ARGO Monitoring System

by GRNET, SRCE and CNRS

Kostas Koumantaros, GRNET

ARGO Service Monitoring

A Flexible & Scalable Framework

- Provides reports for the **status**, **availability** and **reliability** of services
- Can generate **multiple reports** using different profiles (e.g. for management, operations etc)
- Supports **flexible deployment** models
- Can leverage **external information providers** (such as CMDBs, Service Catalogs etc)
- Can take into account **custom factors** during the report generation (e.g. the importance of a service endpoint, scheduled or unscheduled downtimes)
- Supports **customer defined algorithms** for report generation
- Based on **open source** components

Status, Availability & Reliability

ARGO Service Monitoring

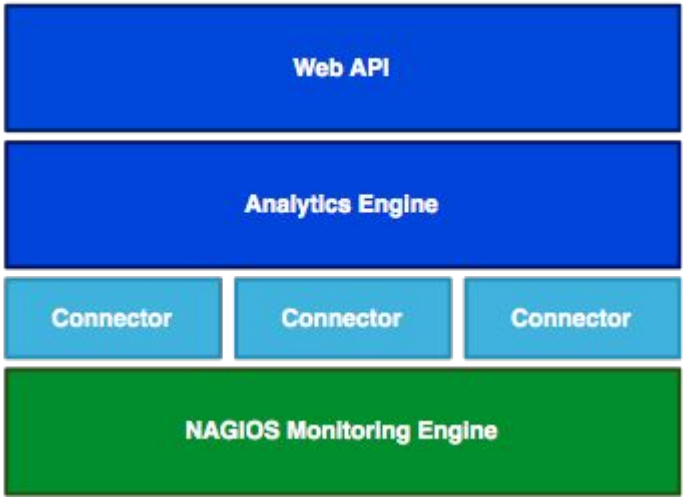
Status. Service Monitoring

For status monitoring, ARGO relies currently on Nagios. All probes developed for ARGO follow the Nagios conventions and ARGO can use any stock Nagios probes.

ARGO provides an **optional set of addons** for the stock Nagios that provide features such as auto-configuration from external information sources, publishing results to a an external messaging service etc

Status, Availability & Reliability

ARGO Service Monitoring



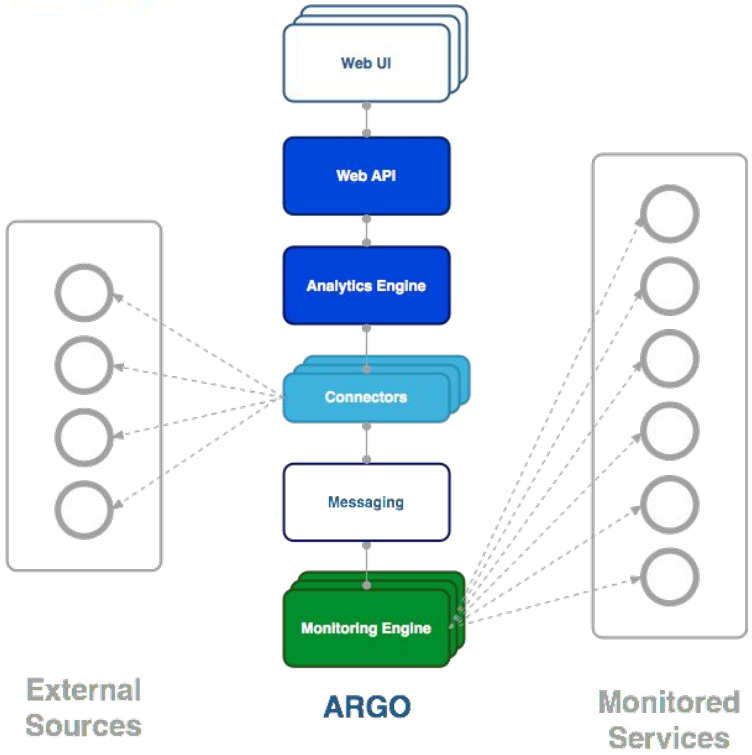
Availability & Reliability. Service Monitoring

For Availability & Reliability monitoring ARGO, introduces a modular architecture, which relies on the Monitoring Engine for service endpoint monitoring and which can ingest the monitoring results in order to **track monitoring metrics**, provide real-time **notifications** and **status reports** and **monitor SLAs/OLAs**

ARGO comes in two flavors: **A standalone version** for deployment in low density e-Infrastructures with a limited number of services and **a cluster version** for deployment in high density e-Infrastructures with a large number of services.

Modular Architecture

ARGO Service Monitoring



ARGO Components. Modular Architecture

At its core, ARGO uses a **flexible monitoring engine**, a **powerful analytics engine** and a **high performance web API**.

Embracing a **modular, pluggable architecture**, ARGO can easily support a **wide range of e-Infrastructures**.

Through the use of **custom connectors**, ARGO can connect to multiple external **Configuration Management Databases** and **Service Catalogs**.

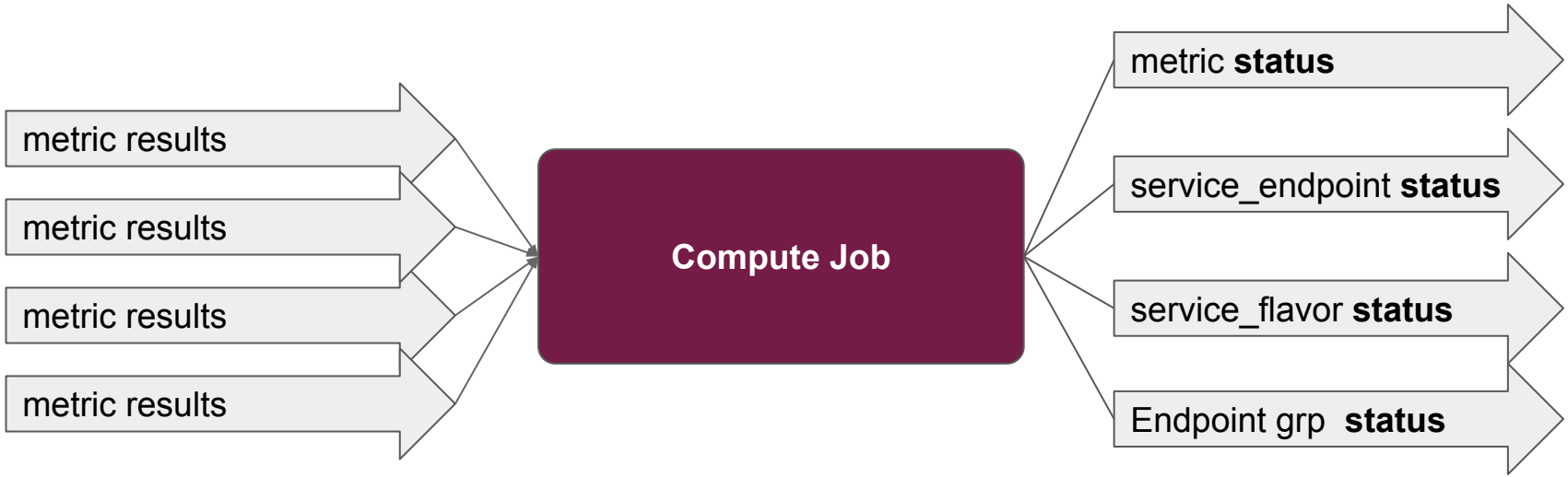
What are the main changes?

- Streaming engine
- Messaging Service
- Notification

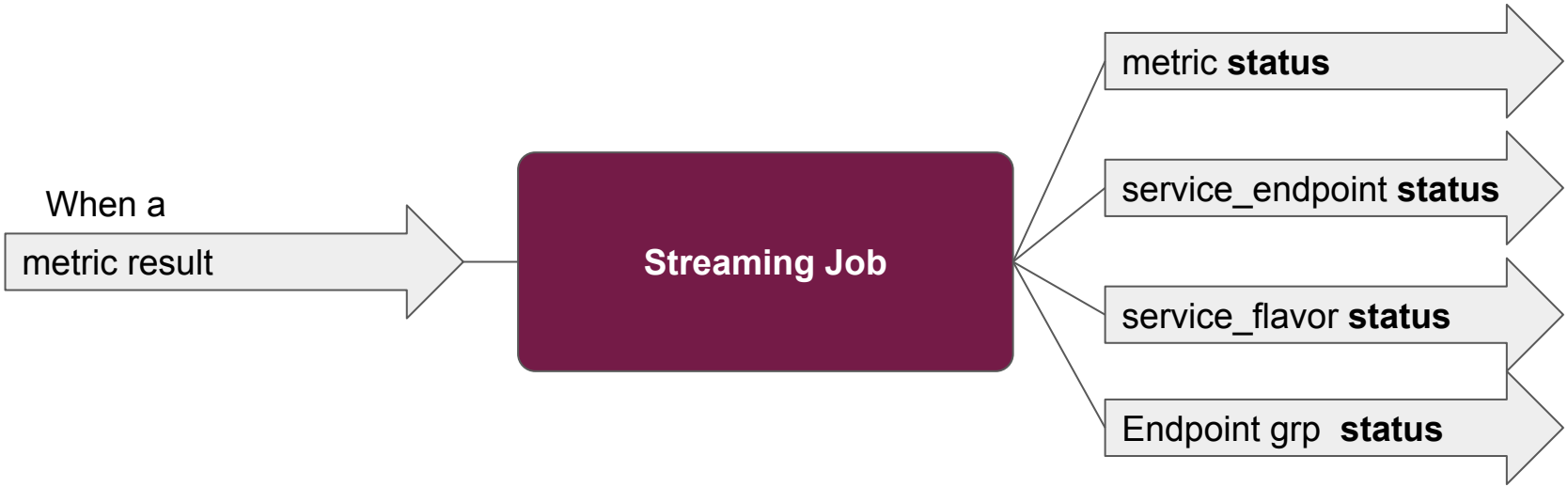
Streaming Engine

Today ....

- Metric results arrive in batches
- Every hour a job is running and status changes

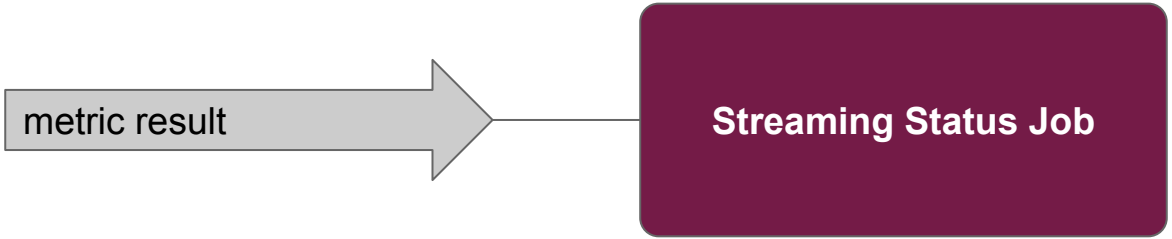


Streaming job i/o



Streaming Status Job

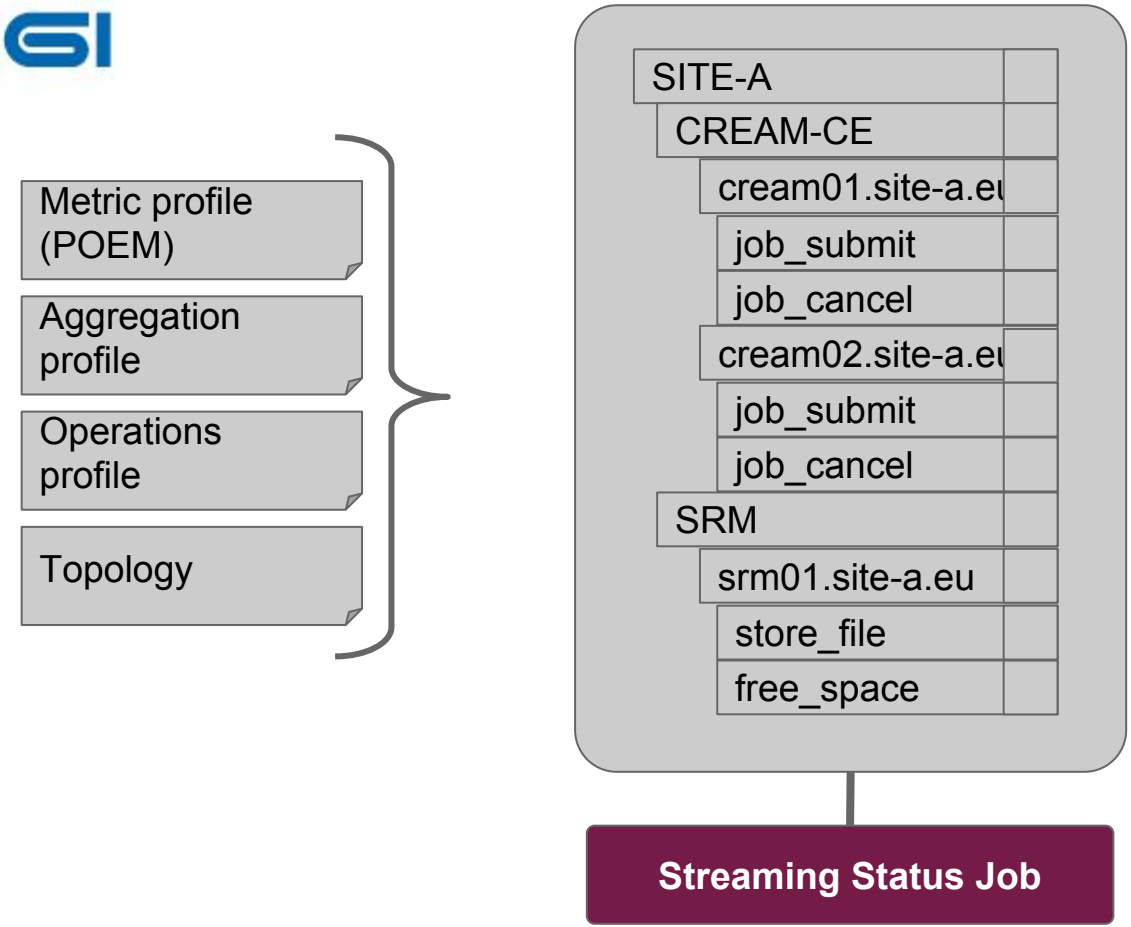
- Each metric result arrives and job should query and find what it affects:
  - Which service_endpoint (cream01.auth.gr)
  - Which service_flavor (CREAM-CE)
  - Which endpoint_group (HG-03-Okeanos)



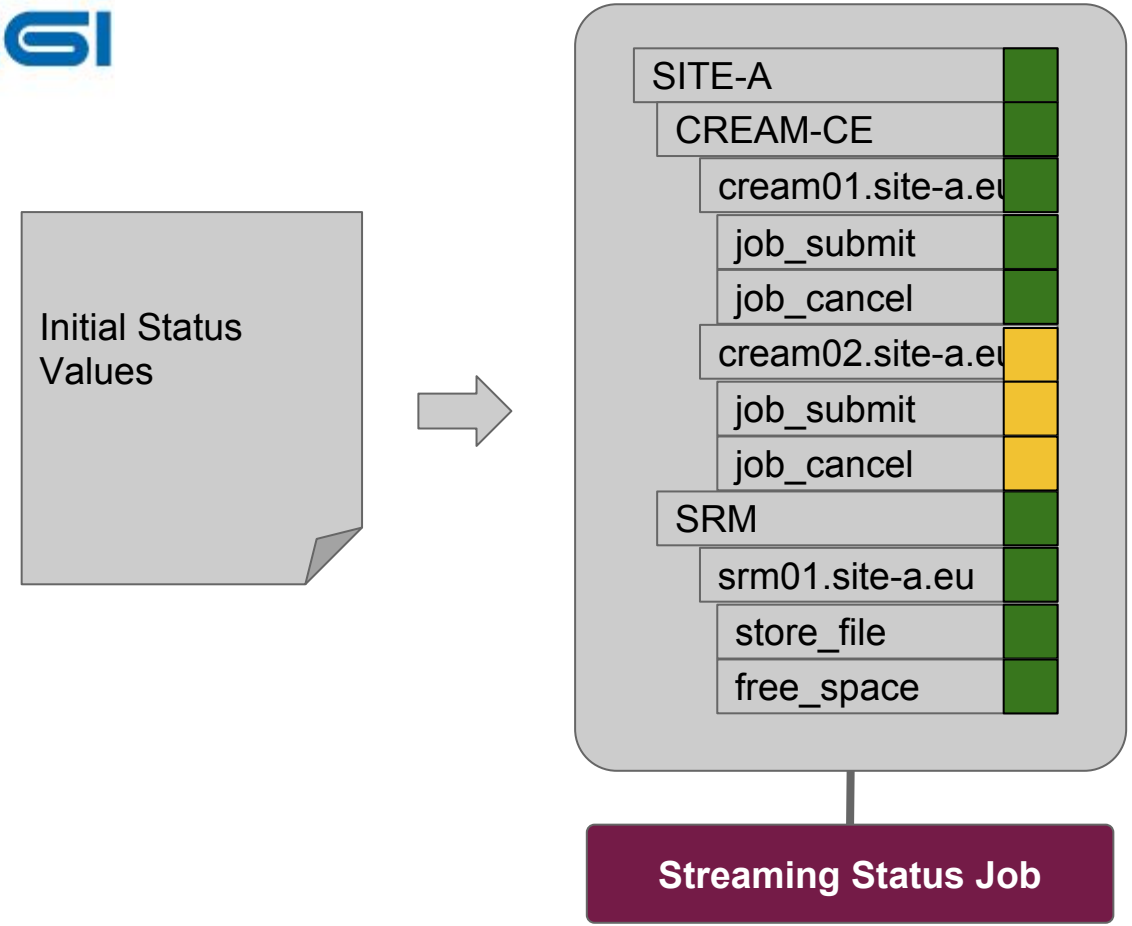
A real example - Job start

Streaming Status Job

A real example - Create structure

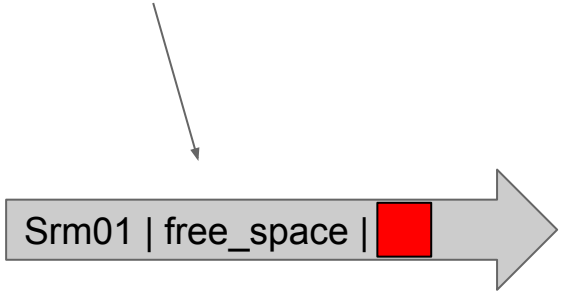


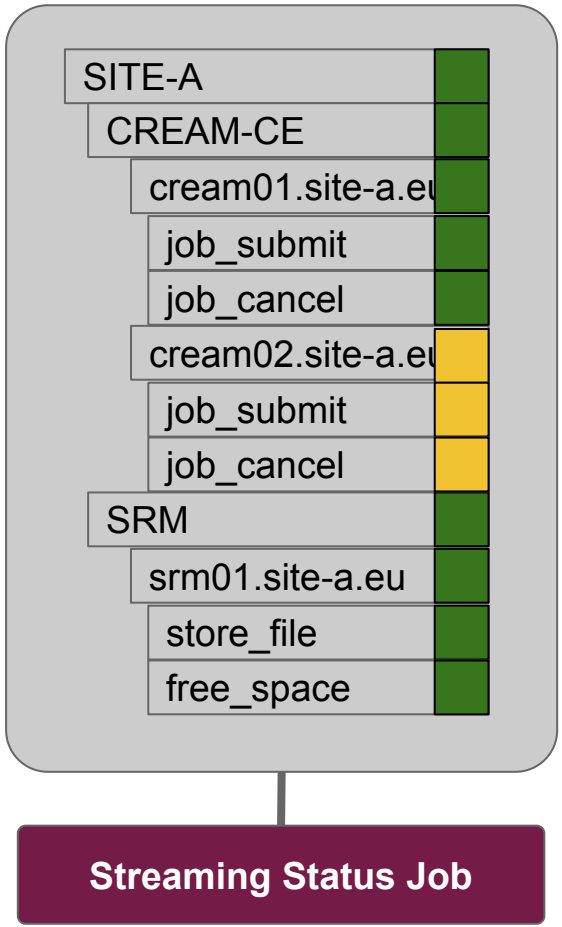
A real example - Load initial values



A real example - First metric arrives

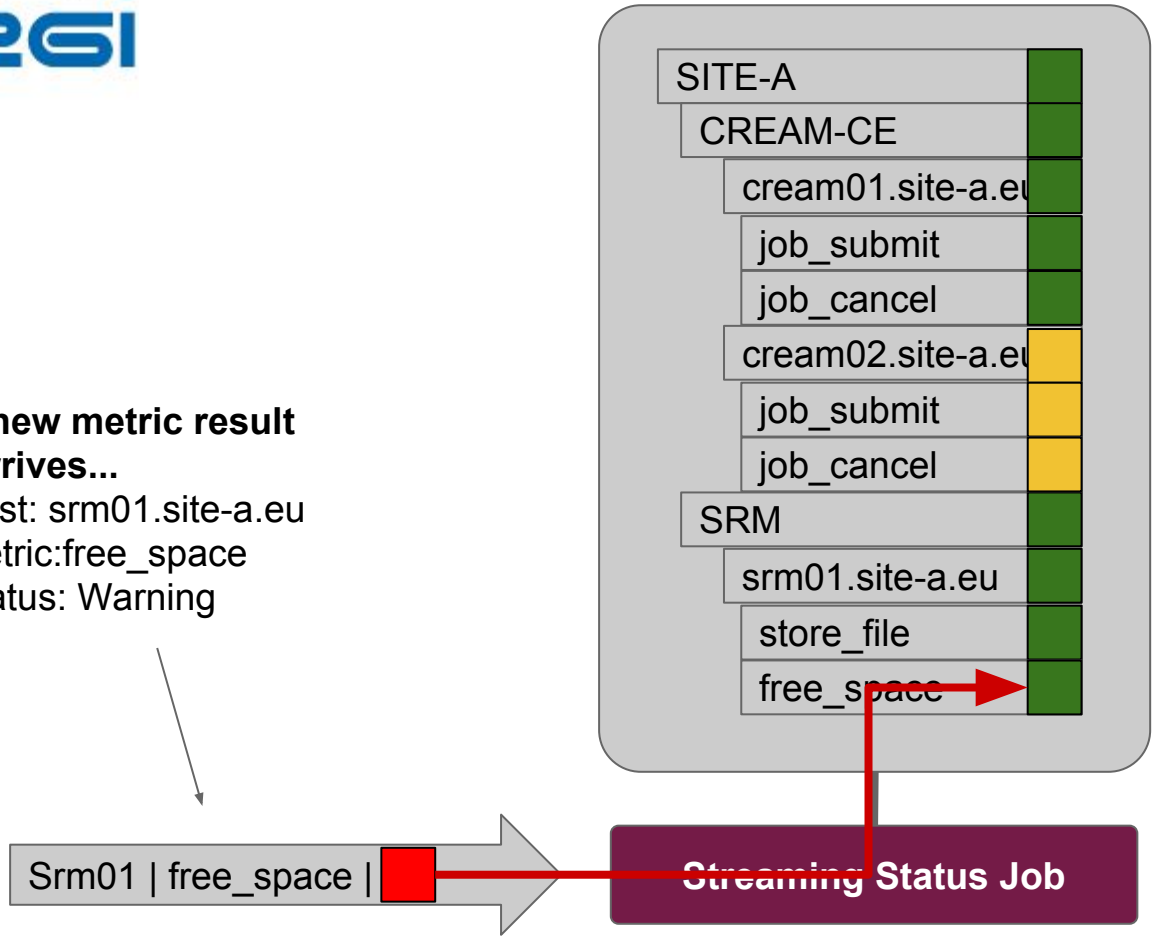
A new metric result arrrives...
Host: srm01.site-a.eu
Metric:free_space
Status: Warning



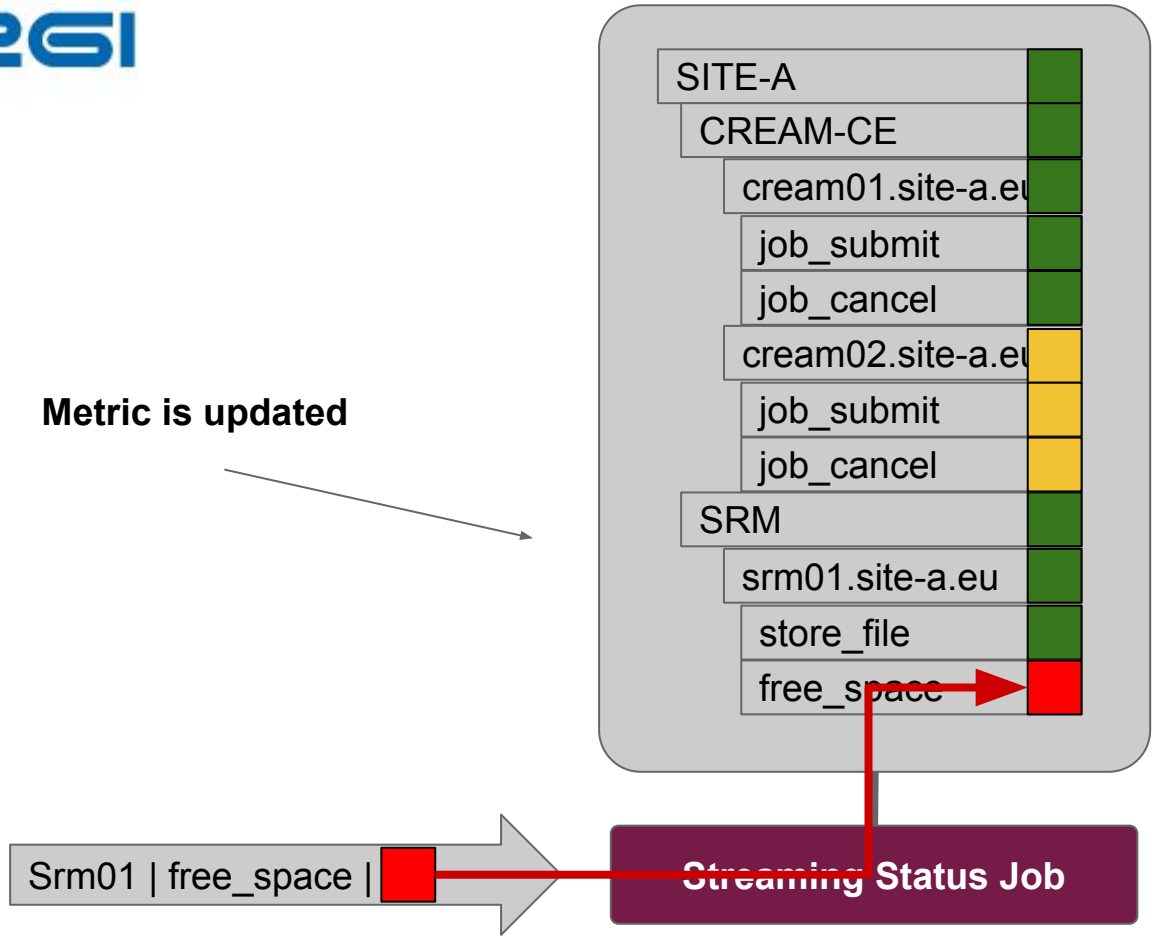


A real example - First metric arrives

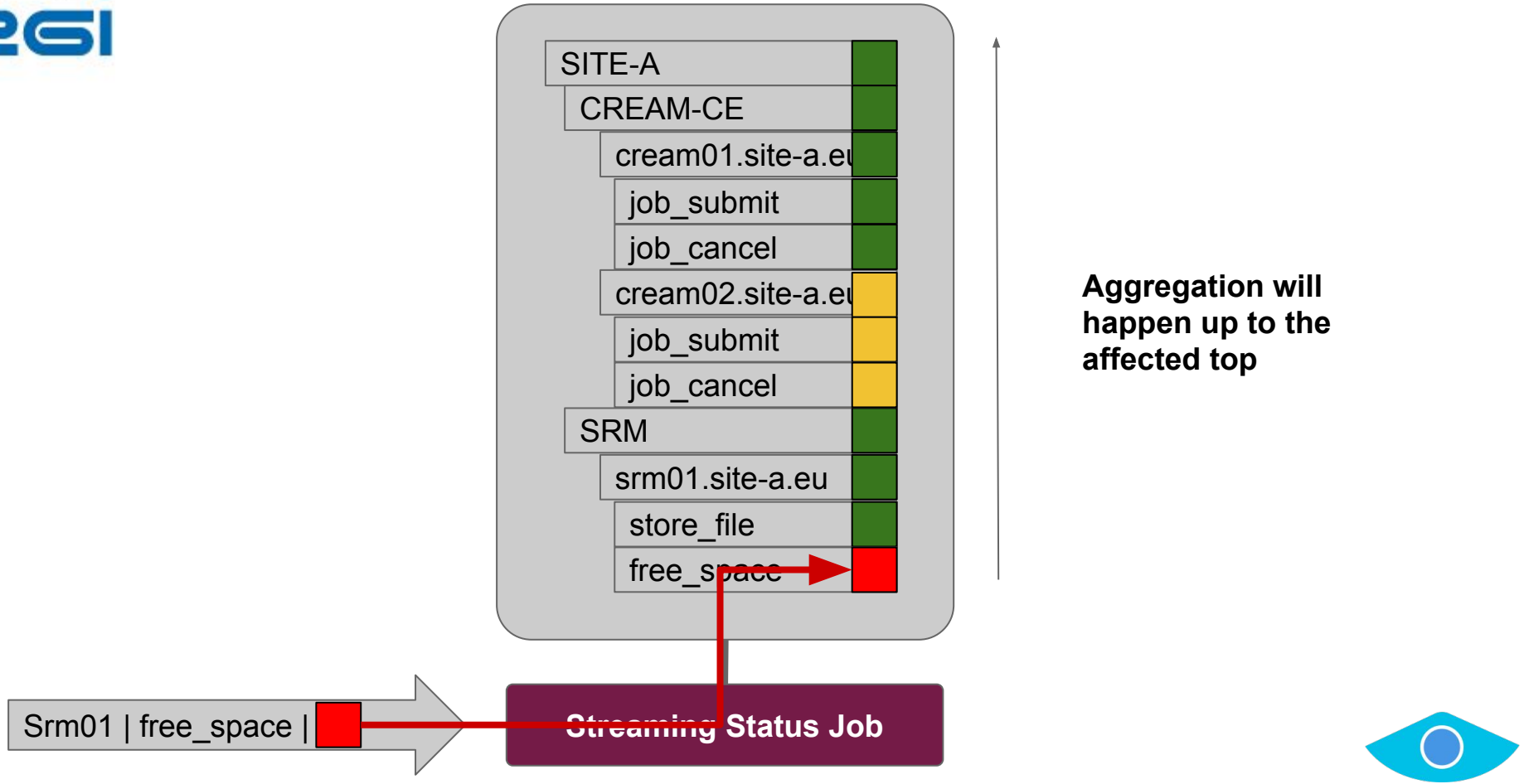
A new metric result arrrives...
Host: srm01.site-a.eu
Metric:free_space
Status: Warning



A real example - metric_update

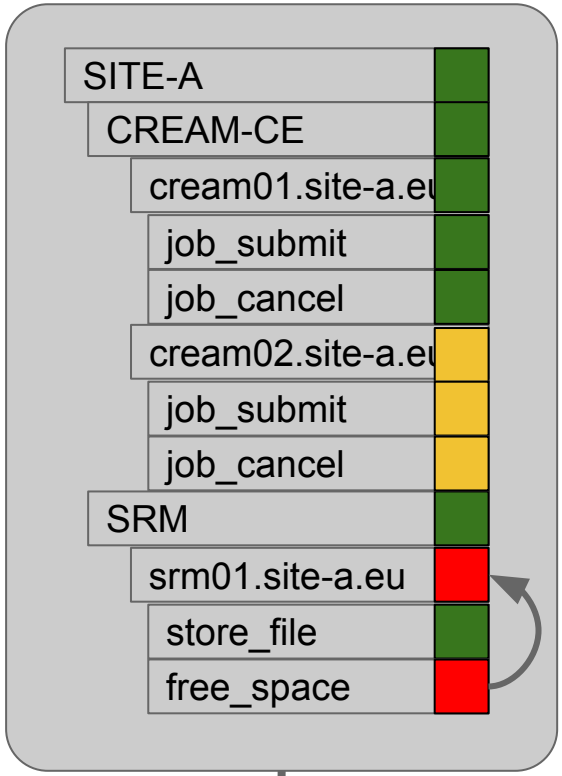


A real example - metric_update



Aggregation will happen up to the affected top

Aggregate metrics to service_endpoint

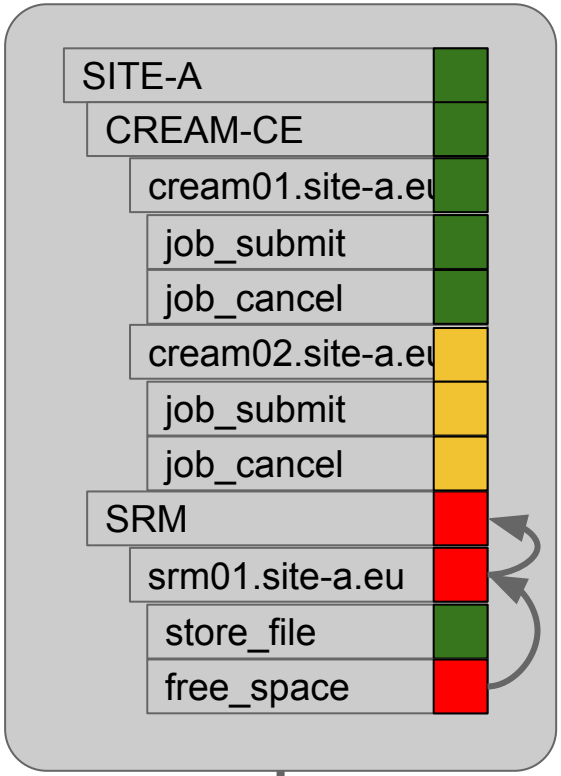


Aggregation will happen up to the affected top

Srm01 | free_space |

Streaming Status Job

Aggregate service_endpoint to service_flavor

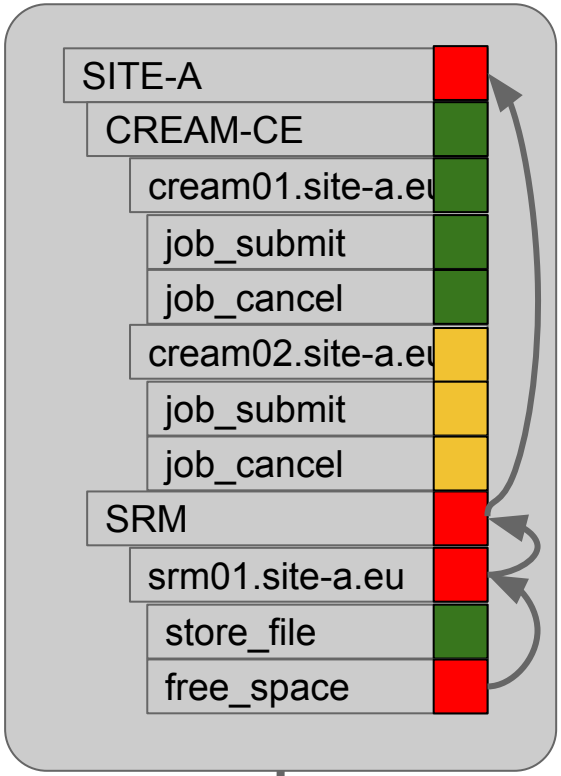


Aggregation will happen up to the affected top

Srm01 | free_space |

Streaming Status Job

Aggregate service_flavor

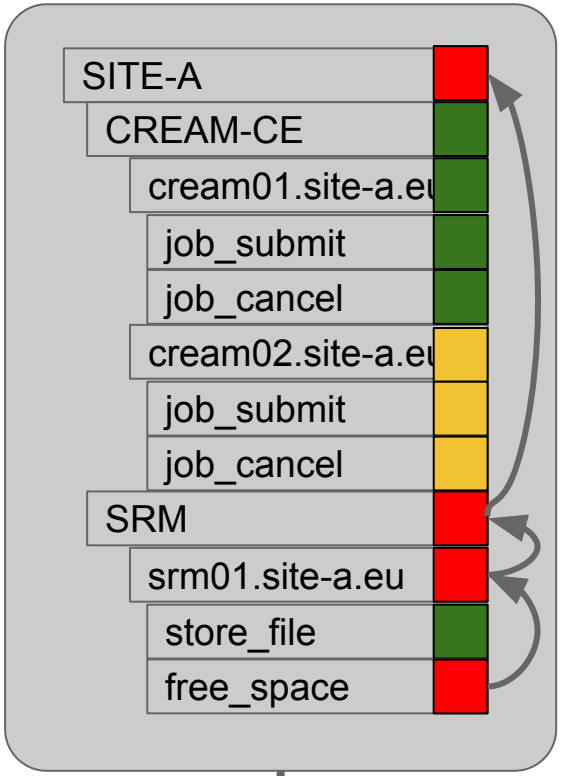


Aggregation will happen up to the affected top

Srm01 | free_space |

Streaming Status Job

Identify changes and produce events



SITE-A

SITE-A | SRM

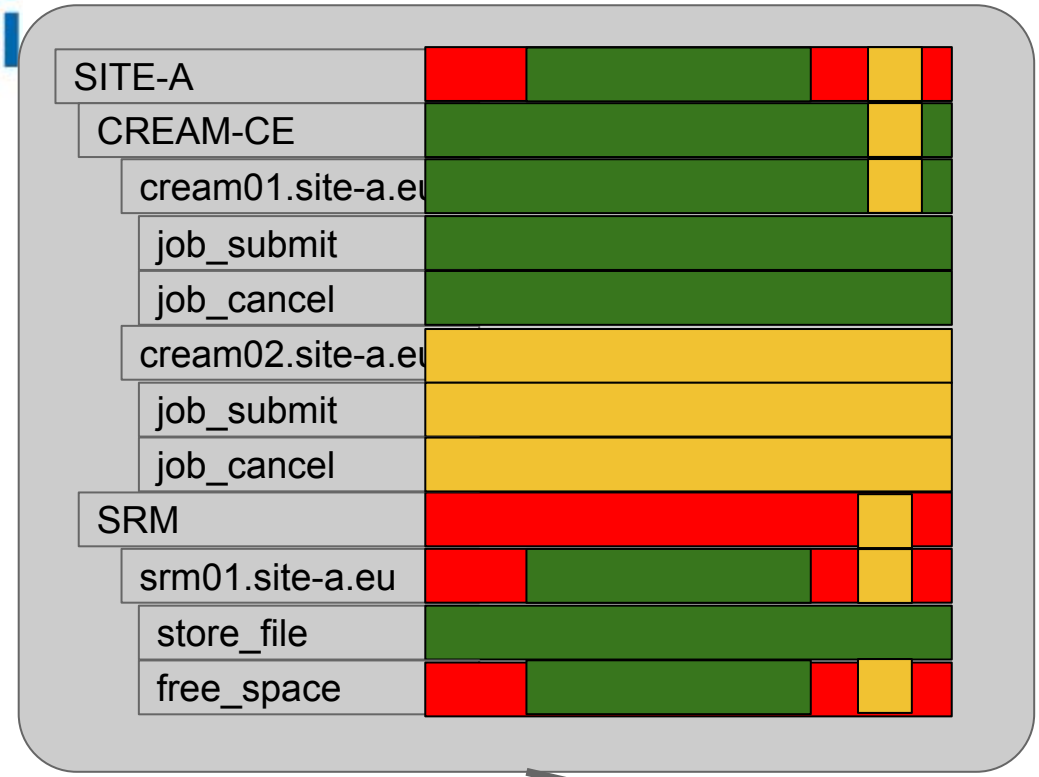
ssrm01.site-a.eu

Srm01 | free_space |

Srm01 | free_space |

Streaming Status Job

Batch Job



Daily status results

Entire daily metrica data

Batch Job

The result

- Near Real time status results

Messaging Service

GRNET ActiveMQ Messaging Service

- Existing Messaging Service is based on ActiveMQ
  - Multi-national ActiveMQ Message Broker Network
- Service is stable
  - Transporting ~1 Billion message per year
- **Used by infrastructure services** mainly
- Very low number of incidents and **always on the client side**
- The Message Broker Network is **a complex service**
  - Coordination of upgrades is a demanding activity
  - Changes to the stable configuration are very time consuming (e.g. support for ACLs)

ARGO Messaging Service

Fundamentals

Project a basis of organizing and isolating groups of users & resources

Resource Types

Topic
A named resource to which messages are sent by publishers.

Subscription
A named resource representing the stream of messages from a single, specific topic

Message
The combination of data and attributes that a publisher sends to a topic and is delivered to subscribers.

Users & Roles

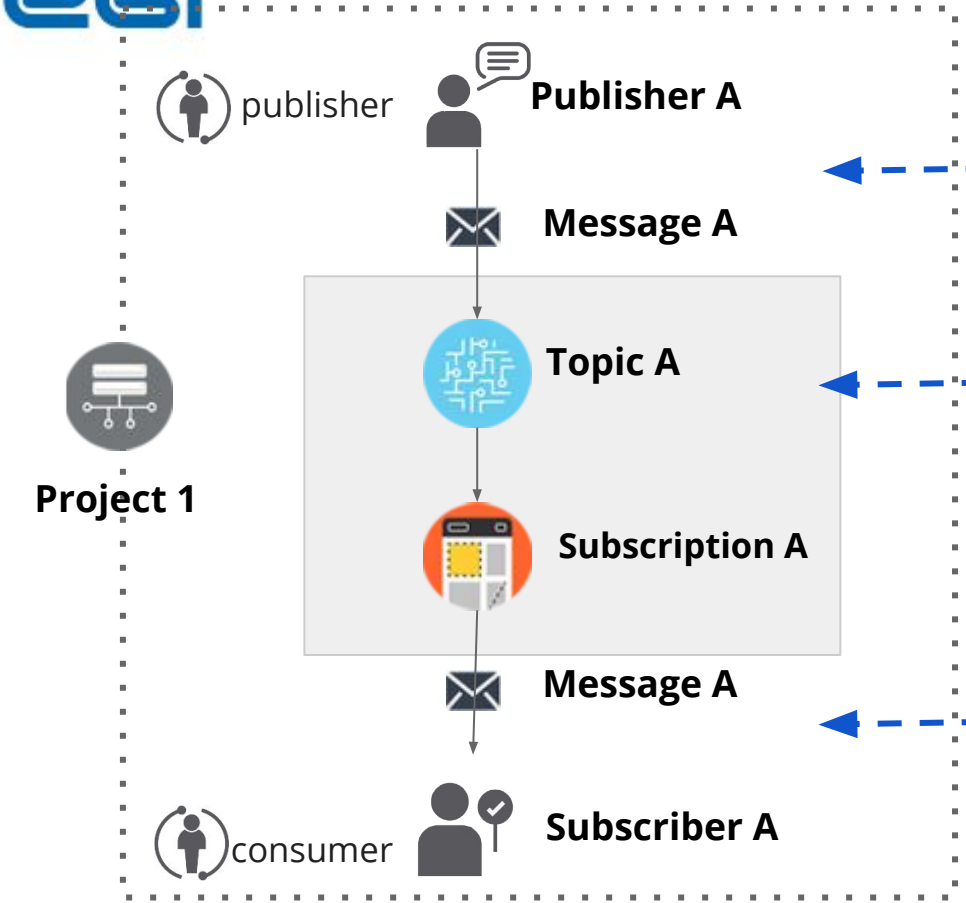
Roles
A list of service-wide roles. Can be extended. (action is a permission)

Publisher
Client that publishes msgs to a topic.

Subscriber
Client that consumes msgs from a topic.

ARGO Messaging Service

Flow



Publisher sends ***messages*** to a topic .

Messages are persistently stored in the *messaging service* until they are delivered to subscribers. The service forwards messages from a topic to all of its subscriptions.

Subscriber receives pending messages from the subscription. Subscriber sends an acknowledgement that he processed the message.

When the message is acknowledged, it is deleted from the AMS.

Status

- Deploy in Production - Q2
- Integration in progress (in devel infrastructure) with
  - ARGO MON
  - AppDB
  - Operations Portal
- A python library for easy integration with AMS

Notifications

Status

- Proposal sent to EGI operations
- Emails to be sent to site & service contacts
- Enable/disable alerts in GOCDB

ARGO Monitoring

Developments

- Opsmon report
  - opsmon.egi.eu decommissioned & tests migrated to central instances (6.4.2017)
  - Report filtered for EGI.eu sites
- Biomed monitoring
  - Agreed to deploy new instance at SRCE
  - biomed to provide GOCDB-equivalent topology feed

Probes in testing

- Tests in progress
  - WebDAV
  - Swift
  - OCCI image list
  - EGI CheckIn
  - NGI Argus

Monitoring uncertified sites

- ARGO MON instance
  - https://argo-mon-uncert.cro-ngi.hr/nagios
- Instructions for sites
  - https://wiki.egi.eu/wiki/HOWTO21
- Report on development web UI
  - http://web-egi-devel.argo.grnet.gr/lavoisier/status_report-site?report=CriticalUncert&accept=html

FedCloud Report

- FedCloud report
  - report based on profile ARGO_MON_CRITICAL
  - filtering sites based on FedCloud tag

Roadmap

Roadmap (1/2)

Task description	Start	End
ARGO Compute Engine & Web API • Streaming processing • Separation of A/R and metric store • Notification System • Adaptation of the Web API to use HBASE • stability and performance improvements	12/2016	06/2017
ARGO Monitoring Engine • Finalize support for GOCDB as a single support of topology • Integration with probe management feature of POEM • AMS integration	12/2016	07/2017
ARGO EGI Web UI • Stability and performance improvements		08/2017
ARGO EGI Connectors & Consumer • Use of AMS for Connectors • Decommission of Consumer and use of AMS • stability and performance improvements	02/2017	06/2017
ARGO POEM • Finalize the probe management feature • Connect to the EGI IdP/SP Proxy • stability and performance improvements	05/2017	07/2017

Roadmap (2/2)

Task description	Start	End
ARGO Messaging • Use of ARGO Messaging production infra by the Monitoring Engine 2017Q2 - [IN PROGRESS] • Use of ARGO Messaging service by the APEL team 2017Q2 - [IN PROGRESS] • Message Service Accounting: Metrics for Messaging Service 2017Q2 • Operational statistics 2017Q2	10/2016	06/2017