slices SC

HORIZON 2020

H2020 - INFRAIA-2020-1

D8.5 Report of the second period of virtual access

Acronym	SLICES-SC
Project Title	Scientific Large-scale Infrastructure for Computing/Communication Experimental Studies – Starting Community
Grant Agreement	101008468
Project Duration	42 Months (01/03/2021 – 31/08/2024)
Due Date	31 August 2024 (M42)
Submission Date	13 September 2024
Authors	Nikos Makris, Stavroula Maglavera (UTH), (PSNC), (CNR), (EURECOM), Konstantinos Filis (OTE), Antti Pauanne (OULU), (INRIA), (IMEC), Barbara Tóth (SZTAKI), Ákos Hajnal (SZTAKI), (TUM), (TUM), Carmen Guerrero, Ignacio Berberana (IMDEA), Hassane Rahich (SU)
Reviewers	Cédric Crettaz (MI), Bartosz Belter (PSNC), Serge Fdida (SU)

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.

Scientific Large-scale Infrastructure for Computing Communication Experimental Studies **Starting Communities**

www.slices-sc.eu

Table of content

TABLE OF CONTENT 2
1. EXECUTIVE SUMMARY 3
2. ANALYSIS ON SLICES VA SERVICES AND TOOLS 4
2.1. SLICES OPEN DATA REPOSITORY 4
2.2. VISUALIZATION TOOLS FOR EXPERIMENTS AND DATASETS 5
2.3. REPRODUCIBILITY FRAMEWORK 11
2.4. SLICES VIRTUAL ACCESS METADATA 12
3. LESSONS LEARNED ON VA SERVICES FROM SLICES TEST-BEDS 13
3.1. SILECS-FIT / ONELAB, FRANCE 13
3.2. NITOS – UTH, GREECE 13
3.3. 5G TEST NETWORK (5GTN), FINLAND 14
3.4. ILABT – IMEC, BELGIUM 15
3.5. HUN-REN CLOUD – HUN-REN SZTAKI 18
3.6. TUM LAB, GERMANY 18
3.7. OTE, GREECE 19
4. LESSON LEARNED ON VA SERVICES FROM OPEN CALLS EXPERIMENTS 20
5. VIRTUAL ACCESS USABILITY TEST 21
6. NEXT STEPS IN SLICES VA SERVICES 24
7. CONCLUSION 25

1. Executive Summary

SLICES-RI is a strategic asset facilitating innovation and knowledge advancement across the domain of digital sciences. The final goal of SLICES is to provide researchers with advanced tools and resources that go beyond individual or institutional capacities and promote collaboration, community-building and the application of scientific standards. SLICES Virtual Access (VA) services enable scientists to use these essential resources without the necessity of being physically present.

SLICES-SC project has as objectives the definition of a common framework for VA services, the initial implementation of VA services by leveraging up by the experience from the Open Calls Experiments, the Transnational Access (TA) services, the deployment of Blueprint services, the training of users and the local activities of the distributed SLICES legacy testbeds. This fruitful experience led us to learn and highlight the implications of the operational processes and the dynamics of iteration between the infrastructure and the user communities. It is important to note that for most, only transnational access to the physical infrastructure was provided and that little experience existed on virtual access. We have identified advantages, obstacles, best-practices, strategies and recommendations to navigate and mitigate the challenges to be faced by VA services in the future implementation and operation phases.

This document includes a report on the activities conducted to develop SLICES VA services during the second period of the project. This report includes an analysis of the VA services and tools initially created during the first period and based on datasets repositories, tools (reproducibility and visualization) and the initial seed on the contribution to a new standardization on a common metadata framework named Metadata Registry System (MRS) and SLICES Fair Digital Object (MRS/SFDO).

This report also includes lesson learned by the locally distributed test-beds and the experience on running open calls from experiments in relation with the definition, provisioning and impact on the research infrastructure of the future SLICES VA services.

2. Analysis on SLICES VA Services and Tools

This deliverable is focusing on the VA related activities that took place during the second period of SLICES-SC project. The vision of SLICES for allowing VA services is to offer a pan-European operational networking and computer infrastructure to facilitate scientific research with instrumentation and experimentation capabilities. The data and tools offered through VA has as objective to assist in either facilitating easy implementation with SLICES-RI, i.e the reproducibility framework, training future users in SLICES framework and technologies, i.e SLICES Academy, or adding up to the knowledge of the community through datasets, analytical results and visualization tools. This section describes the latest updates about the SLICES Open Data Repository and the software and tools that were integrated to SLICES-RI to support and enhance VA.

2.1. SLICES Open Data Repository

During the first period we focused our efforts in the creation of CKAN and GitLab repositories that let the automatic registration of new datasets and tools available for the research community. After this initial setup, during the second period we have been focused on disseminating its usage and consolidation, both internally and externally to attract a wider research community. Internally, the SLICES-SC partners have actively provided new datasets and update the tools. Externally, the main source of activities has come from the external experimenters, mainly from the Open Calls, the SLICES Blueprint deployments and the legacy testbeds that have the commitment of contributing actively by populating the SLICES-SC repositories.

The datasets can consist of experiments descriptions for advanced and complex experiments (5G/6G, wireless sensors devices, IoT, cloud and AI/ML for networking) with the files needed to set up an experiment, the data used in an experiment and the results of the experimentation process. Furthermore, the datasets can encompass the documentation on how to run an experiment. The following figure shows the evolution of that contribution in terms of new datasets and registered users in the SLICES repositories during the two reporting periods.

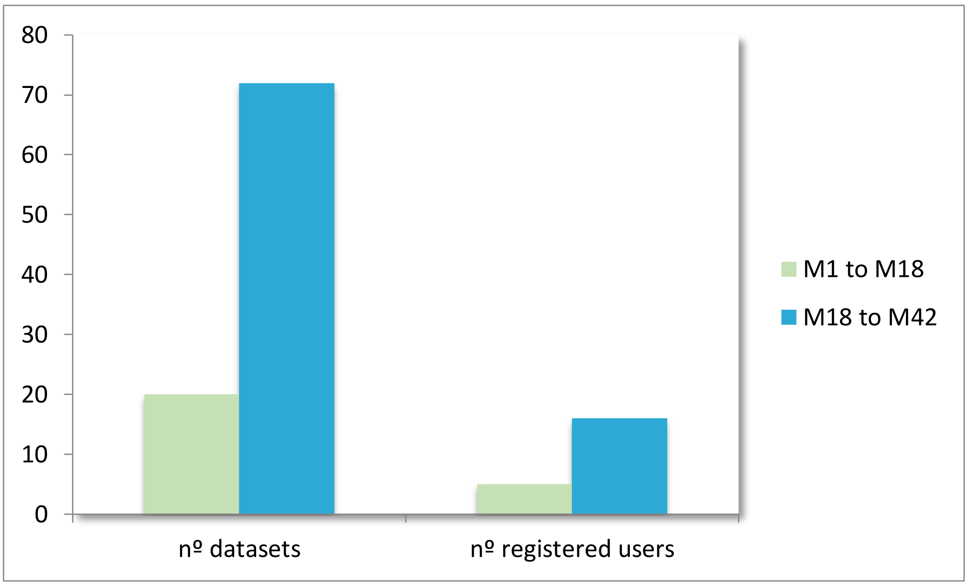

Figure 1. Evolution of datasets and users in the open data repository

The variety of datasets in terms of their classification on the technical fields and characterization currently available in the repository is graphically described in the following figure.

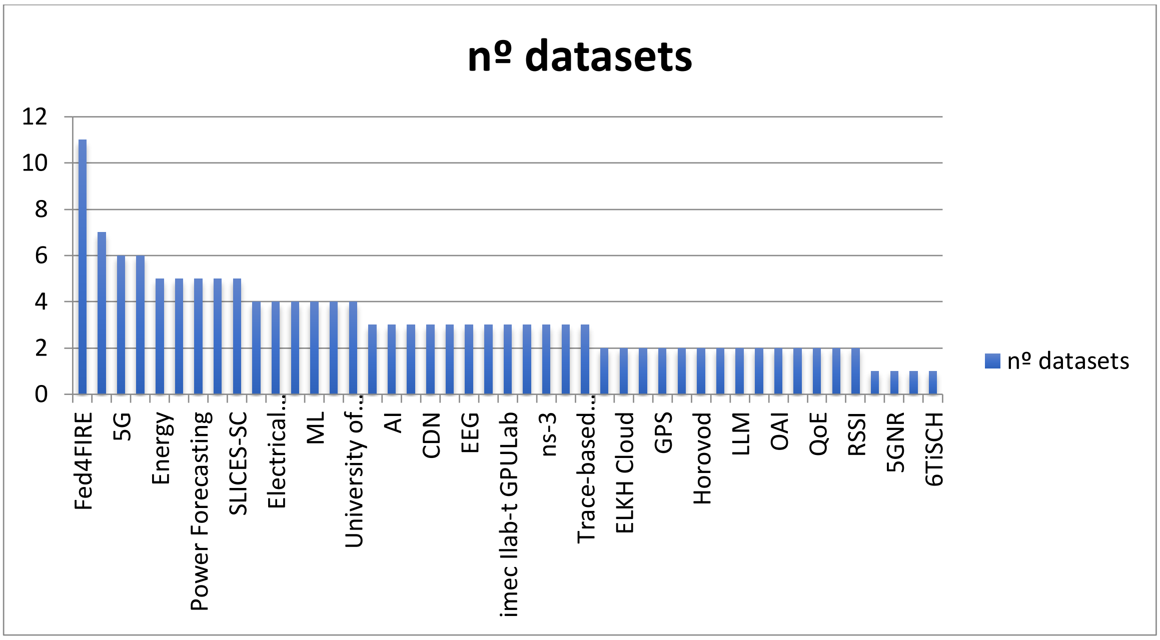

Figure 2. Classification of the datasets

As a main conclusion of this analysis is that even though there exist a high activity in terms of experimental results, there is a limited impact in the number of data sets uploaded in the central repository. The fragmentation on the format and categorization of the datasets is also very limited and clearly identify the need of a standard framework that integrates VA services in the full lifecycle of SLICES services provisioning. This is mostly due to the best practices of the research community and, their little incentive to share their data openly and the lack of a managed solution to host them.

2.2. Visualization tools for experiments and datasets

Data visualization and depiction have seen significant advancements over the past few years. Visualization tools are essential for analyzing datasets and presenting experimental results in an understandable manner, as well as providing the appropriate hooks for cross-correlation of results from different experiments in an easy and user-friendly manner. Such tools are the cornerstone of the SLICES-RI VA, as they present to the experimenter with a direct view of the results of their collected experiment, so as to gain direct feedback on their status (e.g. misconfigured parameters, experiment failed, etc.), as well as enable them to further post-process their results towards producing outcomes with scientific impact.

For the visualization of experiments, several tools have been considered. These tools range from basic charting libraries to sophisticated platforms that offer interactive and dynamic visualizations. Indicative state-of-the-art tools include matplotlib[1], Plotly[2], Tableau[3] and Grafana[4]. Such tools have

[1] S. Tosi (2009). Matplotlib for Python developers. Packt Publishing Ltd.

[2] L. Podo, & P. Velardi, (2022, October). Plotly. plus, an improved dataset for visualization recommendation. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (pp. 4384-4388).

[3] S. Batt, T. Grealis, O. Harmon, & P. Tomolonis, (2020). Learning Tableau: A data visualization tool. The Journal of Economic Education, 51(3-4), 317-328.

[4] M. Chakraborty, & A. P. Kundan, (2021). Grafana. In Monitoring cloud-native applications: Lead agile operations confidently using open source software (pp. 187-240). Berkeley, CA: Apress.

different applications and learning curves, relying either on CLI access, or providing an intuitive GUI. For practicality reasons, and better representation of the results to the experimenters, we opted for the Grafana visualization tool, that allows creating dynamic, real-time dashboards and visualizations. Grafana integrates with a wide variety of data sources, including time-series databases like Prometheus[5], InfluxDB[6], and Elasticsearch[7]. Real-time monitoring and measurement depiction is key for Grafana, while also providing an extensive plugin ecosystem with support for a wide range of data sources.

2.2.1 Metrics Collection

Key for the depiction of measurements is their collection from the different experimental instances. Several tools are able to provide scraping of metrics from different experimental instances. For SLICES we opted for the Prometheus monitoring suite, that is an open-source systems monitoring and alerting toolkit. Prometheus has grown into a robust ecosystem used widely in the industry for collecting and querying metrics, especially for cluster-based data within Kubernetes, with the kube-prometheus-stack[8] providing a complete solution for measurement collection, processing and depiction.

Prometheus

Prometheus stores data as time series, allowing for easy cross-correlation of data coming from different sources based on their timestamps. It mainly supports a query language (PromQL) designed for time series data, enabling complex operations and aggregations. Prometheus is supporting of-the-shelf automatic service discovery, enabling the dynamic discovery of targets to monitor, whenever they are alive. Accompanying libraries allow the seamless integration of any application that generates data to the Prometheus monitoring suite. Known applications of Prometheus include infrastructure monitoring (CPU/Memory/Disk/Network), application performance monitoring, and automatic alerting for applications.

The backend for measurements is usually based on TSDB (Time Series Data Base), that allows flexible measurement collection of such timeseries data. Nevertheless, several other backends are supported (e.g. S3 based access, InfluxDB, etc.). For the initial deployment of monitoring within SLICES-SC, we opted for the TSDB instance. Towards scaling the deployments, and allowing the experimenters to access a single point for collecting the measurements, we selected the case of measurement aggregation to a single central point. Towards achieving this, a local Prometheus instance is instantiated in each local cluster with ephemeral storage, but with the remote-write properties of the server being enabled for pushing each measurement to a central location. The storage of the central location is persistent, and allows the measurements coming from each cluster to be labeled accordingly in order to discriminate from which instance they belong to. The figure below shows the basic architecture for the remote-write functionality of Prometheus.

[5] J. Turnbull, (2018). Monitoring with Prometheus. Turnbull Press.

[6] K. Ahmad, & M. Ansari, (2017). Hands-on InfluxDB. In NoSQL (pp. 341-354). Chapman and Hall/CRC.

[7] Elasticsearch, B. V. (2018). Elasticsearch. software], version, 6(1).

[8] Kube-prometheus-stack, https://github.com/prometheus-community/helm-charts/tree/main/charts/kube-prometheus-stack [Last accessed: 12 September 2024]

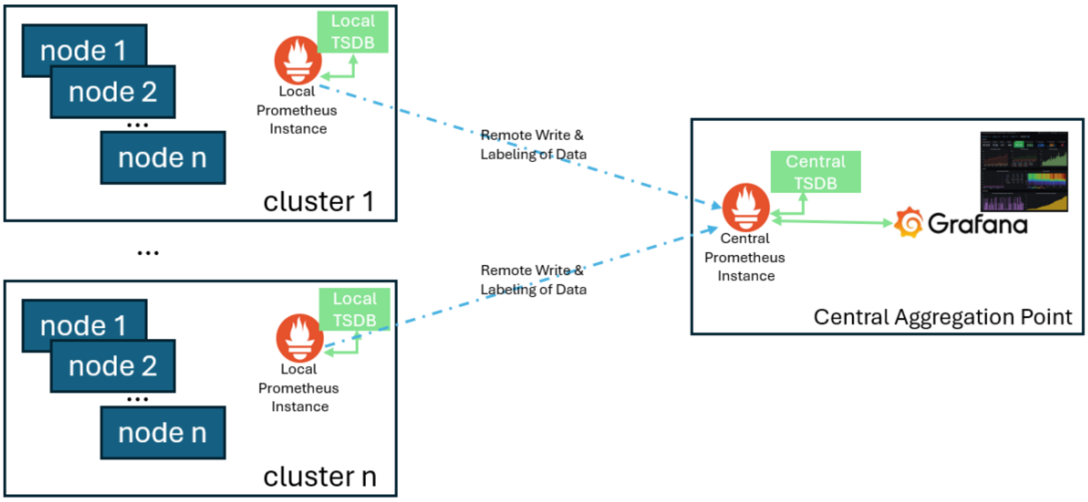

Figure 3: Prometheus instantiation and remote write function to central aggregation unit

2.2.2 Measurement Targets

Prometheus is used for scraping metrics from various targets inside each cluster. In modern cloud-native environments, monitoring and collecting metrics is crucial for maintaining performance, reliability, and overall health. Tools such as cAdvisor, Node Exporter, and kube-state-metrics complement the Prometheus monitoring solutions, and can provide detailed insights into system performance, resource usage, and cluster state. In the existing monitoring solutions, these tools have been integrated within the clusters, in order to provide new targets for metrics for Prometheus. Prometheus acts as a local aggregator, with a common format for data that comes from all these different targets. Below we provide information for each of these targets.

cAdvisor (Container Advisor) [9] is an open-source container resource usage and performance analysis agent. It provides information on the resource usage and performance characteristics of running containers. It collects real-time metrics about CPU, memory, network, and disk usage for containers, and integrates easily with Prometheus. It supports the automatic discovery of containers running on a host and as long as the metrics are exposed in the default API, allows for the automatic collection of metrics without additional configuration.

Node Exporter [10] is a Prometheus exporter for hardware and OS metrics exposed by Unix kernels. It is designed to monitor system-level metrics for individual nodes that host different workloads. It enables the collection of a wide range of metrics, including CPU, memory, disk I/O, and network statistics, while integrates off-the-shelf with Prometheus.

[9] cAdvisor: https://github.com/google/cadvisor [Last accessed: 12 September 2024]

[10] NodeExporter: https://github.com/prometheus/node_exporter [Last accessed: 12 September 2024]

<u>kube-state-metrics</u>

kube-state-metrics [11] is a Kubernetes-focused metrics exporter that provides detailed insights into the state of Kubernetes objects. It exposes metrics about the state of K8s objects such as pods, nodes, deployments, and services. It continuously monitors the Kubernetes API server for updates and reflects changes in real-time metrics, while integrating seamlessly with Prometheus for collection and querying of metrics.

```
ubuntu@cp-bp:~$ sudo kubectl get pods
NAME                                     READY   STATUS    RESTARTS       AGE
cadvisor-78644c6787-s9mps                1/1     Running   2 (5m31s ago)  5d20h
kube-state-metrics-5fbcc79dff-hl4j5      1/1     Running   0              5d20h
node-exporter-9jlq7                      1/1     Running   1 (6m16s ago)  10d
node-exporter-r6gtx                      1/1     Running   2 (6m43s ago)  10d
node-exporter-vfn9s                      1/1     Running   1 (6m18s ago)  10d
prometheus-operator-76469b7f8c-4gxp2     1/1     Running   0              5d20h
prometheus-prometheus-0                  2/2     Running   0              5m34s
prometheus-prometheus-1                  2/2     Running   0              5m30s
promtail-daemonset-fq4wx                 1/1     Running   1 (6m16s ago)  10d
promtail-daemonset-sg27w                 1/1     Running   1 (6m18s ago)  10d
```

Figure 4. Instantiation of monitoring solutions within the clusters

2.2.3 Log Collection

Although metrics regarding the experimentation environment are highly desired for virtual access activities, log collection is also of high importance for ensuring that the experiment has been executed in a correct manner. To this aim, we employ two different tools at two different levels: 1) the Grafana Loki[12] for centralized log aggregation, and further depiction by Grafana, and 2) the promtail[13] log scraping and labelling service, for collecting logs from each cluster and sending them to the centralized Loki instance.

Grafana loki

Loki is a scalable, highly available log aggregation system inspired by Prometheus. It is designed to be cost-effective and easy to operate, focusing on providing efficient log indexing and querying capabilities. It organizes logs into streams with labels, similar to Prometheus' time series model. It uses the same label model and therefore is able to integrate seamlessly with Grafana, allowing users to correlate logs with metrics. It is designed to minimize the cost of storing logs by not indexing the full log content. It consists of three main components:

- Distributor: Receives logs and splits them into smaller chunks.
- Ingester: Writes log chunks to storage and handles queries for recently ingested data.
- Querier: Processes read requests, fetching data from long-term storage and returning results to the user.

Logs can be stored in chunks in a long-term storage backend (like AWS S3 or Google Cloud Storage), with metadata and labels used to index and query the logs efficiently.

---

[11] Kube-state-metrics: https://github.com/kubernetes/kube-state-metrics [Last accessed: 12 September 2024]

[12] E. Bautista, N. Sukhija and S. Deng, "Shasta Log Aggregation, Monitoring and Alerting in HPC Environments with Grafana Loki and ServiceNow," 2022 IEEE International Conference on Cluster Computing (CLUSTER), Heidelberg, Germany, 2022, pp. 602-610, doi: 10.1109/CLUSTER51413.2022.00079.

[13] Grafana promtail: https://hub.docker.com/r/grafana/promtail [Last accessed: 12 September 2024]

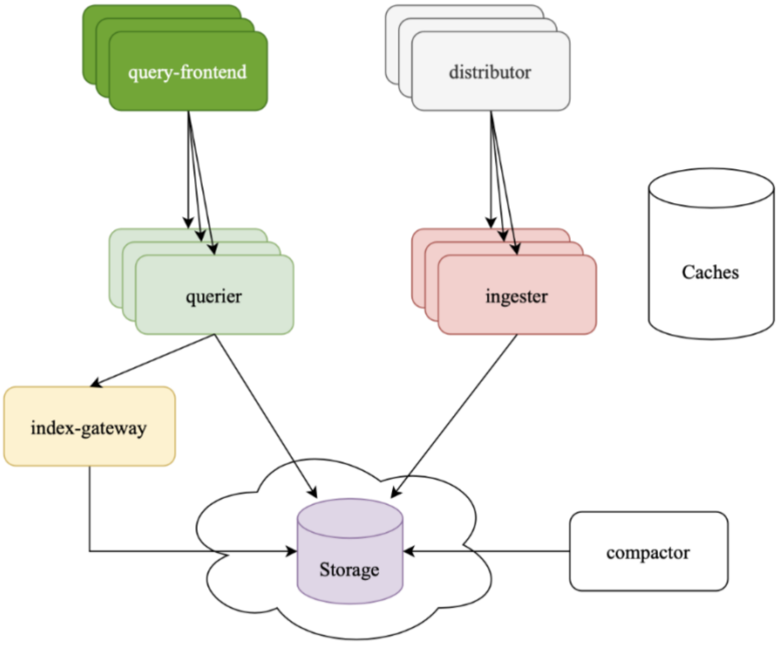

Figure 5. Grafana Loki architecture

Promtail is an agent that is able to collect and transmit the contents of local logs to a Loki instance. It is designed to gather logs from various sources, process them, and send them to Loki for storage and analysis. Logs that promtail collects are comprised from various sources within the cloud-native orchestration environment, including local files, system journal, and K8s pods. The logs are enriched with metadata from their sources, such as labels from Kubernetes pods, which helps in efficient querying and correlation. Further labeling on a per-cluster basis is also performed within SLICES-SC in order to be able to distinguish centrally the data that come from different clusters. Promtail is typically deployed as a daemon on each node in a Kubernetes cluster or as a standalone agent on other systems.

2.2.4 Integration for VA

For the purposes of VA, the tools have been integrated within the post-5G blueprint for SLICES-RI[14]. The tools are configured to be deployed within the cluster, and point to a specific instance that holds the central collection for all the different metrics (Prometheus, Loki and Grafana for visualization). Snapshots of the final metrics that the experimenters get access to are provided below.

[14] Post-5G Blueprint for SLICES-RI (monitoring branch): https://github.com/dsaucez/SLICES/tree/monitoring [Last accessed: 12 September 2024]

Figure 6. Grafana Depiction of Prometheus collected stats

Figure 7. Grafana depiction of node-exporter and cAdvisor generated metrics

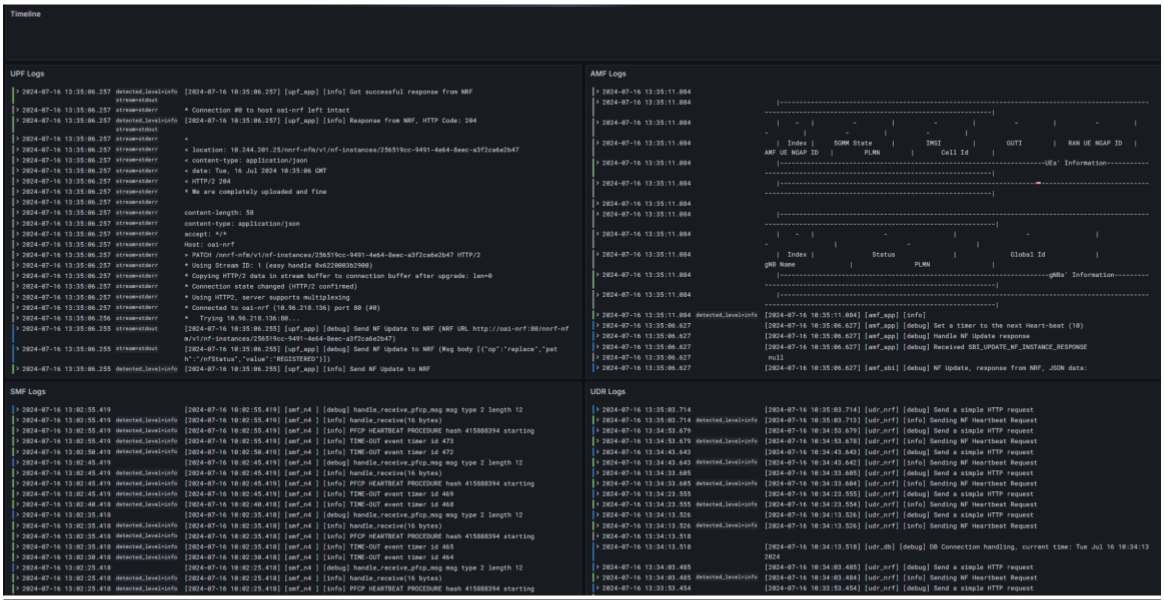

Figure 8. Grafana depiction of logs collected through promtail and Loki on a single experimental instance (logs come from the OAI-CN)

2.3. Reproducibility framework

TUM continued the work on its virtual testbed that was described in D8.4 - Report of the first period of virtual access[15].

The previously described framework for creating virtualized testbeds was extended and adapted to be used as a platform for education. We used the virtual testbed for a practical project that is offered together with the lecture "Advanced Computer Networking" in the winter semester at TUM[16]. This virtual platform is designed to teach students essential concepts, tools, and technologies used in computer networks in domains like cloud computing or cellular networks. This includes Linux command line tools, e.g., SSH or git. The testbed is controlled via the SLICES pos framework, to also teach the students reproducible experiment workflows. Each student implements a software router using different experiment nodes provided by the testbed.

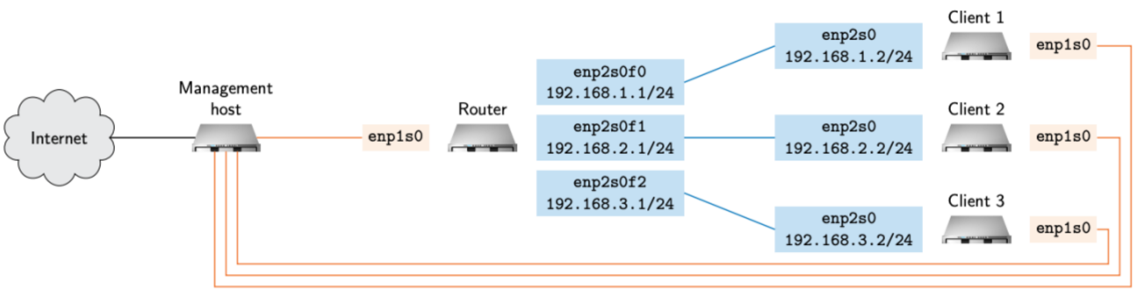

Figure 9. Testbed topology consisting of a testbed management node and four experiment nodes.

[15] D8.4 – Report of the first period of virtual access, https://slices-sc.eu/wp-content/uploads/2023/03/SLICES-SC_D8.4_approval_disclaimer.pdf [Last accessed: 12 September 2024]

[16] Lecture "Advanced Computer Networking", https://net.in.tum.de/teaching/ws2324/acn.html [Last accessed: 12 September 2024]

The topology for a single student is shown in the previous figure. Each student gets a topology consisting of four experiment nodes (a router node and three client nodes). To minimize the hardware requirements for hosting such a platform, we virtualized all experiment nodes. We currently use a server with 2x Intel Xeon Gold 6130 (16 physical cores @ 2.1 GHz) and 384 GB of RAM. The current platform allows hosting 12 independent setups for 12 students. We plan to switch to a dynamic allocation scheme to provide a platform for more than 12 students for the lecture in the next semester. The virtualized testbed relies on a hardware-accelerated network, using 2x Intel E810-XXVDA2 NICs (2 ports, 25 Gbit/s). The hardware acceleration allows us to provide students access to state-of-the-art hardware at reasonable costs.

qosium

On top of the common tools offered in SLICES-SC for the VA users University of Oulu offers Kaitotek Qosium to be used to view the data by the VA users. Qosium is a SW based QoS measurement system that measures and monitors the network quality and performance. Measurements done using Qosium are stored in csv format. University of Oulu license for the Qosium enables to offer a possibility to view the datasets produced in the University of Oulu 5G Test Network using Qosium Scope that resides at the 5GTN server. For interested parties access to that server is offered to view the produced datasets. Qosium license does not allow externals do perform measurements in 5GTN using the Qosium SW unless they have their own license.

2.4. SLICES Virtual Access Metadata

Future SLICES-RI will be a platform designed to provide an all-encompassing experiment experience for researchers, enabling seamless access to specialized laboratory equipment, facilitating collaboration and interdisciplinary research, and fostering and accelerating innovation. One of the key enablers for SLICES-RI to accomplish its vision is the design of appropriate data sharing and integration mechanisms to support complex experiments that orchestrate heterogeneous geo-diverse resources. In addition, data managed within SLICES-RI need to adhere to the FAIR data principles enhancing the discover, accessibility, interoperability, and reusability of research data, thus promoting a more efficient and collaborative scientific ecosystem.

SLICES community within the SLICES-SC project, and targeting the EOSC catalogue/marketplace, started proposing a metadata profile for EOSC. The virtual access metadata follows this already defined profile based in Dublin core metadata schema.[17]During the first period of the project, we decided to start with a practical solution by offering raw data via the CKAN server.

During the second period and in parallel with this initial practical approach, SLICES decided to contribute with a new metadata model, SLICES Fair Digital Object (SFDO), as a unified framework for SLICES users and ecosystem of platforms and systems with the ambition of find, access and use any digital object (data, services, tools and software) in SLICES-RI. To achieve this, SLICES defines a metadata profile scheme, where each digital object is first described using a select set of common metadata attributes and then according to its type, the description is extended with a set of type-specific attributes. Each managed digital object, coined SLICES FAIR Digital Object (SFDO), provides a list of predefined options to inform metadata consumers of what kind of object it describes. The metadata are structured in a hierarchical format, with Level 1 containing attributes that are common to all SFDOs. Level 2 contains the metadata specific to the type of the Digital Object. Some types also

[17] Dublin Core, DCMI Schemas, https://www.dublincore.org/schemas/ [Last accessed: 12 September 2024]

support recording additional domain-specific information, based on the object's subtype, which is recorded in level 3.

The Metadata Registry System (MRS) provides access and management services to SFDOs using three components: (i) *Repository***:** The metadata themselves are persisted in an open-source database. The current implementation employs PostgreSQL due to its extensive feature set, reliability track record and widespread familiarity; (ii) *Backend***:** It exposes the metadata to the rest of the world in form of a REST API and facilitates aspects such as access control, reporting, etc. A core feature of the backend is that it acts as a translation layer, allowing backwards compatibility when a new version of metadata models is released. Finally, it handles background maintenance tasks, such as European Open Science Cloud (EOSC) import/export. The backend is implemented using ASP.NET Core. While it is designed to be as independent as possible, the authentication is delegated to SLICES-global Authentication and Authorization Infrastructure (AAI) to ensure uniform user access across the entire infrastructure; and finally, a (iii) *Web portal*, accessible to all registered users, allows user-friendly access to the MRS functionality. Users can search for objects, manage objects (if they have permissions) and access reporting facilities, such as dashboards. The web portal is accessible to all registered users. Currently, the first version of MRS was released as part of the new blueprint version, led by UCLAN within the SLICES-PP project and with contributions from SLICES-SC partners.

3. Lessons learned on VA services from SLICES Test-beds

The initial seed of SLICES VA services started within the geographically dispersed legacy test-beds. There is a common effort on building an integrated VA services for the coming pre-operation phase and the future next SLICES lifecycle phases. We summarize in this section the rich activity within the more active test-beds to illustrate the future path on building the SLICES VA services.

3.1. SILECS-FIT / OneLab, France

SLICES-FIT testbed has been seamlessly integrated into the SLICES-SC portal through the use of the GENI AM API. This integration significantly enhances the platform's capabilities by offering users the ability to install, customize, and experiment with both new and pre-existing cloud software stacks. By providing such flexibility, SLICES-FIT enables researchers and developers to modify cloud environments according to their specific requirements and test their configurations in real-world scenarios. This makes it an ideal facility for experimentation and innovation in cloud computing. Whether users are developing entirely new cloud infrastructures or adapting existing solutions, the platform provides the necessary tools and resources for comprehensive testing and validation. The combination of SLICES-FIT's powerful infrastructure with different features of the SLICES-SC portal offers a robust environment for cloud development, fostering an ecosystem that supports both academic research and practical applications in the field of cloud technologies. SLICES-FIT will be a candidate for providing open research data to the SLICES Metadata Research Infrastructure. We actively contributed to the definition and articulation of its various components but were not yet in a situation to feed data to SLICES. This will happen in the near future.

3.2. NITOS – UTH, Greece

NITOS implements all the aforementioned tools (see Sect. 2) with respect to the visualization and metric collection from different clusters. A central Grafana, Loki and Prometheus instance is instantiated to allow external users get access to the collected metrics and results and depict them in an easy manner. The datasets collected and offered for depiction regard the experiments that are instantiated within the NITOS testbed, as well as datasets that are offered for external use and generated in the past within NITOS. The second type of datasets has been uploaded to the SLICES-SC CKAN server as well, and is tagged with the appropriate metadata. The datasets offered beyond the current experiments in NITOS are the following:

- AI-driven Application-Aware 5G Network on K8s: This dataset collected through NITOS testbed includes real-world time-series statistics from network traffic on real commercial LTE networks in University of Thessaly (UTH).
- UE statistics Timeseries (CQI) in LTE networks: This dataset includes real-world Channel Quality Indicator (CQI) values from UEs connected to real commercial LTE networks in Greece.
- Data from agricultural installations: Dataset consisting of metrics from sensors (data for over 3 years), regarding ambient weather conditions (temperature, humidity, light luminosity, wind direction and speed), soil conditions (temperature, humidity), and calculated others (e.g. the evapotranspiration of plants)
- Energy consumption data from an office environment: Dataset from a UTH framework for monitoring smart-homes in terms of power consumption, towards concluding on proper and efficient power usage. Depicts energy consumption (in a per-device monitoring case for several sensors) measurements of an office space in the UTH premises, that contains measurements from approx. 2 years.

3.3. 5G Test Network (5GTN), Finland

Instead of virtual access, the University of Oulu recorded lessons learnt from the implementation of the federation of the 5G Test Network that has been done during 2023 and 2024. Federation architecture for the University of Oulu 5G Test Network (5GTN) is presented in the Figure 16.

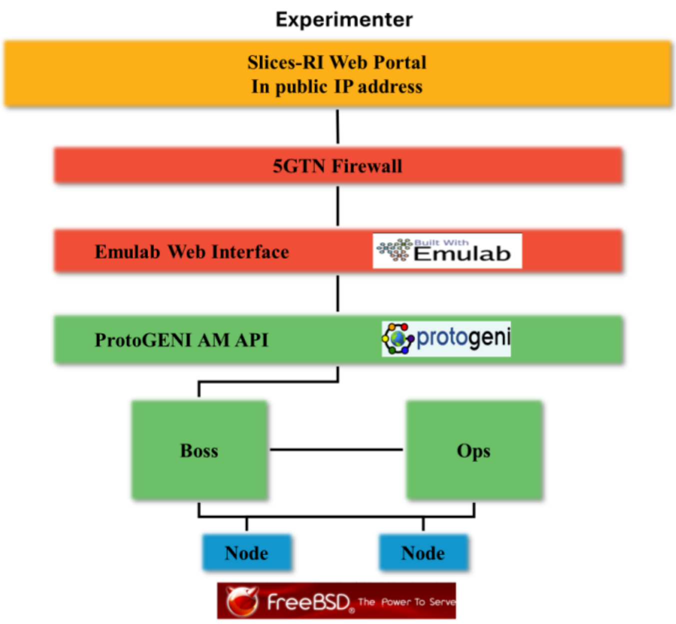

Figure 16. Federation architecture for the University of Oulu

In the architecture, the Web Portal provides user account creation, user authentication and access control. It offers the user interface for the experimenter to interact with testbed setup. It enables the possibility to do the Experiment management and customization. It also helps monitoring the status of the experiment. Emulab software is used to manage testbed resources and overall operations of the testbed. ProtoGENI Aggregate Manager API (AM API) facilitates the federation process. It is implemented on top of Emulab. Finally, the Boss node handles all management tasks like authenticating users, resource allocation and monitoring and Ops node, as a complimentary node to boss node, is hosting file server and deals with external communication.
Following is a list of items University of Oulu learned from the installation process of the SLICES-SC federation:

- Emulab is already quite old technology. Issues appeared it is already so old.
- The benefit for having Emulab is the possibility for reproducibility.
- Support for Emulab is available from Google EmuLab admin group.
- It is a good idea to experiment with the environment to learn outside the instructions and manuals.
- Virtualized environment helps in infrastructure development.
- When issues occur, documentation and manuals do not help much as they describe one way of implementing the system. As Emulab is outdated, on-line help is not available. Peer help from an organization that has already implemented the Emulab is paramount. For us help from imec has been crucial to progress with integration of the system. They have a lot of experience from Fed4FIRE and AM API and the knowledge that they have shared with us has helped tremendously in the integration process.
- Structure for operating this kind of approach is heavy: Base station + compute nodes + data center. This requires special knowledge about the SW and HW components and the overall system. When planning to implement this kind of system, consider that it will be a large operation.
- AM API is good enough for this kind of federation implementation.

3.4. ILabT – IMEC, Belgium

imec has made available multiple testbeds for Transnational and Virtual Access (https://doc.ilabt.imec.be). These testbeds can be used to create datasets of experiments for Virtual Access or to reproduce experiments based on Virtual Access data.

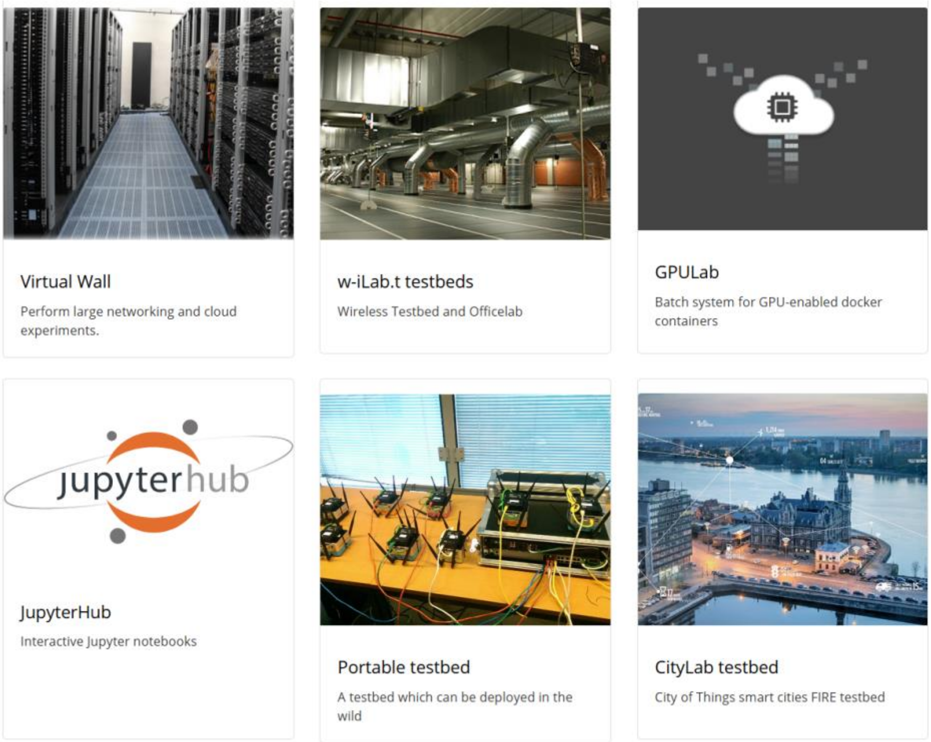

Figure 10. imec testbed portfolio available through SLICES-SC

- 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.

This is the list of datasets on https://ckan.iotlab.com/organization/slices-sc that were produced by SLICES-SC open call experiments using imec testbeds.

NVIDIA V100 matrix multiply benchmark results

NVIDIA V100 GPU benchmark results for various matrix-matrix multiplication implementations.

XLS

NVIDIA V100 memory latency benchmark results

V100 GPU memory hierarchy latency test (register-L1/L2 cache-global memory) benchmark results.

XLS

NVIDIA V100 GPU CUDA kernel data load/store performance benchmark

Results of CUDA kernel data load performance (NVIDIA V100 GPU card)

XLS

NVIDIA V100 GPU CUDA kernel FMA operation performance benchmark

NVIDIA V100 GPU CUDA kernel performance for Fused-Multiply-Add (FMA) operations.

XLS

GPULab HGX-2 File I/O benchmark results

Filesystem performance of the IMEC GPULab HGX-2 system for transferring data from storage to host memory.

XLS

NVIDIA V100 Fast Fourier Transform benchmark results

Single precision FFT execution benchmark results in function of FFT size and batch size.

XLS

CUDA kernel global memory data transfer bandwidth benchmark

Global memory bandwidth measured from a compute kernel on an NVIDIA HGX-2 GPU system using V100 cards.

XLS

Performance evaluation of Horovod distributed deep learning cluster on the...

Performance evaluation based on the ImageNet dataset, Resnet50 models, and the imec Virtual Wall 2 and GPULab environments in various configurations. The dataset contains the...

CSV

The CAP Babel Machine

We present an open-source and freely available natural language processing system designed for comparative policy studies. Manually labeling large corpora can be tedious and...

ZIP

This gives some details of the CAP Babel Machine output (https://ckan.iotlab.com/dataset/cap-babel):

The CAP Babel Machine

We present an open-source and freely available natural language processing system designed for comparative policy studies. Manually labeling large corpora can be tedious and often demands extensive domain expertise. Recent advancements in machine learning and natural language processing hold the potential for language models to surpass human-level accuracies in text classification tasks. In our experiments, we fine-tuned Large Language Models (LLMs) across various language and domain corpora using 21 categories from the Comparative Agendas Project codebook. Leveraging multilingual XLM-RoBERTa models, our pipeline generates state-of-the-art outputs for selected language-domain pairs and domains, such as media or parliamentary speech. We fine-tuned a total of 41 models, with the first being a pooled model trained on the entire (multilingual) dataset, followed by 9 models using language-specific datasets, and 10 using domain-specific datasets, while the remaining models were fine-tuned on language-domain datasets. Out of these, 24 models achieved a weighted macro F1 above 0.75, with 6 reaching 0.90.

The inference platform utilizing these models is freely available for researchers at https://capbabel.poltextlab.com. The dataset includes models, result metrics and the data used for fine-tuning.

Data and Resources

ZIP **Performance metrics** — Explore
Includes fine-grained metrics (precision, recall, F1) for each model and...

DATA **Models** — Explore
Fine-tuned models available as HuggingFace repositories.

DATA **Data** — Explore
Training and test data used for model fine-tuning and evaluation.

3.5. HUN-REN Cloud – HUN-REN SZTAKI

As part of developing the HUN-REN Cloud SLICES node to provide Virtual Access, we improved the RI's monitoring system. We also continued developing reference architectures to lower the barrier of entry for our users to use the RI resources. We started replacing our old Zabbix-based monitoring system with a Prometheus and Grafana-based solution. With this new monitoring system, we have better insight into the usage of the whole RI and can measure the power consumption. This data can help us optimize and improve the RI's physical resource usage. This new monitoring system also provides more useful information about our user's resource utilization during eventual Virtual Access, and we can help them fine-tune their resource usage if necessary.

We also continued developing our reference architectures (RA). These are quickly and easily deployable digital research infrastructures that cover a variety of common use cases. They follow the infrastructure as Code methodology and are based on widespread tools. With these, our users can quickly deploy complex software infrastructures on the Cloud. We continued the development of some significant RAs to support distributed deep learning, workload management, or container platforms. Based on our user experience, these can be helpful during Virtual Access to utilize our RI more easily and efficiently.

3.6. TUM lab, Germany

Beside the development on the virtual testbed, we also continued to provide experimental data created in our testbed. A notable example for this is our paper[18] published at the CoNEXT'23

[18] Markus Sosnowski, Florian Wiedner, Eric Hauser, Lion Steger, Dimitrios Schoinianakis, Sebastian Gallenmüller, Georg Carle, "The Performance of Post-Quantum TLS 1.3," in Proc. International Conference on emerging Networking EXperiments and Technologies (CoNEXT), Paris, France, Dec. 2023.

conference. The provided data can be found on our website[19]. The long-term access to the research data is provided by mediatum[20], a service by our local library. The mediatum platform was chosen due to its support for large files (our artifacts consume 161 GB in total) and the low costs for TUM researchers. The paper contains performance measurements for post-quantum TLS, the provided data set contains our measurement and evaluation scripts in addition to the traffic traces that was recorded during the experiments.

A second notable example is the development of a portable experiment workflow. This paper[21] demonstrates that the pos experiment workflow can be executed on two different US-based testbeds (Chameleon and CloudLab) without modifications to the experiment scripts. We provide the code examples for these experiments via GitHub repositories (https://github.com/hstubbe/wons2024).

Further examples for data sets that were created since the last deliverable are:

- Wiedner et al.[22],[23]
- Helm et al.[24], [25]
- Wiedner et al.[26],[27]

3.7. OTE, Greece

OTE have uploaded four new datasets in the CKAN repository that contain information about the total power consumption of two houses for one year and data collected from 8 environmental sensors deployed in a single room of two additional houses for a one-month period. The datasets are described below in more detail:

- https://ckan.iotlab.com/dataset/leonr-do-iot-lab-total-power-consumption-of-a-house-id-2-for-1-year-period

  This JSON file contains information about the total power consumption of a house (id-2) for one year. The data is stored in a series of objects, with each object representing a measurement of the power consumption at a specific time. The columns field is an array of strings that represent the names of the columns in the values array. The values field is an array of arrays, with each inner array representing a single measurement of power consumption. The first element of each inner array is the timestamp, and the second element is the mean total power consumption of a 15-minute time interval. The power consumption is measured in watts.

---

[19] https://tumi8.github.io/pqs-tls-measurements/ [Last accessed: 12 September 2024]

[20] https://mediatum.ub.tum.de/1725057 [Last accessed: 12 September 2024]

[21] 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.

[22] Florian Wiedner, Alexander Daichendt, Jonas Andre, Georg Carle, "Control Groups Added Latency in NFVs: An Update Needed?," in 2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Nov. 2023.

[23] https://wiednerf.github.io/cgroups-nfv/ [Last accessed: 12 September 2024]

[24] Max Helm, Georg Carle, "Synthesizing and Scaling WAN Topologies using Permutation-invariant Graph Generative Models," in 19th International Conference on Network and Service Management (CNSM 2023), Niagara Falls, Canada, Oct. 2023, p. 6.

[25] https://github.com/tgnn-test/dataset-graphscaling [Last accessed: 12 September 2024]

[26] Florian Wiedner, Max Helm, Alexander Daichendt, Jonas Andre, Georg Carle, "Containing Low Tail-Latencies in Packet Processing Using Lightweight Virtualization," in 2023 35rd International Teletraffic Congress (ITC-35), Oct. 2023.

[27] https://github.com/tumi8/quic-10g-paper [Last accessed: 12 September 2024]

- https://ckan.iotlab.com/dataset/leonr-do-iot-lab-total-power-consumption-of-a-house-id-1-for-1-year-period

  This JSON file contains information about the total power consumption of a house (id-1) for one year. The data is stored in a series of objects, with each object representing a measurement of the power consumption at a specific time. The columns field is an array of strings that represent the names of the columns in the values array. The values field is an array of arrays, with each inner array representing a single measurement of power consumption. The first element of each inner array is the timestamp, and the second element is the mean total power consumption of a 15-minute time interval. The power consumption is measured in watts.

- https://ckan.iotlab.com/dataset/leonr-d-iot-lab-measurements-from-8-different-environmental-sensors-of-room-id-105-for-1-month

  This JSON file contains data collected from 8 environmental sensors deployed in a single room (id-105) for one-month period. The data is structured as a set of objects, each representing a measurement taken by the 8 sensors at a specific time. Each object has two fields: "columns" and "values". The "columns" field is an array of strings that represent the names of the columns in the "values" array. The "values" field is an array of arrays, with each inner array representing a single measurement taken from each of the 8 sensors at a specific timestamp. Each inner array contains the mean value of each measurement taken over a 15-minute time interval. The first element of the inner array is the timestamp at which the measurement was taken.

- https://ckan.iotlab.com/dataset/leonr-d-iot-lab-measurements-from-8-different-environmental-sensors-of-room-id-106-for-1-month

  This JSON file contains data collected from 8 environmental sensors deployed in a single room (id-106) for one-month period. The data is structured as a set of objects, each representing a measurement taken by the 8 sensors at a specific time. Each object has two fields: "columns" and "values". The "columns" field is an array of strings that represent the names of the columns in the "values" array. The "values" field is an array of arrays, with each inner array representing a single measurement taken from each of the 8 sensors at a specific timestamp. Each inner array contains the mean value of each measurement taken over a 15-minute time interval. The first element of the inner array is the timestamp at which the measurement was taken.

4. Lesson Learned on VA services from Open Calls Experiments

Throughout the project’s lifetime, SLICES-SC announced 5 Open Calls. So far applicants requested only Transnational Access to the SLICES-RI. However, thanks to these experiments, we will manage to populate the CKAN server with numerous data sets that derive from actual scientific experiments thus they represent a valuable asset in our infrastructure. Details about this data expansion are described in Section 2.1.

Moving into pre-operation SLICES-RI will create more visibility for this kind of access so that potential users are well aware of this kind of usage. Access to the datasets should be provided in a smooth, user-friendly manner.

5. Virtual Access Usability Test

The deliverable was supposed to include an assessment report from the User Committee assessing Virtual Access services and statistics on data submissions, data access and analysis. Since users are still in the process of populating the repositories with their data, this assessment would not have been meaningful. Nevertheless, we wanted to objectively evaluate our existing services to identify room for improvement. Thus, we asked a junior researcher, who was not previously involved in the SLICES activities to conduct usability testing in order to determine whether the access to the different datasets can be implemented swimmingly, or if there are any bottlenecks or roadblocks requiring fixing.

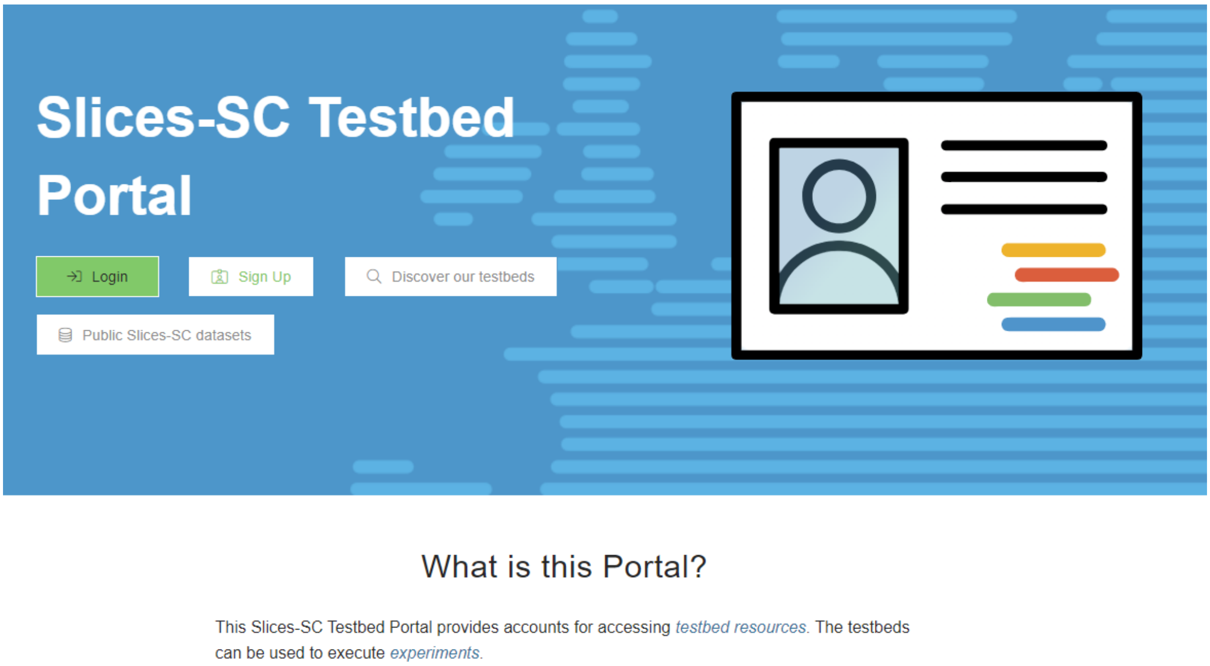

Figure 11. SLICES-SC Testbed Portal

The test participant started by attempting to access the data by clicking on the button that says 'Public Slices-SC datasets on the SLICES Portal. It shortly became clear that it was not possible to access the data without creating an account. Once logged in, the problem still occurred, thus the user needed to contact the operators of the CKAN infrastructure and ask for their advice on how to proceed.

Although the communication with the system operators was smooth and prompt, it is not visible on the website where the users can get help if they come across any problems. Turns out that the problem could be regenerated if the user requested access to internal resources. Apparently, the issue is caused by the browser, thus it could be easily fixed by allowing HTTP download.

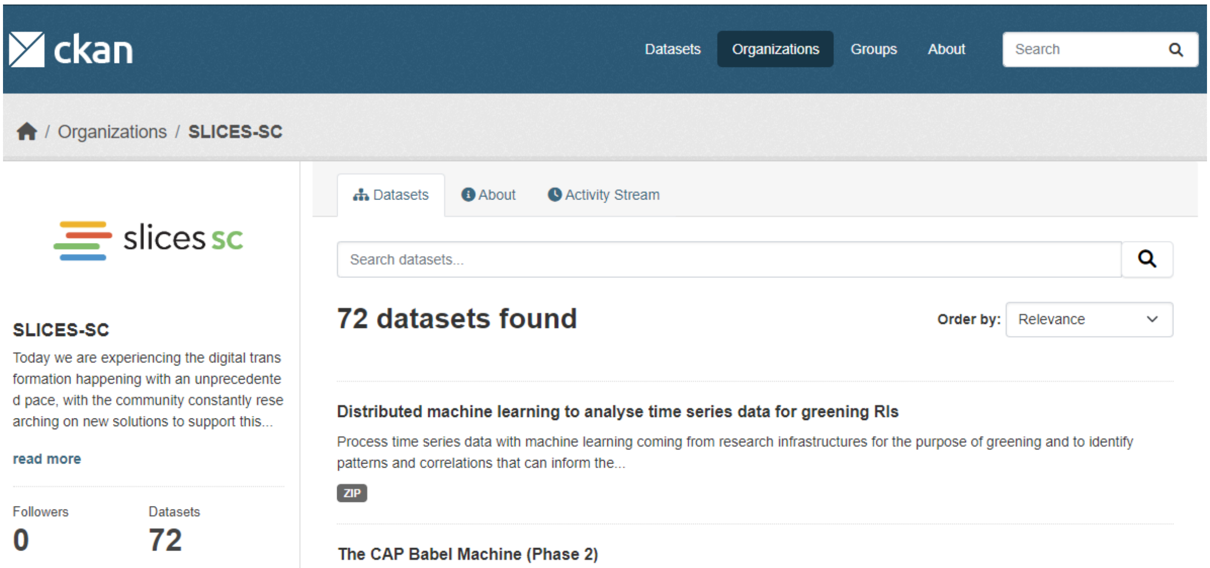

Figure 12. SLICES CKAN server

Having a closer look at the currently available datasets, it is safe to state that they cover a wide range of research topics. Here is a compilation of the uploaded datasets:

AI and machine learning	• deep learning, GPUs, and Machine Learning, • training and fine-tuning large language models (LLMs), • GPU performance analysis and optimization, • classification tasks.
Natural language processing (NLP)	• natural language processing, • text classification, sentiment analysis, or language modelling.
Energy and power	• energy consumption analysis, • power grid optimization, • renewable energy research.
Computer networking	• network statistics, 5G, vehicles, Wi-Fi, and IEEE 802.11n., • network performance analysis, • 5G technology research, • vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communication studies, • Wi-Fi protocol improvements
Internet of Things (IoT)	• smart home technologies, or industrial IoT applications, • connected devices.

At the moments no policy is introduced that would regulate how the data should be uploaded and stored, thus the datasets are heterogeneous, they use various file formats, including:

- structured data: CSV, JSON, XML, TSV, SQL, SQLite
- document formats: PDF, TXT
- spreadsheets: XLSX
- executables, binaries, codes, shell scripts: .PY, .SH

For certain projects, a comprehensive library is included with the dataset, offering solutions to assist with pre-processing, testing, training, validation, and visualisation stages. In approximately 60-70% of cases, these projects feature well-structured and well-documented datasets, making the experiment reproducible, provided the necessary hardware resources are available.

The datasets are accessible the following ways:

- download link
- browser interface
- redirection to the Hugging Face site
- redirection to a shared Google Drive folder
- redirection to a shared Dropbox folder
- redirection to a shared GitHub repository

The next obstacle the user encountered was an internal server error caused by some datasets. Some datasets have accessibility issues due to internal server error results when aiming for download. This issue is currently being addressed at the time of the deliverable.

Descriptions, transparency and ease of use:

- most of the datasets contain well-structured descriptions
- depending on the file format, many datasets include headers inside the files
- the presence of metadata varies across datasets

Conclusion, recommendations

This collection of datasets offers resources for researchers across various domains of computer science and engineering. The diversity in topics, file formats, and accessibility methods presents both opportunities and challenges. To minimise the issues, a user could overcome and to ensure a more user-friendly experience the testing participant proposes the following recommendations.

- Provide a step-by-step guide on the SLICES portal that clearly showcases how to create your account.
- Indicate user support contact information and display it in a visible place on the website.
- Introduce a data curation policy that defines how datasets should be uploaded to the repositories so that they are easily reusable and searchable.
- Provide a search platform that helps users navigate through the different datasets.

6. Next steps in SLICES VA services

SLICES-SC project has been covering a period previous to the SLICES pre-operation phase that will start in Q4 2024 within SLICES-PP project. The experience of defining and starting SLICES VA services during the lifetime of SLICES-SC project has been, on the one hand, very limited due the fragmentation of the legacy testbeds and the data sets produced and the lack of a standardized framework that unify all the resources used to populate the VA services. However, it has also been very fruitful in terms of identifying the needs and key challenges to be faced during the construction of SLICES-RI and the provisioning of the VA services. One of the main lessons learned is the lack of maturity on sharing research data within SLICES research community. We, as research community in the scientific domain of SLICES, have identified the gap between the experimenter's needs and the resources offered to run VA services. SLICES is working currently in filling this gap and change the research methodology to provide real VA services to the SLICES community.

SLICES has identified as crucial the development of a common framework that will contribute to the standardization of metadata model that will allow SLICES users and its ecosystem of platform and systems to uniformly find, access and use digital objects, such data, services, tools and software. SLICES has already started the design of SFDO (SLICES Fair Digital Object) as a common data management framework to support the deployment of more sophisticated SLICES VA services and harmonized and integrated uniform data across distributed facilities to ensure consistency and interoperability in the effective deployment and operation of SLICES-RI.

Another challenge identified is the SLICES VA services usage tracking difficulties. Tracking the usage of digital objects presents significant challenges, complicating the assessment of VA services and utility. There is a clear need on the adoption of Persistent Identifiers (PIDs) at all levels, utilizing analytics tools and software that seamlessly integrate with databases and platforms and ensuring data providers receive due credits and feedback of their contributions that definitely will increase the incentives on participating by the researchers' community.

Regulatory and cybersecurity constrains at National and European regulations on personal and sensitive data sharing are a key obstacle to cross-border data access, requiring careful deployment during the real operation phase of VA services in SLICES-RI.

7. Conclusion

This document reports the activities on defining and starting the provisioning of SLICES VA services during the second reporting period of SLICES-SC project. The experience of seeding and populating repositories like the SLICES CKAN and GitLab servers led us to identify the weaknesses and limitations of this initial approach and to challenge the next steps to build a more standard and consolidated framework that contribute to build the future SLICES VA services ready for the deployment and operation phases of SLICES-RI. This is now well in progress.

slicessc