

Minimum Requirements for a Project Charter

Module 3: Services & Repositories

Topic 1: EOSC Definitions - Services vs. Repositories



Overview of the Repositories & Services Table

Charter Section 3 is a structured table documenting your Node's services and repositories.

3. REPOSITORIES AND SERVICES DELIVERED

Maximum 1 page.

Describe the repositories/services (core functions, exchanges) to be made available by the new Node. Include the access policies for these repositories/services (e.g., specify the type of users who can access them: are they from your organisation only or from your country/discipline only or European-wide?).

Describe whether you intend to use or complement already existing federated capabilities or whether these services are a new addition to the Federation capabilities. See the EOSC Federation Handbook for details.

The resources made available to EOSC users via an EOSC Node must operate at minimum at Technical Readiness Level (TRL) 7 (self-assessment). (General criteria #4)

Service ID	Service Description	Access Policies to the Service	Federation Contributions & Value to Users	TRL

You must provide:

- **Minimum 5 entries** (can be services, repositories, or mix)
- **All five columns completed** for each entry
- **TRL 7 evidence** for operational readiness
- **Accurate capability mapping** to EOSC taxonomy

This is different from

Module 2

Strategic positioning ("what unique value do we bring?")

Module 3

Technical evidence ("what specific services do we operate?")

Key elements of the Table

Service ID	Description	Access Policy	Federation Contribution	TRL
Unique identifier with prefix (more in Topic 4)	User-focused explanation of what it does/holds	Who can use it and how (Topic 3)	Which EOSC capability it provides (Topic 3)	Always 7 with evidence (Topic 2)

What evaluators check:

- Are services clearly described?
- Is TRL 7 credible with evidence?

- Do access policies match service types?
- Are capabilities accurately mapped?
- Does this align with Section 2 value proposition?

Defining Services vs. Repositories

	What is a Service?	What is a Repository?
EOSC Definition	A service provides functionality to users - it performs actions, processes data, or enables capabilities.	A repository provides storage and preservation - it holds, maintains, and provides access to research objects.
Key characteristics	Active functionality: Does something for the user	Storage function: Preserves digital objects over time
	User interaction: Users invoke, configure, or query the service	Metadata provision: Describes holdings for discovery
	Ongoing operation: Runs continuously or on-demand	Access management: Controls who can retrieve what
	Outputs deliverables: Produces results, analyses, or transformed data	Persistent identifiers: Assigns DOIs or handles
Examples	Data discovery service: Allows users to search across repositories	Data repository: Stores research datasets with metadata
	Authentication service: Enables federated login across platforms	Publication repository: Holds and provides access to papers
	Analysis service: Processes datasets and returns visualisations	Software repository: Archives code with version control
	Workflow orchestration: Executes multi-step research pipelines	Metadata catalogue: Maintains descriptive records
	Training service: Delivers educational content and certifications	Digital preservation archive: Long-term bit-level preservation

Services vs. Repositories: The Distinction

The key question: Does it *do* something (service) or *hold* something (repository)?

Aspect	Service	Repository
Primary function	Provides functionality	Provides storage
User interaction	Invokes actions	Deposits/retrieves objects
Example verbs	Analyse, process, authenticate, orchestrate	Store, preserve, archive, catalogue
Outputs	Results, analyses, credentials	Stored objects, metadata records
EOSC role	Enables research workflows	Preserves research outputs

Common Cases

1. **Data Portal** = Web interface for discovering and downloading datasets

Q

Service or repository?

A

Depends on primary function

- If it **stores datasets** → Repository
- If it **searches across multiple repositories** → Service (data discovery service)
- If it does both → List separately:
 - Repository: "Marine Data Repository"
 - Service: "Marine Data Discovery Service"

2. **Jupyter Hub** = Platform providing computational notebooks

Q

Service or repository?

A

Service (computational service)

- Provides functionality (code execution, visualisation)
- Users interact to run analyses
- Not primarily about storage/preservation

Common Cases

3. Metadata Catalogue = Searchable index of dataset descriptions

Q

Service or repository?

A

Repository (metadata repository)

- Stores metadata records
- Primary function is preservation and access to descriptions
- Even though searchable, storage is the core function

4. AAI Infrastructure = Authentication and authorisation system

Q

Service or repository?

A

Service (authentication service)

- Provides functionality (identity verification, access control)
- Active processing of credentials
- Not about storing research objects

Examples from First-Wave Charters

eOSC Node

Life Sciences Connect

Health & Food

Services listed

- Data Submission Service (enables depositing to repositories)
- Data Discovery Service (searches across life science data)
- Federated Authentication Service (ELIXIR AAI)
- Workflow Execution Service (Galaxy, Nextflow platforms)

Repositories listed

- European Nucleotide Archive (ENA) - sequence data repository
- ArrayExpress - functional genomics repository
- MetaboLights - metabolomics repository
- BioStudies - multi-domain repository

They separated "discovery" (service) from "storage" (repositories) even though both involve data

Examples from First-Wave Charters

eOSC Node

SURF

The Netherlands

Services listed

- SURF Research Drive (cloud storage and collaboration service)
- SURF Data Archive (long-term data archiving service)
- National AAI Infrastructure (SURFconext federation service)
- Research Cloud (computational infrastructure service)

Repositories listed

- DANS EASY (social sciences data repository)
- NARCIS (Dutch research information repository)

Even storage offerings were framed as "services" when they provide active functionality beyond basic storage

What NOT to Include

Infrastructure components that aren't services/repositories:

 • **Server hardware**

- "Dell PowerEdge servers" → This is infrastructure, not a service users interact with

• **Network connectivity**

- "10Gb Ethernet backbone" → Enabling infrastructure, not a user-facing service

• **Internal tools**

- "Staff ticketing system" → Not for EOSC users, not in service table

• **Planned/future services**

- "AI recommendation engine (in development)" → Only include TRL 7 operational services

DO include:

Services and repositories that users can access or use today.

Your Section 3 services should reflect Section 2 claims:

Good Example

Section 2 claimed:

"Our Node brings 20-year marine metadata expertise to address Central Europe's standardised data infrastructure gap"

Section 3 must show:

- Marine Data Repository (demonstrating metadata expertise)
- Data Discovery Service (addressing the infrastructure gap)

Both at TRL 7 (operational, not aspirational)

Bad Example

Section 2 claimed:

"We provide quantum computing access for European researchers"

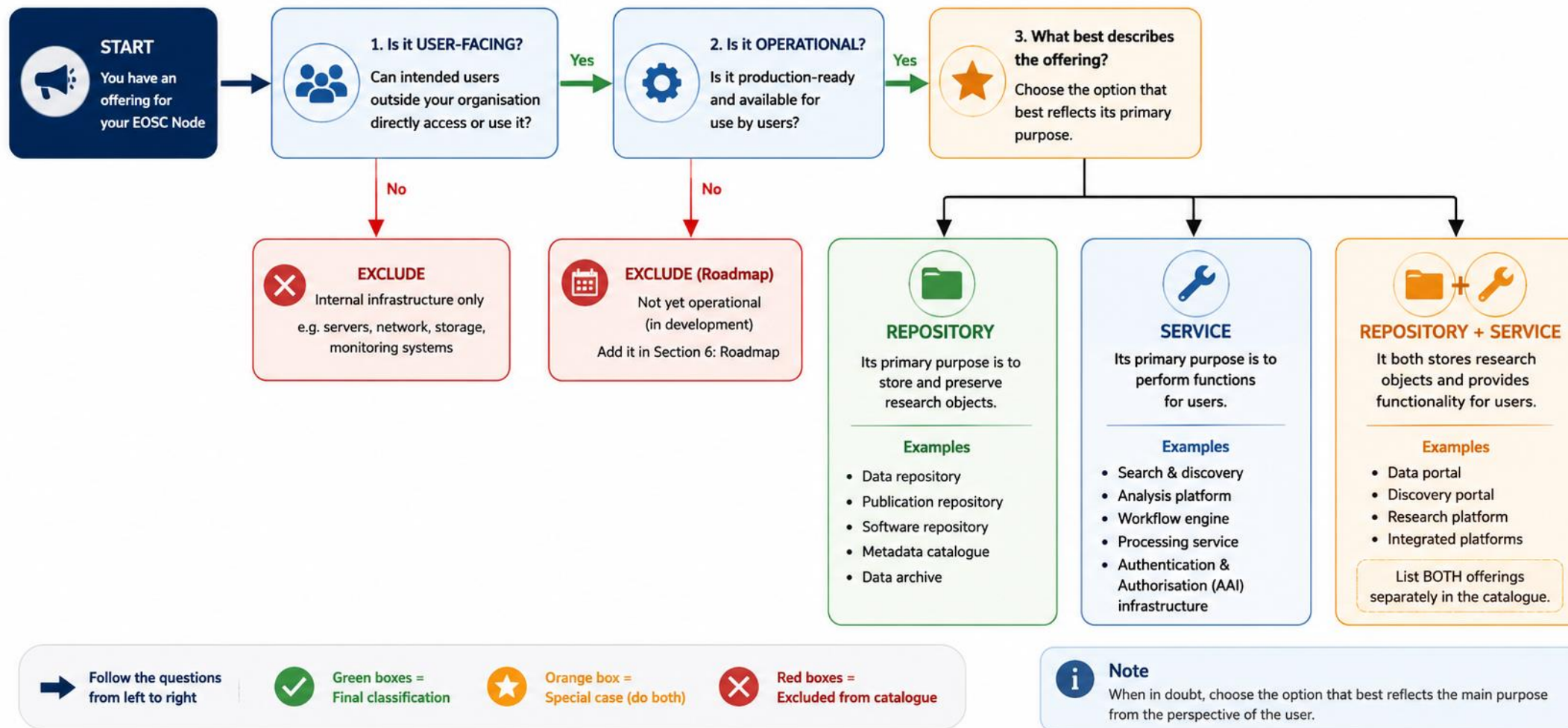
Section 3 shows:

Only traditional HPC services, no quantum

Evaluators will question credibility

Section 3 is the evidence for Section 2 claims

Use this Decision Path to Select your Offering



Run each offering through the decision path above (slide 11) and write down your answers.

Offering 1

Name:

- User-facing? ☐ Yes → continue ☐ No → exclude, stop here
- Operational (TRL 7+)? ☐ Yes → continue ☐ No → note for Section 6 (Roadmap), stop here
- Primary function: ☐ Storage only ☐ Functionality only ☐ Both ☐ Neither

Final category: ☐ Repository ☐ Service ☐ Ambiguous (list both) ☐ Excluded

Brief justification: _____

Offering 2...5 (repeat the same steps for each)

Save these categorisations. You will use them to build your service table in Topic 4.

Thank you!