Lessons Learned From Jointly Using HTC- and HPC-driven e-Science Infrastructures in Fusion Science

Academic Analysis of an Interoperability-enabled Fusion Framework for DEISA and EGEE

M. S. Memon, M. Riedel, A. S. Memon,
F. Wolf, A. Streit, Th. Lippert
Jülich Supercomputing Centre
Forschungszentrum Jülich, Jülich, Germany
E-Mail: m.memon@fz-juelich.de

Marcin Plociennik, Michal Owsiak
Poznan Supercomputing and Networking Center,
Poznan, Poland

David Tskhakaya
Institute for Theoretical Physics
University of Innsbruck, Innsbruck, Austria

Christian Konz
Max-Planck-Institut für Plasmaphysik
Garching, Germany

Abstract— **The interoperability of e-Science infrastructures like DEISA/PRACE and EGEE/EGI is an increasing demand for a wide variety of cross-Grid applications, but interoperability based on common open standards adopted by Grid middleware is only starting to emerge and is not broadly provided today. In earlier work, we have shown how refined open standards form a reference model, which is based on careful academic analysis of lessons learned obtained from production cross-Grid applications that require access to both, High Throughput Computing (HTC) resources as well as High Performance Computing (HPC) resources. This paper provides insights in several concepts of this reference model with a particular focus on the finding of using HPC and HTC resources with the fusion applications BIT1 and a cross-infrastructure workflow based on the HELENA and ILSA fusion applications. Based on lessons learned over years gained with production interoperability setups and experimental interoperability work between production Grids like EGEE, DEISA, and NorduGrid, we illustrate how open Grid standards (e.g. OGSA-BES, JSDL, GLUE2, etc) can be used to overcome several limitations of the production architecture of the EUFORIA framework paving the way to a more standards-based and thus more maintainable and efficient solution.**

Keywords – HTC; HPC; Infrastructure; Interoperability; Fusion

I. INTRODUCTION

In last couple of years, we observe an increasing number of end-users that raise the demand to jointly access both High Throughput Computing (HTC)-driven infrastructures (EGEE, OSG, etc.) and infrastructures driven by High Performance Computing (HPC) needs such as DEISA or TeraGrid. In this context, the fundamental difference between HPC and HTC is that HPC resources (e.g. supercomputers, large-scale clusters, etc.) provide a good interconnection of cpus/cores while HTC resources (e.g. pc pools, smaller clusters, etc.) do not. In many cases, this joint use is typically motivated by the theory and concept that tackle a particular scientific problem leading to a set of corresponding computing codes that are executed as part of a larger cross-infrastructure workflow. While some of these codes can be considered as 'nicely parallel' (suitable for HTC), others require to be computed as 'massively parallel' (suitable for HPC) simulations. In other cases, the joint use of HTC- and HPC-driven infrastructures is motivated by the fact that often end-users perform smaller evaluation runs with their codes on small HTC-like resources before performing full-blown production runs on large-scale HPC resources.

In earlier work, we have shown that large scientific communities (e.g. Bio-med domain [1], e-Health domain [2]) can actually benefit from using jointly HPC and HTC resources on world-wide interoperable e-science infrastructures. In a similar manner, this contribution reveals another use of these different computing paradigms on European infrastructures in the field of fusion science. We provide insights into the architectural setup of the EUFORIA project [3] used in production to leverage the power of EGEE resources (i.e. HTC) and DEISA resources (i.e. HPC). In addition, this contribution thoroughly evaluates the production setup and provides lessons learned with a particular focus on how common open standard improvements based on our earlier published infrastructure interoperability reference model (IIRM) [4] have the potential to improve the EUFORIA framework architecture to make it more supportable (i.e. sustainable, maintainable, etc.) and to use infrastructure resources more efficiently.

The remainder of the paper is structured as follows. After the introduction, we provide the architecture used in production in Section 2. Section 3 describes some use case applications of our architectural setup in detail while Section 4 provides an academic analysis of the gained experience with a particular focus on challenges during the interoperability setup. Section 5 provides a roadmap of how we overcome these challenges by following the design of the IIRM. Section 6 reviews related work and this paper ends with concluding remarks.

978-1-4244-8003-6/10/$26.00

II. ARCHITECTURE AND INFRASTRUCTURE SETUP

The Infrastructure setup of our solution is based on the HPC-driven infrastructure DEISA and the HTC-oriented EGEE infrastructure. Note that we are already partly in the transition process from DEISA to PRACE and EGEE to EGI, but in principle their nature of supporting either HPC or HTC remains the same. While designing an architectural framework, one of the requirements of the fusion community is to have workflow incarnation over homogeneous and heterogeneous resource composites. It thus implies to leverage the potential of the DEISA and EGEE infrastructures with their different types of Grid resources. In our production setup, the components used are described in the following paragraphs.

As part of the ClientLayer, the client tool is a graphical user interface responsible of user interaction and managing multisite, distributed, and heterogeneous workflows. The heterogeneous here implies that a single workflow spanning multi-step activities contains Grid/HTC and HPC application executions. The client side is a thick layer of components where users can easily define workflows with multiple and nested steps. For managing workflows, our production setup is using the Kepler [9] tool, which is a generic scientific workflow engine supporting both grid and HPC based activities. This tool is capable of supporting general job execution use case in terms of actors, therefore we defined Grid/HTC as well as HPC specific additional actors enabling job management and monitoring, including data staging functions. By combining a meaningful sequence of such actors, client applications can create a complete workflow in a very flexible and convenient manner. Once the workflow is verified, the request for each actor is forwarded to the so-called Roaming Access Server (RAS).

The RAS in turn provides general functions of job management and monitoring as well as certain data management functions. As shown in Figure 1 the RAS interacts with Grid/HTC as well as HPC based resources through Grid middleware-based adapters. For Grid/HTC based job requests RAS communicates with gLite [10] services, whereas for HPC services it interacts with UNICORE [11] services. gLite is the middleware used in the EGEE infrastructure, while UNICORE provides access to resources of the DEISA infrastructure. Using jointly these types of services and thus HPC or HTC resources, end-users can conveniently execute serial and parallel jobs. It is mostly a decision by scientists to decide on the application launch, as it cannot be apprehended beforehand for any application's suitability with target site and architecture. In our production setup, UNICORE is used as an interface to HPC and supports a wide range of resource management systems such as Torque, LSF, SLURM, and LL. As shown in Figure 1, RAS interacts with UNICORE using the so-called Vine toolkit.

The Vine Tookit is an adapter to middleware for HPC and a Java based high-level API for managing jobs on specific target sites. The environments in which Vine can be deployed are Java Web Start, Java Servlet 2.3, and Java Portlet 1.0. Currently, the production version supports UNICORE (e.g. SSL key generation, and job-based management), and VOMS (e.g. local proxy delegation, registering and un-registering users). In order to send jobs to a middleware, Vine has respective client classes. As part of a production installation, it has to be deployed as a Web service application deployed in a Web container. Some features that we used for our use case applications are the proprietary UNICORE interface for job management, queries of up to date job statuses (automatically checked by RAS modules), end-user management, VOMS proxy generation, and LFC management.

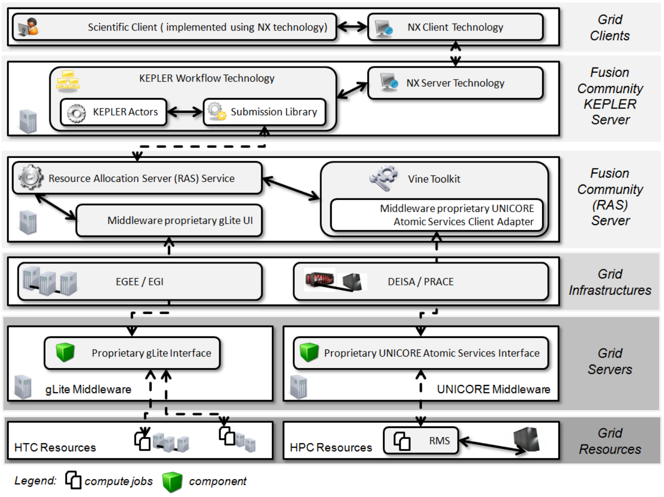

Figure 1. Production Architecture and infrastructure setup.

Other minor elements of the architectural framework are illustrated in Figure 1. A fusion scientist typically uses a scientific client that is implemented in our architecture as a client based on the NX client technology, which handles remote X window connections to the NX Server installed on the Fusion Community KEPLER Server and thus in turn Kepler can be locally displayed on the scientist desktop. The Kepler workflow technology is then used to setup fusion application workflows that uses several middleware-specific Kepler actors (i.e. gLite Actor, UNICORE Actor, etc.) and related submission libraries to access the Resource Allocation Server (RAS) of the fusion community. This server hosts any software necessary to access the different infrastructures in general and Grid middleware adapters in particular.

As shown in Figure 1, two adapters are illustrated that map the functionality of the middleware-specific Kepler actors within the fusion-specific scientific workflow to the corresponding middleware-based requests of the corresponding middleware systems. Hence, a proprietary gLite UI adapter is used to access HTC-oriented resources within EGEE/EGI. In contrast, the vine toolkit is used as another adapter to use the proprietary UNICORE Atomic Services (UAS) [11] and thus HPC-driven Grid resources within DEISA/PRACE. At the time of writing the resources of the corresponding infrastructures are accessible via the EUFORIA Virtual Organization (VO) of EGEE and the EUFORIA Virtual Community (VC) of DEISA. While the usage of compute cycles in EGEE is basically free, the EUFORIA VC got a dedicated amount of agreed compute cycles distributed over the DEISA sites at RZG (Germany), LRZ (Germany), CINECA (Italy), BSC (Spain), and IDRIS (France).

III. FUSION SCIENCE USE CASE APPLICATIONS

The goal of the set of applications mentioned in this section is to simulate aspects of the ITER tokamak, which is a fusion device that may become the basis for future fusion power generating power-plants. Hence, it aims to demonstrate the scientific and technical feasibility of fusion as a sustainable energy source for the future. To exploit the full potential of the device, and to guarantee optimal operation for the device, a high degree of physics modeling and simulation is required - even in the current (construction related) phase of the ITER project. Detailed modeling tools that are required for an adequate description of the underlying physics, are in general very demanding from a computational point of view.

The EUFORIA project enhances modeling capabilities for ITER through the joint use of HTC and HPC resources together with the fusion modeling community by adaptation, optimization and integration of a set of critical applications for edge and core transport modeling as well as turbulence simulations. Most of these application codes allow construction of the complex, combined workflows that produce advance physic results. The numbers of codes of such combined workflows depend on their characteristics and a wide variety of them have been enabled and optimized to be used partly with HPC as well as HTC resources. Because of the paper page restriction, we pick the BIT1 application and the HELENA-ILSA workflow in order to provide a concrete use case of our production interoperability environment. Both are described in detail in the next paragraphs.

BIT1 is a Particle-In-Cell (PIC) code that solves the plasma transport in a divertor. BIT1 is a 1D in real space and 3D in velocity space that takes into account the different relevant phenomena that happen in the scrape-off-layer. It is used to study properties of the plasma-wall transition (PWT) change when the angle is between a wall and the magnetic field and is of the order of, or smaller than, certain values. These values depend on the ion mass and amount of collisions, and on the ion-to-electron temperature ratio in the magnetic presheats [5, 6]. The code was parallelized with MPI and was ported to HPC, profiled and after analysis the code was optimized. In Figure 2 the speed up improvement after the mentioned optimization is compared with the old version of BIT1.

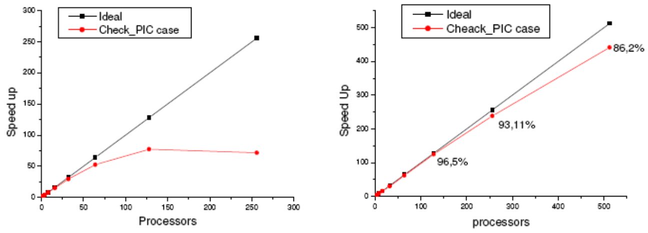

Figure 2. Speed-up improvement before and after the BIT1 optimization.

In addition, the original sequential version of BIT1 was ported successfully to Grid-based environments after several minor corrections. Using a parallel implementation of BIT1, the code has been corrected, profiled and optimized for more efficient parallel computations. This code has been used for performing evaluations since it has been ported to both pure HPC and Grid-based infrastructures. The performance for the realistic runs - the grid version - Average CPU time per Job: 6 days, 12 hours and 23 minutes, Total (Cumulative) CPU Time: 417 days, 2 hours, 32 minutes, 53 seconds.

Another fusion use case application of our framework is HELENA-ILSA, which is one workflow example that includes parameter scan applications. HELENA [7] is a high resolution fixed boundary equilibrium code used to calculate the magnetic flux surfaces in a tokamak by solving the Grad-Shafranov equation. This involves solving a large sparse band matrix equation iteratively. ILSA [8] is a linear Magnetohydrodynamics (MHD) code that is a composite of the former CASTOR, MISHKA1 and MISHKA_D codes (all of them are linear MHD codes with slightly different basic equations). It is used to compute linearly unstable MHD modes (on a mesoscale) in tokamak plasma. Typical modes that can be calculated with ILSA are ballooning modes, peeling modes, kink modes, sawteeth, and neoclassical tearing modes. Also notably, ILSA solves the linearized set of coupled MHD PDEs using various solvers (QR algorithm, inverse vector iteration, Lanczos algorithm, Nyquist method). The major use case for our framework is using the inverse vector iteration and calculated the stability of a ballooning mode in ideal MHD.

Both of the aforementioned codes are used with our framework providing the end-user with a seamless access method to HTC and HPC resources based on Kepler. Figure 3 provides an example screen shot from the end-user perspective showing the Kepler workflow tool. In this particular use case, the HELENA code is firstly computed on HTC resources (using EGEE resources) while its outcome is further used by the ILSA application code on HPC resources (using DEISA).

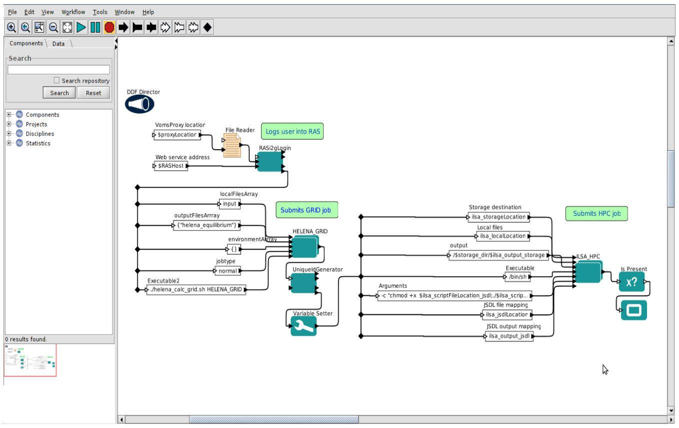

Figure 3. Kepler screen-shot that illustrates the user-perspective of our framework with the fusion HELENA-ILSA workflow using the Kepler tool.

IV. ACADEMIC ANALYSIS OF PRODUCTION EXPERIENCE

We are studying the architectural framework and our use case application experience, because we want to find out how we can improve the efficiency and supportability of our solution in order to enable fusion e-scientists to take a more efficient advantage of interoperable e-science infrastructures and its resources while keeping our solution sustainable over years to come. Although we have been running our framework already in production e-science infrastructure use cases, the outcome of our analysis points to limitations of the framework that are summarized at the end of this section in Table I.

The analysis of the general approach of the framework is the first step to understand overall drawbacks of the approach before we go into details of how specific elements are either supported or not. Riedel et al. provides in [4] a classification of general approaches to reach interoperability and among concepts like Additional Layer, Adapter, Gateway, or Mediator, also the Neutral Bridge approach was defined. Hence, the first major limitation we observed was the use of the well-known neutral-bridge approach to enable interoperability between e-science infrastructures in general and their corresponding Grid middleware in particular. In our case, the neutral bridge is represented by the RAS server that is accessed with the neutral RAS protocol that channels the middleware-specific setups of the Kepler actors through to the bridge.

The problem particularly relies in the transformation logic within the bridge that needs to transform this protocol into either the gLite-specific UI commands or UNICORE Atomic Service (UAS)-specific Web services callouts (cf. Figure 1). This transformation logic basically stands for a couple of drawbacks that are hard to maintain when proprietary interfaces and whole versions of the middleware change over time. In terms of non-functional aspects we can thus conclude that our approach currently taken in production provides less supportability in terms of maintainability and extensibility. The RAS server must be constantly up-to-date with the latest greatest releases of the middleware systems and their versions while preserving older versions (n x versions) in order to not influence ongoing production runs. This clearly leads to high maintenance efforts that are even increased when another adapter might be necessary to work with other production e-science infrastructures like NorduGrid that is using the ARC middleware. Because of the poor extensibility of this approach, also for every new infrastructure support (e.g. TeraGrid with Globus, etc.) another adapter must be provided, deployed, and maintained over time. A desirable solution would be an approach that requires no transformation logic at all based on the 'Open Standards Approach' mentioned as the best solution by the aforementioned approach classification.

Apart from this general approach, a more fine granular academic analysis of our lessons learned provides more clarity and points to specific challenges that we have to address to improve the efficiency and supportability of the EUFORIA framework. Most notably, the RAS server and its middleware-specific entities have not even a (1) common control pattern. This refers to the use of message exchanges that uses principles of Service Oriented Architectures (SOAs) like Web services (WS) that can be seen as the baseline technology within the majority of production e-science infrastructures today. In fact, several 'services' of gLite are still not fully based on WS and thus make the access via common technologies like Web services and SOAP/HTTP difficult. A related limitation was the (2) not commonly agreed message syntax and semantics. Here open standards like OGSA-BES and JSDL define more in detail how the control information is passed in the aforementioned control communication pattern. Although UNICORE adopts JSDL within the proprietary UNICORE Atomic Services this is clearly not enough to promote interoperability since the WS interface remains a non-standard solution and also gLite interfaces do not follow a concrete open standard in our production setup. We define this limitation as no common job control interface. Additionally, gLite is using JDL and not JSDL and thus we identified another limitation by (4) not having a common job description language among the uses Grid middleware setups. In terms of related data-staging activities of computational jobs we clearly found another limitation of having no concrete valid set of data transfer technologies that can be used across the different infrastructures (e.g. GridFTP, HTTP, POSIX, etc.).

Another limitation we observed is (6) the lack of a common information model that addresses the ambiguous terminologies that are used between the production e-science infrastructure EGEE and DEISA and their middleware systems. Differences in the use of terms (e.g. core vs. CPU) across infrastructures was troublesome in our aforementioned fusion use case applications and in many cases the attributes and Grid technology terms that are used are mutually exclusive, conflicting, or have different semantics. Very closely related is the limitation of (7) not having access to this common set of information by using some form of an information system prior to any computational job submissions. This information service typically ensures that up-to-date information about the different resources within the multiple infrastructures can be used by scientists in order to choose the most efficient resource to tackle their problem.

In the particular context of our use case applications BIT1 as well as the HELENA-ILSA workflow we encountered several challenges when accessing the different infrastructures. First, there were (8) no common ways of specifying Grid applications in a meaningful way (e.g. even without specifying the concrete application executable location). Finally, there have been even limitations on the lowest level with (9) having no common way of Grid application execution environments (e.g. environment variables) and (10) a lack of supporting concepts of modern large-scale HPC resources.

TABLE I. OVERVIEW OF LIMITATIONS AFTER ANALYSIS OF APPROACH

No.	Limitation Short Description
(1)	No common control pattern (no common message exchanges)
(2)	No commonly agreed message syntax and semantics
(3)	No common Job Control Interface
(4)	No common Job Description Language
(5)	No concrete valid set of data transfer technologies
(6)	No common Information Model
(7)	No information service query invariant
(8)	No common way of specifying Grid applications
(9)	No common way of Grid application execution environments
(10)	Lack of support of modern large-scale HPC resource concepts

V. USING THE INFRASTRUCTURE INTEORPERABILITY REFERENCE MODEL AND ITS IMPROVED OPEN STANDARDS

To overcome the limitations identified by the academic analysis of lessons learned in the previous chapter, we have worked since several years to improve a dedicated set of open standards towards production usage. Hence, only an approach based on open standards makes it possible to overcome the general limitation of the neutral bridge approach thus avoiding the multiple different middeware-specific adapters as seen in Figure 1 as part of the RAS server. Using open standards for our EUFORIA framework makes it possible to realize more flexibility so that e-scientists are even able to choose the technologies they desire (e.g. prevent vendor-locks). But open standards alone are not sufficient as the next paragraph will reveal by describing numerous improvements of open standards based on outcome of production experience analysis and lessons learned. These improvements of open standards are typically indicated by using ΔX, ΔY, or ΔZ that all stand for the specific improvements that we applied to the open standards.

We start by introducing the overall new approach: we overcome the limitations mentioned in Table I with a set of improved standards and their inter-relationships that have been collectively defined earlier by Riedel et al. as the infrastructure interoperability reference model (IIRM) [4]. This reference model represents currently the roadmap of the Open Grid Forum (OGF) Production Grid Infrastructure (PGI) working group that acts as a channel to numerous other working groups (e.g. OGSA-BES, JSDL, GLUE2, etc.) thus bringing our multiple additions to the open standards into the next generation of specifications. It is worth mentioned that this group consists of member of a wide variety of middleware providers that altogether are in the process of defining and adopting the PGI set of standard improvements. While the first set of these specifications were majorly academically-driven, we clearly emphasize on production experience and needs with this second set of specifications making it thus less likely that other new standard versions must be defined in the near future. In the following paragraphs we will address the limitations that we described in the last Section by comparing Figure 1 with Figure 4, which is a setup based on the IIRM approach.

We overcome limitation (1) by using commonly service oriented architectures in general and Web services in particular. As shown in Figure 4, gLite provides a Web services that adopt the OGSA-BES interface while also UNICORE adopt the same interface. In both cases ΔX stands for some refinements of the interface that are kept out of this section for clarity since they are not being directly related to the EUFORIA framework setup. Since we use the open standard OGSA-BES in each Web service call we thus overcome limitation (2) and also (3), because OGSA-BES is the common job control interface we desire. The use of OGSA-BES implies typically the usage of a the JSDL job description language and as we observe in Figure 4, both middleware systems used by the EUFORIA framework are willing to adopt this standard after applying certain refinements that we refer to as ΔY. As a consequence, we overcome limitation (4).

A more complicated aspect is limitation (7) that we overcome by putting the invariant that the vine toolkit first queries an information service before performing any job management activities. This invariant is also part of the reference model since the use of up-to-date information is a crucial aspect in interoperability of multiple production e-science infrastructures. Very related is the use of GLUE2 as the agreed common information model to overcome limitation (6) and provides a common basis for information services that in turn are queried as described above by the vine toolkit. Here, one feature in context is that the information obtained provides typically information about the supported data-staging technologies thus leading to a solution of limitation (5).

As part of the IIRM, we define several additions (i.e. ΔY) [13] to the JSDL language that represent a common application software concept that re-used GLUE2 elements and also not necessarily requires low-level executables. With this element the EUFORIA framework overcomes limitation (8). Closely related to this, the IIRM defines further additions (i.e. ΔY) [14] to JSDL that provide a common way of Grid application execution environments thus addressing limitation (9).

A wide variety of improvements of open standards have been applied to JSDL and GLUE2 in order to overcome limitation (10). They all increase the support of modern large-scale HPC resources and thus are of valuable use of the EUFORIA framework within DEISA today, but more notably also within PRACE tomorrow. Some of these improvements are network topology support of HPC resources (e.g. torus, mesh, global tree, etc.), shape reservation (X x Y x Z), and task/core mapping definition to list but a few.

Finally, the features of the reference model in terms of security are realized by using the so-called plumbing concept as described in [12]. Because of the page restriction we cannot provide much more information, but the general approach is based on X.509 proxies that are supported by gLite and also optionally more recently by UNICORE.

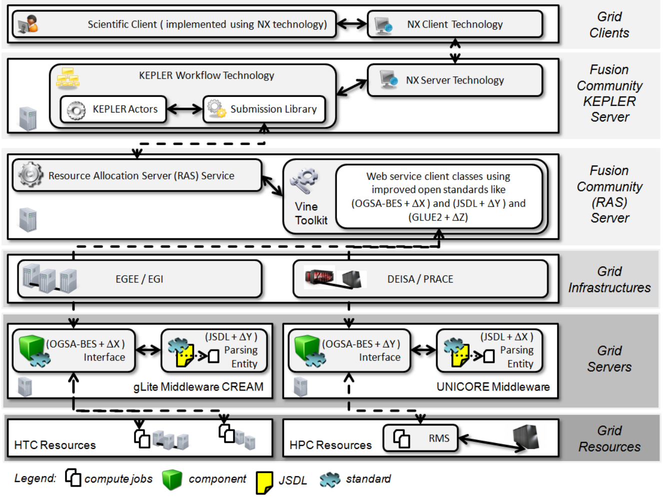

Figure 4. Improved open standards of the infrastructure interoperability reference model lead to a maintenainable and thus sustainable EUFORIA framework for fusion scientists providing HTC and HPC resources.

VI. RELATED WORK

Related work in the field of standardization is found among the members of the OGSA-BES, JSDL and GLUE2 OGF working groups. Several other ideas and concepts arise also from the work of these members and we have discussed and will discuss in future of how we can align our work in order to have new set of specifications that not fundamentally change the existing specifications and thus just improving them without break their emerging stability. To the best of our knowledge there are no other fusion scientists except the ones that collaborate with us in EUFORIA/ITER/ITM and that work with us on improving open standards in order to use the European e-science infrastructures EGEE and DEISA and their Grid middleware systems more efficiently.

Related Work in the field of reference models typically leads to the Open Grid Services Architecture (OGSA) [14], which has in comparison to our approach a much bigger scope. Hence, our approach only represent a subset of this scope but is more focused and thus more detailed. We deliver with our reference model a much more detailed approach of how open standards can be improved and used in scientific applications that require interoperability of e-science production Grids like the fusion community. Neither this contribution nor the reference model in the bigger context aim at replacing OGSA and thus rather represent a medium-term milestone towards a potential full OGSA compliance of Grid middleware systems in future. In comparison with the former commercially-driven Enterprise Grid Alliance Reference model, our model is clearly oriented to support rather scientific-based use cases.

To the best of our knowledge, there are no other fusion science frameworks that apply a greater standards-based reference model to access multiple production e-science infrastructures.

VII. CONCLUSIONS

We have introduced our architectural production setup and sketched two major fusion application use cases that in turn lead to numerous lessons learned about the limitations of the current framework. We have surveyed this set of limitations and addressed each one by using the concepts of the infrastructure interoperability reference model (IIRM) that we earlier defined and currently standardize within the OGF PGI group. We thus have shown that our work on the IIRM was not only driven by the EUFORIA framework, but also can be used to overcome the limitation of the current production framework. We thus can conclude that the framework based on the IIRM leads to a more maintainable (i.e. no transformation logic and middleware adapters) and extensible solution (i.e. use any adoption of open standards) while also enabling a more efficient use of production e-science infrastructures resources. The latter is especially the case in HPC-driven environments where efficiency is often one of the most important aspects.

ACKNOWLEDGMENT

We thank the OGF GIN and PGI groups for fruitful efforts and discussions in the context of this work. This work is partly funded via the DEISA-II project funded by the European Commission in FP7 under grant agreement RI-222919, and the EUFORIA project funded by the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement RI-211804.

REFERENCES

[1] M.Riedel and M.S.Memon et al., "Improving e-Science with Interoperability of the e-Infrastructures EGEE and DEISA", Proceedings of the 31st International Convention MIPRO, Conference on Grid and Visualization Systems (GVS), May 2008, Opatija, Croatia, Croatian Society for Information and Communication Technology, Electronics and Microelectronics, ISBN 978-953-233-036-6, pages 225 - 231

[2] M. Riedel et al., "Exploring the Potential of Using Multiple e-Science Infrastructures with Emerging Open Standards-based e-Health Research Tools", Proceedings of the The 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010), May 17-20, 2010, Melbourne, Victoria, Australia, to be published

[3] B. Guillerminet and I. Campos Plasencia et al., "High Performance Computing tools for the Integrated Tokamak Modelling project",Fusion Engineering and Design, 2010, DOI: 10.1016/j.fusengdes.2010.01.003

[4] M. Riedel et al., "Research Advances by using Interoperable e-Science Infrastructures - The Infrastructure Interoperability Reference Model applied in e-Science", Journal of Cluster Computing, Special Issue Recent Advances in e-Science, Cluster Comput (2009) Vol. 12, No. 4, pp. 357-372, DOI 10.1007/s10586-009-0102-2, December 2009

[5] D. Tskhakaya, S. Kuhn, "Particle-in-cell simulations of the plasmawall transition with a magnetic field almost parallel to the wall", Journal of Nuclear Materials 313–316 (2003) 1119–1122

[6] D. Tskhakaya, R. Schneider, "Optimization of PIC codes by improved memory management", Journal of Computational Physics 225 (2007), pages 829–839.

[7] Huysmans, G.T.A. et al., Proc. CP90 Conf. on Comp. Physics Proc. (1991)

[8] Huysmans, G.T.A. et al., Phys. Plasmas 8(10), 4292 (2001)

[9] I. Altintas and C. Berkley and E. Jaeger and M. Jones and B. Ludascher and S. Mock , "Kepler: an extensible system for design and execution of scientific workflows", Scientific and Statistical Database Management, 2004. Proceedings of 16th International Conference on. Pages: 423—424,

[10] E. Laure et al.: Programming the Grid with gLite, In: Computational Methods in Science and Technology, pp. 33–46 (2006)

[11] A. Streit et al. "UNICORE 6 - A European Grid Technology", L. Grandinetti, G. Joubert, W. Gentzsch (Eds.), Trends in High Performance and Large Scale Computing, IOS Press

[12] M. Riedel et al. "Towards Towards Individually Formed Computing Infrastructures with High Throughput and High Performance Computing Resources of Large-scale Grid and e-Science Infrastructures", in proceedings of GVS Workshop, MIPRO 2010, Optija, Croatia

[13] M. Riedel et al., M.S. Memon, A.S. Memon, A. Streit, F. Wolf, Th. Lippert, B. Konya, A. Konstaninov, O. Smirnova, M. Marzolla, L. Zangrando, J. Watzl, D. Kranzlmüller, "Improvements of Common Open Grid Standards to Increase High Throughput and High Performance Computing Effectiveness on Large-scale Grid and e-Science Infrastructures", Seventh High-Performance Grid Computing (HPGC) Workshop at International Parallel and Distributed Processing Symposium (IPDPS) 2010, Atlanta, USA

[14] I. Foster et al.: Open Grid Services Architecture (OGSA), Version 1.5. OGF Grid Final Document Nr. 80 (2006)