



ELSEVIER

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

TEFIS: A single access point for conducting multifaceted experiments on heterogeneous test facilities



M. Yannuzzi^{a,*}, M.S. Siddiqui^{a,b}, A. Sällström^c, B. Pickering^d, R. Serral-Gracià^a, A. Martínez^{a,b}, W. Chen^d, S. Taylor^d, F. Benbadis^e, J. Leguay^e, E. Borrelli^f, I. Ormaetxea^g, K. Campowsky^h, G. Giammatteoⁱ, G. Aristomenopoulos^j, S. Papavassiliou^j, T. Kuczynski^k, S. Zielinski^k, J.M. Seigneur^l, C. Ballester Lafuente^l, J. Johansson^c, X. Masip-Bruin^b, M. Caria^m, J.R. Ribeiro Juniorⁿ, E. Salageanu^o, J. Latanicki^e

^aNetworking and Information Technology Lab (NetIT Lab), Technical University of Catalonia (UPC), Spain

^bAdvanced Network Architectures Lab (CRAAX), Technical University of Catalonia (UPC), Spain

^cCentre for Distance-Spanning Technology (CDT), Luleå University of Technology, Sweden

^dIT Innovation, UK

^eTHALES Group, France

^fInstitut National de Recherche en Informatique et Automatique (INRIA), France

^gSoftware Quality System, Spain

^hFraunhofer Institute FOKUS, Germany

ⁱEngineering, Ingegneria Informatica, Italy

^jInstitute of Communications and Computer Systems (ICCS), National Technical University of Athens (NTUA), Greece

^kPoznan Supercomputing and Networking Center (PSNC), Poland

^lUniversity of Geneva, Switzerland

^mTechnische Universitaet Braunschweig (TUBS), Germany

ⁿUniversity of Sao Paulo, Brazil

^oActiveEon, France

ARTICLE INFO

Article history:

Received 7 July 2012

Received in revised form 20 December 2013

Accepted 27 December 2013

Available online 4 January 2014

Keywords:

Experimental

Testbeds

Versatility

FIRE

Networks

Living labs

ABSTRACT

A few years ago, an experimental facility composed of networking gear and simulation tools was sufficient for testing the main features of a prototype before the final product could be launched to the Internet market. This paradigm has certainly changed, but the lack of platforms enabling the realistic assessment of the different facets of a product, including cross-cutting trials across different testbeds, poses strong limitations for researchers and developers. In light of this, we present an open platform that offers a versatile combination of heterogeneous experimental facilities called “Testbed for Future Internet Services” (TEFIS). TEFIS provides a single access point for conducting cutting-edge experiments on testbeds that supply different capabilities, including testbeds dedicated to network performance, software performance, grid computing, and living labs. We shall show that TEFIS covers the entire life-cycle of a multifaceted experiment, with the advantage that a single testrun can seamlessly execute across different experimental facilities.

* Corresponding author. Tel.: +34 93 896 72 93; fax: +34 93 896 77 00.

E-mail addresses: yannuzzi@ac.upc.edu (M. Yannuzzi), siddiqui@ac.upc.edu (M.S. Siddiqui), Annika.Sallstrom@cdt.ltu.se (A. Sällström), jbp@it-innovation.soton.ac.uk (B. Pickering), rserral@ac.upc.edu (R. Serral-Gracià), annym@ac.upc.edu (A. Martínez), wxc@it-innovation.soton.ac.uk (W. Chen), sjt@it-innovation.soton.ac.uk (S. Taylor), farid.benbadis@thalesgroup.com (F. Benbadis), jeremie.leguay@thalesgroup.com (J. Leguay), Elvio.Borrelli@inria.fr (E. Borrelli), iormaetxea@sqz.es (I. Ormaetxea), konrad.campowsky@fokus.fraunhofer.de (K. Campowsky), gabriele.giammatteo@eng.it (G. Giammatteo), aristome@netmode.ntua.gr (G. Aristomenopoulos), papavass@mail.ntua.gr (S. Papavassiliou), docentt@man.poznan.pl (T. Kuczynski), sergiusz@man.poznan.pl (S. Zielinski), jean-marc.seigneur@trustcomp.org (J.M. Seigneur), carlos.ballester@unige.ch (C. Ballester Lafuente), Jeaneth.Johansson@ltu.se (J. Johansson), xmasip@ac.upc.edu (X. Masip-Bruin), caria@ida.ing.tu-bs.de (M. Caria), jrobert.junior@gmail.com (J.R. Ribeiro Junior), emil.salageanu@activeeon.com (E. Salageanu), joseph.latanicki@thalesgroup.com (J. Latanicki)

In order to demonstrate the potential and versatility of the TEFIS platform, we describe the deployment of four distinct experiments and provide a set of results highlighting the benefits of using TEFIS. The experiments described in this article cover: (i) the experimentation with an open API called OPENER (which is an open and programmable environment for managing experimentation with SDN applications); (ii) an application for skiers and tourists at the Megève ski resort in France; (iii) an application that can dynamically adapt the Quality of Experience (QoE) of multimedia services for mobile users; and (iv) an augmented reality workspace for remote education and learning purposes based on videoconferencing.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The heterogeneity of applications, protocols, and devices in today's Internet has dramatically increased the complexity and cost of thoroughly testing new technologies, especially in scenarios that aim to be as close as possible to the target ones. Indeed, the success of a new Internet technology now depends both on objective and subjective aspects, so traditional simulation and experimentation strategies are being relegated to the very early stages of prototype testing. In this context, one of the main challenges faced by researchers and developers is the lack of experimental platforms supporting the assessment of the different facets of a service. For instance, consider the case of an experimenter who would like to assess not only the performance of a distributed application along with the experience obtained by the users, but also the quality and reliability of the software packages that support the application itself. Ideally, experimenters would like to run such tests in "a single place", irrespective of how many different test facilities are required, and they would like to be able to manage their experiments throughout its entire life-cycle. Moreover, experimenters would also like to be able to build upon other experimenters' work, as well as learn from their experiences. This would obviously facilitate the overall experimentation process, and would allow experimenters to contribute with their own work to the existing knowledge base.

Our aim in this paper is to present a new platform with the potential to meet these ideals. The "TEstbed for Future Internet Services" (TEFIS) [1] is such a platform and represents a large European effort enabling multifaceted experiments on different test facilities. More specifically, TEFIS offers a versatile combination of heterogeneous experimental facilities through "a single access point", and it is built as an open platform that currently provides seamless access for testing services on six different testbeds, namely: (1) *PACAGrid* [2] (a grid computing facility for experimenting with computationally intensive applications); (2) *ETICS* [3] (an e-infrastructure that provides support for testing the entire software development life-cycle); (3) *SQS-IMS* [4] (an infrastructure for testing protocols and mobile applications over the IP Multimedia Subsystem (IMS) [5]); (4) *The Botnia Living Lab* [6] (a Living Lab that offers end-user engagement in the design and testing of Future Internet technologies and services); (5) *Kyatera* [7] (a high-speed optical network for supporting experimentation of technologies with large bandwidth demands); and (6) *PlanetLab* [8] (a testbed for distributed

systems and network research on a large scale over the Internet).

By seamlessly combining these different testbeds, TEFIS can cover most of the development and testing activities, such as building and packaging software, system integration, Service-Level Agreement (SLA) dimensioning, large-scale deployment and testing, compliance testing, as well as "user-level evaluation" of run-time services. As we shall show, the TEFIS platform provides all the necessary services enabling the management of underlying resources for executing complex experiments on multiple test facilities. In particular, TEFIS handles aspects such as resource management (e.g., resource access, matching and identification of resources that can be activated, etc.), software deployment, measurement-related services for a variety of testbeds, as well as data persistence. One of the key advantages of TEFIS is that this handling is not exposed to experimenters, since the platform abstracts the internals of the different testbeds by means of a flexible interface that enables not only the robust configuration of experiments, but also the automation and repeatability of the same.

Overall, this paper makes the following contributions. We describe an open architecture that is sufficiently general as to enable multifaceted experimentation on multiple (and quite different) testbeds. We show that the combination of testbeds is made possible by means of "connectors", which provide the desired level of abstraction for experimenters while facilitating the process of adding and/or removing testbeds as needed. As a proof-of-concept, we describe four distinct experiments that were recently run on TEFIS, and show how they have benefit from the resources offered by the different test facilities. We also examine some of the future challenges faced by experimental platforms such as TEFIS, including a discussion on business and sustainability models allowing the continuity of these platforms.

We contend that the TEFIS model can be positioned as an enabler for building a community of experimenters, which can increase their expertise and share their knowledge in order to accelerate, and improve, the design and evaluation phases for a broad spectrum of Internet technologies.

The remainder of this paper is organized as follows. In Section 2, we review related work. In Section 3, we describe in more detail the experimental facilities available through TEFIS and the complementarities and synergies that they bring for experimenters. Later, in Section 4, we outline the architecture and the main components of the TEFIS platform. A step-by-step procedure for creating a

new experiment as well as the internal workflow for running a multifaceted experiment through TEFIS are described in Section 5. In Section 6, we depict a set of four different use cases, which perfectly capture the potential and the capabilities of the platform for supporting the inherent heterogeneity of multifaceted testing. Finally, Section 7 discusses the main challenges for maintaining experimental platforms like the one presented in this work, and concludes the paper.

2. Related work

During the last few years, several initiatives across the world have focused on providing support and resources for cutting-edge experimentation under the framework of the so called “Future Internet”. In this section, we review some of the most prominent endeavors in this direction, and highlight the features that are unique in TEFIS. Clearly, we cannot cover here all the initiatives in the area, so we will mainly address a subset of them with special focus on some of the activities recently started in Europe, in the US, and Japan.

In reference to Europe, the advances are mainly driven by a research program launched and managed by the European Commission called the “Future Internet Research and Experimentation” (FIRE) initiative [9]. FIRE seeks to foster the development of experimental facilities capable of supporting the testing of a broad spectrum of ICT technologies, protocols, and applications in Europe. TEFIS is in fact just one of the experimental platforms that are being developed under the FIRE umbrella. In a nutshell, the initiatives in FIRE whose objectives are closer to TEFIS can be categorized into three groups, namely: (i) those addressing *Data and Service Management* experimentation (including clouds); (ii) those addressing experimentation with *Networking technologies and protocols*; and (iii) the ones developing platforms for experimenting in the area of *Content-Centric Networking*. It is worth noting that these are indeed overlapping groups, since, for instance, some of the activities in the area of “Content-Centric Networking” also address “Data and Service Management” issues and vice versa.

In the first group, we found initiatives such as NOVI [10], BonFIRE [11], and LAWA [12]. NOVI is exploring ways to compose and manage virtualized infrastructures in the form of “baskets” of virtual resources and services that are made available through a testbed federation. With a different approach, BonFIRE targets the development of a platform for innovative experimentation on multi-site cloud facilities. LAWA, on the other hand, is mainly focused on developing an experimental testbed for large-scale data analytics—a kind of virtual web observatory—and is also based on a federation of distributed FIRE facilities.

The second group brings together initiatives such as OpenLab [13] and OFELIA [14], as well as a number of already concluded activities, such as PanLab [15], OneLab [16] and OneLab2 [17], and FEDERICA [18]. The portfolio of testbeds offered in OpenLab is rich and diverse and includes: PlanetLab Europe (PLE) [19]; the European Traffic Observatory Measurement Infrastructure (ETOMIC) [20] which offers high-precision network measurements; the

Network Implementation Testbed using Open Source code (NITOS) [21], along with other experimental facilities for testing applications and technologies in the areas of radio, networking, IMS, and data centers [13]. With a different target, OFELIA [14] is one of the initiatives trying to create a flagship test facility allowing experimentation with OpenFlow [22] in Europe. In this second group, we also found the PanLab initiative, which addressed the need for large-scale testing facilities in the areas of telecommunications and information technologies. PanLab implemented an infrastructure for federating testbeds based on the “Teagle” portal [23]. Teagle offers a central coordination instance allowing experimenters to express their testing needs, while also allowing testbed providers to browse, configure, deploy, and register new resources to be provided by the testbed federation. It is worth highlighting that, due to its numerous advantages, the Teagle portal is also being used in our own platform. In addition, OneLab and OneLab2 offered access to four testbeds—three of which were already mentioned—namely, PlanetLab Europe, ETOMIC, NITOS, plus the DIMES topology measurement infrastructure [24]. Last, but not least, FEDERICA focused on exploring ways of virtualizing and slicing an experimental network infrastructure, with the aim of delivering solutions for managing, controlling, and monitoring a set of parallel virtual networks.

In the third group, the initiative that is developing an experimental platform that shares part of our vision in the subject of living labs is EXPERIMEDIA [25]. The latter is developing a test facility especially devised for large-scale future media Internet experiments, and it includes the following testbeds: the Schladming Ski Resort [26], the Multi-Sport High Performance Center of Catalonia [27], the Foundation for the Hellenic World [28], and the 3D Innovation Living Lab [29].

All the initiatives cited thus far run under the FIRE umbrella [9], targeting different aspects around the experimentation challenges in the Future Internet. These initiatives are in one way or another making their mark on the subject, but, as mentioned above, they are not the only ones that are developing platforms for the experimental research sector in ICT in Europe. The flagship initiative of Future Internet Public Private Partnership (PPP) Programme, FI-WARE [30], goes beyond experimental research objectives and targets Future Internet professional services as well. It plans to build a Core Platform for the Future Internet, traversing way beyond the technical scopes of the three groups of FIRE initiatives, mentioned earlier. Among the several initiatives working towards developing a platform for Smart Cities, FI-WARE aims at becoming the open platform for Smart Cities in Europe.

In the US, the flagship initiative is the well-known “Global Environment for Network Innovations” (GENI) [31]. Unlike Europe, GENI is particularly focused on clean-slate research on networking, and to this end, US researchers have adopted a bottom-up approach. GENI is exploring networks of the future through the creation of a virtual Internet-scale laboratory, and, among the many activities carried out under the GENI umbrella, such as, INSTAGENI [32] and K-GENI [33], lies the generation of specifications for testbed management.

It is worth mentioning that similar initiatives are also being developed in Brazil, China, and particularly, in Japan. Among the most important ones in Japan are the activities undertaken by the National Institute of Information and Communications Technology (NICT) [34]. NICT is currently driving the development of the next generation of its national testbed, i.e., the Japanese Gigabit Network (JGN). This effort is being carried out under the framework of the NICT project called JGN eXtreme (JGN-X) [35]. As is the case with GENI in the US, the JGN-X initiative is also strongly focused on developing a platform for testing new networking paradigms, new protocols, new control and management features, as well as network virtualization mechanisms.

In this scenario, we consider that TEFIS has the potential to bring the experimentation processes one step further. Even though all the experimental platforms described above can handle the entire life-cycle for their corresponding experiments, TEFIS promises to add value by covering not only the testing phase during the software development life-cycle, but also the user-level of experience and satisfaction for a broad spectrum of Internet technologies and services. Indeed, along with FIRE, the European Network of Living Labs [36] provides an alternative, and complementary approach to Future Internet experimentation. A Living Lab allows for the testing of aspects which might not have necessarily been considered by the developers of the technology. Such aspects often include user motivation and real-world environment, identifying practical and additional demands for the systems and the services running on them, etc. As shown in Table 1, by combining test facilities that can offer the assessment of the quality and interoperability of software, with FIRE experimental facilities and Living Labs, our platform provides a powerful environment for conducting multifaceted experiments on a broad range of Internet services.

The latest trend in testing and experimentation facilities in Europe shows a keen interest in the concept of federation of cross-technology testbeds for providing a single platform to foster cross-domain innovative experiments for Future Internet. FED4FIRE [37] is one step in that direction which targets to make the numerous diverse initiatives of the FIRE [9] programme, mentioned earlier, reliably accessible through one platform. This way a federation of testbeds, apart from broadening the horizons for many possible and innovative experiments, also facilitates the path to make testbeds sustainable. The TEFIS ideology is very much aligned with this direction as it integrates different experimental facilities. We now proceed to describe in more detail the features of the experimental facilities that are currently accessible through TEFIS.

3. Experimental facilities offered through TEFIS

As outlined in Section 1, TEFIS allows for the combination and exploitation of experiments across heterogeneous test facilities through a common front-end. Fig. 1 captures the versatility, and thus, the potential that this platform has both for experimenters and for testbed providers. As shown in the figure, six different testbeds are currently

accessible through the TEFIS platform. To make this possible, a set of “connectors” has been developed, through which TEFIS basically hides the testbed-specific details of resource and data management from experimenters—the connecting interfaces will be described in Section 4. This level of abstraction clearly simplifies the experimenters’ tasks, and, as we shall show later in Section 6, it has also proved to shorten the learning curve for the experimenters, since they do not need to learn and deal with multiple (and heterogeneous) management systems for conducting experiments involving different test facilities. At present, the testbeds accessible via TEFIS are the following:

1. **ProActive PACA Grid [2]**—A grid computing facility for research labs and enterprises located at the “Institut National de Recherche en Informatique et en Automatique” (INRIA), in Sophia Antipolis, France. It was specifically developed to support computationally intensive experiments as well as to accelerate a plethora of scientific and commercial applications. It is generally used for performance analysis and scientific simulations, e.g., Matlab simulations, Monte Carlo simulations, financial computations, large scale algorithms evaluation, as well as distributed multi-disciplinary optimizations. The test facility is composed of a cluster accessible via graphical interactive interfaces based on the ProActive Parallel Suite (see <http://proactive.inria.fr>) and features 1400 cores, 480 Nvidia GPUs, Infiniband, Distributed File Systems, and currently provides 150 TB of storage. The cluster aggregates dedicated Linux and Windows machines that are manageable through various well-known virtualization tools, including VMware, OpenStack, Hyper-V, Xen, KVM, Qemu, and Amazon’s EC2. The grid is permanently available for INRIA and the University of Nice Sophia Antipolis (UNSA), and, upon request, other labs and enterprise customers can also access the test facility. International and national partners of R&D projects also enjoy access to the platform, including companies such as Renault, Sirhena-DCNS, Thales, and academic institutions, such as the National University of Singapore. The platform has been sponsored by INRIA itself, UNSA, PACA Lander, the European Union (EU), and ICT Labs.
2. **ETICS [3]**—The acronym stands for “e-Infrastructure for Testing, Integration and Configuration of Software”. ETICS offers an e-infrastructure that provides support for testing the entire software development life-cycle. This test facility runs mainly in a dedicated data center in Italy and is managed by one of its main developers, the Italian company “Engineering Ingegneria Informatica”. ETICS is essentially a distributed system for configuring, building, and testing software. It was designed to fulfill the needs of developers for improving the quality, reliability, and interoperability of distributed software in general, and of grid software in particular. In a nutshell, ETICS automates and improves the execution of builds and tests of

Table 1

Some of the most relevant initiatives in Europe, in the US, and Japan, and the test facilities that they offer. TEFIS is one of the European initiatives, and is being developed under the FIRE [9] umbrella.

Name of the Initiative	Test facilities offered							
	Software (quality and interoperability)	Grid computing	Networking technologies	Internet services	Living labs	Cloud computing	Virtualized infrastructures	M2M infrastructures
NOVI				✓		✓	✓	
BonFIRE				✓		✓	✓	
LAWA				✓				
OpenLab			✓	✓		✓	✓	
PanLab			✓					
OneLab			✓	✓				
OFELIA			✓	✓				
FEDERICA			✓	✓			✓	
EXPERIMEDIA			✓	✓	✓			
GENI			✓	✓		✓	✓	
GLIF/StarLight ^a			✓	✓				
G-Lab ^b			✓	✓			✓	
GpENI ^c			✓	✓			✓	
NorNet (Core) ^d			✓	✓			✓	
OpenFlow ^e			✓	✓				
JGN-X			✓	✓		✓	✓	
FI-WARE			✓	✓		✓	✓	✓
FED4FIRE			✓	✓		✓	✓	✓
TEFIS	✓	✓	✓	✓	✓			

^a Advanced experimental network research testbeds based on the Global Lambda Integrated Facility (GLIF) and the StarLight exchange [38].

^b Future internet research and experimentation: the G-Lab approach [39].

^c The GpENI testbed: network infrastructure, implementation experience, and experimentation [40].

^d NorNet (core) – a multi-homed research testbed [41].

^e Stanford SDN/OpenFlow network testbed [42].

distributed, multi-language, and multi-platform software and is also able to provide meaningful measurements of overall software quality. ETICS is offered as a service, on the basis of a shared computing infrastructure that offers new, and improved features to software professionals. The ETICS testbed is based essentially on a software suite for build, test and quality certification developed over the past 8 years by a consortium led by the CERN research centre. The testbed is mainly used by companies and partners within EU research initiatives. ETICS offers an automated way to build any software against any (virtualized) platform and perform a set of tests, e.g, deployment and functional tests. ETICS is critical for ensuring that any software is correctly built and tested before being installed on other premises. In this way, ETICS not only reduces the need to debug potentially error-prone applications, but also supports the compilation of code on the right operating system as well as some initial functional tests that may reduce the complexity of working in heterogeneous and distributed environments.

3. **SQS-IMS** [4]—This test infrastructure was devised for telecom operators and service providers, with the aim of testing and validating protocols and mobile applications over the IP Multimedia Subsystem (IMS) [5] prior to release. The test facility is located in Spain, and allows different aspects of an IMS service to be examined, ranging from functional aspects, such as network and resource-specific loads, up to the compliance with

the existing regulatory framework. The testbed is often used by service providers; for instance, for running interoperability tests and also for examining if their services and the technologies to be deployed are in compliance with specific requirements and standards.

4. **The Botnia Living Lab** [6]—Based in Sweden, this living lab offers support for the evaluation and testing of Future Internet-based ideas, concepts, and prototypes with end-users, wherein the term “end-users” means individuals using IT-based services in their private lives. The availability of the Botnia Living Lab through TEFIS offers a powerful combination for developers and experimenters, since TEFIS provides a single access platform for testing not only the objective aspects of a new technology but also the subjective ones. Overall, the Botnia Living Lab is a testing facility for human-centric research, allowing the assessment during the design and development phases of Future Internet prototypes by end-users. Botnia offers methods, tools, and expertise, as well as access to users for user-testing and evaluation, targeting experimenters both from academia and from industry. Within the environment of a given technology, BOTNIA can offer meaningful evaluation in real life contexts, including scenarios in which the users become co-producers.
5. **Kyatera** [7]—This test facility offers a high-speed optical network located in Brazil, which was designed for supporting collaboration and experimentation with technologies and applications with

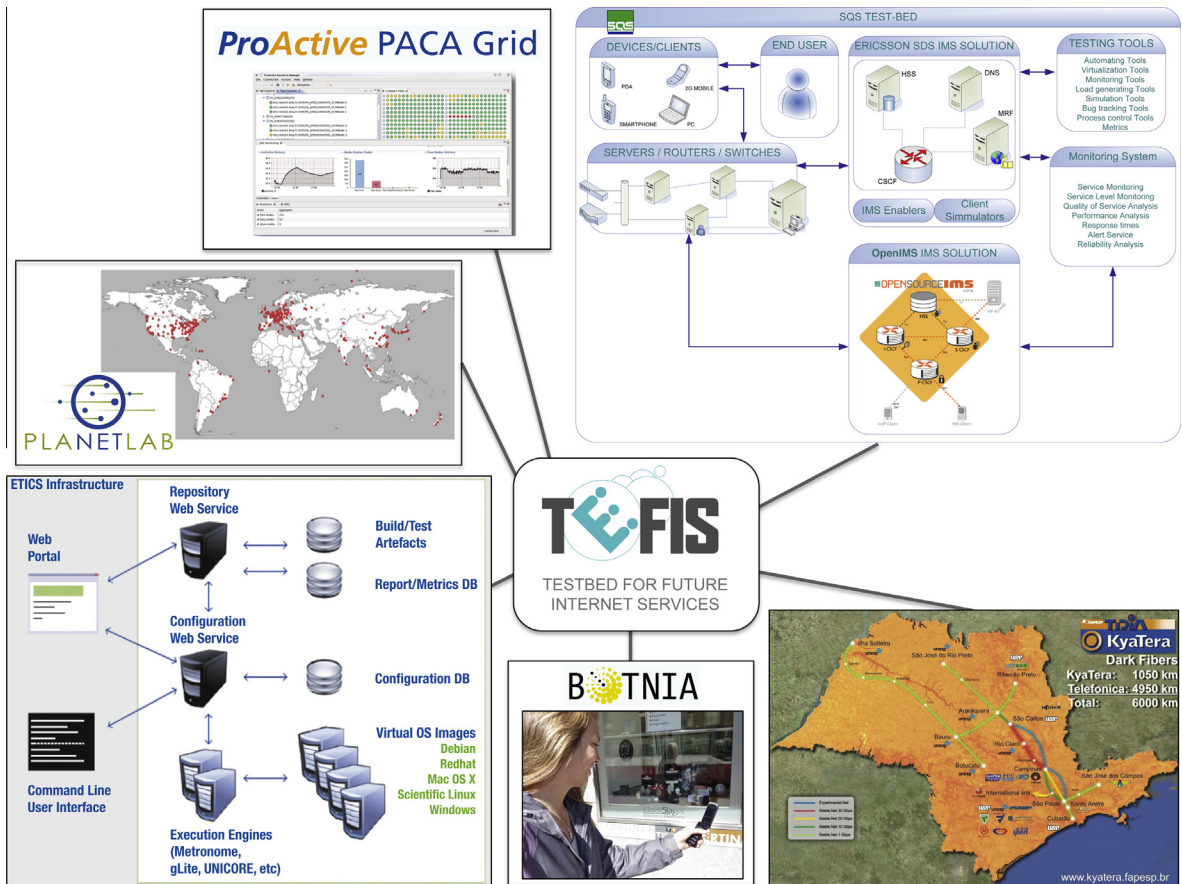


Fig. 1. Six experimental facilities are currently accessible through TEFIS.

large bandwidth demands. KyaTera provides tools for measuring the quality of the services under test, thereby facilitating the specification of the network requirements to run an application with a given level of service. More specifically, the testbed provides the ability for experimenters to evaluate the performance of Internet services, such as multimedia services or critical Data Services. The testbed offers access to advanced network capabilities which can be exploited for investigating different aspects of transmission and quality of service.

6. **PlanetLab** [8]—This testbed represents a large-scale distributed system that offers a set of geographically distributed nodes for experimenters. It is typically used to deploy and run tests on a slice (an independent overlay network composed of a pool of nodes chosen by the experimenter). PlanetLab was basically created to support distributed systems and network research on a large scale over the current Internet. It has become a well-known test facility composed of more than 1000 nodes distributed worldwide and is mainly used by research institutions and industrial labs for testing new technologies in areas such as peer-to-peer systems, distributed storage, distributed hash tables, query processing, etc. It is particularly useful for testing

and validating network protocols and large-scale distributed systems under relatively realistic conditions. At present, this test facility is managed by a centralized authority.

It is worth noting that the pool of testbeds described above are simply the ones that are currently accessible through the platform, but this does not represent in any way a closed list. As we shall show in Section 4, the architectural design of TEFIS is general in scope, and therefore, testbeds can be added or removed as required. In any case, the value for experimenters lies at the end of the day in the list of test facilities offered, in the simplicity of their use (though without compromising versatility), and in the potential synergies of combining heterogeneous testbeds for conducting multifaceted experiments. These are the premises on which TEFIS was conceived. Indeed, by building on top of the ongoing actions for supporting large-scale experimentation for Future Internet services, TEFIS fosters the vision of Testbeds as a Service (TaaS), with the twofold goal of making easier the access to different test facilities and simplifying the running of experimental tasks on them.

In summary, for researchers and experimenters, TEFIS offers the evaluation of multiple facets, such as functionality, performance, scalability, maintainability, standards compliance, as well as usability including user experience

and acceptability. For testbed providers, we consider that our platform could offer greater market opportunities, and could also help to identify future requirements for multifaceted experimentation.

4. The TEFIS platform

The design of the TEFIS platform is based on a bottom-up approach that covers all the requirements identified for conducting multifaceted experiments on different testbeds. As shown in Fig. 2, the overall architecture is broken down into three functional blocks, namely, the TEFIS portal (i.e., the user interface), the TEFIS Middleware, and the Testbed Connectors. The contribution to each functional block is well-balanced in order to provide a full-fledged platform which facilitates coordinated usage not only to the final user (i.e., experimenter) through the TEFIS portal but also to potential TEFIS administrators and testbed providers through the connector framework. This section is devoted to introduce the main components of this architecture, and their role in supporting the motivations that led to the development and implementation of the TEFIS platform. Although other design approaches are certainly feasible, we claim that the architectural model introduced here is sufficiently general, and captures the key components that should be present in any platform that targets our goals. The magnitude of the TEFIS initiative does not permit us to give here a complete explanation of every component in the platform. For more detailed explanations, the reader is referred to [1], and for information on the release and license of TEFIS software modules, to [43].

4.1. The TEFIS portal

The TEFIS portal offers a common user interface for experimenters and testbed providers and is the main entry

point for accessing all the test facilities available through TEFIS. Our portal provides the service-level intelligence that allows for customization of the testing tools, as well as the orchestration methodologies that give access to the required resources for the execution of experiments. It consists of the following five interfaces.

–**The Directory Interface** exposes the list of tools, facilities, and resources provided by the different testbeds, allowing users to obtain information and documentation in relation to the usage and availability of specific experimental resources.

–**The Identity Management Interface** implements the user interface for creating accounts and managing user profiles. It also unveils the user account management features both for experimenters and testbed providers.

–**The Experiment Manager Interface** enables the user to define and structure the experiments to be performed on the different testbeds. It uses the TEFIS directory services to list and configure the available resources, and plan the experiment execution. It also supports the collection of data, in order to extract the results after experiment execution. Note that, the TEFIS portal provides a level of abstraction to ease the process of designing experiments based on a graphical experience. The experiment designer can either select upon a set of predefined tasks—which are specific to each testbed (e.g., validation, testbed initialization, result analysis, etc.)—or design his own custom tasks combining resources and scripts, in a drag and drop fashion, to build his own experiment. Fig. 3 shows an example of a custom task designed through the TEFIS portal. For more details on how to create an experiment through the TEFIS portal refer to Section 5.

–**The Experiments Data Interface** exploits the data management services required for the entire experiment

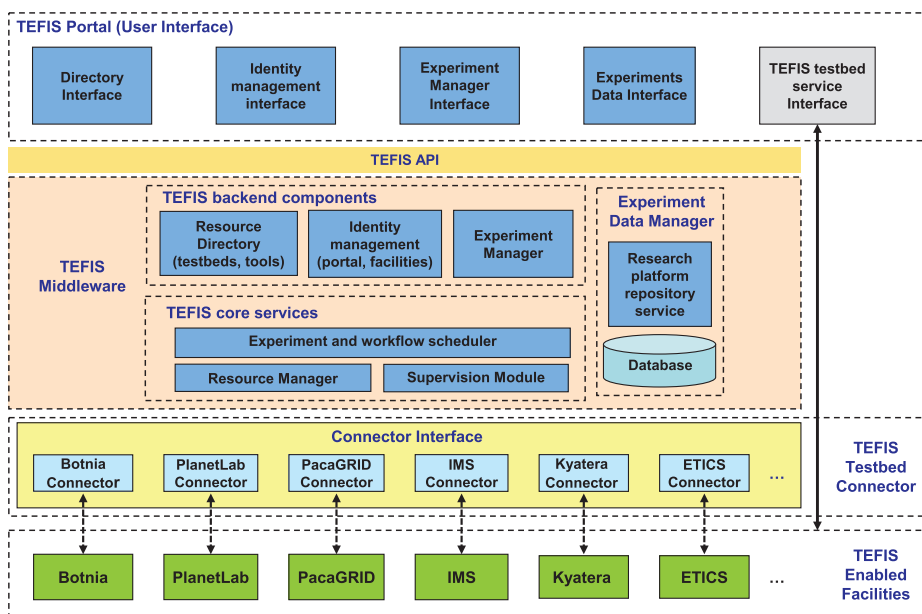


Fig. 2. Outline of the main components of the TEFIS architecture.

life-cycle. The data management services are essential for the design, execution, and storage of experiments as well as for the requisition and utilization of the required testbed resources. It allows users to search for existing experiments, e.g., to locate experiments with similar goals and/or setup, as well as to interact with monitoring data and experimental results once the experiment or an individual testrun completes.

–**The TEFIS testbed service Interface** exposes the platform for testbed providers. To this end, it offers a simple and efficient way for testbed providers to integrate their testbeds into TEFIS. More details on the integration of testbed facilities will be provided later in Section 4.3.

Overall, the portal offers a user-friendly and valuable web interface for experimenters and testbed providers. Through the portal, the former can schedule and use the resources exposed by a number of experimental facilities, as well as collect the results obtained once their experiments have concluded. As for the latter, the portal provides a simple tool for registering and integrating their test facilities into the TEFIS platform. The web-based access to the portal using standard web technologies (e.g., Apache, HTTP, PHP, etc.) assures its scalability in face of a large number of users.

4.2. The TEFIS middleware

As depicted in Fig. 2, the role of the TEFIS API is to expose the TEFIS Middleware functionality to the TEFIS portal. The TEFIS Middleware constitutes the core of the TEFIS platform, as it implements all the management and test execution logic. It consists of three main blocks, which are the TEFIS Backend Components, the TEFIS Core Services, and the Experiment Data Manager.

4.2.1. TEFIS backend components

The Resource Directory—This component manages the repository of tools, facilities, and resources provided by

each testbed. It provides the list of resources for each experiment according to their respective needs and requirements. It also allows for check-in and check-out of software, hardware, laboratories, documentation or any other resource that the repository accounts for. The repository exposes an HTTP-based RESTful interface, as inherited by the Teagle framework, with a number of REST resources presenting the repository as a server entity. Since REST is a stateless protocol, it makes the Resource Directory highly scalable, as all the complexity of state management is moved to the client side. The underlying data model used within the repository is based on the DEN-ng information model [44].

The Identity Manager—The Identity Manager unifies the different authentication and authorization mechanisms handled within the TEFIS platform. By making TEFIS and its facilities accessible via a uniform interface, the experimenters and testbed providers only need to perform authentication once at the TEFIS portal to get access to the components and testbeds allowed. The mechanism is based on *Registration*, on the utilization of *Proxy User Accounts* (used by the TEFIS internal components), and on the management of *Testbed Credentials* (used by the testbeds registered within TEFIS).

The Experiment Manager—Its main goal is to give experimenters the ability to define, configure, and execute experiments, along with the on-line reception of notification alerting upon events or errors. The internal logic of TEFIS's Experiment Manager is performed by Teagle's Directory Services [23]. Among its numerous advantages, including scalability, Teagle's Directory Services natively offers features such as resource configuration, experiment execution, and planning. The Experiment Manager is divided into five interrelated blocks:

- *Experiment Designer*: allows the different parts composing an experiment to be defined. It works by mapping the right tools, testbeds, and methodologies to the requirements of the TEFIS user. To aid experiment

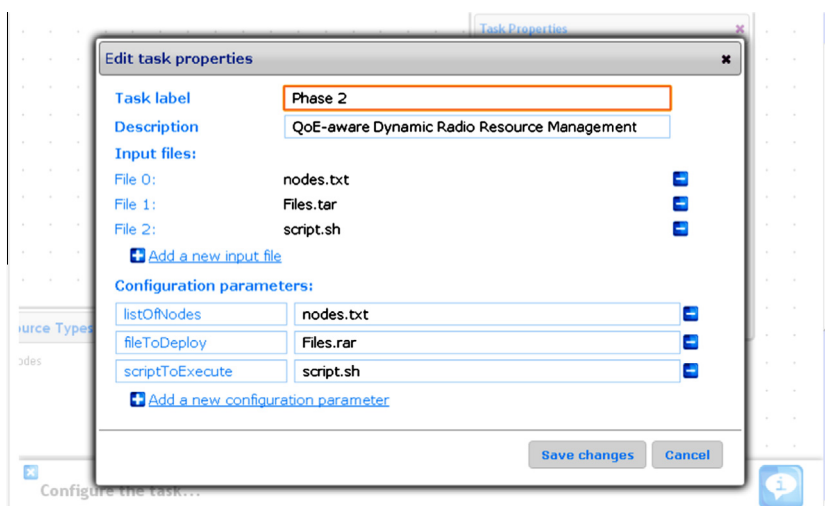


Fig. 3. Design of a custom task through the TEFIS portal.

design, this block offers three different alternatives. First, it is possible to select an experiment from a pool of preset models. Second, experimenters can reuse an already existing experiment; and finally, expert users can create the experiment from scratch. All three alternatives allow customization of their configurations.

- *Experiment Planner*: is in charge of creating a test plan. That is, an abstract view of the experiment, where the Experiment Planner details the set of possible tasks and resource types, as well as allows the booking of such resources at the destination testbeds.
- *Experiment Workflow Manager*: By using the Test Plan definition, the Workflow Manager provides a high-level abstraction capturing the execution workflow of the experiments. Specifically, the Workflow Manager defines the specific testruns of the experiments, which are an instantiation of the settings specified within the Test Plan for each execution.
- *Configuration Assistant Manager*: Provides a unique configuration method for all the different tasks and parts of the experiment workflow. It allows for the definition of variable parameters for each testrun from the Experiment Workflow Manager, along with the particular settings for each experiment.
- *Internal TEFIS Interface*: This block is responsible for interfacing with other TEFIS components within the TEFIS platform, in particular abstracting the interaction with Teagle's Data Services, other core services, or the Experiment Data Manager (see Fig. 2).

4.2.2. TEFIS Core Services

The Experiment and Workflow Scheduler (EWS)—This is the component in charge of the enactment and orchestration of the experiment workflows over the testbeds available at TEFIS. In effect, it provides the execution engine for the TEFIS platform. The EWS implementation is based on the ProActive Scheduling tool [45], a batch scheduler for the execution of workflows on a shared set of computing resources. The ProActive Scheduling tool allows workflows containing one or several tasks (i.e., JAVA or NATIVE), conditional branches, loops and replication, and then assisting the execution of these tasks on the distributed computing resources. This management approach gives TEFIS's platform unique capabilities for seamless resource sharing and resource management among the different users in the system, while ensuring scalability and the failure management of the applications. Moreover, one of the most relevant features available through this design is the simplicity of workflow definition. In particular, an experimenter focuses his/her efforts in the high-level definition of workflows, i.e., the experiments, rather than actually having to write low level code during the experiment design process. The experiment workflow represents and specifies the sequence of activities and testbed resources on which those activities are to be performed.

The TEFIS Supervision Manager (TSM)—provides monitoring capabilities within the TEFIS environment. The TSM identifies three main stakeholders: (i) the experimenter; (ii) the TEFIS administrator; and (iii) the testbed providers. For the experimenter, it monitors the different resources used by the experiment, along with any possible

application-specific output. For the TEFIS administrator, it keeps track of the status of the platform itself. And finally, for testbed providers, it manages domain-specific resources and keeps track of resource usage at the specific testbed.

The TEFIS Resource Manager (TRM)—This is the service responsible of monitoring resources status, storing information on available resources, while providing the Experiment and Workflow Scheduler access to the resources needed as specified in the workflow to be executed.

4.2.3. Experiment Data Manager (Data Services)

The TEFIS Data Services are central to all processing on the platform, as they provide a centralized repository to gather and manage the information related to all the TEFIS components. The Identity Manager creates and manages TEFIS Proxy User Accounts; the Experiment Manager creates the folder structure to store all the data associated with the experiment; the Supervision Manager links monitoring and other experimental data; and the Connector Interface is used to transfer data into and out of TEFIS for the testbeds. Therefore, the Data Services provide a generic framework for active and stateful communication among the different steps during the experiment life-cycle. As can be observed in Fig. 4, the Data Services for an entire experiment are distributed across two logical environments, the Research Platform Repository Service (RPRS), and the Testbed Infrastructure Data Services (TIDS). The former provides a repository service to the TEFIS platform and the TEFIS middleware components. The latter provides the data management services associated with the TEFIS repository within the remote, testbed environment. The RPRS is built on iRODS technology [46], which provides a virtualized filesystem and the associated tools to be able to locate and search any data within the filesystem based on the metadata assigned to the data objects. iRODS was chosen because of its fitness for our purposes, scalability as well as its maturity and highly active user community. In particular, these services will assist in the management of the execution workflow during the experiment runtime. It is worth highlighting that the Data Services built on iRODS provide a RESTful interface for communicating with the other internal components of TEFIS. The RESTful interface allows a client/server style communication, which perfectly suits TEFIS's internal architecture. The REST architecture brings robustness and scalability along with its performance guarantees to the TEFIS core, as also demonstrated in other domains, such as [47,48]. Once the experiment execution is finished, all the data gathered and its meta-information are made available for the experimenter, who can choose to share them with other experimenters or even use them in order to assist in the development of future experiments. Nevertheless, it is important to mention that the experimenter has full control over the entities able to access these data.

4.3. TEFIS connectors

The TEFIS connectors have a twofold goal. On the one hand, there is the necessity to implement a generic, robust,

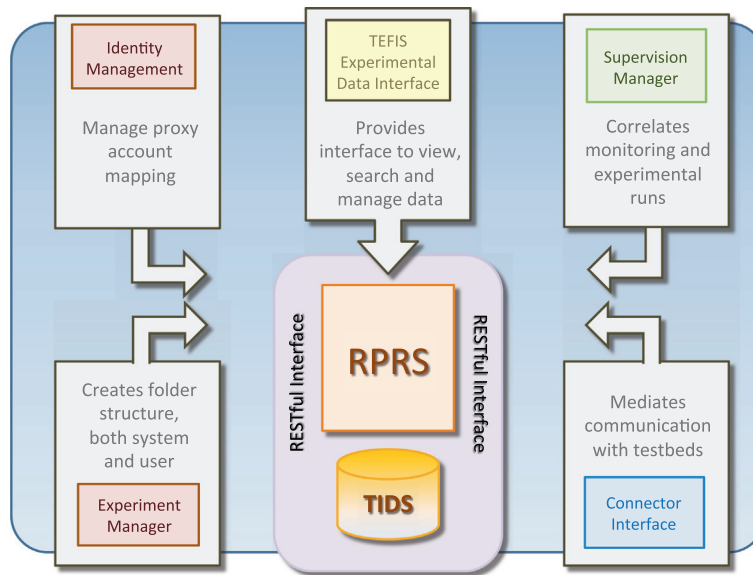


Fig. 4. The TEFIS functional architecture on the basis of Data Services.

and reliable interface between the testbeds and TEFIS. On the other hand, testbed providers must be able to integrate a new testbed into the TEFIS platform as easily as possible. In reference to the first goal, the connectors will translate the generic TEFIS experiment, i.e., resources, tasks, and workflows, to specific configurations, commands, and scripts particular to each testbed. As a result, it is the task of the connector to perform all the necessary processing and adaptation of the information to suit both the TEFIS platform and the testbed itself. The general overview of the interconnection of these features can be observed in Fig. 5. As for the second goal, i.e., the integration of new testbeds, TEFIS proposes to use a TEFIS Connector Interface (TCI), which captures the set of operations that every testbed integrated into TEFIS must implement. Note that the mapping between the testbed specifics and these set of operations may not be direct, however, connectors are responsible of semantically translating general TEFIS concepts to testbed-specific ones.

4.3.1. The TEFIS Connector Interface (TCI)

As mentioned above, in order to abstract the interconnection between the testbeds and TEFIS, it is very important to determine the set of exposed operations (API). To this end, TEFIS defines five subservices (see Fig. 5), which aim to categorize all the functions that will be available to promote the interaction between testbeds and the TEFIS platform. This information is essential for testbed providers to properly implement connectors to their testbeds.

- **Resource Management:** Prior to the execution of an experiment, the system must be able to assess whether it is possible to run the experiment, and, whenever possible, trigger the corresponding resource booking mechanism. To illustrate the functions exposed through the TEFIS Connector Interface Resource Management (TCI-RM) we present in Table 2 a set of these methods which

are based on Teagles' Panlab Testbed Manager (PTM) entity T1 interface, meaning that, integration with the TEFIS Middleware is thereby seamless. These methods target functions related to the management of resources while abstracting the "resource" concept in the context of each testbed. The concept of resource is in fact something like a class in the OO programming language world, this is, a description of a resource that can be instantiated in one or more running instances. Every "running instance" of a resource is uniquely identified and is characterized by a particular configuration, where a configuration object is the set of parameters that characterize that particular resource.

- **Execution:** The ultimate goal of this subservice is to generalize the different execution engines provided by the different testbeds under a single API. This part of the TCI will expose methods to run custom executable scripts in the testbed. The main method is the `execute()` function which takes as a parameter an executable entity, the resource where the execution will take place and the experiment structure and returns an identifier that can be used later onto retrieve the status of the execution, as follows: **execute (executable: ExecutableEntity, resource_id: Identifier, exp: Experiment): uuid**. Other operations to control the execution of the experiment have been defined in the set of methods of the TCI-EXEC, for example, for periodically polling connectors to retrieve job status, pause, stop or cancel execution, among others; for more details, please refer to [44].
- **Data Management:** This subservice allows the TEFIS platform to access the internal testbed monitoring data. Accordingly, it makes available the experimental results for the Experiment Data Manager. The TCI-DM part of the TCI interface will expose all methods to access experimental data, including monitoring data. Table 3 summarizes some of the main methods exposed for this purpose.

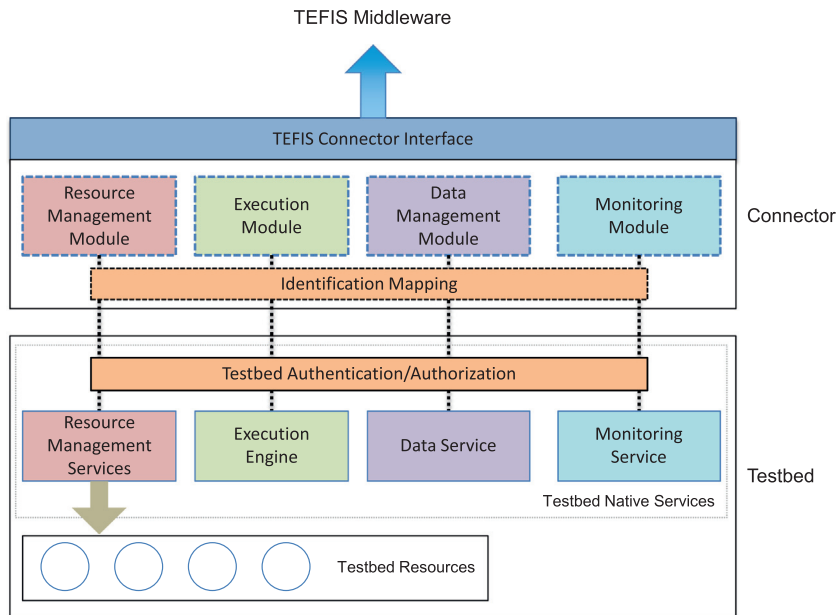


Fig. 5. The TEFIS Connector Interface (TCI) and its main components.

- **Monitoring:** After analyzing the requirements regarding monitoring, it has been observed that its requirements are very much similar to the ones related with Data Management. The main difference lies on the fact that monitoring is usually an on-line process, while the gathering of results can be performed after the experiment execution.
- **Identity Mapping:** Finally, all previous functional blocks need to interact directly with the testbed, which in practice implies that they need to have the proper authentication and authorization tokens. The Identity Mapping abstracts the details required in order to authenticate and authorize the TEFIS user through a well-defined interface. TCI-IDM methods allow TEFIS to control the identity bindings on testbeds. The API is composed of three methods to create, remove and get the identity bindings. Table 4 summarizes the methods of the TC-IDM interface.

It is important to notice that these different functional blocks are wrapped up in a unique entity, which exposes

a complete set of functionalities to TEFIS from the testbeds. For the sake of simplicity, the TCI aims to provide a level of abstraction to experiment designers at the cost of pushing the complexity to the testbed providers side. In this sense, the TCI's counterpart on the testbeds are developed by experts within each domain, in order to wrap up testbed-related complexities and expose them as simple and concrete functionalities to the TEFIS user. The fact that some functionalities of the original testbed are not available or exposed in a different way (i.e., simpler—usually) is one of the goals of TEFIS, in order to simplify configuration and usage of testbeds for non-experts, as well. So far, TEFIS does not consider automatic generation of connectors, mainly because of the extreme heterogeneity of testbeds, which makes automatic generation of connectors very complex (e.g., at this stage, the Botnia LivingLab does not have a programmatic interface at all). There have been studies for mediating connectors for heterogeneous systems, e.g., see [49], which might be considered for automating connector generation in the future.

Table 2
TCI-RM main methods.

TCI-RM method	Description
add_resource (parent_id: Identifier, typename: TypeName, name: LocalName, config: Configuration, vct: VCTName): Identifier	Request the PTM to create a resource of the type passed on the testbed and with the Configuration provided. The identifier of the created resource will be returned
get_resource (identifier: Identifier): Configuration	Return the configuration (that is the object that actually characterizes the resource) for the ID passed
update_resource (id: Identifier, config: Configuration): Configuration	Change one or more values for the parameters of the specified resource
delete_resource (identifier: Identifier): None	Delete the resource
list_resource (parent_id: Identifier, typename: TypeName): Identifier	Return the list of resources that match the listing parameters

Table 3
TCI-DM main methods.

TCI-DM method	Description
getData (dataId: string, exp: Experiment): string	Retrieve data. The actual data will not be exchanged, but an identifier pointing to the real data (e.g., URI)
storeData (dataId: string, exp: Experiment): Boolean	Store Data. Similar to previous operation data will not be exchanged, but instead an identifier pointing to the real data to be stored
startRecordMntData (exp: Experiment)	Start recording monitoring data for a specific testbed
stopRecordMntData (exp: Experiment)	Stop recording monitoring data for a specific testbed
queryMonitoringDB (query: string, exp: Experiment): string	Query monitoring DataBase

Table 4
TCI-IDM main methods.

TCI-IDM method	Description
getIdentityBinding (tefisUser: string): Credentials	This function is intended to be used by TEFIS to retrieve information about an existing identity binding. Null/void will be returned in the case when the requested binding has not been found
updateIdentityBinding (tefisUser: string, testbedCredential: Credentials): Boolean	This function is intended to be used to update the associated Credentials of an existing binding
removeIdentityBinding (tefisUser: string): Boolean	This function will remove the identity binding on the connector

5. Creating and running experiments through TEFIS

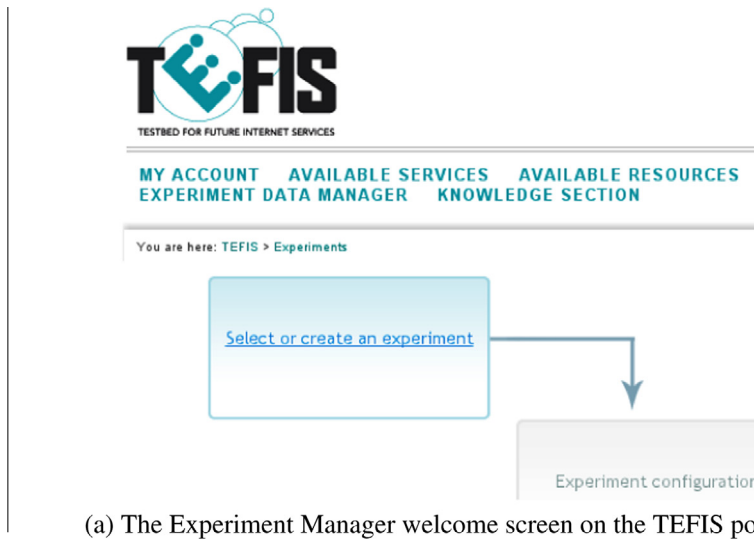
In this section, we describe the life-cycle of an experiment in TEFIS from a user's perspective, while outlining the internal functions of TEFIS as well. Firstly, we provide an overview about the experiment's life-cycle, which, as we shall show in Figs. 6 and 7, cover the creation of a new experiment, requesting and provisioning of required resources, deployment and execution of the experiment, gathering of results, and finally, the publishing of results and the experiment itself through the TEFIS portal. Then, we list the TEFIS internal workings elaborating how an experiment is actually executed within TEFIS, and how the individual TEFIS modules interact with each other, based on an example experiment illustrated in Fig. 8.

The experiment can be planned, designed, deployed and executed according to the requirements through the user interface of the Experiment Manager. As described in Section 4, prior to designing the experiment, the experimenter needs to register at the TEFIS portal. Once logged into the TEFIS portal, the experimenter can click on "Select or create an experiment" to create a new experiment or select an already existing one, as shown in Fig. 6(a). This is an important feature of TEFIS, since it allows new experimenters to browse through the list of published experiments to improve their learning curve about the TEFIS portal. As it can be observed in Fig. 6(b), the experimenter can select or search for an existing experiment by using the options "Select one of your existing experiments" or "Search an experiment", respectively, from the top navigation bar. For the new experiment, the experimenter can click on the "Advanced experiment configuration" option to select the respective testbeds and required resources according to the design of the new experiment. As depicted in Fig. 7(a) next, the experimenter designs and configures the flow of execution of the experiment by creating tasks and binding particular resources to these tasks such that

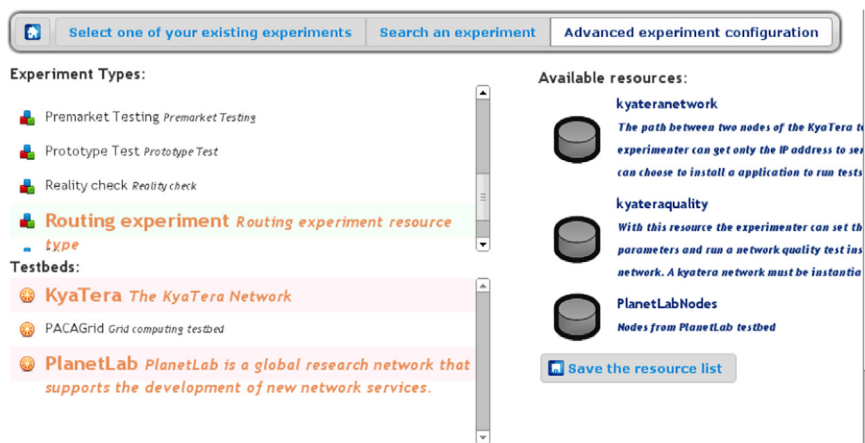
the defined tasks are executed on the configured resources. The TEFIS portal provides total freedom in the definition of tasks by allowing the experimenter to upload input files, scripts, etc., and configure selected resources on a granular level to tailor the experiment execution according to the experimenters' needs. Then, the experimenter creates a *testrun*, that is, a particular instantiation of the generic resources and tasks defined in the previous step, and executes the experiment. As shown in 7(b), after the execution, the results can be fetched from the Experiment Data Manager in the form of text files and images. Furthermore, the TEFIS portal not only allows experimenters to save their experiments and results but also to publish them in the TEFIS knowledge base, which enables a jump start for a new experimenter with little knowledge of the testbeds. Due to space limitations in this paper, we present only a few important snapshots of the TEFIS portal, but the latter assists the experimenter, step-by-step with a user-friendly interface, throughout the entire life-cycle of his experiment. Therefore, interested readers are encouraged to visit the TEFIS portal to appreciate the complete experience of TEFIS.

Fig. 8 depicts the TEFIS experiment life-cycle and the execution internal workings for an already existing experiment that uses two different testbeds. The figure includes the following steps:

1. The experimenter logs into the TEFIS portal.
2. The experimenter selects an already designed and created experiment in the Experiment Manager.
3. The Experiment Manager requests resources from the Resource Manager.
4. The Resource Manager retrieves information about the resources from the Resource Directory.
5. The Resource Manager then provisions adequate resources at the respective testbed(s) (e.g., ETICS and PACA Grid).



(a) The Experiment Manager welcome screen on the TEFIS portal.



(b) Requesting and provisioning of testbed resources.

Fig. 6. Example showing the creation of an experiment through the TEFIS Web portal.

6. Upon completion of provisioning, the Resource Directory is updated regarding the allocated resources.
7. The Experiment Manager now submits the experiment to the Experiment Scheduler for execution.
8. Depending on the defined experiment workflow, the Experiment Scheduler sets off the first task on the corresponding testbed (which is ETICS in this example).
9. Upon task completion, the first testbed returns output data to the Data Service block, i.e., to the Research Platform Repository Service (RPRS). This is done into the appropriate folder defined for this purpose.
10. Depending on the experiment workflow, the Experiment Scheduler initiates subsequent execution on the second testbed (here, on the PACA Grid facility).
11. This testbed retrieves input data for its test run from the RPRS, which may be the output from the previous stage—note that TEFIS supports concurrent execution of experiments in the two testbeds, so it is the experiment designer who decides through the

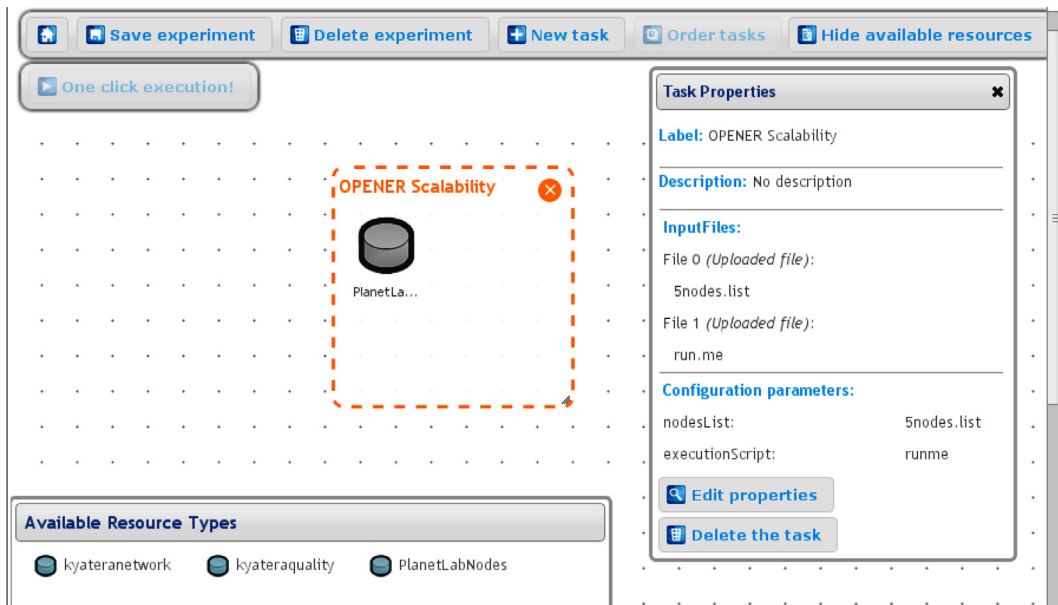
workflow definition whether the execution should be concurrent or sequential.

12. Upon completion, the second testbed returns its output to the respective folder in the RPRS.
13. Triggered by the Experiment Scheduler, the Experiment Manager returns a completion notification to the experimenter.

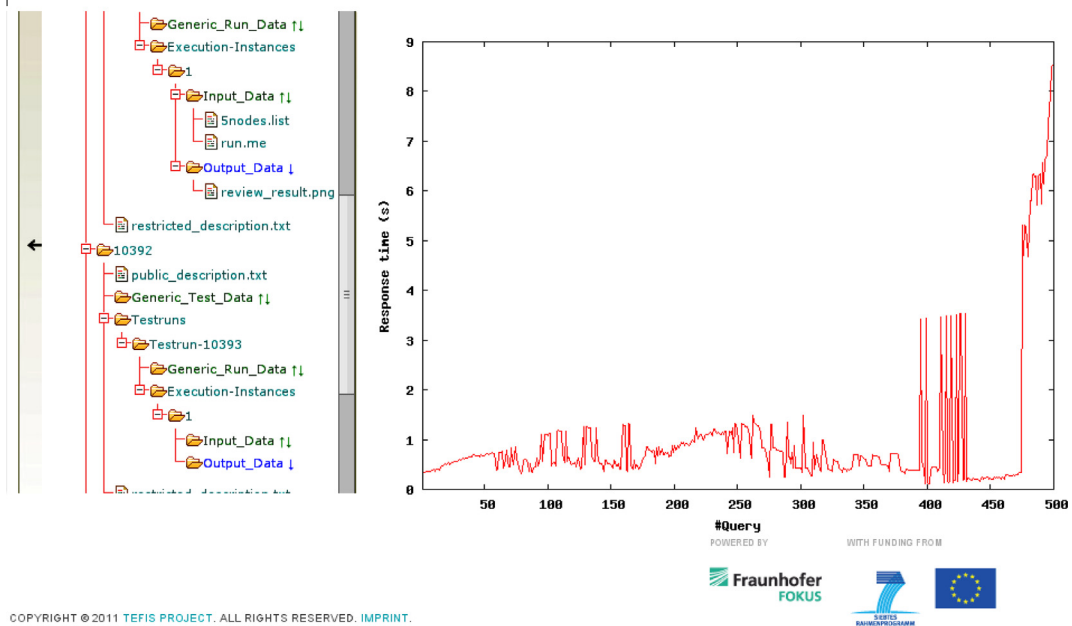
Most significantly, once the experimenter has commenced the experiment run, irrespective of how many testbed facilities are involved, and assuming there are no specific issues with the execution, the experiment is completely managed throughout its life-cycle by the TEFIS platform requiring no intervention from the experimenter.

6. TEFIS use cases

In this section, we describe four real experiments that were tested on TEFIS. This kind of experiments are the ones



(a) Defining tasks and binding resources according to experiment design.



(b) Obtaining results through the Experiment Data Manager.

Fig. 7. Example showing task definition and result gathering of an experiment through the TEFIS Web portal.

that ultimately demonstrate the versatility of our platform for conducting multifaceted tests. We provide a short description about the motivations expressed by the experimenters for conducting their tests on TEFIS, as well as the problems tackled by each of the technologies that were assessed through our platform. In this sense, we also highlight the value of the TEFIS platform in the context of each particular experiment.

6.1. The OPENER experiment

OPENER [50] is an open and programmable tool for managing experimentation with SDN applications. OPENER provides an open access to network element's capabilities, enabling the creation of out of the box applications, which can extend the existing features on network nodes. Moreover, OPENER is not dependent on any vendor-specific

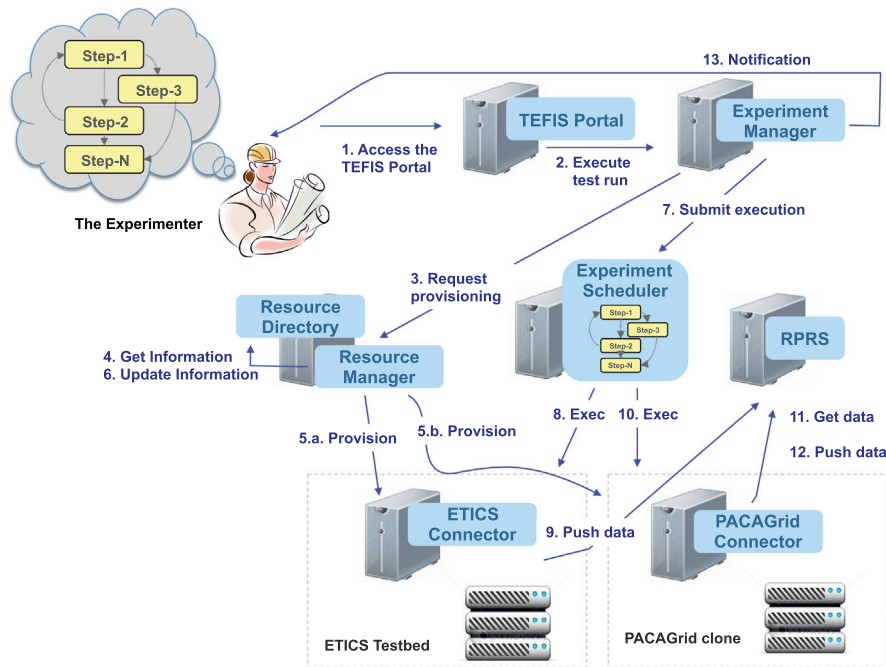


Fig. 8. Example showing the execution of an experiment through TEFIS using two different testbeds.

platform, but rather it is devised to foster non-proprietary solutions, where researchers can easily manage and experiment with SDN applications. In order to offer such an open and programmable environment, OPENER provides a set of interfaces, where the accessible internal features, e.g., routing protocols, interface management, and so on, are exposed to third-party applications. Such enhanced features allow higher levels of automation, management, and configurability than with regular Command Line Interfaces (CLIs).

The experiment consisted of two phases within TEFIS. Firstly, to use TEFIS (with PlanetLab) as a debugging and performance optimization platform for the initial implementation of OPENER's core modules. Secondly, to test a use-case involving OPENER and other third-party applications, to orchestrate automatic IP-offloading between the PlanetLab and Kyatara testbeds, with the objective of optimizing traffic across the network. An illustrative example depicting the goals of the second phase of this experiment can be seen in Fig. 9.

Phase 1 (Performance assessment): The goal in the first phase of the experiment was to assess the performance and scalability of different deployments of the OPENER framework. When developing a network management application, scalability is a key concern, given that such application must allow high concurrency and robustness under stress situations. As a consequence, one of the main goals of this experiment, was the optimization of OPENER to achieve deployments of realistic network sizes, including large scale deployments. In reference to this phase, TEFIS provided a comprehensive and user friendly platform allowing efficient repetition, management, and configuration of the experiment to aid exhaustive

debugging. Subsequent optimizations on the experimenters' tool were evaluated through TEFIS by running a set of benchmarks iteratively, e.g., computation of query response times depending on the number of concurrent managed nodes, while assessing both the performance and the resource usage for each test. In each iteration, the code was optimized for better concurrency level using the knowledge provided by the experimental results, which at the end provided a more resilient codebase for OPENER.

Phase 2 (IP-Offloading): The second phase of the experiment aimed at a real use case of the OPENER framework by targeting coordinated cross-layer interactions in multi-layer scenarios. To this end, a third-party application was deployed and tested—an IP traffic offloading solution in this case. This application allows smart orchestration of IP and transport resources, so as to optimize their usage by offloading part of the traffic between two IP routers through a different optical path. Fig. 9 shows an example deployment, both in Kyatara and in PlanetLAB. As can be observed from the figure, the goal of the experiment is to offload partial IP traffic when the link utilization reaches a particular threshold. This was orchestrated through OPENER by monitoring the utilized bandwidth at all network links. The network traffic was gradually increased by launching a video streaming application and bulk data transfers, and once the network reached a pre-configured threshold, the system offloaded part of the traffic towards Kyatara in order to maintain the quality of the video streaming. The experiments carried out in TEFIS confirmed the adaptability of OPENER to provide coordinated management and enhanced functionality to the IP layer. During this phase of the experiment, TEFIS provided seamless

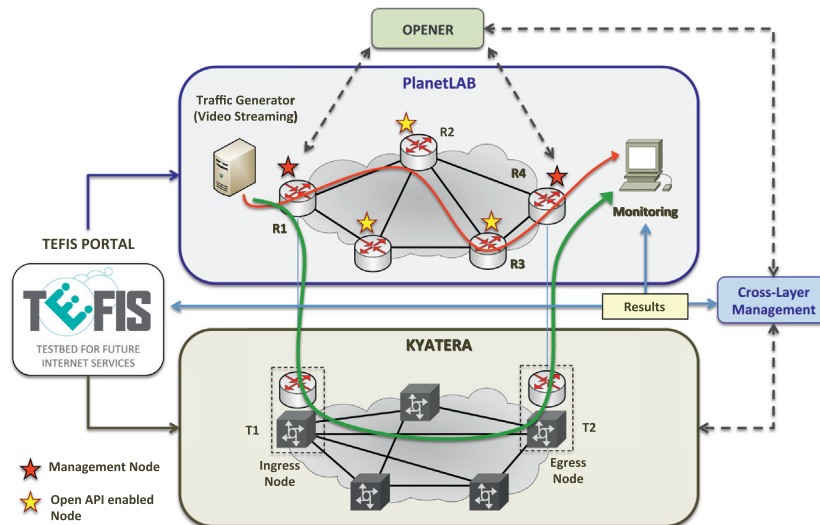


Fig. 9. The second phase of the OPENER experiment in a nutshell.

interconnection between the Kyatera and PlanetLAB testbeds through a single platform, hiding all the particular details of each testbed, which was particularly useful for the experiments.

Overall, TEFIS played a vital role in this experiment, since it provided an abstract mechanism to configure, execute, and most importantly repeat relevant tests to disclose several issues present in OPENER's codebase.

6.2. The Smart Ski Resort experiment

The "Smart Ski Resort" experiment targeted improving the overall performance and user experience of a mobile application used to augment the skiing experience of visitors while on the slopes. This "Smart Ski Resort" application was developed by LUMIPLAN [51] for iPhone and Android platforms, to be used at the Megève skiing resort in France. As shown in Fig. 10, the application requires heavy multimedia content, including ski slopes videos and maps, and the underlying available infrastructure struggles to scale or stay user-friendly, as foreign tourists cannot enjoy all the functionalities due to expensive roaming charges. Furthermore, intermittent connectivity due to limited 3G coverage area and bandwidth, often adds to user frustration. Another important aspect that was evaluated was the user satisfaction level about the available features of the application itself.

In order to enhance the entire user experience, from the application features up to its presentation to the user, systematic evaluations using multiple testbeds through TEFIS were proposed in the experiments. The use of TEFIS for conducting the experiments was highly desirable as it enabled cross-testbed interaction, test management, result monitoring and analysis. The first objective of the "Smart Ski Resort" experiment was to raise users' satisfaction level by improving the next version of the mobile ski application and its features. The TEFIS portal allowed the Megève's tourism board, and the experimenters, to define the profile

of the testers and to request the matching users to participate in the online surveys and trials with the help of TEFIS's "Botnia Living Lab" expertise. The online and on site surveys filled out by the users, and the trials carried out to improve the mobile application and user satisfaction resulted in more than 3000 participating users answering questions designed according to the Botnia Living Lab methodologies.

To evaluate the requirements for improving the underlying infrastructure supporting the mobile application, the experimenters measured certain network parameters using an instrumented version of the mobile application used by skiers in real settings, connected to two testbeds through TEFIS, namely, PlanetLAB and SQS-IMS. PlanetLAB was chosen to monitor the delivery of contents to the mobile application through large scale real networks, whereas SQS-IMS was used to investigate the potential benefits of having the application running over IMS (IP Multimedia Subsystem). The mobile ski application was configured to collect real usage data in order to investigate possible improvements in the performance of its multimedia features through the use of collaborative Wi-Fi sharing on the ski slopes, load balancing on PlanetLab nodes, or even assuming a 4G network with IMS. Collaborative Wi-Fi sharing between skiers was successfully emulated and then carried out on the ski slopes, up to a 40 m distance between two skiers where, one stationary skier was watching a video streamed from the smartphone of the other skier whilst skiing. The technical information obtained by this exercise was analyzed to better understand the users network usage and how it can be optimized through modeling and simulations of the use of parallel shared wireless networks to overcome the roaming cost, bottlenecks and disconnections.

The TEFIS platform served as a one-stop shop for the entire experimentation process as it was successfully used to carry out a multifaceted-testbed experiments, including Botnia Living Lab, SQS-IMS and PlanetLab.



Fig. 10. Megève Smart Ski mobile application.

6.3. The QUEENS experiment

QUEENS is the short name of “Dynamic Quality User Experience ENabling Mobile Multimedia Services”. This experiment aimed at establishing, assessing, and prototyping a novel framework for extending Quality of Service (QoS) to Quality of Experience (QoE) in mobile wireless networks, placing emphasis on mobile on-demand multimedia applications. Instead of viewing QoE as an offline and a priori mapping between the users’ subjective perspectives of their service quality, and specific networking metrics, QUEENS treats QoE provisioning as a dynamic process that enables users to express their preference with respect to the instantaneous experience of their service performance at: (a) multimedia content servers; and (b) wireless access network’s Radio Resource Management (RRM) systems. To facilitate this goal, Network Utility Maximization (NUM) is adopted as the underlying vehicle towards efficiently correlating QoE and user-application interactions with the QoS-aware RRM process, through the dynamic adaptation of users service-aware utility functions.

In a nutshell, QUEENS defined, set, and executed a complete experimental process through the use of TEFIS facilities that essentially covered the entire life-cycle of the multifaceted experiment. With multiple heterogeneous testbeds offered by the TEFIS platform, the QUEENS experiment was able to exploit and combine a human-centered testbed providing real end users (Botnia Living Lab), a distributed network emulated environment (PlanetLab), and a 3GPP IMS emulator including a Quality Assurance (QA) team (through the SQS IMS testbed). The main objectives of this experiment were to refine the design and development of a novel prototype of a mobile IMS-enabled multimedia application, capable of capturing user and network related factors influencing the service quality obtained, and dynamically optimizing its performance, thereby providing a unique enhanced experience to the users. More specifically, the QUEENS experiment had four phases, each one utilizing a different, or a set of different TEFIS testbeds—which allowed experimenters to test and

improve a prototype of an innovative dynamic QoE provisioning framework for mobile applications. The distinct features and requirements of each phase, as well as the reasons that necessitate the adoption and utilization of each testbed are presented next.

Phase 1 (QoE Framework Establishment): Phase 1 focused on gathering the overall specifications and requirements, as well as obtaining real users perspectives in a quantitative and pragmatic manner through experimentation, towards the successful prototyping and deployment of the proposed dynamic QoE mechanism. The goal of this first phase was twofold:

- Devise a flexible and light-weight user feedback mechanism (UI) to efficiently collect the mobile users’ opinion, though without interfering with the content being served (e.g., gather the experience and/or expectation using on demand real-time videos).
- Correlate mobile end-users’ perceptions of the quality of their multimedia service experience and corresponding network and service performance characteristics, in a quantitative and pragmatic manner, via the definition of proper QoE-aware utility functions.

To achieve these goals, an Android-based mobile multimedia application was developed, capable of capturing the users’ behavior with respect to their subjective quality perception. The application allowed users to interact with the video currently watched and request different QoS patterns to increasing their QoE (see Fig. 11). Finally, additional questionnaires requesting users’ feedback and impressions on the service they experienced were also fulfilled. The engagement of real users in such a behavioral analysis is key in achieving the above goals. Exploiting the diverse real user sets provided by the Botnia Living Lab, allowed a realistic and pragmatic view of users’ requirements, expectations and interactions, enabling the establishment of a concrete correlation of QoS and QoE via the definition of proper QoE-aware utility functions. These latter were used as feedback to the radio resource

management processes of the access network so as to enable dynamic resource allocation.

The aforementioned prototype application was executed by numerous Botnia Living Lab users and the statistical analysis of the captured data was able to provide preliminary insights on the effect of the various environmental and networking parameters examined in the analysis on users' perception of quality, highlighting the trends and parameters that shape and influence users behavior and motives.

Phase 2 (QoE-aware Dynamic Radio Resource Allocation): The second phase aimed at devising and evaluating a mechanism for integrating the dynamically changed QoE-aware users' service utility functions derived from Phase 1, into the resource allocation processes for: (a) on demand multimedia application video servers so as to maximize the overall service performance (i.e., minimization of response time, maximization of the number of ongoing sessions, etc.) and (b) heterogeneous wireless networks to enhancing the overall system performance [52]. In realizing the above goals, regressive and stress tests in large scale environments offered by PlanetLab were essential. More specifically, thanks to its distributed nature and the large number of geographically separated nodes, PlanetLab facilitates the emulation of a complete end-to-end networking scenario (from the video server to the end mobile user), and this was essential for evaluating the efficacy and performance of the dynamic utility adaptation scheme. It is important highlighting that, the usage of Planetlab environment via the TEFIS platform eased the burden of accessing and deploying code on numerous nodes, enabling not only one key execution but also the aggregation of results at a single point, besides aspects such as versioning facilitations.

Phase 3 (Mobile QoE Application Prototype Validation): Phase 3 incorporated the previous acquired knowledge and devised mechanisms towards exploiting QoE as an added-value, in the form of an add-on feature in commonly used Web-based services, such as video and audio streaming. Such a tool was built in line with the industry's protocols and standards as well as real-users expectations

and needs, enabling the seamless integration with currently existing services and applications. The proposed mobile QoE-aware multimedia application, *TEFIS* [53], was refined and validated end-to-end via the SQS IMS testbed to assure its compliance with the reference protocols and standards, as well as its interoperability on a realistic environment. This allowed to assess the envisioned mechanisms for interconnection, cooperation, and seamless integration with existing architectures and systems (e.g., 3GPP/LTE). The SQS Quality Assurance (QA) team accessed directly via the TEFIS platform, was responsible for this purpose and for validating the operational correctness of the application prototype. TEFIS offered a single point of access tool facilitating the communication, interactions and versioning of the tests with the human QA team.

Phase 4 (Business Model Validation): The final phase of the QUEENS experiment focused on the related business aspects of the proposed framework for all the involved actors. This was achieved via exploiting the IMS testbed operated by SQS, along with the Botnia Living Lab for a pragmatic evaluation of the proposed QoE-aware mechanism, in terms of: (i) efficacy in optimizing end-users' QoE; (ii) correlation of the proposed mechanisms with realistic pricing schemes; (iii) validation of the expected socio-economic impact; and finally, (iv) an analysis of expected benefits for operators, content providers and end-users.

In summary, the experiments through TEFIS drove a number of design optimizations and improvements on the operational logic of the final *TEFIS* application developed in Phases 3 and 4, thereby enabling both automatically personalized and manual adaptation of the video quality for users' QoE optimization.

6.4. The TEFPOL experiment

TEFPOL is the short name of "AugmenTed rEality collaborative workspace using Future Internet videoconferencing Platform fOR remote education and Learning". This experiment aimed at integrating, deploying and testing of an innovative videoconferencing system, and an advanced visualization and real-time technology on the TEFIS



Fig. 11. Graphical user interface for QoE provisioning within the mobile multimedia application.

testbeds. TEFPOL extends the capabilities provided by current videoconferencing and remote visualization solutions, offering an integrated large-scale platform for online education and training purposes to be used as part of Future Internet Services. The collaborative platform allows simultaneous interactions of high-resolution videoconferencing users with remote 3D objects and augmented reality scenes in real-time.

The videoconferencing solution used in the experiment was HDVIPER [54], an open and scalable high-definition videoconferencing platform developed in a Celtic initiative, with substantial participation of the Poznań Supercomputing and Networking Center (PSNC). Since then, it has been used and actively developed in a few European and national (Polish) projects; for example, in HIPERMED [55], and in “The Future Internet Engineering” project [56]. As for the visualization solution, a distributed Web-based platform called Vitral [57] has been used and extended. It was designed to utilize multi-GPU server installations for highly efficient parallel rendering under well-known data exchange formats. Vitral can be used in collaborative environments where many users can interact, share, or modify 3D models over the Internet simultaneously. Moreover, the 3D models and scenes can be accessed from different clients, such as web browsers or dedicated applications for tablets or even mobile phones. Vitral also offers many natural user interfaces based on sensors embedded within modern devices, such as accelerometers, magnetic field sensors, gravity sensors, gyroscopes, etc.

The motivation behind the experiment was to test and validate the overall performance, scalability and usability of the integrated platform on an European scale. The main questions that were addressed were the following:

- What is the maximum delay that end users accept during a videoconferencing session enhanced with augmented reality? Is a user interface based on movement tracking suitable and sufficiently responsive for such an application? Thanks to the involvement of the Botnia Living Lab, it was possible to carry out the “Biology in English” remote lesson of the future at a Swedish school (see Fig. 12). This gave an opportunity to evaluate not only measurable delays but also correlate subjective

measures of students’ experiences (Quality of Experience) with the service and delays offered by the platform.

- How many end users can use the application and its 3D models simultaneously, while maintaining High Definition (HD) quality streams? This question was answered thanks to the support of the SQS IMS testbed, which provided HDVIPER with signaling, authorization and presence management. The Botnia Living Lab testbed, which supplied the end users, also played a very important role here.
- Is it possible to improve remote rendering operations using multiple GPUs to achieve frame rates greater than 24 FPS? Answering this question was possible thanks to the PACAGrid testbed, which provided the resources for executing Vitral. In addition, the placement of the testbed was on the other edge of Europe—considering the location of the students provided by the Botnia Living Lab—which allowed the experimenters to test the platform in a geographically distributed environment. It is worth mentioning that, another TEFIS testbed, namely ETICS, helped improving the quality of the software, so ETICS also assisted the experimenters to reach the desired FPS bound.

An overview of the TEFPOL architecture, as well as relation between TEFIS’s testbeds and the TEFPOL sub-components is illustrated in Fig. 13. As mentioned earlier, multiple heterogeneous testbeds were utilized to investigate the hypotheses at hand. The decision to use TEFIS was to keep the focus on the objectives of the experiment, and thus avoid the time-consuming preparation and integration of various testbed systems. For example, the task of building all of the software components is carried out automatically in the ETICS testbed. Moreover, these components do not have to be manually transferred and installed on the other testbed – PACAGrid – because all these operations are provided by the TEFIS infrastructure. The data management and transfer mechanisms liberates the experimenter from tedious tasks, while assuring the software quality, e.g., by providing build and test reports. The TEFIS platform can be considered as a ready to use (cloud-like) solution from the experimenter’s perspective.



Fig. 12. The teacher’s video feed with a 3D model of a human heart superimposed.

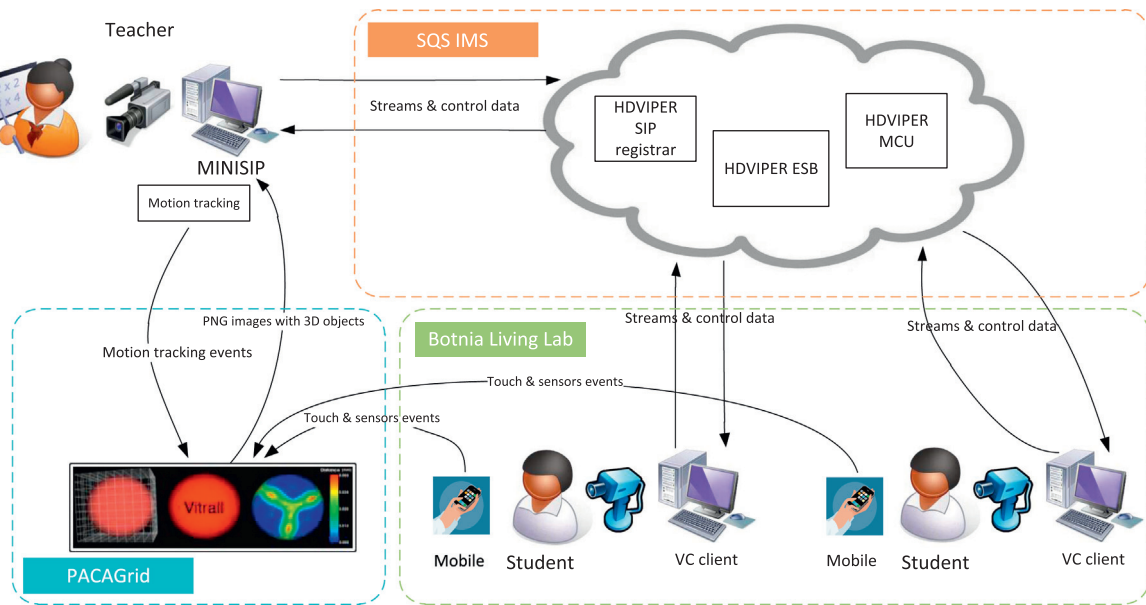


Fig. 13. Relation between Botnia Living Lab, PACAGrid, SQS IMS and the TEFPOL sub-components.

6.5. Summary of advantages for running experiments through TEFIS

Experimenting in a real scenario is a very complex endeavor. This is mostly due to the huge amount of external factors that might bias the obtained results. This can be even worse when dealing with multiple heterogeneous testbeds. TEFIS eases the burden of such complex management problem by providing built-in functionalities and automated processes. In particular, this is achieved by abstracting all the details of each particular testbed under a well-designed user interface. Instead of manually selecting the resources, deploying the experiment application, monitoring the status of the resources and running the tests, the experimenter just sets up the experiment by reclaiming a set of resources, and defining the experiment, while TEFIS deploys the application to the resources, initializes all the required services and runs the experiment as instructed by the experimenter. Furthermore, TEFIS eases the re-run of the experiment multiple times and all these actions are performed without the need of any interaction with the actual testbed.

For the testbeds involving human intervention, e.g., SQS-IMS and Botnia Living Lab, the TEFIS portal automates the experimentation process, as much as allowed by the testbed. In the case of Botnia Living Lab, with TEFIS, it is easy to follow the life-cycle of a Botnia Living Lab expert request because everything is logged in one place: time of initial request, automatic email sent to the expert, time of reply of the expert, automatic attachment of the subject to be reviewed by the expert, etc.

TEFIS also allows contacting all the testbed providers under one umbrella which enables efficient interaction and coordination among the testbeds involved. As an example, if an experimenter with a multi-testbed experiment, is to approach two different testbeds individually,

then it is less likely that they would agree on a common infrastructure (operating system, required libraries version compatibility, etc.) quickly and efficiently. Furthermore, it is easier to negotiate some non-standard changes in the testbeds because the testbeds under TEFIS, form a circle-of-trust. For example, they are less reluctant of modifying their access and security policies in order to perform the experiment through TEFIS.

As a summary of benefits of using TEFIS, we show in Table 5 a comparison between the steps needed to run an experiment using generic legacy testbed features, compared to the cost of running the same experiments using TEFIS in terms of required time. In general, by using TEFIS and the available knowledge base including existing experiments of previous experimenters and researchers, enables novel experimenters, with zero knowledge of the testbeds, to perform experiments faster, and more efficiently and reliably. That is, TEFIS reduces the learning curve for newcomers allowing them to focus on their experiments rather than on their corresponding execution.

7. Challenges and conclusion

In this paper, we have presented TEFIS, a new one-stop platform providing dynamic and heterogeneous experimental facilities for the testing of Future Internet services. TEFIS enables multifaceted experiments on different test facilities, providing complete support for the entire life-cycle of an experiment, including cross-testbed interactions, experiment and data management, resource monitoring, and result analysis. The diverse and distinct use cases presented in this work affirms TEFIS's commitment and its ability to support a wide range of experiments. We have also shown that, through a well-structured and modular design, TEFIS facilitates the integration of new testbeds

Table 5

Productivity comparison during the testing process between legacy testbeds and the utilization of TEFIS.

Phase	Without TEFIS	Duration	With TEFIS	Duration
Learning	– Get familiar with the testbed: learn each present glitch/feature	days/weeks	– Use the built-in features in TEFIS, implemented by testbed experts	hours/days
Preparation	– Manually install all the requirements of the application in an unknown environment – Communicate and coordinate with potential experts among different testbeds as per requirements	days/weeks	– Configure requirements in the TASK Properties in the TEFIS portal, accordingly (c.f., Fig. 7 (a))	hours
Experiment deployment	– Code customized scripts to compile and solve issues in the deployment, e.g., different OS versions, architectures, and so on – Deploy each particular compilation to the corresponding resources	hours/days	– Just instruct TEFIS portal to perform the deployment	min
Experiment execution	– Manually connect/communicate to all the involved resources and run the experiments	min/hours	– Run the experiment from the portal	seconds
Validation	– Manually check the status of each resource involved in the experiment	hours	– Let the portal seamlessly monitor the status of the experiment execution	Not applicable
Gathering results	– Manually download and gather all the result files generated by each testbed in the experiment and process them	min	– Get the results from the Experiment Data Manager and process them (c.f., Fig. 7 (b))	seconds

into its platform, which adds value both from the point of view of its openness and of its evolution.

Based on these strengths, TEFIS is well positioned towards mapping out a robust and flexible sustainability framework for federated testbed platforms, which certainly requires further efforts in the right direction. One step in that direction is the establishment of the TEFIS Partner Network, which aims to envision further exploitation and extension of TEFIS. To conclude, we list the main challenges on the way to developing a long-term and sustainable testbed marketplace, and discuss TEFIS's contributions and possible strategies to approach them.

The gap between testbed providers' offerings and experimenters' needs and expectations: So far, most of the testbeds have been established from a technology push mode, and user involvement in early stages of the development has not been the case. With early user involvement, the likelihood to succeed in creating useful testbeds will increase and user's needs may even become a source for innovation. Within FIRE in Europe, the open calls pose one successful instrument to involve users in the development of testing facilities. The TEFIS open call experiments, detailed in Section 6, acted as a catalyst for TEFIS improvements by pointing out problems, recommending solutions and validating the platform. A challenge in the long term is to keep the testbed movement continuously user-driven.

The complexity of testbed access from the experimenters view: One of the achieved goals through TEFIS, is the provision of easy access for experimenters to different testbed resources and to find mechanisms to overcome the problem with heterogeneous access for users to different testbeds. As the testbed domain still is rather immature, standards for testbed access will need to be implemented to achieve broader usage of testing facilities and to lower the investments of experimenters in testbed usage.

Experimenters' investments in testbed usage: Radical innovation in the fuzzy front end such as test services provided through the outlined experimental facilities in the

Internet market faces two key challenges. First, an uncertain future due to the novel market, and thus also uncertainties in return on investments causing high technological, market and financial risks. Second, resource intensive investments, in terms of technological, financial and human capital, for enabling the provision of innovative services [58,59]. Federated organizations based on collaborating networks such as TEFIS enable exchange of multifaceted key resources, and limit the risks to develop and spread innovative practices, thus overcoming issues such as the individual key partners' limitation of resources and limitation in access to customers in the novel market [60].

Seamless integration of different testbeds: The seamless integration of a heterogeneous ecosystem of different testbeds and (cloud) services is the next step towards an effective commissioning of the connector framework. The framework should work for any type of resource that could be made available through the TEFIS portal, hence allowing growth of the TEFIS ecosystem and paving the way for the future sustainability of the project. The presence of heterogeneous services in a—possibly cloud—infrastructure is one of the potentials for the connector framework: all TEFIS APIs are designed to allow the coexistence of different testbed APIs, ensuring the possibility for expanding the testbed ecosystem.

Test data management and test data exchange: One of the major values of TEFIS is that the platform manages the whole experimental life-cycle on behalf of the experimenter. This involves the management of data generated by the user, directly required for the successful execution of a testrun or subset of a testrun, and generated as a result of the execution of the testrun. All of these data sources may differ in terms of format and what they contain. The TEFIS platform and especially the Data Services already provide a metadata schema based on a number of appropriate standards to describe experimental data. As platform usage increases, it will become increasingly necessary to impose and extend this schema to allow for the seamless transfer of data. At the same time, the integration of monitoring

and other experimental data needs to be accomplished to provide a complete picture for experimenters.

Test data sharing issues and the openness between experiments: It is a significant achievement for TEFIS to be able to enable and support a diverse community of experimenters. One aspect of this is allowing experimenters to search and inquire about the work of others. At the very least, the TEFIS Data Services provide access controls for the experimental data stored there. Based on such controls, TEFIS experimenters are already able to search the work of others, and request contact from the owning experimenter. However, as we move forward, it is important to ensure that data are protected in terms of derivative works (copyright), of identity (privacy) and ownership. The curation and provenance markers within the experimental data schema will need to be managed and controlled to allow for appropriate ownership rights to be monitored and controlled.

Federation vs. competition: We recognize the need of testbed providers to position themselves in the market and their competition with one another as challenges for their federation in a common marketplace. We aim to convince testbed providers to collaborate by emphasizing the fact that the development of markets also calls for the development of marketplaces where exchange can take place (as explained in [61]). We need to show how an online marketplace based on federating partners with multiple service offerings can provide easy access for customers all around the world. However, also the development of such a common marketplace is not without challenges. There is first of all a clear need to design and organize business models able to combine the constraints of various testbed infrastructures with customers' needs and requirements in order to enable a financially viable structure [62]. There is secondly, a clear need to assure for trust among all collaborating partners and a need to assure for trustworthiness in front of current and potential customers [63]. Further, to be sustainable, the realization of a Future Internet test service market place necessitates a critical mass on both sides of the market: testbed providers as well as customers [64].

Financial sustainability: Dynamic business models and an innovative lead strategy in the development of novel Internet service offerings may be clearly linked to national government agencies providing targeted funding [65]. Governmental funding serves as a complement to traditional capital markets and is the base to tackle imperfections of capital markets, which can cause serious financial gaps (also known as financial death valleys) [66]. Earlier studies show that innovative industries with high capital intensity develop faster in nations with well-developed capital markets [67]. Financial theory suggests rationales for governments offering funding to high-technology innovations [68], particularly in cases where innovative organizations are a unique source of new ideas and growth that creates and captures value for other industries and firms. In such cases supporting innovation is suggested as appropriate. Keeping in mind the financial liabilities, there is a clear and urgent need for a comprehensive sustainability model for experimental facilities to assure their future availability. We can conclude that a

viable financial base is critical for the innovative high-technology Internet market to develop into a sustainable market. To this end, governmental funding serves a role as bridging the imperfections of the capital market [69].

Acknowledgements

The authors are grateful for the support received from the rest of the researchers involved in the FP7 Project TEFIS, Contract No. 258142. UPC authors also acknowledge the support received by the Spanish Ministry of Science and Innovation under Contracts TEC2009-07041 and TEC2012-34682, and by the Catalan Research Council (CIRIT) under Contract 2009 SGR1508.

References

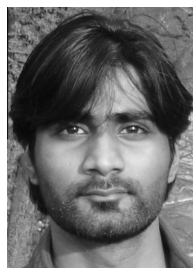
- [1] TEFIS, 2013 <<http://www.tefisproject.eu>>.
- [2] PacaGrid, 2013 <http://www.tefisproject.eu/media/upload/11-2445-blad-paca-grid-korr_110204_2.pdf>.
- [3] ETICS, 2013 <http://www.tefisproject.eu/media/upload/11-2445-blad-etics-korr_110204_2.pdf>.
- [4] SQS-IMS, 2013 <http://www.tefisproject.eu/media/upload/11-2445-sqs-korr_110204_3.pdf>.
- [5] M. Wuthnow, M. Stafford, J. Shih, IMS: A New Model for Blending Applications, Taylor and Francis, 2009. ISBN 1-4200-9285-5.
- [6] BOTNIA, 2013 <http://www.tefisproject.eu/media/upload/11-2445-blad-botnia-korr_1102041.pdf>.
- [7] KYATERA, 2013 <http://www.tefisproject.eu/media/upload/11-2445-blad-kyatera-korr_110204_22.pdf>.
- [8] PlanetLab, 2013 <<http://www.tefisproject.eu/media/upload/11-2445-planetlab-korr2.pdf>>.
- [9] Future Internet Research and Experimentation (FIRE), 2013 <<http://www.ict-fire.eu/home/fire-projects.html>>.
- [10] NOVI, 2013 <<http://www.fp7-novi.eu/>>.
- [11] BonFIRE, 2013 <<http://www.bonfire-project.eu/>>.
- [12] LAWA, 2013 <<http://www.lawa-project.eu/>>.
- [13] OpenLab, 2013 <<http://www.ict-openlab.eu>>.
- [14] OFELIA, 2013 <<http://www.fp7-ofelia.eu/>>.
- [15] PanLab, 2013 <<http://www.panlab.net>>.
- [16] OneLab, 2013 <<http://www.onelab.eu>>.
- [17] OneLab2, 2013 <<http://www.onelab.eu/index.php/projects/past-projects/onelab2.html>>.
- [18] FEDERICA, 2013 <<http://www.fp7-federica.eu/>>.
- [19] PlanetLab Europe, 2013 <<http://www.planet-lab.eu/>>.
- [20] ETOMIC (European Traffic Observatory Measurement Infrastructure), 2013 <<http://etomic.org>>.
- [21] NITOS Wireless Testbed - Network Implementation Testbed Laboratory, 2013 <<http://nitlab.inf.uth.gr/nitlab/index.php/testbed>>.
- [22] OpenFlow, 2013 <<http://www.openflow.org/>>.
- [23] The TEAGLE Portal, 2013 <<http://www.fire-teagle.org/>>.
- [24] Distributed topology measurement infrastructure (DIMES), 2013 <<http://www.netdimes.org/new/>>.
- [25] EXPERIMEDIA, 2013 <<http://www.experimedia.eu/home>>.
- [26] Schladming Ski Resort, 2013 <<http://www.ski-weltcup-schladming.at/>>.
- [27] Multi-Sport High Performance Center of Catalonia (CAR), 2013 <<http://www.car.edu/>>.
- [28] Foundation for the Hellenic World, 2013 <<http://www.ime.gr/fhw/>>.
- [29] 3D Innovation Living Lab, 2013 <<http://www.lafabriquedefutur.org/>>.
- [30] FI-WARE, 2013 <<http://www.fi-ware.eu>>.
- [31] Global Environment for Network Innovations (GENI), 2013 <<http://www.geni.net/>>.
- [32] T. InstaGENI Initiative: An Architecture for Distributed Systems, Advanced Programmable Networks, 2013 <<http://groups.geni.net/geni/wiki/instageni>>.
- [33] K. Testbed Deployment, Federated Meta Operations Experiment over GENI, KREONET, 2013 <<http://groups.geni.net/geni/wiki/k-geni>>.
- [34] The Japanese National Institute of Information and Communications Technology (NICT), 2013 <<http://www.nict.go.jp/en/>>.
- [35] Japanese New Generation Network Testbed (JGN-X), 2013 <<http://www.jgn.nict.go.jp/english/>>.
- [36] European Network of Living Labs, 2013 <<http://www.openliving-labs.eu/>>.

- [37] FED4FIRE, 2013 <<http://www.fed4fire.eu>>.
- [38] GLIF/Starlight, 2013 <<http://www.startup.net/starlight/>>.
- [39] Future Internet Research and Experimentation: The G-Lab Approach, 2013 <<http://www.german-lab.de/>>.
- [40] The GpENI Testbed: Network Infrastructure, Implementation Experience, and Experimentation, 2013 <<http://www.geni.net/?p=1900>>.
- [41] NorNet (Core) – A Multi-Homed Research Testbed, 2013 <<https://www.nntb.no/nornet-core/>>.
- [42] Stanford SDN/OpenFlow Network Testbed, 2013 <<http://archive.openflow.org/wp/stanford-deployment/>>.
- [43] TEFIS Toolkit for Easy Testbed Integration and Management, 2013 <<http://www.tefisproject.eu/results/the-tefis-toolkit-for-easy-testbed-integration-and-management>>.
- [44] TEFIS, TEFIS Connector Framework Documentation, 2009 <<http://grids29.res.eng.it/tcf/doc//index.html>>.
- [45] D. Caromel, M. Leyton, Proactive parallel suite: from active objects-skeletons-components to environment and deployment, in: EuroPar 2008 Workshops – Parallel Processing, LNCS, vol. 5415, Springer, Berlin/Heidelberg, 2009, pp. 423–437.
- [46] Integrated Rule-Oriented Data System (iRODS), 2013 <<http://www.irods.org>>.
- [47] S. Mäkeläinen, T. Alakoski, Fixed-mobile hybrid mashups: applying the rest principles to mobile-specific resources, in: Proceedings of the 2008 International Workshops on Web Information Systems Engineering, WISE '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 172–182.
- [48] Y. Elkhatib, G. S. Blair, B. Surajbali, Experiences of using a hybrid cloud to construct an environmental virtual observatory, in: Proceedings of the 3rd International Workshop on Cloud Data and Platforms, CloudDP '13, ACM, New York, NY, USA, 2013, pp. 13–18.
- [49] R. Spalazzese, A Theory of Mediating Connectors to Achieve Interoperability, PhD Thesis, 2011.
- [50] OPENER, 2013 <<http://www.craax.upc.edu/opener>>.
- [51] Lumiplan, 2013 <<http://lumiplan.com/>>.
- [52] G. Aristomenopoulos, T. Kastrinogiannis, V. Kaldanis, G. Karantonis, S. Papavassiliou, A novel framework for dynamic utility-based QoE provisioning in wireless networks, in: Proceedings of IEEE Globecom, Miami, Florida, USA, December 2010.
- [53] G. Aristomenopoulos, S. Papavassiliou, G. Katsaros, P. Vlahopoulos, User-centric mobile multimedia service delivery: from theory to experimentation to prototyping, in: Proceedings of IEEE INFOCOM 2013 (Demo Session), Turin, Italy, April 2013.
- [54] J. Alcober, G. Cabrera, X. Calvo, E. Eliasson, K. Groth, P. Pawalowski, High Definition Videoconferencing: The Future of Collaboration in Healthcare and Education, in: Proceedings of eChallenges e-2009, ISBN: 978-1-905824-13-7.
- [55] P. Alvarez, J. Benseny, X. Calvo, E. Eliasson, K. Groth, C. Mazurek, P. Pawalowski, W. Pieklik, M. Stroinski, S. Tufan, An open eHealth platform. Solutions for medical services of the future, in: Proceedings of eChallenges e-2011, ISBN: 978-1-905824-27-4.
- [56] W.B. et al., The future internet engineering project in poland: goals and achievements, in: Proceedings of Future Internet Poland Conference, Poznan, Poland, 2011.
- [57] P. Sniegowski, M. Blazewicz, T. Kuczynski, K. Kurowski, B. Ludwiczak, Vitral: web-based distributed visualization system for creation of collaborative working environments, in: Proceedings of PPAM 2011, Part I, LNCS 7203, 2011, pp. 337–346.
- [58] H. Chesbrough, *Open Business Models: How to Thrive in the New Innovation Landscape*, Harvard Business School Publishing, Cambridge, MA, 2006.
- [59] T.C. Powell, A.D. Micallef, Information technology as competitive advantage: the role of human, business and technology resources, tug of war in innovation competitive service development, *Strategic Manage. J.* 18 (5) (1997) 375–405.
- [60] C. Brush, C. Greene, P. Hart, From initial idea to unique advantage: the entrepreneurial challenge of constructing a resource base, *Acad. Manage. Exec.* 15 (1) (2001) 64–78.
- [61] P. Bruun, M. Jensen, J. Skovgaard, e-Marketplaces: crafting a winning strategy, *Eur. Manage. J.* 20 (3) (2002) 286.
- [62] R. Amit, C. Zott, Value creation in e-business, *Strategic Manage. J.* (22) (2001) 6–7, 493–520.
- [63] T.K. Das, B. Teng, Between trust and control: developing confidence in partner cooperation in alliances, *Academy of Management Review* 3 (July) (1998) 491–512.
- [64] A. Afuah, C. Tucci, *Internet Business Models and Strategies*, McGraw Hill, Boston, 2003.
- [65] J. Timmons, *New Venture Creation; Entrepreneurship for the 21st Century*, Irwin, Sydney, 1994.

- [66] P.E. Auerswald, Valleys of death and Darwinian seas: financing the invention to innovation transition in the United States, *J. Technol. Transfer* 28 (3) (2003) 227. sidor.
- [67] R. Levine, Financial development and economic growth: views and agenda, *J. Econ. Lit.* 35 (1997) 688–726.
- [68] J. Lerner, When bureaucrats meet entrepreneurs: the design of effective Public Venture Capital programmes, *Econ. J.* 112 (2002) F73F84.
- [69] T.C. Lawton, Missing the target: assessing the role of government in bridging the European equity gap and enhancing economic growth, *Venture Cap.* 4 (1 sidor.) (2002) 7.



Marcelo Yannuzzi received a degree in Electrical Engineering from the University of the Republic, Uruguay, and the MSc. and Ph.D. degrees in Computer Science from the Department of Computer Architecture (DAC), Technical University of Catalonia (UPC), Spain. He is the head of the Networking and Information Technology Lab (NetIT Lab) at UPC, as well as the head of the Advanced Network Architectures (ANA) research group at UPC. He is involved in several research initiatives and projects in close interaction with European and US companies and research centers. His research interests lie on Software Defined Networks (SDNs), outsourced computation and control of network functions, security, network management, smart orchestrations, and mobility.



Shuaib is a research associate as well as a PhD candidate at Technical University of Catalonia (UPC), Spain. He received his B.Sc in Computer Engineering from King Fahd University of Petroleum & Minerals (KFUPM), Saudi Arabia and M.Sc in Communication Systems Engineering from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.



Annika Sällström (born 1966) is currently employed as a general business manager at Centre for Distance-Spanning Technology at Luleå University of Technology in Sweden. Her formal education includes a bachelor of Education and studies in Computer Science and System Analysis. Of specific competence of her is more than 10 years' experience in user-driven innovation processes as responsible of the Botnia Living Lab.



Brian Pickering has spent many years in ICT Research and Development, working on large automated systems for telco and enterprise customers. He joined the University of Southampton IT Innovation Centre three years ago and has worked on a number of different projects looking at the Future Internet and its socio-economic implications.



Rene Serral-Gracia received his degree in computer science (2003) and a Ph.D. (2009) from the Technical University of Catalunya (UPC). He is the R&D head of the Networking and Information Technology Lab (NetIT Lab) at UPC, where he is leading different research initiatives, including projects under the European FP7 Research Framework as well as with industry. He is also an Associate Professor at the Department of Computer Architecture at UPC. His research interests are focused on Software Defined Networks (SDNs), over-

lay networks, network security, routing optimization, and QoE assessment of multimedia traffic.



Farid Benbadis is a research engineer at Thales Communications and Security, in Colombes, France. He received, in 2007, his Ph. D. degree from the University Pierre et Marie Curie, Paris, France. He spent one year as a postdoc fellow at Orange Labs, France Telecom R&D, France. His research interests include network protocol in wired and wireless environments.



Anny Martinez received her Bachelor's degree in electronic engineering from Simon Bolivar University, Caracas, Venezuela, in 2008. She is currently pursuing a Ph.D. at Technical University of Catalonia (UPC), Spain. Her research interests include future internet architectures, multi-layer networking and ontology-based knowledge management.



Jérémie Leguay is research manager at Thales Communications, in Colombes, France. His current research interests are IP protocols and distributed Internet applications. He received a Ph.D. from the Computer Science laboratory (LIP6) of Pierre & Marie Curie University and Thales Communications where he conducted research in ad hoc (MANET) and Disruption Tolerant networking (DTN).



Willis Chen is Research Engineer at the IT Innovation Centre, University of Southampton. He is experienced in ICT research, in particular such large distributed computing systems as Grid and Cloud. His current focuses and projects include Cloud service federation (the FP7 BonFIRE project), semantic and streaming reasoning (the FP7 SERSCIS), Cloud monitoring (the FP7 TEFIS), Future Media Internet (FP7 3DLive project), Trust and Trustworthiness management in the Future Internet (FP7 OPTET project).



Elvio Borrelli is working as a software engineer at INRIA, France. He received his degrees in Software Engineering and IT Engineering from Università degli Studi del Sannio-Benevento, Italy.



Steve Taylor has worked on many research projects, and his expertise includes system design and architecture, business modeling, trust and security, and service-based computing associated with economics. In the past he has worked many Grid and SOA projects including GRIA, NextGRID and BREIN. Steve has a PhD in artificial intelligence.



Itziar Ormaetxea is working as a TestLab manager at Software Quality Systems S.A., Bilbao, Spain, where she leads the quality assurance and testing activities. She received her Bachelors degree in Mathematics from the Universidad del País Vasco, Spain.



Konrad Campowsky works as a senior researcher at the Competence Center NGNI of Fraunhofer FOKUS in Berlin. After having contributed to several EU funded projects in the field of cloud federation and testbed federation, Konrad joined the M2M group at Fraunhofer FOKUS in 2012 where he currently leads the technical development of the OpenMTC M2M framework.



Tomasz Kuczynski holds MSc Eng degree in Computer Science (Software Engineering) from Czestochowa University of Technology. Since 2004 employed at PSNC. Currently, involved in the E-podreczniki national project. Main areas of expertise are web technologies, visualization and security. He is credited with discovery of numerous security vulnerabilities in portal software (all verified and published by the US-CERT).



G. Giammatteo Graduated (in 2009) in Computer Engineering at the Università degli studi di Roma Tre on Secure Storage in distributed environments. He is currently working in the software engineering field with expertise in software integration, testing and quality assurance. He is involved in three different running FP7 projects: Artist, iMarine and Ocean.



Sergiusz Zielinski has been a member of the Education and eLearning Systems Team at the Network Services Department of Poznan Supercomputing and Networking Center since 2006. He has received his Master's Degree in Computer Science at the Poznan University of Technology in 2007. Currently he is participating in a series of national and international projects such as HIPERMED, PLATON and EXPERIMEDIA. His interests focus on eLearning, eHealth, videoconferencing and multimedia technologies.



Georgios Aristomenopoulos received his Ph.D in Electrical and Computer Engineering from National Technical University of Athens (NTUA), Athens, Greece and his B.S in Computer Engineering and Informatics from University of Patras, Greece, in 2013 and 2007 respectively. He is currently a senior researcher in Network Management and Optimal Design Laboratory (NETMODE) of NTUA.



Jean-Marc Seigneur has obtained his PhD at Trinity College Dublin on computational trust, online reputation and identity management in 2005. He is now an assistant professor and has been consulted regarding his research, especially e-reputation management, by many big companies such as Amazon, Siemens, Thales, Verisign and Philips.



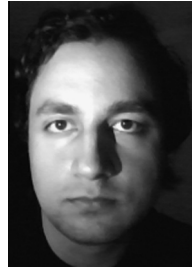
Symeon Papavassiliou is an associate professor in the School of Electrical and Computer Engineering, at the National Technical University of Athens (NTUA), Greece. He received his Ph.D. degree in electrical engineering from Polytechnic University, Brooklyn, New York, in 1996. His main research interests lie in the area of computer and communication networks with emphasis on wireless networks, resource allocation and optimization, and performance analysis of stochastic systems.



Carlos Ballester Lafuente is a PhD candidate at the University of Geneva where he has been researching computational trust and cooperation incentives in wireless networks as part of different EU-funded FP7 research projects, i.e., ULOOP, MUSES and TEFIS.



Jeaneth Johansson is an Associate Professor at Lulea University of Technology, Lulea, Sweden. She received her PhD in Business administration from Lulea University of Technology. She is also the founder of EkonomiStrategerna Fourjo. Her professional interests include Business model, Financial accounting, Entrepreneurial finance, Financial decision making, and Innovation.



José Roberto is graduated in computer engineering at Escola Politécnica da Universidade de São Paulo Brazil, he was a researcher in Laboratório de Sistemas Abertos (LSA). He is also a researcher in SmartLAB, in which his research includes complex systems, and self-organization methods.



Xavi Masip-Bruin, Ph.D. in Telecommunications Engineering from the Technical University of Catalonia (2003) is serving from its creation in 2008 as Director of the Advanced Network Architectures Lab (CRAAX), where he has been involved in several research initiatives funded by both the public and private sectors. Xavi has a large track record of participation in FP6 and FP7 EU research projects and he has been engaged in initiatives in close collaboration with industry including Cisco Systems, IBM, etc. His current R&D activities

are in the areas of network programmability and adaptability, network management, protocol network architectures design, IoT, and smart cities and novel services.



Email Salageanu is working as an R&D Software Engineer at Wall Street Systems, France. Prior to this, he worked as a development engineer at ActiveEon, France, of which he was co-founder as well. He received his MSc. in Computer Engineering and Distributed Systems and Applications from Université de Nice-Sophia Antipolis, France.



Marcel Caria is a Ph.D. student at the Institute of Computer and Network Engineering at the TU Braunschweig. He received his diploma degree in computer science from TU Braunschweig in 2010. His research interests include network optimization, novel network control and management schemes and energy efficiency in carrier-grade networks.



Joseph Latanicki is currently working as a Security Architect in THERESIS which is a THALES innovation laboratory aimed to link academic research and Thales business line needs. He also served as project coordinator for the FP7 project TEFIS. He has been working with THALES Services SAS for last 25 years.