Argo Roadmap

Kostas Koumantaros (GRNET)
& the ARGO Team.
Outline

1. Introduction
2. Latest updates
3. Interoperability and Integration options
4. Next Steps
Monitoring What’s New in 2020

1. Centralised Sources of Truth.
2. Updated POEM to use Web-Api as backend
3. Added Support for Custom Defined Thresholds.
4. Started Publishing actual data.
5. Added Support for flat Topologies
6. Updated UI with more details.
7. Worked on 3 Interoperability Use Cases.
One stop shop
One Stop Shop

Implement a service management web interface through which customers (e.g. VO managers, Infrastructure Managers, thematic services etc) will be able to configure the monitoring service to their liking through just a few clicks.
sources of truth (Before)
Centralized sources of truth
Poem Connections (Before)

ARGO

ARGO Web-API

ARGO Connectors

ARGO Mon

COMPUTE ENGINE

GOCDB

DPMT

CMDB - X

AMS
Updated POEM to use Web-API as backend
React Poem
Customer Defined thresholds
Customer Defined thresholds

- Status and A/R reports based on the metric results
  - Each metric result includes a status and a performance data field that typically contains values related to the provided status.
- Each probe has a hard-coded built-in static logic in order to compute the probe status.
- **BUT** we need flexibility in providing different SLA targets to customers.

**Solution**

A mechanism for each customer to define its own thresholds.

**in progress ...**
Status

- **Probes**: expose the actual data (**done**)
- **Monitoring engine**: Passes actual data to messaging (**done**)
- **Compute engine**: supports customer define thresholds (**done**)
- **Web API**: create and manage profiles (**done**)
- **UI**: manage and display thresholds to users/customers. (**missing**)
### Endpoints Support

#### Availability/Reliability Table

<table>
<thead>
<tr>
<th>Month</th>
<th>2020-07</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Av</td>
</tr>
<tr>
<td>ce94.grid.arm.dz</td>
<td>93.7</td>
</tr>
</tbody>
</table>

Showing 1 to 1 of 1 entries

#### Availability/Reliability Charts

Status for egl tenant - Critical Report

![Graph showing availability and reliability](graph.png)
Metrics Support

Select your metric

<table>
<thead>
<tr>
<th>Metric</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>auvergrid</td>
<td>OK</td>
</tr>
<tr>
<td>BG01.IPP</td>
<td>OK</td>
</tr>
<tr>
<td>budapest</td>
<td>OK</td>
</tr>
<tr>
<td>ca-victoria-westgrid-t2</td>
<td>OK</td>
</tr>
<tr>
<td>camik</td>
<td>OK</td>
</tr>
<tr>
<td>creatis-insa-lyon</td>
<td>OK</td>
</tr>
<tr>
<td>grif</td>
<td>OK</td>
</tr>
<tr>
<td>goeGrid</td>
<td>OK</td>
</tr>
</tbody>
</table>

Report: Critical
Metric: eu.egi.CREAM-IGTF
51 groups - 72 checks

Filter by status:
Documentation

Status

- Introduction
- Status Page: Landing page
- How is the status computed?

Availabilities/Reliabilities

- Availability/Reliability
- Availability/Reliability Table
- Daily Availability/Reliability Table
- Availability/Reliability Charts
- Other Functionalities

Introduction

This is the page where you may see the status for the whole infrastructure.

Before we start some definitions about the services, services are composed of service instances of a specific service type. Each instance has a set of metrics, which we call availability.

Status Page: landing page

The first information you can see is about the groups you have.

Availabilities/Reliabilities

- Availability/Reliability
- Availability/Reliability Table
- Daily Availability/Reliability Table
- Availability/Reliability Charts
- Other Functionalities

Availability/Reliability

Availability: Service Availability is the fraction of time a service was in the UP Period during the reporting period.

Reliability: Service Reliability is the ratio of the time interval a service was UP over the total reporting period.

From this page you can see the latest values for monthly reports for A/R for your infrastructure service and the grouping of the services.

The report may contain A/R values based on the group you chose in the Configuration Manager.

- Sites: List of services that participate in the site
- Project: A list of services that are used in a project.

Availability/Reliability Table

This is a table with the main information. The Availability and Reliability values for the last 4 months are shown.

If you want to learn more about the daily availability or Reliability values of a specific month, you can click on the date.

If you want to learn more about the services or the endpoints of the services you can click on the row.

Dashboard

- Introduction
- Last status checks
- Downtimes

Introduction

This page is a summary view for your monitoring data and a given report (see. Critical) The main information consists of:

- The description of the topology structure (project, sites) and list of the related services
- The results of availabilities/relabilities for the last 30 days
- The failures status chart (more information below)
- The last status change (more information below)
- The failures affecting the service (more information below)
Interoperability guidelines

use case 1
Combine Results of infrastructures in EOSC in a unified UI.

use case 2
Add a Service Provider/Infrastructure to EOSC Monitoring

use case 3
Third-party services exploiting EOSC Monitoring data

EOSC Monitoring

web api

Research Infrastructure

service

service

service
Use Case 1: Combined Results of one or more infrastructures in EOSC in a unified UI.

The general goal of “distributed monitoring” is to allow different infrastructures with the same or different environment to scale. There are a number of different options for supporting this. All of them are based on the concept that **different sources will publish status and performance data in a predefined form** that is read from the core engine of the monitoring service. The data should also be **stamped with their source and timestamp**.
Use Case 2: Add a Service Provider/Infrastructure to EOSC Monitoring

In order to add support for a new Service Provider or Infrastructure in the EOSC Monitoring Service, the provider should only need to provide the **topology of the services to be monitored and the equivalent metric and aggregation profiles**. The system should take care of all the actions required to probe every endpoint in the topology with the metrics/probes defined, and aggregate the results according to the profiles defined and present them in a Web-UI.
Use Case 3: External Usage

Use Case 3: Third-party services exploiting EOSC Monitoring data

Any EOSC service **should be able to retrieve and use the status information and metrics computed by the EOSC Monitoring system.** An API is provided to allow any authorised third-party service to retrieve such data. This API returns:

1. Endpoint, service and service group
2. Raw data and performance data
3. And the data for the sources of truth
Next Steps

ARGO Should:

- Be extended to become an aggregator for monitoring data from other Argo instances or similar systems
- Be able to accept different types of raw data (e.g. logs)
- Be able to discover Service Trends
- Provide Status Pages for Users
- Enable the creation of personalized dashboards in UI
- Investigate Docker Support
- Support open standards (e.g. OpenMetrics.io)
Become an aggregator for monitoring data

- Extend the Compute Engine to become an aggregator of Monitoring data.
- So that it can incorporate Raw Data, Results and Topologies from external sources and include them in its calculations of Status and A/R.
- Also support different types of Data (e.g. Logs, analytics)
Discover Service Trends

Extend compute engine to run additional computations and analytics on data for e.g. discover service trends

The ARGO monitoring engine gathers monitoring data (metric results) for all the services.

to exploit services usage patterns and provide a number of trends.

status behaviour of services, provide ranks

Services with resource problems
Cater for cloud Native Environments

be able to quickly monitor and handle simple url endpoints that might

● expose lightweight portals
● serverless endpoints
● microservices
● staticgen websites
● Ephemeral endpoints.
Status Pages

All the services and their functionalities are properly monitored by the user perspective and owners of the service are alerted when a problem appears.

**BUT**

the end-user is NOT
Status Pages for Users

- Build Community Users trust
- Give some simple building blocks to the user:
  - a/r table elements
  - status timelines
  - custom header/footer
  - basic branding/theme (logos, colors, fonts etc)
- Quickly host it either in argo or in other premises
Support for Open Metrics

Metrics is the main way to monitor services.

- Naming Conventions
- Packaging

Everyone is following its own way

An emerging effort to create an open standard for transmitting metrics at scale, with support for both text representation and Protocol Buffers.

Open Metrics is a Cloud Native Computing Foundation project