



## D45.6 Online Server Testbed

Document Identification	
<b>Date</b>	27/10/2015
<b>Status</b>	final
<b>Version</b>	1.1

<b>Related SP / WP</b>	SP4 / WP45	<b>Document Reference</b>	<a href="https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/6471543">https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/6471543</a>
<b>Related Deliverable(s)</b>	D45.03, D45.4, D45.5	<b>Dissemination Level</b>	PU
<b>Lead Participant</b>	USTUTT	<b>Lead Author</b>	Janina Hofer
<b>Contributors</b>	ECS	<b>Reviewers</b>	TUD, ULD

This document is issued within the frame and for the purpose of the FutureID project. This project has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318424

This document and its content are the property of the FutureID Consortium. All rights relevant to this document are determined by the applicable laws. Access to this document does not grant any right or license on the document or its contents. This document or its contents are not to be used or treated in any manner inconsistent with the rights or interests of the FutureID Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the FutureID Partners.

Each FutureID Partner may use this document in conformity with the FutureID Consortium Grant Agreement provisions

<b>Document name:</b>	D45.6 Online Server Testbed				<b>Page:</b>	0 of 21	
<b>Reference:</b>	<a href="https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/6471543">https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/6471543</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final



### Abstract

This deliverable provides a brief description of the tools and infrastructure of the server testbed. One major outcome of this final server testbed deliverable is the release of the testbed as an online web service. The Jenkins component on <https://ci.futureid.org> is a central component of this test infrastructure. Other components, such as the SoapUI tool, a Git repository and Maven complete the online server testbed. The second access option we are going to offer is the release of the testbed on a virtual machine. In agreement with the Client testbed partners, it has been decided to combine the Server and the Client testbed for the first time into one overall FutureID testbed. A significant advantage of this approach is, that dissemination and distribution of the FutureID testbed becomes easier and clearer, since the new potential user must be given only one instrument for multiple testing purposes. Having the possible impact of the testbed beyond the project in mind, we elaborated several tailored exploitation possibilities for the FutureID testbed. The exploitation possibilities focus on the involvement of new partners in development and evaluation work after the end of the project in terms of future research projects, integration of the testbed into lab and field tests and usage of the testbed within the FutureID foundation for consulting and certification purposes.

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	1 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## Document Information

### Contributors

Name	Partner
Janina Hofer	USTUTT
Daniel Nemmert	ECS

### History

Version	Date	Author	Changes
0.1	27.03.2015	Janina Hofer	Document setup Structure of document
0.2	08.10.2015	Janina Hofer	Chapter 6 added Chapter 8 added Chapter 1 added
0.3	09.10.2015	Janina Hofer	Chapter 5 added Chapter 9 added
0.4	16.10.2015	Daniel Nemmert	Chapter 3 added Chapter 4.1 added
1.0	24.10.2015	Janina Hofer	Integrated ULD Review recommendations
1.1	27.10.2015	Janina Hofer	Integrated TUD Review recommendations and finalized document

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	2 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## Table of Contents

Abstract	1
Document Information	2
Contributors .....	2
History .....	2
Table of Contents	3
List of Figures	4
1. Introduction	5
1.1 Project Scope .....	5
1.2 Deliverable Scope and Outline .....	5
2. Server Testbed and used Tools/Methods	7
2.1 SoapUI.....	8
2.2 Git.....	8
2.3 Maven Project Object Model.....	9
2.4 Jenkins.....	9
2.5 Setup of the Server Testbed .....	9
3. Virtual Machine	11
3.1 Jenkins.....	12
3.2 Sikuli .....	12
3.3 Robot Framework .....	12
3.4 SoapUI.....	12
3.5 Additional Software .....	12
3.6 Setup and Access.....	13
4. Exploitation beyond the FutureID Project	14
4.1 Monitoring.....	14
4.2 Research Projects and Publications .....	14
4.3 Academic Teaching .....	15
4.4 Implementation into Viable Security Laboratory.....	16
4.5 Economic Exploitation .....	16
4.6 FutureID Foundation.....	17
5. Summary	18
5.1 Conclusion .....	18
5.2 Outlook .....	18
6. References	19

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	3 of 21
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU
<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## List of Figures

Figure 1 - Server testbed components and connections ..... 8  
Figure 2 - How to access the latest code from the Git repository (using the Git Client GitGui [8]).. 10

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	4 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## 1. Introduction

### 1.1 Project Scope

The FutureID project builds a comprehensive, flexible, privacy-aware and ubiquitously usable identity management infrastructure for Europe, which integrates existing eID technology and trust infrastructures, emerging federated identity management services and modern credential technologies to provide a user-centric system for the trustworthy and accountable management of identity claims.

The FutureID infrastructure will provide great benefits to all stakeholders involved in the eID value chain. Users will benefit from the availability of a ubiquitously usable open source eID client that is capable of running on arbitrary desktop PCs, tablets and modern smart phones. FutureID will allow application and service providers to easily integrate their existing services with the FutureID infrastructure, providing them with the benefits from the strong security offered by eIDs without requiring them to make substantial investments.

This will enable service providers to offer this technology to users as an alternative to username/password based systems, providing them with a choice for a more trustworthy, usable and innovative technology. For existing and emerging trust service providers and card issuers FutureID will provide an integrative framework, which eases using their authentication and signature related products across Europe and beyond.

To demonstrate the applicability of the developed technologies and the feasibility of the overall approach FutureID will develop two pilot applications and is open for additional application services which want to use the innovative FutureID technology.

Future ID is a three-year duration project funded by the European Commission Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318424.

### 1.2 Deliverable Scope and Outline

The development of complex software, such as the FutureID infrastructure, requires accurate planning and continuous testing. It must be ensured that many different components function and interoperate according to their requirements and specifications. Developers depend on accurate and reliable feedback during the development process. The server testbed serves as the main instrument to guarantee a high quality at all times during the project and beyond. A precise result presentation has already been established and described in deliverables D45.03 [1], D45.4 [2] and D45.5 [3].

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	5 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

Based on all testing related knowledge gathered within FutureID, this final deliverable implements the server testbed as an online web service. In addition, the server testbed will also be made available on a virtual machine. Since the testing infrastructure and approach have been elaborated and discussed already in previous deliverables in detail, they will just be briefly described in Chapter 2. The above mentioned two different access and usage options, the online server testbed via <https://ci.futureid.org> and the testbed on a virtual machine, are explained in the subsequent Chapter 3. In this final step of the FutureID testbed, the Server and the Client testbed partners collaborated with each other and elaborated an approach to integrate both testbeds. This approach has finally been applied and resulted in one overall FutureID testbed that fulfills the requirements of both perspectives [4, 5].

Because the FutureID consortium is convinced that the FutureID testbed is a very valuable instrument that should be used and further developed beyond the scope of the project, some possible testbed exploitation possibilities are introduced in Chapter 4. The listed options focus in particular on the involvement of new partners for further development and evaluation work after the end of the project.

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	6 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## 2. Server Testbed and used Tools/Methods

In this section, the components of the server testbed are briefly described. In agreement with the Client testbed partners, it has been decided to combine the Server and the Client testbed for the first time into one overall FutureID testbed on a Virtual machine. The result will be further explained in Chapter 3. Despite the fact of the integrated testbed, this deliverable will only focus on the server relevant components whereas the client relevant components are described in deliverables from work package 37, for example in the final deliverable D.37.6 [6].

The server testbed consists essentially of two components, which work together. One component, the SoapUI tool [7], is mainly used to create and execute tests right away. The second component consists of an installation of Jenkins, which controls the execution of SoapUI tests and gathers test results accordingly. A Git repository is used as a versioning and storage tool which transfers all created SoapUI tests into the Jenkins instance. This way it can be made sure to provide a high data quality, to avoid double work, and in a worst case, complete data loss. Jenkins applies always the latest code generated in SoapUI to build a test run. Figure 1 gives a brief overview of the used tools and required steps to use the server testbed properly. All used tools and methods that build the server testbed, their connection to each other as well as required steps are explained in more detail in the deliverables D45.03 [1], D45.4 [2] and D45.5 [3]. However, a short overview is given on the following pages.

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	7 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

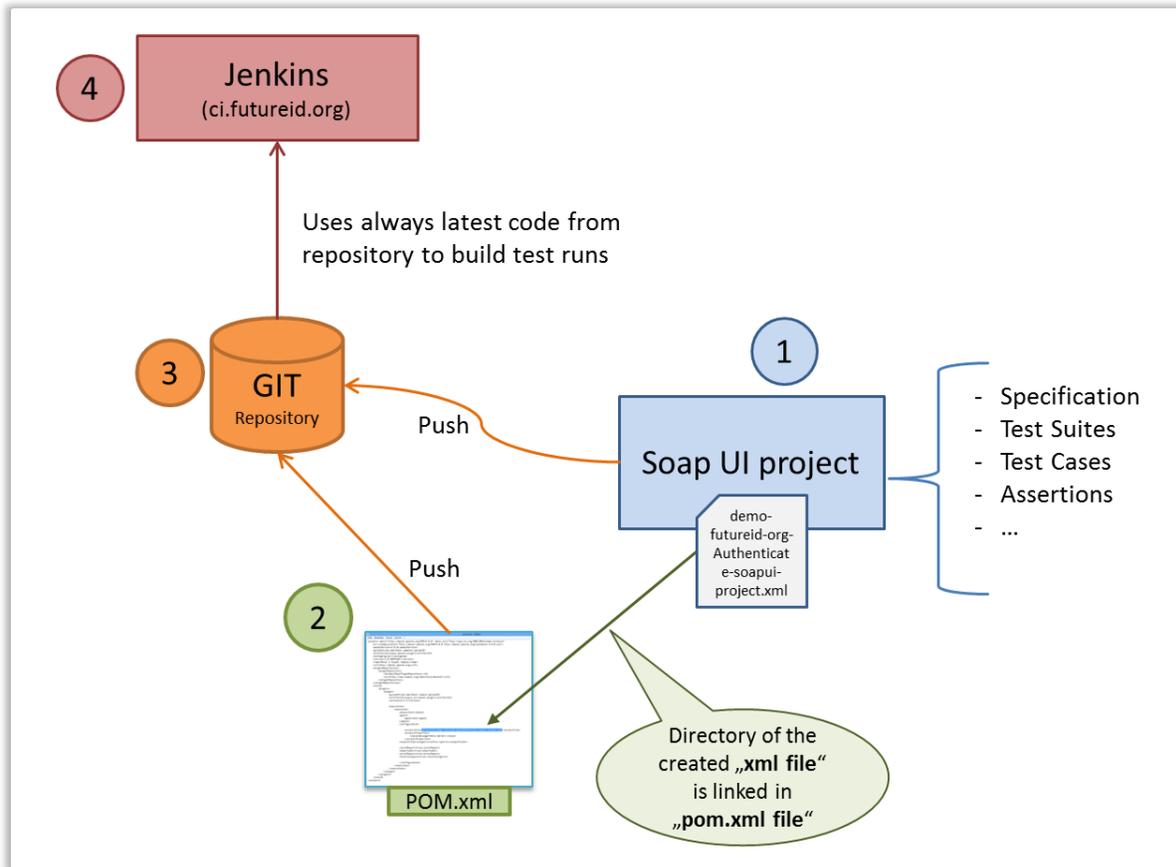


Figure 1 - Server testbed components and connections

## 2.1 SoapUI

SoapUI [7] is a free and open source tool with a graphical user interface, which is designed for functional testing of server components and web services. It provides complete test coverage and supports all standard protocols and technologies. Thus, automated functional tests of different components developed within FutureID can be created and executed.

## 2.2 Git

A Git repository is used to maintain and version the tests created in SoapUI for the different server components. Since Jenkins always uses the latest code from the repository to build a test run, any changes or additions in a SoapUI project that need to be tested have to be pushed into the repository. The different test suites are collected and maintained in this central Git repository: [fidgit@git.futureid.org:testbed](mailto:fidgit@git.futureid.org:testbed) in the folder "servertests". A detailed guideline can be found below in Chapter 2.5.

Document name:	D45.6 Online Server Testbed	Page:	8 of 21
Reference:	<a href="#">livelink</a>	Dissemination:	PU
Version:	Version 1.1	Status:	final

### 2.3 Maven Project Object Model

The central platform Jenkins uses Maven command to execute all uploaded test suites automatically. All test suites the tester wants to integrate into the next test run must be saved in the above mentioned repository folder “servertests” and be included in the designated pom.xml file, which can be found in the same folder. If the content of an already included test suite has been adapted but the name of the test suite has been retained, the test suite must be re-pushed into the repository, but the pom.xml needs no adaption. The other way around, only the pom.xml needs to be adapted by the new test suite name and be re-pushed into the repository, but the test suite itself needs no adaption. A detailed explanation and illustration can be found in deliverable D45.5 [3].

### 2.4 Jenkins

As mentioned, Jenkins applies the latest code that is saved in the repository to build a test run accordingly. The main Jenkins instance is accessible on <https://ci.futureid.org>. The usage of the central and online accessible main Jenkins instance leads to a centric view and execution of all server related function tests that can be found in the designated repository. Jenkins gives developers the possibility to execute many different test types and test cases within only one platform.

### 2.5 Setup of the Server Testbed

This chapter aims to give a brief overview of how already available tests can be accessed, how they can be adapted, and how new test suites can be integrated into the automatically executed tests. The first and very important step is to load the latest code from the repository [fidgit@ci.futureid.org:testbed](https://github.com/fidgit@ci.futureid.org:testbed) to obtain an overview of existing tests and to avoid duplicate work. To gain access to the repository, the user needs to provide a formerly generated public PuTTY key. Naturally, the public/private key pair had to be created and saved beforehand. A password has to be provided in the next step, if one has been entered during the key creation. After a successful password entry, the cloning process starts, that can take several minutes. After completion, the entire latest code for server testing is locally available and ready to use. All described steps are illustrated in Figure 2 on the following page.

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	9 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

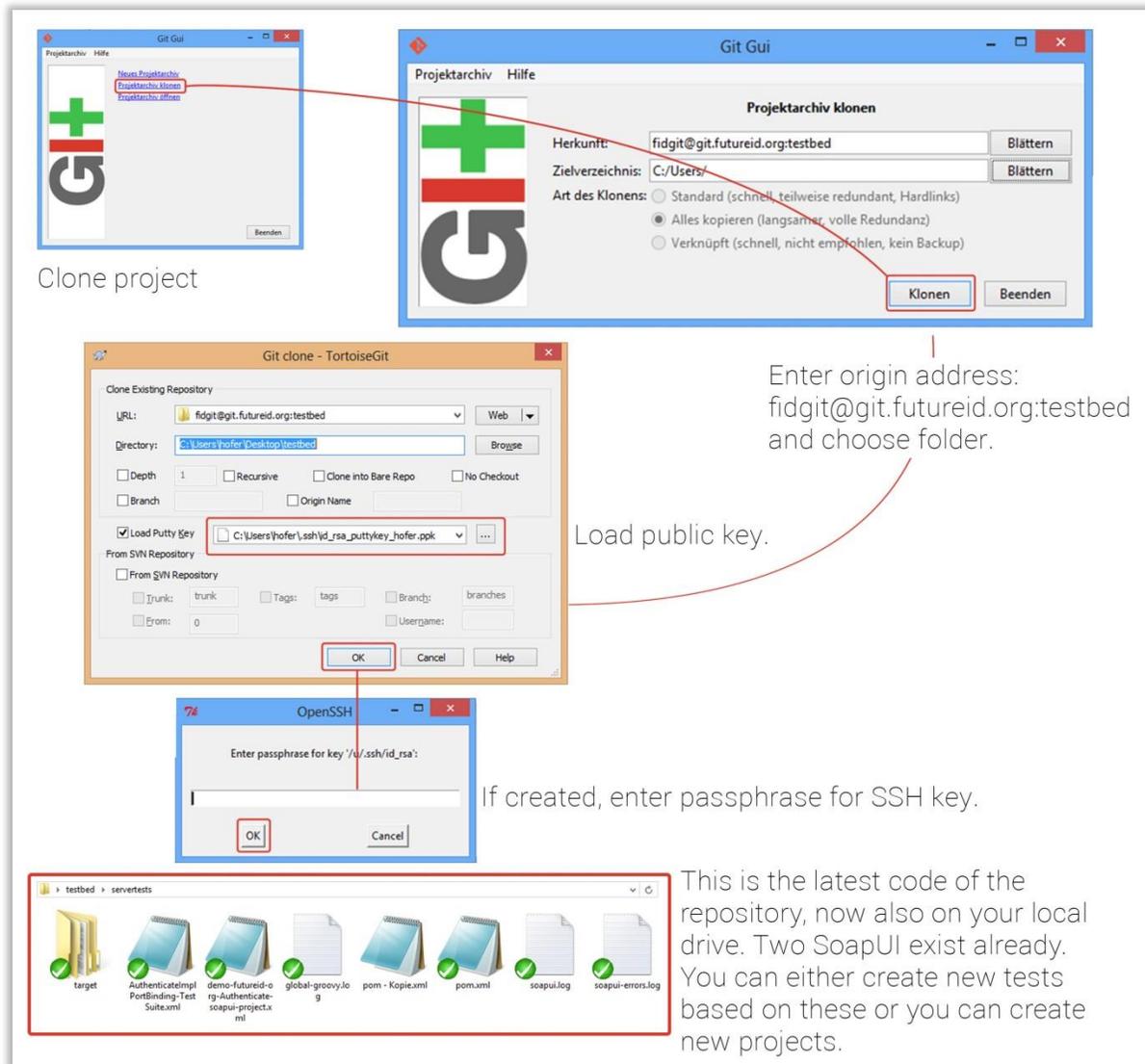


Figure 2 - How to access the latest code from the Git repository (using the Git Client GitGui [8])

Each developer and tester has now the same code as a basis and can now experiment with it, create new test cases and steps, or fix bugs. If the results of the changes are satisfying, the new or adapted SoapUI project needs to be added or updated in the Git repository. Depending on whether the project is a new project or has been renamed, pom.xml needs to be adapted accordingly (§2.3). As soon as all required files are uploaded, the tests can be executed in Jenkins (§ 2.4)

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	10 of 21
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU
<b>Version:</b>	Version 1.1	<b>Status:</b>	final

### 3. Virtual Machine

The FutureID testbed will be available via two access options; in a virtual machine (VM) and as an online web service. The entire testing infrastructure that has been developed within FutureID will be integrated into a Virtual Machine. In agreement with the Client testbed partners, it has been decided to combine the Server and the Client testbed for the first time into one overall FutureID testbed. Despite different requirements towards a testbed, on the one hand dedicated for Client side tests and on the other hand for server-side tests, a joint integration approach has been elaborated and applied. As a result, we can present one overall FutureID testbed which fulfills the requirements from both “sides”. This generates another added value: the dissemination and distribution of the FutureID testbed becomes easier and clearer, since the new potential user must be given only one instrument for multiple testing purposes. The entire infrastructure that is integrated in the virtual machine is additionally available as an online web service that can be accessed via <http://www.ci.futureid.org>.

This section will provide a rough overview over the general structure and administrative procedures to run the basic version of the client and server testbed. For the correct configuration to use Jenkins with the remote Git repository via SSH or with a local repository and a more in-depth look into the various configuration possibilities see deliverable D37.6 Acceptance Test Tools [6]. Although D37.6 focuses on the client testbed, it applies to the server testbed as well, because the provided VM consists of both the client and the server testbed in one running instance.

To facilitate the setup of Jenkins, and the testing environment in general, a virtual machine has been created that can be used “as is” to deploy a working Jenkins instance on a remote server or in a local virtual machine management software. The VM provided with this deliverable is identical to the currently running live web service<sup>1</sup>. Therefore everything described here with regards to the VM applies to the live version as well. The VM that includes the FutureID testbed is available for download via the following link: [http://redmine-iao.izs.fraunhofer.de/testbed/FutureID\\_Testbed.ova](http://redmine-iao.izs.fraunhofer.de/testbed/FutureID_Testbed.ova)

The operating system of the VM is the current 32 Bit long-term support version of Ubuntu (14.04). The following components come preinstalled with the VM:

---

<sup>1</sup> see: <https://ci.futureid.org>

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	11 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

### 3.1 Jenkins

Jenkins<sup>2</sup> is a continuous integration platform with a modular architecture and the ability to extend the functionality via plugins. It is the main entry point for testing and evaluating test results. Jenkins is the central platform in which all additional tools are brought together to unify the testing procedure and the evaluation of the results.

### 3.2 Sikuli

Skuli<sup>3</sup> is a test automation tool that uses image recognition to test graphical user interfaces. Sikuli provides a simple way to define test scripts that verify that the software is working as intended.

### 3.3 Robot Framework

The Robot Framework<sup>4</sup> is a generic test automation framework for acceptance testing and acceptance test-driven development (ATDD). It has an easy-to-use tabular test data syntax, and it uses the keyword-driven testing approach. Its testing capabilities can be extended by test libraries implemented either in Python or Java, and users can create new higher-level keywords from existing ones using the same syntax that is used for creating test cases.

### 3.4 SoapUI

SoapUI is a testing tool which facilitates testing SOAP and REST web services, among other features. Using SoapUI, it is possible to create complete test suites with a wide variety of test cases for any given WSDL. Early simple tests can be set up without any knowledge about the used scripting language (Groovy) with a few clicks.

### 3.5 Additional Software

On top of the tools listed above, these software packages are also installed:

- OpenJDK 1.7
- Python 2.7
- Jython 2.5.3
- Eclipse IDE

---

<sup>2</sup> <http://jenkins-ci.org>

<sup>3</sup> <http://www.sikuli.org/>

<sup>4</sup> <http://robotframework.org/>

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	12 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

### 3.6 Setup and Access

The VM is provided in the open virtualization format as an OVA-File, which can be imported by both VMWare's freely available VMWare Workstation Player for Windows and Linux, or the commercial VMWare Fusion for Mac and Oracle's VirtualBox Manager (available for most operating systems).

After starting the VM, all services should already be running. If Jenkins or the Apache2 server does not appear to be running, use the following credentials to access the system:

- user: futureid
- password: testbedStuttgart

It is of paramount importance that those credentials are changed if the testbed is to be used in a production environment.

On the guest system itself, you can access Jenkins in a browser by opening the url **http://localhost:8080**. If the VM is to be tested locally, there also is the possibility to access Jenkins via browser on the host system. If the VM is running, login and enter **ip address show** in the terminal. Look for an IP that corresponds with the current local network you are on. On the host system, open a browser window and access Jenkins via the IP address of the guest system (i.e. **http://192.168.178.107:8080**).

To start, stop, or restart Jenkins, use the following command:

```
sudo service jenkins [start/stop/restart]
```

The Jenkins service should run again as intended after its startup procedure is completed. Afterwards, the VM can be used to run the tests that come predefined with the VM or to configure custom tests for individual testing needs.

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	13 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## 4. Exploitation beyond the FutureID Project

The test infrastructure established within FutureID is believed to have a value and impact beyond the project. The following sections introduce some possible exploitation options for the FutureID testing approach and infrastructure, especially focusing on the involvement of new partners for development and evaluation work after the end of the project.

### 4.1 Monitoring

The provided client and server testbed VM (described in section 3) can be used to monitor the status of the FutureID project. For the quality management of FutureID there are automated acceptance tests predefined in jobs in the Jenkins continuous integration tool in the VM. The Jenkins acceptance testing jobs use Sikuli script acceptance tests with Robot Framework test cases. This allows for quality management with automated acceptance testing, because new code check-ins will automatically be tested according to the defined jobs.

The predefined test cases cover all test cases specified in the context of the FutureID project. Users can define custom test cases easily by following the steps described in deliverable D37.6 Acceptance Test Tools [6].

The distribution of a pre-configured virtual machine for client and server tests offers an easy way to set up a testing environment in the future with only few additional effort needed to keep it up to date, such as defining new test cases. The set up can be done by everyone with minimal background knowledge in virtual machines and makes the automated testing environment as future-proof as possible.

Additionally, the provided VM can also seamlessly be integrated into an ISO 9001 style quality management system.

### 4.2 Research Projects and Publications

Most of the research institutions from the FutureID consortium proposed and will further propose new and innovative research projects answering different calls in different funding programs (e.g. ENTOURAGE, Industrial Data Space, LIGHTest, etc.). Especially the calls within the Horizon2020 program seem to be very promising. Most potential future research projects that are part of Research and Innovation, especially innovation actions in the scope of Horizon2020, develop some kind of software. Certainly, all software or modules of software need to be tested on a regular basis to fulfill certain requirements, particularly regarding functionality, interoperability, and usability. The

<b>Document name:</b>	D45.6 Online Server Testbed				<b>Page:</b>	14 of 21	
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

Innovation Actions of H2020 require a technology readiness level of 7 by the end of the project, which means that the developed product must be a demonstrable system prototype that functions in an operational environment [9]. In these projects it becomes inevitable and highly important to use a functioning test approach to make sure the developed software is mature and robust enough to fulfill the TRL7 requirements.

Given that the FutureID testbed is modularly built, it will take little effort to align it to another project and context. By actively implementing the FutureID testing approach and infrastructure, the possibility is given to further apply and develop the testbed. It automatically will be introduced to new research and business partners by which it can only benefit from. The more the testing approach and infrastructure is being used in different contexts, the more feedback from different stakeholders can be received. Thus, all feedback can immediately be analyzed and transformed into concrete development and take-up measures. In addition to that, all testbed partners become attractive partners for a wide range of research projects.

Gained knowledge towards testing in general and the testing approach developed within FutureID will certainly be published in dedicated peer reviewed journals and in this way be presented to interested new partners. Future conferences and exhibitions, such as the ICT event [10], ISSE [11], IT&Business [12], Open Identity Summit [13] and many more can and will be used to further demonstrate the FutureID testbed. Such activities are especially helpful to market the FutureID testbed to any interested stakeholder from research over public towards the private sector.

### 4.3 Academic Teaching

The objective of transferring the gained testing knowledge and tools developed in FutureID into the academic teaching domain is to enrich it with external contributions of complementary research fields.

By integrating the specialized knowledge regarding testing in security-relevant software development into dedicated lectures, the FutureID testing approach will be presented to a large group of potential new scientists and developers in the IT security field. We consider the introduction of the testing strategy and infrastructure developed within FutureID to the new generation of scientists as essential and helpful for further development of the FutureID testbed.

Especially research institutions frequently work together with young and innovative students of different study backgrounds, which generate mutual benefits. The students gain insight into the current research topics and results, whereas research institutions

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	15 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

can integrate innovative and new feedback and ideas into their current research and development actions.

#### 4.4 Implementation into Viable Security Laboratory

Partners from the private and the public sector want to develop solutions that are successful in the marketplace and improve security when applied in practice. When creating such solutions, developers must focus on the real-world working environments of users. The VS-Lab viable security laboratory [14] on behalf of Fraunhofer IAO supports vendors in their evaluation and development work, and helps enterprises to identify and integrate viable security solutions. In pursuit of these objectives, the VS-Lab features infrastructure that enables it to incorporate user perspectives and thereby examine IT security technologies from a holistic point of view.

To complement the VS-Lab infrastructure, the FutureID testing approach and tool setup will be integrated. Evaluation and development methods, such as penetration tests, workshops and focus groups, participatory testing and evaluation sessions, and user vulnerability assessments can be conducted in this laboratory and supported by the FutureID testbed. With the embedding of the testbed into the VS-LAB infrastructure, partners from various industries will be introduced to the FutureID testbed, which gives us the possibility to operate the testbed in different branches of industries and to collect feedback from different persons with different backgrounds and requirements towards the testbed. Conducting these tests using the testbed will further provide rich feedback regarding possible enhancements or integration of new components in the testbed.

One actual application of the testbed is advisable when it comes to the planned usage of the FutureID infrastructure by the Fraunhofer-Wiki-Farm. This actual future deployment use case needs an associated testing approach in order to guarantee a successful integration in the ongoing operation of the Wiki-Farm. The logic and the structure of the FutureID testbed and its embedding in the VS-Lab infrastructure enable to conduct critical tests independently without disturbing the habitual use of the Wiki-Farm. Another benefit from using the testbed during this process is that the Fraunhofer-Gesellschaft could introduce this FutureID testing service to their member institutes and their partners.

#### 4.5 Economic Exploitation

Collected and harmonized knowledge regarding testing in general and testing of security-relevant software will be utilized in practical publications and advisory services, and thus will be made accessible to a wider range of enterprises, especially SMEs. An economic exploitation of the FutureID testbed emerges from the use of consulting

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	16 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

industry partners. Since this testing approach developed within FutureID is modular and open for adjustments and additions, it can easily be adapted to any kind of software testing. It therefore can join different applicable corporate products, processes and frameworks and be operationally used. It is envisioned to establish strong liaisons that promote interest in take-up and foster realization of market opportunities. Collaborations with commercial partners will serve to complement the economic exploitation plan of the testbed, garner feedback from the market, and additionally identify potential partners and users.

### 4.6 FutureID Foundation

The FutureID testbed can and will be part of the yet-to-be-formed FutureID Foundation, mainly for consulting and certification purposes. The FutureID Foundation is planned to support the sustainable creation of a European eID community, consisting developers, integrators, vendors, and other interested organisations with the aim to promote the integration and utilisation of FutureID components and compatible services within federated identity schemes. In other words, the FutureID Foundation aims at providing a central market place for FutureID components and other compatible modules.

The testbed, being a very strong and important and yet independent component of the entire FutureID infrastructure, can thereby be introduced to new partners. It is envisioned to use the FutureID testbed for consulting and certification purposes. For the latter, the testbed could be applied as a key enabler for being able to offer certification of FutureID compliance.

In general, the application of the testbed in other use contexts will provide rich feedback from new partners from different sectors, who have not been involved in the FutureID project so far. This generated feedback will be collected and interpreted according to use context. It enables further development of the testbed by means of enhancing and fine-tuning existing components as well as by developing and integrating entirely new components, if there is sufficient user demand.

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	17 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## 5. Summary

In this document, the final server testbed deliverable, we summarize and describe the server testbed. The two different access options, the online web service and the virtual machine, their functionality and set up are being explained in Chapters 2 and 3. Since the testbed is seen as a valuable instrument for future software development projects, we presented some envisioned exploitation possibilities in Chapter 4.

### 5.1 Conclusion

The server testbed has evolved over the entire duration of the project. Several intermediate steps have been achieved, which eventually led to a stable server testbed. Moreover, the requirements for the server testbed were exceeded. It is not just available as an online web service as expected, but can also be used and distributed in a virtual machine. In addition to that, there exist no longer two separate testbeds, where one is dedicated for server tests and the other one for client tests, but one overall FutureID testbed. Having just one instrument for all testing purposes is a major advantage when it comes to dissemination and exploitation of the testbed beyond the project scope of FutureID. Speaking of which, some of the presented exploitation options are very promising with respect to the acquisition of new potential customers or partners.

### 5.2 Outlook

As clearly stated, it is highly envisioned to further operate and develop the FutureID testbed in various contexts. The briefly described exploitation options are able to achieve this goal, provided they will be worked out in more detail in the future, probably as one topic of the FutureID Foundation. However, a first guideline was created with these options. If these are followed, new partners can easily be involved in further operation of the testbed, which will eventually lead to the integration of new components according to possible new use contexts.

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	18 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

## 6. References

- [1] E. Özmü und B. Biallowons, „D45.03 First Internal Version of Online Server Testbed,“ [Online]. Available: <https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/4825629>.
- [2] J. Hofer, T. Schulz und L. Fritsch, „D45.4 Test assertions,“ [Online]. Available: <https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/5442590>.
- [3] J. Hofer, T. Wich, T. Schulz, L. Fritsch und D. Wójs, „D45.5 Transformation tools to present test results,“ 26 04 2015. [Online]. Available: <https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/5463930> . [Zugriff am 08 10 2015].
- [4] E. Özmü, C. Ruff, M. Drabik und L. Fritsch, „D45.1 – Requirements Report Server Testbed,“ 27 05 2013. [Online]. Available: <https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/3119203> . [Zugriff am 09 10 2015].
- [5] E. Özmü, C. Ruff, U. Keskel, L. Fritsch und J. Kubieziel, „D37.1 Requirements Report Client Testbed,“ 02 05 2013. [Online]. Available: <https://dms-prext.fraunhofer.de/livelink/livelink.exe/properties/2985845> . [Zugriff am 09 10 2015].
- [6] J. Hofer, M. Toom und T. Schulz, „D37.6 Acceptance Test Tools,“ [Online]. Available: <https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objId=5628938&objAction=browse&viewType=1>.
- [7] S. SMARTBEAR, „SoapUI download,“ [Online]. Available: <http://www.soapui.org/>. [Zugriff am 03 02 2015].
- [8] GitHub, „GUI Clients,“ [Online]. Available: <http://git-scm.com/downloads/guis> . [Zugriff am 03 02 2015].
- [9] European Commission, „Technology readiness levels (TRL),“ [Online]. Available: [http://ec.europa.eu/research/participants/data/ref/h2020/wp/2014\\_2015/annexes/h2020-wp1415-annex-g-trl\\_en.pdf](http://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf). [Zugriff am 08 10 2015].

- [10] European Commission , „ICT 2015 Innovate, Connect, Transform,“ [Online]. Available: <http://ec.europa.eu/digital-agenda/en/news/ict-2015-innovate-connect>

<b>Document name:</b>	D45.6 Online Server Testbed				<b>Page:</b>	19 of 21	
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final

transform-20-22-october-lisbon-save-date. [Zugriff am 08 10 2015].

[11] EEMA, „ISSE 2015,“ [Online]. Available: <http://www.isse.eu.com/>. [Zugriff am 08 10 2015].

[12] Landesmesse Stuttgart GmbH, „IT & Business,“ [Online]. Available: <http://www.messe-stuttgart.de/it-business/aussteller/>. [Zugriff am 08 10 2015].

[13] Open eCard-Team, c/o ecsec GmbH, „Open Identity Summit,“ [Online]. Available: <https://www.openidentity.eu/home/>. [Zugriff am 08 10 2015].

[14] F. IAO, „VS-Lab: viable security laboratory,“ [Online]. Available: <http://www.iao.fraunhofer.de/lang-en/collaboration/labs-equipment/1242-vs-lab-viable-security-laboratory.html>. [Zugriff am 08 10 2015].

<b>Document name:</b>	D45.6 Online Server Testbed	<b>Page:</b>	20 of 21				
<b>Reference:</b>	<a href="#">livelink</a>	<b>Dissemination:</b>	PU	<b>Version:</b>	Version 1.1	<b>Status:</b>	final