


## EBRAINS2.0

### D5.2 Open Metrics Framework



Figure 1: Open Metrics Framework Dashboard


 Co-funded by the European Union
 
 EBRAINS 2.0 has received funding from the European Union's Research and Innovation Program Horizon Europe under Grant Agreement No. 101147319.

Project Number:	101147319	Project Name:	EBRAINS 2.0
Deliverable No. & Name:	D5.2 Open Metrics Framework		
Deliverable Description:	Software monitoring and conveying end-to-end performance, effort, and satisfaction for the RI		
Deliverable Type:	OTHER		
Dissemination Level:	PU =Public		
Planned Delivery Date:	Project Month M16 / 30.04.2025		
Actual Delivery Date:	Project Month M16 / 24.04.2025		
Author(s):	Vasillios GEORGOPOULOS, ATHENA (P4)		
Compiled by:	Vasillios GEORGOPOULOS, ATHENA (P4), Amaryllis RAOUZAIYOU, ATHENA (P4)		
Contributor(s):	Orfeas AIDONOPOULOS, ATHENA (P4)		
WP and / or Project QC Review:	Evangelia DRYMALITI, ATHENA (P4)		
WP Leadership Sign-off:	Yannis IOANNIDIS, ATHENA (P4)		
Keywords:	Open Metrics Framework, OMF, monitoring, KPI, dashboard, visitors, users, performance metrics		
Abstract:	D5.2 outlines the progress achieved in the development of the Open Metrics Framework (OMF) for EBRAINS 2.0 since the previous deliverable (D5.1). The document provides an overview of key advancements, highlights the challenges encountered, and outlines the roadmap for future developments. The OMF remains a vital component for monitoring, analysing, and reporting performance metrics, ensuring the continued stability and usability of the EBRAINS platform.		

## Table of Contents

<b>1. List of Abbreviations and Acronyms used .....</b>	<b>5</b>
<b>2. Introduction.....</b>	<b>6</b>
2.1 Document structure .....	6
2.2 How to read this deliverable.....	6
<b>3. Overview.....</b>	<b>7</b>
<b>4. Implementation Plan .....</b>	<b>8</b>
<b>5. Implementation Progress M1 - M16 .....</b>	<b>9</b>
5.1 Implementation Progress M1 - M12.....	9
5.1.1 Dashboard for unique visits and visitors .....	9
5.1.2 Dashboard for unique active EBRAINS users .....	10
5.1.3 Dashboard for GitLab issues .....	11
5.1.4 Dashboard for Zammad issues .....	12
5.1.5 Dashboard for CWL workflows executions .....	14
5.1.6 Core Services Operational Monitoring .....	14
5.2 Implementation Progress M13 - M16.....	15
5.2.1 Dashboard for OS Metrics of Services in JSC Rancher (Workload Level) .....	16
5.2.2 Dashboard for OS Metrics of Services in JSC Rancher (Pod Level) .....	18
5.2.3 Dashboard for OS Metrics of Services in JSC VMs.....	21
5.2.4 OMF Dashboard Printing Service.....	23
<b>6. Next steps- Implementation Roadmap M17 - M24.....</b>	<b>24</b>

## Table of Figures

Figure 1: Open Metrics Framework Dashboard .....	1
Figure 2: OMF Structure .....	7
Figure 3: OMF Metric Dashboards Structure .....	9
Figure 4: Visits/Visitors .....	10
Figure 5: Visited Pages.....	10
Figure 6: All Unique Active EBRAINS Users.....	11
Figure 7: New Unique Active EBRAINS Users .....	11
Figure 8: Open Gitlab Issues.....	12
Figure 9: Closed Gitlab Issues .....	12
Figure 10: New Zammad Issues.....	13
Figure 11: Open Zammad Issues .....	13
Figure 12: Closed Zammad Issues .....	13
Figure 13: CWL Workflows Executions .....	14
Figure 14: Services Uptime Status .....	15
Figure 15: OMF Operating System Dashboards Structure.....	15
Figure 16: Namespace Overview .....	16
Figure 17: Storage & Ephemeral Storage .....	16
Figure 18: CPU Usage & Quota per Workload .....	17
Figure 19: Memory Usage & Quota per Workload .....	17
Figure 20: Network Usage and Bandwidth per Workload .....	18
Figure 21: Packet Rate and Dropped Packet Rate per Workload .....	18
Figure 22: Headlines.....	19

Figure 23: CPU Usage and Quota per Pod.....	19
Figure 24: Memory Usage and Quota per Pod .....	19
Figure 25: Ephemeral Storage per Pod.....	20
Figure 26: Network Usage and Bandwidth per Pod.....	20
Figure 27: Packet Rate and Dropped Packet Rate per Pod .....	20
Figure 28: Storage I/O per Pod .....	21
Figure 29: VM Overview .....	21
Figure 30: Disk Usage .....	21
Figure 31: Memory Usage .....	22
Figure 32: CPU Usage.....	22
Figure 33: Network.....	23

### Table of Tables

Table 1: Abbreviations and Acronyms used.....	5
---	---

# 1. List of Abbreviations and Acronyms used

**Table 1: Abbreviations and Acronyms used**

Abbreviation/Acronym	Definition
CPU	Central Processing Unit
ETL	Extract, Transform, Load
JSC	Jülich Supercomputing Centre
I/O	Input/Output
IOPS	I/O operations per second
KG	Knowledge Graph
KPI	Key Performance Indicator
OMF	Open Metrics Framework
OS	Operating System
PVC	Persistent Volume Claim
RAM	Random access memory
RBAC	Role-based access control
VM	Virtual Machine

## 2. Introduction

This document presents the progress made in the development of the Open Metrics Framework (OMF) for EBRAINS 2.0 since the previous deliverable (D5.1-EBRAINS Requirements, Architecture and Guidelines). The OMF continues to play a pivotal role in monitoring, analysing, and reporting performance metrics to ensure the stability and usability of the EBRAINS platform. This deliverable provides an overview of the key advancements, challenges faced, and the roadmap ahead.

### 2.1 Document structure

D5.2 is organized into a series of sections to provide a clear and systematic presentation of its content:

**Section 2** introduces the purpose and scope of the deliverable, including a detailed breakdown of the organization of the document and a guidance on navigating the document effectively.

**Section 3** provides an overview of the OMF structure and technology stack.

**Section 4** outlines the key steps defined in the Implementation Plan.

**Section 5** covers the progress of the OMF Implementation up to the submission of this deliverable.

**Section 6** summarises the next steps and presents the Implementation Roadmap through to the end of the second year of EBRAINS 2.0.

### 2.2 How to read this deliverable

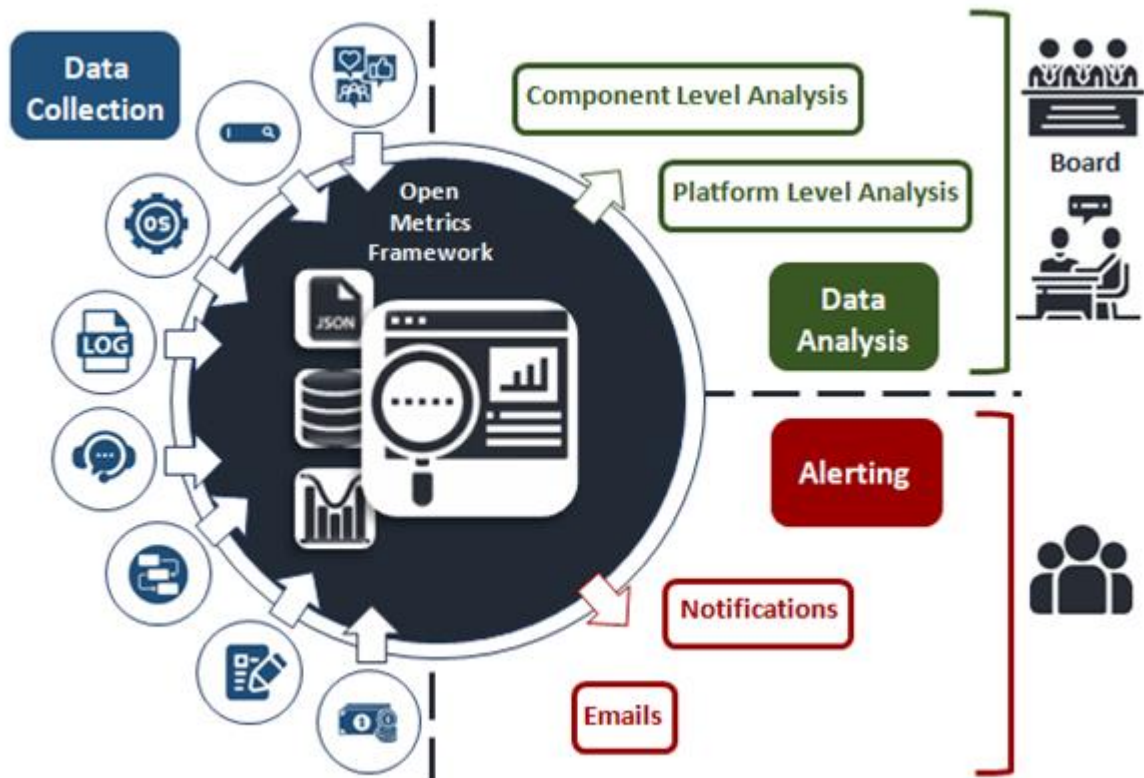
Deliverable **D5.2** details the progress and ongoing efforts in the implementation of the Open Metrics Framework (OMF). It is a Public Report offering a comprehensive overview of the current status and the next steps of the OMF.

This information is already communicated to EBRAINS 2.0 partners through the different meetings, documents or presentations.

**D5.2** is closely linked to **D5.1** [M12], that details the progress in the implementation and operation of the EBRAINS Infrastructure, and sets the stage for future WP5 deliverables, **D5.3** [M18], and **D5.4** [M34]. Furthermore, it maintains an organic connection with **WP8** deliverables, ensuring synergy between the WPs. This deliverable is also anticipated to serve as a reference for future reports and project deliverables.

### 3. Overview

The Open Metrics Framework (OMF) is an analytics platform designed to monitor, analyse, and report on the performance of the EBRAINS infrastructure. It plays a crucial role in tracking Key Performance Indicators (KPIs) and addressing gaps in platform analytics. By providing insights into various data categories such as user behaviour, system performance, and service availability, the OMF enables stakeholders to optimize operations and make data-driven decisions. For more details on its structure (Figure 2) and technology stack, refer to Deliverable D5.1.



**Figure 2: OMF Structure**

## 4. Implementation Plan

To ensure effective implementation of Key Performance Indicators (KPIs) and monitoring requirements within the Open Metrics Framework (OMF), a structured approach is followed. This process ensures data collection, efficient processing, and insightful visualization of metrics. The methodology consists of the following key steps:

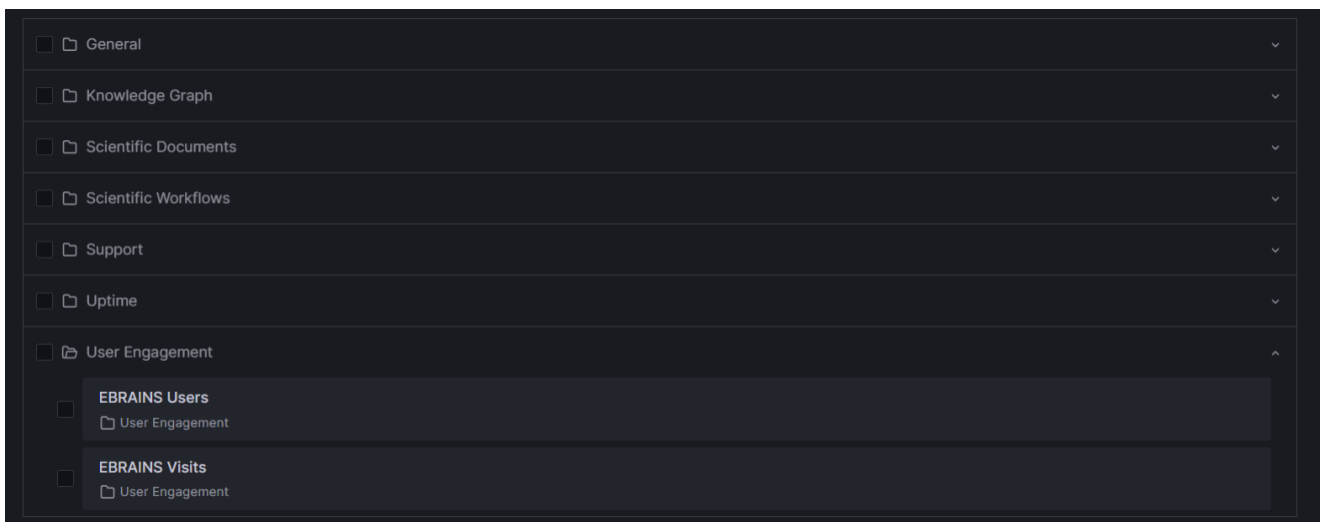
- **Understand the KPI and Define Requirements:** The first step involves analysing the KPI to understand its purpose and the insights it should provide. It is essential to determine what will be valuable for stakeholders in the dashboard, whether it is trend analysis, anomaly detection, or performance comparisons. Different visualization approaches, such as graphs, tables, or heatmaps, should be considered to best represent the data. Additionally, the level of detail (granularity) must be defined to ensure the dashboard provides meaningful insights for decision-making.
- **Identify Data Sources:** Once the KPI is defined, the next step is to locate existing data repositories where relevant metrics are stored. This involves assessing whether reliable data sources already exist or if new data collection methods are necessary. If no suitable repositories are available, further analysis is required to determine how the missing data can be generated, stored, or extracted from alternative sources.
- **Reach Out to the Relevant Point of Contact:** To ensure access to the required data, it is crucial to engage with the appropriate stakeholders or technical teams managing the data sources. Discussions should focus on the specific monitoring requirements and the feasibility of extracting the necessary metrics. Depending on the data access policies, it may be necessary to obtain permissions to retrieve data from existing endpoints or acquire read access to databases and other structured data sources.
- **Design the ETL Data Pipeline:** With access to the required data, the next step is to design the **ETL (Extract, Transform, Load) pipeline**, which will enable the efficient flow of information from data collection to processing and ingestion into the Open Metrics Framework. The **Extract** phase defines how data will be retrieved from various sources, whether databases, APIs, or other repositories. The **Transform** phase focuses on normalizing, and structuring the data to ensure consistency, integrity, and usability for analysis. Finally, the **Load** phase involves storing the processed data in the appropriate OMF backend storage structures, making it readily available for visualization.
- **Develop the ETL Solution:** Develop the ETL Solution: This step focuses on implementing the pipeline as designed, ensuring reliable and seamless data processing within the Open Metrics Framework. A key aspect of the implementation is to ensure that all data entries include the systemic field `@timestamp`, enabling accurate time-based analysis and trend monitoring.
- **Create the Dashboard:** The next step is to build an interactive and visually intuitive dashboard that presents the KPIs in an accessible and meaningful way. The dashboard should be designed with usability in mind, allowing stakeholders to easily interpret the displayed metrics. To enhance user experience and facilitate in-depth analysis, features such as filtering options and time-range selection should be incorporated. Additionally, it is crucial to select the most appropriate panel type for each metric, ensuring that the visualization effectively conveys the intended insights.
- **Provide User Access to the Dashboard:** Once the dashboard is developed, the final step is to ensure that the appropriate users can access their corresponding dashboards securely. Users can log in using their EBRAINS credentials and via role-based access control (RBAC), access only the dashboards relevant to their roles and services. This ensures that users interact only with the data they are authorized to view, maintaining security.
- **Addressing User Requests for Enhancements:** After users access the dashboards, they may request additional features to enhance analysis. If a requested addition is minor and quick to implement, it is provided promptly. For more complex modifications or additions, an effort

estimation is conducted, and the request is prioritized based on its impact, feasibility, and available resources.

## 5. Implementation Progress M1 - M16

### 5.1 Implementation Progress M1 - M12

During this period, apart from the deployment and configuration of the development stack, the primary focus was on the implementation of dashboards containing the metrics and KPIs that need to be reported periodically. These dashboards within the Open Metrics Framework are logically structured based on the category of data they contain, ensuring a user-friendly and intuitive experience for end users. By organizing dashboards into distinct categories (Figure 3), users can efficiently locate and access the relevant insights without unnecessary complexity. This structured approach enhances navigation and usability, allowing stakeholders to focus on specific metrics relevant to their reporting needs.

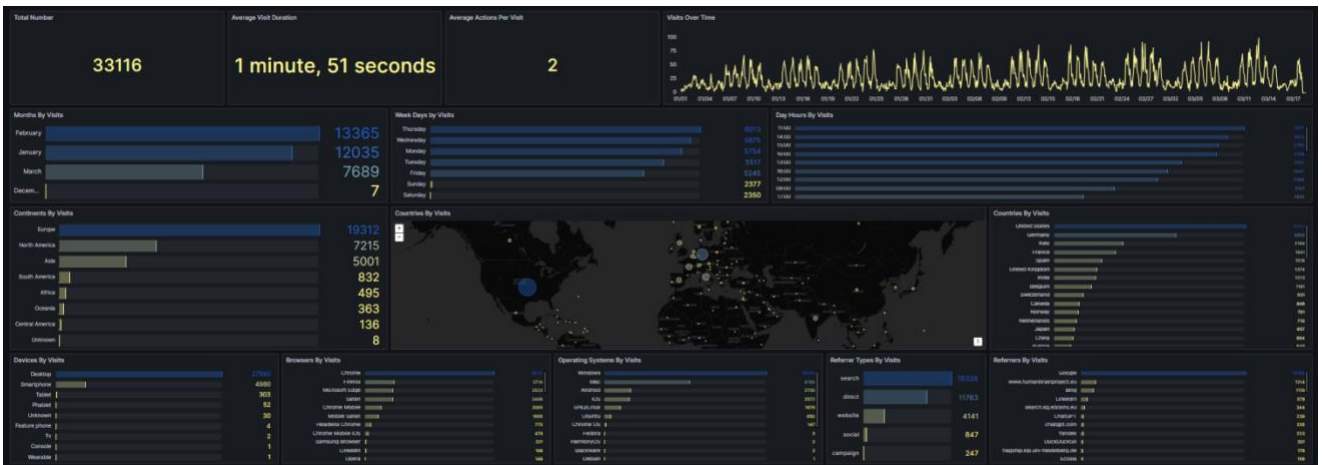


**Figure 3: OMF Metric Dashboards Structure**

This section presents a brief overview of the dashboards that have been implemented and made available up to the submission of deliverable D5.1. These dashboards provide key insights into various metrics within the Open Metrics Framework. A more detailed description of their structure and functionality can be found in D5.1.

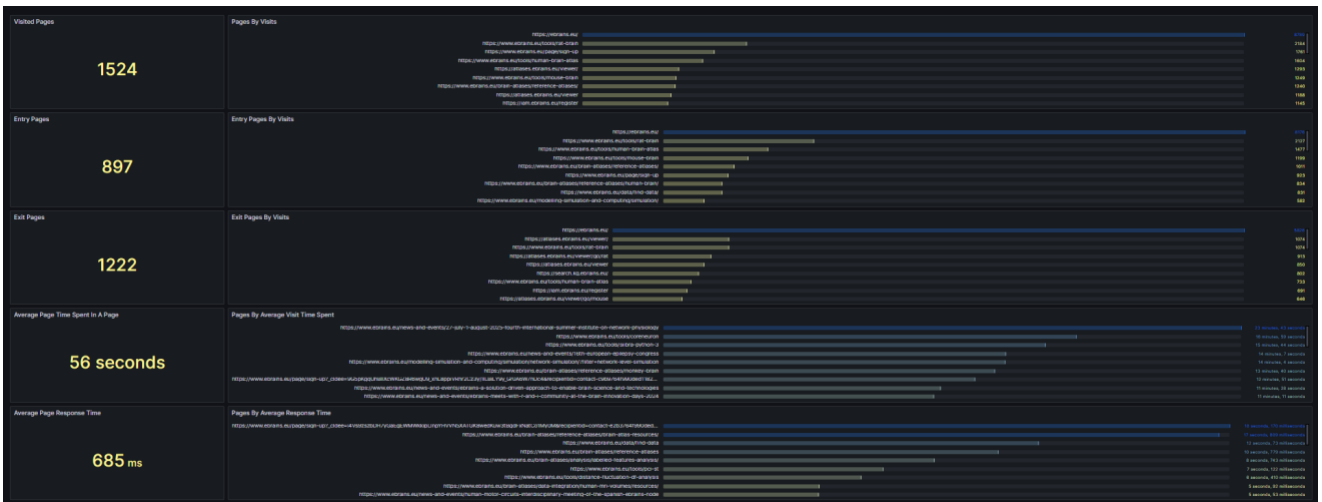
#### 5.1.1 *Dashboard for unique visits and visitors*

The Unique Visits and Visitors Dashboard (Figure 4) provides insights into visits, visitors, user engagement, and trends and allows filtering for different types of analysis.



**Figure 4: Visits/Visitors**

It also tracks key metrics, including total page visits, entry and exit pages for page analysis (Figure 5).

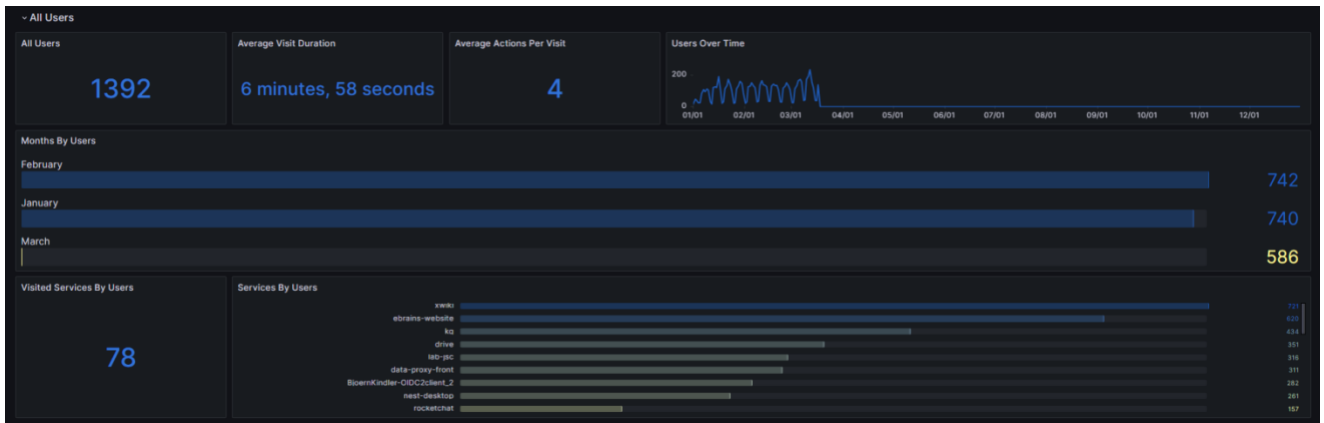


**Figure 5: Visited Pages**

Lastly, the value for **KPI13 – Number of Unique EBRAINS Visitors** is obtained from this dashboard.

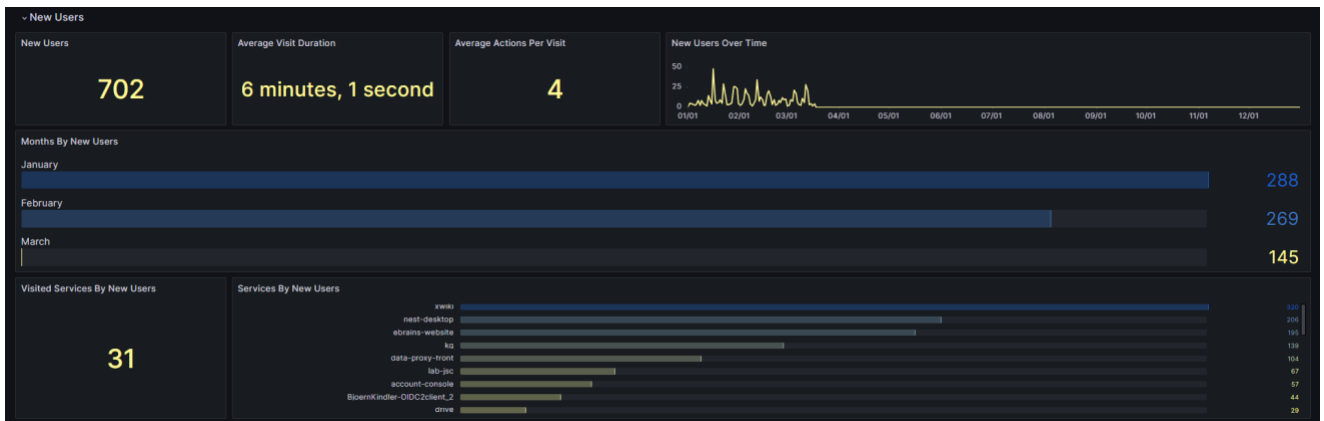
### 5.1.2 Dashboard for unique active EBRAINS users

The Unique Active EBRAINS Users Dashboard (Figure 6) provides insights into user activity, engagement trends, and service interactions, helping to analyse platform usage over time.



**Figure 6: All Unique Active EBRANS Users**

It also includes dedicated sections focused on new users, highlighting their activity and adoption patterns (Figure 7).



**Figure 7: New Unique Active EBRANS Users**

Lastly, this dashboard is used in **KPI22 – Number of user requests for access; number of users of resources, such as collections, data, services.**

### 5.1.3 Dashboard for GitLab issues

The GitLab Issues Dashboard (Figure 8) provides an overview of issue activity, tracking trends in task and incident reporting. It offers insights into issue distribution, severity, and project-level impact over time.

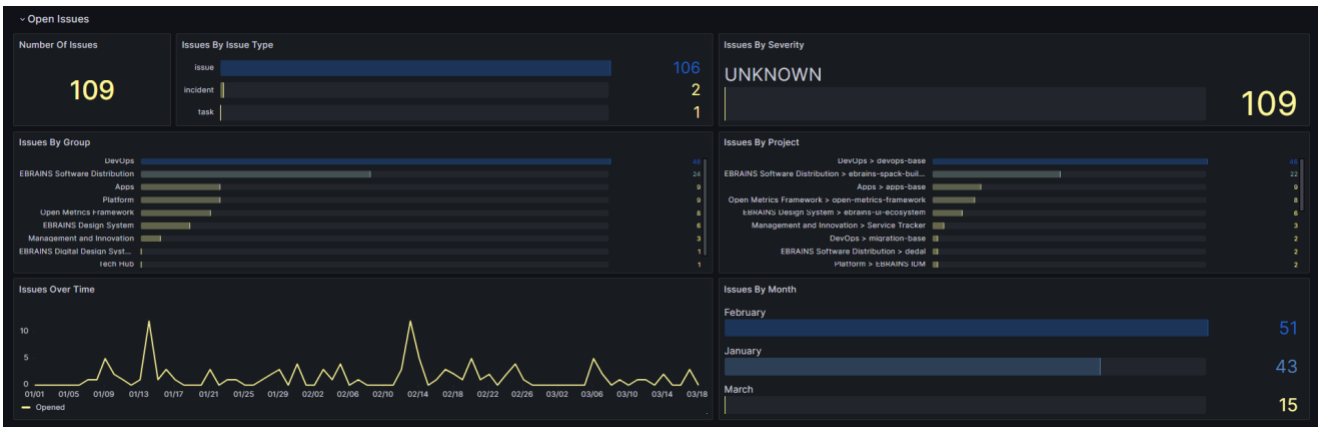


Figure 8: Open Gitlab Issues

It also tracks issue resolution trends, monitoring closure rates, resolution time, and distribution by severity and project (Figure 9).

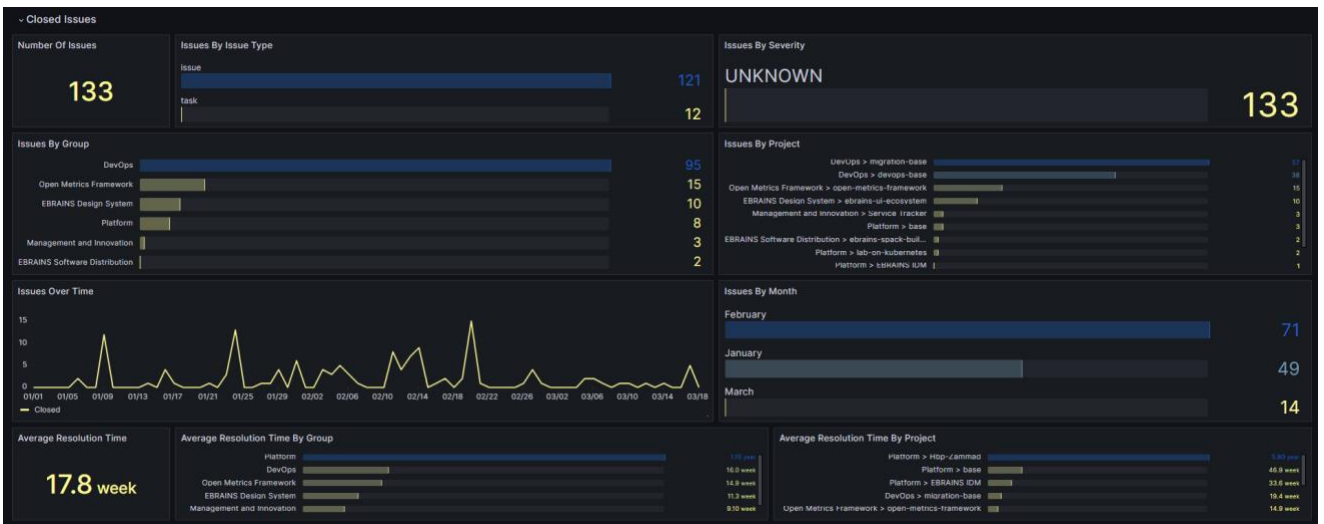


Figure 9: Closed Gitlab Issues

### 5.1.4 Dashboard for Zammad issues

EBRAINS Support uses Zammad<sup>1</sup>, so the Zammad Issues Dashboard (Figure 10) tracks new issues, categorizing them by severity and creation date while also analysing the number of unique users reporting them.

<sup>1</sup> <https://zammad.com/en>



Figure 10: New Zammad Issues

It also includes those with a first response, categorizing them by severity, tag, and date while providing insights into initial response times (Figure 11).

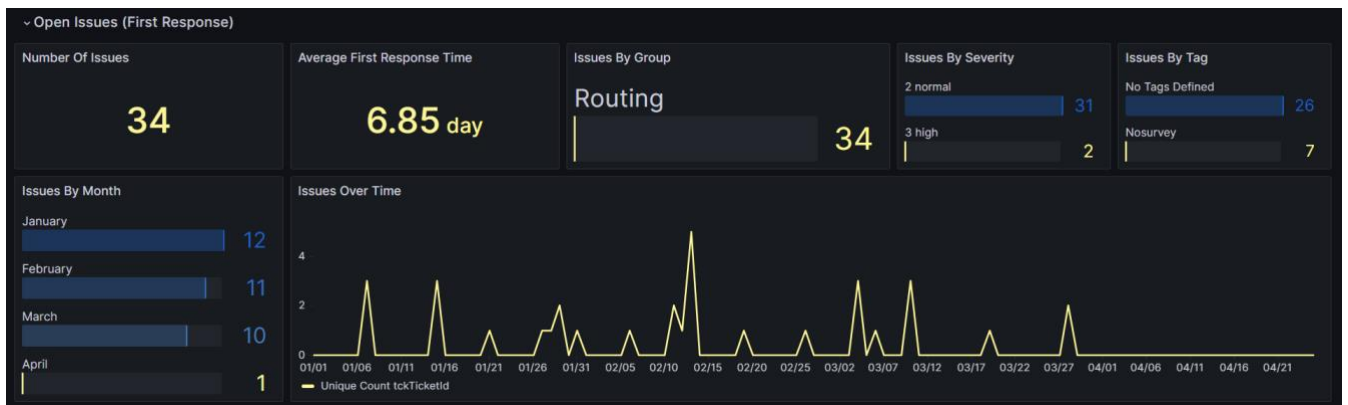


Figure 11: Open Zammad Issues

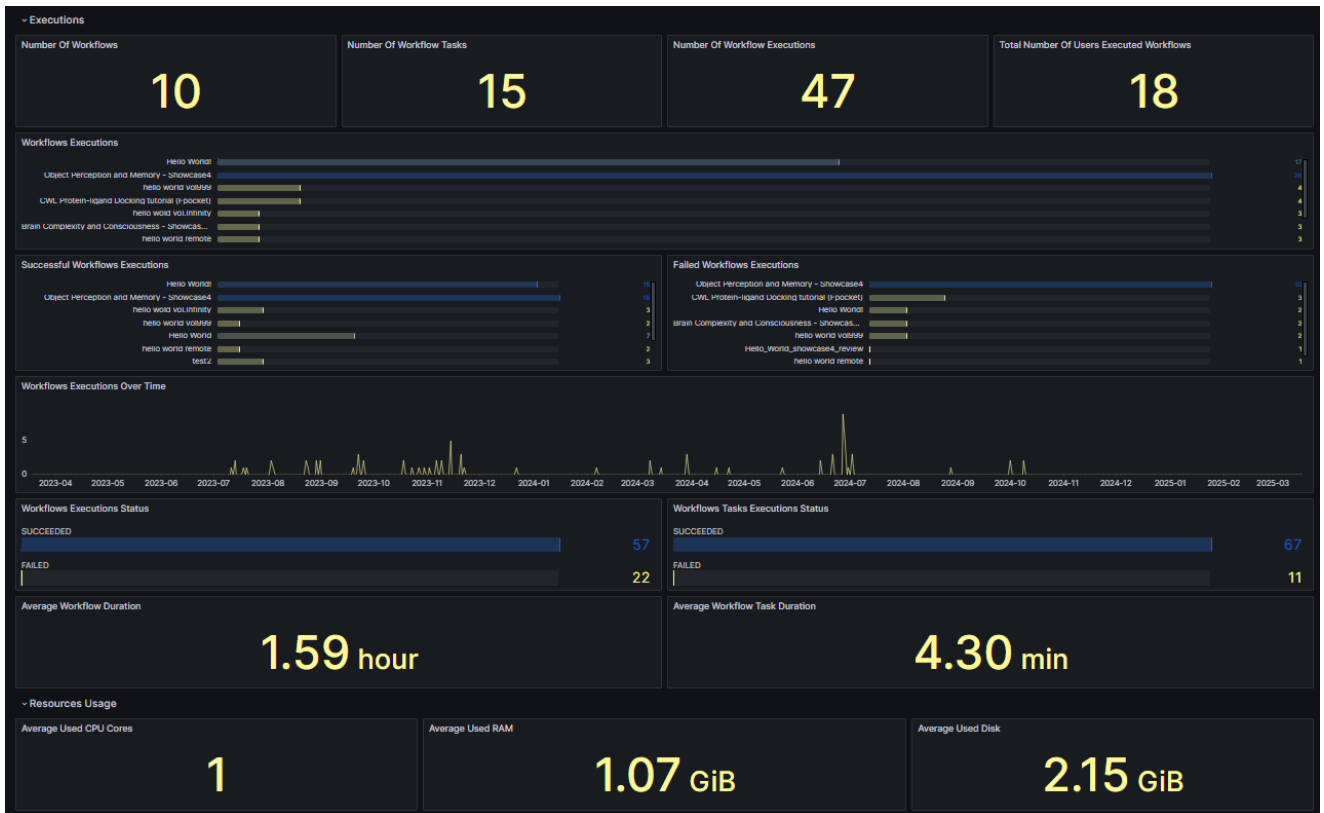
Lastly, it tracks resolved issues, including their severity, tag, and resolution date, while providing detailed insights into resolution times (Figure 12).



Figure 12: Closed Zammad Issues

### 5.1.5 Dashboard for CWL workflows executions

The CWL<sup>2</sup> Workflows Execution Dashboard (Figure 13) tracks workflow activity, monitoring execution counts, success and failure rates, and task durations. It also provides insights into resource usage and allocation during execution.



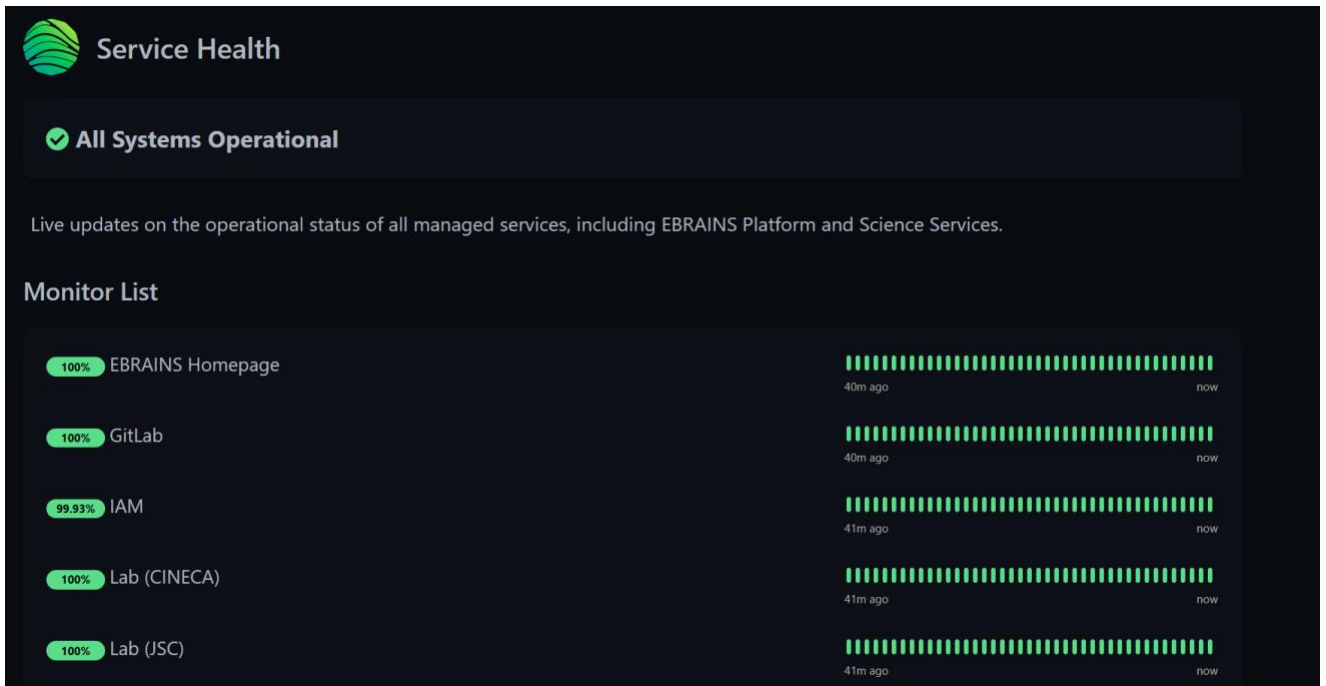
**Figure 13: CWL Workflows Executions**

Lastly, the information of this dashboard is part of **KPI06 – Number of Community Generated Scientific Workflows (notebooks, CWL workflows, services, libraries) extending and/or building on EBRAINS.**

### 5.1.6 Core Services Operational Monitoring

Core services in EBRAINS support neuroanalysis, data management, and user tools. To ensure reliability, Kuma monitors uptime and response times (Figure 14), providing real-time metrics and alerts. These insights are needed for the provision of **KPI07 – Availability (uptime) of EBRAINS RI.**

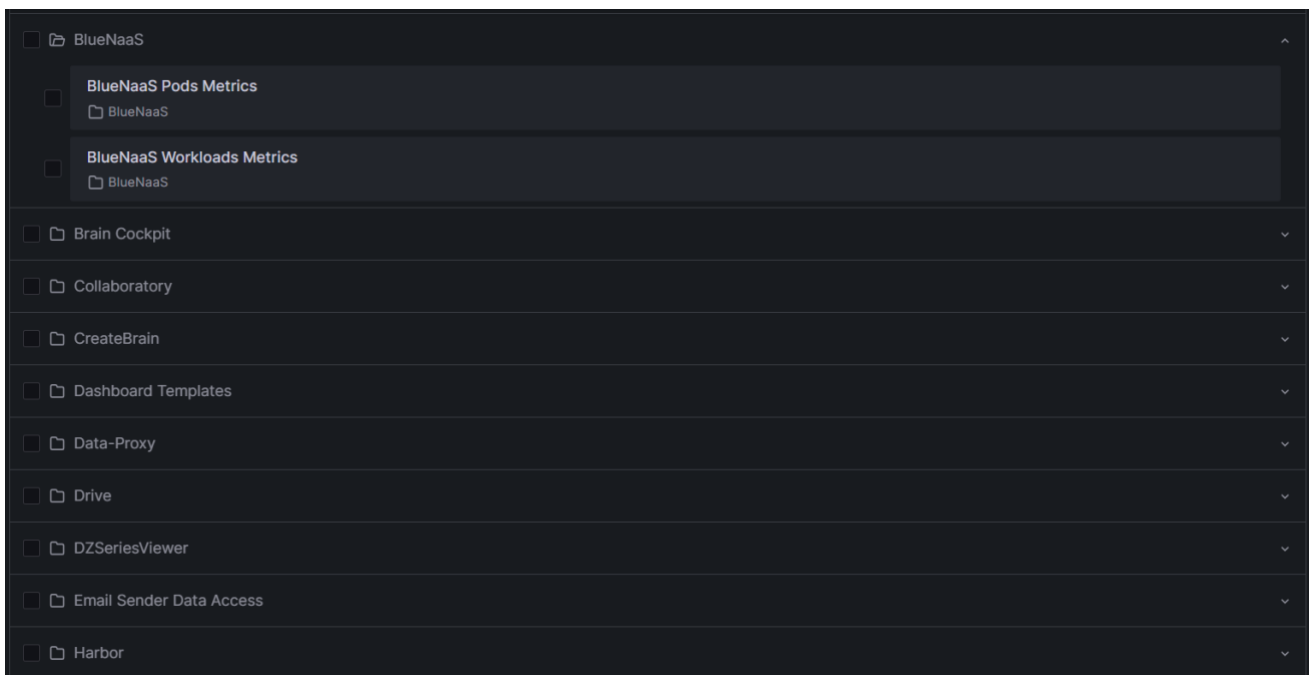
<sup>2</sup> <https://www.commonwl.org/>



**Figure 14: Services Uptime Status**

## 5.2 Implementation Progress M13 - M16

This section provides an overview of the implementations completed between Months M13 and M16 within the Open Metrics Framework. During this period, the primary focus was on the implementation of dashboards for operating system metrics, addressing the high demand from service owners for better visibility into the performance of their services. To enhance security and usability, these dashboards are organized per service name (Figure 15), ensuring that users can easily navigate to the relevant data while maintaining controlled access to service-specific metrics. They offer key insights into resource utilization, system performance, and workload efficiency.



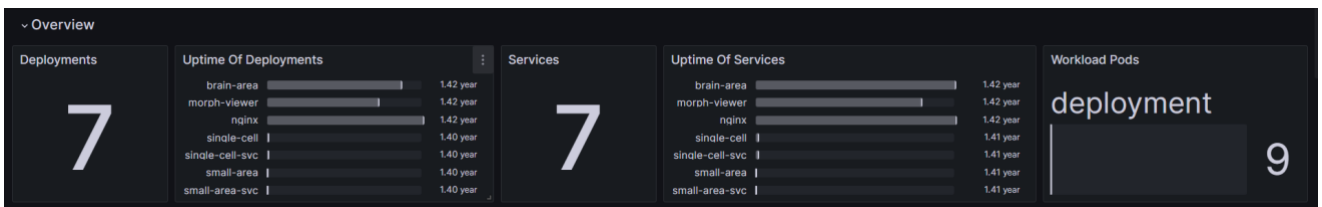
**Figure 15: OMF Operating System Dashboards Structure**

The section aims to present these updates in a clear and structured manner, without delving into technical details.

### 5.2.1 Dashboard for OS Metrics of Services in JSC Rancher (Workload Level)

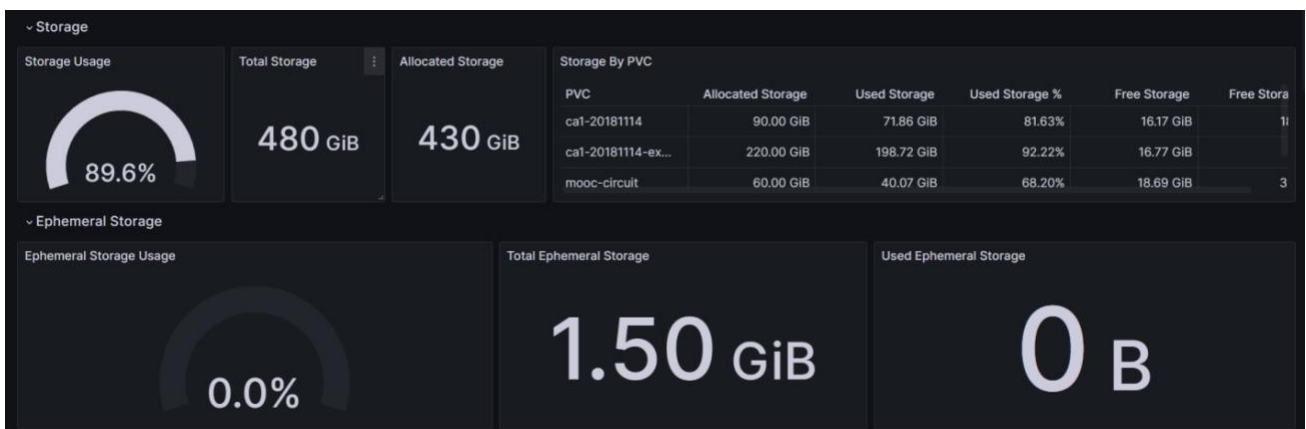
The OS Metrics Dashboard for Services in JSC Rancher provides a comprehensive overview of system performance at the workload level. This dashboard enables monitoring of key operating system metrics, ensuring that deployed services function optimally within the JSC Rancher environment.

The Overview Section of the dashboard (Figure 16) provides key insights into the number of deployments, services, and workload pods within the system. It also tracks the uptime of deployments and services, offering a clear view of their stability and operational duration. This information helps service owners monitor availability and ensure the reliability of their infrastructure.



**Figure 16: Namespace Overview**

The Storage Section of the dashboard (Figure 17) provides detailed insights into total, allocated, and ephemeral storage usage. It tracks overall storage consumption, highlighting the percentage of used storage and available free space. Additionally, the Storage by PVC (Persistent Volume Claim) panel offers a breakdown of allocated and utilized storage for different volumes, helping administrators monitor resource allocation. The Ephemeral Storage Section (Figure 17) displays the total available ephemeral storage and its current usage, ensuring that temporary storage demands are effectively managed.



**Figure 17: Storage & Ephemeral Storage**

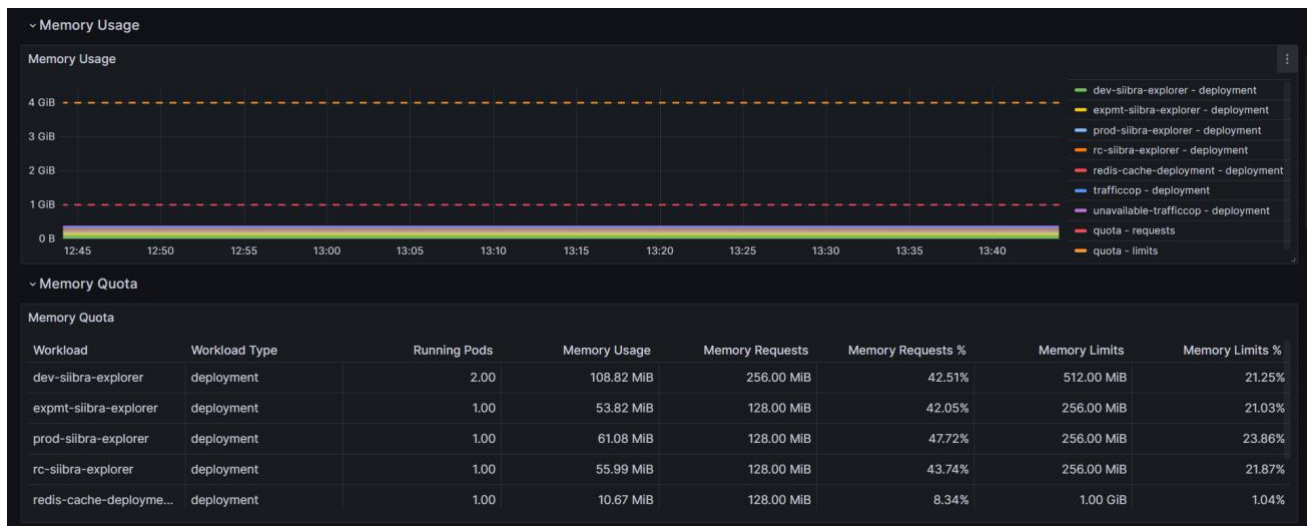
The CPU Usage and Quota Section of the dashboard (Figure 18) provides real-time monitoring of CPU consumption across different workloads. It visualizes CPU usage trends, helping administrators assess resource utilization and detect anomalies. The CPU Quota panel tracks the number of running pods, CPU requests, and CPU limits, offering insights into workload resource allocation. This information enables

service owners to optimize CPU performance, prevent resource bottlenecks, and ensure efficient system operations.



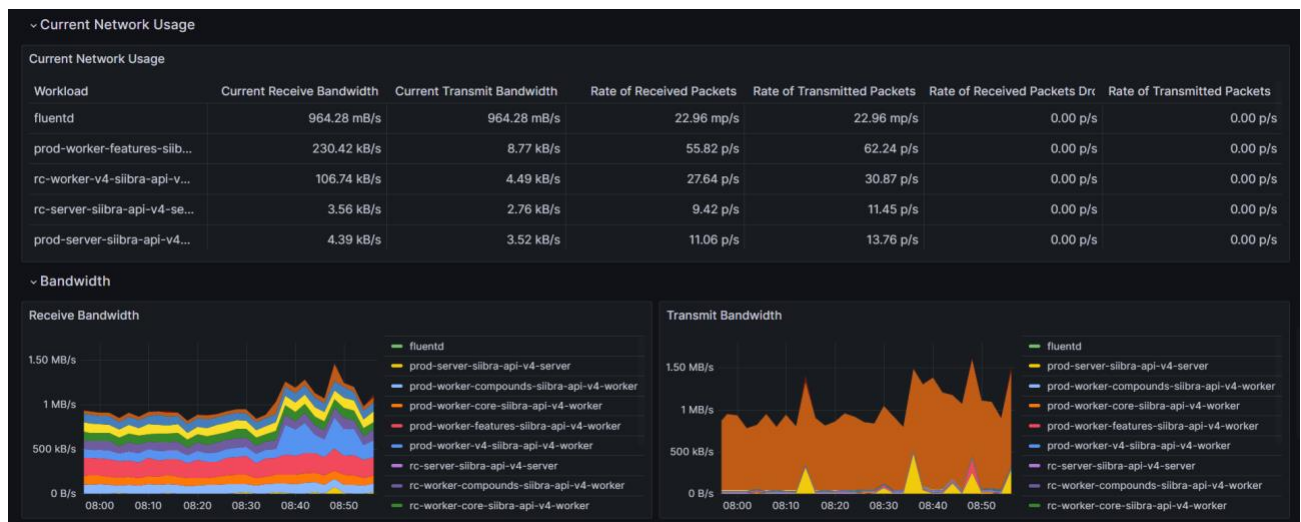
**Figure 18: CPU Usage & Quota per Workload**

The Memory Usage and Quota Section of the dashboard (Figure 19) provides real-time monitoring of memory consumption across different workloads. It visualizes memory usage trends, helping administrators track resource utilization and detect potential inefficiencies. The Memory Quota panel displays the number of running pods, memory requests, and memory limits, offering insights into workload resource allocation. This information supports effective memory management, preventing resource shortages and ensuring system stability.



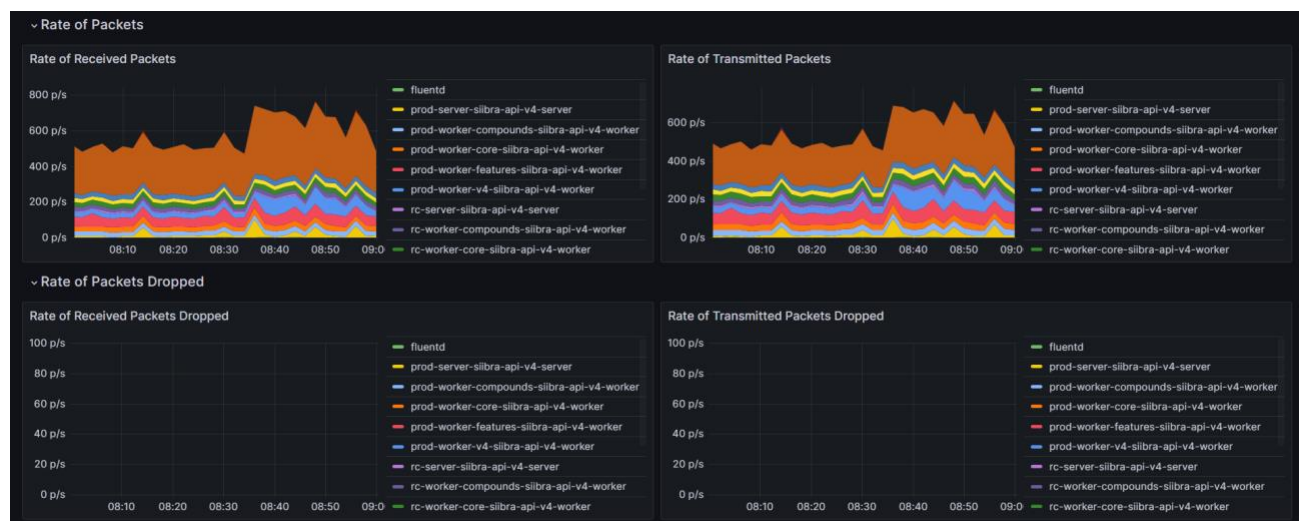
**Figure 19: Memory Usage & Quota per Workload**

The Network Usage and Bandwidth Sections of the dashboard (Figure 20) provide real-time insights into network activity across workloads. It monitors current receive and transmit bandwidth, along with the rate of received and transmitted packets, helping service owners identify network bottlenecks and performance issues. The bandwidth visualization panels display trends in incoming and outgoing traffic, allowing service owners to track data flow and optimize network resource allocation.



**Figure 20: Network Usage and Bandwidth per Workload**

The Packet Rate and Dropped Packets Sections of the dashboard (Figure 21) provide insights into network traffic performance by monitoring the rate of received and transmitted packets across different workloads. The visualizations help identify traffic patterns and potential network congestion. Additionally, the dropped packets panels track lost packets, ensuring network reliability by detecting disruptions or inefficiencies in data transmission.

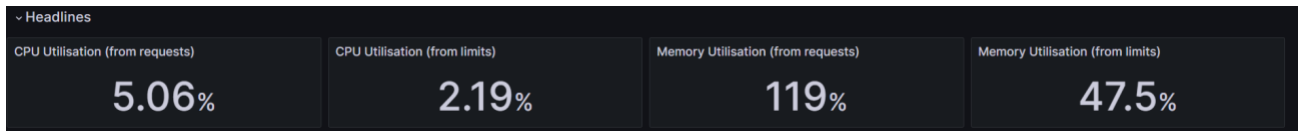


**Figure 21: Packet Rate and Dropped Packet Rate per Workload**

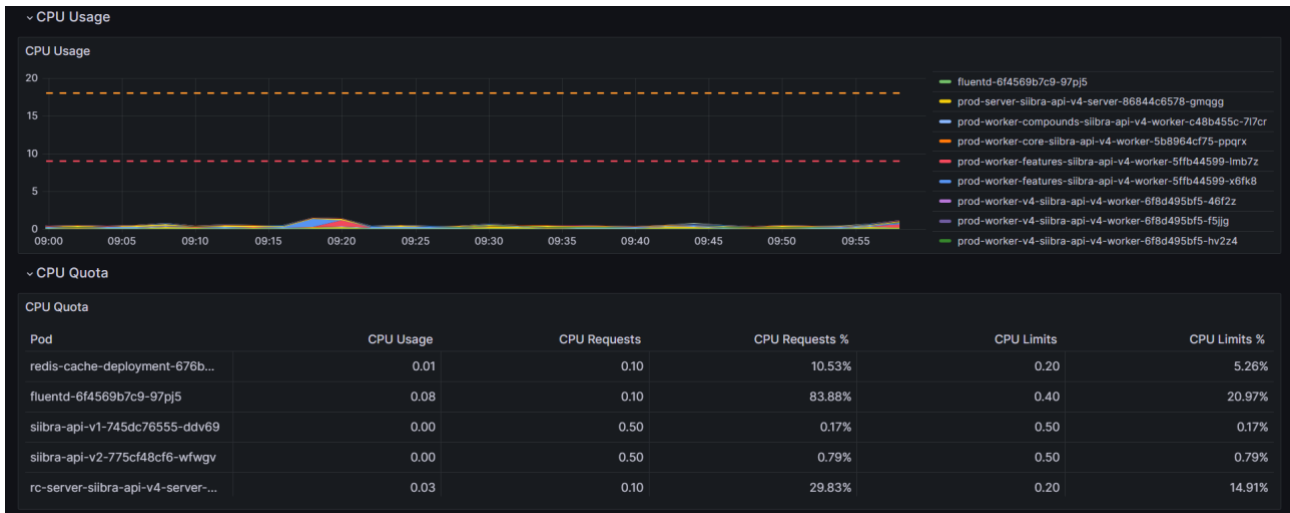
## 5.2.2 Dashboard for OS Metrics of Services in JSC Rancher (Pod Level)

A similar OS Metrics Dashboard has been implemented at the pod level, providing a more detailed view of resource usage and performance for individual pods, allowing for more granular monitoring.

The Headlines Section of the Pod-Level OS Metrics Dashboard (Figure 22) provides a quick overview of CPU and memory utilization, comparing usage against requests and limits. This helps services owners assess resource consumption and detect potential constraints efficiently.


**Figure 22: Headlines**

The CPU Usage and Quota Section (Figure 23) tracks real-time CPU consumption, requests, and limits for individual pods. It helps service owners optimize resource allocation and detect potential bottlenecks for better performance.


**Figure 23: CPU Usage and Quota per Pod**

The Memory Usage and Quota Section (Figure 24) tracks real-time memory consumption, requests, and limits for individual pods, helping service owners optimize resource allocation and prevent overutilization.


**Figure 24: Memory Usage and Quota per Pod**

The Ephemeral Storage Section (Figure 25) tracks ephemeral storage requests, and limits for individual pods. It helps administrators manage temporary storage allocation efficiently, preventing resource constraints and ensuring stable system performance.

Pod	Ephemeral Storage Limits	Ephemeral Storage Requests
copy-cache	256.00 MiB	64.00 MiB
cronjob-pod-usage-29041114-6x8cp	256.00 MiB	64.00 MiB
cronjob-pod-usage-29041115-tcwtq	256.00 MiB	64.00 MiB
cronjob-pod-usage-29041116-n4tb8	256.00 MiB	64.00 MiB
cronjob-trim-old-log-usage-29020421-b7zq8	256.00 MiB	64.00 MiB
cronjob-trim-old-log-usage-29034821-dgqk6	256.00 MiB	64.00 MiB

Figure 25: Ephemeral Storage per Pod

The Current Network Usage and Bandwidth Sections (Figure 26) track real-time network activity, including receive and transmit bandwidth and packet rates for individual pods.

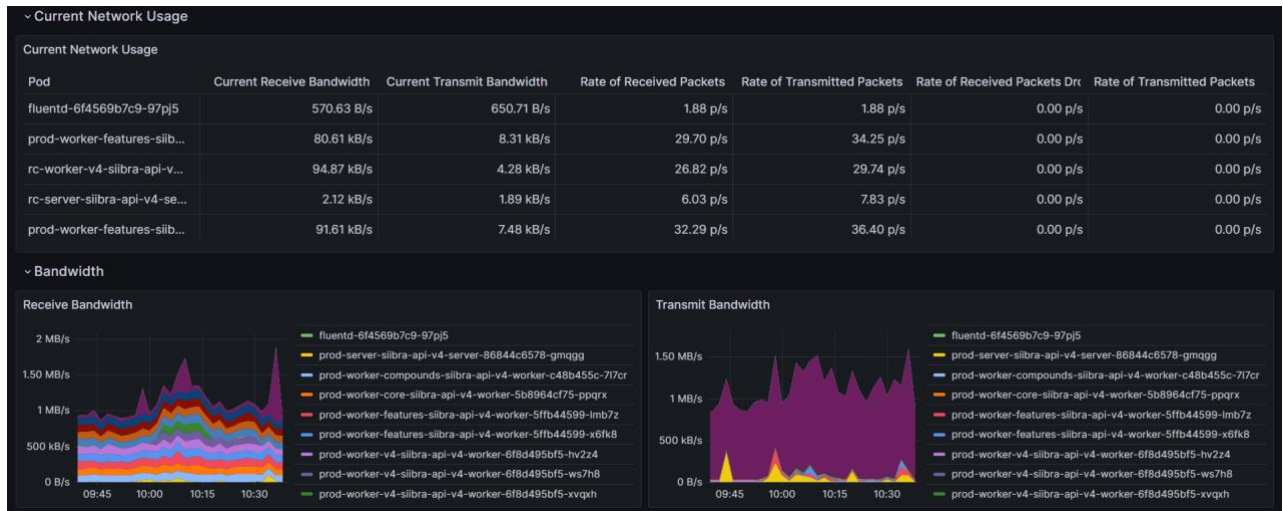


Figure 26: Network Usage and Bandwidth per Pod

The Packet Rate and Drop Packets Sections (Figure 27) track received and transmitted packet rates for individual pods, helping administrators analyse network traffic patterns and potential congestion. Additionally, it monitors dropped packets, ensuring network stability and identifying potential communication issues.

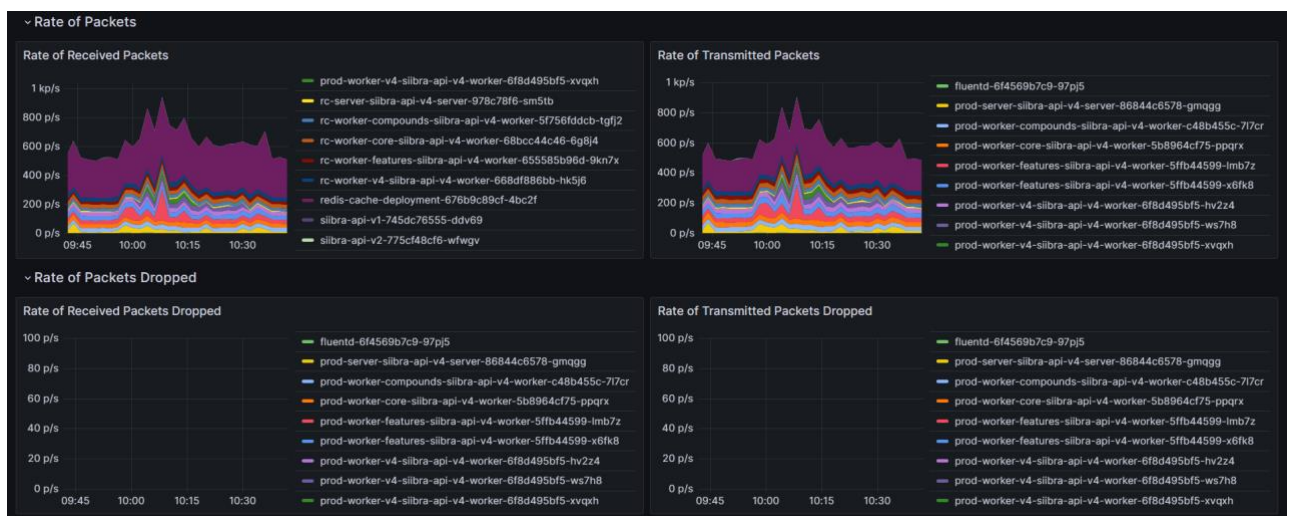


Figure 27: Packet Rate and Dropped Packet Rate per Pod

The Storage I/O Section (Figure 28) tracks I/O operations per second (IOPS) and read/write throughput for individual pods. It helps service owners monitor storage performance, detect bottlenecks, and optimize resource allocation for efficient data processing.

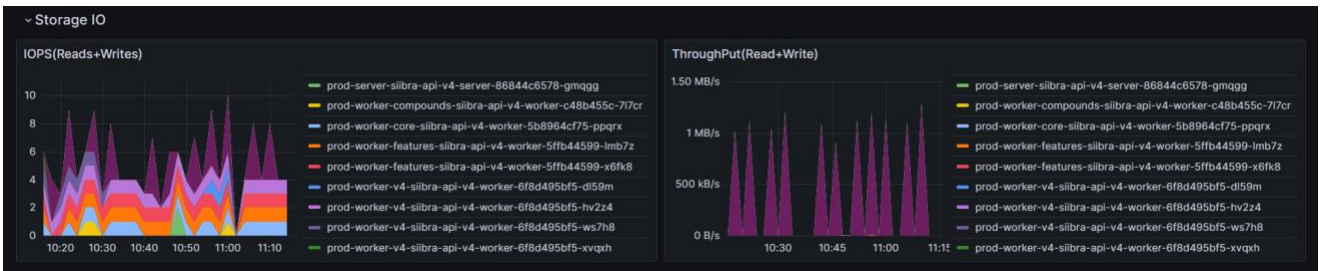


Figure 28: Storage I/O per Pod

### 5.2.3 Dashboard for OS Metrics of Services in JSC VMs

The OS Metrics Dashboards for Services in JSC VMs provide a comprehensive overview of system performance for the services deployed on JSC virtual machines. These dashboards enable monitoring of key operating system metrics, ensuring that services function efficiently while offering service owners detailed insights into resource utilization and performance trends.

The Overview Section (Figure 29) provides a high-level summary of system resource utilization for services deployed on JSC VMs. It tracks key metrics such as CPU pressure, memory usage, swap usage, root filesystem utilization, and system uptime. This enables service owners to quickly assess performance, detect resource constraints, and ensure optimal operation of their services.

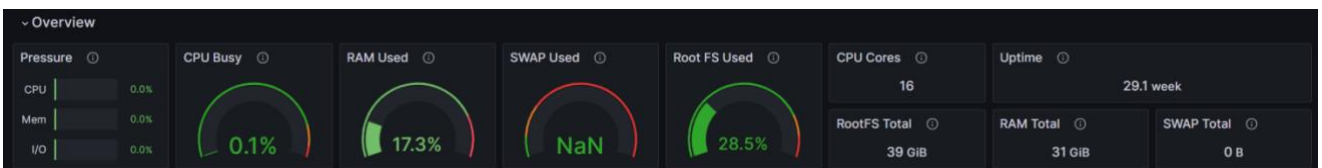


Figure 29: VM Overview

The Disk Section (Figure 30) provides insights into disk usage, I/O utilization, operations, and throughput for services deployed on JSC VMs. It helps service owners track storage performance, detect potential bottlenecks, and optimize disk resource allocation to ensure system efficiency.



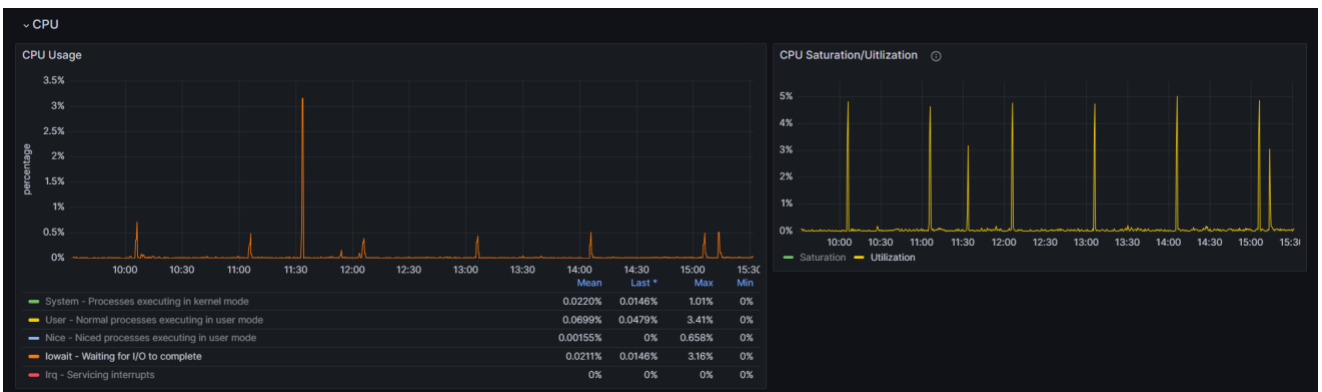
Figure 30: Disk Usage

The Memory Section (Figure 31) provides detailed insights into overall memory usage, RAM utilization, and memory activity for services running on JSC VMs. It tracks active and inactive memory, cache, swap usage, and memory saturation, allowing service owners to identify inefficiencies and optimize memory allocation.



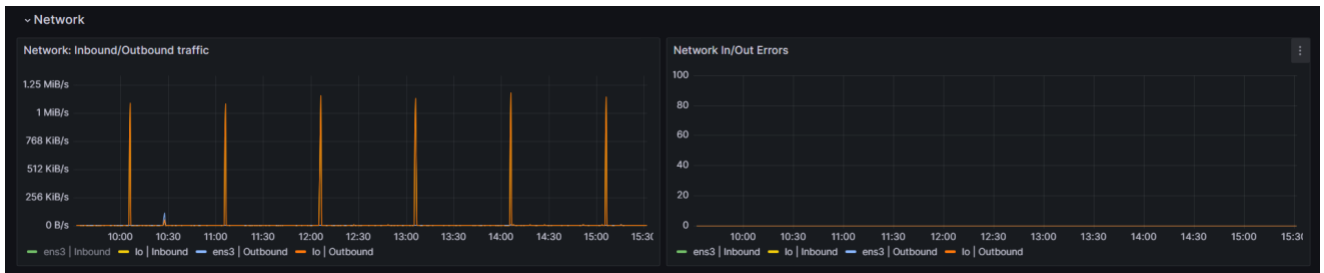
**Figure 31: Memory Usage**

The CPU Section (Figure 32) tracks CPU utilization, saturation, and process execution for services running on JSC VMs. It provides insights into system, user, and I/O wait processes, helping service owners analyse performance trends and optimize resource allocation.



**Figure 32: CPU Usage**

The Network Section (Figure 33) tracks inbound and outbound traffic, along with network errors, for services running on JSC VMs. It provides insights into bandwidth usage and potential connectivity issues, helping service owners optimize network performance and detect anomalies.



**Figure 33: Network**

### 5.2.4 OMF Dashboard Printing Service

To facilitate broader access to Open Metrics Framework (OMF) dashboards, a dashboard printing service has been implemented. This service addresses scenarios where users may require access to dashboard insights but are either not authorized to log into the OMF or need metrics for services they do not own. For example, users preparing specific reports may need structured metric snapshots without direct access to the live dashboard.

The current first version of the service allows us to print any OMF dashboard, but only with default filters and time frames applied. However, as dashboards often contain dynamic filtering options, a more flexible solution is required to accommodate different reporting needs. To address this, future enhancements will extend the service’s functionality, enabling parametric selection of specific dashboards and allowing customization of filters and timeframes in the printed reports. The printing service will be executable only by OMF administrators for requests deemed appropriate for generating static reports, ensuring controlled and secure distribution of key performance insights. These improvements will enhance flexibility and usability, allowing OMF data to be efficiently utilized across various reporting and operational needs.

## 6. Next steps- Implementation Roadmap M17 - M24

During Months M17 to M24, the focus will be on expanding the Open Metrics Framework with new dashboards and monitoring capabilities to enhance data visibility and support data-driven decision-making. Key developments will include the implementation of dashboards for KG Content, KG Search Bar Usage, and EBRAINS Publications/Citations, providing deeper insights into the structure, search trends, and academic impact of EBRAINS.

Enhancements based on both existing user feedback and future requests will be a priority. Users may request new features or adjustments to improve usability and the relevance of displayed metrics. More complex changes will undergo effort estimation and prioritization based on feasibility and impact.

Lastly, the dashboard printing service will be extended, as described in section 5.2.4, to support custom filter selection and dynamic timeframes, allowing users to generate reports tailored to their specific needs.