



Deliverable Title	D4.1.1 – Resource Manager prototype
Deliverable Lead:	INRIA
Related Work package:	WP4
Author(s):	Franca Perrina
Dissemination level:	Public
Due submission date:	30/04/2011
Actual submission:	30/05/2011
Project Number	258142
Instrument:	IP
Start date of Project:	01/06/2010
Duration:	30 months
Project coordinator:	THALES

Abstract

This document gives an overview of the prototype delivered in the task D4.1.1. It presents the main functions provided by the delivered solution and it gives installation instructions and details about the prototype running in the TEFIS infrastructure operated by INRIA.



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

License

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Project co-funded by the European Commission within the Seventh Framework Programme (2008-2013)

© Copyright by the TEFIS Consortium



Versioning and Contribution History

Version	Date	Modification reason	Modified by
0.1	28/04/2011	First Draft	Franca Perrina
1.0	15/05/2011	Final Version	Franca Perrina

TABLE OF CONTENT

1. Execution summary.....	6
2. Introduction.....	7
3. Novelties and design decisions.....	7
4. TEFIS Resource Manager overview.....	10
4.1. Remind about Teagle.....	10
4.2. Teagle adoption in TEFIS.....	12
4.3. The ProActive Resourcing tool.....	13
4.4. The ProActive Resourcing tool in TEFIS.....	14
4.5. The main usage scenarios.....	15
4.5.1. The testbeds' resource provisioning.....	16
4.5.2. The experiment execution.....	17
5. The installed Prototype.....	18
6. Conclusion.....	20
References.....	21

TERMINOLOGY

VCT	Virtual Customer Testbed
REST	Resource State Transfer
PTM	Panlab Testbed Manager
Process step	A task on an experiment
TCI	TEFIS Connectors Interface
TCI-RM API	TEFIS Connectors Interface - Application Programming Interface for Resource Management
TCI-EXEC API	TEFIS Connectors Interface - Application Programming Interface for EXECution

1. Execution summary

This document covers the arguments related to the reasoning and the choices behind the implementation of the Resource Manager component of the TEFIS platform (see deliverable *D2.1.1 – Global architecture and overall design* to get more details about TEFIS platform architecture).

Firstly, in section 3, we identify the features and requirements the TEFIS Resource Manager component must provide and match to realize the management of the resources needed to run the experiments of the users and we argue the choice of existing software products, TEAGLE and ProActive Resourcing tool, to provide those features and match those requirements. In particular, we describe the interaction between TEFIS components and TEAGLE during the different phases of the experiment management (i.e., experiment definition, configuration and execution) to state the usefulness of TEAGLE in the TEFIS platform.

In the section 4, we discuss how resource management is performed in TEFIS through the interaction between the TEFIS Core Service, TEFIS Resource Manager, and the TEFIS testbed connectors TCI-RM API. We present the TEFIS Resource Manager as a component based on the cooperation between two existing software products, TEAGLE and the ProActive Resourcing tool. The section contains also a reminder about TEAGLE, the resource description model it uses and about the most relevant architecture components of TEAGLE like: the Panlab Testbed Manager, the Resource Adapter, the Repository, the VCT tool, the Request Processor, the Policy Engine, the TEAGLE gateway and the Web Portal. Among the TEAGLE components, in the section 4.2, we explain why in TEFIS we adopt all of them except the TEAGLE VCT tool and Web Portal in order to support users in the definition, configuration and execution of their experiments.

Then we move to the description of the ProActive Resourcing tool introducing some common ProActive concepts: Node, Node Source and Infrastructure Manager. Stated that the ProActive Resourcing tool is a client-server application, we explain its architecture. Hence, we talk about the command line and GUI clients and we describe some other component of the ProActive Resourcing tool: the RM Server, the Selection Manager and the Monitoring module. We outline the adoption of the ProActive Resourcing tool in TEFIS as the tool that provides computational resources to the ProActive Scheduling tool that, actually, enacts the low level workflow created from the user experiment.

At the end of the section 4, and precisely in the subsections 4.5.1 and 4.5.2, two scenarios to explain how the TEFIS Resource Manager works are depicted: the testbed resource provisioning and the provisioning of the internal computing resources to the Experimental and Workflow Scheduler during the low level workflow execution.

The last section, 5, presents the installation of the TEFIS Resource Manager prototype and explains why some components, TEAGLE Gateway and TEAGLE Policy Engine, are not installed yet, even if in the document their usefulness is established.

2. Introduction

This document serves as a supporting document for the Resource Manager prototype delivery. It aims to give a general idea of the main functionalities provided by the delivered solution, as well as information about the prototype currently installed within the TEFIS platform.

It is structured as follows:

- the first section gives a functional overview of the system. It starts from an introduction about the Teagle components and the ProActive Resourcing tool used as the basis of the delivered solution. For each of them it presents the role assumed in the TEFIS architecture and the way it is involved in the most relevant interaction scenarios;
- the last section gives the reference of the installed prototypes on the TEFIS infrastructure.

The delivered prototype, at the time of writing, is installed in the TEFIS infrastructure hosting the first release of the TEFIS platform.

3. Novelties and design decisions

TEFIS gives to the user the opportunity to specify and configure the experiment he wants to run. The experiment is made up of different process steps (e.g., the build of an application and then the execution of a performance test for the same application). Each process step requires some resources to be executed. Those resources are provided by the testbeds. During the specification and configuration of the experiment the user has to define the resource types to instantiate or to choose existing resources instances to use. To allow users to instantiate or choose resources, the TEFIS platform must store the information about the resource types provided by the different testbeds and about the existing resource instances.

Testbeds, that provide resources, are heterogeneous and, usually, are managed by different administrative domains. A product that deals with heterogeneous resources that reside under different administrative domains already exists, it is TEAGLE¹. TEAGLE is a central coordination instance that holds together all TEAGLE partner test labs that are needed to provide the user with the maximum range of testing and prototyping possibilities. All the TEAGLE partner labs form a federation of testbeds. Via Teagle the user can:

- Browse resources provided by Panlab Partner labs;
- Configure and deploy the user own Private Virtual Test Lab²;
- Register new resources to be provided by the federation.

¹ <http://www.fire-teagle.org/>

² A Private Virtual Test Lab is a testbed that is composed by a number of distributed software and hardware components situated in existing testbeds across Europe. It provides you with a testing and prototyping environment that supports your very specific requirements.

As it's clear, the functionalities offered by TEAGLE are the same required by TEFIS to perform the management of the testbeds' resources. This is the reason behind the adoption of TEAGLE when implementing the TEFIS resource management.

The different components of TEAGLE are involved in the definition, configuration and execution of the user experiment in TEFIS. Before the experiment is defined (see [10] to get more details), the user must choose the resource type. The choice (as explained in the deliverable [10]) is done following a sequence of steps (domain selection, keywords selection and, finally, resource type selection from a list of resource types) during the execution of which the involved TEFIS components interact with the TEAGLE repository to retrieve the needed information. Information about the domains, the list of keywords associated to each domain and the list of resource specifications (and so the list of resource types) are retrieved from the TEAGLE repository. Once the list of the resource types is available, the experimenter can choose the types of the resources involved in his experiment to complete the experiment definition.

After the experiment is defined, the experiment configuration begins. In this phase the list of resource instances, whose type belong to the list of the resource types the user selected during the definition of the experiment, is retrieved from the TEAGLE repository. The user, then, can select the resource instances from the list or can create new resource instances. While in the first case the interaction with the TEAGLE repository takes place only to load the resource instances, in the second case an additional interaction with the TEAGLE repository is needed to store the new created resource instances. To store the resource instance, TEAGLE will store the information and the configlet³ associated with that instance. The configlet contains the actual data values that are associated with the configuration parameters of the resource specification that is instantiated.

Concerning the execution of the experiment, when the user choose to run a testrun associated to a specific testplan, the TEAGLE repository is used to check the availability of the resource instances to be used to allow or deny the testrun execution. The TEAGLE repository is also used to store the configlets associated to the testrun tasks and resources and, finally, to perform the booking of the resources needed during the execution of that testrun.

Even if the TEAGLE adoption in TEFIS provides to the user the features to allow them to define and configure the resources needed in their experiments, TEAGLE is not enough to fully match the requirements of the TEFIS resource manager component. In fact, it's noteworthy that in TEFIS there are two set of resources to manage. One is the set of resources provided by the testbeds plugged into the TEFIS platform, the other is the set of resources that the Experiment and Workflow Scheduler uses to execute the low level workflows (see [3] for more details) that represent the experiments the TEFIS users run. To perform the management of the set of resources provided by testbeds TEAGLE is used. To manage the resources needed by the Experimental and Workflow Scheduler to enact the user experiment on the testbeds' resources another component is needed since those resources are used only from the Experiment and Workflow Scheduler of the TEFIS platform and are not visible and accessed by the TEFIS users (i.e., the TEFIS user cannot create, register and/or browse those resources).

³ The configlet is a single piece of configuration and, usually, is not stand alone.

Since in the deliverable “*D4.2.1 Experiment and Workflow Scheduler prototype*” we stated we chose ProActive Scheduling tool to implement the features of the Experiment and Workflow Scheduler component of the TEFIS platform, we find suitable and convenient to use the ProActive Resourcing tool as the resource manager of TEFIS internal computational resources. In this way we assure that the two TEFIS internal components, the TEFIS scheduler and the TEFIS resource manager, are highly compatible (i.e., they belong to the same suite, ProActive Parallel Suite).

The resources managed by the ProActive Resourcing tool are computational nodes on which activities of the low level workflow (i.e., ProActive tasks) can be executed. That component provides only the management of the computational nodes (e.g., creation, booking, freeing, monitoring etc...) but does not provide the scheduling of the activities to execute on those nodes. It's the ProActive Scheduling tool that will interact with the ProActive Resourcing tool to book computational nodes and, after the booking is complete, to schedule activities (the tasks of the low level workflow) on those nodes.

4. TEFIS Resource Manager overview

The TEFIS Core Services are in charge of all the main interactions with testbeds, through the TEFIS connectors which implement the TEFIS Connector API defined in [8].

The Resource Manager is the Core Service in charge of the management of the heterogeneous resources that can be provisioned for an experiment by the testbeds plugged into the TEFIS platform (cf. the deliverable D2.1.1 [1] for a global view of the functional architecture of the TEFIS platform). The Resource Manager leverages mainly the TCI-RM API in order to list the available resources, request the instantiation of new resources, release them or access to their configuration.

The solution proposed for the Resource Manager implementation is mainly based on Teagle and on the ProActive Resourcing tool. The following section proposes a quick functional overview of these components and the place they take into the TEFIS functional architecture.

4.1. Remind about Teagle

Teagle [2] is a federation control unit that describes and controls arbitrary and heterogeneous resources across administrative domains. An overview of the Teagle framework has been presented in [6], where an assessment of the framework with respect to the TEFIS architecture and its main requirements has been presented. Here we give a quick remind about the main concepts presented there with the aim to present a mapping between the Teagle components and the TEFIS functional architecture components defined in [1].

Before focusing on Teagle architectural components, the **resource description model** used as a basis for the overall approach adopted in Teagle, is quickly reminded here, for convenience. A resource can be anything that can be controlled with software. A class of resources is described by a Resource Type, that describes common attributes, which are named, typed and can further specify a default value. A specific resource is represented as an instance of a Resource Type and it has individual values for the attributes the type specifies. Each Resource Instance is uniquely identified by an ID across a single domain and, thanks to the domain's identifier, it can be uniquely identified in a multi-domain environment as well. The model allows the definition of two kinds of relationships between resources: 'Containment' relationship, denoting a resource that "lives" inside another resource, and 'References' relationship, denoting a resource instance that "uses" another one to function.

Keeping in mind the mentioned resource model, the Teagle components depicted in Figure 1 are quickly reminded below.

In the picture the underlying Panlab control framework for testbeds is reported as well. The main component of the control framework is the **Panlab Testbed Manager (PTM)**. The picture highlights how for each administrative Domain, such as an organization providing a collection of resources, a specific Domain Manager provides the resource abstraction and exposes the interface to gain access, via the Resource Adapters, to the resources available for external use.

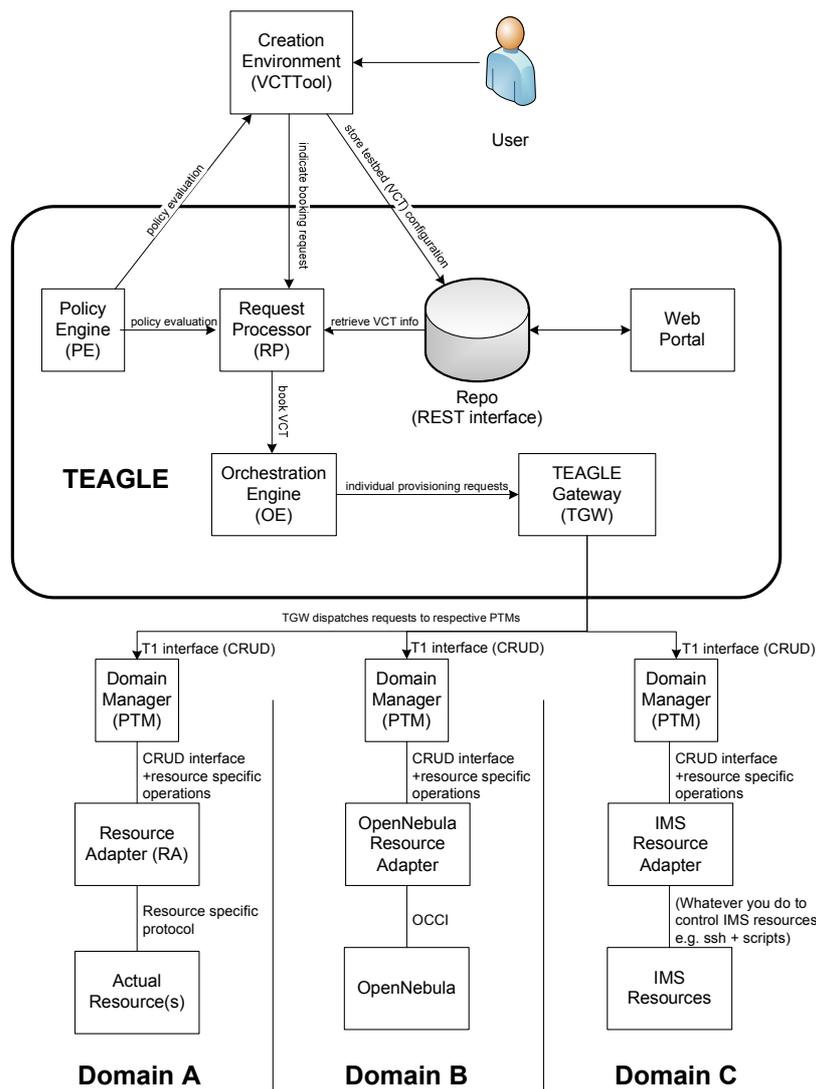


Figure 1 High level view of Teagle and the underlying control framework

The Teagle **Repository** (Repo in the picture) stores information about available resource types, the configuration of existing resource instances, data about individual users and their role, and finally Virtual Customer Testbeds (VCTs) which represent the set of configured resources needed by the final user. The repository is accessed by the other components via its HTTP-based RESTful interface.

The **Orchestration Engine** component is in charge of instantiating a VCT on top of one or more testbeds that are part of the federation. This is achieved by translating the original VCT layout model in the form of an executable orchestration script that includes the necessary sequence of service invocations to achieve the actual provisioning of the requested resources on each testbed. The service calls issued by the Orchestration Engine are redirected to the Teagle Gateway for intermediation purposes.

The Teagle **Web Portal** is a website acting as the primary contact point for customers and partners to use Teagle tools and interact with Panlab resources. A testbed operator that wants to join the Teagle infrastructure by making his resources available to Teagle customers will register his organization and the resources it provides using the functions that the portal provides. An experimenter can retrieve from there the creation environment tool (the VCT Tool) to define his own Virtual Customer Testbed (VCT).

The Teagle **VCT Tool** is a Web Start application where customers can browse resources, design, configure resources in a VCT and request resource booking.

The **Request Processor** is a backend component responsible for validating a user VCT's configuration and transforming it to an XML format suitable as input for the Orchestration Engine.

The validation process is performed by a **Policy Engine**, which enforces a set of policies stored in the repository. Different kinds of policies can be configured to restrict, for example, the access to resources based on the user role or the user's organization, or to restrict the combination of resources where a resource connection does not make sense.

The Teagle **Gateway** instantiates and maintains service endpoints for all the registered PTMs. These service endpoints are utilised by the Orchestration Engine during VCT script execution.

4.2. Teagle adoption in TEFIS

As assessed in [6], the Panlab resource model adopted in Teagle is appropriate to represent highly heterogeneous resources and generic enough to describe TEFIS testbed's resources. The TEFIS resource model has been based, then, on the Teagle resource model and the TEFIS Connector specifications have been defined taking in account this resource model.

In the following picture a mapping between the different Teagle components with the underlying control framework and the TEFIS architectural components is presented. The presented mapping aims at identifying a set of coarse grained functions common to the components involved in the mapping.

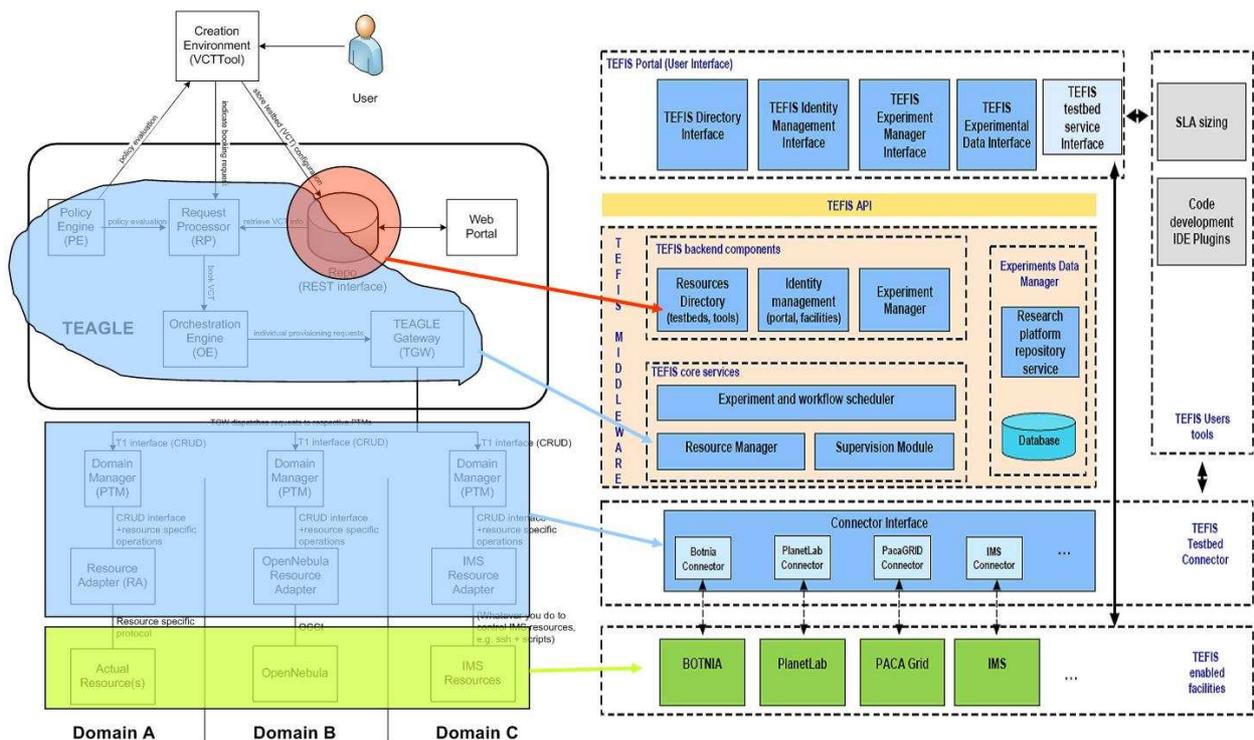


Figure 2 Mapping Teagle and PTM components on the TEFIS functional architecture

Looking at the bottom of the pictures we find a direct mapping between the TEFIS Connectors and the PTM control framework. Both of them represent an abstraction layer for the testbed resources (in green in the picture), even if the abstraction layer provided by the Panlab framework is confined to resource management and a support for experiment execution or for data transfer for example are not in the scope of the Panlab components.

On top of the abstraction layer we find on the left the Teagle components, and on the right the TEFIS middleware functional blocks. As in the case of Teagle the Repository stores the resource descriptions and configuration of existing resource instances etc, in TEFIS the Resources Directory aims to store information about the tools, facilities, and any resources provided by the heterogeneous testbeds. As long as the resource description model is the same, there are many similarities between the two components, from a functional point of view. That's why the picture highlights a direct mapping between the Teagle Repository and the TEFIS Resource Directory.

The other Teagle components, except the VCT Tool and the Web Portal, as mentioned in the Teagle overview, provide together the following main functions:

- Resource provisioning;
- Resource control;
- Resource configuration support;

In TEFIS these features should be provided by the TEFIS Resource Manager component, in order to support the users in the definition and execution of their experiments on the testbed resources. This does not mean that the TEFIS Resource Manager should not provide any other additional feature (such as, for example, a support for internal computing resource provisioning or a stronger support for resource reservation) but the identified mapping suggests that these Teagle components may represent a good base for the implementation of the TEFIS Resource Manager solution.

To complete the discussion, the last two Teagle components reported in the picture are the VCT Tool and the Web Portal. The former is the graphical tool used by end users in the virtual customer testbed definition and booking. It is mainly mapped on a presentation layer entity, but part of its logic could be mapped also on back end components acting as client of the Teagle repository and the Teagle request processor for example. An exact mapping of this tool is not in the focus of the Core Services definition, and the same applies to the Web Portal as well.

4.3. The ProActive Resourcing tool

The ProActive Resourcing tool [7] is a client/server application that aggregates, monitors, controls computing resources and made them available to external clients. A computing resource is represented by a ProActive node and it can be seen as an entry point to the physical resource where the node runs on. The server selects and allocates nodes according to user requests. It restricts the access to these nodes to other users during their allocation, monitors nodes states and provides up-to-date information about resources utilization. Once the suitable nodes are selected and given to the users, they access them directly and at the completion of the computation, the resources are released thanks to a

notification to the server. A client can be an end user or an entity requesting computing nodes. The main client is the ProActive Scheduling tool, which requests computing nodes to execute jobs on. Clients with administrative rights may trigger automatic nodes deployment, manually add and remove nodes to the server.

The picture below, in Figure 3 , reports a synoptic view of the ProActive Resourcing tool architecture.

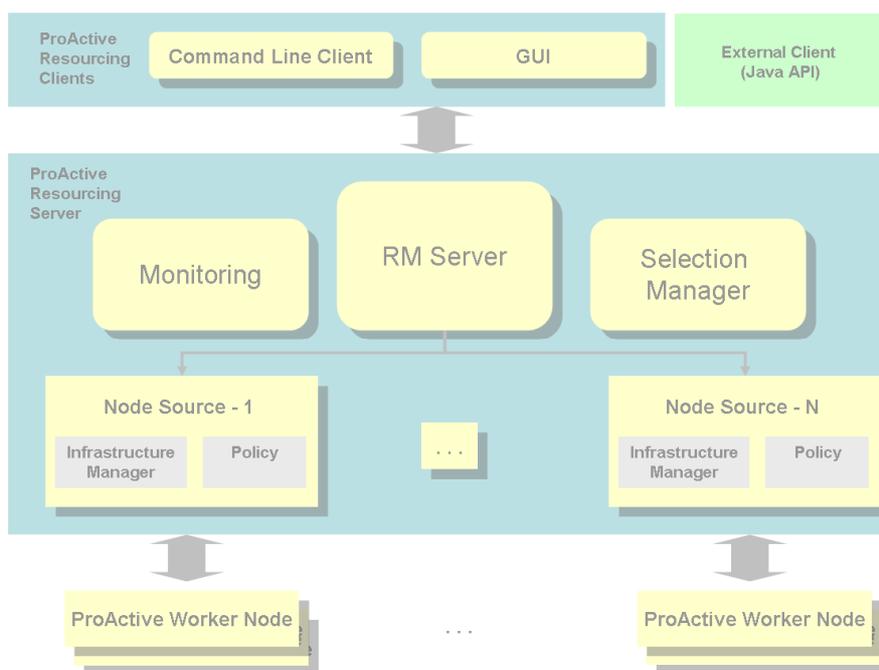


Figure 3 ProActive Resourcing tool architecture

As mentioned above, the ProActive Resourcing tool is a client/server application. **Clients** are remote and may administrate or monitor the server state by means of a Command Line or a Graphical User Interface (GUI). They also may request nodes for computations using Java API. The communication between clients, server and computing nodes are performed by ProActive. Administrative actions, like node deployment, are delegated by the resourcing tool core (**RM Server**, in the picture) to node sources and then to infrastructure managers.

A **Node Source** represents a set of computing nodes and is characterized by an Infrastructure Manager and a Policy. The **Infrastructure Manager** is responsible for node deployment to a particular infrastructure with a given protocol or mechanism. The **Policy** defines rules and limitations of node source utilization, in terms of users allowed to use it or not and in terms of deployment time and so on.

User requests of getting nodes are processed by **Selection Managers** who are able to know whether a node is suitable or not. Once a user has obtained nodes, he contacts them directly without involving the resource manager.

A **Monitoring** module tracks resources and their states and made this information available for external clients.

4.4. The ProActive Resourcing tool in TEFIS

As explained in [3], the ProActive Scheduling tool is used as a base for the implementation of the TEFIS Experiment and workflow Scheduler. It leverages the ProActive Resourcing tool to acquire the computing resources where to execute tasks on.

In TEFIS the workflows enacted by the Scheduler are the experiments defined by the TEFIS users, which involve the heterogeneous resources provisioned by the testbeds. Those experiments are internally represented as ProActive jobs, where the tasks composing each job represent the interactions with the testbeds via their exposed Connectors.

While the testbeds provision the resources required by the experimenters to perform their experiments, the ProActive Resourcing tool provisions the computing resources, internal to the TEFIS platform, needed by the TEFIS Scheduler in order to orchestrate the user experiments on.

For each testbed a specific Node Source is configured in the ProActive Resourcing tool. When the Scheduler needs to interact with a given testbed to execute a process step of the user experiment, it can acquire a computing node from the right Node Source, which is configured with the appropriate code and configuration parameter values needed to properly interact with the testbed's connector. In this way any specific configuration needed for the interaction of a given testbed is hidden in the ProActive Resourcing configuration and the ProActive jobs corresponding to the user experiments can be kept cleaner and simpler.

The picture below, Figure 4 , shows what just explained.

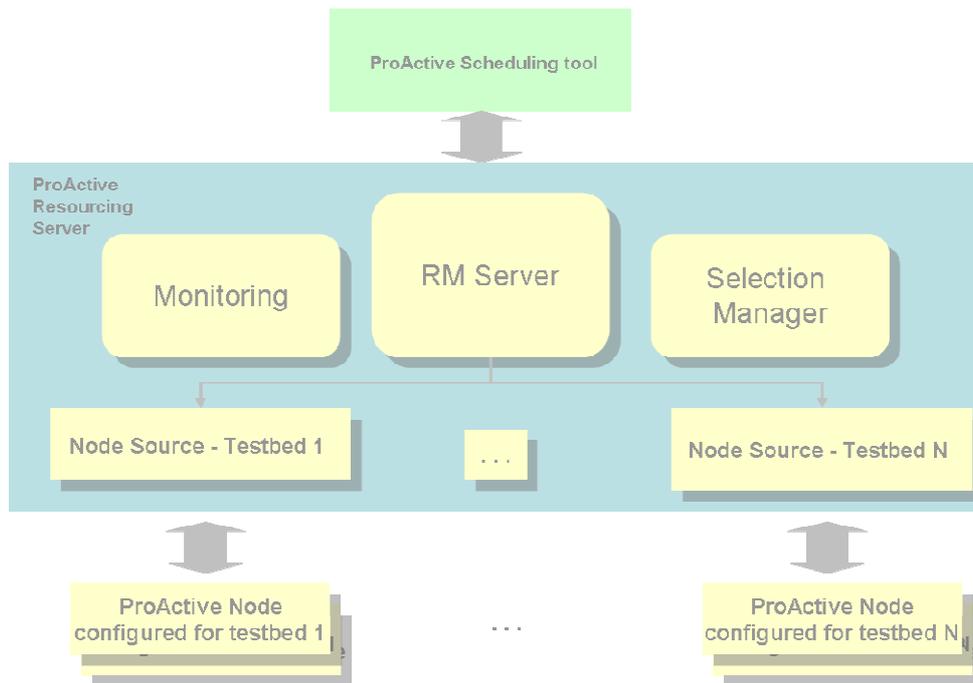


Figure 4 ProActive Resourcing tool in TEFIS

4.5. The main usage scenarios

The TEFIS Resource Manager solution is based on Teagle, as said above, for what concerns the management of the resources provided by external testbeds and involved in the user experiments. The TEFIS Resource Manager solution includes also the ProActive Resourcing tool, for what concerns the provisioning of computing resources internal to the TEFIS platform itself as required by the TEFIS Scheduler that needs to interact with external testbed connectors. The two most significant usage scenarios of interaction between the involved entities are presented below.

4.5.1. The testbeds' resource provisioning

The main flow for resource provisioning starts from the resource configuration design phase made by the experimenter on the Portal. Once this design phase is completed, the experimenter has identified a set of resources from one or several testbeds, each one with its own configuration. This set of information represents the 'resources configuration' required for the experiment execution.

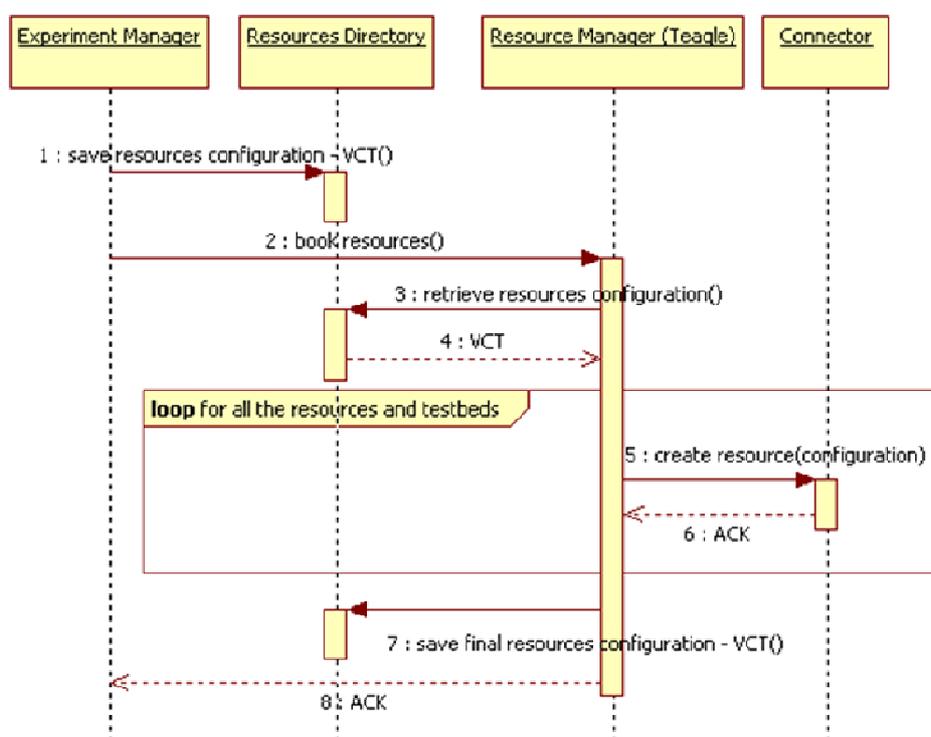


Figure 5 Testbed resource provisioning scenario

This 'resources configuration' is stored, by the Experiment Manager, in the Resources Directory (based on the Teagle repository) in the form of a VCT file. In order to start the experiment execution, these resources have to be provisioned. In the step 2: book resources(), the Experiment Manager asks the Resource Manager (based on Teagle) to provision the required resources according to the VCT previously defined. The Resource Manager, then, retrieves the VCT file and according to this file, for all the resources it contacts the Connector of the appropriate testbed, in order to request the creation of the resource with the required configuration. Once the provisioning process is finished, the VCT file is updated with additional information (like Resource ID, provisioning state) characterizing each resource instance provisioned by the testbed.

4.5.2. The experiment execution

The experiment execution starts when the flow associated to the experiment test run is submitted to the TEFIS Workflow Scheduler by the Experiment Manager component (cf. [5], section “Experiment Manager – Other Building blocks”, for more details about this interaction). The Experiment Manager receives the identifier of the submitted job (the jobId), to be able to monitor its execution.

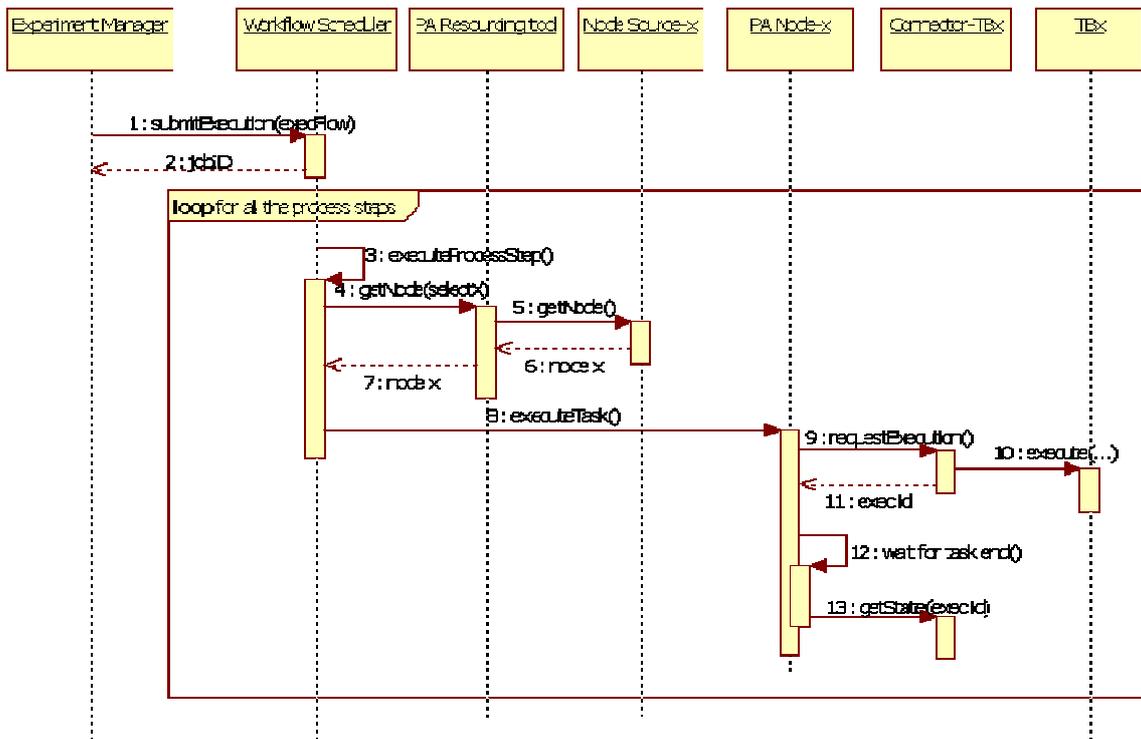


Figure 6 Provisioning of internal computing resources for experiment execution

The TEFIS Workflow Scheduler schedules the execution of each process step according to the defined flow structure. In order to execute a process step, the Scheduler asks the ProActive Resourcing tool (PA Resourcing tool, in the diagram) for a computing node to perform the task execution on the testbed involved in the process step (Testbed X, TBx). The PA Resourcing tool gets a node from the Node Source x and returns a ProActive Node (PANode-x) to the Scheduler. The Scheduler delegates the PANode-x, that is pre-configured with the right properties (such as the testbed endpoint, specific testbed configuration properties like the more appropriate polling time for the execution status check, etc.), to interact with the connector exposed by the testbed x, Connector-TBx. The execution logic is contained in the task template for the testbed x (cf. [3] TEFIS_D4.2.1_Scheduling_prototype.doc) available in the PANode-x environment. An execution id, exec id, is returned in order to let the PANode-x monitor the process execution progress.

5. The installed Prototype

As stated above, the TEFIS Resource Manager prototype includes: the Teagle Request Processor, the Teagle Orchestration Engine, the Teagle Gateway, the Teagle Policy Engine and the ProActive Resourcing tool.

In the settings of the current installation of the prototype the Teagle Gateway is not needed so it is not included in the currently running platform. The same happens for the Teagle Policy Engine since in the scenarios considered for the first version of the TEFIS prototype any specific policy definition has been planned.

The TEFIS Resource Manager prototype is currently installed in the TEFIS infrastructure operated by INRIA (cf [9] for more details on the TEFIS infrastructure). All the involved components are available at the following URLs.

Teagle Request Processor:

<http://tefis1.inria.fr/reqproc>

Teagle Orchestration Engine (its web interface):

<http://tefis1.inria.fr/teagle-site>

ProActive Resourcing tool:

`rmi://tefis2.inria.fr:1099/`

Anyway, since all the components included in the TEFIS Resource Manager prototype are not supposed to be directly used by end users but they are accessed by other TEFIS components, then it's not easy to have a quick view of them.

An easier way to test the installed Teagle components is to use the VCT Tool to browse the repository and to define and request the booking of a new virtual customer testbed. The VCT Tool, to be used testing purposes, is available at

<http://tefis1.inria.fr/web-start-devel/vct.jnlp>

(the access credentials to use are the ones used to access the TEFIS portal).

In order to check the installation of the ProActive Resourcing tool, one of the ProActive Resourcing clients can be used. The version 3.0.x of the clients available for download at

<http://www.activeeon.com/community-downloads>, are actually compatible with the installed version on the TEFIS infrastructure.

Dependencies: the Teagle Request Processor and the Orchestration Engine leverage the TEFIS Resource Directory, that is based on the Teagle Repository, available at:

<http://tefis1.inria.fr/repository> (the Web interface)

<http://tefis1.inria.fr/repository/rest> (the REST interface)

The Resource Directory deliverable is part of the Work Package 3, so refer to WP3 deliverables for more information on this component.

6. Conclusion

In this document we have analyzed the requirements and features an eventual existing component must match to be used to implement the Resource Manager in the TEFIS platform. We stated that the adoption of both TEAGLE and the ProActive Resourcing tool contained in the ProActive Parallel Suite is needed to provide the resource management features of the TEFIS platform.

To validate the proposed solution we analyzed how TEAGLE maps on the TEFIS functional components and how it is used to perform resource provisioning, resource control and resource configuration support. Concerning the ProActive Resourcing tool, we described how it manages and provides computational resources to the Experiment and Workflow Manager of TEFIS to let it to enact an experiment on the resources provided by the testbeds plugged into TEFIS.

We outline two utilization scenarios to show how TEAGLE and the ProActive Resourcing tool are used during the testbed resource provisioning and the experiment execution respectively.

The last part of the document is dedicated to the description of the prototype carried out, how it is installed on the INRIA machines and how to get the access to the installed TEAGLE components and to the ProActive Resourcing tool.

References

- [1] D2.1.1: Global architecture and overall design
- [2] Teagle portal website: <http://www.fire-teagle.org>
- [3] TEFIS_D4.2.1_Scheduling_prototype.doc
- [4] D3 3 User tools implementation.doc
- [5] D3 2 Building blocks integrated into TEFIS portal.doc
- [6] D3.1.1_Teagle_assessment_and_TEFIS_portal_specifications
- [7] http://proactive.inria.fr/release-doc/Resourcing/single_html/ProActiveResourceManagerManual.html
- [8] D5.1.1 - Generic and Specific Connector specifications v1.0.doc
- [9] TEFIS_D2.3.1_Integration_Plan.doc
- [10] D3.1.2 TEFIS Portal implementation