Context:

The DoW for the "VO Services" task under the TNA3.4 activity, states: "*To simplify access to the infrastructure and to promote collaboration within the VO, EGI.eu will (...) operate access to a dashboard infrastructure where the status of the resource fabric being used by a particular VO will be reported upon. Both the portal and dashboard offered by this activity will be basic, but they will provide a core framework around which the particular community can, through their own work, customize their web presence and VO specific monitoring of the infrastructure. The dashboard infrastructure will be based on the work being undertaken in Section 1.3.3.4.3.1: TSA3.2.1 Dashboards.*"

In order to accomplish that task, the VO Services team needs to understand the dashboard framework deployed for the Heavy User Communities (HUCs), how the different dashboard applications work on top of it, and how VOs can customize their VO specific monitoring needs. The VO Services team needs to interact with the Dashboard team in order to be able to refurnish that framework so that it could be used by New Emerging Communities. Please note that New Emerging Communities may just use a simpler version of the HUCs dashboards as long as they can "*,through their own work, customize their web presence and VO specific monitoring of the infrastructure"*.

Questions for clarification:

In the research for the development of this task, the following issues / questions were identified:

1. **Use of ORACLE for the "Data Storage and Aggregation" service; and support status for postgreSQL**
   - From all the presentations [1] we were able to access, and from the information in [2], it seems the HUC dashboard infrastructure is developed on top of Oracle Databases. This is a showstopper for the deployment of this tool by New Emerging Communities. Is it possible to port this framework to MySQL?

   The system was developed having in mind Heavy User Community, like the LHC VOs. Due to performance and scalability considerations, the ORACLE DB is used as a persistency layer. There is no plan to port application to a different DB implementation, though technically it is possible since the Dashboard framework by design foresees various backends, but currently implementation is provided only for ORACLE.

   Also, in [1], it is stated that postgreSQL is supported in some components, but from the documentation it is not clear what is the status of this support, and which modules can be really deployed with postgreSQL. Is postgreSQL really an alternative to ORACLE

   The very first implementation of the Experiment Dashboard used postgreSQL , but as soon as the load had considerably increased, the decision was taken to migrate to ORACLE. The support for postgreSQL implementation was stopped.

2. **Dashboard framework**
   - The dashboard framework documentation [2] is missing important pieces, as for example the DAO section. Is this documentation updated?

Sorry, this is true, this part is missing, we will complete it

- The dashboard components explained in [3] are all installed in the same machine or they could be deployed independently?

They can be deployed independently, they are not at all coupled, and they are installed independently according to the current Dashboard deployment model.

3. **Dashboard Applications**
   - Several dashboard applications are built on top of the dashboard framework namely the Site Status Board, Job Monitoring, Site View, SAM Web portal, etc... However, it is not clear the architecture of those applications, and how these applications are built on top of the dashboard framework. Can you point us to the correct documentation how these applications work and how are they deployed on top of the dashboard framework?

     All applications mentioned above are different but are designed in the common framework, so they use common modules, share the build system, etc…

     Deployment: An automated build system extends Python distutils to enforce strict development, test and release procedures for the various common and application-specific components. The automated build system creates the RPMs of the modules from the SVN and the user can install/upgrade those RPMs by using the 'yum' command with the Dashboard 'yum' repository enabled. The repository can be found at: http://dashb-slc5-build.cern.ch/apt/

     We do not have per application documentation, the most complete description can be found in the following paper: http://www.springerlink.com/content/f750046580857157/fulltext.pdf.

     Main points about application mentioned above:
     1) SiteStatusBoard is a framework allowing VOs to describe and to follow the state of their distributed sites with a set of monitoring metrics, which VOs define by themselves. The application consists of data repository, agents updating the repository from the URLs where monitoring metrics are published by data providers, and user interface. Some of monitoring metrics are standard and are provided as a part of SSB, among them downtime information, results of SAM tests, whether the site is visible in BDII.
     2) Job Monitoring consists of the dashboard agents collecting data from MSG or MonAlisa (two implementations are currently available), data repository and a set of user interfaces on top of a single repository. Currently the main information sources are VO frameworks instrumented for Dashboard reporting. In case Logging and Bookkeeping system propagates job status information to the Messaging System for the Grids (MSG), job monitoring application can use this data as information source and instrumentation of the VO workload management systems would not be mandatory any more.
     3) SAM Web portal or site usability interface in the new implementation will consist of UI which will retrieve data from the information sources via APIs.

- Among the dashboard applications in place, the SAM Web portal seems one of the central pieces which could be used by VOs. However, VOs monitoring is now done using NAGIOS. What is the status of the NAGIOS integration with the SAM Web portal application and with the dashboard framework?

  The new portal is under construction. The SAM system is being completely redesigned and is not yet fully functional. Though we started to develop the new application compatible with the new SAM architecture based on Nagios we need to wait for APIs and availability calculation being provided by SAM developers, which is not yet the case.

- Each application has to have a unique dashboard framework behind, or a single dashboard framework can serve multiple dashboard applications?

  The framework consists of the set of common modules, shared build system, tools for managing and monitoring of the Dashboard agents. All applications are built in the same framework, though depending on their functionality not all of them use the same set of common modules.

4. Requirements

- Does the dashboard framework and the dashboard applications have specific storage needs? What is the typically amount of data generated by each of those components?

  The main storage needs are for database which keeps monitoring data, other components do not have specific storage requirements.

- What are the machine requirements (CPUs, RAM, ...) to run each of those components?

  There are no specific requirements except those from the actual operating system itself. The system specifications list for installing SLC5 which is based on Red Hat Enterprise Linux 5 can be found at: http://linux.web.cern.ch/linux/scientific5/docs/rhel/Installation_Guide/ch-ent-table.html

5. Operation and Support

- What is the effort on the operation of the dashboard framework and on the dashboard applications?

  Operation effort is not high. We are lucky to have excellent ORACLE support provided by CERN ORACLE team since otherwise some effort should go into support of the data repository. There are alarms sent in case some of agents have troubles and have to be restarted, though normally they restart automatically.

- What is the frequency for updates and software releases?
  This can vary from one application to another. We still get a lot of requests from the user communities and the main effort is focused on the development. Typically, we have monthly updates, though some applications are more stable than others.

6. Dependencies

- What are the dashboard framework and dashboard dependencies with respect to external (grid) services?

  Apart of BDII, main dependencies are not so much related to the GRID services but there are some requirements to the VO-specific systems, which are used as data source in case of job monitoring or data transfer. There are two types of data required: VO topology data (which sites and services are used by a particular VO) and monitoring data about a particular computing activity. For the topology information BDII can be used as an information source. Monitoring data about a particular computing activity is coming from the VO workload and data management systems instrumented with standard libraries for Dashboard reporting.

  The reporting is currently implemented via two main transport mechanisms: MonAlisa and Messaging System for the Grids (MSG) based on ActiveMQ message bus. In case MSG is provided as a standard service of the middleware stack , all Dashboard applications would use it for communication with the information sources. For Site Availability Dashboard the main dependency in the current implementation is direct access to SAM DB. This should change in the new version which will get information about SAM tests and calculated availabilities through the APIs provided by SAM system.

- What are the dashboard framework and dashboard dependencies with respect to external software?

  All the following dependencies can be found on the Scientific Linux CERN (SLC) repository : httpd,mod_python ,python, libxslt-python ,docbook-utils, docbook-utils-pdf, docbook-style-xsl, libxslt

  The following dependencies are not on the SLC repository and can be found on the Dashboard repository for the external software http://dashb-slc5-build.cern.ch/apt/RPMS.external/ :
  Pychart, pycurl,cx_Oracle, oracle-instanceclient-basic

  Depending on the Dashboard module, there might be a few more dependencies required such as the matplotlib.

References:

1. http://dashboard.cern.ch/tutorials/
2. http://dashb-build.cern.ch/build/nightly/doc/guides/common/html/dev/index.html
3. http://dashb-build.cern.ch/build/nightly/doc/guides/common/html/dev/index.html#dashboardDesignFigure