Deliverable Title	**D7.1.3 – General procedure to run experiments on the TEFIS platform**
Deliverable Lead:	FUSP
Related Work package:	WP 7
Author(s):	José Roberto (FUSP), Caio César (FUSP)
Dissemination level:	Public
Due submission date:	31/05/2011
Actual submission:	31/05/2011
Project Number	258142
Instrument:	IP
Start date of Project:	01/06/2010
Duration:	30 months
Project coordinator:	THALES

Abstract	This document cover the steps to any experimenter submit, run, test and collect the results of a generic experiment over the TEFIS platform.

Project funded by the European Commission under the 7th European Framework Programme for RTD - ICT theme of the Cooperation Programme.

License

Versioning and Contribution History

Version	Date	Modification reason	Modified by
0.1	25/04/2011	ToC	José Roberto(FUSP)
0.6	02/05/2011	Document Sections	José Roberto(FUSP), Caio César(FUSP)
0.8	24/05/2011	Portal Figures Addition	José Roberto (FUSP), Caio César(FUSP)
0.9	26/05/2011	Send to partners review	José Roberto (FUSP)
0.95	26/05/2011	Reviewer Comments	Brian Pickering (ITI)
0.96	31/05/2011	Reviewer Comments	Ainhoa Gracia (SQS)
1.0	31/05/2011	Final version	José Roberto (FUSP), Caio César (FUSP)

Table of Contents

INDEX OF FIGURES

INDEX OF TABLES

Executive Summary

The main objective of Deliverable D7.1.3 "General procedure to run experiments on the TEFIS platform" is to create a definition of the three main activities necessary to run any generic experiment: the Scenario Definition, the Design of the Experiment, and the Implementation and Execution of the Experiment.

Using the description of aims and intentions of a Scenario, any experiment can be specified in simple terms. This information can be translated into the following categories:

- Summary;
- Hypothesis definition;
- Experimental method and procedure;
- Actors involved;
- Variables;
- Metrics and Measurements;
- Technical Requirements;

In summary this with an example experiment involving an end-user trial of a multi-media social networking application:

Table 1: A generic experiment specified in simple terms and an example involving an end-user trial of a multimedia social networking application

Item	Description	An Example Experiment
Summary	Tell us what your experiment is about and what you are trying to achieve.	Testing the latency in sharing multimedia files for a social networking application running in a highly-distributed environment.
Hypothesis definition	What is the basic question you are trying to investigate? Tell us what you are trying to prove (or disprove!)	Will latencies be less than 500 msec to transfer multi-media files irrespective of the number of users? A secondary issue is whether a 20 to 100 application-server-to-data-server ratio would be sufficient to support the projected traffic.
Experimental method and procedure	Tell us what steps are involved in running your experiment.	The intention is to use a group of real users who will register with the service, take photos, annotate with audio, and then post to a central site. In the second stage, users will browse all content arbitrarily and download any files they wish to. They will be limited to 5 downloads in a given period.

Actors involved	Tell us who and how many people you will need as end users.	500 real users, with multi-media mobile phone devices, from the age group 18 – 25 years.
Variables	Tell us any values you are trying to test and any values you will be keeping constant.	The overall latency of (a) uploading files (b) downloading files (c) browsing files as a function of the number of users and the sizes of the files. We will only try to regulate the number of users and their usage. We will also investigate specific details about the technical infrastructure such as connection speed and bandwidth.
Metrics and measurements	Tell us how you want to quantify the data from the experiment.	We would like to measure the elapsed time in ms between request receipt at the server end until the whole file has been received and stored at the personal device end.
Technical requirements	Tell us about the environment and technical needs of your experiment.	We will need to have access to 500 end-users, connected to and using the service. We expect the service to be installed and running on at least 20 servers and a bank of 100 servers each with approximately 50 TB of storage, and with wireless access from the user to the server and *quasi*-realtime retrieval of data from the data servers to the application server. User devices should be of the *Apple®* family.

1 Introduction

This document describes how to run a generic experiment on the TEFIS Platform, aiming at general procedures based on experience gained from the experiments used in the early stages of platform development.

It describes a walkthrough to guide the experimenter in planning the experiment from scratch through to the collection of the results, via the design in the portal and the deployment of the application on the platform.

In this first section of this document the TEFIS platform will be introduced and the testbeds available initially for use by the experimenter and in the following sections the steps to run a successful experiment will be described in detail.

1.1 About the TEFIS platform

The TEFIS project is developing an open platform to access heterogeneous and complementary experiment facilities for communities of software/business developers to test, experiment and collaboratively create knowledge. It provides appropriate tools and methodologies to address the full development life-cycle of innovative services.

The main feature of the platform is the single point of access to a large number of varied testbed resources, so the experimenter does not need to go to different places for different parts of his experiment. Instead he has access to all experimental facilities in a single portal, where he can plan the experiment, run all necessary tests - even when a single experiment involves separate stages run on different testbeds - and collect the results.

1.2 Experiment Facilities

Heterogeneous and complementary facilities are available in TEFIS for: network performance experiments, software testing, and user-oriented living labs. The initial Experimental Facilities (testbeds) that the experimenter will able to use in the TEFIS platform are:

- **Planetlab:** global research network that supports the development of new network services, with a community of graduate students, researchers and industrial partners. One of its main purposes is to serve as a testbed for overlay networks in which researchers can run experiments on a variety of global services (file sharing, routing and multicasting overlays, scalable object location, etc) with the advantage of being able to run experiments under real-world conditions and on a large scale. This testbed is also designed to support long-running services that support a client base.

- **PACA Grid:** is a set of machines accessible via Graphical Interactive interfaces based on the ProActive Parallel Suite. This Cloud aggregates dedicated machines, both Linux and Windows, and spare desktop machines, dynamically added during the night and at week-ends. The facility mainly targets the scientific and industrial communities that need to speed-up scientific simulations, parallelize and distribute large scale applications, accelerate financial computations and so on.

- **ETICS**: e-Infrastructure for Testing, Integration and Configuration of Software is a collaboration project coordinated by CERN and funded in part by the European Commission. The goals of ETICS are to provide state-of-the-art open-source tools, procedures and resources to manage the software production lifecycle for research projects developing grids and other distributed software and to promote the establishment of standard quality assurance methods and procedures within the scientific research communities to improve the overall quality, stability and reliability of the grid-based research infrastructures. ETICS provides a number of services including a configuration service, a distributed multi-platform test bed, an online repository of packages, build and test reports and metrics, and an ISO-compliant quality assessment model called A-QCM (Automated Quality Certification Model).

- **SQS IMS:** this testbed is offered as an infrastructure to validate and test applications over IMS. The SQS testbed offers both the emulated IMS platform, and connection to real environments (IMS, SS7) to allow tasks to be run and tested; also end-to-end validation and testing services is provided, such as functional validation of the system under test, network performance monitoring, measurement of end to end network QoS and interoperability, for the Telecommunications Sector.

- **Botnia Living Lab:** is the first and largest open Living Lab for human-centric ICT development. The basic idea is to engage end-users, individuals and stakeholder organisations, along a targeted value chain, in the total process from need-finding and idea-generation, through concept-development and prototype/usability testing to pilot service validation of market and marketing principles.

- **KyaTera:** is a testbed that connects academic institutions, institutes of research and funding agencies in an environment for collaborative work, based on a Fiber-to-the-Lab network, dedicated to the study and development of science, technologies and Future Internet applications. The Kyatera network performance meter will be created to measure network quality, targeting the problem of guaranteeing QoS on multimedia data transmission.

1.3 ABOUT THE EXPERIMENT

The TEFIS platform aims to help identify experimental needs and formalize planned experiments, the notion of “high-level workflow” was presented to facilitate users running an experiment. The workflow is illustrated in the figure 1:

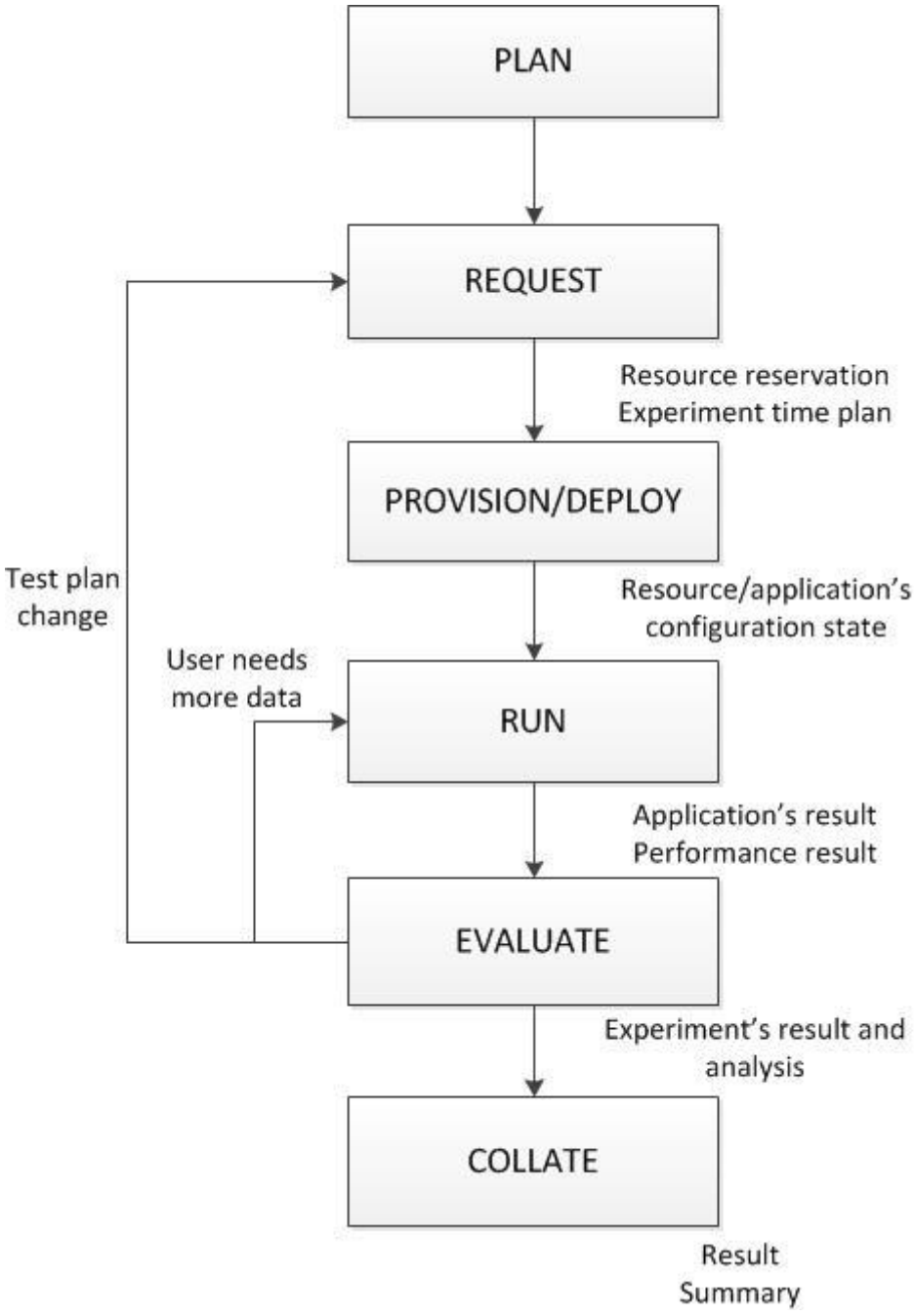

Figure 1: High Level Workflow

The TEFIS portal follows these stages giving the user access to the related functionalities supported by the back end and the testbed facilities. Those stages are as follow:

- **Plan:** is the first phase of the high level flow in which the user is responsible for defining the type of experiment, experimental restrictions, which testbeds are required and global information related to the experiment. In summary, this stage is used for:

  - Expression of needs
  - Identification of testbeds, KPIs, tools
  - Planning & methodology
  - High-level Workflow formulation

- **Request:** is the phase in which the testbeds chosen are contacted, their checked availability discovered, and if necessary, request reservation. In this phase, a time plan for the experiment is also produced. In summary, this stage is used for:

  - Access requests/grants
  - Resource booking

- **Provision / Deploy:** is the stage responsible for the configuration of the required resources and deployment of data and user applications Those stages are grouped here because from a user perspective, there is little intervention required, and so the transition from one stage to the next is largely transparent. In summary, this stage is used for:

  - Test specification (executable workflows)
  - Configuration
  - Deployment

- **Run:** is the realisation of the definition, planning and preparation described in the first three stages. The outputs are the results of the user application and performance results of execution (KPIs measured during execution). In summary, this stage is used for:

  - Execution
  - Testing (tools, Users)
  - Data collection

- **Evaluate:** is the phase where results collected from the run stage are analyzed and used to verify final experiment results. From this point, the user may choose to run another experiment or change configurations if necessary. In summary, this stage is used for:

  - Experimental results analysis
  - Experimental refinements

- **Collate:** is the stage in which final results and conclusions are produced for the experimenter. The results can also be published in a knowledge base, so future experimental cycles are enhanced. In summary, this stage is used for:

  - Access knowledge base
  - Community dissemination

2 General Architecture Overview

Before the experimenter starts the definition and the design of the experiment, it is important to know about the TEFIS architecture. A more detailed explanation can be found in the document D2.1.1. Here the experimenter can find some of the basic principles of the platform architecture.

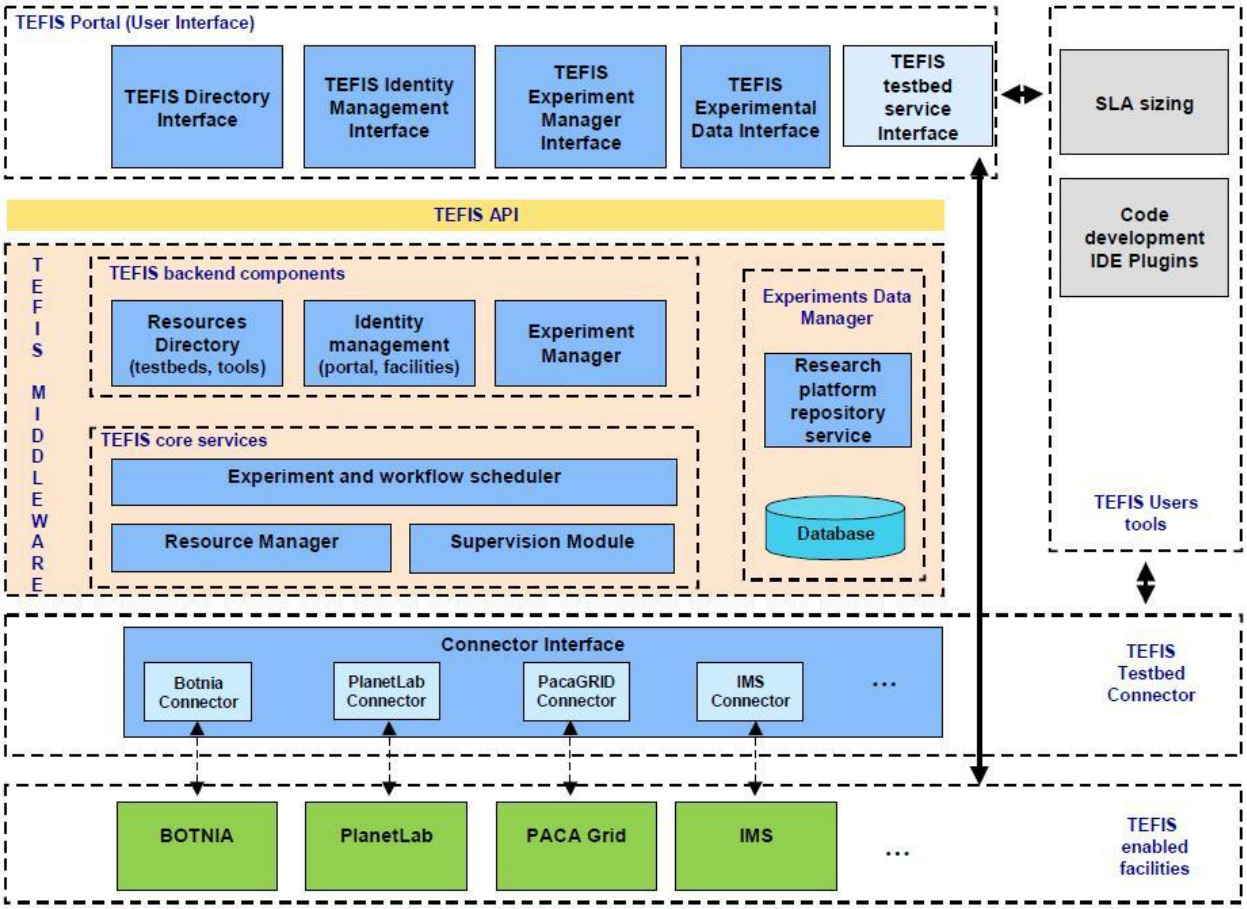

Figure 2: TEFIS Functional Architecture

The TEFIS framework is organized into four functional blocks: the TEFIS Portal, the TEFIS Middleware (including back-end components and core services), TEFIS users' tools, and the TEFIS testbed connectors.

2.1 The TEFIS Portal

The TEFIS portal will be the single access point to specify the experiments, so they can be executed remotely on the testbeds. This portal will allow testbeds to be searched for, the retrieval of reference data, configuration of tests to be performed, gathering of diagnostic parameters and more. It will also provide access to a wide range of tools. This functional block is divided into four different interfaces:

- Resources Directory
  The directory provides the repository of tools, hardware, software, documentation and the list of experiment facilities (testbeds) and its resources, being a fundamental infrastructure supporting the experiment lifecycle. The presentation layer will be implemented in the TEFIS Portal, while the actual directory implementation is a backend component within the TEFIS Middleware.

- Identity Management
  This interface is responsible for creating and managing accounts and access policies for TEFIS users, protecting the platform with simple and effective account management techniques. This system resides in the TEFIS middleware layer, while the TEFIS portal implements the user interface.

- Experiment Manager
  This interface will work with the TEFIS resources directory to list and configure resources in the testbeds, besides planning the experiment execution, allowing it to be executed on a combination of experiment facilities. TEFIS users will be given support in defining, executing and collecting the required data in their experiment.
  The interfaces in this system will be adjusted to the needs of each user. For the end-user (experimenter/researcher) specifically, this will be implemented using these subcomponents:

    - Experiment designer
      This will help users to formalize their requiriments and point out the correct tools, testbed and methodology for that experiment. In this step the inputs are the user needs and the specification and design of the experiment, while the platform outputs are the test strategy and a list with the types of tests to perform (functional, integration, etc.)

    - Experiment planner
      Once the user has specified the experiment, control switches to this interface in order to reorganize the knowledge provided by that user and link his needs to existing resources via the core services. In this step the user inputs are the test strategy and the user specific criteria for it, while the platform output is the test plan

    - Experiment workflow manager
      This step will complete the workflow process, based on the specification of the characteristics of the experiment provided by the user. It will send a request to the core servicesand obtain the complete list of activities and schedules. In this step the user input is the test plan, while the platform outputs are the partial results of the execution of the activities and an activities workflow

    - Configuration assistant
      The Configuration Assistant is responsible for guiding the user in their experiment definition, organization and monitoring tasks. The more specific the description of the

experiment, the more accurate will be the resources, facilities, tools and data provided to the user

- Experimental Data Interface
  This interface is responsible for processing data presented in the TEFIS platform. There are two main types of data to distinguish:

1. *Static data:* data used to identify the user and establish contractual information between user and test provider, as well as the definition of the experiment (configuration and application data)

2. *Dynamic data:* data created during and after the experiment run (performance, user data, consolidated results and reports)

With only two types of data there are still distinctions to be made. The static data will be held locally, since it has a small and tractable footprint, and does not require extensive storage. Meanwhile, the dynamic data will typically be remotely stored and accessed via a TIDS and it will require significant resource to store and manage, since this data will be large and potentially open-ended.

The following items identify the service provided by this interface:

- Registration
  - Users can identify data and information associated with a specific experiment or testrun

- Experiment Request
  - User looks for related experiments
  - User gives details about a new experiment they have run

- Experiment Monitoring
  - User wants to know how the experiment is progressing
  - User wants to decide whether to restart the experiment
  - User wants to evaluate the experiment output

- Experiment Publication
  - User wants to retrieve archived data
  - User wants to send data elsewhere

2.2 THE TEFIS MIDDLEWARE (BACK-END COMPONENTS AND CORE SERVICES)

The main goals of the TEFIS Core Services are:

- Execute low level workflows

  Since the TEFIS platform aims to provide a common framework to a set of several heterogeneous testbeds, it must be considered that each experiment (or kind of experiment) will differ from any

other, so that some of the steps in the high level workflow have to be made more specific for the lower level workflow

- Monitor resource status, matching and identification

  Provide information about the current status of a given process; as well as providing information on performance. This will help future experiment provisioning.

2.3 TEFIS CONNECTORS

TEFIS is expected to be dynamic (support testbeds joining and leaving) and heterogeneous (potentially any type of testbed can join). To manage those interactions (from an architectural point of view), the system relies on the concept of a connector.

The role of connectors in the TEFIS architecture is to handle the heterogeneity presented by the different testbeds, building an abstraction layer above the testbed interface, offering a common interface to all testbeds in the TEFIS platform. A homogeneous and uniform protocol is defined, in order to ease communication between the TEFIS platform and the testbeds.

Due to their location on the periphery of the infrastructure and because they are in charge of communication with external components, the connectors are also the central elements in one aspect of security management.

2.4 TEFIS USER TOOLS

The TEFIS user tools represent a long term vision, once the platform is operational, making it able to host external tools on the platform, in the same way that new testbeds are integrated.

3 Scenario Definition

Before running the experiment over the TEFIS platform the user needs to follow a few steps to create a good plan for their experiment. This planning description starts with the Scenario Definition, since a good experiment needs a good basis.

According to the experiment test life-cycle, this and the next section correspond to the plan phase in the life-cycle workflow.

The Scenario Definition is divided into three phases: the experiment overview, the hypothesis definition and the experimental method and procedure, as described below.

3.1 Experiment Overview

Before the experimenter starts the formal description of his experiment, he can create a quick description of what he is planning to do, making an abstract, to introduce the experiment and its scenario.

This stage can help the experimenter himself to think about the experiment in a high-level view, so he can work over this description to go into detail about the experiment, extracting the variables, metrics, and others features that are necessary.

Finally this section can help other people to understand the experiment. If a second person needs to use or continue the experiment, here will be the first step so he understands it.

3.2 Hypothesis Definition

When the experimenter starts the experiment specification, the first thing to consider is the scenario hypothesis, in which the experimenter will think and provide a formal definition of the proposed scenario hypothesis, including classes being studied, variables and expected causal relationships. Information should also be provided on how the hypothesis was derived.

This way the experimenter will have a formal description of what he intends to do, as well as the scenario background to this experiment, and thereby describing the motivation for this experiment.

To facilitate later verification of what hypotheses are covered, it is recommended that the experimenter make a bulleted list, with a short resumé of each hypothesis to be tested.

Making the bulleted list doesn´t mean that a more extensive description can´t be done. It is highly recommended that the experimenter creates an explanation of why he intends to verify this hypothesis, and why it is important to validate the experiment.

3.3 Experimental Method and Procedure

After the hypotheses are defined, the experimenter needs to focus on how he should proceed in order to validate this hypothesis list. In this section he should describe the experimental

procedure chosen for testing the hypothesis, providing a justification for why it was selected. This will include how independent variables are explored and results analysed.

Considering the testbeds as the principal way that the TEFIS platform offers the experimenter the means to test and validate his experiment, here he will focus on choosing the best testbed for each hypothesis.

So the best practice to do in this section is:

- View all testbeds that the TEFIS platform offers, studying their features;
- List all testbeds that have features that will help validate and test all the hypothesis previously defined;
- Create a short description of how each testbed will be used in the experiment, and why this test was chosen for that hypothesis;
- Planning how the testbeds will be used, i.e. which is the first testbed, if some test can be done in parallel, if any test will use the results of any previous test, etc;
- Create a scheme or a workflow to better visualize the interaction with the testbed.

Figure 3 is an example of the workflow chart based on the e-Travel case. A more detailed explanation can be found in D2.1.1, section 4.2.

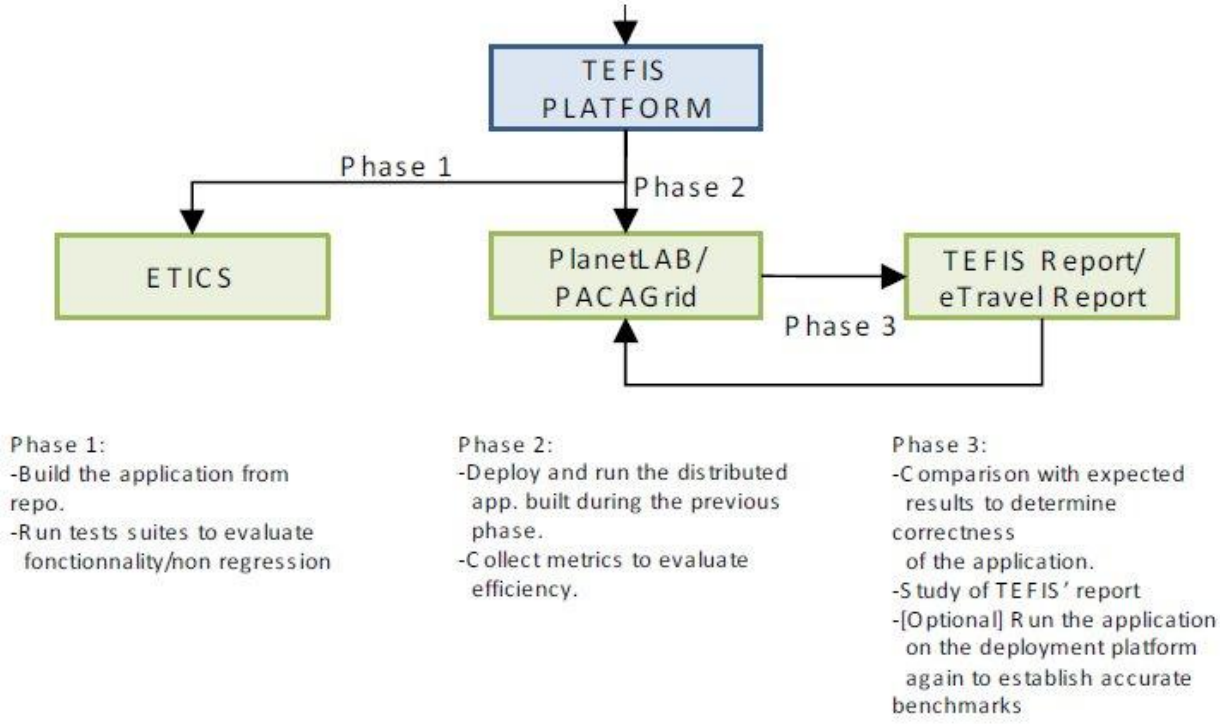

Figure 3: e-Travel use-case use of the TEFIS facilities

4 Design of the Experiment

After the experimenter has defined the scenario, i.e. created a background to the experiment, defined all the hypotheses, chosen the testbeds to perform the tests, he needs to go to a lower level in the experiment to extract variables, metrics, hardware, software and finally think about the steps to run the experiment on the platform.

In this section the experimenter needs to describe the overall objectives of the experiment design, focusing on the specific innovation and technology that the experiment will need to provide.

The information extracted from this section will go to fill some of the fields on the portal interface, when the experimenter needs to describe their experiment. It´s good practice to keep all this information documented in order to easily extract it later.

4.1 VARIABLES

In this section the experimenter should identify all variables in the experiment that he will want to measure, calculate or test.

To assist the experimenter's work, he should aim to answer the following questions:

- What do I change? (independent variable list)
- What do I observe? (dependent variable list)
- What do I keep the same? (controlled variable list)
- What irrelevant variables might mediate the effect of the independent on the dependent variables? (extraneous variable list)

The experimenter should understand that an experiment need not present all types of variable, e.g.: an experiment may have only independent variable.

As an example, some types of variables are listed below:

- Independent variable: Number of users in the system
- Dependent variable: Total bandwidth used by all users in the system
- Controlled variable: Max number of users in the system
- Extraneous variable: Power line jitter

4.2 METRICS AND MEASUREMENTS

Now that the experimenter has all variables listed, he can provide all metrics and measurements related to each variable. To do this, he should aim to provide answers to the following questions:

- What is the feature of interest?
- Name of metric:
- How is it measured?

- How often is the measurement made?
- Is it a raw measurement or pre-processed?
- What are the units of measurement?
- Is there any uncertainty in the measurement?

As an example we describe the measure of *time to transmit a form to the database*:

- Every time a form is sent to the database the platform will measure the time, in milliseconds, that it takes to arrive. This measurement will help establish the system response time, in case of a form sending.

4.3 HARDWARE AND SOFTWARE REQUIREMENTS

In this step the experimenter will describe all hardware and software besides the TEFIS platform that he will need to perform his experiment.

Sometimes the experiment needs an extra device, like a smart-phone or another hand-held device to run an application, or an external data server, containing a massive amount of data to be processed by another physical entity before entering the TEFIS platform. Along with such hardware items, additional software may be necessary beyond what TEFIS provides or which needs to be integrated with the platform to perform the tests.

In this section, thinking of best practices, the experimenter can first list all the requirements related to the hardware and software, and describe their significance to the experiment. e.g.:

- An application server to host the application web-services: Tomcat version 6.0
- A mobile device to run the data collection software, for instance, a smart-phone with the capability to connect to the web (3G, 4G or any other wireless Internet connection) and a GPS system, to run the experiment application.

4.4 GENERAL OVERVIEW SCHEME

At this point of the experiment design, the experimenter has a good idea of what he needs in terms of software, hardware, of all the variables to be measured and of how he will achieve it, along with all the testbeds he will use and the sequence by which to run the experiment.

Thereby, focusing on best practices, the experimenter may create a scheme correlating all the knowledge gathered until now, to facilitate the creation of the next and final stage of the design.

Generally this scheme will contain the testbeds in the background, and on top of that the software, hardware and the data flow inside the experiment and the application.

A sample scheme is show in figure 4, using three generic testbeds, running a generic experimental application, which will generate random data, pass on to an application running over a cellphone, sending the data over a network to a webservice, where the data will be processed.

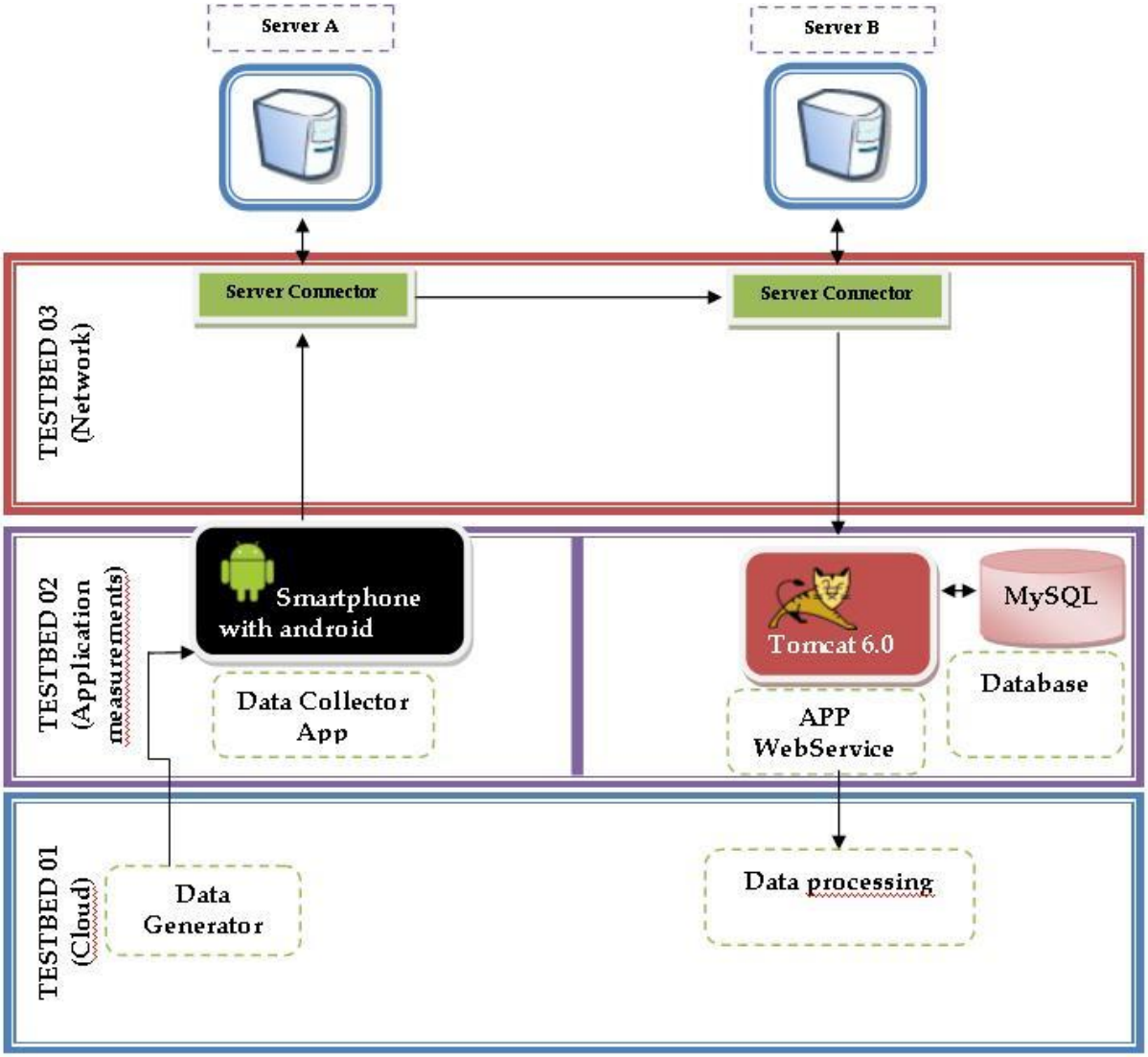

Figure 4: Scheme of a generic experiment on top of testbeds and a hardware/software relationship

4.5 STEPS

The final stage of the experiment design is planning the steps for running the experiment through the TEFIS portal.

Now the experimenter has all the information required to create the steps, as he is already aware of the platform architecture, from the information available in the document 2.1.1 or in the brief explanation in section 2 of this document, and he knows all about his experiment, as planned in the previous sections. So the experimenter will be able to write the steps to run the experiment.

The steps are brief descriptions of what he will do, sequentially, in each part of the portal, including the inputs and outputs expected in the platform.

To easily implement the steps, the experimenter can use the scheme created in the preview section together with the platform architecture, that is mainly the user interface part of the platform (portal). In table 2, the steps for a generic experiment are listed to help him with his own experiment.

Table 2: Description of steps for a generic experiment running on the TEFIS platform

Step	Summary	Description
1	Software developer creates a TEFIS account	All the information required for the account are provided
2	Specification of the experiment by the user via the portal	Once an initial experiment has been described in TEFIS and a test profile created, then for subsequent test runs, the user simply edits what is already held in the TEFIS Experiments Data Manager
3	Create description of the desired SLA, stressing factors (number of resources and input load)	
4	Store experiment description in the TEFIS platform	
5	Selection of the most suitable testbed for the experiment	Choice made based on hardware requirements and the metrics wanted
6	Initial configuration of the chosen testbed is done through TEFIS; any other configuration activities are done using the WS-API; reports are fed back for presentation via the portal to the user	This is related to the TEFIS workflow definition, testbed configurations and then the actual submission of the experiment via the TEFIS platform. The user needs to define the set of resources first that they would like to use from among the available testbeds, then the order/conditions in which they are to be used and finally configure the testbeds directly from their TEFIS workspace.
7	Provide Application and software	1. The user provides the configuration which contains all dependencies required to build the application. Either the user specifies where to get them or they are uploaded directly, if is the case.
8	Prepare a workflow to conduct the test (tasks and number of nodes)	
9	Send a job, which will be executed when resources become available	2. Depending on the result of the build and test suite processes, TEFIS deploys the application to the testbed. The configuration of the application is very dynamic, hence these configuration files need to be

		tagged to show that they will need modification for a successful deployment; and relevant pieces of information need to be updated once they become available (like the machine hosting the given services, for instance)
10	Prepare Virtual machines (install required software: database, libraries, and so forth.)	These steps would be TEFIS internal; the user does not expect to have to configure things manually whilst the application is being deployed or has already been deployed. Any required software, configuration files and so forth need to be dynamically modified and provisioned on the target nodes
11	Deploy the code to the nodes of the selected testbed(s)	
12	Execute the tests	
13	Collect the data manually or directly from directly from the monitoring facility of the testbed in use, if direct access to the testbed is available. Otherwise data will be requested via the TEFIS supervision manager	The user is notified that some metrics are available. Depending on the results, he may then decide interactively to run step 2 of the experiment again or exit the experiment. In the case where the application is to be run again, the experiment flow goes back to steps 6 to 9 where the input for the experiment may be modified and the values to be collected refined
14	The user is notified of the end of the experiment	
15	Experimenter can decide to re-run the experiment depending on the data collected. He can launch the test again and decide to include new worker nodes to the experiment, change experiment input, and refine the data set to be collected. At this point, he can also exit the experiment.	
16	The experiment restarts from step 9	

5 Experiment Implementation

At this point the experimenter will start implemention, passing through the steps described in the preview section, focusing on the first steps, going from account creation to software deployment.

The life-cycle of the experiment is completed by the implementation section, by the execution of the request and by the provision/deploy phases.

As most of the steps are done in the portal, the experimenter may refer to D3.1.1 [2] for a more complete overview of the portal.

5.1 LOGGING INTO THE PORTAL

The first thing to do in the TEFIS portal is to log in, so the user gains access to the platform and is able to design and run the experiment.

If the user doesn't have a portal account, he must create one to join the platform. This is done on the initial page of TEFIS platform, and it is shown in the figure 5.

Figure 5: TEFIS portal - main page

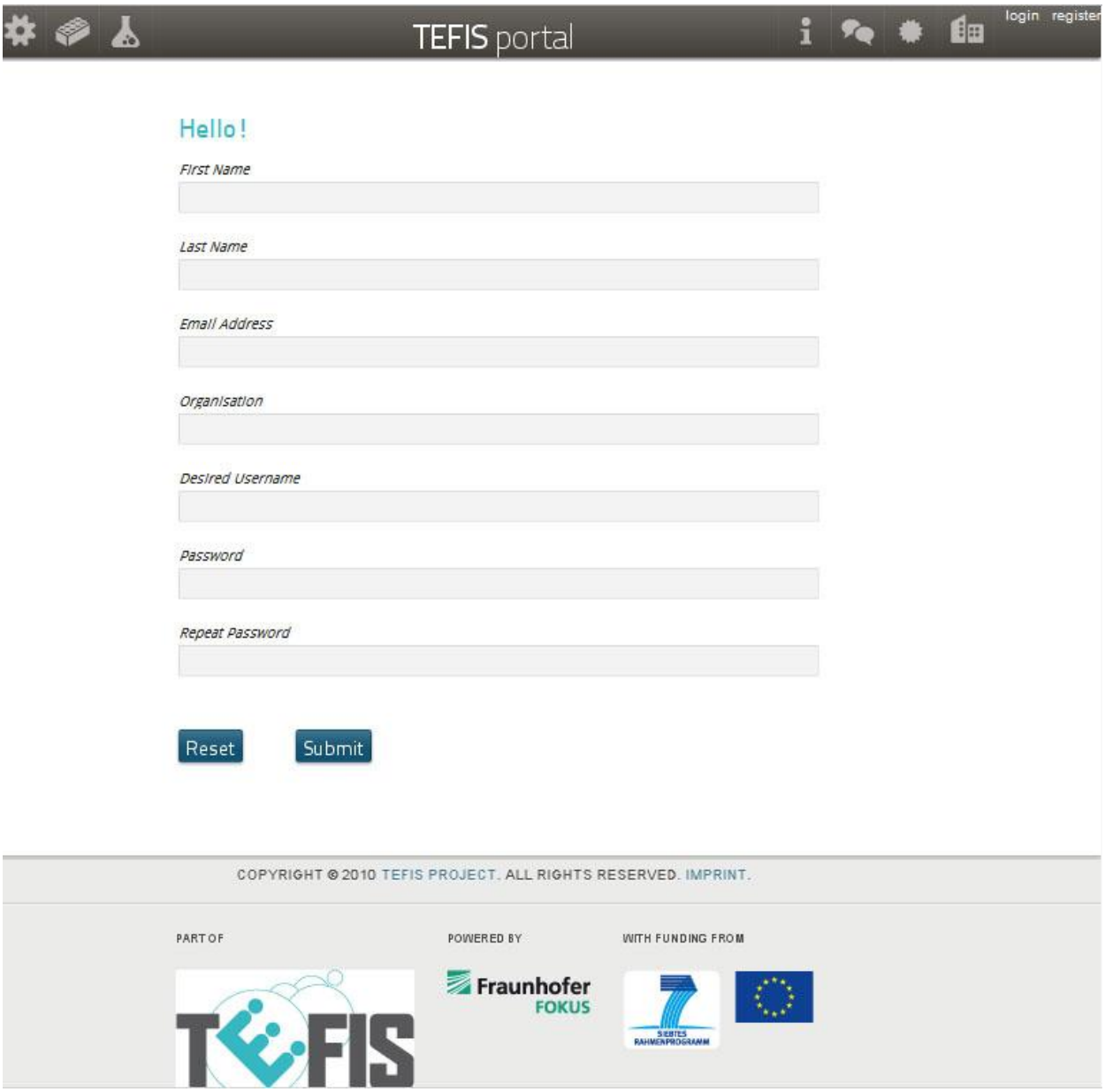

Figure 6: TEFIS Portal - register page

To register, the user simply completes all fields in the form and presses the submit button, as shown in the figure 6 to register into TEFIS platform.

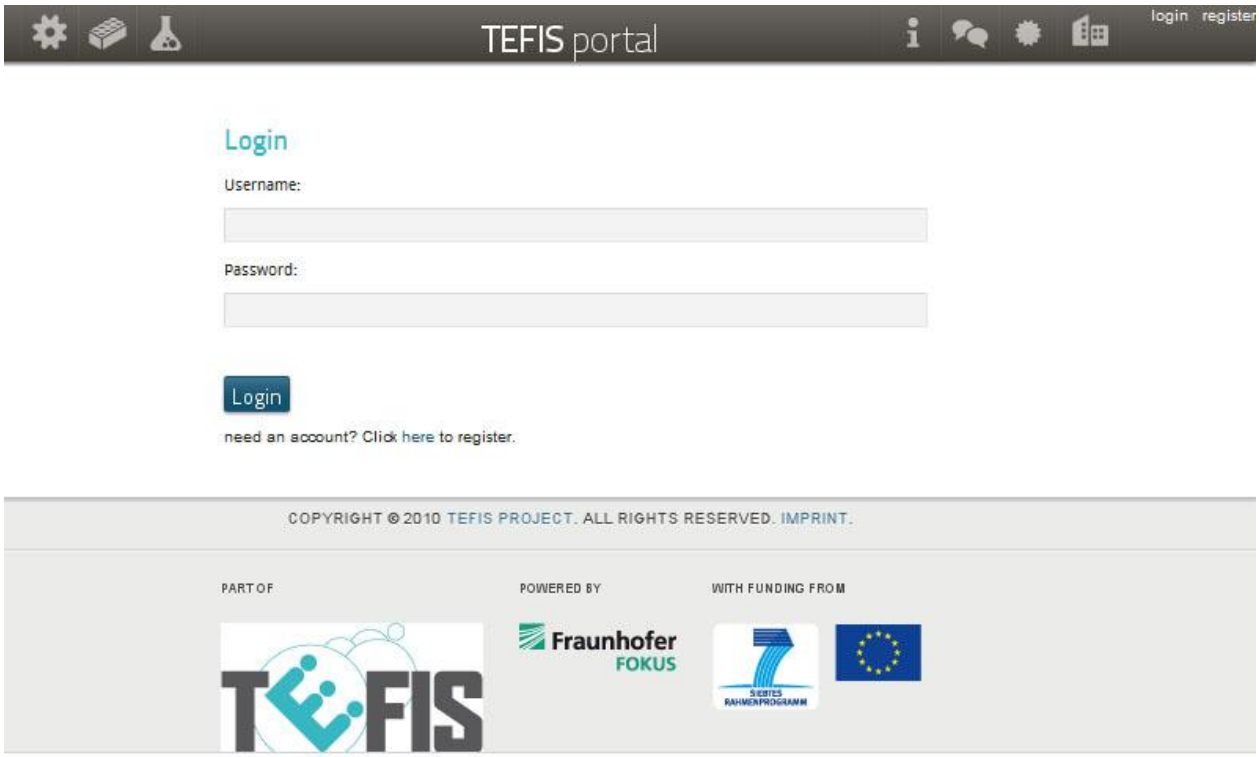

Figure 7: TEFIS Portal - login page

Once the user has created his account and logged into the portal, he will have access to the dashboard areas (PTMs, resources and experiments).

5.2 SPECIFICATION OF THE EXPERIMENT

Logged into the portal, the user enters the experiment area in the dashboard, and will be able to specify the experiment. The user can create the description of the desired SLA, stressing factors (number of resources and input load), variables that are to be measured or tested and the relative KPIs. He will then store his experiment description on the platform and save its specification.

In the experiment interface the user will use the "Experiment Design Flow", in which he will select the phases to create the experiment. In this first step, the user will enter in the "Select or create an experiment" just by clicking the link, as shown in the Figure 8.

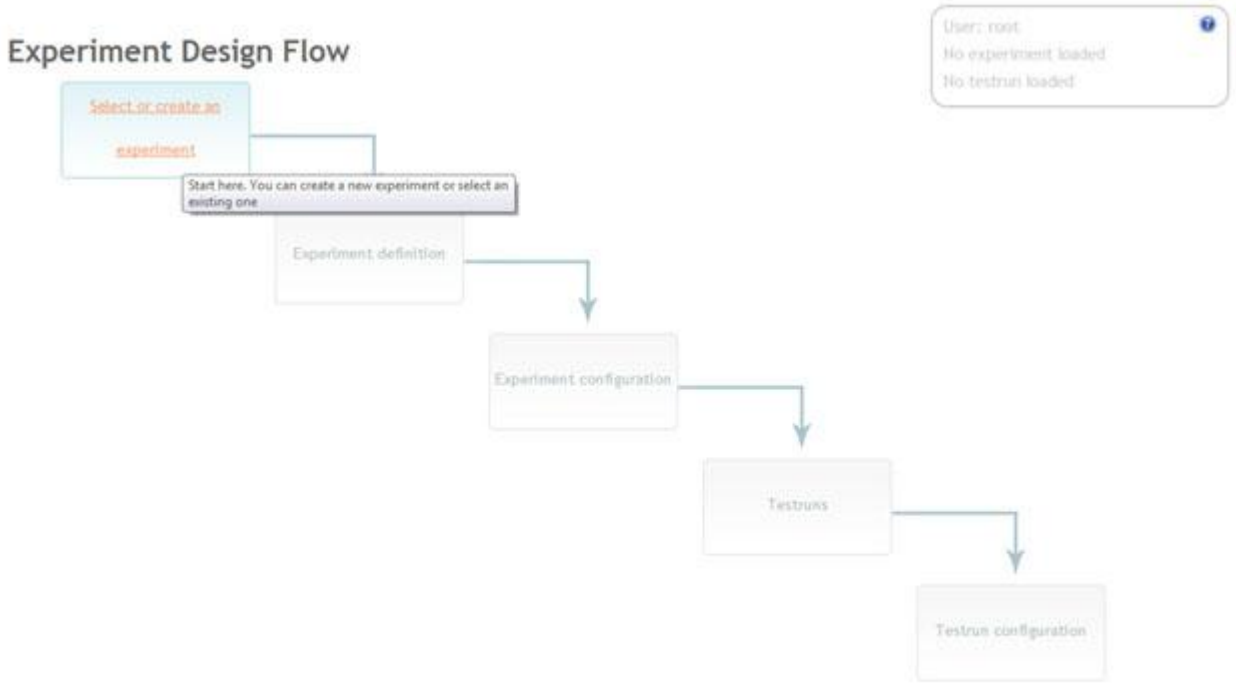

Figure 8: Creation of a new experiment on the Experiment Design Flow

Once in this section, the user is presented with a screen in which he can search for or select an existing experiment, or create a new one by clicking on the "Advanced experiment configuration" link. Figure 9 shows the screen in question.

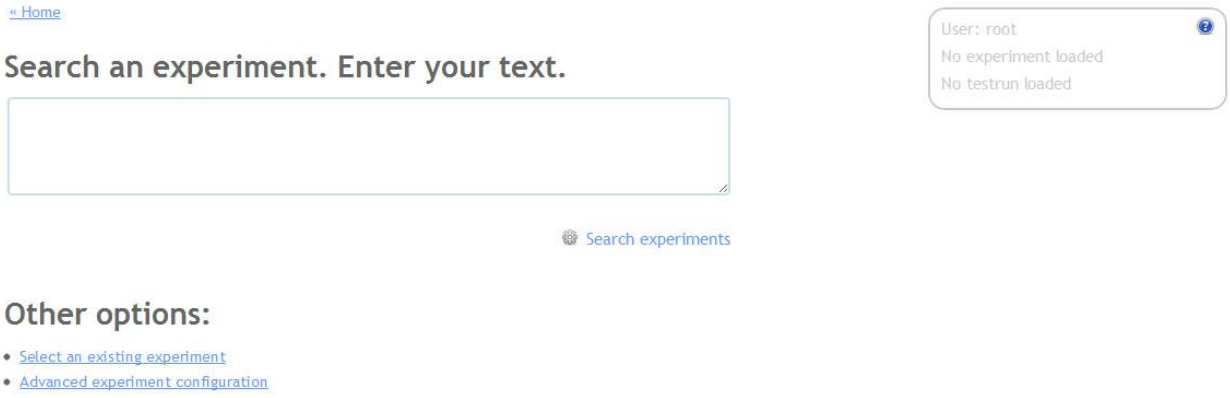

Figure 9: Creating or searching for an experiment

At this point, if the experimenter chooses to create a new experiment, the portal will go back to the "Experiment Design Flow" and the user will be able to proceed to the "Experiment Definition" step, as shown in figure 10.

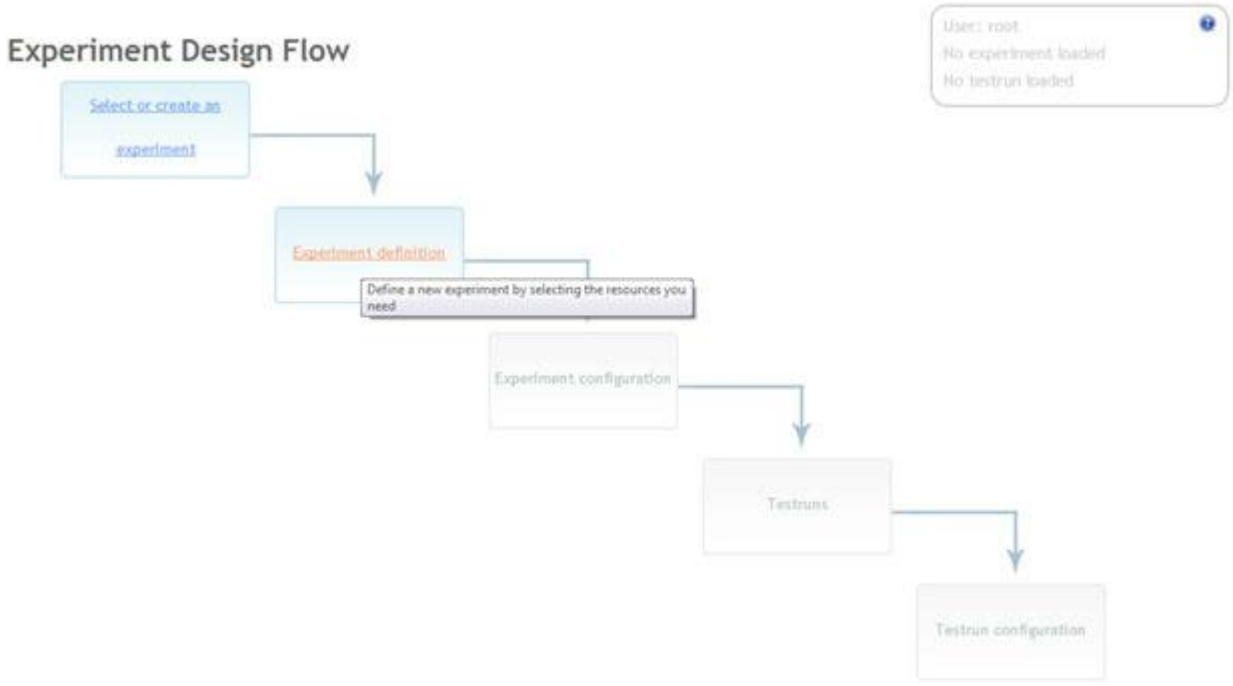

Figure 10: Definition of a new experiment

With this step, the experimenter will use the "Advanced Experiment Definition", where he can choose a name and a description for the experiment.

Once an initial experiment has been described in TEFIS and a test profile created, then for subsequent test runs, the user simply edits what is already held in the TEFIS data store, i.e. he doesn´t need to describe or specify the experiment every time he wants to perform a test. If the experimenter has spent some time to document the experiment, as shown in the previous sections, this part of the implementation will only be a copy of what he has already done.

The user can select the previous experiment, entering on the "select or create an experiment" in the "Experiment Design Flow" "Select an existing experiment". After this, the user will reach the following page shown in the Figure 11.

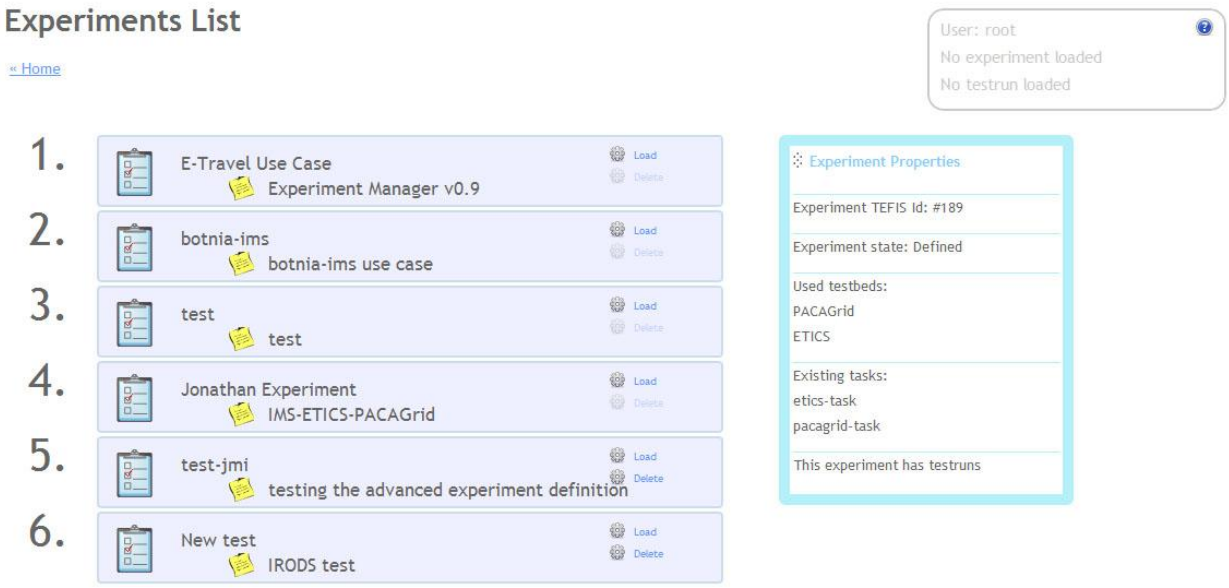

Figure 11: List of existing experiments available

To open an experiment, just select the one desired and hit the “Load” link.

5.3 REQUEST

In this phase the experimenter will focus on the testbeds, mainly on gaining access to the resources available. A list of available resources and testbeds, at the time of the experiment, can be found in the portal, under the resources page, which is accessible after login to the portal.

If the experimenter is following this document, at this point he has already chosen the resources and the testbeds to be used in the experiment, so he can check on this page if the resource is available at the moment of the experiment.

In the main portal, the user can view a list of the available resources that can help the experimenter to choose the correct testbed for his experiment.

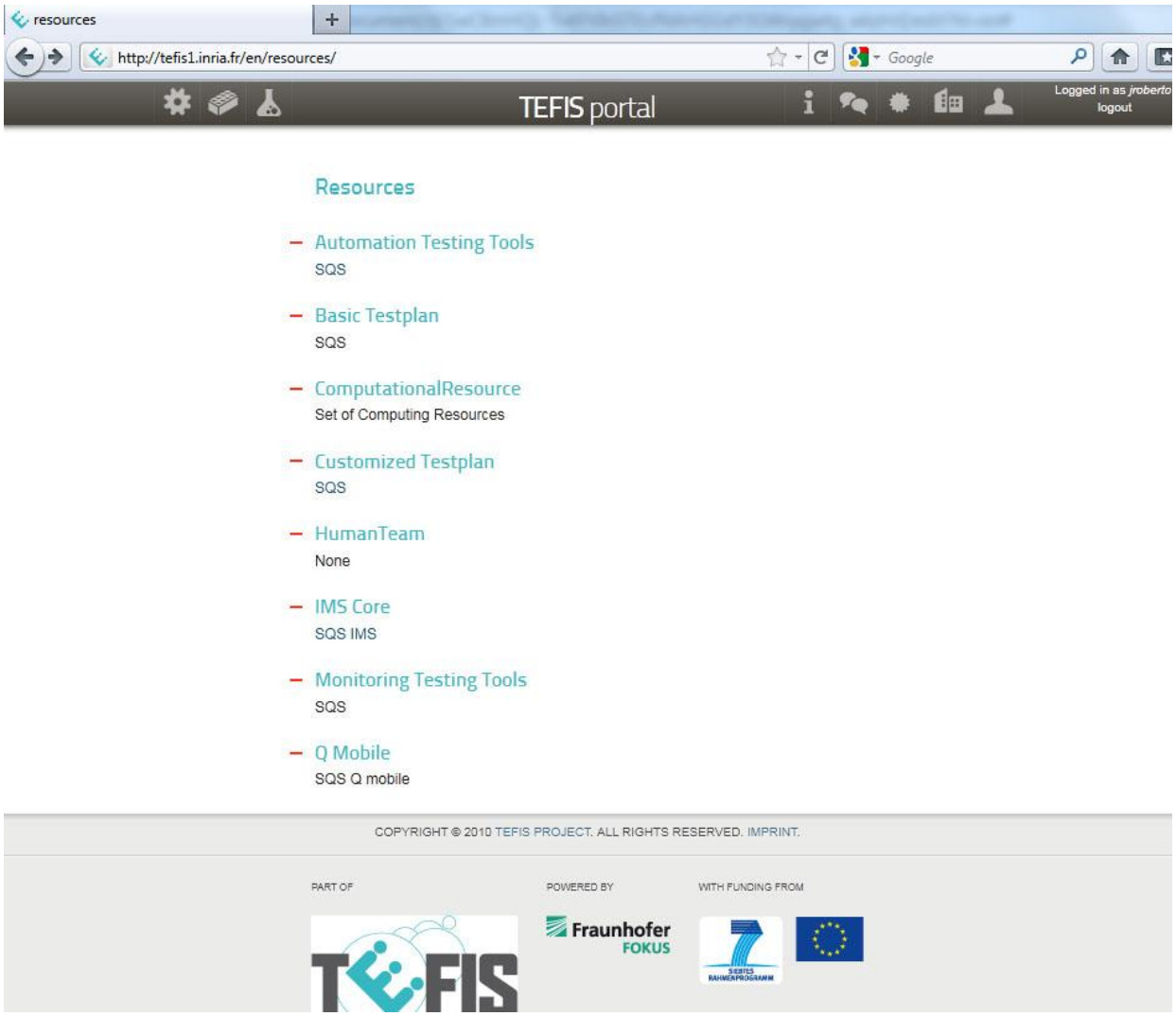

Figure 12: TEFIS Portal - resources page

All the steps of this phase will be done essentially in the TEFIS portal, so the user doesn´t need to interact directly with the testbeds. However, some testbeds used in the experiment need to be contacted to verify availability and to perform any reservation if needed, though for the most part the TEFIS platform will handle this. Cases do exist, due to the individual testbed's policy, in which the experimenter will need to make this contact himself.

In the same location where the user creates the name and description of the experiment, he needs to choose the "Test Types", the "Domains" and the "Keywords", all related to the testbeds that the experiment will use.

Figure 13 shows this page, in which the user can select multiple "Test Types", "Domains" and "Keywords" for the experiment, by clicking then on the " Save this resource list". After that the portal will return to the "Experiment Design Flow" page.

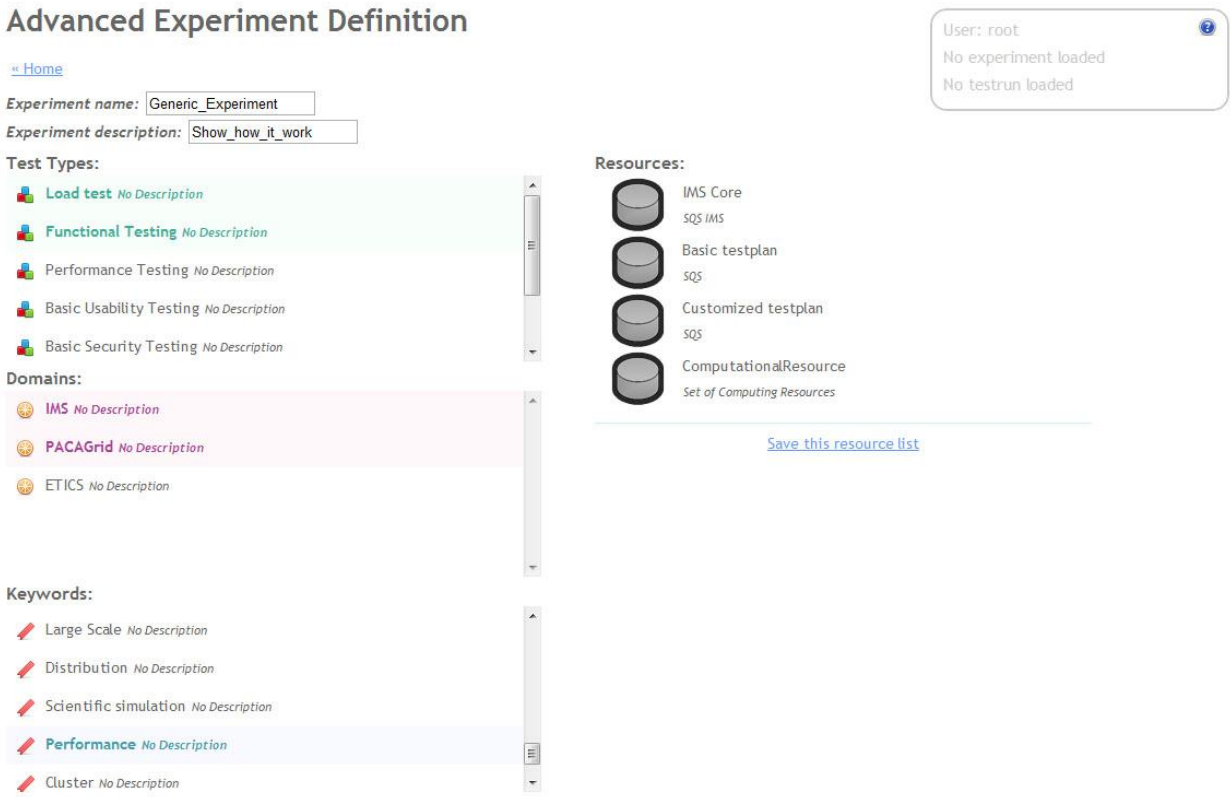

Figure 13: Defining the experiment name and description; and choosing test types, domains and keywords related to the testbeds used by the experiment

After all testbeds have been selected, the user will need to configure them. The initial configuration will be done directly in the TEFIS portal, but any other configuration activities may be done using the WS-API, provided by the testbed. This will produce reports containing feedback for presentation to the user at the portal.

After the portal returns to the “Experiment Design Flow” page, the user will notice that he may not select the definition again, since the experiment has already been defined. However the user can configure his experiment using the “Experiment Configuration” step, shown in the Figure 14.

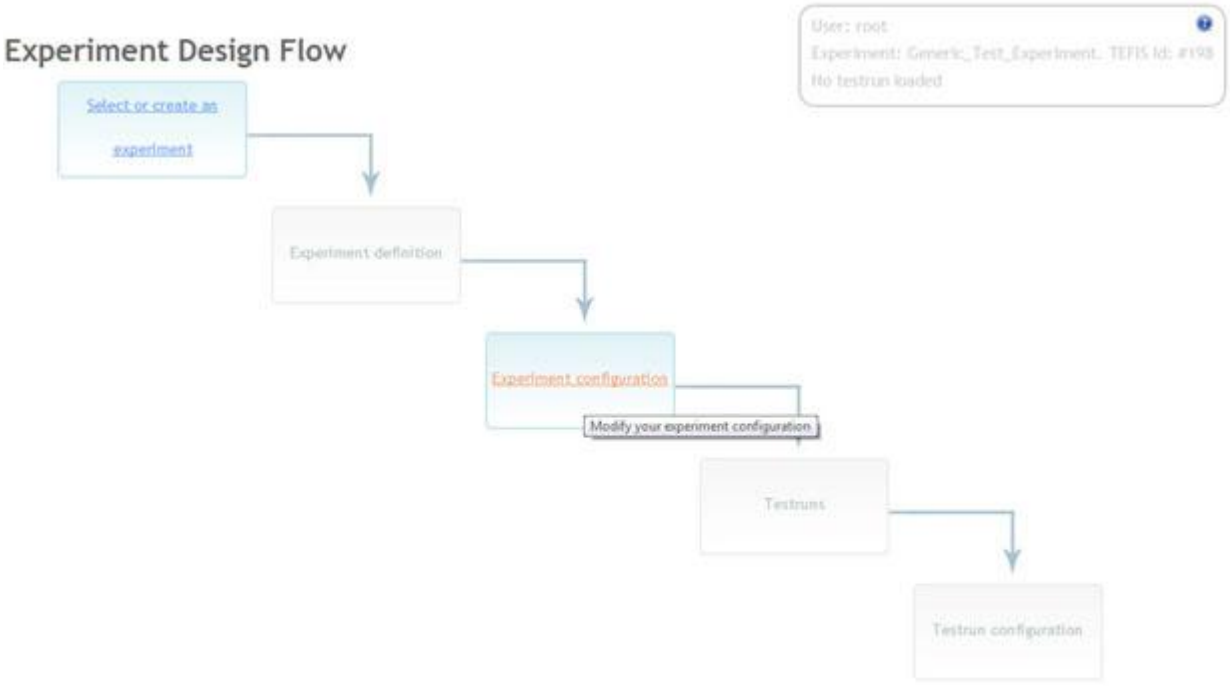

Figure 14: Configuration of an experiment already defined

Inside the "Experiment Configuration" page, the user will find the Available Resource Types, in which the experimenter can drag and drop the resources needed, change the properties of each resource container and, after everything has configured, click on the Save experiment link.

Figure 15 shows a generic experiment configuration, where the portal shows the pop-up box to alert to the Save process. It can be seen in this figure, that the experimenter can change the colour of each resource to help visualize the flow, and he can add tasks to help in the experiment workflow, discussed later in this document.

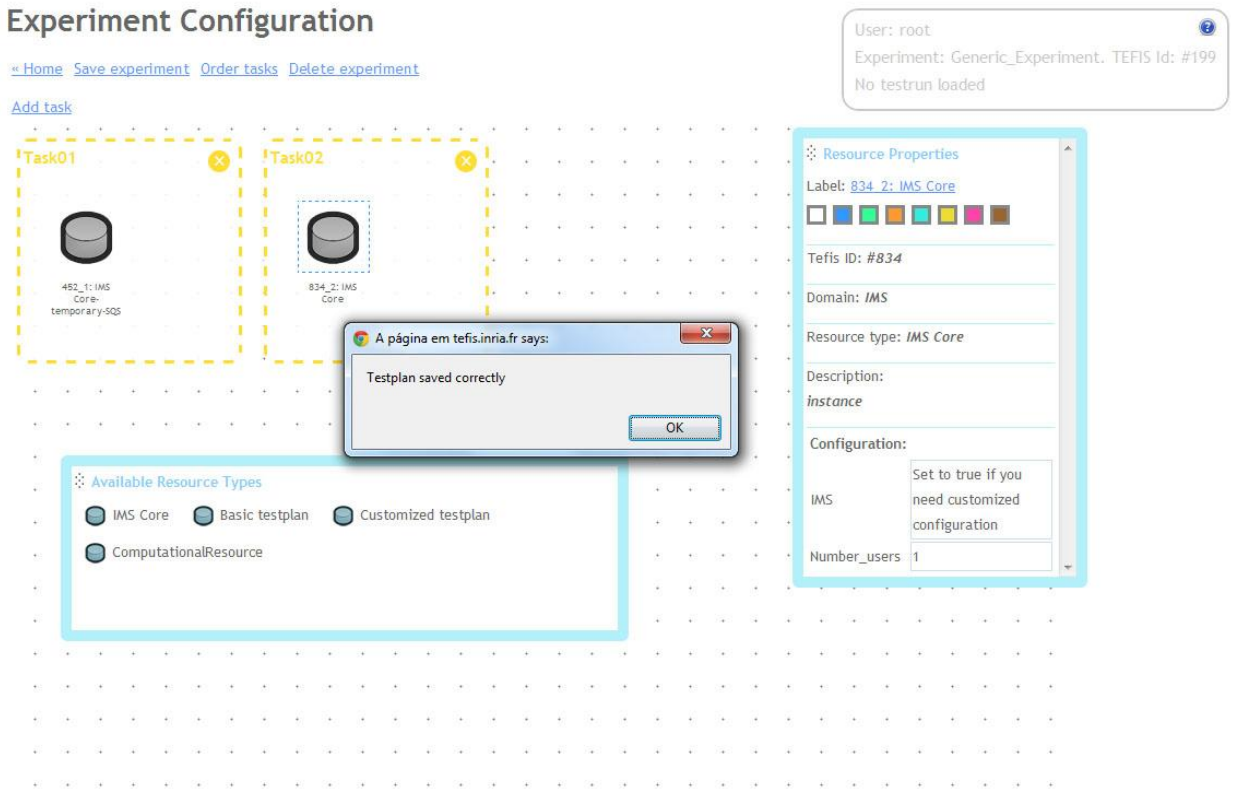

Figure 15: Configuration of a generic experiment

The next step is to prepare the workflow to conduct the test. For this, the user can rearrange the tasks in the Order tasks link under the Experiment Configuration.

Figure 16: Arrangement of taskorder

To arrange the tasks the user can just drag and drop the one selected to the desired place. Once the testrun is done, the experimenter cannot change the workflow. To do this, it is necessary to delete the older testrun, go back to this phase and redo the task order.

After workflow preparation, the user will need to send a job to the portal, which will be executed as planned in the workflow and when resources become available, in accordance with the testbed contract for this experiment.

The user needs to go back to the Experiment Configuration page, select a task and insert the job file in the configuration window as show in the Figure 17

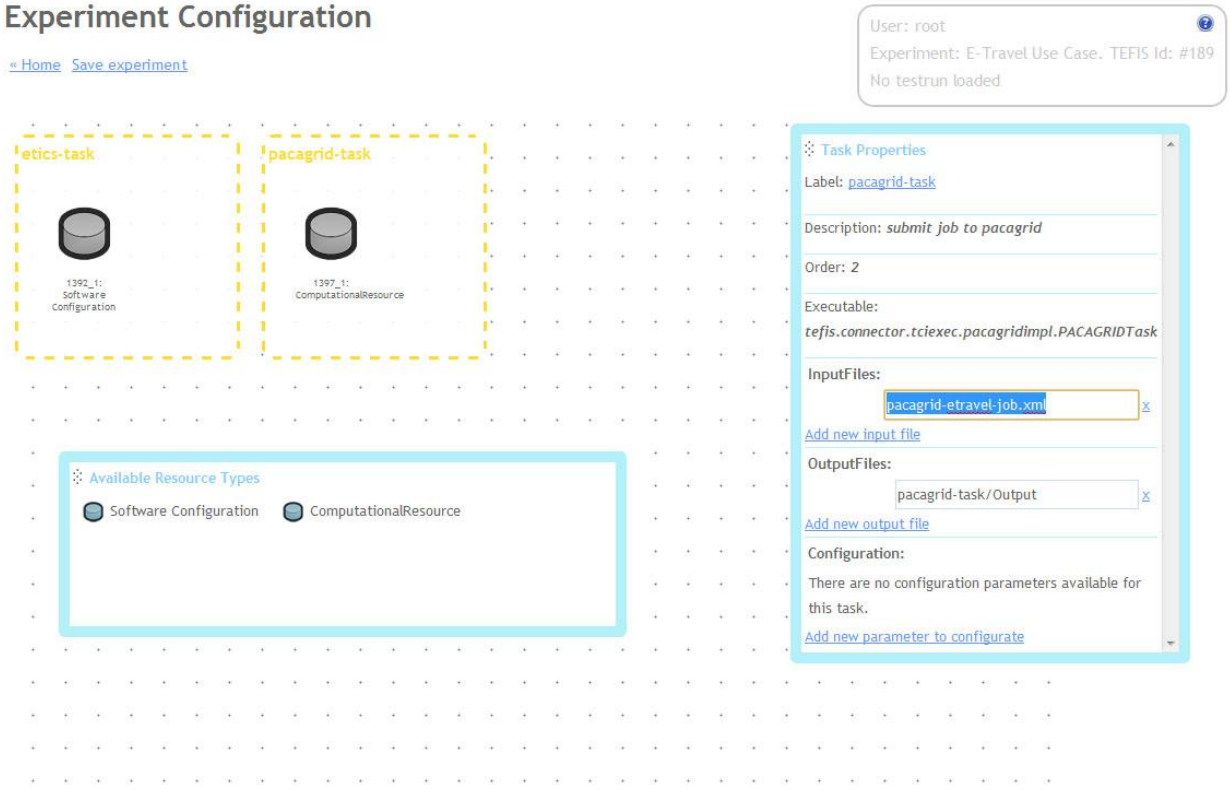

Figure 17: Insertion of the job files into a task during experiment configuration

After sending the job to the portal, the user will then prepare Virtual machines, i.e. he will install the required software to run the experiment: database, libraries, other third-party software etc.

Finalizing the configuration, the request phase of the life-cycle is done. But the experimenter can come back to this phase to choose others testbeds, resources or do another configuration after he has collected the results of the test.

Whenever back at this phase, all previous data are stored in the TEFIS database, so the user doesn´t need to configure them again. He can simply edit the data or start over again.

As a reminder once the experiment is described, the user can´t redo it; he will need to create a new experiment from the first step, in the event that he has missed a resource type or wants to change the experiment type.

5.4 PROVISION / DEPLOY

After the request phase, with the testbeds properly selected and configured, the user will start the provision and deploy phase.

In the provision phase, the user provides the configuration which contains all the dependencies required to build the application, including any third-party software that the experiment needs. Either the user specifies where to get them or they are uploaded directly into the portal.

Depending on the result of the build and test suite processes, the TEFIS platform will deploy the application to the correct testbed. The configuration of the application is very dynamic. In concequence, the configuration files must be identifiable as needing modification to make the deployment succeed; and relevant pieces of information will need to be updated once they become available, i.e. the machine hosting the given services.

All this is done in the Test Run phase of the Experiment Design Flow. After the experiment configuration has been done, the portal will turn to the test run task, and the user will be given access to this phase as shown in the Figure 18

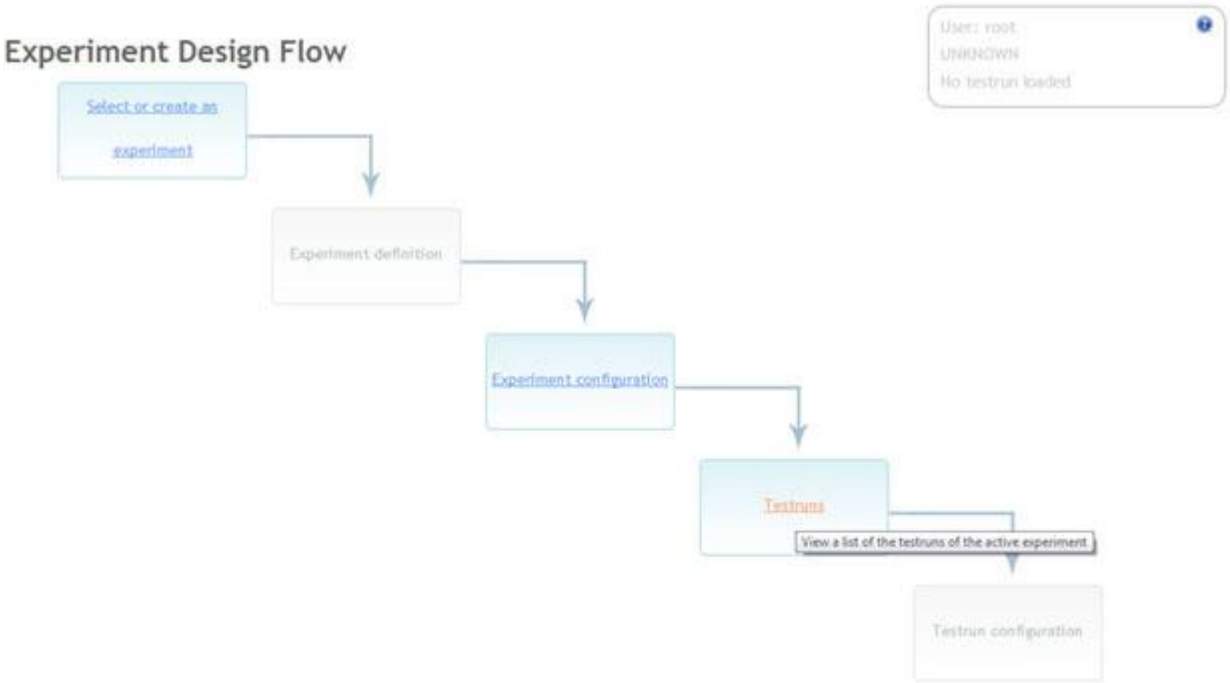

Figure 18: Platform allowing for the creation of testruns for already configured experiments

Entering this task, if a new experiment, the TEFIS portal will ask the experimenter to create a new test run, as shown in Figure 19.

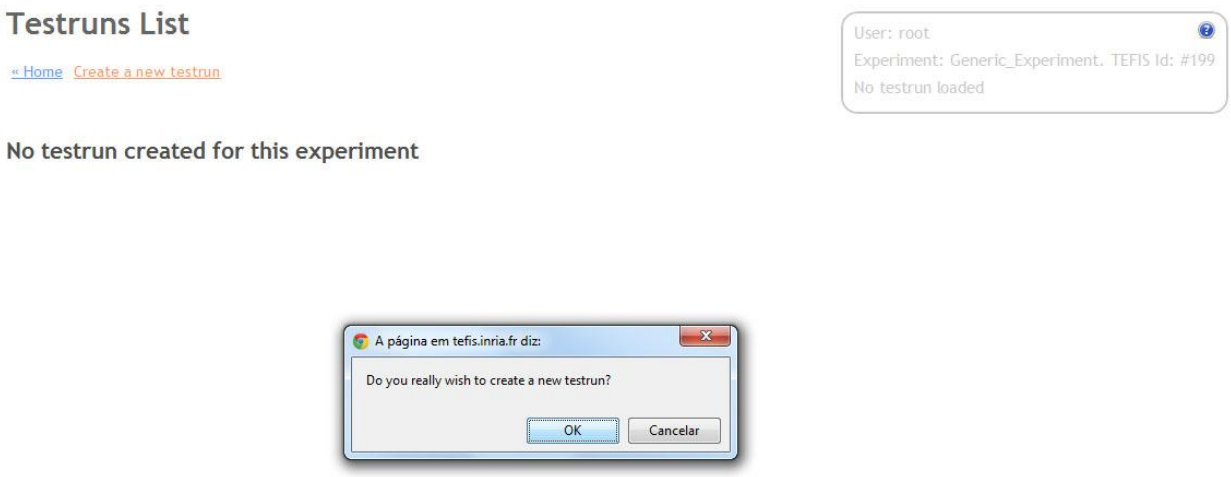

Figure 19: Creation of a testrun for new experiments

After the user confirms the creation of a new testrun clicking in the OK button, the system will show the process status and when it's done, the portal will show the testruns list as can be seen in Figure 20.

Figure 20: List of created testruns

If this experiment has been loaded, the portal will directly show the testrun list with all the testruns that have been previously saved.

To continue the experimenter the testrun just needs to be selected and click on the load button, and the testrun configuration page will open.

Figure 21 shows the testrun Configuration page, with the ordered tasks.

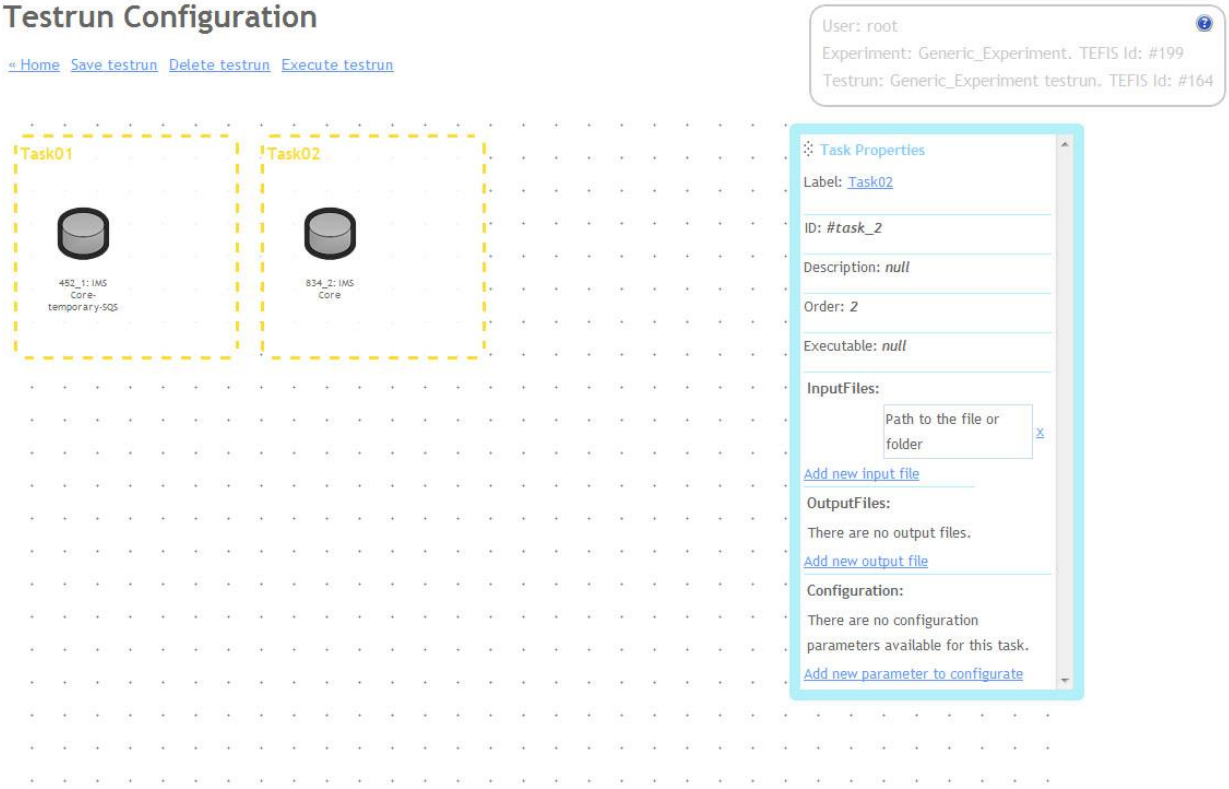

Figure 21: Testrun configuration page with ordered tasks

Now that all has been set up in the portal, the last thing to be done before the test battery is started is to deploy the experiment code to the nodes of the selected testbed, and the experiment is ready to be tested by the platform.

It should be remembered that these final steps are internal to TEFIS; the user is not expected to configure things manually whilst the application is being deployed or has already been deployed. Any required software, configuration files and so forth need to be dynamically modified and provisioned on the target nodes.

6 Experiment Execution

6.1 RUN PHASE

With everything having been set up in the portal, the experimenter can finally run the tests. For that the TEFIS platform just needs to have the range of possible actions initiated, depending on the type of testbed and the type of experiment.

In TEFIS there are three approaches to execute actions on testbeds:

1. Actions are performed manually by the experimenter on the testbed
2. A workflow is executed by the TEFIS Resource Manager
3. The TEFIS system exploits, if available, the testbed's execution engine to submit a specific workflow for execution, using the testbed connector, which is responsible for wrapping the execution engine in an abstract and general interface usable within the TEFIS system.

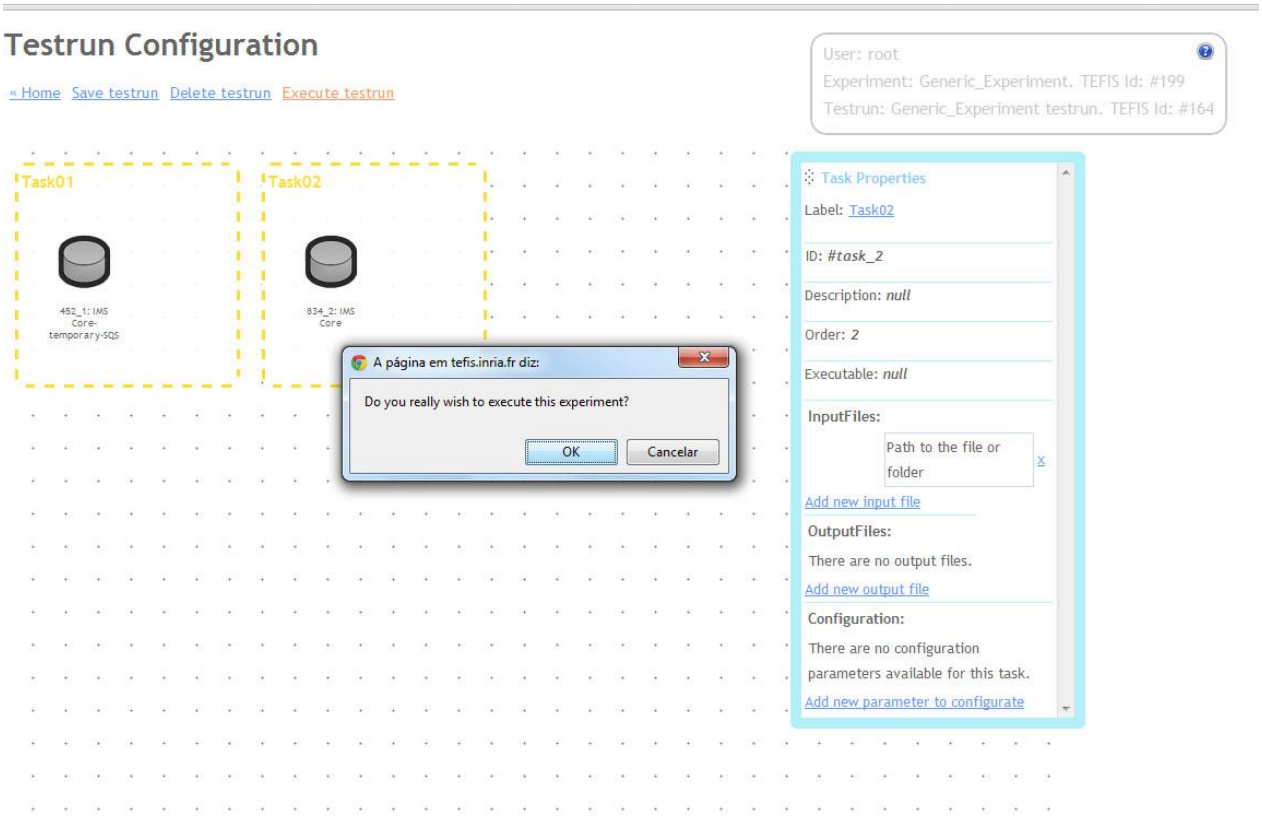

Figure 22: Execution of a testrun on the TEFIS portal

6.2 COLLECT AND EVALUATE PHASES

As the platform finishes the tests the experimenter can collect the data manually or directly from the monitoring facility of the testbed in use, if direct access to the testbed is available. Otherwise data will be requested via the TEFIS Supervision Manager.

Once all the test information has been collected, the user will be notified about the end of the experiment and now he has three options:

- He can simply decide that the information available at this point is enough and finish the tests.

- He can decide just to re-run the test without modification to see if any of the results change in a second run.

- Finally he can launch the test again from the beginning, with new workings, changing the experiment input and refining the data set to be collected.

7 Acronyms

3G, 4G	3rd/4th Generation
A-QCM	Automated Quality Certification model
ETICS	e-Infrastructure for Testing, Integration and Configuration of Software
GPS	Global Positioning System
IMS	IP Multimedia Subsystem
ISO	International Organization for Standardization
KPI	Key Performance Indicator
PTM	PanLab Testbed Manager
QoS	Quality of Service
SLA	Service Level Agreement
SQS	Software Quality Systems
SS7	Signaling System 7
TIDS	Testbed Infrastructure Data Service
WS-API	Web Services Application Programming Interface

8 References

[1] TEFIS deliverable: D2.1.1
[2] TEFIS deliverable: D3.1.1
[3] TEFIS deliverable: D3.3
[4] TEFIS portal: http://tefis1.inria.fr/
[5] TEFIS experimente manager: http://tefis.inria.fr/tefis/
[6] CoMon: http://comon.cs.princeton.edu/