

D2.3 SLICES-SC final portal for accessing the RI

Acronym	SLICES-SC
Project Title	Scientific Large-scale Infrastructure for Computing/Communication Experimental Studies – Starting Community
Grand Agreement	101008468
Project Duration	42 Months (01/03/2021 – 31/08/2024)
Due Date	30 June 2024 (M40)
Submission Date	26 June 2024 (M40)
Authors	Nikos Makris (UTH), Brecht Vermeulen (imec), Thijs Walcarius (imec), Wim Van de Meerssche (imec), Bartosz Belter (PSNC), Sebastian Gallenmueller (TUM), Cedric Crettaz (MI), Esa Posio (UOULU), Konstantinos Filis (OTE)
Reviewers	Serge Fdida (SU), Walid Dabbous (INRIA)



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101008468. The information, documentation and figures available in this deliverable, is written by the SLICES-SC project consortium and does not necessarily reflect the views of the European Commission. The European Commission is not responsible for any use that may be made of the information contained herein.





Executive summary

This document outlines the design, implementation, and functionalities of the final SLICES-SC portal, which serves as the primary entry point for accessing the project's research infrastructure. The portal has been developed to align with the SLICES-SC branding and ensure a seamless user experience. It incorporates essential features such as user-friendly interfaces, integration with external authentication providers, detailed user and project management, comprehensive logging and auditing, and advanced statistics and analytics. Additionally, the portal supports OAuth API integration, allowing other web-based services to utilize the same user account base, thereby enhancing interoperability and ease of use. Key sections of this document include: 1) Portal Requirements: this section details the necessary requirements for the portal, including branding, user interface design, terms and conditions approval, and robust user and project management functionalities. 2) Portal Walkthrough: this section provides a detailed walkthrough of the portal, showcasing its features, supported facilities, and dashboard functionalities. 3) Authentication and Authorization: this section describes the methods used for user authentication and authorization within the portal. 4) API Integration: this section outlines the specific APIs developed for the portal, including methods to achieve experiment reproducibility. 5) AI Tools for Experimenters: this section discusses the integration of AI tools to assist experimenters, including a detailed evaluation of their performance.



Table of content

- EXECUTIVE SUMMARY 2**
- TABLE OF CONTENT 3**
- ACRONYMS..... 5**
- LIST OF FIGURES..... 6**
- 1. INTRODUCTION..... 7**
- 2. PORTAL REQUIREMENTS 7**
 - 2.1. BRANDING AND LOOK & FEEL 7
 - 2.2. USER-FRIENDLY PORTAL..... 7
 - 2.3. TERMS AND CONDITIONS FLOW 7
 - 2.4. INTEGRATION WITH EXTERNAL AUTHENTICATION PROVIDERS (E.G. EDUGAIN) 8
 - 2.5. USER INFORMATION MANAGEMENT..... 8
 - 2.6. USER AND PROJECT MANAGEMENT..... 8
 - 2.7. AUDITING AND LOGGING 8
 - 2.8. STATISTICS AND ANALYTICS 8
 - 2.9. OAUTH API INTEGRATION..... 8
- 3. PORTAL WALKTHROUGH 9**
 - 3.1. DISCOVER OUR TESTBEDS..... 9
 - 3.2. SLICES-SC SUPPORTED FACILITIES 11
 - 3.2.1. *OneLab CloudLab* 11
 - 3.2.2. *NITOS testbed*..... 12
 - 3.2.3. *5GTN testbed*..... 12
 - 3.2.4. *IoT Lab testbed* 12
 - 3.2.5. *R2Lab testbed*..... 12
 - 3.2.6. *imec iLab.t* 13
 - 3.3. DASHBOARD PER PROJECT FUNCTIONALITY 13
- 4. PORTAL AUTHENTICATION & AUTHORIZATION METHOD 14**
- 5. APIS SPECIFIC TO THE PORTAL..... 15**
 - 5.1. PORTAL APIS 15
 - 5.2. SLICES/POS REPRODUCIBILITY FRAMEWORK AND INTEGRATION WITH THE PORTAL..... 15
- 6. AI TOOLS FOR EXPERIMENTERS..... 17**
 - 6.1. LLM BASED CHATBOT - SLICES..... 17
 - 6.1.1. *LLM Architecture and Configuration* 17
 - 6.1.2. *Data acquisition and augmentation* 18
 - 6.1.3. *In context learning and fine tuning*..... 20
 - 6.1.4. *Evaluation and indicative results* 21





- 6.1.5. *Future Work and exploitation plans*.....23
- 6.2. EOSC BASED AI TOOL FOR SUGGESTING RESOURCES BASED ON THE BEHAVIOUR OF USERS24
 - 6.2.1. *System Architecture*24
 - 6.2.2. *Workflow of the recommendation generation process*26
 - 6.2.3. *Integration with research infrastructures*28
 - 6.2.4. *User interface and experience*.....29
 - 6.2.5. *Evaluation and results*.....30
 - 6.2.6. *Partnerships and potential future roadmap*31
- 7. CONCLUSION 33**





Acronyms

AI	Artificial Intelligence
AM	Aggregate Manager
ANN	Approximate Nearest Neighbours
API	Application Programming Interface
DI	Digital Infrastructures
EOSC	European Open Science Cloud
Fed4FIRE	FEDeration for Future Internet Research and Experimentation
FT	Fine Tuning
GCF	GENI Control Framework
GDPR	General Data Protection Regulation
GENI	Global Environment for Network Innovations
GPT	Generating Pretrained Transformer
HPC	High Performance Computing
HTTP	HyperText Transfer Protocol
ICL	In-context Learning
LLM	Large Language Models
MA	Member Authority
OAuth	Open Authorization
PEM	Privacy Enhanced Mail
RAG	Retrieval Augmented Generation
RBAC	Role-Based Access Control
RI	Research Infrastructures
RSpec	Resource Specification
SA	Slice Authority
SSO	Single Sign-On
URN	Uniform Resource Name



List of figures

Figure 1: Portal Front page (<https://portal.slices-sc.eu>)9

Figure 2: Testbed directory page10

Figure 3: Testbed directory: overview of filters and information about a single infrastructure10

Figure 4: imec testbed portfolio available through SLICES-SC13

Figure 5: Per project functionality in the dashboard.....14

Figure 6: Project extension request.....14

Figure 7: SLICES-SC documentation structure at <https://doc.slices-sc.eu>15

Figure 8 Porting the pos experiment workflow (in the hosted testbed) to other testbeds (the hosting testbed).....16

Figure 9: sIAIces LLM system architecture18

Figure 10: Percentage of Correctly Answered Questions.....21

Figure 11: Indicative responses from sIAIces on the 5G blueprint.....23

Figure 12: Indicative responses from sIAIces, when trained with the OpenAirInterface documentation23

Figure 13: Architecture of the recommendation system25

Figure 14: Simplified workflow of the recommendation system26

Figure 15: User interface of the recommendation system29





1. Introduction

The SLICES-SC project aims to establish a comprehensive and user-friendly portal that serves as the primary entry point for accessing the offered research infrastructure. The SLICES-SC portal is designed to facilitate large-scale experimental studies in computing and networking communication. This document outlines the requirements and features necessary to ensure the portal not only aligns with the SLICES-SC requirements, but also meets the functional needs of its diverse user base, including researchers, academics, and students.

The document is structured as follows. In Section 2, we present the requirements for the portal operation, as concluded by the consortium during its design phase. In Section 3, we present a walkthrough of the delivered portal, along with the supported facilities, and the manner in which they integrated to the portal. Section 4 presents the authentication mechanisms, while Section 5 refers to the specific portal APIs, and the methods introduced to achieve experiment reproducibility. In Section 6 we present how AI tools have been integrated to the portal, in order to assist experimenters, along with a brief evaluation of their operation. Finally, in Section 0 we conclude the document.

2. Portal requirements

As the initial entry point for external users accessing the infrastructure, the portal fulfills several critical requirements identified and analyzed by the consortium during the development stage. These requirements ensure that the portal is not only visually consistent with the existing SLICES-SC branding but the portal is also functionally robust and easy to use. Below are the detailed requirements and features necessary for the portal:

2.1. Branding and Look & Feel

- **Consistency with SLICES-SC Branding:** The portal should visually align with the SLICES-SC website (<https://slices-sc.eu>). This includes using the same color palette, fonts, and overall design aesthetics to create a cohesive brand experience.
- **User Interface Design:** The interface should be modern, clean, and intuitive. Elements such as buttons, menus, and forms should follow the same styling guidelines as the SLICES-SC website to provide a seamless user experience.
- **Navigation Structure:** The portal should have a similar navigation experience to the SLICES-SC website, including a consistent header, footer, and sidebar menus. This familiarity helps users easily find information and navigate through the portal.

2.2. User-Friendly Portal

- **Invitation System:** Implement a straightforward system for inviting new users to projects or student classes. This system should support bulk invitations via email and provide clear notifications and tracking for both invitees and project administrators.
- **Intuitive Dashboard:** Develop a user-friendly dashboard that presents an overview of the user's projects, pending invitations, and recent activities. The dashboard should be customizable to cater to individual user preferences.

2.3. Terms and Conditions Flow

- **Approval Process:** Create a clear and straightforward process for users to review and accept terms and conditions, including GDPR compliance terms. This process should be integrated into the account registration and first login procedures.



- **Accessibility:** Ensure that the terms and conditions are easily accessible at all times, allowing users to review them whenever necessary.

2.4. Integration with external authentication providers (e.g. eduGAIN)

- **Single Sign-On (SSO):** Integrate eduGAIN for SSO, enabling users to log in with their university credentials. This simplifies access for academic users and enhances security.
- **Account Linking:** Allow users to link their eduGAIN account with their portal account, ensuring seamless transitions and consistent access across platforms.

2.5. User Information Management

- **Detailed Account Information:** During the registration process, collect detailed information necessary for creating user credentials. This information will facilitate fine-grained access control on testbeds.
- **Profile Management:** Provide users with tools to update their personal information, manage their credentials, and view their account activity within a dedicated profile management section.

2.6. User and Project Management

- **Project Creation and Collaboration:** Enable users to create and manage projects, inviting multiple experimenters to collaborate on the same project. Each project should have configurable settings and permissions.
- **Role-Based Access Control (RBAC):** Implement RBAC to define different access levels and permissions within projects. This ensures that only authorized users can perform specific actions.

2.7. Auditing and Logging

- **Comprehensive Logging:** Maintain detailed logs of user activities, including login attempts, project activities, and changes to account settings. This supports security and accountability.
- **Audit Reporting:** Generate audit reports that administrators can review to monitor compliance and security within the portal.

2.8. Statistics and Analytics

- **Usage Metrics:** Provide detailed statistics on the number of user accounts, active projects, and other relevant metrics. This data helps in understanding portal usage and planning improvements.
- **Activity Reports:** Generate reports that detail user and project activities, providing insights into how the portal is being utilized.

2.9. OAuth API Integration

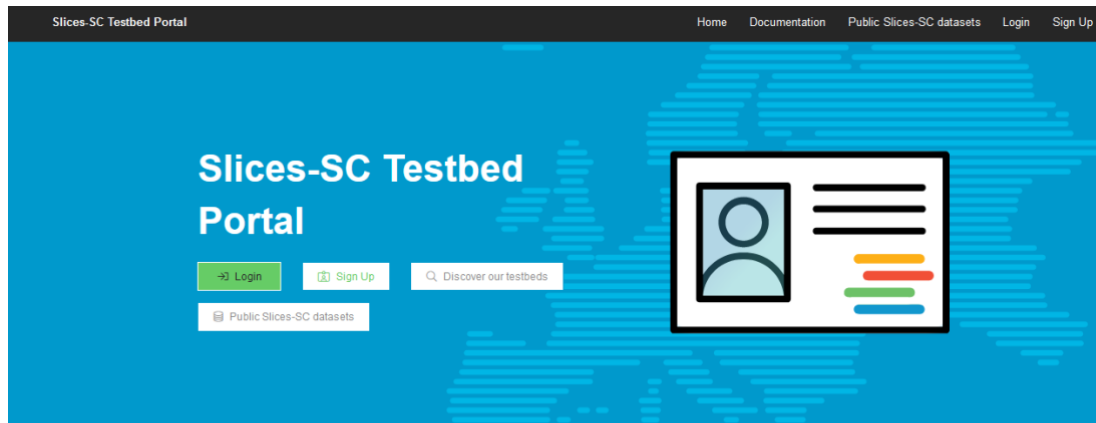
- **API Development:** Develop an OAuth API that allows other web-based services to integrate with the portal. This facilitates the use of the same user account base across multiple services.



- **Comprehensive Documentation:** Provide clear and detailed documentation for the OAuth API, ensuring that developers can easily understand and implement it.

3. Portal walkthrough

D2.2 included a portal walkthrough of the initial portal. In this section we will highlight the changes and updates since then.



What is this Portal?

This Slices-SC Testbed Portal provides accounts for accessing *testbed resources*. The testbeds can be used to execute *experiments*.

Figure 1: Portal Front page (<https://portal.slices-sc.eu>)

Two buttons have been added to the front page:

- Discover our testbeds: see next section
- Public SLICES-SC datasets: this is a link to <https://slices-sc.eu/slices-datasets/> to make sure all experimenters are aware of the public datasets

3.1. Discover our testbeds

This is a new section on the portal, that can be found at <https://portal.slices-sc.eu/explore/directory/> and offers an overview of all the relevant infrastructure in SLICES-SC, see the figures below. On the left side, multiple filters are present to search for the right testbed, while on the right you have a short overview of each infrastructure, with links to documentation.



The screenshot shows the 'Explore our offering' page of the Slices SC Testbed Portal. It features a navigation bar at the top with 'Home', 'Documentation', 'Public Slices-SC content', 'Login', and 'Sign Up'. A left sidebar contains a 'Category' list (Wired, Wireless, 4G, 5G, 6G, Edge, IoT, Big Data, OpenFlow, Cloud, SDN, AI, Cognitive Radio, HPC, GPU, Multimedia, Optical) and a 'Filters' section for 'Technologies' (GPU, Wireless Networking, OpenFlow, Infiniband, Cloud Software Stack, OpenStack Cloud, Ansible Automation, OptiTrac, 3D/2D, DWDM, FTTH, P4, O-RAN, 100GbE) and 'Properties' (IPv6 Support, Job Based, Large Storage, Resource reservation, City environment, International L2 connectivity, Sensor Observations as a Service, Long Running experiments, Bare-metal access, 4G connectivity, 5G NSA and SA connectivity, Different Open Source 4G and 5G cores, IoT network with hundreds of sensors, 3D scanner for 3D modeling). The main area displays a grid of testbed cards, each with a title, location, categories, size, technologies, properties, and a description. Examples include GPULab (Ghent University, Belgium), W-Lab 1 (imec - Ghent University, Belgium), CityLab (University of Antwerp, Belgium), OneLab (Sorbonne University, France), and OneLab Fit CloudLab (Sorbonne University, France).

Figure 2: Testbed directory page

This screenshot provides a detailed view of the filter options and the information for a specific testbed, GPULab. The 'Category' list on the left is expanded to show 'Show All (18)'. The 'Filters' section on the left shows 'Technologies' with 'Wireless Networking' selected, and 'Properties' with 'Large Storage', 'Resource reservation', and 'City environment' selected. The main content area displays the details for the 'GPULab' testbed at imec - Ghent University, Belgium. It lists categories as 'Big Data, HPC, AI, GPU', size as '25-100 nodes', and technologies as 'nVidia GPUs'. Properties include 'IPv6 Support', 'Job Based', 'Large Storage', 'Resource reservation', 'City environment', 'International L2 connectivity', 'Sensor Observations as a Service', 'Long Running experiments', 'Bare-metal access', '4G connectivity', '5G NSA and SA connectivity', 'Different Open Source 4G and 5G cores', 'IoT network with hundreds of sensors', '3D scanner for 3D modeling', '8K Visualization/streaming/recording', 'ambisonic sound', 'OLT/ONT/ODM', 'optical network generator', and 'IoT commercial and custom sensors'. The description states: 'GPULab is a distributed system for running jobs in GPU-enabled Docker-containers. GPULab consists of a set of heterogeneous clusters, each with their own characteristics (GPU model, CPU speed, memory, bus speed, ...) allowing you to select the most appropriate hardware. GPUs vary from Nvidia 1080Ti up till the Nvidia V100 and DGX-2 system. Each job runs isolated within a Docker containers with dedicated CPUs, GPU's and memory for maximum performance.' The card also indicates 'Is Federated: Yes' and provides buttons for 'Learn more', 'Ask a question', and 'Technical documentation'.

Figure 3: Testbed directory: overview of filters and information about a single infrastructure



3.2. SLICES-SC Supported facilities

The participating facilities in the project offer several technologies to external experimenters. Nevertheless, due to the complexity of each facility, and the fact that many of them have different APIs and offer different functionalities, the unification of all tools under the portal was not rendered feasible. To this end, two options were offered to the facilities: 1) to fully integrate with the portal, adopting a respective Aggregate Manager (AM) that allows the translation of basic commands of an experiment to the specific API of each testbed, and 2) a loose integration model, that allows accounts registered with the SLICES portal to access each of the facilities. The detailed technologies supported by each facility, as reported and in D2.1 are presented in Table 2. Details on the integration model per each facility that follows the first integration method are reported in the subsections below. The rest of the facilities, follow the single sign-in mechanism provided by the project.

Table 1: SLICES-SC sites and capabilities as reported in D2.1

Testbed	Country Location	WiFi	4G LTE	5G NR	IoT	HPC	Cloud	SDN	Open-Flow	Optical networks
SILECS-FIT/OneLab	SU/FR						√			
NITOS	UTH/GR	√	√	√	√		√	√	√	
Open5GLab	EUR/FR		√	√						
PIONIER-LAB	PSNC/PL						√			√
5GTN	OULU/FI	√	√	√	√					
LeonR&D	OTE/GR				√					
CNR-IIT Lab	CNR/IT				√	√				
imec iLab.t	IMEC/BE	√	√	√	√	√	√	√	√	
ELKH Cloud	SZTAKI/HU						√			
R2Lab	INRIA/FR	√	√	√		√	√	√		
IoT Lab	MI/CH				√					
SLICES Virtual pos	TUM/DE						√			
5TONIC	IMDEA/ES		√	√	√			√		

3.2.1. OneLab CloudLab

FIT Cloudlab Europe <https://portal.onelab.eu/> is a meta-cloud or more precisely it's a facility to build clouds. It provides bare-metal access and control over a substantial set of computing, and storage resources to users. FIT Cloudlab Europe is now fully integrated into SLICES-SC portal using the GENI AM API, thus users can install new, existing or modified cloud software stacks and test them using this facility.



3.2.2. NITOS testbed

NITOS testbed (<https://nitos.inf.uth.gr>) is fully integrated with the SLICES-SC portal. The testbed is using the “Advanced Federation”¹ option for connecting to the portal, supported through the implementation of a new AM, called the NITOS broker^{2,3}. The broker is supporting several interfaces (REST,SFA) that can be used in order to handle the reservation of testbed nodes, deploying images on the reserved nodes, experimenting with them, and releasing after the experimentation process. Low-level configuration for wireless radios (WiFi, 4G, 5G) is exposed through the API.

3.2.3. 5GTN testbed

University of Oulu’s 5G Test Network (5GTN), the world’s first open access 5G Test Network, which is currently developed towards 6G offers a complete test environment, from infrastructure to apps and services, allows unique testing possibilities from prototypes to solutions in a controlled environment. 5GTN testbed (<https://5gtn.fi>) is currently connected with the SLICES-SC portal, and it is opted for “Advance Federation”. 5GTN uses Emulab as testbed management software and on top of Emulab it is using Protogeni AMAPI to integrate testbed with the portal. Both are instrumental in facilitating this versatile federation and support for modifying, creating, and monitoring the resources. Work is ongoing on the resource addition to the testbed. 5GTN has two dedicated servers for initial node setup. As 5GTN is committed to support cutting-edge research in wireless technology and enhance our testbed capabilities, we are working on integrating 5G Pico base stations equipped with Open5GS core. 5G Pico base stations use frequency band n78. 6G PoC devices will be integrated once available.

3.2.4. IoT Lab testbed

The IoT Lab testbed is available on the SLICES-SC portal. Initially, the IoT Lab testbed was integrated into the Fed4FIRE testbed federation using the advanced federation model. An Aggregate Manager (AM) was created based on the GENI Control Framework (GCF) and the communication between the IoT Lab testbed and the SLICES-SC portal is realized through the GENI AM API version 3. The process of federating the IoT Lab testbed corresponds to the requirements and the official technical documentation provided by the SLICES starting community. The technical documentation related to the federation of the IoT Lab testbed into respectively Fed4FIRE and SLICES is available online at <https://gitlab.distantaccess.com/iotlab-fed4fire/fed4fire-documentation>.

3.2.5. R2Lab testbed

R2lab is an open tested located in an anechoic chamber for reproducible research in wireless WiFi and 4G/5G networks. R2lab is part of the FIT federation, an open large-scale testing infrastructure for systems and applications on wireless and sensor communications. Located at Inria Centre at Université Côte d'Azur, R2lab proposes thirty-seven customisable commercial off-the-shelf wireless devices,

¹ Advanced Federation with the SLICES-portal: <https://doc.slices-sc.eu/#advanced-federation> [Last accessed 25 June 2024]

² Creating your Aggregate Manager (AM): software options, https://doc.slices-sc.eu/testbed_owner/amsoft.html [Last accessed 25 June 2024]

³ D. Stavropoulos, A. Dadoukis, T. Rakotoarivelo, M. Ott, T. Korakis and L. Tassiulas, "Design, architecture and implementation of a resource discovery, reservation and provisioning framework for testbeds," 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Mumbai, India, 2015, pp. 48-53, doi: 10.1109/WIOPT.2015.7151032.



together with USRP nodes, 5G remote radio units, commercial LTE phones, and programmable 5G User Equipment to create rich experimental setups. The testbed also features advanced software like leverage GnuRadio and OpenAirinterface, as well as efficient software tools, to support easy experimentation. These tools allow to book the whole testbed, to remotely control the wireless devices, to easily deploy various scenarios and to collect results.

3.2.6. imec iLab.t

The imec iLab.t facility consists out of multiple infrastructures (<https://doc.ilabt.imec.be>).

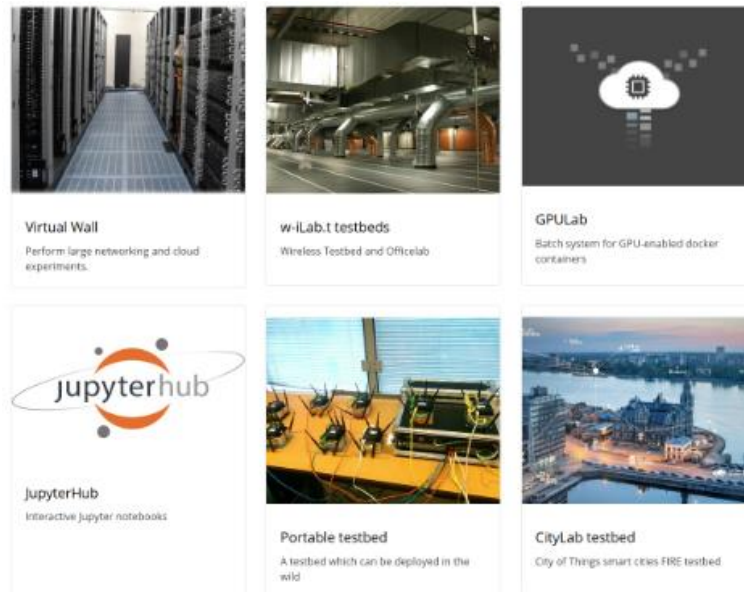


Figure 4: imec testbed portfolio available through SLICES-SC

- Virtual wall (Gent): to perform wired networking, cloud, distributed software, service backends and scalability experiments. 550+ installed servers.
- w-iLab.t (Gent): pseudo shielded environment for wireless and IoT research with over 150 wireless nodes (fixed and mobile), including software defined radios.
- Officelab (Gent): a real office environment for wireless and IoT research with over 110 embedded PCs spread over the building.
- GPULab (Gent and Antwerpen): testbed with 125+ GPUs with over 570.000+ cuda cores and 1.8TB+ GPU RAM for AI research and everything which needs GPUs. Available through interactive jupyter notebooks and scheduled jobs.
- CityLab (Antwerpen): testbed for wireless networking experimentation in the unlicensed spectrum in the city of Antwerp. 50 nodes are spread over an area of 1 square km.

All testbeds are available through the SLICES portal and tools. The testbeds are open to externals. All testbeds are perfectly accessible through remote access, which is also the main usage of the testbeds.

3.3. Dashboard per project functionality

In the specific page per project, 2 things were added:



- As we got more questions to extend project durations (see Expiration field), we added a request extension button so there is a form asking for motivation and new expiration data, see figures below.
- Demanded by the SLICES-PP project, a field was added with the specific project S3 storage bucket name that is used in storage infrastructure set up for SLICES-PP pre-operations.

As you can see, the upper menu bar also contains a button to open the jFed tool, so it's easy to access from everywhere.

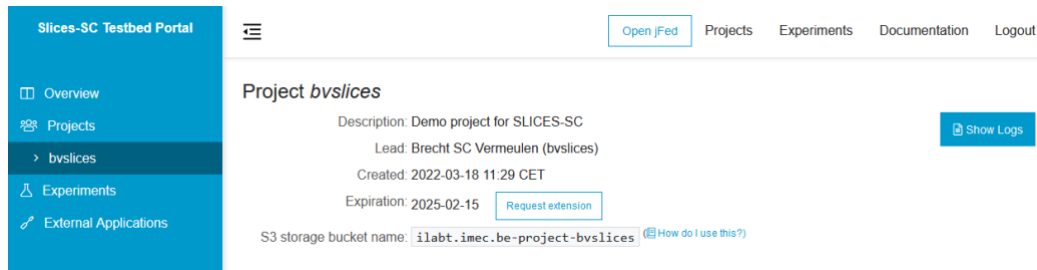


Figure 5: Per project functionality in the dashboard

Request extension for project *bvslices*

Please motivate the extension request for your project:

Motivation

Project Description

Demo project for SLICES-SC

Please review your project description to make sure that it still reflects the actual work being done.

New Expiration Date

15 / 02 / 2025

Request Project Extension

Figure 6: Project extension request

4. Portal Authentication & Authorization method

The authentication methods have not changed since D2.2. We have however, improved the documentation for infrastructure developers/owners at https://doc.slices-sc.eu/testbed_owner/oauth.html.

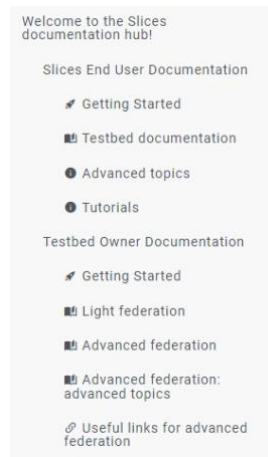


Figure 7: SLICES-SC documentation structure at <https://doc.slices-sc.eu>

5. APIs specific to the portal

5.1. Portal APIs

The APIs have not changed since D2.2.

5.2. SLICES/pos reproducibility framework and integration with the portal

The SLICES/pos framework was originally designed to orchestrate testbeds and execute experiments. Currently, within SLICES-SC, pos is being developed and managing the TUM testbed infrastructure. Towards integrating the framework with the portal, the TUM testbed has been used as the testbed upon which the contributions are developed. Based on the pos architecture, the integration of the authentication service into the management host of the pos testbed was deemed necessary. In the past, we used a local authentication service at TUM to create users and manage the authentication for the testbed. However, this approach does not scale for SLICES, which needs to handle accounts distributed over multiple testbeds and institutions.

To allow authentication in such an environment, we implemented the OpenID connect standard. This is a standardized single-sign-on service (SSO) that is widely used in the Internet. The OpenID connect service and its implementation for pos testbeds are described in more detail in one of our publications⁴. OpenID connect can provide different properties that can be used to offer additional services for the testbed, such as an SSH key that can be used to provide an SSH login to the testbed for users. OpenID can also provide groups for its users. The group membership can be used to grant access to the testbed for a specific user group. We use this group feature of the SLICES portal to manage the access to one of our virtualized testbeds to authenticate its users.

Using the pos framework to orchestrate the TUM testbed together with OpenID and the SLICES portal is only one of the supported deployment scenarios. To increase the flexibility of the pos framework, we split the architecture into two components: a *testbed orchestrator* and an *experiment controller*.

⁴ Sebastian Gallenmüller, Eric Hauser, Georg Carle, “Prototyping Prototyping Facilities: Developing and Bootstrapping Testbeds,” in 2022 IFIP Networking WKSHPS: SLICES Scientific Instruments to support digital infrastructure science (IFIP Networking 2022 WKSHPS SLICES), Catania, Italy, Jun. 2022, <https://ieeexplore.ieee.org/document/9829817> [Last accessed 25 June 2024]

The *testbed orchestrator* is the component that manages the testbed. This involves typical tasks such as the reservation of hardware resources for testbed users, or the rebooting of experiment hosts. The *experiment controller* is the component of pos that executes the actual experiments. The experiments are defined by scripts. The experiment controller deploys and executes the scripts on the previously reserved and rebooted experiment nodes. It monitors the execution of the scripts, collects the experiment results, and also manages the evaluation process. It is the responsibility of the experiment controller to ensure the reproducibility of experiments. Reproducibility is achieved through the specific structure of the experimental workflow supported by the pos framework. This workflow provides its own Linux image ensuring that each experiment always starts from a well-defined state. After each experiment, the experiment nodes reboot a specific Linux image. This avoids having an unwanted, residual state between experiments. Experimenters know about this state reset at the beginning of an experiment, ensuring that they include all configuration steps as part of the automated experimental workflow. In one of our publications⁵, we describe how to use the SLICES pos experiment controller to bring the pos experiment workflow and its reproducibility feature to other testbeds. The testbed still is managed by its original testbed orchestrator, e.g., the GENI framework. Inside this testbed, an experiment is created that deploys the pos experiment controller, i.e., a pos-hosted testbed inside the hosting testbed. This allows porting pos experiments that use its experiment API and its reproducibility feature to other testbeds. We demonstrated this for an experiment in two US-based testbeds, CloudLab and Chameleon.

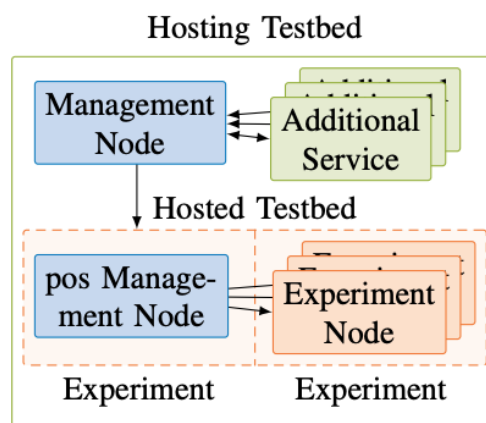


Figure 8 Porting the pos experiment workflow (in the hosted testbed) to other testbeds (the hosting testbed)

We briefly described the two different deployments currently supported by pos: (1) The traditional deployment of the pos testbed orchestrator and its experiment controller and (2) the deployment of the experiment controller inside other experiments (cf. Fig. 1). For the first deployment the SLICES portal (or any other OpenID connect provider) can be used to authenticate its users. The API to execute experimental workflows, this deployment provides the pos experiment controller. The second deployment only deploys the API to execute experimental workflows. The authentication is provided by the operator of the hosting testbed. This may be a testbed that utilizes the SLICES portal or other means of user authentication.

⁵ Henning Stubbe, Sebastian Gallenmüller, Georg Carle, “The pos Experiment Controller: Reproducible & Portable Network Experiments,” in 2024 19th Wireless On-Demand Network Systems and Services Conference (WONS), 2024, pp. 1–8. <https://ieeexplore.ieee.org/document/10449532> [Last accessed 25 June 2024]



6. AI tools for experimenters

Towards lowering the barrier for new-users to start experimenting with SLICES-SC, AI tools have been integrated in order to support experimenters in getting started with the experimentation process, or recommending resources for them based on their experimentation needs. Two tools have been designed, developed and evaluated in order to assist experimenters, as described below.

6.1. LLM based chatbot - sIAIces

Research Infrastructures (RIs) have emerged as key components for validating research outcomes in domain sciences. Such instruments need to keep up with the most recent advances in technology, offering bleeding-edge resources to researchers, and integrating the most advanced features that accelerate the materialization of new ideas to research outcomes. As such, SLICES⁶ is the first RI defined as a scientific instrument in the Digital Infrastructures (DI) domain, able to offer cutting-edge resources for experimentation to researchers of the field, while providing them with structured experiment lifecycle management from the design phase, to the deployment, evaluation, archiving and post-experiment analysis.

The resources offered by SLICES are defined based on thought experiments, spanning a wide range of programmable infrastructure, from Open-RAN and beyond 5G networks to edge and High-Performance Computing (HPC) and IoT. However, retrieving documentation needed for bootstrapping the experimentation process, as well as identifying the appropriate resources in such complex environments can be a very tedious task, especially for new users of the RI that are not yet familiar with the specific methodologies that need to be followed in order to get started with the experimentation process.

To this aim, tools that increase the user-friendly interface for on-boarding users with the facilities are needed to increase the attractiveness and overall usability of the RI. Generative AI and Large Language Models (LLMs) have emerged as tools that can simplify to a large extent the creation of new human interpretable content based on existing knowledge that they have access to. Based on their training inputs, such models are able to create content by combining knowledge from multiple sources, including even automatic code generation, that can be used in the context of user support for RIs.

In this context, sIAIces was created, an LLM-based chatbot for lowering the barrier for new users experimenting with the SLICES-RI. sIAIces uses the well-established Generating Pre-trained Transformer (GPT) 4 model for generating output for accessing the infrastructure, including selecting the most appropriate nodes depending on the type of experiment that the users need to conduct, while receives as input the documentation of each SLICES-RI testbed (incl. hardware capabilities, supported types of experiments, etc.). Hereunder we present the design choices, lessons learned and recommendations from the development and the model training process, and a framework evaluation based on the inputs provided from the online documentation sites of the different SLICES-RI nodes.

6.1.1. LLM Architecture and Configuration

To design sIAIces, we employed a structured framework development strategy, which consists of four key stages: data preparation, designing and implementing a RAG system, LLM fine-tuning, and user querying via an appropriate environment. The overall integrated architecture can be observed in

⁶ Serge Fdida, et al., SLICES, a scientific instrument for the networking community, *Computer Communications*, Volume 193, 2022, Pages 189-203, ISSN 0140-3664, <https://doi.org/10.1016/j.comcom.2022.07.019> [Last accessed 25 June 2024]

Figure 9. The initial stage involves acquiring SLICES data from diverse websites using the web scraper further described below. Subsequently, raw data is appropriately enhanced by enriching both the textual content and the code content. This enhancement process involves leveraging the online GPT-4 model, which is manually utilized to upload our data for further enrichment. In addition, our domain expertise is leveraged to supplement this process.

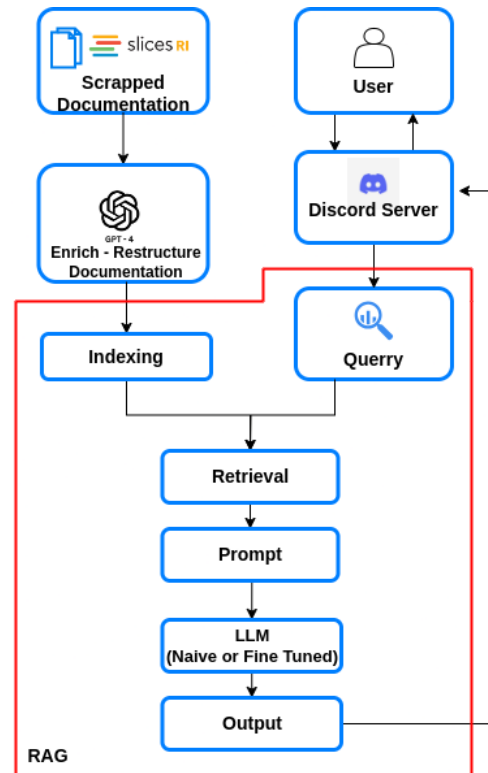


Figure 9: sAlces LLM system architecture

The Retrieval-Augmented Generation (RAG) system consists of a pipeline of different functions that are responsible: 1) for breaking and storing the data in suitable data structures (indexing), 2) retrieving the most similar information to the user query from the data (retrieval), 3) constructing a prompt that contains the retrieved information as long as the user query (prompting), 4) using an LLM to get the response of the user query (LLM-Output).

Following these, we proceed with deploying the chatbot service onto one of our SLICES servers. In order to provide a user-friendly interface, we have established a link between the SLICES server (housing the deployed chatbot) and a Discord server. This integration allows users to interact with the chatbot directly through the Discord server, simplifying the process and providing a more intuitive interface for the end user. The server is also integrated to the online portal, to provide a conversational experience to the users that want to conduct experiments within the SLICES infrastructure. The basic concepts and functionalities of our system are further explained in the subsections below.

6.1.2. Data acquisition and augmentation

The data that feeds our model are sourced primarily from the official website of SLICES-RI as well as the links within its contents that point to educational and training material, such as tutorials for



students and researchers to become familiar with various frameworks and technologies that each integrated testbed of SLICES-RI is equipped.

To be able to automatically retrieve the required information from the aforementioned documentation websites, a web scraper was developed, designed to operate in two stages. In the first stage, all the links and URLs inside the documentation pages are tracked recursively from the root website, with the help of the *urllib* library.

After gathering all the referral URLs on the visited websites, duplicate URLs are removed to create a unique set of URLs. Subsequently, a manual review and evaluation of the content of the web pages is performed, to evaluate its relevance and value to be included in the dataset. In the second stage of the web scraper, the web page data is retrieved and converted from HTML format to text through the *Beautiful Soup* library.

Given the variation in website designs among the SLICES-RI testbeds, the code was adapted according to each website-specific design to extract the information correctly. At this stage, the data from each website is stored in distinct files, allowing a modular approach to removing/adding new information to the model.

Fine-tuning is mandatory to make our system efficient. Although the retrieved data from the documentation sites are interpretable by humans, they can be less beneficial for LLMs. LLMs, in contrast to humans, struggle to understand complexity and ambiguity since they lack real-world understanding.

The learning process for humans versus the one for LLMs significantly differs, especially in tutorials. A human can approach tutorials already having a basic understanding of the technology and the ontology of the various entities and components used in them. Foundational knowledge makes it easier for humans to make new connections among the different parts of information as long as they can easily grasp new concepts. Tutorials mainly point to hands-on experience, not providing depth in theory. In addition, the reader of the tutorials can refer to other sources besides the tutorial and its own references. However, none of these applies to LLMs, since LLMs lack ontological understanding. They realize text based on patterns observed using statistical correlations in their training data and not using relationships to convey meaning. LLMs also suffer from Referencing Limitations since their knowledge is restricted only to their training and In-context Learning (ICL) dataset, without external post-training resources.

Our raw dataset, provided by the documentations, presents gaps and deficiencies for appropriate LLM training, with respect to clarity, depth, or specificity of different information. To address such vulnerabilities, we enriched our dataset with comprehensive vocabulary as well as with an extensive and precise description of the theory (e.g. component's architecture, terminology, etc.). We employed the popular ChatGPT AI system⁷ based on GPT4 model⁸ to perform this enhancement. Following that, we restructured the generated text, into smaller paragraphs with titles emphasizing the context of their body. We validated rigorously, and thoroughly checked the data enhancement to ensure that we did not add vague information and pollute our data with noise, since ChatGPT can make mistakes.

⁷ OpenAI, "Introducing ChatGPT," <https://openai.com/blog/chatgpt> [Last accessed 25 June 2024]

⁸ OpenAI, "GPT-4," <https://openai.com/gpt-4> [Last accessed 25 June 2024]



6.1.3. In context learning and fine tuning

A diverse amount of publicly available internet text data, such as websites, Wikipedia, Common Crawl, articles, as well as scientific papers, have been used to train GPT. This extensive training leads LLMs to identify patterns observed using statistical correlations, and identify grammar, syntax, and semantics, achieving a significant amount of general knowledge. Although GPT has tremendous performance in general-purpose knowledge, it looks very vulnerable to specific domains, such as providing assistance to SLICES RI infrastructure, where domain-specific knowledge and contextual understanding are crucial. For an LLM model to be updated with special-purpose data, there are two popular methods: in-context learning via a RAG system and fine-tuning.

The first approach, in-context learning, feeds the LLM with special-purpose data and by providing extra information in the prompt, without any parameter updates or fine-tuning. This method utilizes a RAG system to insert context into the input prompt, allowing the LLM to generate responses by leveraging its reasoning capabilities. Our RAG implementation follows the Naive RAG architecture as described in ⁹ and has three necessary axes: Data Ingesting, data indexing and data retrieving.

Data ingestion refers to how the contents of files in different formats (CSV, JSON, XML, PDFs, etc.) can be converted into data structures. Subsequently, Data Indexing is responsible for organizing, storing, and efficiently accessing the converted data.

Several proposed index schemes can be used in different use-case scenarios. A node in the data Indexing world represents a chunk of source data from a document, regardless of its type. There are multiple index schemes available, such as List Index, Vector Store Index, Tree Index and the key-word index. The data retriever is responsible to identify the most similar data nodes of the indexed dataset to include its content to the prompt. Our system utilized the open-source library Llama¹⁰ to manage the data ingestion, indexing and retrieving procedures of sIAlces implementation.

Fine-tuning (FT) is another strategy to integrate specific-domain knowledge into an LLM than ICL. The strategy mainly focuses on training a pre-trained model using domain-specific datasets to incorporate data from specialized fields. Using a pre-trained model offers considerable advantages compared to an LLM that is exclusively trained with our private data. One of the most important ones is efficiency, since training the model from scratch requires significant computational resources, time, and an enormous dataset to learn language understanding. FT changes the model's weight through additional training, improving performance on domain-specific tasks. During the FT process, great attention was given to the parameters that will be modified so that they maintain the performance of our model and not to become overfit.

The strategy of FT is not as straightforward as that of In-context learning. Since the FT process is designed to adjust the model's parameters using examples of input-output pairs, we have to change the format in which the data are fed to the model. Instead of giving the information in raw text, all information intended for training should be converted into a query-answer format to be used as a prompt. Following this, our dataset should be structured in a prompt-completion format as described in the Fine-tuning Guide¹¹ We decided to fine tune the model with a portion of the dataset that LLM had difficulty responding when using only ICL. OpenAI does not support fine-tuning for GPT-4 yet,

⁹ Gao, Yunfan, et al. "Retrieval-augmented generation for large language models: A survey." arXiv preprint arXiv:2312.10997 (2023).

¹⁰ J. Liu, "LlamaIndex," 2022, https://github.com/jerryliu/llama_index [Last accessed 25 June 2024]

¹¹ OpenAI, "Preparing Your Dataset for Fine-Tuning," <https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset> [Last accessed 25 June 2024]



since it is at an experimental stage at the time this document is written. For our results, we used GPT-3.5-turbo-0125 limiting the model's capabilities compared it to the GPT4.

ICL and FT are two complementary techniques that can enhance a model's capabilities on distinct levels without being mutually exclusive. Our chatbot assistant is of considerable complexity, primarily because it must deliver not just theoretical knowledge but also detailed instructions and advanced information across a multitude of configurations, console outputs, commands, architectural concepts and many more. To deal with this issue, it is critical to integrate both FT and ICL into our development process. With ICL we are capable of providing the model with all the information, including even the slightest detail. On the other hand, FT is inevitable, since it helps the model to deeply understand our specific dataset, enabling it to interpret the meanings of different data types and follow advanced procedures.

6.1.4. Evaluation and indicative results

To assess the chatbot's performance before and after data augmentation, we have generated 25 multiple-choice questions from our datasets. It is important to mention that each question was formulated across a diverse range of data sources and was structured to have answers that had small deviations from the correct one, in order to increase the level of difficulty. Subsequently, the questions were first presented to the RAG system with the original dataset, and then to the same model indexed with the enhanced dataset. The RAG system was prompted the same way for both the raw and the enhanced dataset, outlining the role of RAG as a chatbot assistant, as well as generating responses based only on the information provided to it. As depicted in Figure 10, the correctly-answered questions percentage by the model with the original-data was 76%, in contrast to the enhanced-data model that achieved 96%. This difference suggests a huge performance improvement, as it underscores the importance of data augmentation in enhancing the chatbot's understanding and response accuracy.

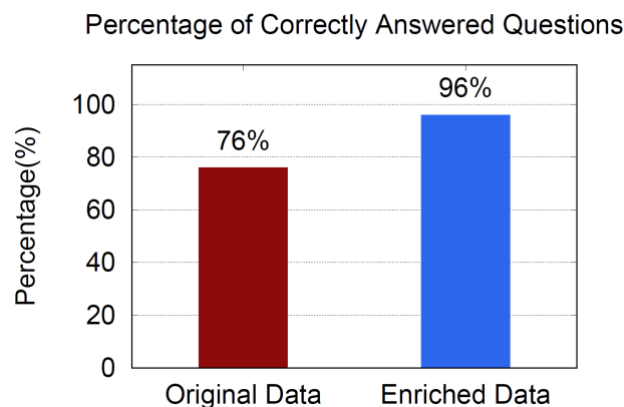


Figure 10: Percentage of Correctly Answered Questions

Regarding the open-ended answers, the model that uses the enhanced data provided responses with more precision and included more information compared to the model using the raw dataset. When it came to modifying code and configuration files, both models achieved the expected outcomes, with the responses from the RAG system using the enriched data being more solid, especially in the reasoning part. For example, as depicted in Table 2, when asking the same question "Which nodes of



the NITOS testbed should I use to deploy the 5G RAN using USRPs according to 5G Blueprint¹²?” to the GPT-4 model, the chatbot did not respond in the case of raw data, in contrast to the enhanced one, which answered correctly. At this point we note that the same information was present in both datasets, but the GPT-4 model could not successfully identify or interpret it to produce good responses. Some more indicative responses to different questions are provided in the accompanying figures below.

Table 2: Question Responses per Model & Dataset

Question	Which nodes of the NITOS testbed should I use to deploy the 5G RAN using USRPs according to 5G Blueprint?
Original Data with GPT-4 Model	The given context does not provide specific details about the nodes of NITOS that can deploy 5G RAN using USRPs according to the 5G Blueprint. Detailed information regarding this would be required to provide a precise answer.
Enhanced Data with GPT-4 Model	The NITOS testbed nodes that are capable of deploying the 5G RAN using USRPs, according to the 5G Blueprint, are: node055, node056, node059, node060, node067, node068, node071, node072. These nodes are equipped with USRP B210 models, which are connected via USB 3.0. The USRP B210 devices are particularly significant for 5G research due to their wide frequency range and high bandwidth capabilities, enabling researchers to simulate and analyze various aspects of 5G networks.
Enhanced Data - GPT-3.5	To deploy the 5G RAN using USRPs according to the 5G Blueprint in the NITOS testbed, you should utilize the nodes equipped with USRP N210 and USRP B210 devices. Here are the specific nodes that you should use for this purpose: USRP N210 Nodes: node052, node064, node080 USRP B210 Nodes (Connected via USB 3.0): node055, node056, node059, node060, node067, node068, node071, node072.
Enhanced Data - Fine-Tuned GPT-3.5	The NITOS testbed encompasses specific nodes equipped to facilitate the deployment of 5G RAN experiments utilizing USRP devices. According to the 5G Blueprint, the following NITOS nodes are suitable for this purpose: node055, node056, node059, node060, node067, node068, node071, node072. These nodes are instrumental in realizing advanced 5G RAN scenarios, enabling researchers to leverage the versatile capabilities of USRP B210 models for comprehensive and cutting-edge experiments.

¹² SLICES Starting Community, “SLICES-RI Blueprint,” <https://doc.slices-sc.eu/blueprint/>, 2023 [Last accessed 25 June 2024]

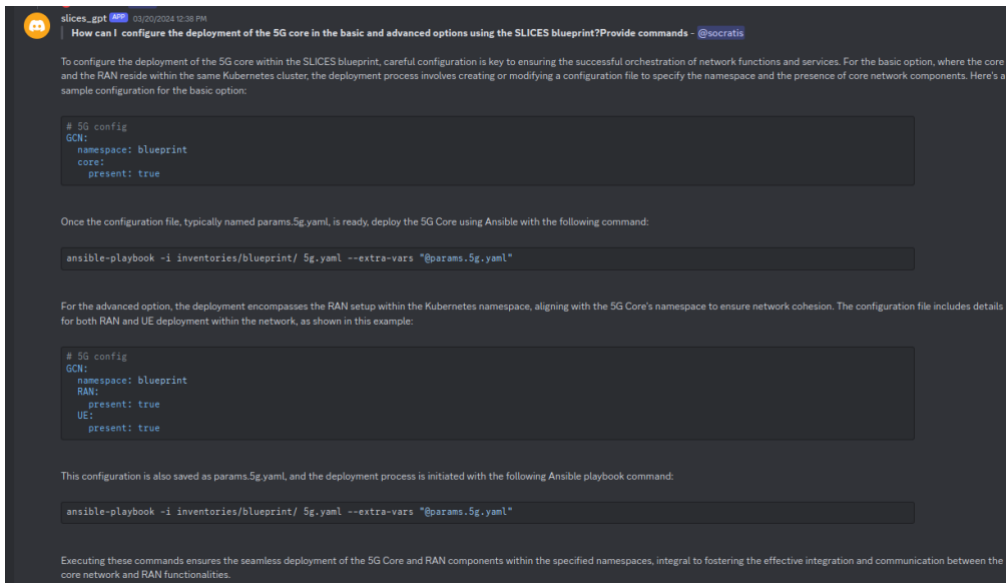


Figure 11: Indicative responses from sAlces on the 5G blueprint

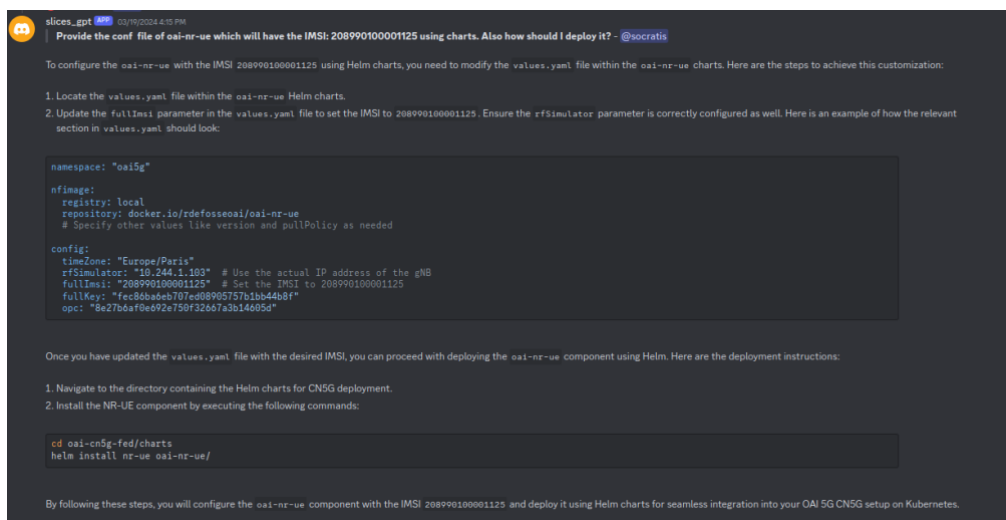


Figure 12: Indicative responses from sAlces, when trained with the OpenAirInterface documentation

6.1.5. Future Work and exploitation plans

The initial results of the sAlces chatbot seem very promising, as they can aid the experimenter to easily identify the resources of the SLICES testbeds, based on their needs. In the future, we foresee to improve the capabilities of our chatbot by developing an execution multi-agent mechanism, that will automate the experimental workflow and the process of experimentation. This includes several stages such as experiment design, resource reservation, experimentation and result collection. By invoking the respective APIs in the portal, and the rest of the SLICES testbeds, this process shall significantly ease the experimentation process, as no site-specific and API specific knowledge will be required by the experimenters. In addition, we aim to develop a closed control loop system that will utilize valid experiment results that have been conducted in the SLICES-RI Infrastructure and subsequently include this data to the main dataset. This will enhance the chatbot's ability to adapt to the needs of the research community, moving beyond tutorial support.



6.2. EOSC based AI tool for suggesting resources based on the behaviour of users

In this section, an outline is presented of the fundamental architecture, operational mechanisms, and the impactful role of a system aimed at improving access to resources tailored to individual user preferences. The introduction of a recommendation system marks a significant stride towards improving resource discovery and utilization across various research infrastructures. Through the utilization of a diverse set of technological components and a recommendation engine, the system aims to transform the traditional approach to resource recommendations and access, thereby enhancing user engagement and supporting the effectiveness of research and exploration efforts.

Overview of the recommendation system:

The recommendation system was designed as a platform to help users navigate through a wide array of resources, steering them towards those that align with their interests and requirements. By combining user interaction data and algorithmic analysis with feedback mechanisms, the system aims to provide personalized recommendations.

Objectives and scope:

The primary objective of this recommendation system is to implement a solution that not only simplifies the process of resource discovery but also ensures that the recommendations are relevant, personalized, and valuable to the user. The scope of this system spans various dimensions, including the integration with existing research infrastructures, harnessing the power of data analytics and machine learning for recommendation generation, and providing a seamless user experience through a user-friendly end interface. Furthermore, the system aims to support a wide range of users, from researchers and academics to students and industry professionals, thereby broadening the horizon of its applicability and impact.

Significance for research infrastructure:

The deployment of this recommendation system within research infrastructures transforms the approach to accessing and utilizing resources. By automating the process of recommendation based on user-specific data and preferences, the system elevates the efficiency of resource discovery. It enables users to uncover relevant resources that might have remained unnoticed, thus fostering a deeper exploration of subjects and facilitating cross-disciplinary discoveries. Moreover, by streamlining access to resources, the system aids in the effective utilization of available information, promoting a culture of informed research and learning.

6.2.1. System Architecture

The architecture of the recommendation system is crafted to facilitate a seamless and efficient process of generating and delivering personalized recommendations. The foundation of this system is the integration of various technological components, each playing a key role in its overall operation. From the user interface to the algorithm powering the recommendation engine, the architecture is designed to ensure scalability, reliability, and performance. This chapter provides an in-depth exploration of the system's architecture, detailing the roles and interactions of the user interface, the API layer, the database, and the recommendation engine.

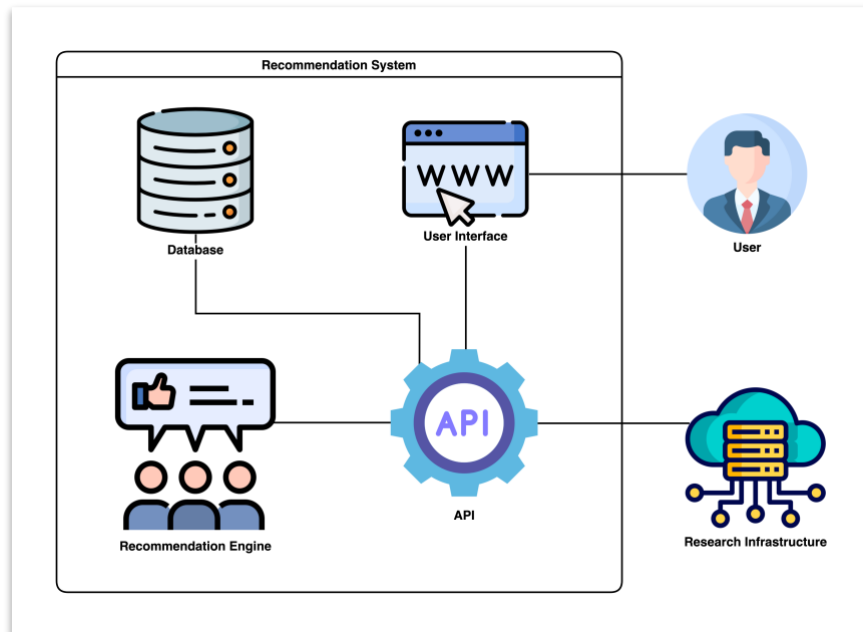


Figure 13: Architecture of the recommendation system

User interface:

The user interface, developed using Angular, serves as the primary point of interaction between the user and the recommendation system. Angular, known for its capability to build dynamic and responsive web applications, facilitates the creation of a user interface that is not only intuitive but also adaptable to various user needs. This interface allows users to navigate through different resource categories, make selections, and receive personalized recommendations with ease. Its design focuses on user experience, ensuring that the system is accessible, engaging, and efficient in guiding users toward their areas of interest.

API:

The API layer is central to the system's functionality, utilizing Django to handle the intricate flow of data and requests between the front-end and back-end components. This essential layer ensures robust data processing and seamless interaction across the system's architecture. It serves as a foundational pillar, enabling the system to manage complex queries efficiently and maintain consistent performance across various operations. Its design underpins the system's capability to evolve, supporting enhanced features and integrations that cater to the dynamic needs of users.

Database:

The PostgreSQL database stands as the central repository for all data within the system. Chosen for its performance, reliability, and scalability, PostgreSQL efficiently stores and manages user data, resource information, and interaction logs. Its role is critical in ensuring that data is accurately captured, stored, and available for retrieval, which is essential for the recommendation engine's operation. The database's design and structure are optimized to support complex queries and analyses, facilitating quick and efficient data access and manipulation.



Recommendation engine:

The recommendation engine is the core component responsible for generating personalized resource suggestions. Powered by Elasticsearch and leveraging the Approximate Nearest Neighbors (ANN) algorithm, the engine analyzes user data to identify patterns and preferences. It dynamically generates recommendations based on similarities among users and their interactions with resources. Elasticsearch enhances the engine's ability to process large volumes of data swiftly, ensuring that recommendations are both accurate and timely. This component embodies the system's intelligence, adapting to user behaviors to improve recommendation relevance continually.

6.2.2. Workflow of the recommendation generation process

Central to the efficacy of the recommendation system is its ability to generate tailored suggestions for users, a process that hinges on a detailed interplay of data collection, analysis, and application of machine learning algorithms. This chapter delves into the intricate process through which the system curates personalized recommendations, illustrating the blend of technology and methodology that underpins this dynamic operation.

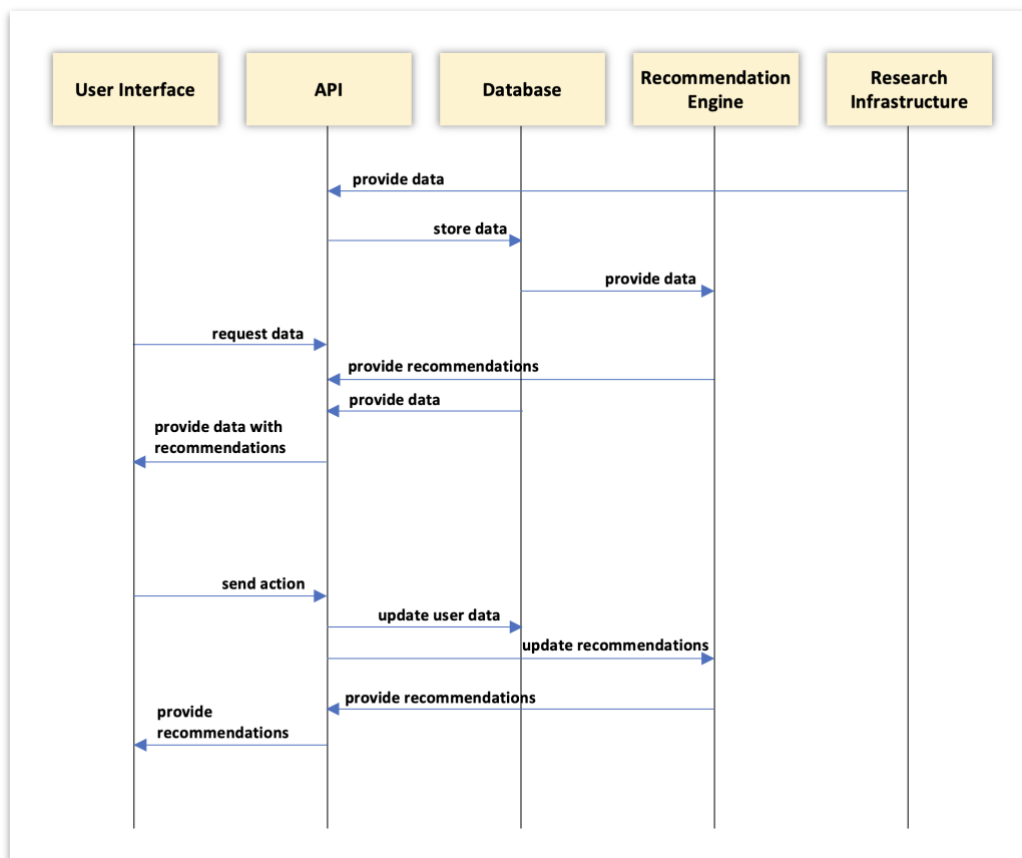


Figure 14: Simplified workflow of the recommendation system



Data collection and preprocessing:

The process of generating a recommendation begins with the careful collection of user interaction data. The Angular user interface is instrumental in capturing these interactions, which are subsequently relayed to the back-end through the Django API layer for processing.

Once collected, this data undergoes a series of preprocessing steps designed to optimize its utility for recommendation generation. These steps include data cleaning to remove any inconsistencies or irrelevant information, normalization to ensure data is in a standardized format, and transformation processes that prepare the data for analysis. This stage is critical for ensuring the accuracy and relevance of the recommendations generated.

User profile building and segmentation:

With preprocessed data in hand, the system then constructs comprehensive user profiles that reflect each individual's preferences, behaviors, and interaction patterns. These profiles are dynamic, evolving with each user interaction to accurately represent changing interests and preferences over time.

Users are segmented based on similarities in their profiles, with the system identifying groups of users who share common interests and behaviors. This segmentation enables the recommendation engine to efficiently identify and leverage patterns across user interactions, enhancing the personalization of the recommendations provided.

Recommendation engine operation:

At the core of the recommendation generation process lies the recommendation engine, powered by Elasticsearch and the ANN algorithm. This engine takes the segmented user data and applies advanced analytical techniques to identify patterns and relationships among users and resources.

The ANN algorithm is particularly adept at identifying users with similar profiles (nearest neighbors) in real-time, enabling the system to adapt its recommendations based on the most current data. This dynamic aspect of the recommendation process ensures that suggestions remain relevant and closely aligned with each user's interests and behaviors.

Generating recommendations:

With the nearest neighbors identified, the engine aggregates the resources most popular and relevant among these users. It calculates the frequency and significance of each resource within the neighbor group, identifying those with the highest relevance. The top resources are then compiled into a list of recommendations. This list is dynamic, adjusting in real-time to changes in user behavior and preferences to ensure the highest degree of personalization.

Delivering recommendations to users:

The final step in the recommendation process is the delivery of the curated list of suggestions to the user. Through the API layer, these recommendations are transmitted back to the front-end interface, where they are presented to the user. This interface is designed to maximize engagement and ease of access, encouraging users to explore the recommended resources further.



6.2.3. Integration with research infrastructures

The seamless operation of the recommendation system is contingent upon its adept integration with existing infrastructures and its meticulous management of data flow. This chapter elucidates the mechanisms through which the system integrates direct and indirect data inputs, as well as the internal workflow that ensures the efficient processing of this data to generate recommendations. Understanding these aspects is crucial for appreciating the system's capacity to leverage diverse data sources and maintain a dynamic, responsive recommendation process.

API integration for direct data input:

A pivotal aspect of the system's integration capabilities is its API layer. This layer facilitates direct data input from users and other systems, acting as a bridge that connects the recommendation system with external data sources. The API is designed to be robust and flexible, allowing it to be adapted to handle a wide range of data formats and structures, thereby enabling the system to ingest a diverse array of data with ease.

The direct integration via the API not only allows for data acquisition but also ensures that the recommendation engine has access to the most current and comprehensive data sets. This immediacy and breadth of data are critical for maintaining the relevance and accuracy of the recommendations generated, allowing the system to dynamically adapt to new information and evolving user interests.

Netbox integration for research infrastructure data collection:

In parallel to direct data inputs, the system incorporates Netbox integration for gathering resources from research infrastructures. This specialized integration allows the system to passively collect a variety of resources, which are essential inputs for the recommendation engine, without the need for active user or system input.

By synchronizing data with the Netbox platform, the recommendation system maintains continuous access to infrastructure resources, facilitating a comprehensive understanding of the available information. This integration enriches the recommendation engine's capacity to tailor suggestions with heightened relevance and specificity, enabling it to adapt more fluidly to changes within the research infrastructure. As a result, users benefit from a dynamic and robust set of recommendations that reflect the current landscape of resources.

Workflow of data through the system:

The workflow of data through the system is orchestrated to ensure that data collected via both direct and indirect methods is effectively processed, analyzed, and utilized in the recommendation generation process. Upon collection, data is funneled into the PostgreSQL database, where it undergoes initial preprocessing to standardize and cleanse the dataset. This preparation is crucial for ensuring the data's utility in subsequent analytical processes.

From the database, data is directed to the recommendation engine, where it is subjected to a comprehensive analysis. Leveraging Elasticsearch and the ANN algorithm, the engine identifies patterns, similarities, and user preferences embedded within the data. This analysis forms the basis for generating personalized recommendations, which are then passed back through the API layer to the front-end interface for presentation to the user.





6.2.4. User interface and experience

The effectiveness of the recommendation system is not just based on the complexity of its algorithms and data processing capabilities but equally on the user interface's intuitiveness and responsiveness. This chapter delves into the system's user interface and the overall user experience it creates, highlighting the design principles it follows and the functionalities that improve how users interact with the system.

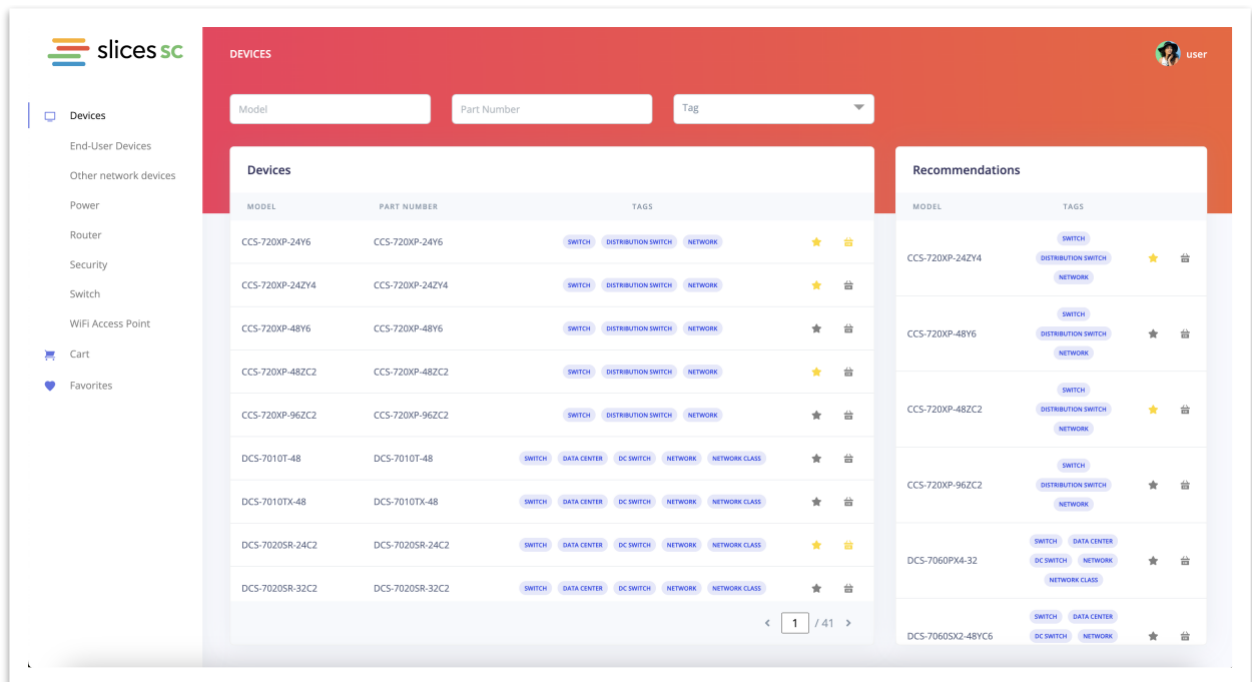


Figure 15: User interface of the recommendation system

Enhanced user interface functionalities:

The user interface, as shown in the figure, is furnished with functionalities that enable efficient and engaging system interaction. Central to these is the dynamic presentation of recommendations, structured in a format that is both visually compelling and easily navigable. Recommendations are displayed alongside related tags, offering users immediate contextual understanding and facilitating swift navigation.

Interactive features within the interface, such as search bars and filterable tags, allow for advanced querying capabilities. Users can effortlessly search for specific models, part numbers, or tags, and the system responds with relevant suggestions, enhancing the overall discovery process. The interface's design, as showcased, integrates these functionalities into a cohesive experience, enabling users to explore resources through a combination of filters and category selections.

Moreover, the system provides users with the ability to directly interact with listed resources. Icons for actions like adding to cart or favorites underscore the interface's capacity for personalization and convenience. Users can mark items of interest, which not only aids in resource management but also informs the recommendation engine of their preferences, further refining the suggestion mechanism.



Personalization through user feedback and interaction:

User feedback is seamlessly integrated within the interface, affording individuals the opportunity to rate resources, articulate their preferences, and add items to favorites or the cart directly. This feedback loop is critical for the system's adaptability, ensuring that the recommendations evolve to align with user behavior and input.

Within the user interface, resources are enhanced with categorized tags, facilitating quick identification and organization of items according to their relevance and user preferences. These personalization features are integral to the interface, focusing solely on streamlining resource interaction and customization, thus ensuring a tailored and efficient user experience.

Optimizing engagement through user-centric design:

The system employs a rigorous analysis of engagement metrics to fine-tune the user interface continuously. This demonstrates user-focused design, highlighting categories and favorite mechanisms that encourage users to interact with the system in a manner that aligns with their unique needs and preferences. The incorporation of these features is a direct response to user behavior, ensuring that the system not only meets but anticipates user needs, fostering a more engaging and satisfying experience.

6.2.5. Evaluation and results

The comprehensive evaluation of the recommendation system is vital for assessing its effectiveness, efficiency, and overall impact on the user experience. This chapter outlines the methodologies employed in testing and evaluating the system, presents the results obtained from these evaluations, and discusses the implications of these results for the system's performance and future development.

Methodology for testing and evaluation:

To rigorously assess the recommendation system, a multifaceted approach was adopted, incorporating both quantitative and qualitative metrics. Quantitative metrics provide objective measures of the system's performance in delivering relevant recommendations to users, while qualitative evaluations offer insights into user satisfaction, perceived usefulness, and the overall user experience through methods such as user surveys, feedback sessions, and usability studies.

A/B testing played a pivotal role in evaluating the performance of various recommendation algorithms and interface layouts. This approach entailed presenting two variants of the system to analogous groups of test users and scrutinizing the differences in how they interacted with each version, their levels of engagement, and their feedback. The insights gained from these comparative analyses were key in pinpointing the strategies that most effectively enhance the recommendation process and the design of the user interface.

Results from implementation:

The implementation of the recommendation system yielded significant findings, highlighting the strengths of the system and areas for further enhancement. Key results include:



- Significant engagement metrics were noted. Test users showed a pronounced interest in exploring the resources suggested by the system, highlighting the relevance and attractiveness of these recommendations.
- There was a marked improvement in the precision of recommendations over time, with the system becoming increasingly proficient in tailoring its suggestions to match the preferences and behaviors of test users. This advancement was primarily attributed to the system's advanced algorithms, which continuously adjusted recommendation criteria based on the feedback and interaction data from these users.
- Feedback collected from test phases, including surveys and interactive feedback sessions, consistently underscored the system's success in delivering highly pertinent recommendations. Test users commended the interface for its ease of use and the system's overall efficacy in uncovering new resources and information.
- Insights from A/B testing highlighted distinct preferences among test users for certain interface designs and recommendation approaches. These findings have been crucial for directing subsequent enhancements to the system, with the aim of refining its configuration to further improve engagement and satisfaction levels among test users.

Accuracy and efficiency of recommendations:

The evaluation effectively highlighted the system's ability to deliver accurate and efficient recommendations, with a particular focus on its adaptive performance as the number of test users increased. This scalability not only allowed the system to gather more comprehensive data but also to continually refine its approach based on user interactions. The feedback and data collected from these interactions enabled the recommendation engine to fine-tune its algorithms, enhancing its ability to predict and meet user needs more accurately. Through this process of constant learning and adaptation, the system fostered a positive feedback loop, leading to increased user satisfaction and a cycle of continuous improvement.

6.2.6. Partnerships and potential future roadmap

In this chapter, an outline is presented for the plan regarding partnerships and future development, with a focus on practical steps and strategic collaborations to enhance the recommendation system. The partnership with partners of EOSC Future is identified as a cornerstone of this plan, serving as a solid foundation for both current achievements and future initiatives. All proposals contained in this chapter are suggestions on how the continuation of this collaboration could potentially enhance and expand the capabilities of the system.

Leveraging collaborative partnership with EOSC Future:

Throughout the project's duration, collaboration with the EOSC Future initiative has been crucial in determining the development path of the recommendation system. This partnership, marked by a comprehensive exchange of knowledge and collaborative efforts, has been especially significant in the creation and refinement of recommendation algorithms and strategies. The joint work has not only propelled advancements in the system's capabilities but has also significantly increased its effectiveness. By leveraging the combined expertise and resources of the teams, a significant improvement in the precision and relevance of the recommendation algorithms has been achieved.



Furthermore, this collaboration has fostered a conducive environment for practical progress, enabling swift adaptation to technological changes and user expectations. The synergy between the teams has facilitated a deeper understanding of complex data patterns and user interactions, further refining the algorithmic approaches. Moving forward, the alliance with EOSC Future is set to deepen, with plans to explore advanced computational techniques and novel data processing methodologies. The ongoing partnership is built on a mutual commitment to practical advancements and a focus on developing reliable solutions that deliver tailored and meaningful recommendations. The collective efforts are aimed at enhancing the system's functionality and relevance, ensuring it effectively responds to the current and future needs of users.

Strategic enhancements to recommendation algorithms:

Continual improvement of the recommendation algorithms is paramount. Future initiatives may explore the integration of more sophisticated machine learning models, including deep learning approaches, to refine the system's ability to understand and predict user preferences with greater accuracy. Enhancements could focus on improving the system's contextual awareness, enabling it to discern and adapt to the nuances of user behavior and emerging trends in real-time.

Expansion of data integration capabilities:

The system's efficacy is fundamentally linked to the richness and diversity of the data it can access. Plans might include expanding its integration capabilities with a broader spectrum of data sources. This could involve forging partnerships with academic institutions, research organizations, and industry consortia to access a wider array of content and user interaction data. Such expansions would not only enrich the recommendations provided but also enhance the system's applicability and relevance across different domains.

User experience and interface innovations:

The user interface, the primary conduit of interaction with the system, is slated for significant enhancements. Future versions could incorporate more dynamic and interactive elements, leveraging advances in web technologies to create a more immersive and engaging user experience. Personalization features could be deepened, offering users more granular control over their preferences and how recommendations are presented to them.

Embracing new technologies and platforms:

Recognizing the rapid pace of technological evolution, the system's development roadmap might include the exploration and adoption of emerging technologies. This might encompass advancements in artificial intelligence, data analytics, and even blockchain for secure and transparent user data management. Additionally, the system could be adapted to new platforms and devices, ensuring accessibility and optimal performance across the spectrum of user interfaces.

Community building and collaborative features:

A strategic focus on community building and collaboration could lead to the introduction of features designed to foster interaction among users. This could include tools for sharing resources, collaborative filtering, and community-driven recommendation mechanisms. By enabling users to



connect, share insights, and contribute to the refinement of the recommendation process, the system could harness the collective intelligence of its user base, enriching the discovery experience for all.

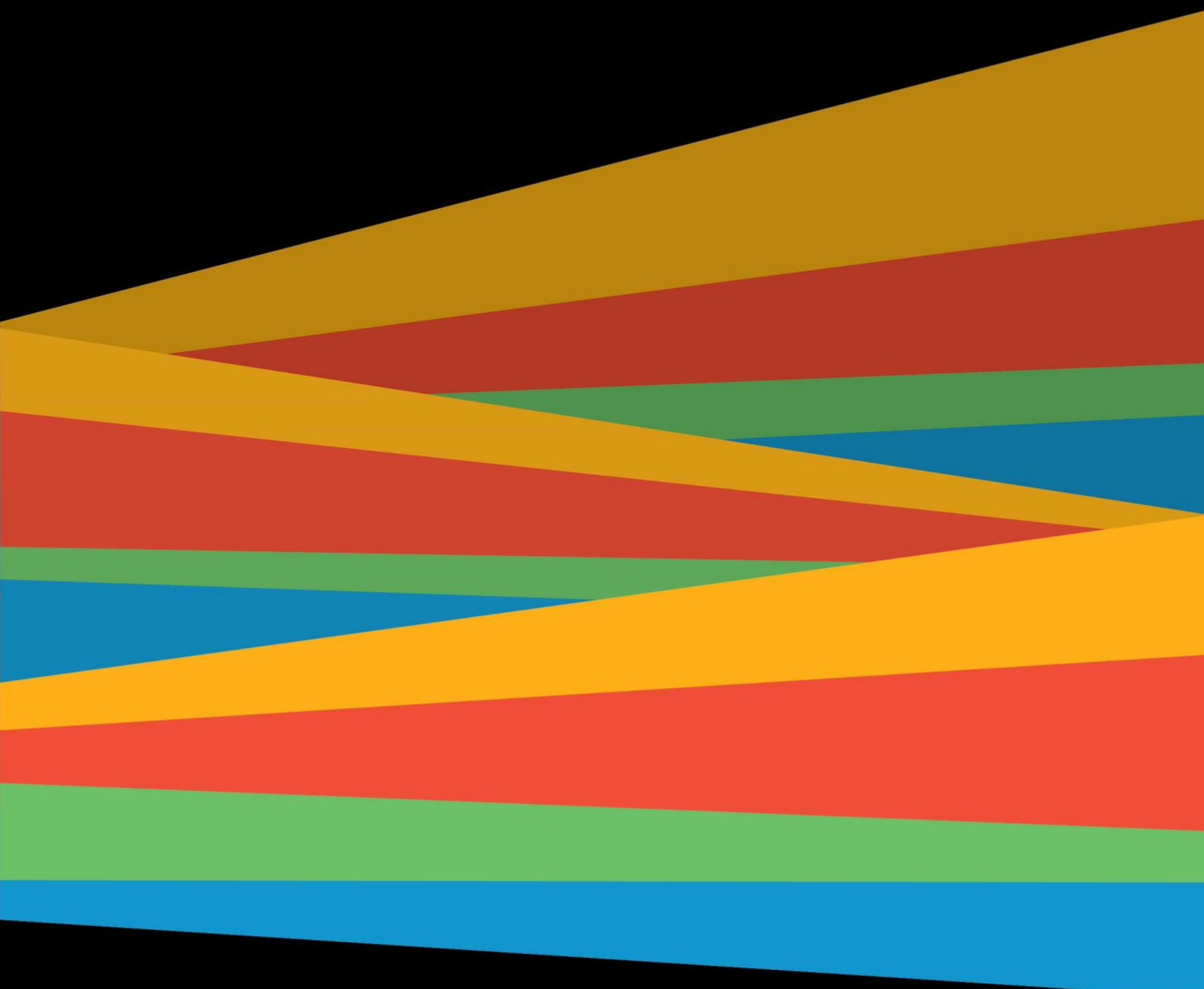
Commitment to data privacy and security:

As the system expands and evolves, a steadfast commitment to data privacy and security will remain central. Future developments should adhere to the highest standards of data protection, implementing state-of-the-art security measures and ensuring compliance with global data privacy regulations. This commitment is fundamental to maintaining user trust and ensuring the system's integrity and sustainability.

7. Conclusion

The SLICES-SC portal represents a significant advancement in providing a centralized, single sign-on, user-friendly platform for accessing and managing large-scale research infrastructure. By adhering to the outlined requirements, the portal ensures consistency with the SLICES-SC branding, enhances user experience, and supports detailed project and user management. The integration of advanced authentication methods, comprehensive logging and auditing, and robust API capabilities positions the portal as a vital tool for researchers and academics. Towards lowering the learning curve for experimenters and novice users of the facilities, AI assisting tools have been integrated in order to assist them with the process of getting their experiment started. The development and implementation of this portal not only streamline the experimental process but also foster collaboration and innovation within the research community. As we move forward, ongoing enhancements and user feedback will be crucial in maintaining the portal's relevance and effectiveness, ensuring it continues to meet the evolving needs of its users.





slicessc