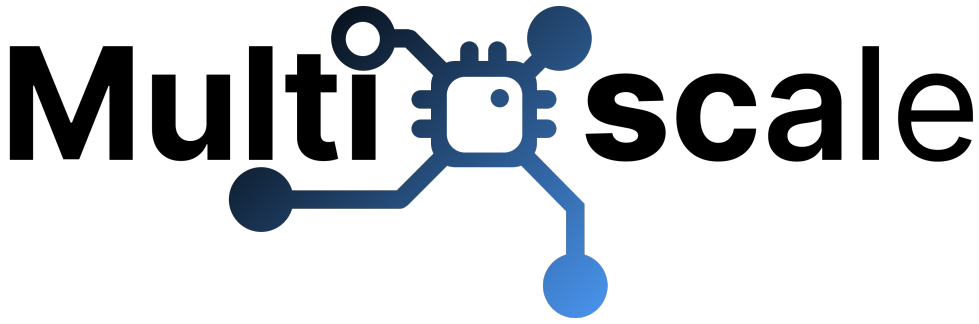

Support portal

MultiXscale Deliverable 5.2
Deliverable Type: Other
Delivered in December, 2023

MultiXscale
EuroHPC Centre of Excellence for
Multiscale Modelling

Co-funded by
the European Union

Acknowledgement
Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) under grant agreement No 101093169.

Project and Deliverable Information

Project Title	MultiXscale: EuroHPC Centre of Excellence for Multiscale Modelling
Project Ref.	Grant Agreement 101093169
Project Website	https://www.multixscale.eu
EuroHPC Project Officer	Dr. Linda Gesenhues
Deliverable ID	D5.2
Deliverable Nature	Other
Dissemination Level	Public
Contractual Date of Delivery	Project Month 12(31st December, 2023)
Actual Date of Delivery	28th December, 2023
Description of Deliverable	Report on alternatives that were considered for support portal and motivated selected solution + working support portal and accompanying documentation.

Document Control Information

Document	Title:	Support portal
	ID:	D5.2
	Version:	As of December, 2023
	Status:	Accepted by Steering Committee
	Available at:	https://www.multixscale.eu/deliverables
	Document history:	Internal Project Management Link
Review	Review Status:	Reviewed
Authorship	Written by:	Kenneth Hoste (Ghent University)
	Contributors:	Lara Peeters (Ghent University), Thomas Röblitz (University of Bergen)
	Reviewed by:	Alan O'Cais (University of Barcelona)
	Approved by:	Matej Praprotnik (National Institute of Chemistry)

Document Keywords

Keywords:	MultiXscale, HPC, EESSI, support portal

28th December, 2023

***Disclaimer**:This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements.*

[1] kenneth.hoste@ugent.be

Contents

List of Figures

List of Tables

Executive Summary

In the scope of Task 5.1 of the MultiXscale project, a support portal was set up for the shared software stack, which is being developed in Task 1.1 of the project.

Project partners were consulted to share their experience with various support portals. This led to a set of objectives, from which a list of requirements and nice-to-have features were derived for a support portal in the context of this project.

An initial set of candidate solutions for setting up a support portal was selected, including Bugzilla, GitHub, GitLab, Jira, OTRS, Redmine, Request Tracker, and Trac. These were thoroughly evaluated in terms of the outlined requirements and desired features, which resulted in an objective comparison that takes into account the features and limitations of each candidate solution.

This process revealed that *GitLab* is currently the best suited alternative among the considered solutions for setting up a support portal for the shared software stack.

To solidify this result further, a test setup of the support portal using the free hosted offering of GitLab was assessed by project partners to verify that it aligns with the objectives that were set out. This led to the unanimous conclusion that a support portal using GitLab as underlying technical solution was indeed fully able to meet the expectations. The feedback that arose during this process was taken into account when setting up the production version of the support portal at `https://gitlab.com/eessi/support`. Documentation focused on end users that outlines how to submit a support request was made available, along with a clear description of the provided support level.

A test support rotation among project partners who will be involved in Task 5.4 was executed from October-December 2023. Each project partner was made principally responsible for managing and processing support requests for 2 weeks, and weekly support meetings were held to syncronize among support team members. This setup was found to work well, and the support rotation planning was extended into 2024 to prepare for Task 5.4.

At the time of writing, about 25 support tickets had been opened in the support portal for the shared software stack by a dozen different people, mostly corresponding to partners in the MultiXscale project, but also beyond it, including for example the Square Kilometre Array Observatory (SKAO) project.

1 Introduction

1.1 Scope of the deliverable

This document covers the effort on setting up a support portal for the shared software stack for the MultiXscale project, along with defining the level of support that will be provided.

We start by defining the support team and giving a high-level overview of how the effort was planned and executed in Section 2. Section 3 defines the objectives for the support portal, and derives a set of requirements and desirable (but not essential) features. These are used in Section 4 to compare a select set of candidate solutions, which leads to recognising GitLab as best suited candidate solution.

Further evaluation by project partners through a test setup is covered in Section 5, before presenting the production setup of the support portal in Section 6. In Section 6.5 we outline the preparation that was done for Task 5.4 by setting up an experimental support rotation, before concluding in Section 7.

1.2 Target audience

The support portal and accompanying findings presented in this report are intended primarily for the partners of the MultiXscale project who will need to provide support for the shared software task that is developed in the scope of the project, as well as for the end users of the shared software stack.

2 Organisation of the effort

In this section, we define the support team who will eventually manage and use the support portal for the shared software stack, and give a high-level overview of how the effort for setting up a support portal was planned and executed.

2.1 Support team

Providing support for the shared software stack, which is being developed in Work Package (WP) 1, is the goal of Task 5.4. The coordination of this task is the responsibility of Ghent University (UGent), who is also the lead beneficiary of WP5. The support team is completed by additional project partners including SURF, HPCNow!, Rijksuniversiteit Groningen, Universitat de Barcelona, and Universitetet i Bergen.

The objective of Task 5.1 was to set up a support portal, along with a description of the level of provided support, and a define a support rotation among project partners, to facilitate the effort planned in Task 5.4.

In this deliverable we will refer to the MultiXscale project partners involved in Task 5.4 as *support team members*. They will be responsible for processing support requests, and hence they will be the primary users of the support portal (next to end users who create support requests).

2.2 Planning and execution

The effort covered by this report was planned and executed efficiently and effectively, in a series of steps:

- Identify objectives & requirements for the support portal, by consulting support team members (Section 3);
- Select an initial set of candidate solutions, and compare them based on requirements (Section 4);
- Evaluate primary candidate solution(s), and collect feedback from support team members (Section 5);
- Install the production setup of the support portal (Section 6);
- Test support rotation schedule (Section 6.5);

3 Objectives and requirements

This section describes the objectives for the support portal, and derives requirements that must be met by it.

3.1 Objectives for the support portal

The support team members have extensive experience in providing support to their respective users in various scenarios. Key considerations for organising an efficient and effective support service include the organisation of the support team (centralised versus distributed), the technical experience of the user base (homogeneous versus diverse), the anticipated duration of the support service (fixed-term versus long-term), and the nature of the infrastructure or service being supported (simple versus complex).

Based on the experience of the support team members and inspired by existing in-house and external support services, for example the support portal for Slurm, we identified the following key objectives regarding the support portal for the shared software stack:

OBJ-1 Facilitating **high-quality** and **responsive** support to end users, contributors, and fellow team members;

OBJ-2 Ensuring a **low threshold** to ask questions, report problems, or propose changes and enhancements;

OBJ-3 Catering to **different scenarios**: reporting problems, requesting additional software, asking questions, etc.;

OBJ-4 Enabling **efficient collaboration** among the members of the distributed support team;

OBJ-5 Providing effective means to **build up and share knowledge** from resolved support cases;

OBJ-6 Using a **sustainable** technical solution, which requires limited cost and effort to operate and maintain;

OBJ-7 Being able to **control visibility** based on the sensitivity of support cases (e.g., security concerns).

3.2 Requirements for a technical solution

In this section, we derive the requirements that a technical solution must meet to achieve the objectives listed above.

3.2.1 (REQ-1) Features to facilitate efficient processing of support requests

The support portal should provide an easy way to search through existing support requests, organise and filter support requests in various ways, and assign and re-assign support requests to support team members.

This helps achieving objectives OBJ-1, OBJ-3, OBJ-4, and OBJ-5.

3.2.2 (REQ-2) Support for controlling visibility of support requests

Support requests for the shared software stack must be processed in public. This can empower end users by being able to see what other users struggle with. It also allows them to potentially find an answer to their questions if they have been raised by others previously, which can reduce the total number of support requests.

It is important however to keep in mind that having *all* support requests publicly available could be counter-productive. Consider for example reports of security-related issues, or relatively inexperienced end users who may be reluctant to ask their question publicly.

As a result, it is imperative that the visibility of support requests can be controlled case-by-case, both by support team members and the creator of a support request, and that sensible defaults are used where appropriate.

This requirement help achieving objectives OBJ-2, OBJ-3, OBJ-4, OBJ-5, and OBJ-7.

3.2.3 (REQ-3) Intuitive interface for support team members and end users

The user interface of the support portal must be easy to understand and use, both for support team members and end users. Ideally it is intuitive enough such that no training effort is required to use and manage the support portal, and that providing basic documentation is sufficient. A user interface that is similar to that of established platforms like GitHub to which many end users and support team members are likely to be familiar with, would be beneficial.

This requirement supports the objectives OBJ-2 and OBJ-6.

3.2.4 (REQ-4) Low operational cost

Although several project partners would be able to host the support portal on their existing in-house cloud infrastructure should the need arise, there is a strong incentive to use a low-cost hosted solution since that helps significantly with reducing the cost of operating and maintaining the support portal.

The maintenance and upkeep of the support portal should not consume a significant amount of time and effort that could be better spent on processing support requests. If a solution is at some point no longer maintained by the original provider, this could leave a significant burden on the support team to maintain it, which would interfere with providing good support.

This requirement supports achieving objective OBJ-6.

3.2.5 (REQ-5) Actively maintained project

The technical solution used for implementing the support portal must be an actively maintained project. This way, we can be more confident that the support portal can be kept operational for the duration of the project (and beyond), at a reasonable cost and effort.

Using an actively maintained solution is important not only with respect to making sure that the setup can be kept secure, but also to decrease the likelihood that migrating to another technical solution may be required. This would imply that both support team members and end users need to familiarize themselves again with the new setup, which should be avoided.

This requirement helps achieving objectives OBJ-2 and OBJ-6.

3.2.6 (REQ-6) Flexibility to transition to another setup

While we do not anticipate to migrate the support portal to another setup in the short-term, we must be able do so should the need arise.

The support portal must provide a means (directly via an export mechanism, via an API, etc.) to export all support requests and corresponding replies along with all relevant metadata (associated labels, support team members involved, etc.). This enables transitioning to another setup if and when required.

Reasons for having to migrate to another setup could be multi-fold, including hitting the limitations in terms of capabilities or available resources of the current setup which hinder the support effort (see REQ-1), a significant increase in cost (see REQ-4), a lack of sufficient maintenance of the underlying solution (see REQ-5), etc.

By making sure that the adopted solution allows for easy migration to another setup, we effectively avoid being locked-in to the initial setup of the support portal.

This requirement supports the objectives OBJ-4 and OBJ-6.

3.3 Optional features for a technical solution

In this section, we outline a couple of nice-to-have features for the support portal.

These were deemed optional: although they would increase the value of a particular setup, they are not strictly necessary in order to meet the objections defined in Section 3.1.

3.3.1 (FEAT-1) Support for creating and following up on support requests via email

The ability to send support requests via email and to reply to inquiries via email may simplify the interactions between end users and the support team. In addition, it could lower the threshold to ask for help.

This feature could help support objectives OBJ-1 and OBJ-2.

3.3.2 (FEAT-2) Support for using templates to create and reply to support requests

Templates for creating a support request help guide end users on what information should be provided when asking for help. This enables the support team to provide higher quality support more quickly.

Similarly, templates for replies to support requests can help the support team to provide quality-checked answers to common questions.

This feature helps achieving objectives OBJ-1, OBJ-2, OBJ-3, and OBJ-5.

3.3.3 (FEAT-3) E-mail notifications

Providing e-mail notifications, for example, if a new support request has been created or an update to an existing support case was made, can help to foster smooth follow-up of support requests by both the support team members and the users.

This feature supports the objectives OBJ-1 and OBJ-4.

3.3.4 (FEAT-4) Integrated documentation for the support team

The support team should have easy access to documentation that provides guidelines to handle support requests efficiently, which should (ideally) be integrated into the support portal itself.

This feature helps in achieving the objectives OBJ-1, OBJ-4, and OBJ-5.

4 Selection and comparison of candidate solutions

In this section, we describe the process that was followed to select and compare candidate solutions for setting up the support portal for the shared software stack, which eventually led to identifying a single well suited candidate solution.

4.1 Initial selection of candidate solutions

There is a large variety of potential solutions for setting up a support portal. Doing a thorough analysis and comparison of all possible candidate solutions was deemed to be an unreasonable effort, so we focused on evaluating solutions that the support team members were already somewhat familiar with.

If one or more of the solutions in this initial selection would meet the requirements that were outlined in Section 3.2, we could proceed with only evaluating those more in-depth. If that would not be the case, which seemed unlikely based on a rough assessment, we could still consider evaluating additional solutions.

4.2 Overview of the candidate solutions

The project partners who will be involved in Task 5.4 are all part of support teams at their own organisation.

Collectively, they use or are familiar with the following projects (alphabetically ordered): Bugzilla, GitHub, GitLab, Jira, OTRS, Redmine, Request Tracker, and Trac.

Each of these projects was considered to be a candidate solution for setting up the support portal for the shared software stack. We briefly describe each of them in the following subsections.

4.2.1 Bugzilla

Bugzilla is an open-source software project designed to drive software development, and is generally used as a bug-tracking system. It provides a rather minimalistic user interface.

There is no hosted offering provided by the Bugzilla project itself, a self-managed setup is required. Bugs can be created and followed up on via e-mail. Alternatively, an account can be created in the particular Bugzilla instance being used to subscribe to updates for particular bugs. The visibility of issues in Bugzilla can be controlled via user groups. Partners in the MultiXscale project only have limited experience with Bugzilla.

Bugzilla is actively maintained, and is used by a wide range of open-source software projects, for example GCC, Gentoo Linux, and Slurm. The latter in particular was an initial inspiration for the support portal for the shared software stack, because it aligns well with many of the requirements outlined in Section 3.2.

4.2.2 GitHub

GitHub is a popular developer platform, in particular for open-source software projects.

It offers several hosted tier plans ranging from free to enterprise, which provide features that cater to the needs of different types of projects and organisations. GitHub can also be self-hosted via the commercial GitHub Enterprise Server offering.

While accessing public information in a GitHub repository can be done by anyone, some actions like creating an issue (which could be equivalent to a support request) do require a GitHub account. Issues in a GitHub repository can only be fully public, or fully private (by using repository that is entirely private).

GitHub is actively maintained and is used by many (open-source) software projects that serve large distributed communities. Most partners in the MultiXscale project have extensive experience with GitHub.

4.2.3 GitLab

GitLab is an open-source developer platform, which is similar to GitHub in terms of user experience.

GitLab projects can be created in the free hosted offering, or through the premium GitLab Enterprise Edition license that provides additional features. As an open-source project, it can also be self-managed in various ways, without feature limitations.

Issues in a GitLab project can be marked as confidential, such that they are only accessible to members of the project (which could be only support team members, for example). Through the Service Desk feature that GitLab supports, issues can be created and followed up via email, without requiring a GitLab account.

GitLab is actively maintained and used by many (mostly open-source) projects serving large distributed communities. Most partners in the MultiXscale project have experience with using GitLab.

4.2.4 Jira

Jira is a suite of several products, including a pure issue tracking solution and a dedicated service desk solution.

Jira products are available in hosted tier plans and for self-hosting. Hosted tier plans range from free for up to 10 users (support staff) to enterprise (up to 35,000 support staff). The free plan is also limited in features, while the paid tiers provide additional capabilities. Creating issues (support requests) does not require an account, and issues can either be fully public or fully private.

Jira is actively maintained and integrates with various third-party services. Most partners in the MultiXscale have experience with Jira products.

4.2.5 OTRS

OTRS is a service management solution which can be used to manage support to customers.

It is provided as a managed hosted solution with several paying tiers, or can be run on-premise. Pricing information is not publicly available. End users do not need to have an account to create a support request, and support requests can be either fully public or fully private.

OTRS is actively maintained. Partners in the MultiXscale project only have limited experience with OTRS.

Znuny is an open-source fork of the previously available OTRS Community Edition, which was available with a free hosted offering. Znuny requires using a self-hosted setup. It provides similar features as OTRS regarding ticket creation and visibility.

4.2.6 Redmine

Redmine is a flexible project management web application. It is available under an open-source license, and must be self-hosted and -managed.

End users do not need an account to create support requests, which can be either fully public or fully private.

Redmine is actively maintained. Partners in the MuliXscale project have limited experience with Redmine.

4.2.7 Request Tracker

Request Tracker (RT) is a tool to track and manage workflows, customer requests, and internal project tasks.

It is released under an open-source license. The fully featured version can be self-managed. It is also offered as a hosted solution, but only with payed tiers. Professional support plans exist for self-managed setups. End users do not need an account to create support requests, which can be either fully public or fully private.

Request Tracker is actively maintained. Partners in the MuliXscale project have limited experience with Request Tracker.

4.2.8 Trac

Trac is an open source, web-based project management and bug tracking system.

Trac is released under an open-source license, and must be self hosted. End users do not need an account to create support requests, which can be either fully public or fully private.

At the time of assessment (May 2023), Trac was not actively maintained: there were no published releases for several months (since then, maintenance seems to have resumed though, with new releases as recent as December 2023). Partners in the MuliXscale project have limited experience with Trac.

4.3 Comparison of candidate solutions

We evaluated and compared the candidate solutions that were considered with respect to our requirements (see Section 3.2) and desired optional features (see Section 3.3). We describe our findings for each of these, and provide a summary of the results in Table 1. In the subsections below as well as in Table 1, we use the same markers for indicating a positive evaluation (✓), a negative evaluation (×) and a neutral evaluation (N).

	Bugzilla	GitHub	GitLab	Jira	OTRS	Redmine	RT	Trac
(REQ-1) Features for efficient processing	✓	✓	✓	✓	✓	✓	✓	✓
(REQ-2) Visibility of support requests	✓	×	✓	×	×	×	✓	×
(REQ-3) Intuitive user interface	✓	✓	✓	✓	✓	✓	✓	✓
(REQ-4) Low operational cost	N	✓	✓	×	N	N	N	N
(REQ-5) Actively maintained project	✓	✓	✓	✓	✓	✓	✓	×
(REQ-6) Transitioning to other setup	✓	✓	✓*	✓	✓	✓	✓	✓
(FEAT-1) Support request via e-mail	✓	×	✓	✓	✓	✓	✓	✓
(FEAT-2) Templates for requests/replies	✓	✓	✓	✓	✓	✓	✓	✓
(FEAT-3) E-mail notifications	✓	✓	✓	✓	✓	✓	✓	✓
(FEAT-4) Integrated documentation	×	✓	✓	×	×	✓	×	✓
Experience level of project partners	N	✓	✓	✓	N	×	N	×

Table 1: Comparison of candidate solutions considered for support portal based on requirements and nice-to-have features; ✓ indicates positive evaluation, × indicates negative evaluation, N indicates neutral evaluation.; the special ✓* marker for REQ-6 and GitLab is explained in Section 4.3.5.

4.3.1 Non-discriminating requirements and features

Several of the requirements and nice-to-have features outlined in Sections 3.2 and 3.3 turned out to be non-discriminatory, since they are supported by all considered solutions (marked ✓ in Table 1). This includes:

- (REQ-1) Features to facilitate efficient processing of support requests
- (REQ-3) Intuitive interface for both support team members and end users
- (FEAT-2) Support for using templates to create and reply to support requests
- (FEAT-3) E-mail notifications

While requirement REQ-6 is also evaluated as non-discriminatory, it is discussed in detail separately in Section 4.3.5.

4.3.2 (REQ-2) Support for controlling visibility of support requests

We found that only Bugzilla, GitLab, and Request Tracker (marked ✓ in Table 1) provide control over the visibility of support requests, by letting support team members specify whether or not a particular support request should be publicly accessible. For all other candidate solutions (marked × in Table 1), support requests can only be either all public, or all private.

As such, this requirement turned out to be a significant factor in the comparison of candidate solutions.

4.3.3 (REQ-4) Low operational cost

Open-source solutions that do not provide a free hosted offering (which includes Bugzilla, Redmine, Request Tracker, and Trac) were evaluated as neutral (marked N in Table 1) regarding the requirement to have a low operational cost, since these would require to set up and maintain the solution ourselves. OTRS was also evaluated neutrally regarding this requirement, since the open-source fork Znuny (which needs to be self-managed) provides an alternative to the paid offerings provided by OTRS.

Jira was evaluated negatively (marked × in Table 1) because only their paid offering would be sufficient in terms of limitations and features.

The free hosted offerings of both GitHub and GitLab (marked ✓ in Table 1) are sufficient in terms of available features and capabilities, and hence were evaluated positively.

4.3.4 (REQ-5) Actively maintained project

All candidate solutions (marked ✓ in Table 1) except for Trac (marked × in Table 1) were actively maintained at the time of assessment (May 2023).

4.3.5 (REQ-6) Flexibility to transition to another setup

For candidate solutions that provide both free and paid hosted offerings, transitioning between these options is assumed to be friction-less. Most considered solutions also provide a way to export all data in some particular format, and to import it again in another setup using the same solution. Hence all candidate solutions are evaluated positively (marked ✓ in Table 1).

GitLab (marked ✓* in Table 1) was considered to have an extra edge regarding this requirement, because it provides detailed documentation on migrating between hosted and self-managed GitLab instances, and importing/exporting projects via their API. In addition, being able to set up a self-managed instance using the open-source offering of GitLab provides additional flexibility, should the need arise to transition to another setup.

4.3.6 (FEAT-1) Support for creating and following up on support requests via email

All but one of the candidate solutions (marked ✓ in Table 1) provide built-in support for creating support requests via email without requiring an account, except for GitHub (marked × in Table 1).

4.3.7 (FEAT-4) Integrated documentation for the support team

Half of the candidate solutions (GitHub, GitLab, Redmine, and Trac, marked ✓ in Table 1) provide built-in wiki-like functionality that can be used for integrated documentation. The others (Bugzilla, Jira, OTRS, and Request Tracker; marked × in Table 1) do not provide support for built-in documentation.

4.3.8 Experience level of project partners

MultiXscale project partners involved in this effort have extensive experience with 3 out of 8 candidate solutions being considered (GitHub, GitLab, Jira; ✓ in Table 1). For 3 other candidate solutions (Bugzilla, OTRS, Request Tracker; N in Table 1) only a small minority of partners have experience, while for the two others (Redmine and Trac, × in Table 1) there is little to no (recent) experience.

Although this is a rather minor aspect, since support team members can always familiarize themselves with a solution that is new to them, it is relevant with respect to several requirements, including REQ-3 and REQ-4.

4.4 GitLab as best candidate solution for setting up support portal

The comparison presented in Section 4.3 indicates that **GitLab** is the best alternative among the candidate solutions that were considered.

All requirements that were outlined in Section 3.2 are clearly met by GitLab, and the optional features covered in Section 3.3 are supported as well. The free hosted offering provided by GitLab is sufficient to set up the initial support portal, since it does not come with significant limitations that would interfere with the objectives of the support portal. Moreover, project partners already have extensive experience with GitLab.

GitLab provides features (and accompanying extensive documentation) to allow for migrating the support portal to another setup, should the need arise. This could be either a payed hosted solution via GitLab Enterprise Edition, or a self-managed instance using the open-source offering that can be hosted on-premise by one of the project partners.

In addition, it is the only candidate solution that was considered which meets *all* outlined objectives (cfr. Section 3.1), including ones that are considered particularly important, like being able to control the visibility of support requests, and ensuring a low threshold for creating support requests by not requiring that an account needs to be created.

While this objective comparison put the spotlight on GitLab as an excellent candidate for setting up a support portal for the shared software stack, we wanted to make sure it also met the expectations in practice. Therefore, a test setup was created to evaluate GitLab in more detail before making a final decision, which is discussed in Section 5.

5 Preliminary testing of the support portal created with GitLab

To make sure that a support portal created with GitLab aligns with the objectives outlined in Section 3.1, a temporary test setup was created using the free hosted offering that is provided by GitLab. This was used to explore the capabilities and limitations of such a setup, and to get detailed feedback from project partners who will be involved in Task 5.4 on the proposed setup of the support portal.

The explicit intention of this testing phase was to weed out potential problems or concerns early on, and to gain more extensive experience with the setup and maintenance of a GitLab instance, before setting up the production-ready version of the support portal.

5.1 Test setup and exploration of GitLab features

The test version of the support portal was set up early June 2023 to explore the features supported by GitLab, and to evaluate how we could best employ them to meet the requirements outlined in Section 3.2.

This consisted of a **private GitLab project** that was only accessible to relevant project partners. The code repository in this project was populated with a README file containing **basic instructions** for hypothetical end users on how to use the support portal, alongside **templates** for creating different types of support requests (asking a question, reporting a problem, submitting a software installation request), as well as for responding to support requests.

A couple of **hypothetical support requests** were created in the issue board, and a set of **labels** was defined to organise support requests with respect to their type, to which part of the shared software stack they relate to, or other aspects. The **confidentiality setting** was enabled for a few support requests, which effectively made them "private" issues that can only be accessed by support team members or the author of the support request.

The built-in wiki provided by GitLab was configured to only be accessible to support team members, so it can be used for **internal documentation**. It was populated to provide information on how to configure your GitLab account to get e-mail notifications, and how to leverage the provided templates for responding to support requests or create new templates. Support team members were granted a "maintainer" or "developer" **role**, which gives them permissions to either manage the configuration of the support portal, or to access the internal documentation and non-public support requests.

The Service Desk feature of GitLab was employed to create a **contact e-mail address** `support@eessi.io` that can be used by end users to create support requests or respond to existing ones, without using a GitLab account. Support requests created this way automatically have the confidentiality setting enabled by default.

5.2 Evaluation using hypothetical support requests

Mid June 2023 project partners were invited to thoroughly evaluate the test setup of the support portal.

Two testing rounds were performed involving additional hypothetical support requests, each of which from a different point-of-view: end users and support team members.

5.2.1 First evaluation round: End User Role

In a first evaluation round, project partners involved in Task 5.4 focused on the end user experience of the support portal.

Six different test cases were provided, each to one partner, consisting of three types of support requests (question, problem report, software installation request) that were created either directly in the support portal, or via the contact e-mail address `support@eessi.io`.

These test cases were enacted as if they would be submitted by an end user of the shared software stack, to assess the end user experience.

5.2.2 Second evaluation round: Support Team Member Role

In a second evaluation round, project partners assumed the role of support team members.

Five test cases were provided, which consisted of processing a variety of support requests. The project partners involved in this round were asked to use the available internal documentation of the support portal, and use the appropriate features of the support portal, without further specific instructions.

The objective of this evaluation round was to make sure that support team members were comfortable with using the support portal, and to check whether the internal documentation provided was sufficient.

5.2.3 Feedback from project partners

Both evaluation rounds of the test setup of the support portal were generally positive, with project partners being able to work through the provided test cases relatively easily. The available internal documentation proved sufficient to configure their GitLab account as needed.

Some detailed points of feedback were provided by project partners, including:

- The structure of the internal documentation in the wiki of the support portal could be improved.
- The available templates for creating a support request were deemed to be potentially overwhelming, since they were asking for a lot of information that a typical end user may not be able to provide on the spot.
- The set of labels for tagging support requests were found to be somewhat unclear and disorganised, and their intended use was not always clear.

This feedback was taken into account when setting up the production version of the support portal, for example by adding a clear description to the labels, using a consistent color scheme for them, and adding documentation on managing them to the internal wiki.

Overall, the project partners involved in Task 5.4 unanimously agreed that the test setup of the support portal with GitLab met the set of requirements and provided the optional features that were put forward, and hence was in line with the objectives that were outlined (see Section 3).

5.2.4 Test migration to on-premise GitLab installation

Next to evaluating the use of the support portal, we also tested the migration of the test setup using the free hosted solution offered by GitLab to a temporary on-premise installation of the open-source version of GitLab.

GitLab provides extensive documentation on this (see Migrating projects using file exports), and also here the experience was overly positive since no significant problems arose during the process.

This further strengthens our confidence that GitLab is the best suited solution for setting up a support portal for the shared software stack, since it validates that requirement REQ-6 (see Section 4.3.5).

6 Production setup of support portal

Following the evaluation of test setup using GitLab (see Section 5), the production version of the support portal for the shared software stack was set up end of September 2023 using the free hosted offering of GitLab:

https://gitlab.com/eessi/support

6.1 Creating and following up on support requests

Creating a support request for the shared software stack can be done either directly in the support portal using a free GitLab account, or by sending an email to `support@eessi.io` (even without having a GitLab account). Likewise, following up on support requests can be done in the support portal itself, or via email.

Support requests that have not been marked as confidential can be consulted publicly via the issue board in the support portal; ones that have been marked confidential are only accessible by support team members.

Figure 1 shows a selection of ongoing and resolved support requests in the support portal along with their associated labels. These were opened by support team members, by partners of the MultiXscale project partners who are not directly involved in WP5 (see issue #13), or by people external to the MultiXscale project (see issue #14).

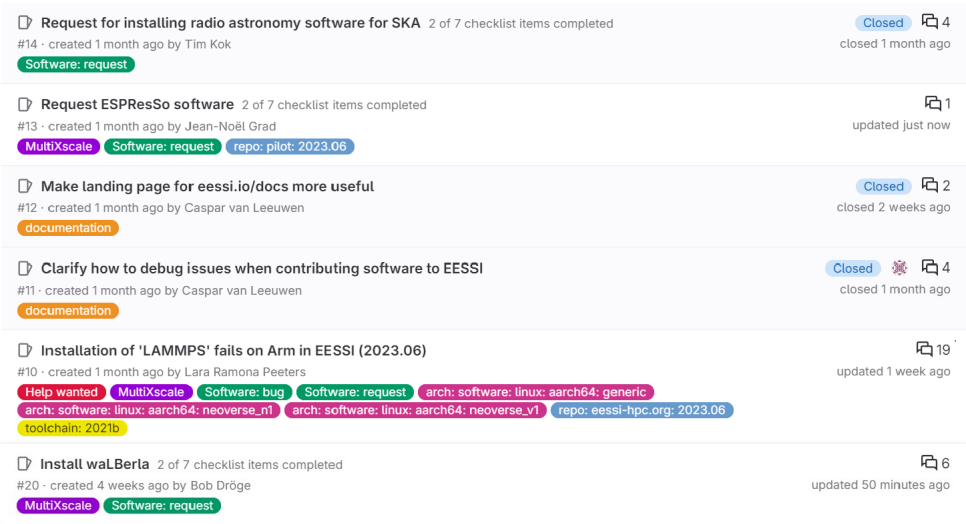

Figure 1: Overview of issues in the support portal on GitLab, corresponding to support requests.

6.2 Documentation

User-facing documentation on how to create a support request was made available via a dedicated page in the EESSI documentation, which also covers the level of support (see also Section 6.4). Basic instructions on how to create a particular type of support request are also provided in the `README` file of the repository in the support portal.

Internal documentation for support team members, that provides information on efficiently handling support requests, how to configure your GitLab profile, and includes the planning of the support rotation (see Section 6.5) is integrated in the support portal through the wiki that is provided by GitLab; see also Figure 2.

6.3 Internal communication for support team

To ensure smooth communication among support team members, a couple of different channels are used:

- A Slack channel for (semi-synchronous) informal day-to-day discussions among support team members;
- Weekly 1-hour online meetings to discuss and align the ongoing tasks and support requests, split in two types:
  - Handover meetings between the two partners who are handing and taking over in the support rotation, along with other active support team members, every other week.
  - Sync meetings on new and ongoing support requests, for all active support team members, led by the partner who is then principally responsible for the support portal, every other (non-handover) week;

Meeting notes for the weekly handover or sync meetings are saved in a dedicated section of the internal documentation in the support portal.

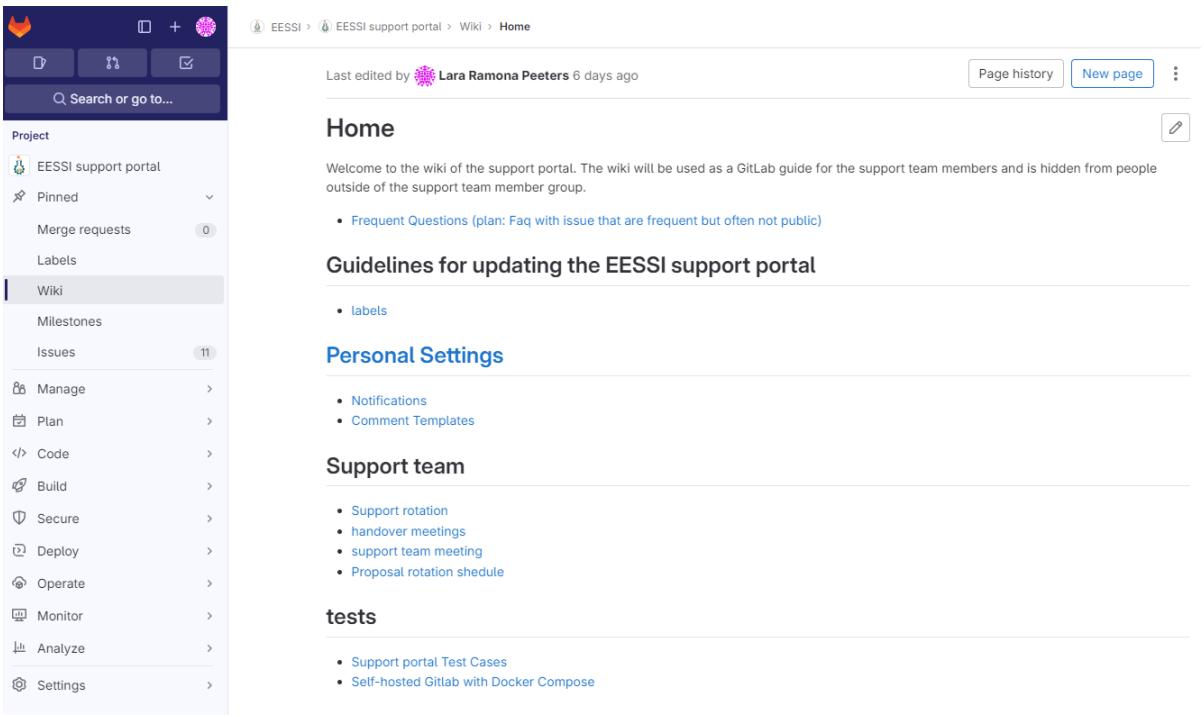

Figure 2: Screenshot of the internal documentation for the support portal, via the wiki that is provided by GitLab.

6.4 Level of support

The support level that is being offered for the shared software stack is defined in the user-facing documentation.

It clearly states that:

- Support is offered according to a "reasonable effort" standard, similar to what is typically offered by open-source software projects;
- The support team focuses primarily on resolving problems and answering questions related to accessing and using the shared software stack, requesting software installations, and providing support for the tools and services that are closely related to the shared software stack (like the EESSI test suite, or the build-and-deploy bot);
- Software-specific questions or problems that are not related to how the software was installed, should be directed to alternate appropriate support channels (such as, for example, the forums or mailing lists associated with a particular software package).

6.5 Support rotation

From October to December 2023, the project partners involved in Task 5.4 conducted a test for the planned support rotation using the production version of the support portal.

Every project partner who will be actively involved in the support effort for the shared software stack was made principally responsible for managing incoming or ongoing support requests for 2 consecutive weeks. As such, every project partner who will be involved in Task 5.4 had taken the principal support role once by the end of Task 5.1.

Every other week, another project partner took over the principal support role during the scheduled handover meetings, which served as a way to exchange information on the current status of open support tasks. The support team sync meetings that were held during the other weeks served as a way to synchronize on recently created and open support tasks with active support team members. Detailed meeting notes were kept and shared with all support team members via the internal documentation in the support portal.

This approach proved to work very well to ensure that support was provided smoothly and effectively, and hence the support rotation was extended into 2024 following the same principle, in preparation for Task 5.4.

About 25 issues were opened in the support portal during this period, to track support tasks that should be handled by one of the project partners, for example the installation of key applications in the MultiXscale project like ESPResSO, waLBerla, and LAMMPS.

7 Conclusions

In this report, we have presented the process of selecting and comparing candidate solutions for setting up a support portal for the shared software stack, along with corresponding documentation. This was done via a well-defined set of objectives and requirements.

By thoroughly comparing an initial selection of candidate solutions, GitLab was identified as the best suited option. An additional evaluation using a test setup of the support portal confirmed that GitLab indeed provides the necessary features to meet the outlined objectives and requirements.

Subsequently, the production version of the support portal was set up using the free hosted offering provided by GitLab at https://gitlab.com/eessi/support, and accompanying user-facing guidelines were made available in a dedicated page of the EESSI documentation.

A concrete proposal for a support rotation was tested with all project partners who will be involved in providing support for the shared software stack. This was found to work well, and serves a good basis in preparation for the effort planned in Task 5.4.

During the first months of operation, about 25 support requests and tasks were tracked via the support portal, which relate to both efforts within the scope of the MultiXscale project, as well as beyond it, including by people external to the project.

References

Acronyms Used

HPC High Performance Computing
WP Work Package
EESSI European Environment for Scientific Software Installations
UGent Ghent University

URLs referenced

Page ii

https://www.multixscale.eu ... https://www.multixscale.eu
https://www.multixscale.eu/deliverables ... https://www.multixscale.eu/deliverables
Internal Project Management Link ... https://github.com/multixscale/planning/issues/28
kenneth.hoste@ugent.be ... mailto:kenneth.hoste@ugent.be
http://creativecommons.org/licenses/by/4.0 ... http://creativecommons.org/licenses/by/4.0

Page 1

https://gitlab.com/eessi/support ... https://gitlab.com/eessi/support
Square Kilometre Array Observatory (SKAO) ... https://www.skao.int/en

Page 4

support portal for Slurm ... https://bugs.schedmd.com

Page 7

large variety of potential solutions ... https://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems
Bugzilla ... https://www.bugzilla.org/
GitHub ... https://github.com/
GitLab ... https://gitlab.com/
Jira ... https://www.atlassian.com/software/jira
OTRS ... https://otrs.com/
Redmine ... https://www.redmine.org/
Request Tracker ... https://bestpractical.com/request-tracker
Trac ... https://trac.edgewall.org/
Bugzilla ... https://www.bugzilla.org
user groups ... https://bugzilla.readthedocs.io/en/latest/administering/parameters.html#group-security
GCC ... https://gcc.gnu.org/bugzilla
Gentoo Linux ... https://bugs.gentoo.org
Slurm ... https://bugs.schedmd.com
GitHub ... https://github.com
GitHub Enterprise Server ... https://docs.github.com/en/enterprise-server@3.11/admin/overview/about-github-enterprise-server
GitLab ... https://gitlab.com
GitLab Enterprise Edition ... https://docs.gitlab.com/ee/administration/license.html
self-managed ... https://docs.gitlab.com/ee/install/install_methods.html
confidential ... https://docs.gitlab.com/ee/user/project/issues/confidential_issues.html
Service Desk ... https://docs.gitlab.com/ee/user/project/service_desk/index.html

Page 8

Jira ... https://www.atlassian.com/software/jira
OTRS ... https://otrs.com/otrs-software-solutions/otrs
Znuny ... https://www.znuny.org/en
Redmine ... https://www.redmine.org
Request Tracker (RT) ... https://bestpractical.com/request-tracker
Trac ... https://trac.edgewall.org

Page 10

on migrating between hosted and self-managed GitLab instances ... https://docs.gitlab.com/ee/user/project/import/
importing/exporting projects via their API ... https://docs.gitlab.com/ee/api/project_import_export.html
GitLab Enterprise Edition ... https://docs.gitlab.com/ee/administration/license.html

Page 11

test version of the support portal ... https://gitlab.com/eessi.io/support-test
templates ... https://docs.gitlab.com/ee/user/profile/comment_templates.html
labels ... https://docs.gitlab.com/ee/user/project/labels.html
confidentiality setting ... https://docs.gitlab.com/ee/user/project/issues/confidential_issues.html
built-in wiki provided by GitLab ... https://docs.gitlab.com/ee/user/project/wiki
role ... https://docs.gitlab.com/ee/user/permissions.html#project-members-permissions
Service Desk feature of GitLab ... https://docs.gitlab.com/ee/user/project/service_desk

Page 12

Migrating projects using file exports ... https://docs.gitlab.com/ee/user/project/settings/import_export.html

Page 13

https://gitlab.com/eessi/support ... https://gitlab.com/eessi/support
issue board in the support portal ... https://gitlab.com/eessi/support/-/issues
a dedicated page in the EESSI documentation ... https://www.eessi.io/docs/support

Page 14

user-facing documentation ... https://www.eessi.io/docs/support

Page 15

https://gitlab.com/eessi/support ... https://gitlab.com/eessi/support