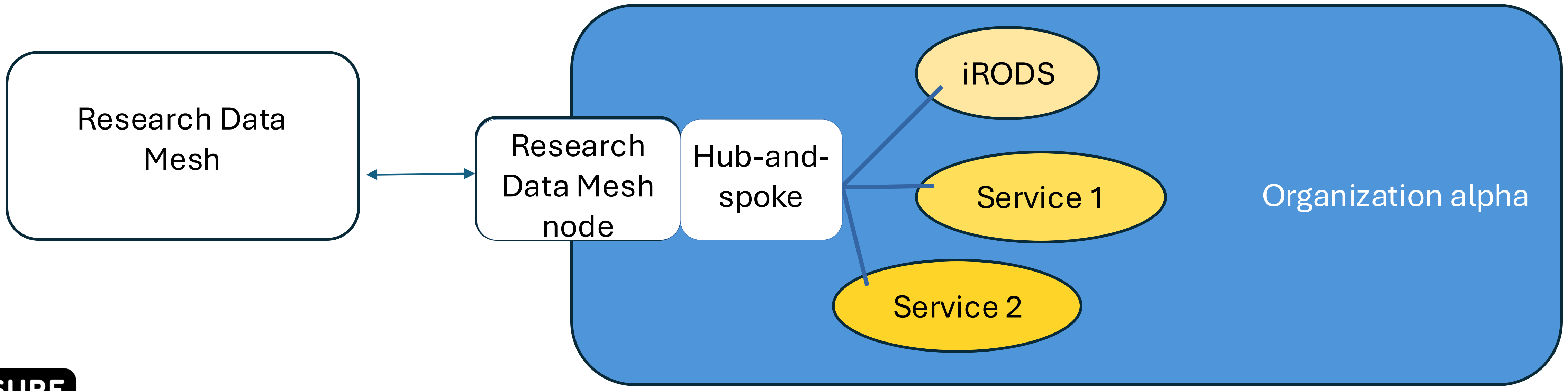
One organization, multiple services

- Sometime, an organization has multiple services that are involved in the data lifecycle, not just a single iRODS or YODA instance
- We propose to mitigate that complexity, adopting a hybrid mesh architecture. Where the external Research Data Mesh is coupled with a hub-and-spoke architecture within the organization.



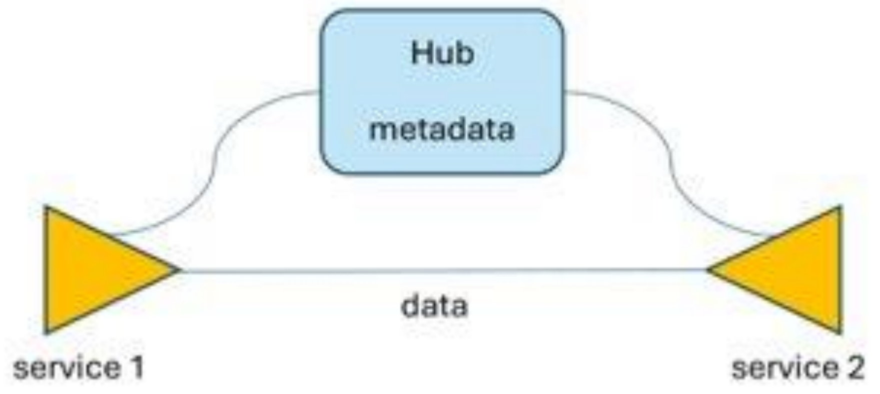
Hybrid Mesh architecture

- A hub-and-spoke design provides a central point, the hub, where it is possible to apply the governance rules and, in general, organization-wide data policies.
- This central point would represent the whole organization as node in the external Research Data Mesh, guaranteeing consistency in the exchange of data and metadata between the organization and the other nodes of the mesh.



Just metadata, please

- However, the hub in the hub-and-spoke architecture could become a bottleneck and a single point of failure.
- A redundant implementation can mitigate the latter weakness, but in order to remove the former one, we propose to limit the scope of the hub to metadata.
- In fact, any data workflow can be designed so that it is metadata driven, hence if we keep the metadata consistent, the governance and data policies rules will be consistent as well.
- On the other hand, the data will be transferred through a direct connection to the service hosting them, so that the performance and the scalability will be optimized.



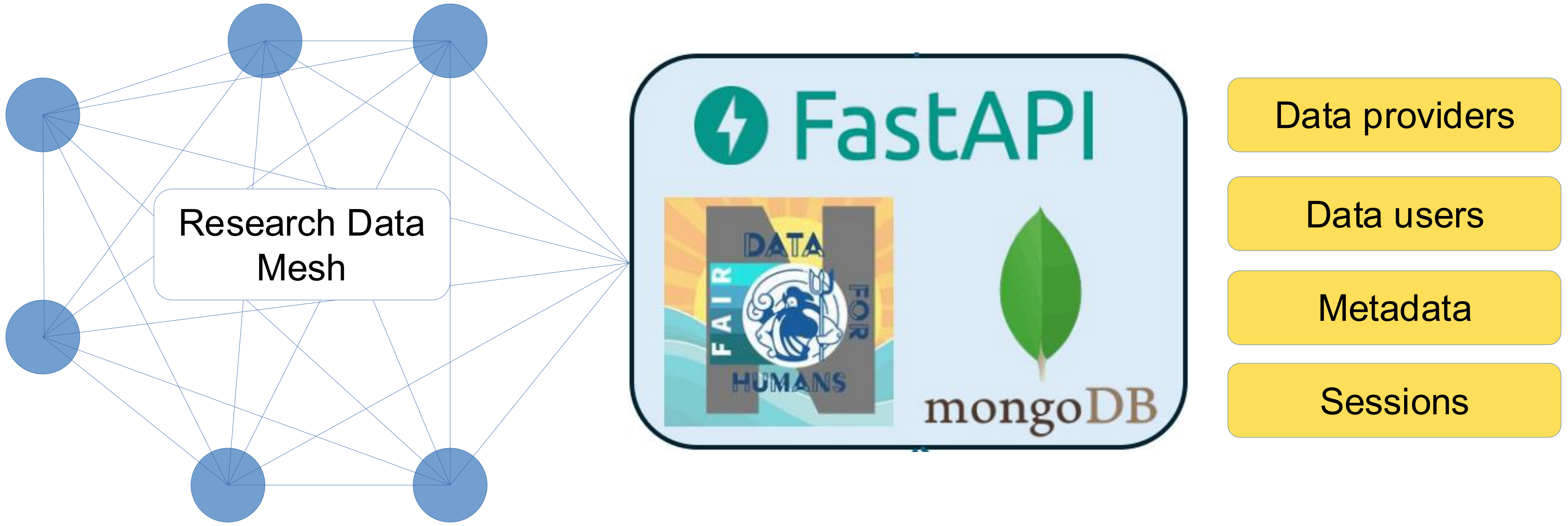
Research Data Hybrid Mesh node

Which features should offer the node of this Research Data Hybrid Mesh, being at the same time the hub of a hub-and-spoke topology?

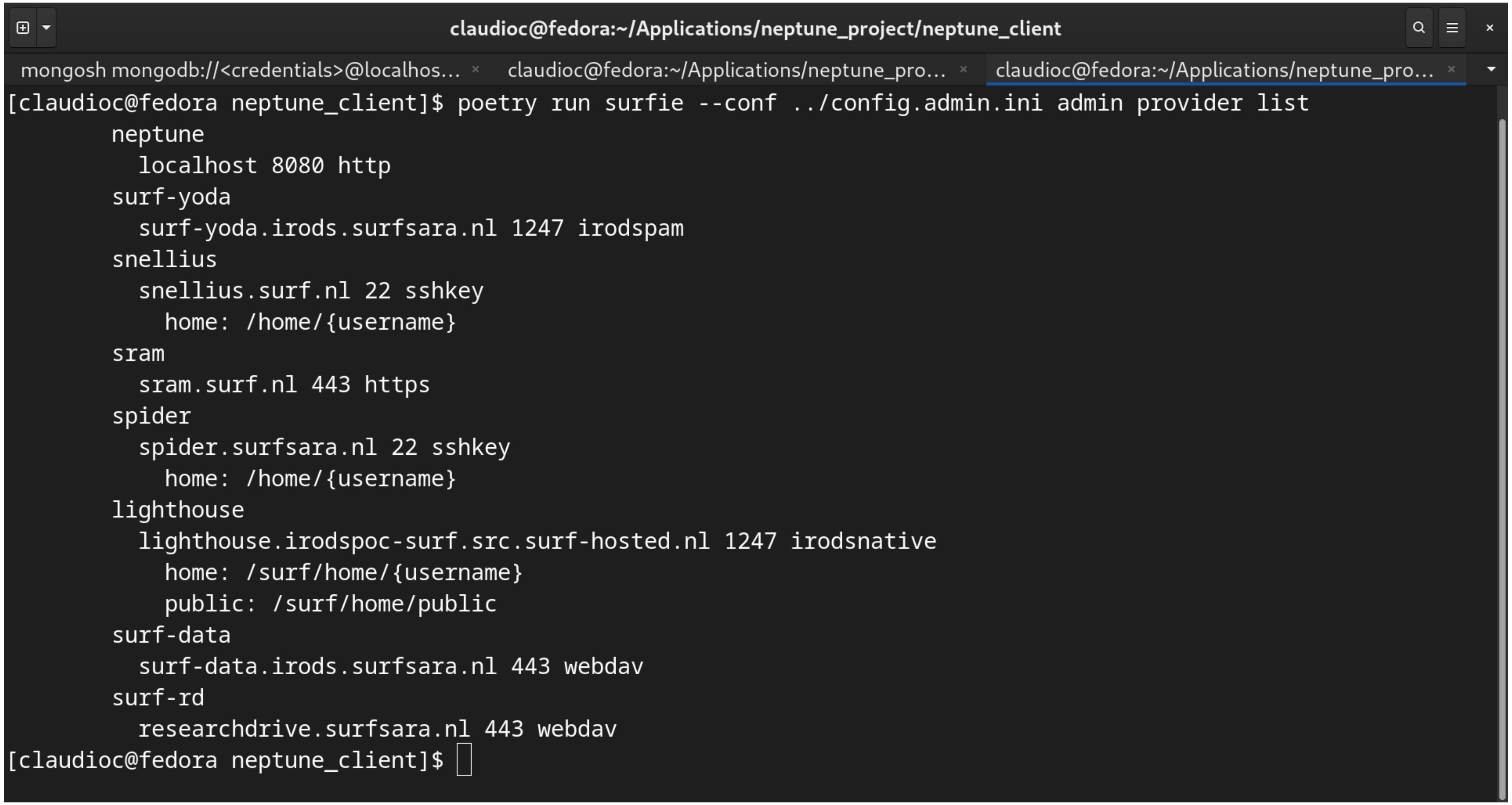
We propose that it is defined by the following properties:

- It is a **service registry**: it needs to be aware of all the services within the organization to advertise them to the other nodes of the mesh and support data workflows that use multiple services.
- It is a **user registry**: it needs to identify any “actor” that interacts with it to support decisions based on identity and role.
- It is a **metadata catalog**: the metadata related to users, providers and data are necessary to enforce policies based on them, automate as much as possible the decisions and to fulfill the definition of data product.
- It is an **API** gateway: it needs to route and control the API requests between the organization domain and the external world.
- It is a **data sharing** point: where the available data can be advertised to other users, provider, nodes … And where data can be discovered through metadata and requested from the data provider hosting them.

A prototype implementation: Neptune

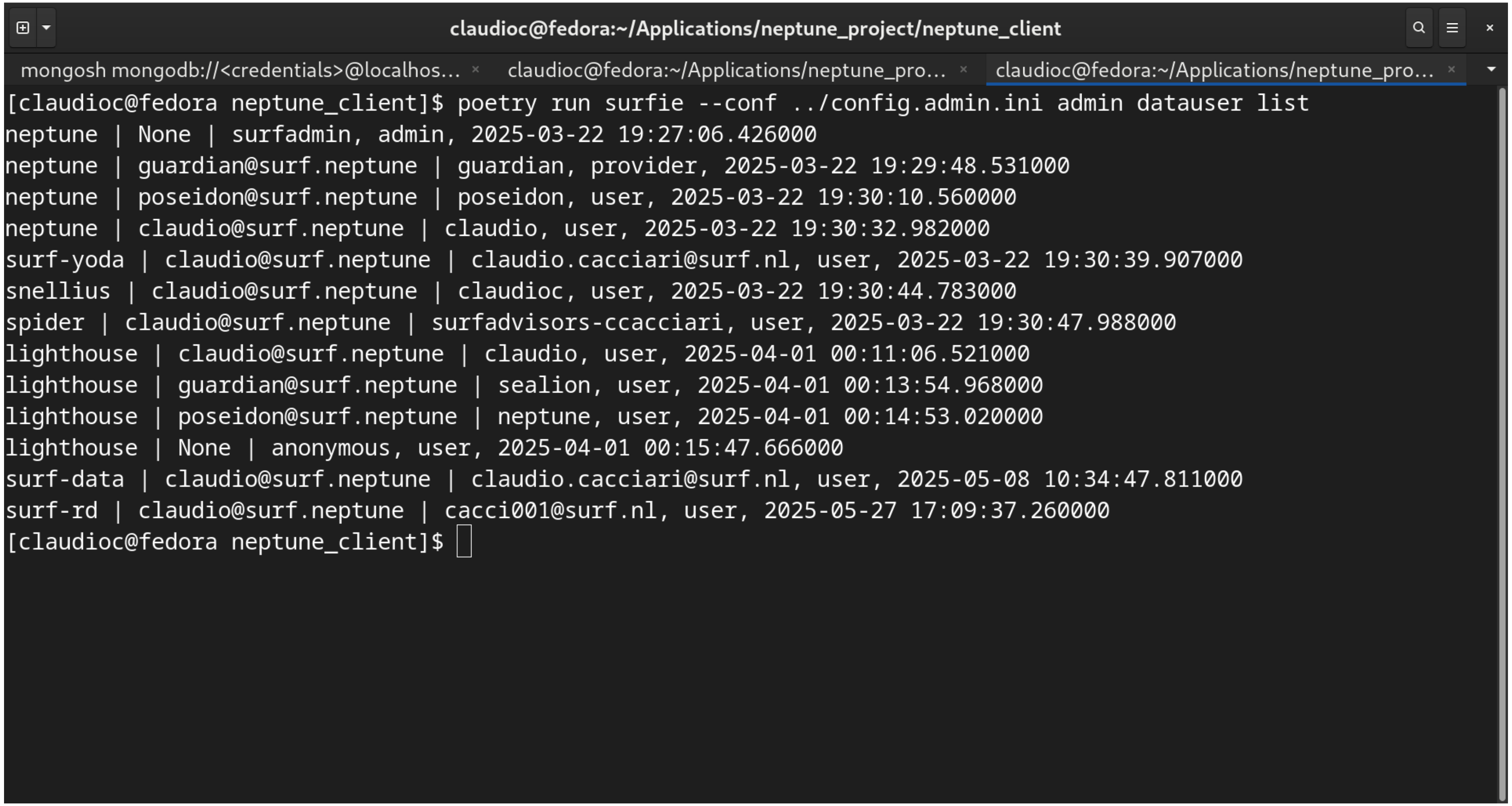


Service registry


claudioc@fedora:~/Applications/neptune_project/neptune_client
mongosh mongodb://<credentials>@localhos…
claudioc@fedora:~/Applications/neptune_pro…
claudioc@fedora:~/Applications/neptune_pro…

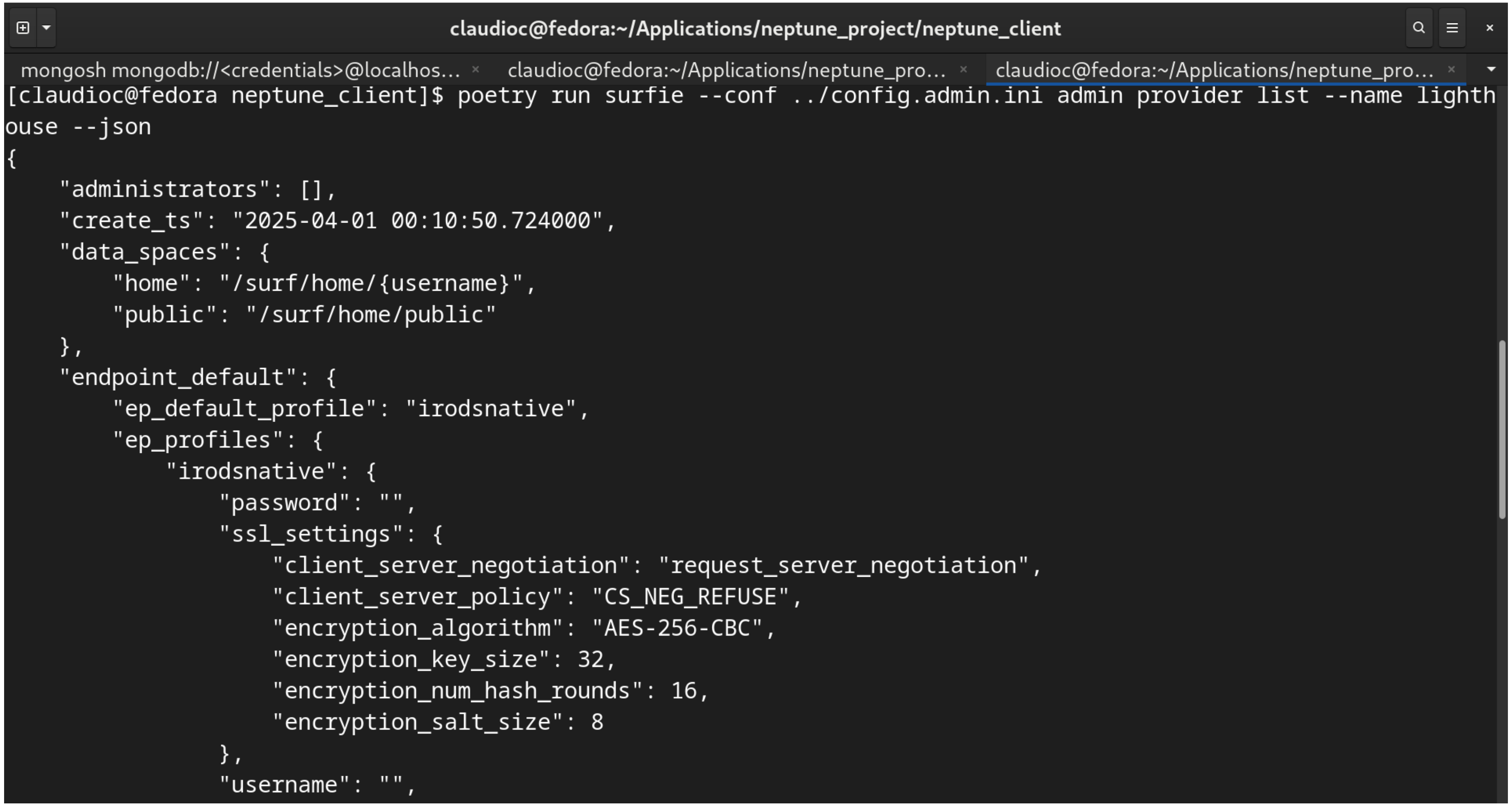
```
[claudioc@fedora neptune_client]$ poetry run surfie --conf ../config.admin.ini admin provider list
        neptune
          localhost 8080 http
        surf-yoda
          surf-yoda.irods.surfsara.nl 1247 irodspam
        snellius
          snellius.surf.nl 22 sshkey
            home: /home/{username}
        sram
          sram.surf.nl 443 https
        spider
          spider.surfsara.nl 22 sshkey
            home: /home/{username}
        lighthouse
          lighthouse.irodspoc-surf.src.surf-hosted.nl 1247 irodsnative
            home: /surf/home/{username}
            public: /surf/home/public
        surf-data
          surf-data.irods.surfsara.nl 443 webdav
        surf-rd
          researchdrive.surfsara.nl 443 webdav
[claudioc@fedora neptune_client]$ 
```

User registry


claudioc@fedora:~/Applications/neptune_project/neptune_client
mongosh mongodb://<credentials>@localhos…
claudioc@fedora:~/Applications/neptune_pro…
claudioc@fedora:~/Applications/neptune_pro…

```
[claudioc@fedora neptune_client]$ poetry run surfie --conf ../config.admin.ini admin datauser list
neptune | None | surfadmin, admin, 2025-03-22 19:27:06.426000
neptune | guardian@surf.neptune | guardian, provider, 2025-03-22 19:29:48.531000
neptune | poseidon@surf.neptune | poseidon, user, 2025-03-22 19:30:10.560000
neptune | claudio@surf.neptune | claudio, user, 2025-03-22 19:30:32.982000
surf-yoda | claudio@surf.neptune | claudio.cacciari@surf.nl, user, 2025-03-22 19:30:39.907000
snellius | claudio@surf.neptune | claudioc, user, 2025-03-22 19:30:44.783000
spider | claudio@surf.neptune | surfadvisors-ccacciari, user, 2025-03-22 19:30:47.988000
lighthouse | claudio@surf.neptune | claudio, user, 2025-04-01 00:11:06.521000
lighthouse | guardian@surf.neptune | sealion, user, 2025-04-01 00:13:54.968000
lighthouse | poseidon@surf.neptune | neptune, user, 2025-04-01 00:14:53.020000
lighthouse | None | anonymous, user, 2025-04-01 00:15:47.666000
surf-data | claudio@surf.neptune | claudio.cacciari@surf.nl, user, 2025-05-08 10:34:47.811000
surf-rd | claudio@surf.neptune | cacci001@surf.nl, user, 2025-05-27 17:09:37.260000
[claudioc@fedora neptune_client]$
```

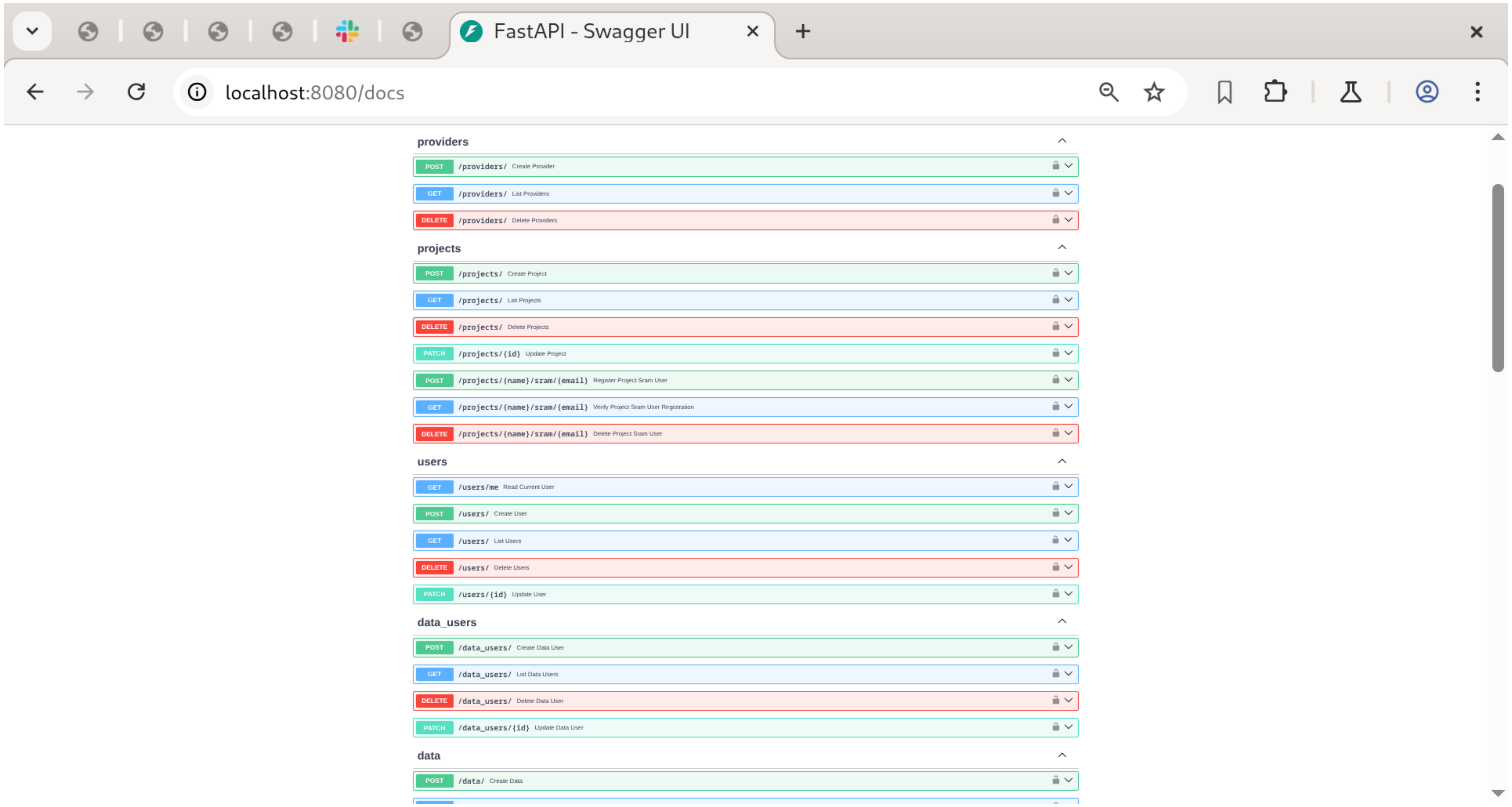
Data provider’s metadata



```
[claudioc@fedora neptune_client]$ poetry run surfie --conf ../config.admin.ini admin provider list --name lighth
ouse --json
{
    "administrators": [],
    "create_ts": "2025-04-01 00:10:50.724000",
    "data_spaces": {
        "home": "/surf/home/{username}",
        "public": "/surf/home/public"
    },
    "endpoint_default": {
        "ep_default_profile": "irodsnative",
        "ep_profiles": {
            "irodsnative": {
                "password": "",
                "ssl_settings": {
                    "client_server_negotiation": "request_server_negotiation",
                    "client_server_policy": "CS_NEG_REFUSE",
                    "encryption_algorithm": "AES-256-CBC",
                    "encryption_key_size": 32,
                    "encryption_num_hash_rounds": 16,
                    "encryption_salt_size": 8
                },
                "username": "",
```

claudioc@fedora:~/Applications/neptune_project/neptune_client
mongosh mongodb://<credentials>@localhos...
claudioc@fedora:~/Applications/neptune_pro...
claudioc@fedora:~/Applications/neptune_pro...

API



providers

POST `/providers/` Create Provider
GET `/providers/` List Providers
DELETE `/providers/` Delete Providers

projects

POST `/projects/` Create Project
GET `/projects/` List Projects
DELETE `/projects/` Delete Projects
PATCH `/projects/{id}` Update Project
POST `/projects/{name}/sram/{email}` Register Project Sram User
GET `/projects/{name}/sram/{email}` Verify Project Sram User Registration
DELETE `/projects/{name}/sram/{email}` Delete Project Sram User

users

GET `/users/me` Read Current User
POST `/users/` Create User
GET `/users/` List Users
DELETE `/users/` Delete Users
PATCH `/users/{id}` Update User

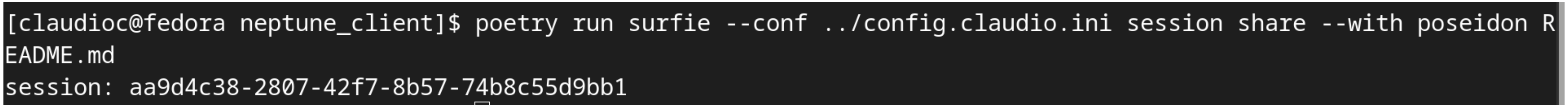
data_users

POST `/data_users/` Create Data User
GET `/data_users/` List Data Users
DELETE `/data_users/` Delete Data User
PATCH `/data_users/{id}` Update Data User

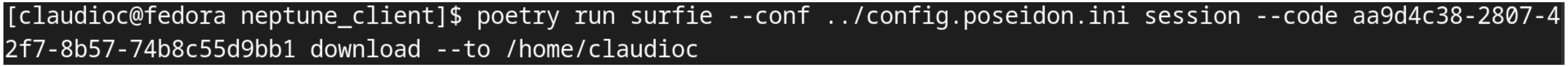
data

POST `/data/` Create Data

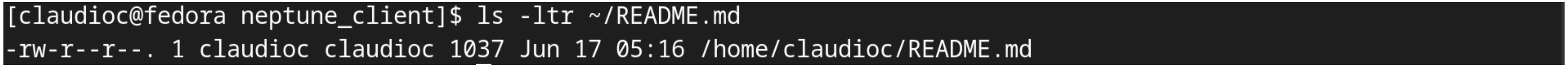
Data sharing



```
[claudioc@fedora neptune_client]$ poetry run surfie --conf ../config.claudio.ini session share --with poseidon R
EADME.md
session: aa9d4c38-2807-42f7-8b57-74b8c55d9bb1
```



```
[claudioc@fedora neptune_client]$ poetry run surfie --conf ../config.poseidon.ini session --code aa9d4c38-2807-4
2f7-8b57-74b8c55d9bb1 download --to /home/claudioc
```



```
[claudioc@fedora neptune_client]$ ls -ltr ~/README.md
-rw-r--r--. 1 claudioc claudioc 1037 Jun 17 05:16 /home/claudioc/README.md
```

Thank you for your attention!

SURF