

# QoE Testbed Infrastructure and Services: Enriching the End User's Experience

Frances Cleary Grant<sup>1</sup>, Eileen Dillon<sup>1</sup>, Gemma Power<sup>1</sup>, Thomas Kaschwig<sup>2</sup>,  
Ahmet Cihat Toker<sup>2</sup> and Christian Hämmerle<sup>3</sup>

<sup>1</sup>Waterford Institute of Technology, TSSG, Cork Rd, Ireland  
<sup>1</sup>{fcleary, edillon, gpower}@tssg.org

<sup>2</sup>Technische Universität Berlin, DAI-Labor, Berlin, Germany  
<sup>2</sup>{thomas.kaschwig, ahmet-cihat.toker}@dai-labor.de

<sup>3</sup>Fachhochschule Vorarlberg, Hochschulstr. 1, 6850 Dornbirn, Austria  
<sup>3</sup>christian.hammerle@fhv.at

**Abstract.** Quality of Experience (QoE) is the subjective judgment of the satisfaction an end user perceives from an application running over a given network topology and configuration. The information provided by end users regarding their QoE preferences, experience and feedback is invaluable in providing a service that meets with their mobile activity needs within various access networks. The PERIMETER project progresses the QoE thematic research area by taking end user-related QoE factors for end user-centric mobility experimentation, thus empowering them to always have a service in which their QoE is high. This paper will detail the components of the PERIMETER framework and the user centric scenario based process adopted to implement and develop such a framework. This paper provides an insight into the federated testbed infrastructure, testing methodology and tools, operating system and applications used in the project, thus demonstrating PERIMETER's innovative advances within the QoE end user domain.

**Keywords:** Quality of Experience, testbeds, federation, PERIMETER, Future Internet, Always Best Connected

## 1 Introduction

Quality of Experience (QoE) is a measure of the end-to-end performance at the service level from the end user perspective and an indication of how well the system meets the end user's needs [1]. Consideration of QoE parameters and preferences allows a more *user-centric*, rather than network-centric, approach to be adhered to in areas of seamless mobility. Enabling the end user to control the way their identity, preferences and credentials are used empowers them to be Always Best Connected (ABC) in multiple access and multiple operator networks of the Future Internet. The PERIMETER project [2] establishes a QoE middleware framework which targets these needs in advanced networking architectures. To demonstrate such a QoE specific PERIMETER framework, a scenario based approach was adopted and a suitable federated testbed infrastructure was created to provide a valuable environment to verify the innovative QoE aspects of the PERIMETER project.

## 1.1 PERIMETER Middleware

The PERIMETER middleware is composed of a QoE management system, QoE delivery system and PPA<sup>3</sup>R (Privacy Preserving Authentication, Authorisation and Accounting and Reputation) system. These systems are supported by a Storage Layer (for storing and retrieving information using a distributed peer-to-peer approach) and an Application Layer (which contains an application manager and a GUI that provides the end user with control over their QoE parameters, preferences and settings). The PERIMETER middleware, depicted in Figure 1, is hosted on PERIMETER aware mobile devices and support nodes. Both terminals and support nodes can be directly connected to the Internet or to Virtual Private Networks (VPN) behind a Network Address Translation (NAT)/Firewall.

The central goal of PERIMETER is to devise a framework where end users are always in an ABC state. To achieve this goal, PERIMETER must gather relevant information, and make decisions on whether to generate a network switch based on the analysis of this information. The ABC state is measured using QoE metrics. PERIMETER makes its decisions using information on the end user's preferences, end user's context (application under use, location, etc.), network performance parameters, other PERIMETER end users' QoE information and PERIMETER end users' feedback, which are collected in a QoE Descriptor (QoED) [3]. The Data Network Processor is responsible for computing these QoEDs. The Decision Maker uses local and a selected subset of remote QoEDs to decide on the most suitable available network. Dedicated Trust and PPA<sup>3</sup>R components are employed to handle trust and security issues related to the sharing of QoEDs between the end users.

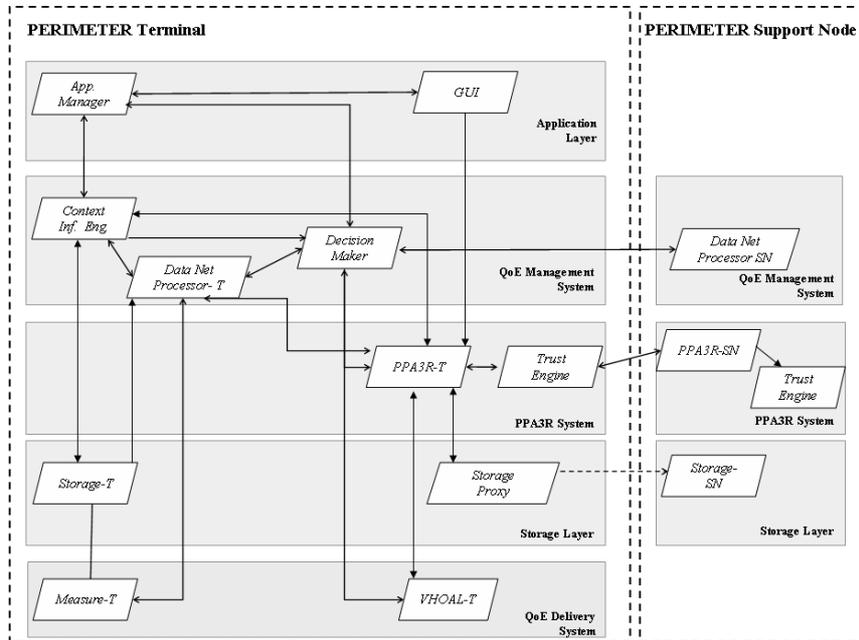


Figure 1 PERIMETER Support Node and Terminal

## 1.2 User Centric Scenario Process

Adopting an end user scenario based process allowed the definition of a suitable federated testbed infrastructure that will be capable of supporting an end user QoE PERIMETER demonstrator. Through a scenario sub-step breakdown and component mapping activities, the PERIMETER consortium detailed the relevant component functionalities, interfaces and network technologies, therefore kick starting the process of identifying the testbed initial requirements. PERIMETER Scenario 1, entitled 'user-centric agnostic ubiquitous communication', follows the a daily activity of an end user as they seamlessly roam between different technologies, connecting to services using various access technologies and devices.

A summary of the PERIMETER Scenario 1 is as follows:

Yvette is waiting for a taxi in the same room as Bob who is connected with his laptop attached to the building Ethernet network. A **phone conference** starts. Bob participates with his laptop and Yvette with her handheld device. Yvette's handheld contacts Bob's laptop through ad-hoc wireless local area network (WLAN).

As the taxi arrives, the handheld begins to face network problems and when the Taxi moves off no more packets can be delivered. Yvette's handheld has not received any more beacon from the WiFi network with Bob's laptop so the user terminal decides to get ready for a public network handover.

Yvette's handheld has already detected a pair of Universal Mobile Telecommunications System (UMTS) networks, the agent chooses the cheaper network. It has, in fact, analyzed a few statistics, previously cached from the PERIMETER overlay infrastructure, reporting that the majority of the end users had satisfactory experience with this cheaper UMTS network in that part of the city for IP connectivity.

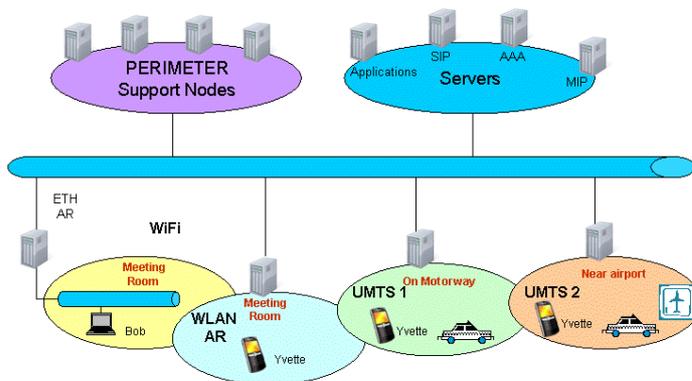
While in the taxi, Yvette receives an important **video call** from her boss. Given its nature, the call requires high quality video parameters and a stronger confidentiality. QoE descriptors of the end users on this network makes sure that a handover to this network will satisfy the needs of the video call. When connection is established with UMTS network, the video session is handed over which guarantees exceptional video-call performance.

The following work flow shows the step by step flow of actions taking into account the testbed infrastructure and the PERIMETER middleware:

- Scan for connection options and discover the nearby WLAN device.
- Establish an ad-hoc connection to the nearby device.
- Upon establishing the connection, it should request connection authorization to PERIMETER agent.
- Join the phone conference.
- Monitor the active network interfaces for acceptable QoE.
- Detect changes in the location and degrading channel quality.
- Scan for other connection options.
- Collect QoS (Quality Of Service) data and interact with the PERIMETER Support Node to update QoS information of current geographic location.
- Collect QoE data over the PERIMETER overlay.

- Process collected data using the Decision Maker component of the system.
- Switch from WLAN to UMTS based on Decision Maker's decision.
- Receive the incoming call and detect QoE requirements (high quality and confidentiality).
- Probe the QoE descriptors of other users on this network.
- Select the more expensive and reliable UMTS network and switch from UMTS1 to UMTS2.

Figure 2 conveys a high level overview of the mapping of the PERIMETER scenario 1 to the initial testbed requirements.



**Figure 2 Mapping Scenario to Testbed Requirements**

## 2 PERIMETER Intermediate Testbed Infrastructure

Two main testbeds co-exist within the PERIMETER project, the first is housed at Waterford Institute of technology (WIT) Ireland and the second main testbed is housed at Technische Universität Berlin (TUB) in Berlin. The mapping of the scenario helped identify the main testbed requirements such as terminal devices (mobile and fixed), network support technology hardware and equipment and software requirements. For the preliminary and intermediate testbed infrastructure, WIT focused on the application services, while TUB focused on the network access infrastructure required to demonstrate the scenario, as illustrated in Figure 3.

Between the two official test sites layer 3 Internet Protocol security (IPsec) interconnectivity was adopted to allow interconnection in a secure manner (Figure 4). The IPsec protocol encrypts packets before they are sent across the interconnection and also authenticates the transmitted packets. The PERIMETER testbed used IPsec tools [4] such as MOn0wall [5], a complete embedded firewall package, Racoon [4] for Internet key exchange and Setkey [4] to manipulate security associations and policies within the implemented IPsec tunnel on the hosts from both testbeds. These tools were installed on the main gateway machines, and then a common agreement for

configuration was implemented between the testbed managers from both WIT and TUB, thus providing the layer 3 testbed federation between WIT and the testbed located in TUB.

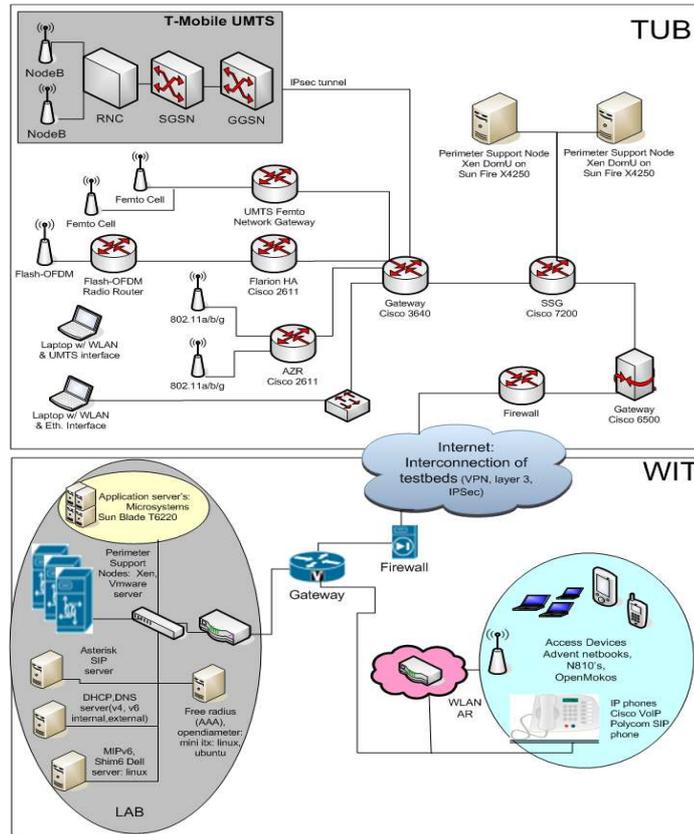


Figure 3 Intermediate PERIMETER Testbed Infrastructure

TUB and WIT are both connected over the GÉANT2 [6] network via VPN, bearing in mind that GÉANT2 is very well established all over Europe as the main national research network. TUB and WIT have the experience of conducting demos at remote sites while the communication is running in their respective testbeds.

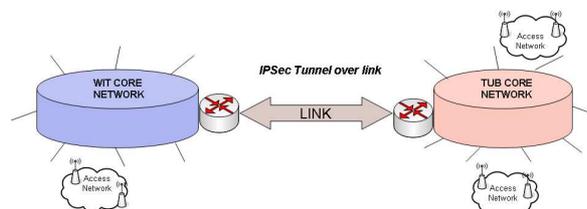


Figure 4 Interconnection between WIT and TUB test sites

The testbed interconnection has been designed with the following functionalities in mind:

- Service environment component integration.
- Testbed adaptation.
- Exposure, composition and redeployment of services and components.
- Horizontal Interconnection: to achieve greater scale. This is a relatively inexpensive and simple way to dynamically pool and use resources for experimentation of new network paradigms.
- Vertical Interconnection: to support system-level testing of new Internet networking and services paradigms across layers.

The purpose of the PERIMETER testbed federation is to deliver the scenario testing, architecture validation testing, end user evaluation testing and final project demonstration for PERIMETER by interconnecting the diverse wireless network access systems and terminals available in TUB with an IP Multimedia Subsystem (IMS) Core [7] and Application environment based on 3GPP [8] Release 7 in WIT, which can provide a multi-faceted mobile communications environment.

## 2.1 Operating System and Applications used in the QoE testbed

Adopting a user-centric scenario based approach incurs the need to address and have available certain applications in order to support the final end user demonstrator. In the PERIMETER scenario the two main applications required included a phone conference and a video conference application. It was agreed upon in the project to incorporate the use of the Google Android [9] operating system. This is a mobile operating system that runs on top of a Linux kernel. Android is free and open source software (Apache License and GPLv2), which was initially developed by Google and later by the Open Handset Alliance [10].

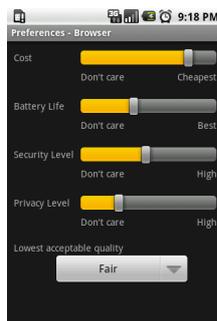
The use of the Google Android operating system has allowed the PERIMETER consortium to invest in Android compatible mobile devices. Using the Android operating system allows the project to avail of applications distributed by the Android Market. Through an application called “Market” that is preinstalled on all Android devices, this in turn allows end users access to browse and download various application published by third-party developers for the Android Market. PERIMETER assessed suitable applications to check if they would perform and address the requirements of the scenario. The following Google applications were initially assessed on the G1 mobile device containing firmware version 1.5, to determine their compatibility with the PERIMETER project:

1. For the phone conferencing application PERIMETER are examining **Sipdroid** [11]. Sipdroid is an open source SIP client implemented in Java which is capable of running on the Google Android platform. Sipdroid allows the use of an Android phone with almost any SIP provider. In the PERIMETER testbeds this is configured using an Asterisk server.

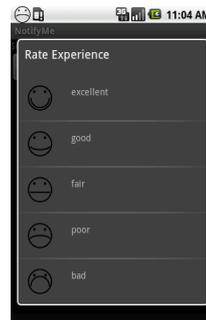
2. For the video call application PERIMETER are investigating **Semantic IPTV** [12] provided by TUB. Semantic IPTV permits video streaming on the Android phone.

All applications used within the project must be made PERIMETER aware. This involved interfacing a PERIMETER specific Application Manager (AM). The AM's main functionality is to provide the end user with the ability to control the applications running, edit their preferences and set the network selection manually if needed. To validate the usability of the AM's graphical user interface (GUI), usability tests were performed with a group of actual users from the PERIMETER consortium. This step yielded important clues about the end users' acceptance of the system and their ability to grasp and utilize the functionality represented by the GUI.

As a first step for the GUI testing, a usability pre-test was initiated to elevate statistical data of end users with different demographical (Ireland, Germany, Austria and Turkey) and technical background, with a set of interview questions. After the pre-test, the end users were provided with the Android G1 mobile phones with the PERIMETER GUI installed, shown here in Figure 5 and Figure 6. If the end user was not familiar with using an Android device, a tutorial was handed out to them, which explained the basic tasks the end user needed to know for the tests. The end users were allowed to familiarize themselves with the phone and the operating system.



**Figure 5 Browser Preferences**



**Figure 6 QoE Rating**

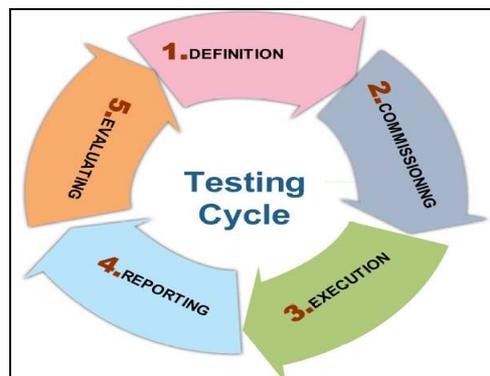
Subsequently a task list was handed out to the end users with associated questions about the look, feel and usability of the GUI. Based on this usability test, some recommendations for improving the GUI in the next iteration can be derived.

The main feedback points were in the following areas:

- The application launcher needed to be redesigned. Functionality such as the play/pause/stop metaphor for the applications was not deemed intuitive.
- Recommendation that the privacy and security preferences, shown in Figure 5, should be merged
- The dynamic feedback, involving the 'smileys' (Figure 6) was well received.
- The meaning of the cost preference (Figure 5) was not transparent.

### 3 PERIMETER Testing & Test Tools

An integrated test plan was scheduled with strict delivery deadlines for both the component code and corresponding test cases. The testing cycle for PERIMETER must be proven to be robust, scalable, interoperable and secure. The PERIMETER testing cycle [13] consists of five phases, as shown in Figure 7. These steps are mapped to the PERIMETER testing procedures for unit, functional, integration and scenario testing of the code.



**Figure 7 PERIMETER Testing Cycle**

1. **Definition:** agreement of scope and functionality to be tested.
2. **Commissioning:** setting up the test environment.
3. **Execution:** performing the tests.
4. **Reporting:** recording test results and communicating these to the interested partners.
5. **Evaluating:** dissemination of the test results and taking the appropriate actions

In order to achieve the PERIMETER testing methodology goals test tools such as Hudson [14], Trac [15], and Subversion [16] (SVN) [6] were chosen to support the integration, validation and verification process.

Hudson [14] is an open source Java-based tool for continuous integration allowing for seamless connectivity to the PERIMETER Subversion repository in addition to the automated execution of Ant [17] scripts. Hudson can be configured to constantly execute builds as new or modified source code is checked in to the PERIMETER SVN. This means that while the PERIMETER team of software developers periodically checks in new or modified code, Hudson continuously validates that the software build is not being made invalid by the new code. This reduces the need for developers to check with each other on changes to interdependent components thereby improving productivity. An extension to the continuous automated build process is continuous testing. This process ensures that newly committed or modified code does not cause predefined tests on the built entity to fail. Both build validation and testing, failures can generate notifications to alert interested parties, indicating

that a build or some tests have failed by using one of the many plugins available for Hudson such as Google Calendar or Mozilla which give an informative and unobtrusive update on the state of the build.

Trac [15] is used in the PERIMETER project as a wiki and ticket system for tracking software development and related issues as well as providing a Web based interface to the PERIMETER subversion repository and for showing the roadmap ahead listing and assigning upcoming milestones

Subversion [16] is used in the PERIMETER project for a centralized repository and a version control system, maintaining current and historical versions of documents, source code and configuration files.

## **4 Conclusion and Future Work**

This paper introduced PERIMETER's approach for user-centric seamless mobility in Future Internet. The PERIMETER middleware, which manages the QoE and PPA<sup>3</sup>R system and pictured the PERIMETER architecture, consisting of terminals and support nodes was described. A typical scenario for ubiquitous networking is shown that utilizes PERIMETER's overlay infrastructure with QoE based handover decisions to achieve an ABC networking environment. Within the PERIMETER project, a federated testbed was built with main sites at TUB and WIT, interconnected over the GÉANT2 research network. The testbed offers large-scale experimental facilities, sharing specialized networking infrastructure and services and enables integration and validation of the PERIMETER system in a heterogeneous and realistic environment.

On the terminal side, PERIMETER is based on the Android operating system. An Application Manager and GUI is being implemented, which allows the user to start PERIMETER aware applications, set preferences, returns QoE feedback to the user and enables the user to evaluate his QoE. The usability of the projected terminal software was evaluated in a field test, with primary focus on the GUI and the overall acceptance of the PERIMETER approach.

Consideration in the future testbed steps of the project will be given to use of the FEDERICA [18] infrastructure. This would enable the WIT and TUB federated testbeds to have core connectivity. As the project progresses the chosen scenario will mature to include more innovative QoE development aspects within the project.

Currently the PERIMETER system is being further developed in order to provide a complete system which can be used by an end user. This is being done in a user co-creation process, for which further usability and Living Labs [19] testing needs to be performed.

## **Acknowledgement**

This research activity is funded under the EU ICT FP7 project, PERIMETER (Project No.: 224024).

## References

- [1]. DSL Forum Technical Report TR-126, Triple-play Services Quality of Experience (QoE) Requirements, December 2006, Produced by Architecture & Transport Working Group.
- [2]. PERIMETER project website, Available: <http://www.ict-PERIMETER.eu/>
- [3]. E. Dillon, G. Power, M. O. Ramos, M. A. Callejo Rodríguez, J. R. Argente, M. Fiedler, D. S. Tonesi, *PERIMETER: A Quality of Experience Framework*, FIS 2009: Future Internet Symposium 2009, Berlin (Germany), Sep. 1-3 2009.
- [4]. IPSec Tools, Available: <http://ipsec-tools.sourceforge.net/>
- [5]. M0n0Wall, Available: <http://m0n0.ch/wall/>
- [6]. Geant2, Available: <http://www.geant2.net/>
- [7]. IMS-ARCS Project, Available: <http://www.ims-arcs.com/>
- [8]. The Third Generation Partnership Project (3GPP), Available: <http://www.3gpp.org/>
- [9]. Android Operating System, Available: <http://www.android.com/>
- [10]. Open Handset Alliance, Available: <http://www.openhandsetalliance.com/>
- [11]. SIPdroid, Native SIP/VoIP client for Android, Available: <http://sipdroid.org/>
- [12]. Semantic IPTV, Available: <http://semantic-iptv.de/>
- [13]. Testing Cycle, Available: <http://www.automated-testing.com>
- [14]. Hudson, Available: <http://hudson-ci.org/>,
- [15]. Trac, Available: <http://trac.edgewall.org/>
- [16]. Subversion, Available: <http://subversion.tigris.org/>
- [17]. Ant, Available: <http://ant.apache.org/>
- [18]. FEDERICA, Available: <http://www.fp7-federica.eu/>
- [19]. LL M. Eriksson, V. Niitamo, S. Kulki, State-of-the-art in utilizing Living Labs approach to user-centric ICT innovation – a European approach. Centre for Distance-spanning Technology at Luleå University of Technology, 2005.