Table of Contents

Multi Level Monitoring Overview

Introduction

The EGEE SA1 Multi-Level Monitoring (MLM) system is an operational monitoring system built to meet the needs of EGEE during the EGEE-III project. It is based on existing operational monitoring tools such as SAM and Gridview. The new system is enhanced by the addition of Nagios as a way to schedule and execute site tests. It also uses the MSG messaging system to integrate other operational tools with Nagios instances.

This document gives an overview of:

- the main components within the multi-level monitoring system
- an understanding of how these various components interact
- the deployment model, showing where the various components will be deployed
- the teams involved in the development of the system

Aims and Objectives

The aim of the multi-level monitoring system is to provide an integrated project-level monitoring system for EGEE-III. The model is based on concepts outlined in the EGEE-III Operations Automation Strategy. It should also fit within the principles of future federated infrastructures such as EGI.org.

It enables regional entities, such as a ROC in EGEE-III or a NGI in EGI.org, to take on the responsibility for the smooth running of the sites within the region. It provides the facility to monitor these sites remotely in order to detect operational problems which could affect users. It also provides the tools (e.g. regional operational dashboards and debugging facilities) to enable the ROC to aid sites to follow up problems.

The data produced by these regional monitoring systems is also used by project-level systems to carry out tasks such as SLA calculations.

Sites also deploy a local monitoring system. We have a reference, recommended, implementation based upon the standard components of MLM, e.g. Nagios, Metric Description DB. The aim of this local monitoring system is to improve site reliability by giving site administrators better visibility of the state of grid services at their site.

Components in multi-level monitoring

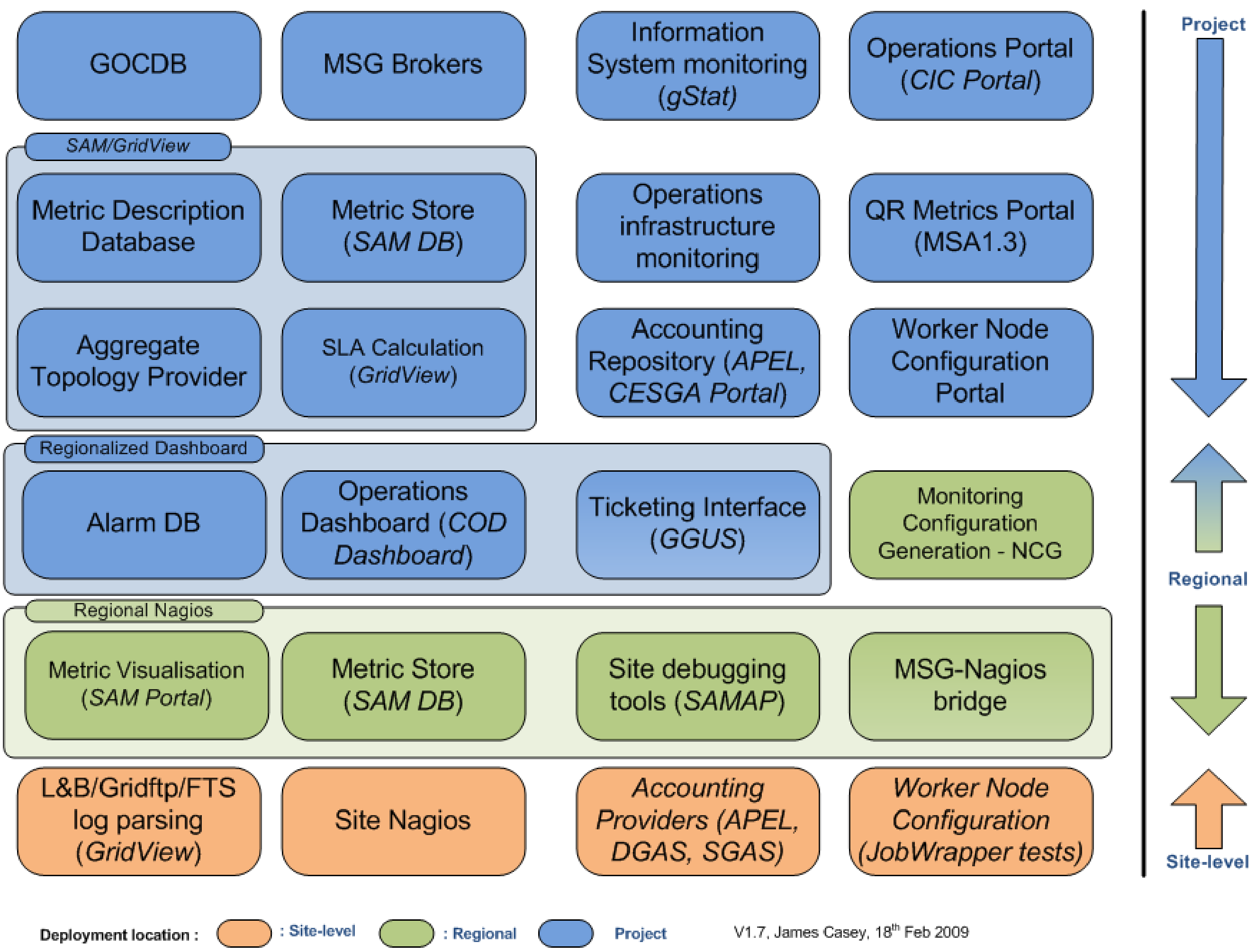
Technology Choices

Nagios, **Django** and **MSG** are the two underlying commodity technologies used in the system:

- **MSG** is used whenever two distributed components wish to communicate in an asyncronous manner.
- **Django**, a python based development framework, is used in the Metric Description DB, Aggregated Topology Provider, Worker Node configuration system, Metric Results Store and the WLCG Topology Provider
- **Nagios** is used extensively: in site montoring; regional monitoring; gStat; and project and regional Metric Stores

The diagram below shows the various components in the system, according to their deployment scope - project; regional; site.

Components in multi-level monitoring by deployment location*


GOCDB
MSG Brokers
Information System monitoring (gStat)
Operations Portal (CIC Portal)
SAM/GridView
Metric Description Database
Metric Store (SAM DB)
Operations infrastructure monitoring
QR Metrics Portal (MSA1.3)
Aggregate Topology Provider
SLA Calculation (GridView)
Accounting Repository (APEL, CESGA Portal)
Worker Node Configuration Portal
Regionalized Dashboard
Alarm DB
Operations Dashboard (COD Dashboard)
Ticketing Interface (GGUS)
Monitoring Configuration Generation - NCG
Regional Nagios
Metric Visualisation (SAM Portal)
Metric Store (SAM DB)
Site debugging tools (SAMAP)
MSG-Nagios bridge
L&B/Gridftp/FTS log parsing (GridView)
Site Nagios
Accounting Providers (APEL, DGAS, SGAS)
Worker Node Configuration (JobWrapper tests)
Project
Regional
Site-level
Deployment location :
: Site-level
: Regional
Project
V1.7, James Casey, 18th Feb 2009

* Items in (..) refer to similar components that exist and are deployed centrally at this moment.

We will now outline the roles and responsibilities of each of these components.

Repositories of Information

In order to configure the monitoring system we need two different types of information:

- A list of sites and services which should be tested
- The metrics that should be obtained for each of the services.

The definitive source of information for these two types of information are the **GOCDB** and **Metric Description Database**

GOCDB

- **Provider:** STFC
- **Deployment Model:** Project-Level
- **Description:** A project-level database which contains the definitive list of sites belonging to the EGEE infrastructure.
- **Dependencies:** None
- **Responsibilities:**
    - Contains personal contact information for staff at sites
    - Contains list of roles for project members

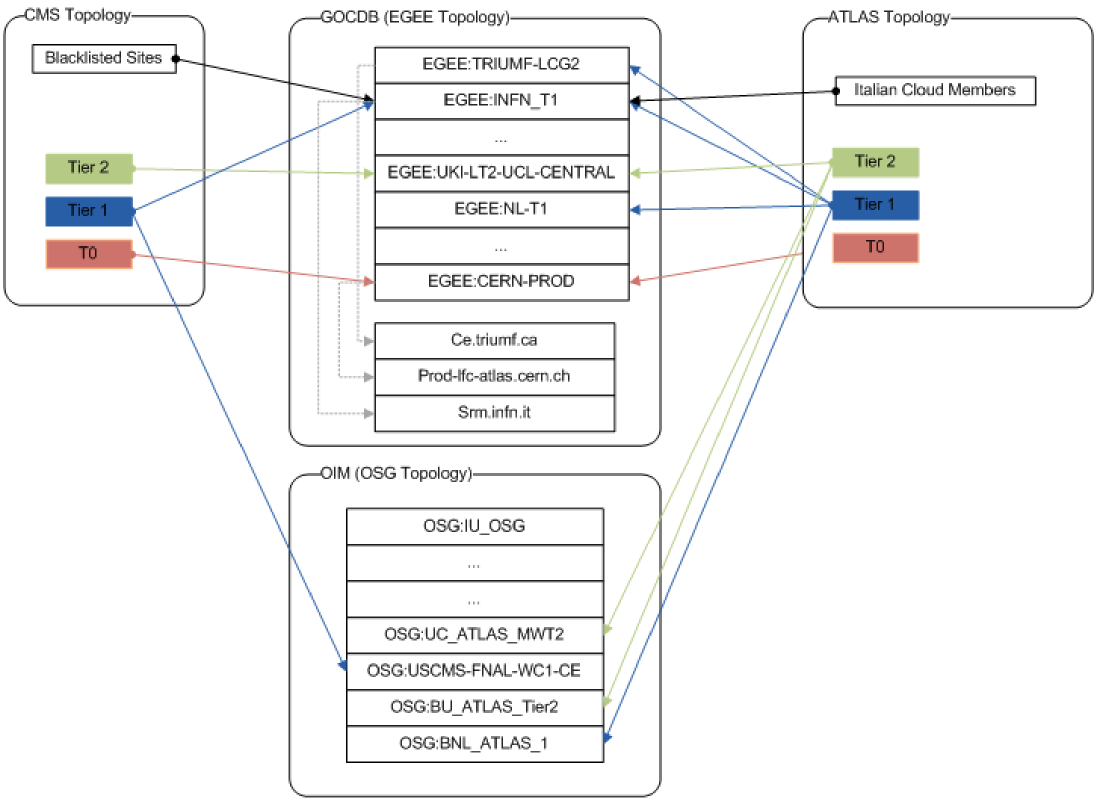
    - Contains list of downtimes and monitorable services
- **Interfaces provided:**
    - Web interface for site and regional administrators to create and update information
    - Programmatic query interface for other operational tools and VO tools to export/query information
- **Comments:**
- **Links and Documentation:** https://wiki.egi.eu/wiki/GOCDB/PI/Technical_Documentation

Metric Description Database

- **Provider:** BARC/CERN
- **Deployment Model:** Project-Level
- **Description:** A project-level database which contains the metadata about metrics that can be used to test services, and also how these metrics can be combined to calculate availability or generate blacklists.
- **Dependencies:**
    - GOCDB contains the list of services types
    - GOCDB contains the lists of roles for users which limit which actions they may take
- **Responsibilities:**
    - Provide descriptions of tests which should be run against grid services at EGEE sites.
    - Allow customization of tests that should be run.
    - Provide a list of calculations based on the tests that can then be used for e.g. availability, blacklisting.
    - Allow project members to provide their own calculations
    - Maintains history of which metrics and calculations were valid at which point in time
- **Interfaces provided:**
    - Web interface to add new metrics, metricsets and calculations
    - Programmatic interface for NCG and SLA calculation system to query and obtain the list of metrics to be tested for each service type
    - Programmatic interface for the SLA calculation system to obtain the current set of VO-specific SLAs that should be calculated
- **Comments:**
- **Links and Documentation:** Design Docs

Aggregated Topology Provider (ATP)

Projects such as WLCG need access to a wider range of topology sources than just the GOCDB - for instance OIM for OSG and also specific VO information sources e.g. ATLAS AGIS system. The **Aggregated Topology Provider** (Previously known as the **Topology Database**) combines information from several topology sources into a single coherent view. In essence, infrastructure sources (such as GOCDB and OIM) provide the list of sites and services in the infrastructure. Then, VO and project topology sources annotate and group these basic sites and services. Eg. ATLAS could create a group called 'ATLAS Tier1 sites' which have sites from both OSG and EGEE. These groups can then be used in other applications such as the availability calculator. The types of information, and the linkages between them are shown in the following picture:

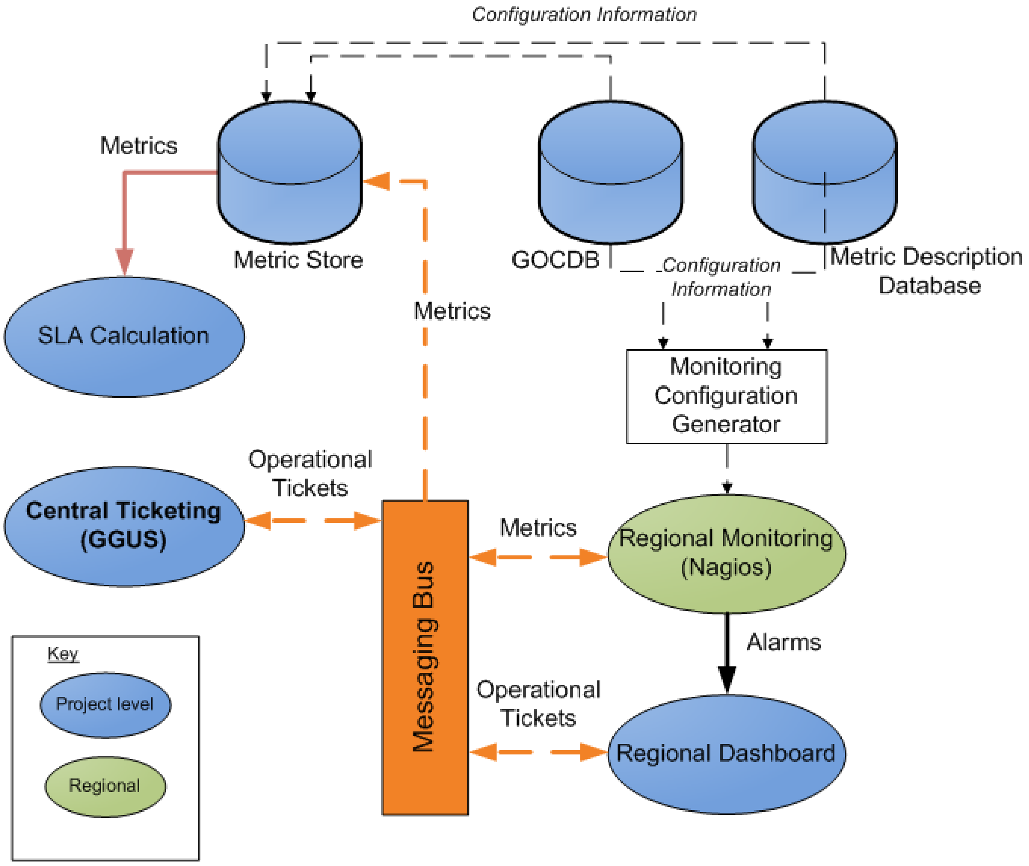


Links between Infrastructure and VO topology sources in the ATP

- **Provider:** BARC/CERN
- **Deployment Model:** Project
- **Description:** A project-level aggregator of various Infrastructure, VO and topology sources
- **Dependencies:** The ATP depends on topology sources making their information available in a standard format
- **Responsibilities:**
    - Integrate information from various topology providers
    - Provide a snapshot of the current topology
    - Provide a history of the aggregate topology
- **Interfaces provided:**
    - Programmatic interface to extract the aggregated topology information
    - Direct SQL access for data-intensive applications e.g. availability calculations
    - Publish topology changes on the messaging system
- **Comments:** The ATP is mainly based on the requirements from WLCG for SAM and Gridview (i.e scheduling of tests and calculation of availabilities).
- **Links and Documentation:** Data Model requirements

Regional monitoring with Nagios

The **regional monitoring system** is the core of MLM, and is responsible for the actual testing of the sites. Once the tests have been run, they're sent to the other components which need them, in particular the **regional dashboard** and **central metric store / SLA calculation**. This is shown below in more detail. Note the use of the **Messaging system** to communicate results to geographically distant components.



Interactions between EGEE Operational Monitoring Components

Regional grid monitoring (Nagios)

* **Provider:** Nagios.org

- **Deployment Model:** Regional
- **Description:** The regional grid monitoring Nagios tests sites and services in the region. It uses the components below to be configured, to store its historical data and to publish its data to other systems via the messaging system. It has a pluggable architecture into which these other components are deployed, including extra visualization tools.
- **Dependencies:**
    - Nagios Configuration Generator (NCG)
    - Regional Metric Store
    - MSG-Nagios bridge
    - User credentials for grid tests are stored in a MyProxy server
- **Responsibilities:**
    - Schedule and run tests against sites and services
    - Generate notifications in case of problems
    - Publish results of tests to message bus
- **Interfaces provided:**
    - Standard Nagios web interface provided to ROC and site admins for viewing and scheduling tests
- **Comments:**
- **Links and Documentation:**
    - How to validate ROC Nagios boxes
    - Nagios: http://www.nagios.org/docs/

Nagios Configuration Generator (NCG)

- **Provider:** SRCE
- **Deployment Model:** All locations where Nagios is needed - Project, Regional, Site
- **Description:** NCG is a configuration tool which generates an appropriate configuration file for Nagios
- **Dependencies:**
    - GOCDB (or any other topology source, such as Aggregated Topology Provider or Information System) provides the list of sites and services
    - Metric Description Database provides the list of metrics that are to be configured
- **Responsibilities:**
    - query the GOCDB and Metric Description DB and produce a suitable Nagios configuration file
- **Interfaces provided:**
    - Command line tool which produces a set of Nagios configuration files.
- **Comments:**
- **Links and Documentation:** GridMonitoringNcgOverview

Regional Metric Store

- **Provider:** Nagios.org
- **Deployment Model:** Regional
- **Description:** The regional metric stores provides a database of current and historical metrics results
- **Dependencies:** MySQL
- **Responsibilities:**
    - Provide the Alarm DB with a list of current metrics which are in CRITICAL or WARNING states
    - Provide regional admins with a local historical store of metrics for their sites for debugging purposes
- **Interfaces provided:**
    - Direct SQL interface for query by Alarm DB
    - Integrated into Metric Visualization system (e.g. RRD graphs of historical results)
- **Comments:** this is simply the standard Nagios DB extension - **NDOUtils**. It's available for MySQL, PostgreSQL and Oracle.
- **Links and Documentation:** http://nagioswiki.org/wiki/Addon:NDOUtils, http://www.nagios.org/docs/

Metric Visualization

* **Provider:** nagios.org

- **Deployment Model:** Regional
- **Description:** (similar to what is in current SAM Portal) - Provide visualization of current tests results and history
- **Dependencies:**
- **Responsibilities:**
- **Interfaces provided:**
    - Generate web-accessible graphs of history for embedding in other tools
- **Comments:** Not sure if this is development work, or just configuration of existing tools like pnp4nagios and NagViz
- **Links and Documentation:** Laurence's notes on pnp4nagios, http://www.pnp4nagios.org/pnp/start, http://nagvis.org/

MSG-Nagios bridge

* **Provider:** SRCE/CERN

- **Deployment Model:** Regional/Project/Site
- **Description:** This provides a mechanism to submit tests results to a Nagios, and also consume test results from a Nagios
- **Dependencies:**
    - Messaging system
- **Responsibilities:**
    - Listen on messaging system for message destined for this Nagios instance and push them to Nagios
    - Publish the results of all grid tests to the messaging system
- **Interfaces provided:**
    - Test results can pushed to Nagios via messaging
    - Test results can be consumed from Nagios via messaging
- **Comments:** This is a set of scripts and daemons which integrate tightly into a Nagios to provide this functionality
- **Links and Documentation:**

Site Debugging tools

* **Provider:** Nagios.org / ???

- **Deployment Model:** Regional
- **Description:** Provides the existing SAMAP functionality e.g. Provide the ability for site admins to reschedule tests via the Nagios interface
- **Dependencies:**
- **Responsibilities:**
- **Interfaces provided:**
    - allow site/ROC admins to re-schedule testing of their sites
- **Comments:** Not clear why this is different from just the standard Nagios functionality - need further clarification with SAMAP team. Indeed this could be seen to be completely satisfied by a site Nagios instance.
- **Links and Documentation:** SAMAP

Project-level metric results and SLA calculations

The EGEE project has a need to keep a reliable long-term store of metric results and availability calculations performed on this data. this includes infrastructure availability (e.g. OPS tests) and also VO specific availabilities.

Project Metric Store

* **Provider:** Nagios.org

- **Deployment Model:** Project
- **Description:** This is a long-term store of metric results gathered from all regional monitoring instances. It would provide a much longer history of the results that the regional instances.
- **Dependencies:**
    - Regional monitoring to test sites and push data to repository via messaging system using standard APIs.
    - SLA Calculation provides the availabity calculation results
    - The Metric Description Database provides additional metadata on the metrics

- **Responsibilities:**
    - Store authoritative EGI.org calculated availabilities.
    - Provide long-term historical repository of all site testing.
- **Interfaces provided:**
    - Interface for management to obtain availability reports (per region, country, site etc…).
    - Interface for other operational tools to obtain calculated availability.
    - Interface to extract historical data from repository
    - Provide a query interface for point intime queries of the metric status
    - Provide an interface to request dumps of historical data
- **Comments:** This would probably leverage NDOUtils much like the regional metric store.
- **Links and Documentation:** http://nagioswiki.org/wiki/Addon:NDOUtils

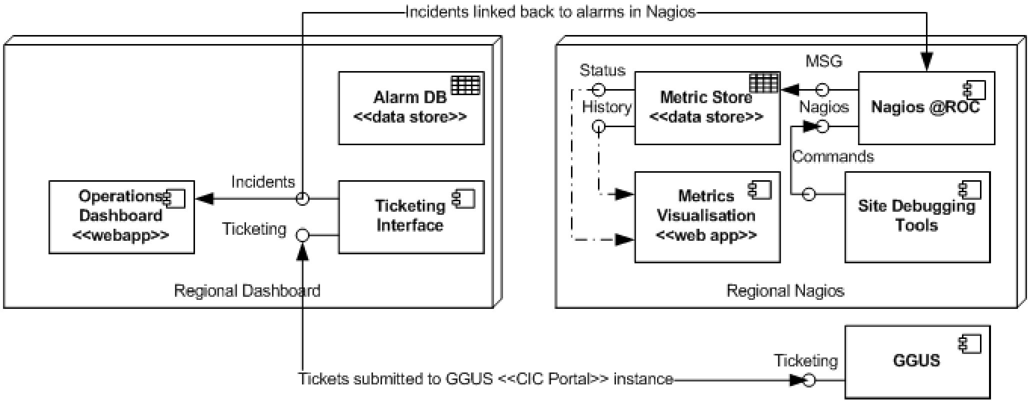
SLA Calculation

* **Provider:** BARC

- **Deployment Model:** Project
- **Description:** This is an active component which calculates availabilities based on the metric results stored in the project metric store
- **Dependencies:**
    - GOCDB or other topology sources to provide list of sites and services that should be used for availability calculation
    - Metric Description Database to provide the list of tests and algorithm to be used for the calculation
- **Responsibilities:**
    - Store authoritative EGI.org calculated availabilities.
- **Interfaces provided:** None - all interaction is via the project metric store
- **Comments:** This component could be deployed regionally too if the regional and project metric store use the same underlying component (e.g. NDOUtils)
- **Links and Documentation:**

Regional operational tools

The regional operational tools implement the EGEE operational model. Their interaction with the other regional components is shown below:



Components in Regional Monitoring

Alarm DB

* **Provider:** IN2P3

- **Deployment Model:** Project (Regionalized views)
- **Description:** Store a list of currently active alarms for the R-COD operations staff to act on
- **Dependencies:**
    - the metric store provides the underlying metric results for which alarms could be generated (via direct SQL)
- **Responsibilities:**
    - from problem metric results provided by Metric store, raise alarms if certain threshold conditions are met. These are used by operators using the operations dashboard
- **Interfaces provided:**
    - A list of alarms are provided to the operations dashboard
- **Comments:** This will be a regionalized view implemented in a central Dashboard
- **Links and Documentation:** OperationalDocumentation

Operations Dashboard

* **Provider:** IN2P3

- **Deployment Model:** Project (Regionalized views)
- **Description:** A dashboard which implements the R-COD workflow for tracking problems at sites, and creating tickets based on alarms
- **Dependencies:**
    - The Alarm DB will provide the list of current alarms to the dashboard
    - ???
- **Responsibilities:**
    - Create tickets for alarm
    - Provide R-COD handover logbook
    - Site notepad for communication with the site
- **Interfaces provided:**
    - Web interface
    - Interface to/from GGUS via Ticketing Interface
- **Comments:** This will be a regionalized system implemented in a central Dashboard
- **Links and Documentation:** OperationalDocumentation, New RCOD Model

Ticketing Interface

* **Provider:** IN2P3/ FZK

- **Deployment Model:** Project (Regionalized views)
- **Description:** A system to create tickets in GGUS, and receive updates
- **Dependencies:**
    - GGUS will provide a standard mechanism to interface with
- **Responsibilities:**
    - inteface between GGUS and the operations dashboard
- **Interfaces provided:**
    - Allow creation of a new ticket for an alarm
    - query interface to receive updated information on a ticket from GGUS
- **Comments:** Need docs for this ! This will be a implemented in a central Dashboard
- **Links and Documentation:**

Site components

At a site, we assume there is a fabric monitoring component deployed. A full set of tests are provided for the grid services. These are integrated into a reference implementation based on **Nagios**, but sites are free to use a different site monitoring tool, provided they test the services to an equivalent level.

There are also probes deployed on grid services to

- extract accounting information
- publish into the messaging system state changes happening in middleware components (e.g. Job status, transfer start/end, ...) that could be of interest to VOs and operators

Site Monitoring (Nagios)

* **Provider:** Nagios.org

- **Deployment Model:** Site
- **Description:** A site monitoring system based on Nagios
- **Dependencies:**
- **Responsibilities:**
    - Generate a configuration for a site based on the information stored about them in topology sources
    - Test the grid services at regular intervals
    - Publish the results of the tests to the messaging system for consumption by interested parties
    - Generate notifications to site managers on service problems
- **Interfaces provided:**
    - Web interface for browsing the status of the site
    - Metrics published via MSG
    - Notifications via email, SMS, IM
- **Comments:** This is all based on standard Nagios components. YAIM is provided to properly configure all the components
- **Links and Documentation:** How to configure Nagios + NCG to monitor your site

Site Monitoring (Probes)

* **Provider:** Nagios.org / CERN / SRCE / ...

- **Deployment Model:** Site
- **Description:** A set of probes which test the grid services at a site.
- **Dependencies:**
- **Responsibilities:**
    - Test the public interface of grid services according to specifications provided by development and operations team
    - Provide checks that can be run directly on service nodes to monitor their health
    - Integrate into Nagios
- **Interfaces provided:**
    - Command line tools which produce output that can be fed directly into Nagios
- **Comments:** these are based initially on SAM tests - other tests are also provided by CERN and SRCE
- **Links and Documentation:** SAM Probes for Nagios

Middleware event publication

* **Provider:** CERN/ BARC

- **Deployment Model:** Site
- **Description:** Extract state changes and summary information from grid middleware and publish them into MSG for consumption by VO and operations clients
- **Dependencies:**
- **Responsibilities:**
    - Publish summary information for data transfers based on internal gridftp/FTS/SRM

information (gained by log mining or querying public and private APIs)
    - Publish summary information for job status based on L&B notifications
    - Publish summary of jobs running in a computing element
- **Interfaces provided:**
    - Publish into MSG in a standard format
    - ...
- **Comments:** Sufficient information for gridview + RealTimeMonitor information should be published
- **Links and Documentation:**

Accounting providers

Worker Node Configuration (Tests)

Infrastructrure

MSG Brokers

* **Provider:**

- **Deployment Model:** Regional
- **Description:**
- **Dependencies:**
- **Responsibilities:**
    - ...
    - ...
    - ...
- **Interfaces provided:**
    - ...
    - ...
- **Comments:**
- **Links and Documentation:**

Operational infrastructure monitoring

* **Provider:**

- **Deployment Model:** Regional
- **Description:**
- **Dependencies:**
- **Responsibilities:**
    - ...
    - ...
    - ...
- **Interfaces provided:**
    - ...
    - ...
- **Comments:**
- **Links and Documentation:**

Portals

Worker Node Configuration Portal

Accounting portal

Operations Portal

QR Metrics Portal

Information System monitoring (gStat)

This topic: EGEE > MultiLevelMonitoringOverview
Topic revision: r11 - 2011-06-20 - AndresAeschlimann