The technology evolution in web design, development and deployment supporting web scientific applications

S. Pastore

Abstract— In recent years Internet and it famous application - the web - have seen a wide diffusion and have been subjected to technological changes. The underlying platforms supporting web applications are now based on layered software and complex hardware solutions in order to support a growing load request. Sciences in general and Astrophysics in specific take advantage of Internet and the web since most of its applications are executed on the Internet and manifest a web user interface. For modern applications is important both the architectural framework where to execute them and the languages used in the design and development of web applications that could be scientific applications or web tools used to communicate this science. Among the different solution available in these two topics, this paper proposes a framework in an open environment as a combination of several software. The aim is to adhere to standards in every stage of the application lifecycle to guarantee main features as openness, accessibility, interoperability and internationalization. The solution implements the cloud paradigm through an open platform like the Eucalyptus software for the deployment environment and web specifications (i.e., HTML 5, CC3 and ECMAScript) together with different standards proposed by international bodies as languages to design and develop user interface accessing applications.

Keywords— cloud computing; grid computing; web applications; web standards technologies; markup languages.

I. Introduction

THE mission of our Institute (INAF aka Italian National Institute of Astrophysics) is to carry out, promote and to make the most of scientific and technological research in astronomy and astrophysics and to spread and popularize the results in the optics of promoting and encouraging a transfer towards the business world. The ambitious scope of the research is to answer main questions about the universe and its components through the several projects ranging from terrestrial to spatial ones. The promotion and the outreach of astronomical knowledge and discoveries could contribute to encourage the scientific and technical grow of a country. From a more technical point of view, Internet is the natural medium hosting the different applications devoted both to computation and communication goals. In most cases, such applications manifest a web user interface since the web browser is becoming the main tool through which a person can execute every category application. Web technologies, standards and platforms are thus applied also in this scientific field and the choice of such parameters become essential in design, development and deployment of whatever web applications. This paper discusses our approach in searching for suitable platforms and frameworks when developing and deploying applications trying to follow technological changes. Working in a research institute, our focus is on an open platform that adheres to standards since only common standards could guarantee interoperability solution and a real culture diffusion. This preliminary study meant to evaluate the state of the art standards and implementations in two areas related to web applications: its design and development that is used methods and standards in application programming and its deployment that is specific platforms and architectural frameworks needed for application execution. Our aim is to create a sort of "best practice" in selecting platforms, architectures and web standards, both for specific web-based astronomical applications (i.e., those that interact with database) and for generic applications that contribute to web communication and science dissemination goals. The area is that of an open environment, and thus implementations should as much as possible create an open community and avoid lock-in effects related to commercial solutions. In this paper we want to show as an open solution could be found even if with some uncertainties and obstacles in its implementation due to the rapid changes of technologies in these fields. The paper initially describes the influence of Internet and the web in the Astrophysics sciences by analyzing the two aspects of platforms and languages for web user interfaces. It approaches the evolution of the distributed paradigm by examining the experience in grid computing and the promises of cloud computing. Then it discusses about the evolution in application design and development that goes through the evolution of markup and web languages together with the dynamic features of the web. Then it describes the proposed framework that makes use of

Manuscript received March 237 June 6, 2011: Revised version received March ??, 2011. (Write the dates on which you submitted your paper for review as well as the revised version).

S. Pastore is with the Italian National Institute of Astrophysics, Astronomical Observatory of Padova, vicolo Osservatorio 5, 35122, Padova, ITALY (corresponding author to provide e-mail: serena.pastore@ oapd.inaf.it).

open source solution in the platform implementation and follows web standards in languages used for applications development.

II. Astrophysics, Internet and the Web

The Internet and the web are two fundamental aspects of actual science, both from the research point of view and in the communication and dissemination areas. Most astronomical applications need great storage and computation capacities and such applications make use of distributed platforms that are mostly based in an Internet network in order to exploit some paradigms. The grid paradigm [1] is an example: the research and educational environment experimented on grid computing [2] through the establishment of a European grid infrastructure within the EGEE project [3] reporting some successes and some failures [4]. On the other side the web is the ideal platform to disseminate information about astrophysics, its projects and its discovery; it is also the ideal software for e-collaboration, website management and social network. Examples are the use of specific collaboration or groupware software like the adoption of the Google Apps [5] (i.e. Google Docs or Google Calendar) for scientific research project [6] or the implementation of a Web Content Management System to deploy the Institute's website [7].

We shall focus on two aspects like distributed architectures and web technology needed in web application development and deployment, considering that the web is going for another evolution and the battle between open source and proprietary software is escalating. Scientific applications in general and astronomical one in specific, take advantage of complex hardware and software infrastructure needed to a growing request of storage and computation facilities. Moreover most of them should manifest a web user interface thanks to their execution through a browser, the software component that is currently present in all the users' devices. The diffusion of the so called web operating system or Web OS [8] manifests the trend towards desktop environment entirely usable from a web browser and just with an Internet capable devices allowing from access to all own applications. The design and development of web applications go through two peculiar aspects: the adoption of software architecture supporting load request and web technologies used to design and develop such applications. We approach the available solutions first examining the distributed computing platforms [9] and then the web languages that could be adopted to develop applications in order to guarantee their execution in every device and on behalf of every potential user. The goal is to propose a possible solution implemented in an open environment outlining the obstacles that could be found. Astrophysics experimented grid computing [4] as distributed platform where to execute application, but cloud computing [9] could be an interesting solution especially for it peculiar feature to provide services on demand. An important aspect for example is that such a paradigm has been chosen as a platform for the astronomical data processing in the Gaia mission [10] of the European Space Agency (ESA) [11]. This mission will conduct a survey of our solar system to discover new celestial objects and the collected data will be massive. The ESA decided to prototype a cloud-based system to analyze the data and thus outline as this paradigm could be useful in such a science.

On the other hand scientific applications present a web user interface both for scientific and communication goals and such applications should be designed and developed following standard web languages for interoperability, accessibility and stability needs. This implies the use of specific markup languages for separating presentation, structure and behavior of the content. Web standards and specifications are in large part dictated by the Word Wide Web Consortium (W3C) [12] whose goal is to develop open specification (de facto standards) to enhance interoperability of web-related products. However there are other organizations, groups and standard bodies like the European Computer Manufacturers Association (ECMA) [13] or the Web Standards Projects (WaSP) [14] that fight for adoption of standards. In these fields two specific aspects should be considered: the commercial trend of cloud computing and the diffidence in standards adoption that led to proprietary solution of which the Adobe Flash technology [15] is an important example.

III. Evolution of Distributed Computing Platforms

Distributed computing platforms are the natural choice in providing Internet services to guarantee service quality (i.e., redundancy, load balancing, geographical distribution, and etcetera), even if these require complex management of hardware and software infrastructure. Many distributed paradigms from cluster computing to grid computing and cloud computing have succeeded in order to provide the growing request of platform able to support today applications' requirements. The trend is however toward a service-oriented vision [16] thanks to web services technologies that enables exporting application trough open interfaces (e.g., Application Programming Interfaces or APIs).

The service concept allows an abstraction of software al code and gives integration capabilities.

In the last years, both grid computing and cloud computing have emerged as the right platforms to solve some of the problems that current applications have. We analyze the adoption of such infrastructures related to our specific scientific environment.

A. Grid computing experience

Our experience on grid computing started in 2003 and has gone through several grid projects from DataGrid (2002-2004) to EGEE-I, II and III (2004-2010) that contribute to the establishment of a grid infrastructure usable by different European entities. Now the infrastructure is supported by the European Grid Infrastructure (EGI) [17] through the EGI.eu Foundation. The foundation aimed at collecting several national grid initiatives (i.e., NGIs) like the Italian Grid Infrastructure (IGI [18]) and guaranteeing the long-term availability of a generic e-infrastructure. The grid experience in the astronomical environment reported some successes, but also some failures. This kind of architecture could be successfully use for the execution of some kinds of applications (i.e., batch applications requiring a large amount of computing and storage capabilities); but in the web application area, we have experimented some difficulty in platform's adoption. Applications that required web services in order to interact with distributed databases manifested some issues that required an improvement in the entire infrastructure. This was partially solved, for example, through the introduction of a new element that is able to model database resource [19]. Probably, this latter problem was due principally to the software features of the middleware adopted (i.e., the gLite [20] toolkit). This middleware was developed around job scheduler mechanisms, and was initially poorly customized for web applications since it lacks specific services as the discovery mechanism able to search and find the best grid resources for a web application. Another aspect of grid security is the need for restrictive requests in terms of authorization policies that limit the usability of shared resources whose control remains on the owner of the site. Moreover resources are related essentially to storage and computing capabilities with a lot of limitations, even if the resource concept in the model comprises other resources like software, applications and so on. Actually, there is an effort, within the European Middleware Initiative (EMI) [21] in order to create an upgraded middleware known as Unified Middleware Distribution (UMD) as the fusion of the used grid middleware gLite, the ARC [22] and the UNICORE [23].

B. The promise of cloud computing

Starting from our experience in grid computing and deployment of web applications, we begin by analyzing cloud computing that seems to solve some of the issues introduced by the adoption of the grid. Thanks to virtualization technology [24], the resource is effectively extended to include an entire virtual machines. The core concept of cloud computing starts with services provided via the web, but the provision is on demand that is the resource is given when the user makes the request and then released. This aspect is extremely important in applications that require massive capabilities for a limited period (i.e. in processing data for example), but also in network services like the web server provision where traffic load is asymmetric (i.e. increasing of web access in occasion of some special astronomical events). As the NIST [25] definition states "cloud is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable resources that can be rapidly provisioned or released with minimal management effort or service provider interaction". The cloud paradigm manifests three delivery service models, as the Fig. 1 shows and different types of community created according the type of network they establish (i.e. private, public, hybrid and community).

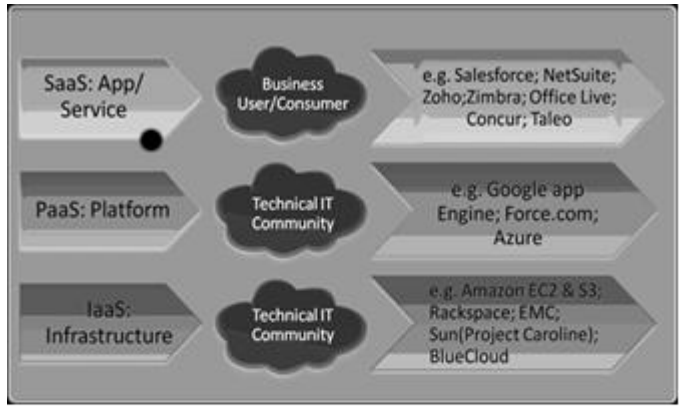

Fig.1: Different types of cloud service models

The service models range from SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service). There are thus different deployment solutions related to the type of offered service that is also outlined by the presence of various cloud software implementations, both commercial and open source. Actually each IT and software enterprises (e.g., Oracle [26], HP [27], and etcetera) propose offer their specific cloud models and platforms. Among the offering we could cite those that offers a SaaS cloud by providing software (i.e., Google App engine solutions [5]), those that gives on demand frameworks to develop and deploy applications in various programming languages (i.e., Microsoft Azure [28]) and those that implement a IaaS solution providing a complete machine (i.e., the Amazon Elastic Compute Cloud – EC2 [29]). Amazon in specific provides a great amount of web services [30] integrated with a cloud such as the Amazon Simple Storage Service (S3) and it could be collocate as the primary cloud provider. The IaaS implementation is a complete solution that solved ownership issues related to grid environment, since a virtual machine (or instance in Amazon-like term) is provided with all privileges giving the possibility to install the software stack needed for a specific execution. Summarizing the cloud could be seen as a mixture of Service Level Agreement (SLA), web services, and virtualization technologies. In this description seems to be en evolution of already used technologies that follow a different business model. Cloud computing's best implementation rests on it capability to answer user demands for computer resources.

Within this plethora of commercial solution, the open source panorama is very limited. Among academic and research offerings (i.e. the Nimbus cloud [31] a toolkit that allows to turn a cluster in a IaaS cloud), a competitive solution seems to be Eucalyptus [32]. This solution, present in a commercial and open source version, provides a software available under GPL that helps in creating and managing a private or publicly accessible cloud maintaining a compatibility with Amazon EC2/S3 services. Its popularity is due since it is one of the few implementation of an open cloud and is adopted into the Ubuntu project as Ubuntu Enterprise Cloud (or UEC) [33] included in the server edition of this operating system that is one of the most widespread Linux distributions. The architectural components of the software run on top of existing resources (i.e., Linux distribution and hypervisor agnostic) and its adoption could satisfy the goal of an open implementation of a IaaS cloud. Users' of Eucalyptus are also part of a community, called the Eucalyptus Community Cloud (ECC) [34]. This is an environment where they can test drive and experiment this software.

C. Grid vs. Cloud

Cluster, grid and cloud computing apparently solve the same request, but while grid is devoted more to hardware-like resources, cloud extends this concept to software-like resources. Moreover, the type of access seems to be different: grids are born as an execution environment and thus are tied to parallel software and toolkits able to distribute execution. Most commands are as yet command-line and the kind of software implementation (especially with the European grid infrastructure) has precluded the adoption of such technology in many organizations. Most implementations in, for example, Italy remain project efforts [35]. In addition, the security model, following Public Key Infrastructure (PKI) and an authorization model based on attributes, leaves the control of the resource provided to the owner of the resources. Cloud computing promises to avoid this limitation and gives total control to the consumer, although this entails security trade-offs. Table 1 summarizes the difference between such technologies in terms of some features such as resources type, hardware and network characteristics. Cloud computing comprises a connection and interconnection of utilities, while cluster is an access to real physical hardware, operating systems, and applications present at one physical location.

Table 1: Differences on distributed paradigm

Category	*Cluster*	*Grid*	*Cloud*
Size	Small	Great	Small→great
Initial cost	Very high	High	Low
Resources type	Homogenous	Heterogeneous	heterogeneous
Typical ROI	Very High	Medium	High
Hardware	Very expensive	Expensive	VM use upon hardware
Network type	Private /Proprietary	Private based on Ethernet	Public Internet on Ethernet
Security requirement	Low→ High	High	High
SUMMARY	Supercomputer	workstation	Groups of VM

A grid system offers distributed access to real physical hardware, operating systems and applications present across the world. Cloud guests access virtualized resources distributed on one system on demand. As the last entry of Table 1 indicates, while cluster systems are, in practice, supercomputers, grid systems become fast workstations, while cloud systems are groups of virtual machines. A comparative study between grids and clouds has been done within the EGEE project [36]; they study looked at EGEE grid implementations and the Amazon Web Service (AWS) for cloud. This is an important study, the main results of which are reflected in the Fig. 2. The main differences between these two platforms based on different environments where these two distributed platforms are used, is as follows: grids are typically used for job execution, while clouds are used to support long-terms services. However, the different distributed paradigms do not seem to be in competition; they seem to be complementary platforms customized according to the type of application to be deployed on them. Probably, the best solution could be their integration; such, however, could be possible only if open source software implementation were made available to encourage standardization.

	EGEE Grid	Amazon Cloud
Target Group	Scientific community	Business
Service	short-lived batch-style processing (job execution)	long-lived services based on hardware virtualization
SLA	Local (between the EGEE project and the resource providers)	Global (between Amazon and users)
User Interface	High-level interfaces	HTTP(S), REST, SOAP, Java API, BitTorrent
Resource-side middleware	Open Source (Apache 2.0)	Proprietary
Ease of Use	Heavy	Light
Ease of Deployment	Heavy	Unknown
Resource Management	probably similar	
Funding Model	Publicly funded	Commercial

Summary of „An EGEE Comparative Study: Grids and Clouds Evolution or Revolution" by Markus Klems

Fig. 2: A comparison between features of a grid computing (i.e., EGEE grid) and a cloud computing implementation (i.e. Amazon cloud)

Unfortunately, this is not yet the case even if one of the IGI objectives that is under study is the possible effective integration of the two platforms.

IV. APPLICATION DESIGN AND DEVELOPMENT IN THE WEB 2.0 ERA

The Web is an ecosystem of different languages i.e. standard markup and programming languages and proprietary technologies. Even if the core language remains HTML in least years it has been subjected to an evolution. Nearby this language to structure information, there are the Cascade Style Sheet (CSS) [37] for presentation aspect and scripting languages i.e. ECMAScript [38] that contribute to introduce interaction and dynamic effects in web application. These concepts allow following the three layers architectural model of web programming that creates a separation in web design between structure, presentation and behavior. Most current applications are developed as web applications thanks to the advanced features of web browser that are becoming the ideal execution environment with their simplified development curve and management. The web technologies field shows different trends. Applications are becoming more rich and interactive, but the standards supporting such features seem to create obstacles in applications' development, as the case on the future of HTML languages has shown [39]. Web applications are comprised of user interface code and back-end data. The user interface code runs in markup and client-side languages, while the back-end code, which can be proprietary, can run in whichever language. We focus on the front-end part of web applications following the way to present web content according to accessibility, usability and internationalization both for people and machines.

A. *Evolution of software models*

Software development followed a series of changes over time, as Fig. 3 shows.

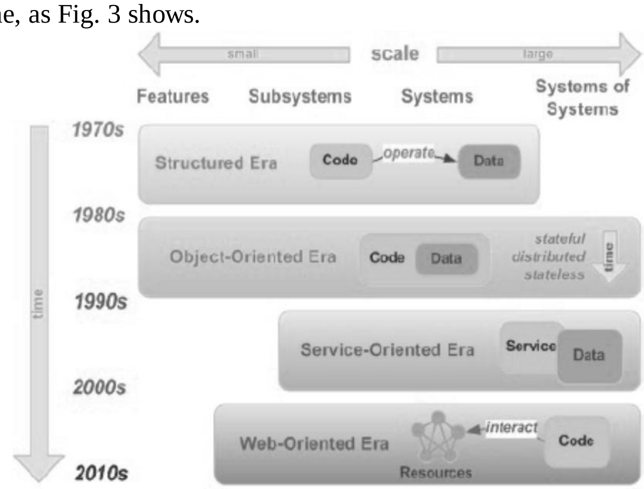

Fig. 3: Evolution of software models

These changes, involved paradigm models and approaches in writing code. In the last years, we have seen a shift towards the web as the ideal platform to execute applications: web languages are gained a great popularity through standard development efforts in order to establish such a platform.

We could think of such a deployment and execution environment as a web-oriented platform (Fig. 4) that is comprised of a core element: that is, the data offered as a web resource in a wider meaning. Through the web, such data is distributed in different ways that go beyond its simple delivery as hypertext.

Developing an application today is a matter of deploying services in the Internet. Although some software remain locally installed in users' machines, frequently used applications are executed through a web browser and thus are developed with the use of web technologies. Clients are becoming more and more powerful and able to execute complex web applications that enhance the user's experience.

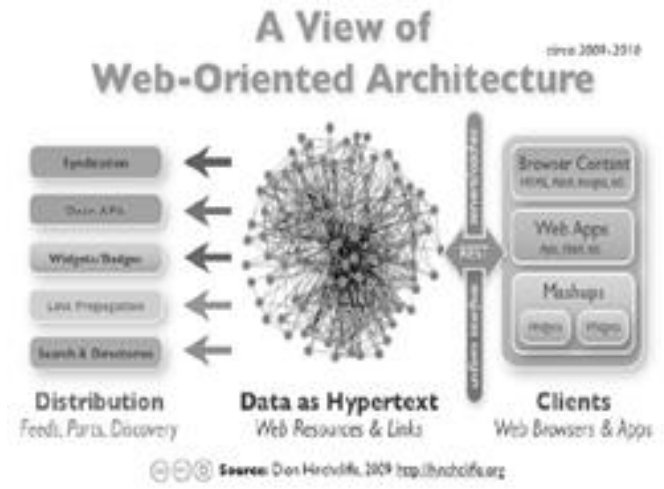

Fig. 4: Components of a web-oriented architecture

However all technologies used in design and development should follows web standards that is the languages and specifications used for creating and interpreting web-based content: their application allows for design and publish content to be delivered by the greatest number of web users in long-term availability.

B. *Evolution of markup languages*

HTML is the core language for the web that from its born has been subject to changes (Fig. 5) even due to the publication of new type of content, especially in the multimedia field.

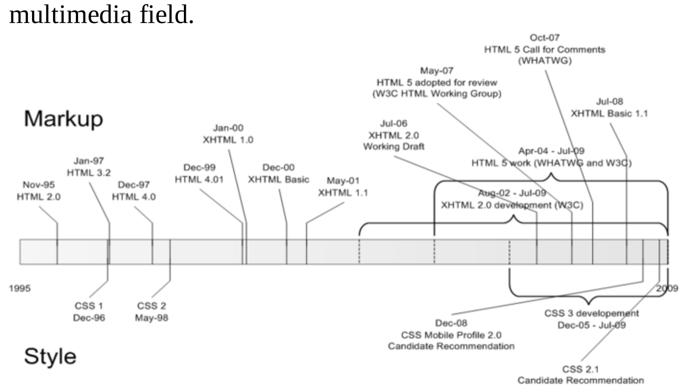

Fig. 5: Current frequently used versions of the hypertext markup languages

HTML is designed as a structure language, but for many years was used also with presentation goals. The clearly separation between the three layers (structure, presentation and behavior) in web design has encouraged the adoption of presentation and behavior languages, but the need of structuring content different from simple text has contributed to a new formulation of HTML [40]. The W3C shifted to a formulation of HTML according the XML language (forming XHTML [40]) since XML represents a family of languages able to structure every kind of data in a simple and interoperable way being in simple text format and thus easily converted in every other format.

While W3C initially has given the impression to abandon HTML in flavor of its redesign in XML-like language. XHTML version 2.0, unlike previous specifications, is based solely on XML and it does not provide compatible with existing content or previous markup language versions. This was a great problem for web developers that continue to use previous versions of the language that are more simple to use. Thus to overcome the issues related to HTML lack of specific element, other technologies, most of them proprietary, have taken popularity in the web world. This is the case of plug-in technologies (e.g., Adobe with Flash platform technologies [15] or Microsoft with Silverlight [41]) especially used for multimedia content. Also for this reason, a group called the Web Hypertext Application Technology Working Group (WHATWG) [42], initially founded by individuals from most commercial enterprises (i.e., Apple [43], Mozilla Foundation [44] and Opera Software [45]), works actively at HTML development in the optics of web applications, and thus of APIs and rich interfaces. The result is HTML 5 [39] that sounds like new version of the previous markup languages specifications (i.e., HTML 4.0, XHTML 1), plus the enhancement of the document object model (i.e., DOM [46]); nevertheless, it is more a transitional technology than a revolutionary one and is meant to address deficiencies in HTML4. After this scission and the fact that XHTML 2 finds few implementations in the web world, W3C decided to review the adoption of the HTML5 specification and invested in its development. The WHATWG and the W3C are now working together to create this next generation language. However, at the present, there two versions of HTML5 that manifest minor differences, even if their overall goals are equivalent.

A. *HTML5 promises*

HTML 5 tries to collect all the functionalities needed by actually interfaces by introducing some important element and attributes (Fig. 6).

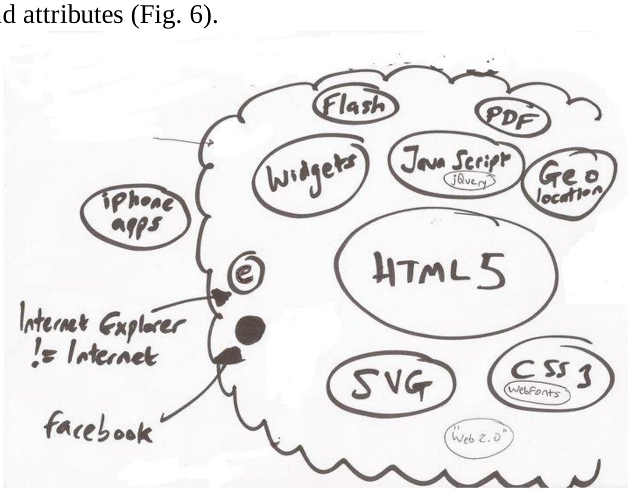

Fig. 6: HTML 5 ecosystem

It refers to a collection of individual features that define new tags and how such tags interact with objects and languages. Its force is that it would be based only on standards (i.e., HTML versions, CSS for presentation and ECMAScript applied to the DOM taking advantage of the objects elements, properties and methods to provide the dynamic effect. Among new elements the video element allows to embed such a content directly on the web page avoiding the need of plug-in technologies.

Another is the canvas that will able to include any type of graphics in the page. Finally the geolocation allows for delivering customized content basing on user's location. Actually the problem is the existence of two versions of the specification and the fact that not all the browsers now implement such features.

Moreover even if web browsers support the language as in the past, they support in several ways (i.e. with different element) due also to the existence of different implementations of ECMAscript (Fig. 6) and its features (events and objects). However the potentiality to improve web applications and its main elements (i.e. web forms, video, etc.) together with the facility of learning since so tied to the old version of the language will guarantee its success. Even if the standardization will take a long period, its adoption at least in some of the main features could be done by its rendering in the last version of the used web browsers.

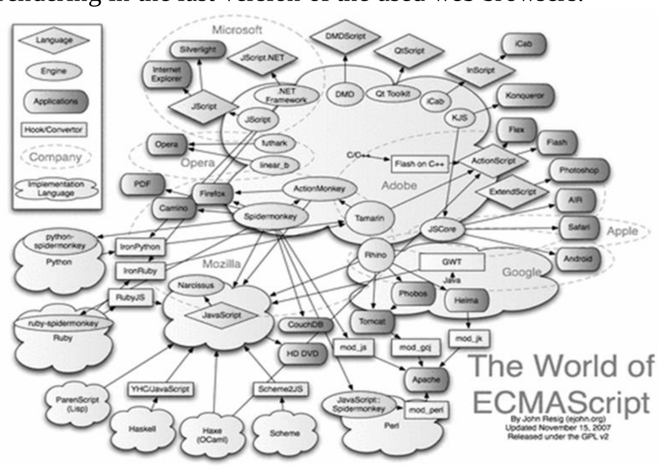

Fig. 7: The World of ECMAScript

The debate is between the adoption and how HTML5 pages are rendered on each browser since in some case there are some supporting issues. The large number of browsers and version is an obstacle in the adoption of such a language requiring a great effort in the development in order to support all of them.

B. *About content internationalization and multilingual web*

Web standards and interoperability play an important role in this context not only as a set of technologies able to deliver content but also to make it usable and readable by potentially all people in the world. In this optics when developing web content for applications it is necessary to consider the multilingual features vital to reach as much as people as possible and thus increment international collaborations. People that, in our context, need to be reached are heterogeneous: they range from the public that is in some way interested in such subjects to science users at different levels and business organizations that are or potentially could be involved in astronomical projects with a return in term of investment and market. Taking into account internationalization is vital to could reach all these users at every level and to internationalize web content deployed on sites. This aspect is part of the decision about standards and languages to choose in web application development and content authoring. The W3C Internationalization (I18n) Activity [47] works to ensure that content will be available apart from different languages, cultures and writing systems. It provides best practices and standards to develop web content with i18n features. In our context, several obstacles are present in implementing and adopting standard solutions in general and in specific whose standards and practices regarding internationalization since the lack of knowledge and culture in web standard. Main challenge is to educate heterogeneous web content authors and to persuade them that such effort cloud be useful fiving an added value to their content that repay from the efforts done. The context and some issues related to reach at least internationalization are the following:

- a lot of web authors spread in the several sites composing use different authoring tools (from specific audio/video software makers like Flash to a vast numbers of CMSs like Wordpress, Joomla, Plone, etc.) and some that use yet a simple text editor to compose web pages directly in HTML/XHTML/CSS languages. But these tools require a customization or a special attention in order for example to declare and apply an encoding.
- Web content is hosted in several web servers managed by different users with little experience since the facility to setup and run in a default way a web server. Thus it is difficult that such managers remember to set character encoding, or HTTP charset parameter.
- A lot of web authors are not specifically skillful in web technologies and sometimes are reluctant in following web standards or practices that require special attention in making content.

The great availability of authoring tools that makes web publishing very easy for everyone and thus without paying attention in what is the final web product, bars very often with developing content actually accessible, usable and international.

V. Our suggested solution

The solution proposed should be as much as possible an open-source model both in software implementation and in the distribution and dissemination of information. The creation of a research community aimed at redistributing technical knowledge is also a vision of our study. Our case studies regard applications that are related on two main fields: those based on web user interface and interaction with databases that allow users to extract specific data and provide multimedia files, and those that enable communication and outreach in astrophysics and require rich interfaces.

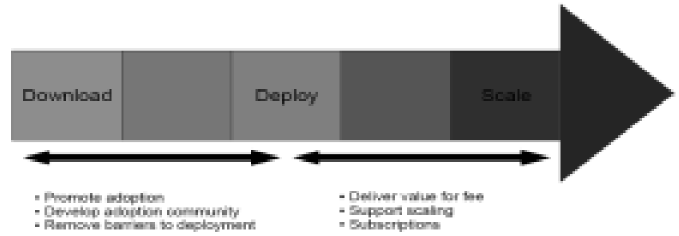

Fig.8: The open-source business model

In these two areas, the interaction with users means the creation of dynamic applications that in many cases could take advantage of resources, but only for specific and limited period (i.e., related for example to astronomical events).

Our study is focused on providing the right platform able to satisfy all these constraints; it could become the ideal platform to develop and deploy these kinds of applications. Our attention is both on distributed platforms and on methods to develop applications that could be adopted in an open environment, such as that of research institute. Even if with some uncertainties and obstacles we propose the adoption of the Eucalyptus system as implementation of a cloud platform and web technologies based on HTML 5 [48], CSS 3[49] and ECMAScript to design and develop applications (Fig. 9).

The grid platform was initially born and developed as an open environment, even if it is subjected to security constraints related to owner resources; the cloud, on the other hand, is born primarily as a commercial environment with the aim of pay and use. However, there are several incentives for the creation of an open cloud community, in some sense parallel but interoperable with the commercial one.

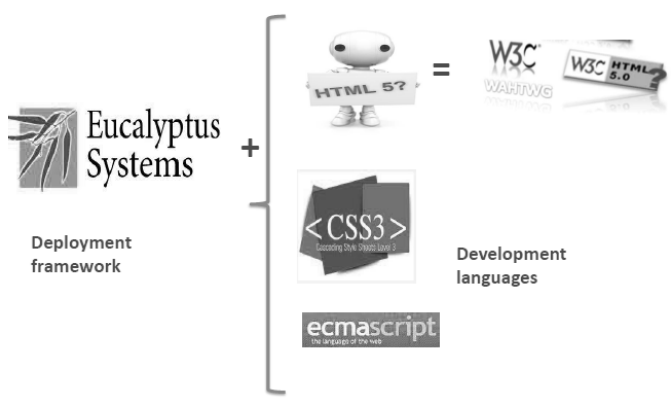

Fig.9: Solution proposed

The creation of an open cloud could be a starting point to verify the efficacy of such a platform. Example would be, the establishment of the Open Cloud Consortium [50] that supports the development of standards for cloud computing and frameworks for the interoperation between clouds, and the Open Cloud Manifesto [51] initiative dedicated to the belief that the cloud should be open.

Eucalyptus software provides a feasibility solution even if with interaction with commercial solutions. We hope looking at the effort of all different community (i.e. the grid ones) that will be a convergence between the platform adoption. From the web applications field, the choice to support web standard is dictated by accessibility, usability, interoperability and openness goals that a scientific and research environment should obtain. Even if specific applications take advantage of being stand-alone to maintain a great performance, most applications could be enhanced if executed through a browser, both for the possibility of execution in most devices and the lack of installation, and thus the need for specific interdependency. In writing any kind of flexible code and specifically for web applications, the two key concepts are portability (and thus the ability to run components or systems written for one environment into another) and interoperability (and thus the ability to write a code that works with multiple providers regardless of the differences between them). HTML 5 promises to help in writing applications that work across-platforms and thus our study suggests to work in this direction and in using HTML5, CSS 3 [41] and Javascript for programming applications especially as sophisticated user interfaces. These three languages should allow to write the code for every web application that includes every type of content even multimedia one. This is in spite of the continued browsers war with major players that develop some technology of their own that results in the fact HTML 5 has little support in modern browsers, just like all the various standards in the past. HTML5 could be the unifying language since the market is investing in this language even if the diffusion of proprietary technologies will be very hard to overcome.

VI. Conclusion

Cloud computing and HTML5 are two buzzwords in the IT environment that in some way demonstrate that we are now really going into an always connected environment. Network services and distributed applications could take advantage of on-demand resource requests without reconfiguring the hardware and software infrastructure, while application could include multimedia and interaction capabilities without plug-in technologies, and through a technology that cloud really be cross platform. For these main reasons, we try to improve the design, development and deployment of application that could satisfy accessibility, interoperability, internationalization and openness goal. We propose a framework make of software as results of open solutions and based on standard implementations of web specifications. We think of web applications in general both as user interface to astronomical applications and as tools and websites used to disseminate knowledge about astrophysics and its projects. Eucalyptus software is as an open implementation of the IaaS cloud infrastructure that is able to offer a web application deployment environment on demand but in the form of a virtual machine resource and thus with the full flexibility. However technical issues are outlined in cloud implementation concerning the great demand of network bandwidth even if tied to the type of application and of the method in which it is distributed. In addition, security problems in resource accessing and the sharing with the other open public clouds is an issue here. Other problems concern the lack of documentations and the need of specific skills in installing and managing such an infrastructure. Nevertheless, testing and adopting such a model with a network service as the web hosting could demonstrate the validity of such a model in our field. On the other hand, we propose to follow HTML 5 specifications on making the application developer follow web standards, even if that could require complex solutions and need more attention in considering specific turnaround to support the different browsers. This implies a great effort and testing capabilities in developing web applications. However, we think that an open community means more than transparency, and the problem with any new technology is how to avoid vendor lock-in. Moreover uncertainties and obstacles remain in the full adoption of such solution even if to the rapid changes of technologies.

References

[1] H. B. Higgings, “The grid book”, The MIT Press, 2009.

[2] S. Pastore, “Internet technologies and the grid paradigm: design a custom environment for web service-based applications),” in *Proc.6th WSEAS Int. Conf. on Simulation, Modelling and Optimization*, Lisbon, 2006, pp. 639–698.

[3] The EGEE (Enabling Grids for E-science) project. Available: http://www.eu-egee.org. 07.03.2011

[4] S. Pastore, Description and Discovery of Web Applications in Grid, Proceedings of The European Computing Conference, Vol. 2, Lecture Notes in Electrical Engineering, Vol. 28, Springer ,2009, pp. 375-382.

[5] Google Apps. Available: http://www.google.com/apps

[6] S. Pastore, “Social networks, collaboration and groupware software for the scientific research process in the web 2.0 world”, *in Proc. 7th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES (AIKED'08*), University of Cambridge, UK, 2008, pp. 403-408

[7] S. Pastore, C. Boccato, A Web Multi-Tiered Framework for the Italian National Institute for Astrophysics: Expanding Functionality by Incorporating Collaborative Tools to Promote the Institute, *WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS*, Issue 11, Volume 3, November 2006, pp.2207-2214.

[8] J. Strickland, How Web Operating System Works. HowStuffWorks.com. 14 May 2008. Available [Online]: http://computer.howstuffworks.com/web-operating-system.htm

[9] ITU-T Technology Watch Report 9, Distributed Computing: Utilities, Grids and Clouds, ITU, 2009.

[10] The GAIA mission. Available: http://www.esa.int/science/gaia

[11] The European Space Agency (ESA). Available:http://www.esa.org

[12] W3C, the World Wide Web Consortium, http://www.w3c.org

[13] ECMA, www.ecma-international.org

[14] The Web Standards Project. Available: http://www.webstandards.org

[15] Adobe Flash Platform technologies, http://labs.adome.com/technologies/flash

[16] T. Erl, SOA Principles of Service Design, Prentice Hall, 2008

[17] The European Grid Infrastructure (EGI). Available: http://www.egi.eu

[18] The Italian grid Infrastructure, Available: http://www.italiangrid.org
[19] E. Ambrosi, A. Ghiselli, G. Taffoni: GDSE: A New Data Source Oriented Computing Element for Grid. Parallel and Distributed Computing and Networks 2006, pp. 53-57
[20] The Lightweight Middleware for Grid computing, http://glite.cern.ch
[21] The European Middleware Initiative (EMI), http://www.eu-emi.eu
[22] The Advanced Resource Connector (ARC), http://www.nordugrid.org/arc
[23] Uniform Interface to Computing Resources (UNICORE), http://www.unicore.eu
[24] G. Shields, The Shortcut Guide to Selecting the Right Virtualization Solution, RealtimeNEXUS, 2008
[25] The National Institute of Standards and technology, http://www.nist.gov
[26] Oracle Cloud Computing, http://www.oracle.com/us/technologies/cloud/index.htm
[27] Finding the right cloud solutions for your organization, Business White Paper, HP, 2011, http://www.hp.com/go/cloud
[28] Windows Azure Platform, http://www.microsoft.com/windowsazure
[29] Amazon Elastic Computing Cloud EC2, Available: http://www.amazon.com/ec2
[30] Amazon Web Services (AWS), Available: http://aws.amazon.com
[31] The Nimbus project, http://www.nimbusproject.org
[32] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, D. Zagorodnov, "The Eucalyptus Open-source Cloud-computing System"" in Proc. of 9th IEEE International Symposium on Cluster Computing and the Grid, Shanghai, China.2009
[33] Ubuntu Enterprise Cloud (UEC). Available: http://www.ubuntu.com/business/cloud/cloudcomputing
[34] Eucalyptus Community Cloud, http://open.eucalyptus.com/CommunityCloud
[35] Grids and clouds: the new computing, GridTalk, January 2009 -4, http://www.gridtalk.org/briefings/gridsandclouds.pdf
[36] S.Pastore, Investigating frameworks to address issues involved in sharing service-based software resources within grid environments, WSEAS Transactions on Computers, Issue 1, Volume 6, January 2007, pp. 72-79.
[37] CSS – Cascade Style Sheets, http://www.w3.org/Style/CSS
[38] ECMA-262: ECMAScript Specification, www.ecma-international.org
[39] HTML, Living Standard, http://whatwg.org.html, Last updated 5 february 2011
[40] Potet, C., HTML 5 vs. XHTML 2: The Future of Web Standards, 2009, http://www.digital-web.com/articles, html4_xhtml2_and_the_future_of_the_web
[41] Microsoft Silverlight Site, http://www.silverlight.net
[42] The Web Hypertext Application technology Working Group, http://www.whatwg.org
[43] Apple website, http://www.apple.com
[44] Mozilla Foundation, http://www.mozilla.org/foundation
[45] Opera, http://www.opera.com
[46] DOM (Document Object Model), http://www.w3.org/DOMI
[47] The W3C Internationalization Activity, http://www.w3.org/International/
[48] J. Keith, "HTML 5 for Web Designers", A Book Apart editor, 20
[49] Dan Cederholm, CSS3 for Web Designers, A book Apart, 2010
[50] . Open cloud manifesto, http://www.opencloudmanifesto.org
[51] Cloud computing Use cases: a white paper produced by the Cloud computing use case discussion group, 2009.