Abstract

The objective of the Panlab project is to define a general framework for the interconnection of independent testbeds and laboratories and to facilitate the establishment of a Pan-European laboratory. The concept of such a Panlab is based on the federation of distributed interconnected testbeds that provide the required facilities for a broad range of testing capabilities for technologies, product interoperability and telecommunications services.

This document investigates the means to set-up and to control a federation of testbeds from a technical point of view and describes processes that allow for an automated, within the context of a certain contract, access of remote assets and resources.
Disclaimer

This document contains material, which is the copyright of certain Panlab consortium parties, and may not be reproduced or copied without permission.

All Panlab consortium parties have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the Panlab consortium as a whole, nor a certain party of the Panlab consortium warrants that the information contained in this document is capable of use, or that use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information.

Impressum

[Full project title] Pan-European Laboratory for Next Generation Networks and Services
[Short project title] Panlab
[Number and title of work-package] WP2: Approach to technical infrastructure
[Document title] Approach to technical infrastructure
[Editor: Name, company] Fabricio Gouveia and Sebastian Wahle, Fraunhofer FOKUS
[Work-package leader: Name, company] Lourdes Calvo, Telefonica I+D
[Estimation of PM spent on the Deliverable] 18

Copyright notice
© 2008 Participants in project Panlab
## List of authors

<table>
<thead>
<tr>
<th>Company</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alcatel-Lucent</td>
<td>Olivier Audouin</td>
</tr>
<tr>
<td>Dimes</td>
<td>Susanna Avéssta</td>
</tr>
<tr>
<td>Eurescom</td>
<td>Anastasius Gavras, Halid Hrasnica</td>
</tr>
<tr>
<td>Fokus</td>
<td>Thomas Magedanz, Fabricio Gouveia, Sebastian Wahle</td>
</tr>
<tr>
<td>Synchromedia</td>
<td>Mohamed Cheriet, Mathieu Lemay</td>
</tr>
<tr>
<td>Italtel</td>
<td>Giuseppe Monteleone</td>
</tr>
<tr>
<td>Nokia</td>
<td>Juha Saarnio</td>
</tr>
<tr>
<td>Telefonica I+D</td>
<td>Lourdes Calvo</td>
</tr>
<tr>
<td>Thomson</td>
<td>Dennis Mischler</td>
</tr>
<tr>
<td>France Télécom</td>
<td>Jean-Claude Imbeaux</td>
</tr>
<tr>
<td>RAD</td>
<td>Yuri Gittik</td>
</tr>
</tbody>
</table>
## Table of Contents

List of authors ................................................................................................................................. 3  
Table of Contents ............................................................................................................................. 4  
List of figures ...................................................................................................................................... 5  
List of tables ....................................................................................................................................... 5  
Definitions of terms ........................................................................................................................... 6  
1 Executive summary ........................................................................................................................ 7  
2 Introduction ...................................................................................................................................... 8  
3 Panlab Technical Infrastructure Overview and Definitions .......................................................... 9  
   3.1 General Use Case ....................................................................................................................... 9  
   3.2 Overview and Description of the General Panlab Testbed Federation Concept .................. 9  
   3.3 Panlab Infrastructure and Roles ............................................................................................. 10  
      3.3.1 The Panlab Office ............................................................................................................... 11  
      3.3.2 The Panlab Partner ............................................................................................................ 11  
      3.3.3 The Panlab Customer ......................................................................................................... 12  
   3.4 The Panlab Evolution Path ....................................................................................................... 12  
      3.4.1 Centralized Approach ....................................................................................................... 12  
      3.4.2 Manual Configuration Approach ..................................................................................... 13  
      3.4.3 On-demand Configuration Approach ............................................................................... 13  
4 Functional Infrastructure Components .......................................................................................... 15  
   4.1 Establishing Connectivity ......................................................................................................... 15  
      4.1.1 Telecom Layered Approach .............................................................................................. 17  
      4.1.2 Utilization of IMS as a Federation Tool ............................................................................. 18  
   4.2 Teagle ..................................................................................................................................... 19  
      4.2.1 Teagle Operation, Plans and Roadmap ............................................................................. 19  
      4.2.2 Data Needed From the Users ............................................................................................ 20  
      4.2.3 Teagle Model ...................................................................................................................... 21  
   4.3 Panlab Repository ..................................................................................................................... 22  
      4.3.1 Partner Testbed Repository ............................................................................................... 23  
      4.3.2 Testing Results Repository .................................................................................................. 27  
      4.3.3 Panlab Repository Data Model .......................................................................................... 30  
4.4 Virtualized Resources .................................................................................................................. 31  
   4.4.1 Resource Allocation .............................................................................................................. 31  
   4.4.2 Access Right Management .................................................................................................. 32  
   4.4.3 Scheduling ............................................................................................................................ 32  
   4.4.4 Resource Economics ............................................................................................................ 33  
   4.4.5 Management Tools .............................................................................................................. 33  
   4.4.6 Complementary Supporting Tools ...................................................................................... 34  
4.5 Existing Technologies .................................................................................................................... 34  
   4.5.1 Teagle Technologies ............................................................................................................. 34  
   4.5.2 Middleware Resources Technologies .................................................................................. 35  
5 Conclusion ...................................................................................................................................... 37  
References .......................................................................................................................................... 39
List of figures
Figure 1: The general Panlab Testbed Federation Concept ................................................................. 10
Figure 2: Basic Panlab Architecture and Components ................................................................. 11
Figure 3: First Phase - The Centralized Approach ........................................................................... 12
Figure 4: Second Phase - The Manual Configuration Approach ................................................. 13
Figure 5: Third Phase - On-Demand Configuration Approach .................................................... 14
Figure 6: General case of connectivity between testbeds ............................................................ 15
Figure 7: Controlling Connectivity ............................................................................................... 17
Figure 8: Separation of Layers .................................................................................................... 17
Figure 9: Teagle Model ............................................................................................................... 22
Figure 10: Panlab Repository Data Model (Simplified) ................................................................. 31
Figure 11 Bottom up Approach to Virtualization .................................................................... 32
Figure 12: Virtualized Resources Workflow ............................................................................. 34
Figure 13: IaaS Framework Capabilities ..................................................................................... 36

List of tables
Table 1 – Hardware Resources .................................................................................................. 29
Table 2 – Software Resources .................................................................................................. 29
Table 3 – Productivity and Support Tools ................................................................................ 30
Table 4 – Test Environment (Application Dependent) ............................................................ 30
Table 5 – Teagle Technologies .................................................................................................. 35
Table 6 – Middleware Resources Technologies ......................................................................... 35
Definitions of terms

**Infrastructure as a Service (IaaS)** – is a concept where users buy infrastructure resources (like servers, components, software) as an outsourced service.

**Pan-European Laboratory (Panlab)** – a concept of enabling Pan-European testing services for a demo/trial and evaluation of new technologies, system solutions and telecom service concepts

**Panlab Activities** – actions and processes that are launched and carried out within the Panlab framework.

**Panlab Partner** – an entity that participates in Panlab activities by providing testing or any other supporting services.

**Panlab Customer** – an entity that uses (consumes) any service provided by the Panlab. Customers typically carry out R&D projects, implement new technologies or products, or introduce new telecom services.

**Panlab Community** is composed of Partners and Customers, along with other relevant participants from the industry, research and supra-governmental institutions, for example, EU organizations.

**Panlab Mechanisms** – a set of rules and procedures that enables effective collaboration between Panlab partners as well as cooperation with Panlab customers.

**Panlab Organization** – organization applying the concept of the Pan-European Laboratory.

**Panlab Office** – coordination and support entity of the Panlab organization.

**Resource Consumer** – Panlab customer or partner that is consuming resources that have been shared to the Panlab Community.

**Resource Provider** – Panlab partner providing resources to be shared with the Panlab Community.

**System Under Test (SUT)** - refers to the system that is being tested. In the context of Panlab, the SUT might need to be deployed at a Panlab Partner testbed.

**Virtual Private Testbed (VPT)** – is the resulting testbed configuration after orchestration of Panlab resources distributed in the federation based on the user testing requirements.
1 Executive summary

The objective of the Panlab project is to define a framework for the interconnection of independent testbeds and laboratories so as to facilitate the establishment of a Pan-European Laboratory. The concept of such a Pan-European Laboratory is based on the federation of distributed test laboratories and testbeds that are interconnected and provide access to required platforms, networks and services for broad activities, to demo/trial and evaluate new technologies, system solutions and telecom service concepts.

This deliverable focuses on the technical infrastructure necessary for implementing and maintaining the Panlab concepts. It is intended to be used as a generic reference model for the technical infrastructure rather than a technical specification.

From a general point of view, the Panlab testbed federation concept is composed by multiple test laboratories and testbeds interconnected through a Virtual Private Network (VPN), as well as the Panlab Office, all empowered by the Teagle (search and orchestration engine).

There are different entities involved in Panlab’s infrastructure, each one with a different role: the Panlab Office, the Panlab Partner and the Panlab Customer.

- Panlab Partner: the entity responsible for providing test and/or support facilities.
- Panlab Customer: the entity that uses the resources provided by Panlab infrastructure to test new technologies, system solutions and telecom service concepts.
- Panlab Office: the entity in charge of coordinating and supporting the Panlab organization as well as managing technical, business and marketing activities.

The Panlab infrastructure will be deployed in stages, evolving from a simple to more advanced configurations. The Panlab evolution path will go from a manual phase (where testbed descriptions are manually entered by each testbed representative) to a total automated solution allowing both Partners and Customers to set-up and configure the required testbeds on demand.

The technical infrastructure model involves the use of the following components to achieve connectivity and uniformity among Partners, Customers and the Panlab Office:

- Connectivity component: It can be considered as VPN-based tools to enable the interconnection between Panlab partners. These tools may be presented as resources themselves, hiding the technical complexities and allowing Panlab partners to have multiple overlay networks for testbeds on “as required” basis.
- Teagle: It is the federation search and orchestration engine. Teagle will provide a smart interface for Customers making available necessary information and means for serving their needs. That is, a web-based service to fulfil Customer’s needs related to Panlab offerings. This interface should provide access to a database containing information on testbeds and exploit a repository of testbed functionality descriptions. This interface will also be able to combine different testing components into a functional entity making it possible to meet various testing requirements that a single testbed may not be able to do.
- Panlab repository: It is a component to maintain the description of the Panlab testbeds, detailing the testing services and caching resources offered by each Panlab Partner to the Panlab community. It will also be able to maintain the tests’ results, configurations, etc. for future tests.
- Resource virtualization: Virtualized resources are intended to be used both by Teagle or other testbed administrators in order to access hardware or software devices that need to be shared, reserved, discovered, and controlled. The advantage of virtualizing the resources is that each testbed remains the owner of its physical infrastructure giving limited access to the Panlab Office.

The technical infrastructure offered by Panlab is not static; it is a dynamic structure that will evolve, including new platforms, services and systems offered by previously connected or new laboratories.
2 Introduction

The concept of the Panlab is based on the federation of distributed test laboratories and testbeds that are interconnected and provide access to required platforms, networks and services for a broad range of testing types. The realization of such infrastructure demands investigation of available technologies that can be reused for the implementation of this federated framework, as well as new concepts and technologies.

This deliverable describes the Panlab approach for the technical infrastructure. The intention of this document is to be used as a reference and is not meant to be a technical specification.

Section 3 gives a high level technical infrastructure overview and defines the most important terms. It begins with the description of a general scenario to facilitate the understanding of the general concept. The general concept of federation is accompanied by a definition of the infrastructure components and the different roles, namely Panlab Customer, Partner and Office. The infrastructure shall be dynamic and adaptive where Panlab Customers and Panlab Partners interact through a central control office for connecting to the Panlab Partner laboratories. Also, via the central office new partner laboratories can be added to the Panlab federation. Furthermore, the aim of section 3 is to describe the Panlab evolution path, which starts from a centralized and manually configured approach towards a more robust, distributed and automatically configured one.

Section 4 is the main part of the document and defines the infrastructure components and procedures for establishing connectivity. Also, a central search and configuration tool named Teagle is discussed. In addition, the Panlab Repository, storing Panlab Partner testbed details as well as obtained results, is introduced in this section. Generally, the Next Generation Network (NGN) layered approach has been taken as a basis for decoupling applications, signalling control and service support functions. In order to deploy a testbed federation reusing these concepts, functional blocks are defined. The main functional components such as the searching and orchestration engine Teagle, the Panlab Repository and their respective data models and functionality are described.

The resulting orchestrated infrastructure is called a virtual testbed. This virtual testbed offers resources to the Panlab Customer, matching their requirements on the requested testing and other Panlab usage interests. Supporting tools, like scheduling and managing tools, necessary for the federation control and operation, are also described.

Finally, available technologies that could be used to implement Teagle and middleware resources are analyzed.
3 Panlab Technical Infrastructure Overview and Definitions

3.1 General Use Case

A telecommunication operator is currently developing an innovative application for converged networks. This application allows users to watch TV and Video on Demand (VoD) from different devices such as PDAs, smartphones, notebooks, PCs, etc. With market analysis, this operator discovered a trend on offering the same service using an IP Multimedia Subsystem (IMS) infrastructure. The IMS offers converged functionalities, for instance charge the video stream service differently from other media traffic (e.g. video conferencing). As this operator does not have an IMS infrastructure, it uses Panlab services for finding an appropriate IMS testbed that can fulfil testing requirements on this view. With the testing possibility Panlab offers, the telecommunication operator can make performance tests and evaluate better the risks involved on deploying this new architecture (e.g. cost/benefit, interoperability between the service and the different equipment available, etc.).

The Teagle tool provided by Panlab helps the telecom operator (Panlab Customer) to search in the Panlab federation for specific testbeds, shared resources and possible testing scenarios. This tool is described in section 4.2. The testing configurations are based on the very specific requirements resulting from concrete use cases (for example different access networks, IMS support, etc.).

3.2 Overview and Description of the General Panlab Testbed Federation Concept

As depicted by the basic use case above, the general idea of Panlab is to make use of the federation concept for distributed test laboratories that are interconnected and provide access to local platforms, networks and services. The coordination of resources and access to the laboratory services shall be controlled by a centralized entity – the Panlab Office (as described in section 3.3.1).

The technical infrastructure offered by Panlab is not static, but rather dynamic and adaptive; it will be extended, adjusted and improved by new available platforms, services and systems that can be offered by the connected laboratories or new laboratories joining the federation. The entire mechanism, the rules and procedures of how to achieve the effective testing collaboration, are currently being developed in the Panlab project. In this document we focus on the technical infrastructure.

Panlab relies on existing and future testing facilities and targets collaboration mechanisms between these. The federative approach to distributed testing has the benefit of flexibility, but at the cost of the management of the distributed environment.

The ultimate goal of Panlab is the establishment of the foundation of a future operational and long-term self-sustainable Pan-European laboratory. This includes a business model on the continuation of Panlab as an independent entity, targeting customers, such as European or national collaborative projects and the industry, during its pre-competitive research and development phase.

As shown in Figure 1, the general setup is composed by the following functional entities: The Panlab Partner testbed, the Panlab Office and Teagle. The Panlab Customer role (not shown in Figure 1), is described in section 3.3.3. The testbeds are geographically distributed across Europe and are interconnected by an interconnection mechanism as described in section 4.1. The Panlab Office has many functions such as the coordination of resources, offering legal advice and documents, as well as maintenance of the website and the Teagle tool.

The Teagle tool as a whole consists of a collection of web services that provide the means for a customer to express the testing needs and to get feedback on where, how, and when testing can take place. It enables finding a suitable site for one’s testing needs. It does this through a database of the partner testbeds. The objectives of Teagle in its fully operational form are to manage the complete set-up, necessary resource reservations, and needed interconnections of any foreseen testing needs. Details on Teagle operation are described in section 4.2.
The long-term target for Panlab includes the automated testbed setup and configuration (spanning multiple underlying physical testbeds) in order to realize a “virtual testbed” as an environment to run the selected test suites. This virtual test environment makes use of any number of physical testbeds offering different types of technologies required to set up the desired environment.

![Figure 1: The general Panlab Testbed Federation Concept](image)

The Teagle tool has different interfaces such as a customer interface that is used for lookup of Panlab capabilities, resources and testing configurations as well as a Panlab Partner interface.

### 3.3 Panlab Infrastructure and Roles

The Panlab infrastructure is based on Figure 1 where different distributed test laboratories and testbeds are interconnected for providing services to customers of the infrastructure. The different interfaces that provide access, networks and services are depicted in Figure 2. The members of the Panlab Community are connected to the infrastructure using Virtual Private Network (VPN) connections to communicate with all parties. These VPN connections are conducted based on trusted relationships among Panlab Partners defined by the operational issues [3].

The role of each Panlab entity and their respective interfaces are described in the next subsections.
3.3.1 The Panlab Office

The Panlab Office (office of the Pan-European Laboratory) is the entity that realizes a component brokering service for the test facilities, coordinating and supporting the Panlab organization. The Panlab Office is responsible for coordinating the provision of the testing services, partly also by using various tools and, when possible, web interfaces, but also directly coordinating the Panlab test-sites and communication path between them and customers. The Panlab Office develops and maintains the web interface and in this way offers testing services of the Pan-European Laboratory. Besides coordination and operational activities, the Panlab Office is responsible for the technical and business development of the Pan-European Laboratory as well as marketing and public relations.

The Panlab Office has two operational interfaces: one connecting to the Customer (called Teagle interface) and the other connecting to the Panlab Partners. The Teagle tool is located at the Panlab Office, where it can be controlled.

The Teagle interface is mainly used for providing the user the web services developed in Panlab. It provides user interaction during the search transferring customers’ requests and providing them with results. Through a user-friendly graphical interface users can search (either by clicking or defining advanced search) within different categories from a set of options that are checked directly from the information available in the Teagle database.

The other interface is based on Virtual Private Network (VPN) connections and is used for providing test partners the necessary tools, Teagle configurations, testing results, etc., which are used by the tests and Virtual Private Testbeds (VPT) configurations.

3.3.2 The Panlab Partner

The Panlab Partner is an entity that participates in Panlab activities by providing testing or any other supporting services. The entities are testbeds offering testing services, which will be used by customers or other partners in order to do different kinds of tests in cooperation with other industry partners or research and academic institutions.

The Panlab Partners are connected to the Panlab Office for offering their capabilities and services and to other partners or customers, for executing tests between them. Testing and other cooperation traffic is transported through secure VPN connections.

Figure 2: Basic Panlab Architecture and Components
3.3.3 The Panlab Customer

The Panlab Customer is an entity that uses any service provided by the Panlab. Customers typically carry out R&D projects, implement and evaluate new technologies or products, or introduce and evaluate new telecom services, benefiting from the Panlab infrastructure.

They use the Teagle interface and a System Under Test (SUT) interface for obtaining the Panlab functionality. The SUT interface is used by customers to connect directly to the testbeds for executing the agreed testing/configuration or deploy systems.

3.4 The Panlab Evolution Path

This section provides a high level overview of the possible evolution of the proposed architecture. While at the beginning most functions are to be carried out manually it is expected that, while the concept evolves, more and more function are going to be automated. The Panlab Partner testbeds for example are expected to provide a description of what their offerings are (in terms of supported interfaces, technologies, etc.). This description (testbed metadata) is expected to be entered manually for early deployments of the Panlab infrastructure. However, the long-term vision is that this information is generated by the infrastructure itself in order to automate search and setup processes. In the following subsections we discuss the expected phases of the Panlab evolution.

3.4.1 Centralized Approach

In this first phase, the testbed metadata held in the Panlab Repository (as described in section 4.3) is entered manually by the partner testbed representatives such as an administrator. In order to achieve this, Teagle should offer an online form where the testbed representatives can enter the relevant data describing the testbed and its resources. Panlab Customers may search the Panlab Repository using Teagle to find suitable resources needed for doing their tests (Figure 3).

![Figure 3: First Phase - The Centralized Approach](image)

Each testbed resource is directly entered in the Panlab Repository manually by the Panlab Partner administrator. This centralized approach provides a mean to quickly federate the different testbeds. A user may then use Teagle to search for needed testbed resources. Based on these search results from Teagle the user will later contact the Panlab office which should then kick-off the setup and configuration process manually with the appropriate testbed representatives.
3.4.2 Manual Configuration Approach

In the second phase, the Panlab Partner testbeds will advertise themselves the resources they want to share with the different virtual organization partners by using a specialized middleware. Teagle will then search both static information kept in the repository as well as the different virtualized resources representing available infrastructure components or services (Figure 4). This approach, called Infrastructure as a Service (IaaS), allows other Panlab partners to make “allowed” configuration changes on the equipment by themselves, thus removing the need for having all testbed representative intervention in the initial setup and configuration of the testbed resources.

![Figure 4: Second Phase – The Manual Configuration Approach](image)

The principal enhancement of this second step approach is that a single Panlab Partner may do the configuration required on the different devices that are offered using Infrastructure as a Service resource. Strict policies are enforced by the virtual instances that restrict efficiently the resource consumer’s scope of action therefore, allowing a single consumer to put the complete testbed configuration in place by interacting with the Resource Management System interface.

3.4.3 On-demand Configuration Approach

In the final Panlab phase, Teagle will directly interact with the virtualized resources in order to establish the testbed configurations on-demand based on the customer descriptions, access rights and resource policies (Figure 5). Please note that Figure 5 abstracts from different administrative domains and that policy enforcement is needed. The Modular Gateways, as introduced in section 4.1, act as policy enforcement and signalling conversion points.
Figure 5: Third Phase - On-Demand Configuration Approach

This third phase enables Teagle to offer an advanced search tool, which allows Panlab Customers to search for required technology. As the information on the underlying testing resources is gathered automatically, customers can rely on data that is always up to date. The configuration of the testbed can be done directly within Teagle by the customer or partners themselves. However, for more complex configuration, a resource consumer may directly access the resources in order to manually do configuration changes as in phase 2 if it is required.
4 Functional Infrastructure Components

4.1 Establishing Connectivity

In order to establish connectivity, a Virtual Private Network must be set-up using open VPN technologies like OpenVPN [4] or Tinc [5]. These tools may be wrapped as resources themselves, hiding the complexities and allowing Panlab partners to dynamically provision multiple overlay networks for testbeds on “as required” basis. A VPN being a stateful entity should be modelled as a stateful web service via Web Service Resource Framework (WSRF) standards, using the factory-resource software pattern. If desired, once the connectivity is established, notifications can be sent out using Web Service-Notifications to Teagle or another service, initiating the tasks to be performed by the testbed. The configuration of the interconnection resources (in addition to testbed resources) could also be integrated into the scope of the Infrastructure as a Service approach described in the previous section.

As illustrated in Figure 2, testing traffic must be able to flow between Panlab partners and thus controlled connectivity with pre-defined properties must be established between two or more remote sites. An abstract view on such a general case is illustrated in Figure 6 below.

![Diagram showing general case of connectivity between testbeds](image-url)

**Figure 6: General case of connectivity between testbeds**

The established connection between the different testbeds must serve a specific objective, i.e. it must serve the interactions between (i) applications, (ii) signalling control and service support, or (iii) user data transport. Connections can be requested to serve one or more of these objectives.

- Connections with the objective to serve applications will permit the testing of interactions between applications such as messaging, streaming or data services.
- Connections with the objective to serve signalling control and service support will allow the provisioning of dedicated functions enabling the interactions of signalling and control for optimization of the interactions between different network and service control systems (for example centralized versus peer-to-peer control).
- Connections with the objective to serve user data transport will allow the interconnection of different network technologies in the access segment and border functions of the core network.
For the purpose of a testbed federation, the connections that need to be established must at least satisfy specific reliability, quality of service (QoS) and security properties. In particular, when the tests involve performance measurements, the impact of the performance of the interconnection resource on the end-to-end performance has to be fully known and controlled. Such properties need to be specified explicitly by the testing service user or implicitly by the type of configuration requested. The selection of testbeds to perform a given test should take into account, besides the properties of each individual testbed, the available possibilities (capacity, achievable QoS …) for interconnecting them in order to select an optimum combination of testbeds.

For establishing different requested connections, a service oriented approach should be used. Connections should be defined as a service by a set of properties that can be managed at each testbed site by a modular gateway (see Figure 6). One of the tasks of this modular gateway is to match property requirements to the connectivity service with the properties of the available connectivity. Both the available and required properties must be described in a uniform format (for example UML or XML) so that they can be automatically brokered. Connectivity service properties can be distinguished:

- Inherent properties, which are necessary and non negotiable in the process of matching properties. Examples of such properties are reliable transport, or secure transmission and these can be seen as hard requirements on the connectivity service.
- Qualitative properties, which can be used to rate a connectivity service. Examples of such properties can be bandwidth, latency, or quality of encryption.

Inherent and qualitative properties can be related. In the examples above the quality of encryption only makes sense if there is secure transmission.

In many cases Virtual Overlay Network (VON) technologies can be used to connect resources and sites with a common set of connectivity properties. Especially Virtual Private Network (VPN) or Virtual LAN (VLAN) technologies are well established means to create a logically dedicated network for a specific purpose. Although VON technologies are typically used to allow secure communications over the open Internet, a VON can also be subject to a defined service level agreement between the VON customer and the VON service provider. In the Panlab federation the VON customer is the testing customer and the system under test (SUT), while the VON service provider is the federation itself.

From the federation point of view a logical connectivity control and service support function must be implemented, which is able to receive and interpret requests for connectivity configurations and in turn to instruct the modular gateways, sitting at the edge of the individual testbeds, to establish the requested connections to the peer site or sites. This is illustrated in Figure 7 below.
4.1.1 Telecom Layered Approach

In a service oriented approach the notion of layering is not used as defined by the OSI layering model, but rather to partition the type of services provided by the architecture.

The services offered in the Panlab testbed federation can be partitioned following a generic common telecom layered approach, which can be seen as a three layers structure decoupling (i) applications, (ii) signalling control and service support functions, as well as (iii) user data transport. This separation of layers is illustrated in Figure 8 below. In order to deploy a testbed federation following such architecture, functional blocks and reference points, representing interfaces between different layers, must be defined.
The following interfaces between the layers are identified:

- **Applications to signalling control functions** as an interface to the network interactions resources. This interface is required by the applications to access the signalling plane of the testbed which is part of the overall test configuration. Access to the signalling control functions of the test configuration is specified and determined during the initial step when the testing user is specifying his requirements.

- **Applications to service support functions** as an interface to service primitives allowing access to functions supporting the overall service execution. The primitives are triggered, or instantiated by the applications when actually needed during a specific test.

- **Signalling control functions to network transport functions** as an interface to the transport layer is required to manage the resource allocation. This interface is used to control transport layers parameters and settings, according to the requirements of the application.

- **Service support functions to network transport functions** as an interface to interrogate the status of the network transport functions. This interface is used by the service support functions to collect network specific information.

At the middle tier (Figure 8), functional blocks are classified in two categories:

1. **Signalling control functions** which are very generic capabilities to route and manage signalling information. On top of IP routing functions, signalling data need to route according to addressing plans and service specific routing capabilities (example: an end-user has requested to re-route all call to a temporary voice machine).

2. **Services support functions** which include:
   - Common functions such as user databases, network management, administration, etc.
   - Service specific functions which are dedicated to different classes of service such as voice, TV, or testbed federation functions. Examples of such functions include control for media transport or delivery, service discovery functions, and functions for user registration to service.

Transport functions require appropriate control and management, related at least to: network attachment functions, network identification and authentication functions, network resources reservation and allocation, network QoS negotiations and further test bed federation functions.

4.1.2 Utilization of IMS as a Federation Tool

The IMS concept can be used to implement testbed federation functions, but has to be extended. Thereby, the IMS service support layer and IMS application layer are impacted.

The signalling control layer in Figure 8 can easily be mapped to the IMS core functionality.

In the service support layer the following function blocks need to be developed: (i) User related functions, (ii) Service Operator functions (the service being here to offer running of tested interactions), (iii) Network Management functions.

- **User related functions**: Preferences, Presence, and Directories
- **Service operator functions (testing services)**: Charging, Session management, Identity, and Service Authorization
- **Network Management functions**: Access control, Localization, Authentication, and Authorization

This list above is not an exhaustive list but rather an initial orientation for leading deeper specifications of tasks to be carried out during the deployment of a testbed federation. A comprehensive list of service functions and their exact specifications as well as definition of their interaction with application and transport layers must be developed.

The overall process to instantiate a testing configuration requires tools on the application layer (Figure 8) to define, search, selection, configure and control testbed components. It is a multi-step
process starting with an interactive customer session to describe the testing requirements, search by the system for available and capable network resources, and finally provision of relevant access tools to the customers.

To execute properly such a process, specific knowledge is constantly required to match the customer demand with testing capabilities. Accordingly, there are consequently needs for serving and managing description data for all possible services, scanning and searching applications, as well as serving and managing description data about the network topologies. All these components will be realized as specialized application servers on top of the overall system architecture, as is presented in Figure 8:

- **Service description** providing knowledge and information about the service primitives available and how to utilize them. A data model has to be defined. All services will then be described according to this model. The service description server is then providing such information on demand and according to real resource availabilities.

- **Search engine** as a specific application in relationship with the customer interface module, the service description server and a global testbed database.

- **Network topology management** providing the knowledge related to network resources and topologies. There are different network technologies entering potentially in a user configuration. When mapping the customer demand into a specific setup, knowledge about utilization of the different network technologies is required. Consequently, data model for resource and topology representation will be needed.

### 4.2 Teagle

It is elementary for a testbed federation to have a repository for information on testbeds. It is also mandatory to be able to search, sort and combine this information as easily and conveniently as possible. On the other hand processes are developed into more and more automated. This ensures cost effectiveness in the long run, but more importantly it induces systematic handling of activities also in terms of righteous, just and trustworthy operations, being not that much subject to human flaws. Automation should also correspond to the increased speed of transactions. Hence the concept of Teagle was born in Panlab.

This section describes the Teagle tool, as it is seen to be an elementary part of the Pan-European Laboratory.

#### 4.2.1 Teagle Operation, Plans and Roadmap

The simplest way of describing Teagle is to call it the electronic Reception Desk of Panlab. It is a smart interface for customers with necessary information and means for serving the customer requests. As the ‘rating’ of Panlab increases, the functionalities and features in the Teagle tool will also evolve. As in the compared hotel business a three-star hotel cannot provide the same services nor can it support such diversified requests at its Reception Desk as a five-star hotel is able to provide. Moreover, as the philosophy of Teagle evolves and Panlab matures, the stakeholders are all to be considered customers of the federation to some extent. Equally, Teagle service capabilities for other stakeholders evolve and expand. Teagle is hence a web service to satisfy customer requests and needs related to Panlab offerings.

The same way a hotel reception acts as the operations centre, giving out asks to be completed and serving as the interface to various directions, so does Teagle for the testbed federation. It has connection to a testbed database containing information on testbeds and uses this Panlab Repository of testbed functionalities description in order to be able to combine different testing components into a functional entity being able to satisfy testing needs that a single testbed may not be able to do. Teagle also needs to have access on data on testing history, giving well-proceeded constellations prioritized listing position.

**Search and orchestration engine**

It has been referred earlier in this document that Teagle is a web service, the interface for customers entering Panlab when searching for resources, services or information on federated testbeds. This is the starting point from the customer point of view. The customer having a specific need for testing or
testing related expertise can describe his/her needs on the Teagle site. The usability of the web service should be designed so that even a non-expert of the specific test requirements can choose correct options and ends up with relevant search results. The web service would provide the means for a customer to express the testing needs in as much detail as desired and get feedback on where, how and when testing could take place. Such a tool provides benefits for the customers and testbed operators alike.

Teagle needs to be able to combine services offered by different testing entities in light of their interconnecting capabilities. In some testing requests it can happen that no single testbed can satisfy the testing needs alone. In this case Teagle should be able to match possible Panlab Partners for a joint offering of the test services required. This is an important feature of Teagle for the second and third phase of the development. Also, this is where Teagle differs from traditional and only search engines. Firstly, Teagle handles structure data unlike its idols. Based on the structured information, for example correct descriptions of resources available, Teagle can compose a service offering not existing in any of the testbeds alone. Hence Teagle is a search and orchestration engine. This means that the resources, functional components of the testbed, in the repository need to be described in a standardized manner, for example in natural XML. Moreover, required details on interconnection and applicable protocols need to be provided for Teagle to know which components and how they can be combined.

The overall target would be to gather enough information from the potential testing service user to make an intelligent listing of potential possibilities serving the needs. The returned short list needs to provide for rationale of the selection and provide for opportunity for the user to make a selection from the list. Information required from the user would include such items as: planned test type, required technological capabilities, equipment, services, person-power and users, timing and the confidentiality level. Breaking down the needed data we arrive at components detailed in the paragraph 4.2.2.

**Sorted offering**

As Teagle provides the customer with the information on the testbeds and returns a list of providers matching the request, it should also be able to sort the list according to a set of criteria. The criterion can be for example schedule for next possible testing time, price, location or quality. The quality measure can be achieved from both monitored test cases within the Panlab run-time or collected feedback from the customers or co-operators. Such quality assurance measures will be established and they are depicted in the roadmap for Panlab services. Obviously, the quality-level affects the pricing of the test sites, but more importantly, builds the trust of the federation and improves the overall quality of the offerings.

**Web brokerage for testing**

As Panlab matures and Teagle grows, Teagle should evolve from an intelligent web service with abilities to combine separate viable offerings into one offering, to a Manager/Administrator of Panlab processes. Offer request handling, availability checks, offer generation, standard testing contracts, bookings, etc. should be eventually automated processes that Teagle can handle [2], [3]. Clearly, human intervention will always be needed and especially in miscellaneous cases. Nevertheless, it should be a common goal in all activities related to Panlab offerings to standardize the processes as far as possible and therefore develop the progress of Teagle into an automated test service procurer.

### 4.2.2 Data Needed From the Users

**Test type**

This data needs to describe the type of test desired. The type can range from low-level interface tests to acceptability tests with real users. Other types could include, e.g., throughput tests, spreading measurements, beam forming, protocol tests, interface tests, interoperability tests, user behaviour observations and application tests. The selection of the test type greatly affects the list of candidate test facilities and may in cases require the federation of more than one testing location.

**Technological capabilities**

The technological capabilities required in the testing are many and make for a wide variety of selections. These capabilities include items such as: spectrum, frequency, bandwidth, switching, throughput, computational, radio resource and interface requirements. Additionally, there may be
operating system, programming environment, test content and application requirements. Again, the requirements specified will narrow down the possible selection of a testing facility.

Equipment

This set of selections refers to the equipment provided by the testing environments. There may be requirements, e.g., for modelling, measuring, emulation, simulation, calculation and sensor equipment. Also, many tests require certain types of terminal equipment. In some cases it may be desired to bring the required terminal equipment to the tests, but it could also be a requirement by the party needing the tests to have these provided by the facility.

Support and person-power

The services required refer to the facilities and environments themselves and are closely connected with the person-power requirements. The required services could include, e.g., the assistance to use an equipment or installation (e.g., an anechoic chamber), having a cellular communications or a broadcasting network at one’s disposal or having the use of a real-life-simulating environment. Many of these services will require service personnel to operate the facilities and this will drive the person-power requirements of the tests. The testing customer may also opt to hire dedicated personnel for carrying out the tests in total or in part.

Users

Service and application tests will often require test users. These may range from trained users to friendly users and finally to real users selected in a manner representative of a natural demographic and / or geographic population. It may also be that several different user groups are needed for the testing. If some services or applications are meant for a range of different market conditions and need to be, for example, localized, the user population needs to reflect these needs. It may become necessary to interconnect test facilities in totally different geographies or to carry out the tests at multiple regions subsequently.

Timing

The timing is critical to know in order to schedule for the testing. The short list of potential test environments is dependent on the availability of resources, equipment and service capability. There could be situations where the best possible facility is fully or partially occupied during the desired testing time. Feedback of this needs be provided to the user. It might even be desired to attempt to obtain the availability of a certain facility through changing the testing times. The possibility to adjust the parameters in this manner would be possible using scheduling tools describe in section 4.4.3.

Confidentiality level

Finally, the confidentiality level determines, for example, whether the full use of the facility needs to be provided for one tester or whether the premises could possibly be shared. The confidentiality requirement also affects the requirements set for the testing facility to determine proper handling of the information provided for the testing, data gathered through the testing and even the setup and equipment used for the testing.

Teagle user interface (Panlab Customer interface)

The service offered and its user interface needs to be dynamic in nature. The selections and choices the customer makes greatly affect the options offered subsequently. For example, the whole range of questions and options offered to the user may depend on him/her selecting the test type. The whole information-gathering phase of the web service needs to be planned with a phased approach, i.e., the user will be greeted with simple information and a question on the test type required with explanations regarding the different existing test types. This initial selection will determine the full range of the next prompts to the user. Likewise, the answers to these questions will set up the next page of input requested. Thus, the user is guided through a natural-feeling questions and answers session giving meaningful output and a set of selections to choose from to the user and leads to accurate service setup.

4.2.3 Teagle Model

This section provides a Teagle model (shown in Figure 9) to facilitate the understanding of the Teagle tool as a central coordination instance. Teagle shall be composed of a search engine, a service
composition engine, its own service repository, several tools (used by the other components) and the Panlab Repository (as described in section 4.3). In the following, the NGN service repository, the composition engine and the Teagle tools such as the service description and discovery tools, as shown in Figure 9, will be described in more depth.

Figure 9: Teagle Model

The search engine and the composition engine make use of the service repository, the tools, and the Panlab Repository. The service description shall be realized by means of a suitable description technique such as a meta language that is able to properly describe the offered functions, data types, protocols, interfaces, parameters and return values. All necessary information must be available in a machine-readable format in order to facilitate the automation of the processes depending on the service description such as the searching and service composition.

Teagle must know which services are available in the underlying physical infrastructure as this is required by the search engine, the service composition engine and the service repository. Therefore, service discovery is an important mechanism for the proper functioning of a testbed federation. It shall be provided by a suitable registry function that allows to identify and locate available services such as Universal Description Discovery and Integration (UDDI) [6] for stateless services, or Grid’s Monitoring and Discovery Services (MDS) [7] for stateful resources.

The service composition engine makes use of the service description and the service repository (which serves as a database of all the discovered services in this case) to compose high level services to be tested. The resulting services are a composition of services with lower granularity that are offered by any testbed part of the testbed federation.

The service repository is used for NGN services that are already specified and can be used via a network abstraction layer and well known Application Programming Interfaces (APIs) that enable for example ParlayX [8] web services.

Furthermore, Teagle makes use of the Panlab Repository and middleware resources. As outlined in section 4.3, the Panlab Repository as well as the middleware resources represent physical infrastructures, devices, human resources, and the relevant metadata.

Generally, Teagle shall make use of resources via Web Service Resource Framework (WSRF) introspection to then present options to the users based on the various resource types. Teagle should possess enough knowledge about the resources to be able to create a basic configuration. The resources may then further be tweaked via Infrastructure as a Service (IaaS) [9] concepts to achieve the target configuration needed to perform the experiment.

4.3 Panlab Repository

The structure and objective of the Panlab repository is twofold:
• The Partner Testbed Repository maintains the description of the Panlab testbeds specifically detailing the testing services and caching resources offered by each Panlab partner to the Panlab community of users.

• The Testing Results Repository maintains the test results for the testing projects conducted in the Panlab so to provide useful information for documenting the activities performed within the Panlab and for sharing knowledge on past experiences (test cases, test plant configuration, key measures, etc).

The data model of the Panlab Repository undergoes a model transformation when exposed towards the Teagle users.

4.3.1 Partner Testbed Repository

Each Panlab partner shall express its availability in Teagle for offering testing services of different kinds related to the areas of interest by indicating the services offered.

Specifically a set of information shall be included in the Partner Testbed Repository.

The repository location and administration is at Panlab Office level for ensuring better control and security of the information. It shall define the set of policy rules to access the repository.

The main information to provide is:

• the composition of testing infrastructures;
• the tools offering;
• the description of the network solutions or technological areas covered in the testbed;
• the description of the available services;
• the experts available for supporting testing activities;
• calendar and timing aspects.

4.3.1.1 Access Policy and Security

The foreseen actors are:

• the repository manager, responsible for the Panlab Repository;
• the “Panlab testbed” managers;
• the Panlab partners;
• the Panlab customers.

It is suggested to provide the following use cases:

• Allow the complete set of management functions for the responsible of the Panlab repository. An on-line section shall be provided along with a backup copy of the repository. The admitted operations on the on-line section shall be: create the description for a new testbed joining the Panlab; validate input data provided by the Panlab testbed manager, remove a testbed description when the testbed is no more available.

• Allow the access for a “Panlab testbed” manager (read and write) to the portion of repository corresponding to its testbed.

• Allow the access for Panlab Partners and Customers (read-only) to the entire repository or to a subset of the information maintained in the repository.

4.3.1.2 Testing Infrastructures

The Panlab testbeds shall be described expressing in an unambiguous and uniform way the composition of the infrastructure. The description shall be sufficiently clear for all the stakeholders so to argue if a testbed hosts the required resources and services. This means detailing the nodes composing the testbed giving evidence of the functional entities that can be hosted in each node. Other information that can be provided are the general configuration aspects (as pre-condition for starting testing activities), the list of standard reference points exposed by each node, etc.

The following is a template for describing the testbed composition:
• description of the nodes composing the testbed (type, quantity, configuration; e.g. basic processor hardware, peripheral devices, operating systems, databases servers, third-party products or components, etc);

• description of functional entities available in the testbed with reference to a standard reference architecture (e.g.: referring to the IMS TISPAN architecture CSCF, HSS, RACS, etc.) with:
  • the list of standard available reference points;
  • information about the implementation;
  • the related protocol implementation conformance statements;
  • the preconditions such as the kind of network connections;
  • other requirements that must be fulfilled for integrate them in a complete network solution;

• internal network infrastructures;

• connection to external network infrastructures.

4.3.1.3 Tools Offering

The tools offered can be related to different testing activities (for instance it is possible to have test management tools as well as protocol emulators, etc).

Tools can be further classified as:

• testing management tools;

• testing automation tools;

• testing support tools.

4.3.1.3.1 Test Management Tools

Testing management tools are generic tools for defining, planning, organizing and managing the testing activities (see section 4.4.3 and section 4.4.5). They may be of different kind. For instance they may consist only in a collaborative portal where it is possible to define roles and store and manage all the documents needed for the testing activities (for example test plan, test suites, test scenarios, test cases, test results, etc).

It is possible to allow access to more complex test management tools that may include a repository or a database containing the test cases and test results, a failure management tool, etc.

4.3.1.3.2 Test Automation Tools

Test automation capabilities may be provided within a Panlab testbed. In this case they shall be described in the repository along with the description of the software tools available for scheduling and controlling tests execution.

These tools shall be integrated with manual processes. They may consist of scripts for organizing test activities. It is also possible to take advantage of a complex test automation environment providing means for:

• prepare a formal description of the “use cases” foreseen in the test case;

• set up the test preconditions;

• collect results for the “use case”;

• analyse and describe differences between the expected test outcomes and actual ones.

Test automation, being expensive, shall be evaluated case by case and it shall be explicitly considered when it is necessary to perform regression testing.

4.3.1.3.3 Test Support Tools

For each Panlab testbed there shall be a description of the available tools. Besides, it has to be declared if specific testing tools can be developed “on demand” for covering specific customer requests. Tools offering can be also related to the need of experts capable of using them.
We can mention the following:

- traffic load generators: tools and testing facilities to replicate, whenever possible, an operating environment closer to the real network conditions so to perform more significant test campaigns;
- passive traffic monitors;
- provisioning tools;
- simulation tools;
- tools and testing facilities for performance evaluation of systems and architectures characterized by high availability and high throughput;
- tools for measuring system and network performances under normal and heavy-load operating conditions;
- static and dynamic measurement tools;
- application performance profiling tools, especially needed when applications performance is a critical characteristic. Specific tools can help in identifying the portions of an application that require high execution time (CPU, memory utilization);
- installation-monitoring tools (registry, hard disk, CPU, memory, etc.)
- resource-constraining tools;
- base configuration imager and restorer;
- backup and recovery tools;
- database SQL utilities and tools;
- data acquisition tools;
- data generation tools;
- "hacker" security breach and probing tools;
- OS Security Administration tools;
- transaction load scheduling and control tool;
- usage monitors;
- benchmarking tools;
- conformance tester.

4.3.1.4 Network Solutions Covered

Testing facilities can be made available for different services, applications and network technologies. It shall be specified which areas are covered within the testbed. For instance a testbed can be oriented to the testing of IMS network solutions or to generic Next Generation Network (NGN) and to Next Generation Services (NGS). Within these main areas of interest we can foresee to provide the following services:

- testing and validation of network protocols, network and system architectures with the availability of testing tools such as protocol emulators for conformance testing;
- IMS network solutions and NGS feasibility considering interoperability aspects and so providing an heterogeneous network environment;
- IMS network and system architectures solutions with carrier grade requirements with emphasis on reliability, distributed systems guaranteeing disaster recovery, redundancy, etc.

Services should cover prototype services in order to distribute services that are still at the conceptual stage considering the evolution in voice services, video-telephony, IP television, Fixed Mobile Convergence, security and including innovative functions such as Collaborative Working, Unified Messaging, etc. These services should be accessed from existing and new kind of terminals (e.g.:
video-phones, palmtop, smart-phones) with different access technologies (e.g. WiFi, UMTS, fixed networks) and implemented with different network protocols.

Interoperability Testing can be performed considering also different kind of services, applications and network technologies. The main interest should be related to IMS and NGN networks and to NGS.

4.3.1.5 Test Types Offered

The testbed description shall include the list of services offered according to a general classification of testing services such as:

- **Functionality**
  - Function testing: verify proper data acceptance, processing, retrieval, and the appropriate implementation of the functions based upon black box techniques.
  - Security test: Security and Access Control Testing focuses on two key areas of security:
    - application-level security, including access to the data or business functions;
    - system-level security, including logging into or remotely accessing to the system.
  - Volume test: Volume testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail. Volume testing also identifies the continuous maximum load or volume the target-of-test can handle for a given period.
  - User Interface (UI) testing: User Interface (UI) testing verifies a user's interaction with the software.

- **Reliability**
  - Stress test: Stress testing is a type of performance test implemented and executed to understand how a system fails due to conditions at the boundary or outside of the expected tolerances.
  - Failover and recovery test: Failover and recovery testing ensures that the target-of-test can successfully failover and recover from a variety of hardware, software, or network malfunctions with undue loss of data or data integrity.

- **Performance**
  - Load test: Load testing is a performance test that subjects the target-of-test to varying workloads to measure and evaluate the performance behaviours and abilities of the target-of-test to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload.
  - Performance profile: Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated.

- **Supportability**
  - Configuration Testing: Configuration testing verifies the operation of the target-of-test on different software and hardware configurations.
  - Interoperability Testing: Interoperability testing verifies that expected functionalities and services are available when the target of test is integrated with other equipments or networks.
  - Installation Testing.

The offering of each service whenever possible may be further described with specific and detailed use cases.

- The use cases define the generic test sequences by giving the description of each sequence in terms of stimuli and observations for the involved entities along with the description of the message exchange on the network interfaces.
• Test prerequisites: such as the kind of network connections (for example the use of IPv4 or IPv6, the IP bearer allowed) and other requirements that need to be fulfilled regarding network protocols, authentication rules, network databases, etc.

4.3.1.6 Experts
The repository shall contain the description of the human resources available in a Panlab testbed providing the following roles:

• **Test manager** is responsible for overall management of a testing project considering: planning, reporting, logistics, etc.
• **Test analyst** identifies and defines the specific tests to be conducted.
• **Test designer** defines the technical approach to the implementation of the test effort.
• **Tester** is responsible for tests execution providing logs and test results. It is also responsible for activating recovery procedures in case of test failures.
• **Test system administrator** is responsible for setting up test environments and managing and maintaining test management systems, installing and supporting access to, and recovery of, test environment configurations and test labs.

4.3.1.7 Other Information
The repository may contain additional information if relevant for the description of a Panlab testbed.

4.3.2 Testing Results Repository
The testing results repository shall contain all the “deliverables” provided after the execution of the tests within a specific “Test Plan”.

The aim is twofold:

• provide a repository for the various artefacts created during the testing activities;
• store testing data, for instance configuration parameters and measurements results.

This concept aims at facilitating future configurations that follow the same behaviour or at providing means to compare these results when needed.

With regard to these points precedence shall be given to information that is significant for the different stakeholders.

It is not practically possible to think that a uniform model can be adopted from Panlab Customers and Partners for describing the “Test Plan” and the test results. Nevertheless, it is important to define the general rules that can help creating a homogeneous repository where the information maintained are complete and unambiguous.

As indicated in Figure 1, the testing services composition requires the cooperation of different Panlab testbeds. Therefore, the testing results repository shall be a common infrastructure.

4.3.2.1 Access Policy and Security
The repository location and administration is at Panlab office level for ensuring better control and security of these resources.

There needs to be an option NOT to save any deliverable or test results due to confidentiality/security issues.

A set of policy rules to access to the repository shall be defined.

The foreseen actors are:

• the manager of the Panlab repository;
• the managers of a “Panlab test plan”;
• the managers of a “Panlab testbed”;
• the Panlab Partners;
• the Panlab Customers.
It is suggested to organize the repository according to testing projects so to provide the following use cases:

- allow the complete set of management functions to the responsible of the Panlab repository;
- allow a limited set of users to access (read-only) the entire repository;
- allow a Panlab test plan responsible to access (read and write) the partial repository corresponding to its test plan;
- allow the Panlab test plan responsible and a limited set of Panlab partners to access (read and write) to the repository (for instance, to the portion corresponding to a specific test plan);
- allow a limited set of users belonging to the Panlab users to access (read-only) the partial repository corresponding to a specific test project; within the test project the Panlab customer can be allowed to maintain a private section until the passage of the information to the final status.

4.3.2.2 General Contents of the Repository

The test result repository shall include a concise but unambiguous description of the test performed in the Panlab within a “Test project”.

Testing results can be maintained organized according to each “test suite” performed. A test suite, according to ISO 9646, is as a logical grouping of test purposes and test cases.

The test purposes shall provide a description of a well-defined objective of testing activities (what was tested).

The foreseen deliverables for a test project are the following:

- The “Test Plan” that was prepared and preliminarily approved in order to define and guide testing activities. The test plan shall define as clearly as possible:
  - the Panlab users and partners involved;
  - the requested test plan configuration;
  - “testing scenarios” and “test cases”.
- The “Test Evaluation Summary” containing various information, mainly those significant and valuable for review and assessment activities, including:
  - the real test plan configuration including performance parameters of components and services (e.g. IP Metrics, SIP Metrics and Service Execution metrics);
  - a summary description of the tests performed;
  - a summary analysis of the test results;
  - the kind of services executed;
  - key measures of test.
- Detailed test results in form of tables listing the pass/fail result for each test case;
- Additional information, such as:
  - test guidelines (goals of testing, testing standards, key measures, test completion criteria, defect management guidelines, change management criteria);
  - logs, if any;
  - test scripts, if any;
  - reporting on test coverage;
  - quality reports;
  - incident logs and change requests.
4.3.2.3 Test Plan

Each test project shall be clearly defined by a test plan. The test plan shall provide all the needed information for defining and guiding testing activities. Also, it is necessary to identify the Panlab Customers and Partners involved in the testing activities.

4.3.2.3.1 Human Resources

It is necessary to define the responsibilities and the needs for dedicated personnel for executing the testing activities. It shall be defined both the initial request and the final situation for the human resources considering minimum effort recommended and real effort allocated for all the necessary roles.

The following roles shall be provided by the Panlab Partner (more roles may be attributed to a single person):

- Test manager (part of the Panlab organization);
- Testers, if any;
- Test System Administrator.

The following roles shall be provided by the Panlab Customer (more roles may be attributed to a single person):

- User test manager (provide equivalent functions from the Panlab Customer perspective);
- Test analyst and test designer;
- Testers.

4.3.2.3.2 Environmental Resources Used

The test results repository shall contain the “environmental resources” that is the description of the non-human resources that were requested and allocated in the Panlab testbed for executing the tests. The information shall be in line with the test plan that was prepared for executing the tests.

The following tables provide the templates

<table>
<thead>
<tr>
<th>Table 1 – Hardware Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware resource name</td>
</tr>
<tr>
<td>Node Server Name</td>
</tr>
<tr>
<td>Network or Subnet</td>
</tr>
<tr>
<td>Client Test PCs</td>
</tr>
<tr>
<td>Special configuration requirements</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2 – Software Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software resource name</td>
</tr>
<tr>
<td>Operating system</td>
</tr>
<tr>
<td>Web application container</td>
</tr>
<tr>
<td>Databases</td>
</tr>
<tr>
<td>Web browsers</td>
</tr>
<tr>
<td>Protocol analyzers</td>
</tr>
</tbody>
</table>
Table 3 – Productivity and Support Tools

<table>
<thead>
<tr>
<th>Tool Category or Type</th>
<th>Tool Brand Name</th>
<th>Vendor or In-house</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test management tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defect Tracking tool</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional testing tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance profiling tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(...)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4 – Test Environment (Application Dependent)

<table>
<thead>
<tr>
<th>Configuration Name</th>
<th>Description</th>
<th>Implemented in Physical Configuration</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average user configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimal configuration supported</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(...)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3.2.4 Testing Scenarios and Test Cases

The test plan shall define the “testing scenarios” and “test cases”.

Usually, these shall derive from the requirements analysis and use case definition phase that shall be preliminarily conducted. Note that the scenarios refer the different combinations of the main and alternate flows used in the use case definition. From scenarios it is possible to identify the test cases. Test cases provide a specification of the tests performed: test inputs, execution conditions, expected results, so to clearly identify conditions that are implemented when the test is executed.

4.3.2.5 Measurements Results

Performance measurements are important for verifying if the application requirements were satisfied with reference to specified performance behaviours. Typically, performance is stated as a measure of response time and/or resource usage as measured under various conditions, including:

- different workloads and/or system conditions;
- different use cases;
- different configurations.

Key measures may include the following:

- response time or timing profiles;
- number of events or use cases in a stated period of time;
- usage of resources;
- operational reliability (mean time to failure or MTTF) over a period of time.

Measurements may compare the application behaviour considering different configurations.

4.3.3 Panlab Repository Data Model

Figure 10 shows the data model associated to the Panlab Repository, detailing the testbed repositories and should be considered as a simplified, not-exhaustive example of a possible classification of the relevant data.
4.4 Virtualized Resources

Virtualized resources are used both by Teagle or other testbed administrators in order to access hardware or software devices that need to be shared, reserved, discovered and controlled as part of a virtual organization. The advantage of virtualizing the resources is that each Testbed remains the owner of the physical infrastructure giving limited access to the Panlab Office. For example, an organization that shares an Enterprise Switch between production and testing may decide to virtualize only a subset of ports on this switch and give them to the Panlab Community. This will give the Panlab Community or Teagle control over the VLANs, IGMP or Multicast settings over these ports while isolating the production environment. Another example of what virtual resources provide is they enforce strict policies on a tool, software or result. For example, virtualized resources containing digital media will enforce the DRM, by having a limited lifetime, enforcing the number of times this digital media can be read and who can read it.

While this approach can be compared to a centralized approach (Domain Controller), distributed resources are more scalable (no limits) and may be shared by different subgroups as the information (scheduling, policies, status) are self-contained.

4.4.1 Resource Allocation

Allocation of physical resources is at the heart of the middleware’s role. This solution must be capable of following any resource change whether the process is automatic or manual. In the case that the virtualized resource represents a physical device, a notifications based bottom-up approach (Figure 11) must be undertaken to avoid any inconsistencies between the virtualized resources and the physical devices. This goes against the usual approach which consists of creating a virtual middleware resource...
that controls the devices. The top down approach, or stateless services can cause governance issues and state mismatch when manual changes are done to the resources.

![Application Services](image1)

**Figure 11 Bottom up Approach to Virtualization**

Resources may be created from data stored in databases, communications with devices or even manual input from users. Resource doesn’t mean that automation is required; however a stateful service should be used to represent the logical object. Once these resources are virtualized, they can be exchanged between Panlab users/testbeds via the means of resource lists. The resource lists must have the following information:

- Metainformation about the resource types.
- Semantic Web techniques (RDF) should describe relationship between resources.
- Only REST URLs or End Point References (WS-Addressing) should be used to represent resources.
- Can optionally be classified under resource groups for improved listing in clients.
- Resource lists may be resources themselves.

When a list is exchanged between users, the client applications should validate the access and well behaviour of the newly acquired resources before using them to create new testbeds.

### 4.4.2 Access Right Management

Receiving resources in resource lists doesn’t mean users have access to the resources. For the middleware, three layers of security are needed. First, the container level settings, this applies to an application server running the middleware and represents the defaults settings for specific credentials. The middleware should not depend on the way user credentials are obtained, users may be validated using certificates such as X.509 certificates, Kerberos and/or SAML assertions. It is also the reason why authorization should be based on User, Group, Organization and optionally Role if attributes / Virtual Organization are supported (i.e. Shibboleth or VOMS). The second security setting should be the service level security which applies to specific methods offered by the middleware services. Finally, resource level security should have precedence over both service and container level security settings to provide resources specific permissions. This access right system allows for users to have partial access to resources. It is also important that these resources have a flag that determines if they are dedicated to specific users (only 1 user can use it at a time) or shared (multiple users can use it at the same time).

### 4.4.3 Scheduling

One key objective of the Panlab federation should be to assure an optimal use of all resources (testbeds, tools, etc.) provided by the partners. The better the resources are organized, the more users could use them simultaneously. In order to accomplish this key objective in an efficient way, it is desirable to employ automatic scheduling tools. In general, such tools should allow for planning testing events in a given testbed, planning human resources and equipment needed and even helping in purchasing equipment/materials. When requesting resources from a testbed, the scheduling tool should
be able to eliminate automatically double bookings and assure correctness of appointments, staff, meeting and resources scheduling and availability.

The scheduling tools should operate automatically making use of a wide set of expertise rules. That is, no human intervention should be required in a normal situation.

More precisely, the selected scheduling tools should have the following features.

- **Calendar**: to show the resources available in a specific date/time. It would be configurable the period to be checked, (initial and end date/time) and the measurement unit (month, week, day or hour). The system will allow only logical combinations of period and measurement units. The calendar will have different views, depending on the information required.
  - General: each simple time unit (cell) in the calendar will have associated information of every resource offered in the selected date and time, and the state of the resource.
  - Specific resource: to check availability or to book only one specific resource.
  - Resource combination: it would be possible to select several resources simultaneously in order to check its combined availability, or to book them together.

- **Resources state**: there will be different state cases, such us free, booked and not available, booked but possibility to be shared, etc.

- **Search engine for availability**: to find the date/time when a resource or combination of resources are available.

- **Reservation rules**: in order to ensure a fair resource sharing, there will be established rules related to date limit to book in advance, time limit of reservation, etc.

- **Actions to the resources**: locking, unlocking, sharing (if compatibility is possible), etc.

- **Display of information**: it would be possible to show the information related to a reservation: resource user, resource owner, etc.

- **Users’ level**: it would be possible to define different type of users with different permissions, depending on the relationship with the resource (owner, user, Panlab Partner, super-user, etc.).

### 4.4.4 Resource Economics

Part of the virtualized resources design needs to take into account the economics, if any, that will govern the resource exchange and policies. The middleware economics usually have two drivers, the price for the resources themselves that represents the value of the device or information virtualized and a fee for service price based on the quality of that service or deadline. If you are booking the resources for a time period where there is a high demand for them you should be paying more than if it is being used when there is no demand at all. Other initiatives like the IaaS Framework brokers are using the following formula to fix prices. Partner discounts are then removed from the final resource price.

\[
\text{Service Price} = \sum \left( \text{Resource Price} + \frac{\text{Time Period Usage}}{\text{Evaluation Time Period}} \times \text{High Demand Price} \right)
\]

### 4.4.5 Management Tools

Management can be in a more general way defined as the control of a group of people/entities for the coordination and harmonization of that group towards accomplishing a goal. The goals of Panlab are at establishing integration, testing, validation/verification and possibly certification services for product prototypes in Europe. With this infrastructure in place, there is the need of a straightforward Panlab management in order to reach Panlab goals and offer a reliable infrastructure.

The identified management functions, with which the management of Panlab should operate, include:

- **Planning**: deciding future developments or actions that need to be addressed for obtaining the desired functionality (here the above described scheduling tools play an important role);

- **Organizing**: assuring the required resources are ready to use when needed, in order to enable the successful carrying out of plans;
• Monitoring: analyzing the progress and accomplishment of tasks. Triggers for controlling the resources could be based on feedback;
• Self-Management: In the federation each participant should guarantee good self-management, and then facilitating overall management of the federation from the Panlab Office.
• Business management: a system to allocate the real implementation of the business model framework described in [2]. The tool will provide the required functionalities to put into practice the concepts defined in the business model.

4.4.6 Complementary Supporting Tools

In order to facilitate the testing development there will be complementary tools described below:
• Tracking system (troubles and development): it could appear the necessity of informing the resource’s provider about a specific issue (for example problems detected, improvement suggestions, etc.). In this scenario, it is needed to have specific tools to manage this information and to control the solution’s development.
• New releases information: availability of a repository with the information related to the new features and fixing patches of resources susceptible of being updated, improved, etc.

The Panlab Office can make use of tools to monitor traffic (with the help of distributed network taps) as well as by active performance tests (by means of User Agent emulators) based on the interfaces with testing sites. The test sites should also monitor their network, and notify the Panlab office in case of faults or undesired behaviour. This approach will facilitate the management of the entire federation. Figure 12 shows the virtualized resources workflow that illustrates this concept.

![Figure 12: Virtualized Resources Workflow](image)

4.5 Existing Technologies

4.5.1 Teagle Technologies

While it is clear that Teagle must offer a Web Service interface, the user front-end must also be created for the Panlab Office, this section goes over some available technologies that can be used to implement these features. However, it favours no implementation of these technologies, as this document is not intended to be a technical specification document. It will also not cover the internal
design technologies like EJB, Spring, Hibernate, etc. that are needed for a successful design and implementation.

### Table 5 – Teagle Technologies

<table>
<thead>
<tr>
<th></th>
<th>Portal</th>
<th>Webapp</th>
<th>Webservice</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Extensible</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Multiple Service</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Interfaces</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Easy to use</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Developer’s Learning Curve</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Security</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Maturity</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
</tbody>
</table>

Portals having been evolving in the past five years, using the JSR-286 specifications [10], portlets now have more access to the underlying portals. Some portals offer Web Service interface to their portlets via the Web Services for Remote Portlets (WSRP) specification [11]. They seem to be the solution that provides the best user extensibility where Panlab Community member can quickly add new tools and applications on the portal.

Web applications on the other end are the most proven technology and provide a solid way to develop, the learning curve is small and many web application developers can be recruited. However, for any change to occur the web application itself will need to be updated.

Web services themselves don’t provide user interfaces and therefore are harder to use. They must be integrated in Web Application Portals or AJAX client pages using a suitable set of public Javascript APIs (“mashups”). Web services have no built-in security.

The solution that seems to be most suited for the Panlab Community is a community Portal, because it handles many different security levels for each document, service and tool. Moreover, it offers a Web Service interface automatically for every tool. For more user control over the resources, applications may interface directly with virtualized resources as depicted in the following section.

#### 4.5.2 Middleware Resources Technologies

To virtualize the resources, different technologies may be used. The following table offers a quick comparison chart on important features for Panlab and how each technology offers needed features.

Axis Web Services are developed by the Apache software foundation [12].

Globus toolkit is a toolkit for setting up grids, that contains a java implementation of the WSRF [13] standard for representing resources as Web Services.

### Table 6 – Middleware Resources Technologies

<table>
<thead>
<tr>
<th></th>
<th>Axis Web Services</th>
<th>Globus Toolkit 4</th>
<th>IaasS Framework (GT4 Extension)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introspection</td>
<td>⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Uniform Read/Write</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Lifetime Management</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Notifications</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐</td>
</tr>
</tbody>
</table>
In order to create stateful web services, the WSRF standard by OASIS [13] should be used.

To put together the technologies, a unified resource framework is needed. IaaS Framework open source components could be used as a basis framework full of utilities/tools needed to help Panlab virtualize the different types of resources. Figure 13 shows the capabilities provided by the IaaS Framework [14] which is an extension of the Globus Toolkit 4 providing more governance and functionalities as needed to quickly develop resources.

![Figure 13: IaaS Framework Capabilities](image-url)
5 Conclusion

Acknowledging some simple facts, such as that a single testbed cannot provide every possible testing environment or every possible testing resource, or that testing resources such as high guaranteed bandwidth network links or dedicated testing equipment are very expensive, lead to the Panlab concept and elaborated proposal to federate existing testbeds. By doing so, the scattered available testing resources become available through a single logical entry point, which increases visibility and potential utilization of these expensive resources.

For realizing the proposed Panlab federation a supporting technical infrastructure must be developed and deployed that takes into account basic technical requirements; among others:

- How to interconnect remote testing resources
- How to describe and locate testing resources
- How to enable access in a uniform way

The necessity to interconnect different, potentially remote, testing resources for the provisioning of a specific testing environment leads to the first set of technical requirements. An observation from previous Panlab deliverables clearly suggests that the diversity of potential testing resources that could be federated and made available via the proposed infrastructure is very high, allowing the comparison that its complexity exhibits similarities with the complexity of controlling converged networks in the context of next generation networks and services (NGN). Allowing for this comparison, leads to the investigation of the suitability of the IP Multimedia Subsystem (IMS), which is the control plane of the NGN, as the basic framework for controlling the Panlab federation.

The first conclusion of this study is that it should be possible to reuse the architectural framework of IMS, and potential existing implementation technology thereof, to implement the technical control infrastructure of the Panlab federation. This proposal conforms to the fundamental Panlab approach, which is to maximize re-use of existing knowledge, technology and resources and only invent/introduce things that do not exist. Certainly in the course of a Panlab deployment a number of support functions that adhere to the IMS-based controlling framework will need to be implemented extended.

The large variety and different character of available testing resources that must be made visible, accessible and comparable through a single logical interrogation point leads to the second set of technical requirements, which relates to the way how the testing resources are described, stored, discovered, brokered and provisioned. To satisfy this set of requirements Panlab proposes the implementation of a new tool, called Teagle, which unifies the representation of testing resources for the purpose of later automated processing. Resources under consideration in this area include, testing environment configurations, test suits and testing results.

In the course of implementing and deploying the proposed Panlab technical infrastructure by a follow-on initiative, the Teagle concept being an essential element of the technical infrastructure, deserves particular attention.

The second conclusion is that research work has to be carried out, in order to identify and develop a suitable representation by re-using and extending existing state-of-the-art description techniques. The description of Teagle currently suggests that the Unified Modelling Language (UML), combined with the Extensible Markup Language (XML), provide the basis for this development.

A large number of different stakeholders will be granted access to the Panlab federation, which leads to the third set of technical requirements. These stakeholders will be owners and operators of testing resources and testbeds, customers of the federation seeking access to specific testing environments, and different tools, such as Teagle, to configure, provision, manage and generally administer assets of the enabling Panlab technical infrastructure at the federation level, or in the individual testbeds.

The third conclusion is that the Panlab technical infrastructure must grant controlled access at different levels to different resources, allowing for resource allocation, scheduling, access rights management, as well as access to information related to management of the federation for the purpose of fulfilling service level agreements and generally to assure quality of offerings. In order to satisfy
these requirements in a controlled way, research and development work must be carried out to build on existing resource abstraction and virtualization techniques. Overall, the Panlab technical infrastructure must impose minimum overhead to the owners of testing resources and their customers, thus a fine balance must be found between efficiency and fine grained management, so that “market dynamics” could be induced in the European testbed landscape. The Panlab vision is that the market dynamics will work towards state-of-the-art high quality testbed offerings in the federation.
References

[3] Panlab Deliverable D2.3 “User and operations manual”
[9] Infrastructure as a Service (IaaS): http://en.wikipedia.org/wiki/Infrastructure_as_a_Service