On Realizing the Concept Study ScienceSoft of the European Middleware Initiative

Open Software for Open Science

Alberto Di Meglio
European Center for Nuclear Research
CERN
Geneva, Switzerland
Alberto.Di.Meglio@cern.ch

Morris Riedel
Juelich Supercomputing Centre
Forschungszentrum Juelich
Juelich, Germany
m.riedel@fz-juelich.de

Abstract—**In September 2011 the European Middleware Initiative (EMI) started discussing the feasibility of creating an open source community for science with other projects like EGI, StratusLab, OpenAIRE, iMarine, and IGE, SMEs like DCore, Maat, SixSq, SharedObjects, communities like WLCG and LSGC. The general idea of establishing an open source community dedicated to software for scientific applications was understood and appreciated by most people. However, the lack of a precise definition of goals and scope is a limiting factor that has also made many people sceptical of the initiative. In order to understand more precisely what such an open source initiative should do and how, EMI has started a more formal feasibility study around a concept called ScienceSoft – Open Software for Open Science. A group of people from interested parties was created in December 2011 to be the ScienceSoft Steering Committee with the short-term mandate to formalize the discussions about the initiative and produce a document with an initial high-level description of the motivations, issues and possible solutions and a general plan to make it happen. The conclusions of the initial investigation were presented at CERN in February 2012 at a ScienceSoft Workshop organized by EMI. Since then, presentations of ScienceSoft have been made in various occasions, in Amsterdam in January 2012 at the EGI Workshop on Sustainability, in Taipei in February at the ISGC 2012 conference, in Munich in March at the EGI/EMI Conference and at OGF 34 in March. This paper provides information this concept study ScienceSoft as an overview distributed to the broader scientific community to critique it.**

Keywords-software; maintenance; collaboration; market-place; sustainability; middleware

I. Introduction

There is a wealth of open source software in use across scientific communities but the value of its contribution to science is under-estimated, under-utilised and often poorly coordinated. Some websites such as ohloh [2] offer directories that attempt to rate the quality and impact of open source software projects, but currently lack the means of attracting developers and users from academic communities and harvesting a large enough body of essential data to make their results meaningful for the scientific research environments. Being able to aggregate the power of cataloguing services, trends and statistics would provide a sound basis for judging the popularity of specific software enable social-networking amongst users and developers, create active communities and promote citizen science. Rating software and providing a means by which it can be cited in a similar manner to publications and datasets would enable the authors to gain merit and career advancement for their work and accelerate the open source software movement in scientific communities. Being able to quantify the impact of open source software would allow funding agencies, companies and venture capitalists to better target their investments leading to a more vibrant and sustainable open source market for open science.

The European Middleware Initiative (EMI) [3] is delivering open source software and is a joint project of the middleware technology partners around ARC, dCache, gLite, and UNICORE. Recently, EMI products in particular and Grid computing in general contributed to the success of eventually finding the ‘higgs boson’ [1]. In terms of sustaining its activities, a new initiative named as ScienceSoft [4] emerged that is well embedded in the broader overall future strategies of the project. This strategy aims to engage with new markets that go beyond the traditional research Grid and High Performance Computing (HPC) market.

This paper outlines the state-of-the-art of open source activities in the scientific research environments and describes a number of problems and potential solutions identified by discussing with user and developers of existing projects and communities around the distributed computing ecosystem in Europe and beyond. The potential benefits of the proposed solutions are described and how scientists, developer, administrators, managers and funding bodies could exploit them to take decisions and plan their activities.

This paper is structured as follows. After the introduction in Section I, the EMI project is briefly introduced in Section II with a focus on its future strategies where ScienceSoft is one crucial part of it. Section III then list identified key challenges and problems in context of scientific software development today, while Section IV lists a potential set of solutions. The ScienceSoft concept study itself is introduced in Section V and Section VI briefly surveys related work in the field. The paper ends with some concluding remarks.

II. Future Strategies of the EMI Project

The activities of exploitation and sustainability (E&S) of the EMI software are of paramount importance for the success of the project. Exploitation represents the measure of how successfully the software products are used not only by EMI users, grid sites administrators, application developers, National Grid Initiative (NGI) managers, but also by the project partners themselves in their future activities. Sustainability gives the measure of how well the EMI project has managed to create, support and develop existing and new markets through a proper development of its traditional channels and the exploration of new channels. The creation of value and the continuous innovation of the EMI ideas and products are the ultimate goals that drive the plans.

After the initial explorations in year 1 of the concepts of exploitation and sustainability and the definition of a generic set of drivers, it became clear that a number of changes in the project structure and in the focus of some of its activities were needed. Similar recommendations were the result of the first year EMI review. The creation of a dedicated activity around sustainability and long-term strategies was performed in order to foster future plans. On the wave of the changes, the E&S team in the second year has actively worked on the definition of an ambitious, but realistic exploitation and sustainability plan, drawing ideas and support from existing and new collaborations with other projects, user communities, commercial companies and from existing open source community models. This paper highlights some parts of it.

The EMI E&S strategy finds its roots in the core business of providing the best possible support for European research grids. The relationship with EGI, PRACE, WLCG, LSGC application developers and user projects in this domain have been expanded and strengthened by establishing formal collaboration agreements and technical cooperation. Important parameters as the cost of supporting and developing the EMI products, the core business plans of the EMI partners, the available and expected funding sources have been analysed and discussed. Long-terms support plans have been defined and shared with major stakeholders.

The core market as illustrated in Figure 1 and represented by the European infrastructures and the researchers using them must be preserved and kept efficient and operational. These activities require as expected most of the EMI effort, energy and funds. At the same time, the fact that the research grid market is a niche, mature market with a relatively moderate growth must be acknowledged. The EMI E&S activities have therefore been also focused in this second year in understanding how innovation can be better sustained and what new markets could provide the highest chances of returns on investing the available limited effort. Based on the ideas already explored in the first year, the EMI sustainability team have decided to focus on two distinct areas, namely the implementation of at least one good example of commercial exploitation of some of the most promising EMI services and the creation of wider-scope open source initiative for science with the potential of serving the longer-term goals of the European Commission (EC) open science strategies. Both are illustrated in Figure 1 alongside the traditional market.

The first objective, the commercial exploitation, has resulted in the collaboration with dCore Systems [5], a Luxembourg-based holding of a number of high-tech SMEs. The company (previously known as The Syrrus) has an ambitious business plan based on the concept of commercial distributed services that fits very well the distributed nature of the EMI products.

The second objective, the creation of wider-scope open source initiatives, has resulted in the creation of ScienceSoft, a vertical open source community dedicated to software for scientific application. The ideas behind ScienceSoft have been widely discussed and presented to as large an audience as possible at this stage. A plan for its further implementation has been agreed and it’s being followed. This paper aims to shed light on these particular activities while some of its activities are still ‘not clear’ in the sense that ScienceSoft itself is still in a concept phase moving towards a more production status.

The illustration in Figure 1 summarizes the overall EMI exploitation and sustainability activities that have led to the current state of the art and how the different parts fit together in a broad-scoped strategy that will guide EMI for its third and final year and the EMI partners for the next two to three years. The subsequent sections reveal some insights about these future strategies.

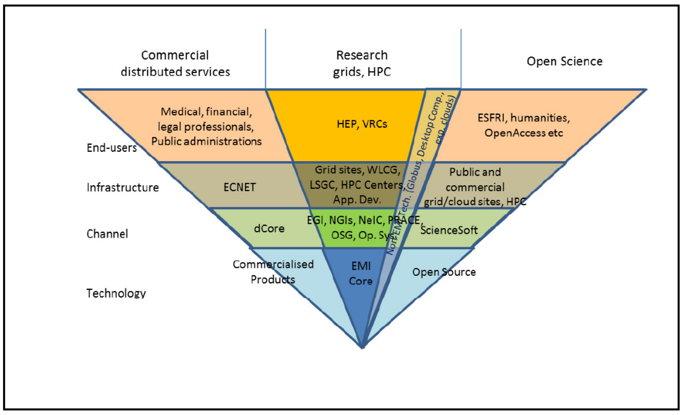

Commercial distributed services
Research grids, HPC
Open Science
End-users
Infrastructure
Channel
Technology
Medical, financial, legal professionals, Public administrations
HEP, VRCs
Non-EMI Tech. (Globus, Desktop Comp., exp. clouds)
ESFRI, humanities, OpenAccess etc
ECNET
Grid sites, WLCG, LSGC, HPC Centers, App. Dev.
Public and commercial grid/cloud sites, HPC
dCore
EGI, NGIs, NeIC, PRACE, OSG, Op. Sys.
ScienceSoft
Commercialised Products
EMI Core
Open Source

Figure 1. The Open Science market and its relation with other EMI markets

A. *Model Generation and Market Drivers*

The EMI exploitation and sustainability strategy requires an understanding of the EMI markets and users. As part of the implementation of the strategy, a market analysis has been performed using a model composed of as a set of stacked layers, each one with associated responsibility and scope that is summarized in Figure 1 while the layers are defined as follows.

Technology: This layer represents EMI and its products and services and other technology providers. Within the scope of the EMI plan this is essentially the set of middleware services that provide distributed compute and data functionality.

Channel: this is the means that EMI uses to distribute its product to its users. It can be considered a distribution channel made of entities providing entry points to the infrastructures where the services are deployed.

Infrastructure: this is where the EMI services are deployed or directly used. This layer is made by the projects, initiatives and people deploying, managing and monitoring the services and the resources where the services are deployed. It includes also the communities of grid application developers directly using the EMI Application Programming Interfaces (APIs) to produce the high-level services used by end-users. This layer is the actual reference market segment for EMI.

End-users communities: this is the layer made of the professionals, researchers, scientists doing work using the infrastructure services. This layer is composed of several different types of users, from high-level domain-specific application developers, to the scientists having (and wanting to have) little or no knowledge of how things work under the hood.

Market domain: this is a specific market defined in terms of its scope, composition, mission, requirements, etc. and structured by the above-described layers.

An important concept in the proposed EMI layered stack concept is the '*amplification factor'* (the size of market impact) that each layer provides for the layer above. The actual entities in each layer must be selected to provide an added value so that a relatively small improvement in the technology layer can be turned in a large improvement in the domain market size. This is something that requires a realistic verification of the assumptions and continuous collaboration among the different entities.

The current and potential EMI markets and the work done in terms of creating, maintaining and extending collaborations is illustrated in Figure 1. Three market domains have been identified and exploited to different extents. The market domains are (1) Research Grids and High Performance Computing, (2) Commercial distributed services, and (3) Open Science.

Given the above-mentioned definition of a Market domain, the concepts of Exploitation and Sustainability can be specified in this paper as foundation as follows. Exploitation is defined as the use of the EMI products and services by the layers above the Technology layers and by the EMI partners themselves as part of their core business. Each layer has different exploitation characteristics and different expected and actual results based on the EMI Exploitation plan.

Sustainability is defined as the capacity of producing continuous innovation, enabling the layers above to provide increasingly better services and to be willing to pay directly for that or to support and require the EMI partners to seek additional funding in the future. Also in this case the values that each layer perceive as critical are different and so are the expected and actual results of the EMI exploitation plan.

B. Open Science Market Analysis Model

During the discussions about what can generate innovation and produce sustainability, EMI contacted many projects, infrastructure, communities and commercial companies' representatives to share problems and ideas. Over the past 12 months it became obvious that most of the people interviewed on the subject share a common set of problems. Some of these problems are more felt by software developers, others are ore felt by users. However, they all have common elements of lack of communication, lack of information, lack of interaction across different projects, different communities, and different scientific disciplines.

In most cases people from different contexts meet and talk in occasion of conferences and other similar events. However the level of interaction is often limited to a presentation followed by a few question. Many issues are briefly discussed, but are not efficiently followed up, recorded, solved.

As far as software development and usage is concerned, there are limited ways of understanding what software exists, who is using it, what they think about it. Despite the fact that all software produced is released under a valid open source license, the structure that animate a community around that software are currently missing in most of the production of scientific software.

The cases were software is released via email from a developer to an end-user to be soon forgotten are not rare. Good quality software has limited chances to be publicly praised by satisfied users and bad software often survives because users have limited ways of clearly saying that it should be stopped. Many scientific publications describe fundamental results obtained in many different fields and cite past related papers with fully searchable identifiers that increase the academic rating of their authors. However, the software used to generate those results is often mentioned in the text of paper as a note or little more.

The successful open source initiatives do organize conferences and events, but have mechanisms in place by which community members can interact on a daily base to exchange information, discuss problems, find help from other users, and organize ad hoc technical interest groups and so on. Due the traditional structure of limited duration projects typical of the research infrastructures, such mechanisms are very difficult to set up and maintain after the end of the project. In turn the end-user communities find it difficult to fully commit to rely on the software produced by projects without long-term operational lifetimes.

In order to change what is seen by many as a status-quo, a different approach has to be taken. EMI has started the process of adopting standard open source methodologies and will release by its end date most of its products in standard operating systems repositories. However, the missing element of an active open source community must also be established. Interactions within the community are stronger if supported by motivated individuals within the more general interests of institutes or companies. Of course both developers and users must see advantages in being part of the community, in terms of more visibility, better support, career enhancements, personal technical interests, etc.

Institutes in turn have to implement policies to encourage this behaviour and take advantage of the possibility of having themselves more visibility and better tools to prove their value in the research community and therefore increase their chances to be funded to continue providing that value.

III. Identified Key Challenges and Problems

Most of the software developed today by research institutes, university, research projects, etc. is typically stored in local source and binary repositories and readily available for the duration of the project lifetime only. Finding software based on given functional characteristics or field of application is very difficult especially for new projects or young researchers. Binaries to be run on the most used operating systems are available from many different places ranging from local university repositories to mainstream community repositories like EPEL [6] or Debian. Cases of conflicts are often found between different versions distributed by different people from different places. Source code is even more difficult to locate and access and contributing with comments, patches and fixes, which is a very common activity in the open source world, is traditionally very difficult to do in the research communities. This has been for years a primary complaint from users.

Similar requirements have been expressed by application developers, infrastructure managers and users. Within the HPC community the organization of common repositories of application code is a known concern. Although such code is usually highly dependent on the hardware on which they are run, the lack of code sharing and availability is considered one of the reasons why the European HPC efforts are falling behind similar activities in the US and Asia (see the recent IDC report presented at the EGI Technical Forum in Lyon in September 2011). Most of the reported problems can be categorized as a lack of consistent and transparent information about the software being used in scientific research. The problem is not necessarily a lack of technical information (such as documentation or user guides, although this has also been described as a problem in many cases), but rather a lack of metadata.

Information about who develops, contributes and uses a given program is very difficult to find out and yet the widespread availability of such information would give more visibility and credibility to the software products. In addition, the EC invests considerable amounts of money into funding projects that directly or indirectly need to develop software. A single repository of metadata information about software products would allow projects to avoid re-developing existing solutions and would provide valuable statistics about software usage. Such information could be used also by the EC to monitor the outcome and impact of funded projects, the extension of adoption of open source software and the compliance with Open Source Initiative licenses [7] and possible as input to future EC calls objectives and framework programmes. In the same way, the information could give more strength and credibility to project proposals, which could be backed by realistic information about usage, impact and exploitation of the software.

Users and developers of software for scientific research projects have been contacted during conferences and presentations and asked about what problems they are confronted with in their normal working activities. The identified problems and challenges identified as most critical are listed further in the following sub-sections.

A. *Lack of Continuity*

The lack of continuity in support, development, coordination of software is a hindrance in supporting science today. This issue is mostly felt by users and developers of software developed by short-term projects. Users face difficulties in adopting or relying on software without longer term support commitments. At the same time developers, who may be willing to continue developing and supporting the software, find themselves without a release channels and have no easy way of keeping contact with their users.

B. *Non-optimal Communication*

A non-optimal communication between users and developers is often experienced. Most projects tend to be rather horizontal and focus on specific, limited aspects of the scientific research software stack. Although this is a quite natural approach to setting up projects, it has the drawback of creating boundaries between different categories of users. The interaction happens often in occasion of conferences or other public events, but many users have expressed the need for a more direct and continuous relation with the developers in order to discuss requirements, features, issues, etc. It is felt that there is excessive formalization and loss of information due to being in different projects.

C. *Lack of Consistent Real Usage Information*

It is easy to count downloads from a web site, but information on actual usage is more difficult to collect. Information systems are partly available to provide this data, but not for all services and applications. The lack of this information prevents understanding the actual user base and evaluating the relative importance and impact of the produced software.

D. *Limited Access to other Users Experience*

Before using an application or a software service and investing time and money in them, users would like to know what other people think of those applications or services, what problems have been found, if and how they have been solved. Problems solved in one case can be applied in other cases and save time for both users and support people.

E. *Limited ways of Finding Existing Software*

Before using an application Limited or complex ways of finding what exists already. A lot of duplication is often found in software and due to real needs of producing something that fits better a community need, but quite often it is due to the lack of knowledge of what exist already. Having information about existing software and services, who develops and supports them and who uses them may help taking informed decisions and quickly bootstrap new projects without waste of time and resources.

F. *No way of Influencing the Production of Software*

Many users feel that they have little saying on what functionality software should have or what software should be supported and which should be stopped. Although functionality and quality are always a matter of negotiation and priorities, the lack of the possibility of expressing opinions often drives users away or prompt them to develop their own solutions in the hope of having more control.

G. Lack of Visibility

The Lack of visibility of the software activities are also a key problem identified by many projects and developers. Software developers working in research institutes produce fundamental tools used by scientists to make discoveries. Academic recognition is usually measured in terms of publications and the contributions of programs and tools cannot be formally acknowledged with a reliable citation system. Having formal ways of citing and listing software used in conjunction with published scientific results would increase motivation and help career recognition and development.

H. No Way of Accessing the User Market

The Lack of visibility of the software activities are also a key problem identified The main of foremost barrier to the involvement of commercial ventures into the development and support of software and services for scientific research is the difficulty of sizing the market and the potential revenue streams. Having ways of performing realistic market analysis and offering targeted services to the scientific communities may help SMEs if not larger companies to define achievable business plans.

IV. Possible Solutions and Benefits

From the discussions with users and developers and the outcome of the CERN Workshop in February, a number of desirable functionality has been defined that is presented in this section as some possible solutions to the problems and challenges listed in Section III.

The subsequent solutions lead to several benefits that we provide in advance to motivate and better understand the ideas behind some of the solutions. The creation of links or relationships not only among pieces of software, but equally among the people interacting with the software, would foster a more active community and create the conditions for sharing ideas and skills and a more rapid improvements of the software quality.

The use of modern social networking techniques would greatly help the establishment of active open source communities and focused sub-communities around specific scientific and technical interests. For example, sub-communities could be established for people interested in testing or in writing documentation, communities of packagers, or experts of specific standards or security and so on.

The possibility of sizing and profiling the usage of software by user communities would potentially allow commercial companies to offer added-value services based on concrete needs.

The availability of value-added information about software and its usage would bring several benefits. It would allow Institutes to perform more realistic assessment of costs and optimize resources by focusing on unique propositions rather than duplicating existing software. The possibility of concretely sizing the user base of an application or a service would provide tangible supporting evidence for funding requests with concrete impact analysis. Open source software licenses adoption and compliance could be verified in order to enforce legal requirements.

The establishment of a software rating system based on both technical criteria (Is a given platform supported? Is a certain package format available?) and usability criteria (What do existing users think of it? Is documentation up-to-date and well written?) would allow filtering mature products from less mature products and would increase the developers motivation to improve certain aspects of their software like documentation that traditionally receive less effort that the actual code writing.

An interesting functionality is the possibility of creating community-specific virtual software stacks using the software catalogues. Once the profile is defined, it can be kept updated as the products evolve and dedicated community integrator could provide pre-packaged appliances to be shared for example through appliance marketplaces like the one implemented by StratusLab.

A. Software, Services, and People Catalogues

The first and foremost desired functionality is the provision of catalogues of information about software products, software-related services and people. The catalogues should provide information based on tags or taxonomies that allow to group together sets of related products, services and people based on flexible search criteria. The provision of technical metrics about software and services is also desired, for example license, programming languages, compatibility, supported systems, etc.

B. Generation of Statistics

The information collected and processed should not only be used to search about software, but also to general relevant and useful statistics. The most requested statistics are for example actual usage information, as opposed to downloads of packages from a web site, geographical distribution of usage and production, involvement of Institutes and Companies in projects based on scientific discipline, etc.

C. Honour System

The information community users should be able to rate the registered software and services based on their experience. Ratings can be provided based on predefined categories such as reliability, support quality, documentation, ease-of-use, standards support, etc. Ratings should be supported by comments describing the user experience with the software or service.

D. Citation System

The citation system should allow software to be referenced in papers: registered software should receive some sort of unique identifier like the DOIs [8] used for papers, so they can be reliably cited in scientific publications.

E. *Marketplace*

The citation system Marketplace for products, services, and people: this is one of the most interesting features and one that may well define a community. Matching demand and offer of software products, service and people skills should be enabled based on the catalogues maintained by the community tools. Some form of advertising could be considered both for non-profit and for profit activities, although different models should be envisaged in this case.

F. *Links to Technical Services*

One of the marketplace-related activities is the provision of technical services. A range of services could be designed and provided by community members and provided to other members for free or for a fee depending on conditions. Such services could go from straight access to Infrastructure as a Service (IaaS) resources, to software testing services, to consultancy and support, etc. Services can be generic or community-specific. The common requirement is to provide a way to advertise, promote and access the services.

G. *Platform Integration Support*

By using the collected information and the product catalogues, it should be possible to define community-specific software stacks to be supported by platform integrators. These community-specific profiles or stacks can then be pre-packaged and easily deployed using the more and more standard virtualization and cloud technologies.

H. *Ad-hoc Community Creation*

A system is required that support the creation of ad-hoc community and groups that may actually define the system itself as a kind of ‘community-enabler’. The members of the system per se should act as a super-community and provide a framework and tools to enable more specific communities to interact without being isolated from other communities. Cross-community groups, for example for common activities like security, configuration, etc. could also be created and populated with the relevant products and people.

I. *Coordination, Collaboration and Discussion Tools*

Collaboration and coordination tools typical of distributed online infrastructures should be available through a kind of community portal. It should be possible for each hosted community to have independent channels for discussions, but a general social network for science should be supported allowing direct interaction among users in addition to standard forums.

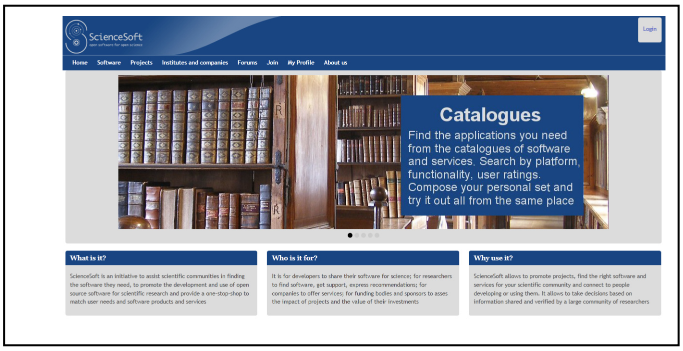
J. *Organization of Technical Events*

A system is needed that supports the organization of technical events. It should be possible to advertise and manage technical events related to the hosted communities or projects. Support can range from dedicated event pages, to agendas, advertising, collection of material, etc. Where possible, it would make sense to co-locate these events with domain-specific science events that might drive and steer some of the activities of the system while also being one of its customer.

V. ScienceSoft Concept and Governance

The core functions of ScienceSoft aim to implement many of the solutions presented in Section IV. Nevertheless, it should be clear that each of the individual possible solutions are driven from the community needs over time and thus while some functions become perhaps very prominent others might fully disappear or reappear in several years.

The ScienceSoft community portal [4] as shown in Figure 2 gives access to the community services like member and product registration and management and the different functionality described earlier in the document.



Catalogues

Find the applications you need from the catalogues of software and services. Search by platform, functionality, user ratings. Compose your personal set and try it out all from the same place

What is it?

ScienceSoft is an initiative to assist scientific communities in finding the software they need, to promote the development and use of open source software for scientific research and provide a one-stop-shop to match user needs and software products and services

Who is it for?

It is for developers to share their software for science; for researchers to find software, get support, express recommendations; for companies to offer services; for funding bodies and sponsors to asses the impact of projects and the value of their investments

Why use it?

ScienceSoft allows to promote projects, find the right software and services for your scientific community and connect to people developing or using them. It allows to take decisions based on information shared and verified by a large community of researchers

Figure 2. Science Soft Portal free for everybody to login and to use.

Based on the preceding list of possible solutions therefore ScienceSoft could be configured as a community of users and developers of software targeted at scientific applications. The community members contribute software and information and provide services to other members. The members within ScienceSoft are organized in focused communities or collaborations around a specific scientific or research topic. People, software, services, etc. are tagged within the super-communities based on their particular focus in order to build community-specific sets of resources. Resource tagging is not exclusive. The same resource can be tagged with more than one community focus, so that it becomes possible to understand the overall usage of that resource within a more general scientific context.

Most of the common base functionality required to operate the ScienceSoft portal already exists in some form. As a first implementation, the ScienceSoft portal can be configured as an aggregation of such functionality within a coherent container. Functionality like software inventories, source-code repositories, social networking, forums, etc., can be provided in this way using existing open source services like ohloh.net [2] or inventories, Drupal modules [9] for the web based collaboration tools and existing social networks like FaceBook, Linkedin or Google+ for user management.

The community specific services would instead be provided by the members through links or applications running in the portal. Community-specific micro-sites can be easily established from common templates to create well-defined identities and focused sets of people, programs, services, tools, information, etc.

The governance structure for ScienceSoft could therefore be based on two general roles. Firstly, the *ScienceSoft maintainer's role* stands for maintainers that are in charge of implementing and supporting the portal and the base common features by integrating as much as possible existing functionality from other compatible open source initiatives, where compatible means that the license and usage rights must allow the integration within the ScienceSoft portal. The maintainers could also provide support for general items like community templates, although it can be envisaged that such functionality would be also contributed by other community members.

Secondly, the *ScienceSoft contributors* are any ScienceSoft community member providing information, software or services to ScienceSoft or one or more communities hosted within ScienceSoft. This category includes the people responsible to defined and maintain collaborations or communities within ScienceSoft. For example the EMI Collaboration, the Grid and Cloud Security Group, the Earthquake Modelling and Prediction project could be hosted communities with one or more person managing their definition, memberships, activities, software stacks, etc.

Although the initial organization and governance structure of ScienceSoft should be as lean and lightweight as possible, it is foreseen that depending on the success and evolution of its activities it could become in the future a Not-for-Profit foundation on the model of existing successful open source foundation.

The requirements and possible implementation strategies for Science Soft are currently being investigated by the ScienceSoft Steering Committee. Initial requirements and desired functionality have been discussed in occasion of the ScienceSoft Workshop held at CERN in February 2012. The outcome of the workshop has been incorporated in this document. The workshop participants have agreed to keep providing feedback and collaborate in the definition of implementation priorities. A proposed timeline is given in Table I.

TABLE I. SCIENCE SOFT IMPLEMENTATION TIMELINE

Time	Actions
March to June 2012	Definition of priorities and identification of volunteers ScienceSoft maintainers.
July to December 2012	Progressive implementation of a prototype community portal, dissemination, engagement of scientific communities in trying the functionality and providing feedback.
January to April 2013	Start of the regular activities, further requirements and implementation cycles. Until this date the community is incubated with the EMI project, which provides overall coordination
May 2013 onward	Regular operations, fund raising for continuing activities based on the success of the initiative. Phase down and discontinuation if no interest has emerged.

VI. RELATED WORK

Related work in the realm of ScienceSoft can be considered in two major directions. Firstly, lessons learned from the open source software community and its different modus operandi. Secondly, the wide variety of existing international collaboration platforms and tools that all have similar or same functions as those identified above. It is obvious that for the second part we cannot provide a full list of related work but aim to provide a few well known representatives.

The first part addresses the described issues and provides the desired functionality to exploit lessons learned from successful open source software communities. Most of the software used in scientific research and developed by academic institutes is generically ‚open source‘ in the sense that it uses some type of OSI license. However, it takes more that source code and license to have a ‚community‘. The general definition of an open source software community is a group of developers and users interacting to produce free-software. The interaction among users and developers, the sharing of resources and common objectives and the benefits deriving from sharing are of course fundamental to have a community and not just software in a repository. We can distinguish primarily between four different types of open source communities that are surveyed as part of related work.

The *‚technology-specific (or horizontal) projects:* this type of communities includes projects focused around a specific technology or framework which all members contribute to. Usually the membership rules are quite stringent both in technological and legal terms. For example it's not uncommon to have to adopt a mandated IP model and a license for all contributing products. Notable examples of this category are the Apache Foundation, the Eclipse Foundation or the Drupal Association.

The *operating system distributions:* communities focused around different flavours of Linux operating systems have been among the first to emerge and have in many cases enabled most profitable open source business models. Although in general there is no formal membership into these communities, the engagement rules to contribute are quite strict and require a peer review level of competence and quality for both contributors and products. Most notable examples of this category are Fedora, EPEL, Debian, CentOS, etc.

Services and tools: these open source communities usually provide a software application and often services based on that application. They have dual usage models, whereby access to the service is free for personal or non-commercial use, while professional use is charged a fee. Most notable examples include SourceForge, GitHub, Zarafa and many others.

End-to-end (or vertical) open source communities: at the end of 2011 Andrew Aitken, president of the Olliance Group, a leading open source consulting firms, wrote an article about the appearance of ‚super-communities‘ or communities of communities [10]. The super-communities instead of focusing on a particular piece of open source technology are built around the entire end-to-end supply chain of an industrial sector, like the aerospace industry (Polarsys launched by Airbus), stock exchange management (OpenMama launched by the New York Stock Exchange) or OSEHRA (electronic healthcare records launched by the US Department of Veteran Affairs). The Olliance Group predits that this kind of communities will rapidly increase in number. Microsoft launched in 2011 the OuterCurve community to host open source software and communities and provide general-purpose IP management services with a focus on Windows applications. In the scientific communities, portal like NanoHub [11] or CyberSKA [12] focus on the general needs of the nano technologies and radio astronomy communities respectively.

The software produced and used in scientific applications is by its nature very diverse. It uses different programming models, technologies, IP and licensing models. In addition the end users, the scientists using the software to perform their research are not overly interest in what technologies are used under the hood, but are very concerned with having a working set of tools. The first three types of open source communities described about could therefore be used for parts of the software produced in the academic world, but would to bring the level of communication and organization needed to provide the functionality described earlier. A suitable model for ScienceSoft could therefore be the fourth one, where the overall end-to-end software needs of specific scientific communities could be modelled and addressed with a more global approach than just individual pieces of software.

The second part of the related work survey is related to the wide variety of social network-based approaches for science and society. This survey is not exhaustive, because of the paper restriction, but still aims to give a reasonable picture that similar activities exist, but are often not as focussed as the ScienceSoft activity described in this contribution.

Ohloh [2] is an activity to discover, track and compare open source available, but currently lack the means of attracting developers and users from academic communities and harvesting a large enough body of essential data to make their results meaningful for the scientific research environments.

Academia [13] aims to share related and has currently a community of roughly 1.6 million researchers. It is optimized to search scientists, research interests, and universities. ResearchGate [14] also has a user basis of 1.7 million and is optimized and offers functionality to expose the reputation of users including a breakdown of scientific disciplines. Many similar Web activities exist that in one form or another share functions as described in this paper.

VII. Conclusions

This paper introduced the ScienceSoft concept study as one of the crucial parts of the EMI future strategies. We can con clude that ScienceSoft is an initiative to assist scientific communities in finding the software they need, to promote the development and use of open source software for scientific research and provide a one-stop-shop to match user needs and software products and services.

The second conclusion we are able to derive clarifies for who ScienceSoft is intended to be. It is for developers to share their software for science; for researchers to find software, get support, express recommendations; for companies to offer services; for funding bodies and sponsors to assess the impact of projects and the value of their investments being potentially supporting when evaluating scientific grants around software engineering activities.

Thirdly, we can conclude the reason why using it at all. ScienceSoft aims to allow for promoting projects, find the right software and services for your scientific community and connect to people developing or using them. It allows to take decisions based on information shared and verified by a large community of researchers.

Finally, we conclude that this activity can be only successful if it is possible to establish a reasonable momentum in the scientific community by broadening significantly its players, actors, and stakeholders. Although many other activities have been started in the past and stopped, the time might be right now building on top of the Web 2.0 era that led to a new 'mindset' and 'skillset' of people requiring a new 'toolset'.

Acknowledgment

We thank our collaborators of the projects that joined the Science Soft activities as well as the members of the Science Soft steering committee for their fruitful advice and comments. This work has been partially funded by the European Commission as part of the EMI (Grant Agreement INFSORI-261611) project.

References

[1] EMI Project, "On the Higgs boson's track. Grid computing and EMI empower a step forward in fundamental knowledge" Press Release, July 2012, Online: http://www.eu-emi.eu/press-releases

[2] OHLOH Project, "Discover, Track and Compare Open Source", July 2012, Online: http://www.ohloh.net

[3] EMI Project, "European Middleware Initiative", July 2012, Online: http://www.eu-emi.eu

[4] ScienceSoft Community Portal, "ScienceSoft – Open Software for Open Science", July 2012, Online: http://sciencesoft.web.cern.ch/

[5] dCore Systems, "dCore Systems – International Niche Technology Company", July 2012, http://www.dcore-sa.com

[6] EPEL, "Fedora EPEL Project", July 2012, Online: http://fedoraproject.org/wiki/EPEL

[7] OSI, "Open Source Initiative – Licenses", July 2012, Online: http://opensource.org/licenses/

[8] DOI, "The DOI System", July 2012, Online: http://www.doi.org/

[9] Drupal, "Drupal Modules – Download and Extend", July 2012, Online: http://drupal.org/project/Modules

[10] Andrew Aitken,"The Advent of Super Communities" , Dec 2011, Online: http://opensourcedelivers.com/2011/12/20/the-advent-of-super-communities/

[11] NanoHub, "NanoHub – Online Simulation and More for Nanotechnology", July 2012, Online: http://nanohub.org/

[12] CyberSKA, "CyberSKA – Cyberinfrastructure platform ", July 2012, Online: http://www.cyberska.org/

[13] Academia, "Academia.edu – Share Research", July 2012, Online: http://academia.edu/

[14] ResearchGate, "ResearchGate – Your reputation, your terms", July 2012, Online: http://www.researchgate.net/