

EOSC Monitoring: Architecture and Interoperability Guidelines

Version 0.9
September 2022

EOSC Monitoring: Architecture and Interoperability Guidelines

Lead by National Infrastructures for Research and Technology – GRNET SA

Authored by Kostas Koumantaros, Themis Zamani, Kostas Kagkelidis, Angelos Tsalapatis, Chrysa Thermolia, - National Infrastructures for Research and Technology - GRNET, GREECE

Cyril L'Orphelin, IN2P3 Computing Center (CNRS), France

Emir Imamagic, Daniel Vrcic, Katarina Zailac, University of Zagreb University Computing Centre (SRCE), Croatia

Dissemination Level of the Document

Public

Abstract

Monitoring is the key service needed to gain insights into an infrastructure. It needs to be continuous and on-demand to quickly detect, correlate, and analyse data for a fast reaction to anomalous behaviour. The challenge of this type of monitoring is how to quickly identify and correlate problems before they affect end-users and ultimately the productivity of the organisation. Management teams can monitor the availability and reliability of the services from a high level view down to individual system metrics and monitor the conformance of multiple SLAs. The EOSC Monitoring service combines two operational monitoring services: the EOSC-CORE and the EOSC-Exchange Monitoring Services, respectively monitoring the EOSC-Core services (EOSC Core Monitoring) and the services onboarded to the Marketplace (EOSC-Exchange Monitoring). The EOSC Monitoring services were implemented adopting the ARGO technology. This document describes the current architecture of the EOSC Monitoring and provides guidelines for five (5) integration Options available.

Version History

Version	Date	Authors/Contributors	Description
Vo.1	01/04/2021	Kostas Koumantaros, Themis Zamani - GRNET	Initiation – Proposed ToC – First draft
Vo.2	01/04/2021	Kostas Koumantaros, Themis Zamani - GRNET	1 st draft
Vo.3	05/04/2021	Kostas Koumantaros, Themis Zamani – Kostas Kagkelidis GRNET	Chapters 1,2
Vo.4	20/5/2021	Kostas Kagkelidis, Angelos Tsalapatis, Chrysa Thermolia, - GRNET	Chapters 3,4,5
Vo.5	1/6/2021	Kostas Koumantaros, Themis Zamani, Kostas Kagkelidis, Angelos Tsalapatis, Chrysa Thermolia, - GRNET, Cyril L'Orphelin -IN2P3 Emir Imamagic, Daniel Vrcic, Katarina Zailac - SRCE	Revision Chapter 5
Vo.6	1/4/2022	Kostas Koumantaros, Themis Zamani, Kostas Kagkelidis, Angelos Tsalapatis, Chrysa Thermolia, - GRNET, Cyril L'Orphelin -IN2P3 Emir Imamagic, Daniel Vrcic, Katarina Zailac - SRCE	Revision of all chapters
Vo.7	1/6/2022	Kostas Koumantaros - GRNET	Minor updates – edits based on feedback received.
Vo.8	1/8/2022	Michelle Williams – GEANT	Review – Feedback.
Vo.9	20/9/22	Kostas Koumantaros - GRNET	Minor updates – edits based on feedback received.
V1.0			

Copyright Notice



This work by Parties of the EOSC Future Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/). The EOSC Future project is co-funded by the European Union Horizon Programme call INFRAEOSC-03-2020, Grant Agreement number 101017536.

Table of Contents

1	Glossary.....	3
2	Introduction.....	4
2.1	Licensing Information:	4
2.2	Intended Audience.....	4
2.3	Description and Main Features.....	4
3	Response to Community Need	5
4	High-level Architecture	5
4.1	Definitions	6
4.2	How the EOSC Monitoring service checks the status of a Service	8
5	Related Guidelines.....	8
6	Adopted Standards	8
7	Integration Options	9
7.1	Integration Option 1: Monitor an Onboarded Service	9
7.2	Integration Option 2: Monitor an Infrastructure.	10
7.3	Integration Option 3: Integrate External Monitoring service.....	11
	The monitoring team assists the Provider to create the necessary profiles:	11
	Supported monitoring Engine and Operating System (Nagios on Centos 7 or Debian 8):	11
	Other monitoring systems:.....	12
7.4	Integration Option 4: Combine Results of existing ARGO Tenants.	14
7.5	Integration Option 5: Third-party services exploiting EOSC Monitoring data	14
7.5.1	Example used:	15
	Example	16
	API call examples for status reports	16
	Detailed documentation: https://argoeu.github.io/api/v3/status/	16
	Example	16

1 Glossary

EOSC Future project Glossary is incorporated by reference: <https://wiki.eoscfuture.eu/x/JQCK>.

2 Introduction

2.1 Licensing Information:

For software licensing information, please visit: <https://github.com/ARGOeu/argo-monitoring/blob/master/LICENSE>

2.2 Intended Audience

Technical experts of service and resource providers that would like their services and/or resources to be interoperable or integrate with EOSC-Core Services.

2.3 Description and Main Features

Monitoring is the key service needed to gain insights into an infrastructure. It needs to be continuous and on-demand to quickly detect, correlate, and analyse data for a fast reaction to anomalous behaviour. The challenge of this type of monitoring is how to quickly identify and correlate problems before they affect end-users and ultimately the productivity of the organisation. Management teams can monitor the availability and reliability of the services from a high level view down to individual system metrics and monitor the conformance of multiple SLAs. The key functionalities offered by the EOSC Monitoring service are:

- Monitoring of:
 - EOSC-Core services
 - EOSC-Exchange Services (see Integration Options section)
- Reporting availability and reliability,
- Visualisation of the services status,
- Provide dashboard interfaces that can be target towards both Providers and End Users (e.g. Researchers),
- Sending real-time alerts to Providers (Service operators) and EOSC-Core service operators to varying levels of complexity (for example, for the purposes of alerting operators to availability issues, or to alert the EOSC Provider Onboarding Team of issues with Resource Profiles).

The dashboard design should enable easy access and visualisation of data for end-users. APIs are supported to allow third parties to gather monitoring data from the system .

The EOSC Monitoring service was designed to:

- Support multiple entry points (different types of systems can work together),
- Being easily interoperable with other monitoring systems,
- Operate the different components of the systems in High Availability,
- Support for Multiple Tenants, Configurations, Metrics and profiles to add flexibility and ease of customisation.

The EOSC Monitoring service combines two operational monitoring services: the EOSC-CORE and the EOSC-Exchange Monitoring Services, Respectively monitoring the EOSC-Core services (EOSC Core Monitoring) and the services onboarded to the Marketplace (EOSC-Exchange Monitoring).

The EOSC Monitoring services were implemented adopting the ARGO technology.

3 Response to Community Need

Two needs have been identified:

1. Services operated by Providers that do not have easy access to monitoring solutions, and would benefit from integrating with a federated monitoring service.
2. Providing Service metadata in a Resource Catalogue alone may not be sufficient to convince a user that the service is reliable and properly supported. Publishing a monitoring dashboard for a service can increase the confidence of a potential user of a service, that the service/resource is actively supported and available for use.

4 High-level Architecture

The EOSC Monitoring service collects status (metrics) results from one or more monitoring engine(s) deployed across distributed infrastructure and delivers daily and/or monthly availability (A) and reliability (R) results for monitored services. Status results and A/R metrics are presented through a Web UI, with the ability for a user to drill-down from the availability of a site to individual test results that contributed to the computed figure.

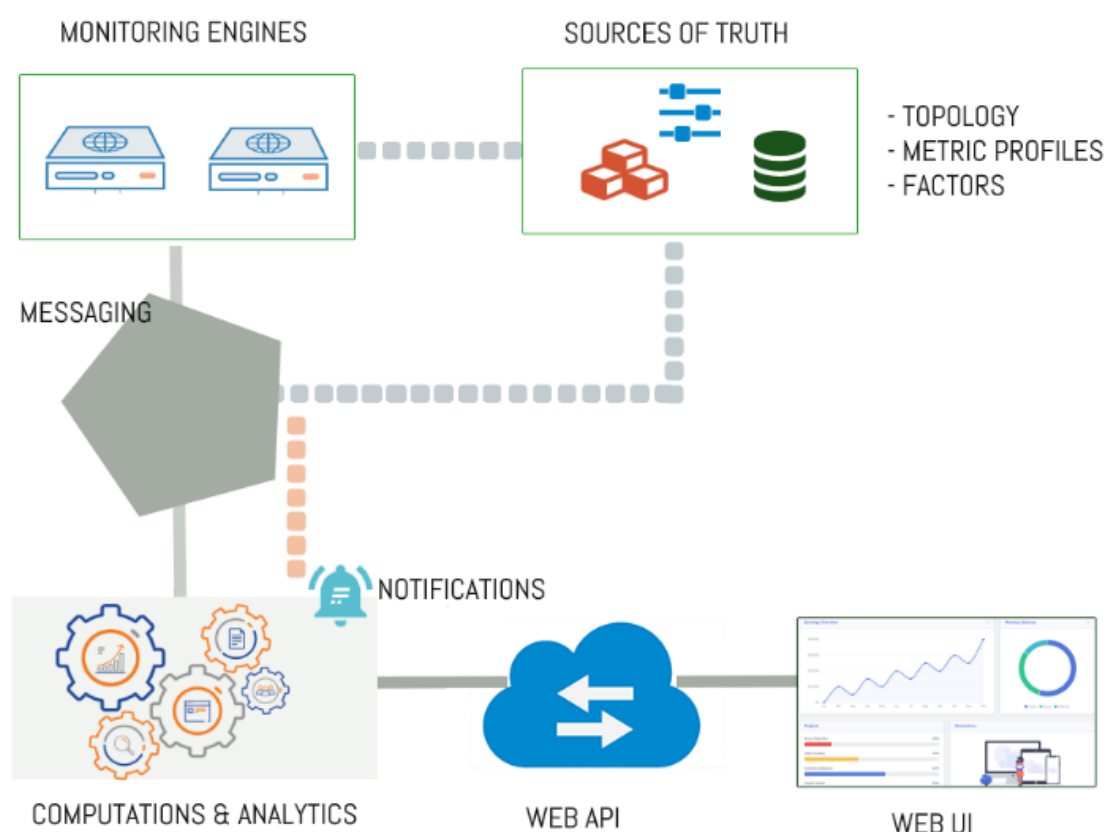


Figure 4 High level architecture of a Monitoring service

The main components of the EOSC Monitoring service are depicted in the high-level architecture diagram and described in Figure 1.

Monitoring Engine(s): this component executes the service checks against the distributed infrastructure and delivers the metric data (probe check results) to the Messaging Service.

Sources of Truth: The Monitoring system supports a number of connector plugins that are able to fetch topology, metrics and factors from various sources such as the CMDB and the EOSC Resource Catalogue. A Metric and Profile Management Component allows checks (probes) to be defined and associated with specific Services. Each combination of checks and service types forms a profile.

Messaging: The monitoring system uses a Pub/Sub Messaging Service to connect its components.

Computations & Analytics: Computational jobs are defined for ingesting data, calculating status and availability/reliability and a management service automatically configures, deploys and executes those jobs on a distributed processing engine for stateful computations. This component analyses the monitoring results and sends notifications based on a set of rules, to inform the Service Providers about the status of their services.

WEB API: RESTful HTTP API service that provides access to status and availability/reliability results. It supports token based authentication and authorisation with established roles. Results are provided in JSON Format.

WEB UI : The Web UI is the component to present the information about the status of the services. The global information from the primary and heterogeneous data sources is retrieved by means of the different plugins. The collected information is structured and organised within configuration files in the service and, finally, made available to the web application without the need for any further computations. This modular architecture is conceived in order to make it easy to add new data sources and to use cached information if a primary source is unavailable. The resulting data is exposed through a [RESTful](#) web service interface.

The following table defines key terms that help explain how the EOSC Monitoring service works and the types of information a Provider should supply/define to start to use the service.

4.1 Definitions

Tenant	<p>The tenant is an isolated instance of the EOSC Monitoring service that relies on common components and provides the user with its own environment.</p> <p>The EOSC Monitor provides default UI and POEM URLs in following form:</p> <ul style="list-style-type: none"> • UI: <a href="https://<tenant_name>.ui.argo.grnet.gr">https://<tenant_name>.ui.argo.grnet.gr • POEM: <a href="https://<tenant_name>.ui.argo.grnet.gr">https://<tenant_name>.ui.argo.grnet.gr <p>Custom URLs can also be used, in such cases the customer is responsible for providing valid certificates and DNS aliases.</p> <p>The EOSC Monitoring service requires following topology information in order to monitor services:</p> <ul style="list-style-type: none"> • the services and service endpoints they are running, • the way they are organised (e.g. groups of sites, groups of services),
---------------	---

	<ul style="list-style-type: none"> the service actors (owners, admins, contact points).
Topology	<p>EOSC Monitoring service requires the following topology information in order to monitor services:</p> <ul style="list-style-type: none"> the services and service endpoints they are running the way they are organised (e.g. groups of sites, groups of services), the service actors (owners, admins, contact points). <p>The topology can be further extended with attributes needed for individual probes (e.g. service port or URL, path to be used in case of storage services, e.g.).</p> <p>The topology sources currently supported are:</p> <ul style="list-style-type: none"> EOSC Resource Catalogue EGI Configuration Database (GOCDDB) EUDAT DPMT JSON feed in the predefined format.
Metric	<p>A metric is a small piece of software that checks specific functionality of a given service. For example a metric such as Portal-WebCheck runs on a site and checks if the HTTP connection responds correctly or not.</p>
Probe	<p>A probe is a small piece of software that implements single or multiple tests. The probe must comply with the guidelines for monitoring probes.</p>
Registry of probes and metrics (POEM)	<p>EOSC Monitoring Service provides a registry of probes and metrics. New probes and metrics can be added to the registry with the support of the EOSC Monitoring team.</p>
Metric Profile	<p>A Metric Profile is used to associate a Service with the corresponding metrics.</p>
Aggregation Profile	<p>An Aggregation Profile defines how to aggregate service statuses into higher hierarchical grouping (i.e. a service_group) status results. They are actually used to define logical rules on how to aggregate individual service status computations into groups.</p>

4.2 How the EOSC Monitoring service checks the status of a Service

In order to evaluate the operational state of the Service, all or part of the metrics that check the service's functionality should be taken into account. The Metrics Profiles encompass, for each Service, all the metrics whose results are included in the calculation of the Service's state. For example, an example service "WebSite" running on host1.example.com should be considered properly operating if it is accessible and some actions are available such as downloading or uploading material (documents, images etc). In this case three metrics might be used to check the service's functionalities:

- *Portal-WebCheck* is a metric to check if the http service responds;
- *http.download* is a metric to check if download functionality operates well;
- *http.upload* is a metric to check if upload functionality operates well;

The Service is assumed to operate properly if it is accessible and can support downloading material. Uploading material does not affect the state of the service (whether it is working properly or not). So in the Metrics Profile, the metrics Portal-WebCheck and http.download will be defined in order to be taken into account for concluding the status of the service.

5 Related Guidelines

The following table presents two perspectives of the EOSC Monitoring Interoperability Guidelines, i.e., a description of the standards, protocols, APIs, Guidelines, etc, to which compliance is recommended or required in order to interoperate with or integrate EOSC Monitoring, and also provides a description of the material standards, protocols, APIs, Guidelines that are adopted by the EOSC Monitoring Service.

ResourceType	Title	Short Description	relatedIdentifier	relationType
Guideline	ARGO Guidelines for Monitoring Probes	Probes must comply with the guidelines for monitoring probes.	https://argoeu.github.io/argo-monitoring/docs/Monitoring/guidelines/	Requires

6 Adopted Standards

Standard	Short Description	References
REST	All the APIs exposed by the EOSC Monitoring Service follow the REST standard.	https://www.ics.uci.edu/~fielding/pubs/

SAML2	Used to authenticate/authorise users in the Web interfaces.	https://wiki.oasis-open.org/security/FrontPage
X509	Used by metrics/probes to authenticate with the service to be tested. It is also used to authenticate in some web interfaces (legacy)	https://www.rfc-editor.org/info/rfc5280
Apache Avro	Used to encapsulate monitoring data as they are transferred between its internal components.	http://avro.apache.org/
HTTPS	All Web UIs of the EOSC Monitoring Service are exposed via the HTTPS Protocol	https://tools.ietf.org/html/rfc2818

7 Integration Options

EOSC Providers can choose one of five approaches to integrate their Services, with the EOSC Monitoring service:

1. Monitor an Onboarded Service: monitor a single EOSC Service;
2. Monitor an Infrastructure: monitor a complete infrastructure supporting multiple Services and Resources;
3. Integrate External Monitoring service: configure the EOSC Monitoring service to accept monitoring data from third-party monitoring engines;
4. Combine Results of existing ARGO Tenants: allow to combine the topology and the results of multiple tenants in a number of reports;
5. Third-party services exploiting EOSC Monitoring data: a customer retrieves results from the EOSC Monitoring Service to use them in an external service/dashboard.

These are described below. Details are available in the EOSC Future wiki.

7.1 Integration Option 1: Monitor an Onboarded Service

This option covers the scenario to monitor one EOSC. The results of this process will become available via the EOSC-Exchange Monitoring WebUI.

After a service has been successfully onboarded to the EOSC, its provider can enable the EOSC Monitoring service by supplying some extra information.

First, the monitoring service requires the probes and metrics to be associated with the service. This can be enabled by the Service Provider provider by visiting their EOSC Providers Dashboard and

selecting the Resource (Service) that needs to be monitored. The Provider should select “Add Monitoring Extension” and enter the following Information:

- Select which organisation’s monitoring service is to be enabled for your Service from the drop down list. If none is enabled yet or you wish that the EOSC Monitoring Service is used select “EOSC” otherwise select the appropriate one.
- Select the Service Type that matches the technology used by your Service For example, eu.eosc.generic.http represents generic service speaking HTTP protocol.
- Add the URL of the Service Endpoint to be Monitored (e.g. <https://argo.eosc-portal.eu>)

Once all the information has been provided, the monitoring of the service starts and the EOSC Monitoring Computation and Analytics component calculates availability and reliability of the service and creates a report. The Service Provider can review A/R and status results from the EOSC-Exchange Monitoring WebUI.

7.2 Integration Option 2: Monitor an Infrastructure.

This option covers the scenario when an infrastructure (e.g. an e-infrastructure) with multiple services and a custom topology needs to be monitored by the EOSC Monitoring. In this case, the first integration option is not appropriate because the infrastructure provider needs to take additional steps:

- define a custom topology and of the way in which monitored endpoints will be aggregated for reporting purposes;
- select from existing range of probes and adding custom ones;
- manage profiles and metrics for different services.

This integration option requires the following steps:

The Provider (typically its infrastructure manager) must request the creation of a dedicated EOSC Monitoring service instance. This can be done by opening a ticket in the EOSC Helpdesk where the following minimum information should be provided:

- Infrastructure topology
- Personnel responsible for managing profiles
- URLs for the registry of probes and metrics (POEM) and UI components

The EOSC Monitoring team will use the provided information to create a new instance (tenant) of the Monitoring Service, within the EOSC Monitoring Infrastructure, and inform the infrastructure manager that the instance is ready for use.

As a next step, the infrastructure manager must define the minimum set of profiles to allow the monitoring to start:

- Selection of a list of metrics from the metric repository
- Definition of the Metric Profile
- Definition of the Aggregation Profile

Once all the information has been provided, the monitoring of the service starts and the EOSC Monitoring Computation and Analytics component calculates availability and reliability of the service and creates a report. The Infrastructure Manager can have a look at the A/R and status results from the dedicated UI.

7.3 Integration Option 3: Integrate External Monitoring service.

In order to be able to scale-out and take advantage of existing Monitoring systems, the EOSC Monitoring service is capable of accepting data from external sources. When referring to external sources we mean other monitoring engines that want to connect with the EOSC Monitoring Service.

This integration option covers the case when a service or an infrastructure provider is already operating its own monitoring system and is willing to publish information about the status of its service(s) in EOSC to, for example, demonstrate their reliability.

The connection of a third-party monitoring system with EOSC Monitoring is mainly based on the necessary data to create the final monitoring report. In this use case an external monitoring system replaces the internal monitoring engine and is thus **reliable for the validity of the monitoring data that is published**.

This integration option requires the following steps:

The Provider opens a ticket on EOSC Helpdesk requesting to start the process to connect to the EOSC Monitoring Service. They need to prepare their systems to be able to share the following information:

- The type of system used;
- Infrastructure topology;
- Personnel responsible for managing the necessary profiles;
- URLs for the registry of probes and metrics (POEM) and UI components.

The monitoring team creates a new instance/tenant in the EOSC Monitoring service and sets up all the necessary configuration on the EOSC Messaging service. As a result, the monitoring team will then send to the Provider the necessary instructions and the access tokens to connect to the EOSC Monitoring Service.

The monitoring team assists the Provider to create the necessary profiles:

- Metric Profile
- Aggregation Profile.

The Provider will need to make the necessary configuration on their monitoring engine in order to start publishing metric data via the EOSC messaging service. The EOSC Monitoring Service supports two options:

Supported monitoring Engine and Operating System (Nagios on Centos 7 or Debian 8):

if the Providers use Nagios as its monitoring tool, EOSC Monitoring offers the argo-nagios-ams-publisher tool that is currently supported on Centos-7 and Debian-8. argo-nagios-ams-publisher is a component acting as a bridge from Nagios to the EOSC Messaging system and finally to the EOSC Monitoring Engine. It is responsible for forming and dispatching messages that wrap up results from the monitoring engine. In order to use the this solution the customer will need to :

1. Install argo-nagios-ams-publisher and ams-library
2. Configure argo-nagios-ams-publisher
3. Enable OSCP in Nagios:

In /etc/nagios/nagios.cfg add this configuration

```
obsess_over_services=1
ocsp_command=argo_service_check
ocsp_timeout=15
```

1. Add OSCP command:

should add an OSCP command in /etc/nagios/objects/commands.cfg

```
define command {
    command_name argo_service_check
    command_line /usr/bin/ams-metric-to-queue --queue /var/spool/argo-nagios-ams-
publisher/metrics/ --hostname "$HOSTNAME$" --status "$SERVICESTATE$" --summary
"$SERVICEOUTPUT$" --message "$LONGSERVICEOUTPUT$" --servicestatetype
"$SERVICESTATETYPE$" --actual_data "$SERVICEPERFDATA$" --service
"$_SERVICESERVICE_FLAVOUR$" --metric "$_SERVICEMETRIC_NAME$"
}
```

1. All the Services to be published must have following attributes set:

```
define service {

    use          generic-service; Name of service template
to use
    host_name      grnet.gr
    service_description HTTP
    check_command   check_http
    check_interval  5
    _service_flavour WebPortal //the service
    _metric_name    org.nagios.WebCheck
}
```

1. Start argo-nagios-ams-publisher by executing

```
service ams-publisherd start
```

Other monitoring systems:

In this case the client cannot or doesn't want to use the solution described in the case 3.1 . Then the external monitoring system should find a way to send the monitoring data (metric data) to the EOSC Monitoring . These data should follow a predefined format.

The data should be stamped with their source and timestamp. Every metric should be prefixed with [source_type], following the metric naming best practises. Every metric is also labelled with the

hostname and service description. These predefined messages should be sent to the EOSC Messaging service which is the service responsible to pass them to the computations engine which performs the necessary calculations to produce the reports.

```
{
  "hostname": "host101.example.com",
  "service": "eu.eosc.portal.services.url",
  "metric": "org.nagios.WebCheck",
  "timestamp": "2022-01-02T00:24:38Z",
  "status": "OK",
  "tags": {
    "endpoint_group": "GroupA"
  },
  "summary": "200 OK",
  "actual_data": "time=0.085796s;;;0.000000 size=1126B;;;0",
  "monitoring_host": "monbox.example.com", //name of the external monitoring
  box
  "message": "a more detailed message about the monitoring result"
}
```

Metric data comes in the form of avro files, (json files support currently in development) and contains timestamped status information about the hostname, service and specific checks (metrics) that are being monitored. A typical item of information in the metric data contains the field listed in the table below.

```
{ "namespace": "argo.avro", //currently this type is supported.
  "type": "record",
  "name": "metric_data",
  "fields": [
    { "name": "timestamp", "type": "string"},
    { "name": "service", "type": "string"},
    { "name": "hostname", "type": "string"},
    { "name": "metric", "type": "string"},
    { "name": "status", "type": "string"},
    { "name": "monitoring_host", "type": ["null", "string"]},
    { "name": "summary", "type": ["null", "string"]},
    { "name": "message", "type": ["null", "string"]},
    { "name": "tags", "type": ["null", { "name": "Tags",
      "type": "map",
      "values": ["null", "string"]
    }]
  }
}
```

The monitoring team will validate the published metric data against the supplied topology and perform a number of dry runs to ensure that there is no issue with the supplied data. As soon as the metric data is validated by the Monitoring Team these will be the main data to compute A/R and status results.

After the previous steps are completed, the monitoring of the service starts and the EOSC monitoring Computation and Analytics component calculates availability and reliability of the service, and creates the monitoring report. The Infrastructure Manager can have a look at the A/R and status results from the dedicated UI.

7.4 Integration Option 4: Combine Results of existing ARGO Tenants.

This integration option covers the scenarios where the topology and the results of multiple monitoring instances/tenants need to be combined in a number of reports. It allows the creation of a monitoring report including services coming from multiple infrastructures like, for example, when a research community is using services from more e-infrastructures. Through this option, a research community is able to create a unique monitoring report including all the services it is using regardless of who is operating them.

As a prerequisite, in order to combine results from tenants, e.g. A and B, those tenants should be already monitored by EOSC Monitoring service:

- **Latest Data available:** Each tenant should be checked that has an active stream of incoming monitoring data.
- **Topology:** Each tenant should already have a well defined source of topology that includes lists of groups, endpoints and services.
- **Metric Profile:** In simple terms, a list of all services to be checked along with all relevant metrics per service

This integration option can be enabled by following these steps:

The Provider should submit a request on the EOSC helpdesk describing:

- Tenants to be used in the combined report;
- Services and metrics;
- Aggregation profile.

The monitoring team creates a new tenant that will host the combined report. This tenant acts as a host tenant for the combined results and will rely on the data of the other tenants as input for the computations of the availability, reliability and status results.

After the previous steps are completed, the monitoring of the service starts and the EOSC Monitoring Computation and Analytics component calculates availability and reliability of the services, and creates a report. The User can have a look at the A/R and status results from the combined reports from the UI.

7.5 Integration Option 5: Third-party services exploiting EOSC Monitoring data

This option covers the scenario according to which the Provider needs to use the results of the EOSC Monitoring Service in an external service/dashboard.

The customer can access the following information via an API:

- A/R information about the service and its service components;
- Status information about the service and its service components;
- The topology and grouping of the service.

This integration option can be enabled by following these steps:

The user who wants to gain access to this type of monitoring information will get a token with read-only access to the A/R and status results. The user via the EOSC helpdesk may send his request to the monitoring team by sending:

- The name of the service that requires the information ;
- An email to create the user able to access to the monitoring information;
- The type of information (A/R results, status results, or both)

The monitoring team will provide the required token and information, guidance on how to retrieve the information.

7.5.1 Example used:

In this example we are going to present how the user can get the availability, the reliability values and the status of the AMS (Messaging Service) (endpoint: <https://msg.argo.grnet.gr>) of the Organisation GRNET.

The Monitoring Service Monitoring Service is checking the services at regular intervals. It actually runs explicit tests (checks) in order to assess the status of the service. The result of the checks decides on the status of the service. In order to display status information it uses reports where it keeps all the necessary information.

At the same time it produces useful conclusions about the monitoring item via the monitoring analytics engine. One very useful conclusion is to decide if the item is available for usage and if it is considered as reliable. To succeed this, availability/reliability values (hourly, daily, monthly) are calculated. These different types of information are also encapsulated in a report.

The EOSC monitoring service monitors the Messaging Service and it performs the following checks

- `cert_validity_check` : a metric that checks the validity of the certificate used by the service
- `ams_check`: a metric that checks a list of functionalities provided by the messaging service.

Based on the explanation provided above, the information about the service follows:

Definition	Value	Description
GROUP	GRNET	A collection of services
SERVICE	AMS	The type of one of the services of the collection

SERVICE endpoint	msg.argo.grnet.gr (AMS)	is defined as the combination of a hostname and Service Type. (a Service Type of AMS listening on port/s <ams-port/s> on the host msg.argo.grnet.gr is a service endpoint)
Grouping used in the report	SERVICEGROUPS	the way the services are organized (e.g. in groups of sites, in groups of services) in the monitoring engine
A/R report	Default	The place where the A/R results are provided.
Status report	Default	The place where status results are provided.

This is the configuration that the user will have to use to use the api calls.

API call examples for A/R reports

The api authenticates the user using the api-key within the x-api-key header. Users can specify time granularity (monthly or daily) for retrieved results and also format using the Accept header. Depending on the form of the request the user can request a group, service or service endpoint.

Detailed documentation: <https://argoeu.github.io/api/v3/results/>

Example

For the AMS the corresponding api call to get the A/R of the service group GRNET is:

Request for A/R results for service group GRNET

```
$ curl -X GET -H "Accept: application/json" -H "Content-Type: application/json" -H "x-api-key: secret-token"
https://api.argo.grnet.gr/api/v3/results/Default/SERVICEGROUPS/GRNET?start\_time=2021-08-05T00:00:00Z&end\_time=2021-08-05T23:59:59Z
```

API call examples for status reports

The api authenticates the user using the api-key within the x-api-key header. Users can specify time granularity (monthly or daily) for retrieved results and also format using the Accept header. Depending on the form of the request the user can request a group, service or service endpoint.

Detailed documentation: <https://argoeu.github.io/api/v3/status/>

Example

For the AMS the corresponding api call to get the status of the service group GRNET is:

Request for status results for service group GRNET

```
$ curl -X GET -H "Accept: application/json" -H "Content-Type: application/json" -H "x-api-key: secret-token"  
https://api.argo.grnet.gr/api/v3/status/Default/SERVICEGROUPS/GRNET?start\_time=2021-08-05T00:00:00Z&end\_time=2021-08-05T23:59:59Z
```