The “Common Solutions” Strategy of the Experiment Support group at CERN for the LHC Experiments

M Girone, J Andreeva, F H Barreiro Megino, S Campana, M Cinquilli, A Di Girolamo, M Dimou, D Giordano, E Karavakis, M J Kenyon, L Kokozkiewicz, E Lanciotti, M Litmaath, N Magini, G Negri, S Roiser, P Saiz, M D Saiz Santos, J Schovancova, A Sciabà, D Spiga, R Trentadue, D Tuckett, A Valassi, D C Van der Ster and J D Shiers
CERN IT-ES
CH-1211 Geneva 23, Switzerland

E-mail: Maria.Girone@cern.ch

Abstract. After two years of LHC data taking, processing and analysis and with numerous changes in computing technology, a number of aspects of the experiments’ computing, as well as WLCG deployment and operations, need to evolve. As part of the activities of the Experiment Support group in CERN’s IT department, and reinforced by effort from the EGI-InSPIRE project, we present work aimed at common solutions across all LHC experiments. Such solutions allow us not only to optimize development manpower but also offer lower long-term maintenance and support costs. The main areas cover Distributed Data Management, Data Analysis, Monitoring and the LCG Persistency Framework. Specific tools have been developed including the HammerCloud framework, automated services for data placement, data cleaning and data integrity (such as the data popularity service for CMS, the common Victor cleaning agent for ATLAS and CMS and tools for catalogue/storage consistency), the Dashboard Monitoring framework (job monitoring, data management monitoring, File Transfer monitoring) and the Site Status Board. This talk focuses primarily on the strategic aspects of providing such common solutions and how this relates to the overall goals of long-term sustainability and the relationship to the various WLCG Technical Evolution Groups.

1. Introduction
There are four main experiments at the Large Hadron Collider (LHC) at CERN. Two of them, ATLAS and CMS, are large multipurpose detectors and two, ALICE and LHCb, have somewhat smaller collaborations and are more specialized. All four experiments make use of infrastructure and computing resources of the Worldwide LHC Computing grid (WLCG) [1]. The workflows and the related technical tools have been developed by the experiments over more than a decade of preparation and several years of operations. There are differences and specializations in all of the systems, but there is also significant commonality in the workflows executed and the techniques used.

While all the experiment collaborations are very large, running into the thousands of physicists, the amount of effort in computing development is limited. Each experiment is expected to succeed at the core mission of producing physics results. Computing infrastructure is a needed piece for that goal,
but the focus is on the physics analysis. As a consequence the development effort available to the experiments is steadily decreasing.

In this paper we present the work on common solutions for the LHC experiments through the Experiment Support group of CERN-IT. Common solutions try to take advantage of the similarities in the experiment activities and make the most efficient use of the limited development effort.

2. The group
The Experiment Support (ES) group in CERN IT is composed of approximately 30 people ranging from permanent staff, to fellows and students. A significant fraction of the group is supported on project funding through the European Grid Initiative (EGI) [2]. The effort is a balance of contributions to the development and operations activities of the individual experiments, and contributions to common projects resident in IT-ES. This balance has helped create a unique resource in LHC computing. The team has experience with systems than spans experiment boundaries and this experience makes IT-ES particularly well suited to find and develop common services for the LHC experiments, in line with the WLCG and IT goals.

The current LHC schedule presents an upcoming window of opportunity for large technology changes in the next few years and many of the LHC experiment computing frameworks will undergo major revisions to address issues related to long term sustainability and scalability while IT and WLCG will continue to evolve technically. The strategy for this has been documented by some Technical Evolution Groups [3].

The main activities of the group can be summarized as:

- Data Management support: covers the development and integration of the experiment-specific and shared grid middleware;
- Monitoring and Experiment Dashboards: allows the experiments and sites to monitor and track their production and analysis activities across the grid;
- The LCG Persistency Framework: handles the event and detector conditions data from the experiments;
- Distributed production and analysis: design and development for many critical experiment workload management and analysis components.

On the time scale of the LHC upgrade shutdown and beyond, computing challenges faced in common by the LHC experiments will motivate work in the following areas:

- Factorization and consolidation of the experiment computing frameworks in order to promote a long-term sustainability;
- Evolution of data and workload management systems to cope with the ever increasing data volumes.

3. Identifying common solutions
When looking for potential common solutions we look at how existing experiment services can be broken into elements, as shown in Figure 1. The four experiments rely on many of the same sets of common site infrastructure and lower level grid interface components. One of the big successes of the WLCG has been the broad adoption of the common grid infrastructure. On the other end of the diagram all experiments have elements developed to interface with experiment specific components that were designed to match physics needs or early technical decisions. The interface layer between is the prime location for common solutions.
The intermediate layer typically translates the experiment-specific aspects, hopefully through a well-defined interface, with the common site and grid components. The work performed in this layer can be substantial, and often has strong similarities across experiments. Work performed in the interface layer can be applied to several experiments and reduces the overall development effort, while maintaining experiment-specific components.

4. Examples of successful common solutions

Over the last year and a half, IT-ES has developed a number of successful common components that have been rolled out for production use by the LHC experiments. We describe several in the following sections.

4.1. Monitoring and Experiment Dashboards

Monitoring and Experiment Dashboards allow experiments and sites to monitor and track their production and analysis activities across the grid. We distinguish:

- Data management monitoring: services for data popularity, data cleaning and data integrity
- HammerCloud: a site functionality and stress testing framework;
- Dashboard framework and applications.

4.1.1. Data Management Monitoring: Data Popularity and Cleaning

The Data Popularity Service is designed to help the LHC experiments track what data is used, how much it is used, where it is used, and who uses it [4]. The data popularity relies on the fact that all the experiments open files and how those files are opened and accessed can be tracked in a common way. The experiment-specific elements are how those files are mapped into the logical quantities the experiment tracks like datasets, reprocessing campaigns, and simulation campaigns. The mapping between them is provided by a common solution that provides the database of accesses and can make projections, plots and tables of the access and use based on logical mappings provided by the experiment specific components. The data popularity service is used by the experiments to assess the importance of computing processing work, and to decide when the number of replicas of a sample needs to be adjusted either up or down.

The service, originally inspired by the ATLAS experience [5], is designed with a modular architecture. As shown in Figure 2, the system is decoupled from the data sources and gathers information from them via dedicated plugins, which populate a table in the Popularity database. This guarantees flexibility and allows extension to different providers of information.
Currently, plugins have been implemented for the CMS and LHCb workload management systems and for XRootD [6]. The latter is a step forward as it gives the possibility to trace all storage back-ends using the XROOT framework, including the EOS system [7] used at CERN. Monitoring the backend directly also allows tracing accesses that have not been scheduled via the experiment workload systems.

The Site Cleaning Agent is used to suggest obsolete or unused data that can be safely deleted without affecting analysis. Figure 3 shows the agent workflow. The information about space usage is taken from the experiment data management and transfer system (PhEDEx [8] for CMS, DQ2 [9] for ATLAS). The pledges are taken from a catalogue of the resources of the experiment computing sites. In the second step the agent will try to select replicas to clean full sites until a quota of 30% or 25TB of free space per site. In order to choose the unpopular replicas, the agent will get a dump of the replicas owned by the groups at each site from the Popularity service and the data management system, and will identify the datasets that can be removed without risk for the experiment analyses.
The service exposes data through a web user interface that presents a basic storage accounting system (Figure 4) and also provides the interface for physics groups and site administrators who can select the suggested replicas in a way convenient to forward the request to the deletion services.

![Figure 4: Space usage overview for the CMS UK Tier-2 sites showing for each group-site association the allocated space (green), the used space (blue) and the space limit (red).](image)

**4.1.2. Data Management Monitoring: Data Integrity**

In the distributed computing model of WLCG Grid Storage Elements (SE) are, by construction, completely decoupled from the File Catalogues (FC) where the experiment files are registered. On the basis of the experience of managing large volumes of data in such environment inconsistencies have often happened, either causing a waste of disk space in case the data were deleted from the FC but still physically on the SE, or serious operational problems in the opposite case, when some data registered in the FC was not found on the SE. Therefore, three of the LHC experiments, ATLAS, CMS and LHCb, have implemented dedicated systems to ensure the consistency of the data stored on the SEs with the information reported in the FCs implementing systematic checks. The objectives of these checks is to spot any inconsistency above a certain threshold that cannot only be due to the expected latency between data upload and registration, and in such case try and identify the problematic data. The consistency checks have been implemented by every experiment according to its needs and integrated in its data management framework; however they all rely on the information produced by sites, who should make available to the experiments a full dump of their SEs on weekly or monthly basis.

The IT-ES group has taken the initiative to define a common format and procedure to produce storage dumps in coordination with the experiments, in order to provide a solution as generic as possible that can suit all LHC experiments and contributes to reduce the effort for the sites that are asked to provide such data. This also streamlines the procedure on the experiment side, as the storage dump format follows a defined standard, thus limiting the effort needed to parse and process them. Currently a standard format and procedure are defined for dCache, StoRM, Castor, DPM and instructions are made available for sites that are asked to provide such information.
4.1.3. HammerCloud
The need for testing grid sites and their services with realistic experiment workflows and in an automated way was recognized very early in the LHC computing projects. The first instance of a stable service capable of that is the CMS Job Robot, developed for the CMS Computing Software and Analysis Challenge in 2006 (CSA06) and used in production by CMS ever since. The CMS Job Robot had some important limitations: its design made it almost impossible to adapt it to the workload management systems of other experiments and it had a very crude user interface. A similar tool, the GangaRobot, was developed in 2007 to automate the submission of jobs via Ganga [10]. Ganga is a generic grid job submission and management tool, with its main LHC usage coming from the ATLAS and LHCb user communities. Using the GangaRobot, ATLAS developed a site functional testing service for the EGEE/LCG sites used by ATLAS. HammerCloud [11], developed from 2008 by CERN IT in collaboration with the LMU in Munich, was designed to improve the efficiency and scalability of the ATLAS GangaRobot.

One key feature that HammerCloud added was a stress-testing service; this feature was successfully used in a series of global stress tests to validate the overall performance of the ATLAS Grid prior to the start of LHC data taking. HammerCloud features a web front-end, a database of historical test statistics, and a back-end implementing the test submission logic; the job submission is done via Ganga (with the GangaRobot plugin) and the interface to the experiment-specific software is almost completely contained in dedicated Ganga plugins. According to the general diagram, HammerCloud interacts with low level services (computing and storage services at sites) via experiment drivers (the Ganga plugins). Given the success of HammerCloud in ATLAS, IT-ES was requested in 2010 to develop prototype instances of the service for the CMS and LHCb experiments. Very recently, after several improvements in functionality and performance, CMS has replaced its Job Robot with HammerCloud in production. This convergence of site testing tools will lead to a decrease in the amount of support and development effort required, to a higher level of uniformity among experiments (particularly beneficial to sites supporting more than one) and – as a consequence – to experiments automatically benefitting from new features requested by other experiments.

4.1.4. Dashboard framework and applications
The Dashboard framework [12] is one of the original common services. It relies on the principle that all experiments submit jobs, though how those jobs map to activities or to sites may be experiment specific. The dashboard is used by the LHC experiments to track historically and interactively how the resources are being used and by what activities.

The framework provides the necessary components to build monitoring applications, as shown in Figure 5. Collectors gather information and store it using a Database Access Object (DAO) layer, which decouples the application from the database. Data can be accessed via the DAO, using web actions and views allowing HTML, XML, JSON, text or image export.
A set of JavaScript libraries make it possible to build functionally rich and intuitive web applications that allow users to access and analyse data in novel ways. To take advantage of these possibilities, the Dashboard team has adopted an approach to UI development that builds on top of the Dashboard framework. The approach is based on a common architecture and cocktail of technologies, principally jQuery and selected plug-ins. Interaction with the server passes entirely via web API, decoupling UI and server code. This decoupling has enabled the Dashboard team to build new UIs for a set of applications, as described below.

The Experiment Dashboard job monitoring system collects information from various information sources such as VO-specific job submission systems and running jobs. These are instrumented to report job status updates to a messaging server, either MonALISA monitoring system or Apache ActiveMQ Messaging System for the Grid (MSG). Several dashboard collectors gather job status updates from different sources and store them in the Dashboard Data Repository, implemented in Oracle. In order to ensure the reliability of the service, the collectors are monitored on three different levels: database, runtime and service. The collected information is then available to the user via web UIs. The same data repository is used by multiple applications within a VO. Each application focuses on a particular use case and targets a specific category of users. The architecture is shared between ATLAS and CMS even though they use different job submission systems and execution back-ends. The web front-ends, both for the ATLAS and CMS collaboration, expose modern user-interfaces.

Data transfers for the LHC experiments are monitored by experiment-specific systems, such as MonALISA for ALICE, DDM Dashboard for ATLAS, PhEDEx for CMS and Dirac for LHCb. WLCG Transfers Dashboard [13] has been developed to provide a common solution and a cross-experiment view of WLCG transfers facilitating analysis of transfer performance. Rapid development has been achieved by re-using the server design and UI code of ATLAS DDM Dashboard. WLCG Transfers Dashboard has been designed to work transparently across experiments and transfer technologies, thanks to data source decoupling provided by the use of MSG. Experiment-specific monitoring systems can integrate at several levels: consuming transfer events via MSG, retrieving statistics via web API, or including plots directly. In the first instance, transfers performed by FTS are being monitored. In the future, other technologies, such as XRootD, will also be monitored.

The Site Status Board [14] also follows the normal pattern of a successful common service, facilitating the monitoring of sites and allowing the experiment to assess the operational status of the global infrastructure.
VO administrators can define metrics that they consider to be important, such as number of running jobs, status of a particular service, downtime information, tickets assigned to sites and so forth. It offers the flexibility to add or modify metrics on the fly. These metrics can be combined into views. Currently, CMS and ATLAS have defined more than 300 metrics, which are combined into more than 20 different views. ATLAS and CMS are heavily using SSB on their production monitoring systems.

The four LHC experiments perform sanity tests of the important services at WLCG sites with SAM/Nagios [15]. A group of tests define the status of a service, which are then aggregated into profiles. The Experiment Dashboard Framework provides Site Usability Monitoring (SUM) to visualize test results, and availabilities and reliabilities of sites and site services. The SUM UI is used by all the LHC experiments. The SUM UI provides two main views: latest results to show the current status of all endpoints at VO sites, and status history views. Separate history views are provided for site availability or reliability, aggregated from site services history, for site services, and for test results. The different history views are linked together. This way, users can start with site status and drill down through site services, test results, and finish with detailed log files per test result. The visualized quantities follow a simple colour scheme to enable users to immediately identify ongoing issues with services, and to determine their duration. With these features the SUM UI targets the experiment computing operations experts, shifters, and site administrators, who can easily identify and address or escalate issues with services at the sites.

4.2. Persistency Framework
The LCG Persistency Framework [16] consists of three packages (CORAL, COOL and POOL) that address the requirements of the LHC experiments for the handling of their event and detector conditions data. POOL is a generic hybrid store for C++ objects, metadata catalogs and collections, using streaming and relational technologies. CORAL is a generic abstraction layer with an SQL-free API for accessing relational databases. COOL provides specific software to handle the time variation and versioning of conditions data. All packages are written in C++, but Python bindings are also provided for CORAL and COOL.

From the architectural point of view, the Persistency Framework is a typical example of a common solution across LHC experiments. As shown in Figure 6 CORAL, COOL and POOL sit at the boundary between the experiment-specific software frameworks and the lower-level computing services. The framework libraries provide abstract APIs, which are used directly by the experiment frameworks for data access using file streaming or database technologies such as ROOT, Oracle and SQLite. The Persistency Framework software is used by experiment jobs to access conditions, events and other types of data. CORAL is also used internally by COOL and POOL to access relational databases.
The Persistency Framework is also a typical example of a common solution across LHC experiments from the point of view of the development process through a well-established collaboration of developers from the LHC experiments with a team in the CERN IT department, now in IT-ES, which ensures the overall project coordination. The PF benefits strongly from the close collaboration with other WLCG Application Area projects, especially ROOT (that provides the object streaming software for POOL and the Python binding software for COOL) and WLCG Software Project and Infrastructure (that provides and operates the build and test infrastructure for the software). The close collaboration with the team operating the relational database services has been essential for efficient deployment of the PF software. The PF team is an example of how IT-ES provides the glue at the human level between the user community in the LHC experiments and the service providers in CERN IT and other Grid sites. The development priorities to meet the requests of the LHC experiments are set with their representatives in the Architects Forum, where all WLCG Application Area projects are also represented.

5. Current Developments for a Common Analysis Framework

Up to now, most of the common services that have been put into production are monitoring services. Monitoring is important to the experiment and a place where groups expend effort, but it is also outside the central workflows. With the success of projects of more limited scope, there is a feasibility study to develop a common analysis submission framework between ATLAS and CMS. This would represent the highest-level service yet attempted as a common solution. If this service gets out of the study phase, it could be a substantial savings of development effort in the long term.

The analysis infrastructure of each experiment essentially has common goals: to enable access to experiment data for processing and to insulate the user from the complexity of the distributed computing infrastructure. The analysis infrastructure interfaces with the data management components to discover the data location, tracks the processing requests and resubmits failed jobs to deal with transient failures, and returns the produced data products. The analysis submission system
is optimized to support the entire experiment and to make the most efficient use of the resources. Prioritization and resource utilization is controlled in the distributed analysis layer.

In the common solution the experiment specific components will handle the data discovery elements, the job splitting and the packaging of the user environment. The common infrastructure is the job submission and execution. The layer in between that is being examined for a potential common solution is the workflow engine that handles prioritization, job tracking, and job resubmission. The systems that performs this functionality currently in ATLAS is called PanDA [17] and in CMS is WMAgent [18]. PanDA has managed the production and analysis activities for more than five years, reaching peaks of up to 100k-150k concurrently running jobs globally and up to 1 million jobs per day.

The interface between the experiment specific and common infrastructure contains a lot of functionality and has involved substantial effort from both experiments over several years. The substantial investment does not discourage the adoption of a common LHC solution. Even though adopting common components involves potentially modifying or abandoning services developed over long periods of time there are still long-term benefits in operations and sustainability.

The work on analysis tools is just completing the feasibility study. A formal process with an initial joint feasibility study, followed by a proof of concept prototype, was proposed for analysis because it is a service central to the experiment’s success and impacts many analysis users. The eventual goal of the project is for the experiments to use a common framework based on elements from PanDA and GlideInWMS [19]. The present feasibility study has covered the following steps:

1. review the architecture and functionality of the current analysis frameworks;
2. determine which elements could be provided in common;
3. identify how to interface to existing external services;
4. develop an architecture identifying which experiment specific services could be replaced with common elements.

We will gain additional experience with the design of the proof of concept prototype.

6. Looking Forward

If IT-ES and the experiments are able to work together and develop common components for a service as critical as the analysis submission tools, it will be possible to complete additional common projects in other critical service areas. Data management is another possible area to work in. With the success of data popularity there is some commonality in the support infrastructure for data management, but the core data management services are experiment-specific. Looking at the standard common services diagram, the service that tracks files in transfer and handles consistency and retries looks logically similar to the same component in the analysis submission. In one case workflow tasks are mapped to jobs in the experiment-specific services, tracked and prioritized by the common components, and submitted to the common grid infrastructure. In the case of data management, the experiment specific concepts like datasets are mapped to files by the experiment services, handed to a common service for tracking, and then given to the common grid infrastructure for the transfer of individual files.

7. Outlook

The Experiment Support group in IT is a unique resource in LHC Computing development. The developers in the group have the ability to cross experiment boundaries and the group is well situated to develop common solutions useful to several groups. There have been successful examples of common solutions deployed in the production environments of the LHC experiments. Generally these
have been support components and monitoring elements, services that facilitate the core mission but are not critical to the core functionality.

The success of the service components has given us confidence in the process, and has developed the trust of the stakeholders. We are now attempting to expand the development of common solutions into the more critical workflows. The first is a feasibility study of common analysis workflow execution elements between ATLAS and CMS. We look forward to additional common development in the future.

8. References
[1] I. Bird et al. “LHC computing Grid. Technical design report” CERN-LHCC-2005-024 (2005)
[2] EGI (European Grid Infrastructure) http://www.egi.eu/
[3] https://espace.cern.ch/WLCG-document-repository/Boards/MB
[4] D. Giordano et al., “Implementing data placement strategies for the CMS experiment based on a popularity model”, to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference.
[5] A Molfetas et al. “Popularity framework to process dataset tracers and its application on dynamic replica reduction in the ATLAS experiment”, CHEP, Taipei, Taiwan, October 18-22, 2010
[6] XrootD: http://xrootd.org/
[7] A. Peters et al., Exabyte Scale Storage at CERN J. Phys.: Conf. Series 331 052015.
[8] Rehn J, Barass T, Bonacorsi D, Hernandez J, Semeniouk I, Tuura L and Wu Y, “PhEDEx high-throughput data transfer management system”, Computing in High Energy Physics 2006 (CHEP06) (Mumbai, India)
[9] M Branco et al. “Managing ATLAS data on a petabyte-scale with DQ2”, CHEP, Victoria, Canada, September 2-7, 2007
[10] J.T. Moscicki et al., “Ganga: A tool for computational-task management and easy access to Grid resources”, Computer Physics Communications. v.180-11. pp.2303-2316. ISSN 0010-4655. 10.1016/j.cpc.2009.06.016.
[11] D. van der Ster et al., “Experience in Grid Site Testing for ATLAS, CMS and LHcb with HammerCloud”, to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference.
[12] J. Andreeva et al., “Experiment Dashboard for monitoring of the LHC distributed computing systems”, 2011, J. Phys.: Conf. Ser. 331 072001 doi:10.1088/1742-6596/331/7/072001; J. Andreeva et al., “Designing and developing portable large-scale JavaScript web applications within the Experiment Dashboard framework” to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference.
[13] J. Andreeva et al., “Providing WLCG Global Transfer monitoring” to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference.
[14] Saiz P et al, “Collaborative development. Case study of the development of flexible monitoring applications”, to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference.
[15] A. Di Girolamo et al., “New solutions for large scale functional tests in the WLCG infrastructure with SAM/Nagios: the experiments experience”, to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference.
[16] R. Trentadue et al, “LCG Persistency Framework (CORAL, COOL, POOL)”, these proceedings
[17] Maeno, T. (2008). PanDA: distributed production and distributed analysis system for ATLAS. Journal of Physics Conference Series, 062036(6), 62036. Retrieved from http://www.iop.org/EJ/abstract/1742-6596/119/6/062036 .
[18] Frank van Lingen et al (2010) “Job life cycle management libraries for CMS workflow management projects”, J. Phys.: Conf. Ser. 219 042024 doi:10.1088/1742-6596/219/4/042024
[19] I Sfiligoi, glideinWMS—a generic pilot-based workload management system 2008 J. Phys.;
