SOCIETIES
Deliverable D7.6v2
First Integrated Prototype
Test Report

Editor: TRIALOG
Deliverable nature: Software Prototype Deliverable
Dissemination level: PU
Contractual delivery date: Oct 31th 2012
Actual delivery date: Nov 2nd 2012
Suggested readers: WPL, Developers, Integrators of the SOCIETIES Platform
Version: 2.0
Total number of pages: 37
Keywords: SOCIETIES Platform, First Prototype of the Integrated SOCIETIES Platform, Testing, Development.

Abstract
D7.6 v2 is a software deliverable. Platform developers are the targeted users of this deliverable. This document describes the first integrated prototype of the SOCIETIES Platform delivered by WP7 following the development performed by WP4 and WP5. It includes a description of the platform architecture and the services it provides, the deployment nodes, how to get and use the software distribution, how the components are defined and integrated.

The main goal of the v2 of this D76 is to report the results of tests performed up to now to obtain a functional platform.

A developer manual of each component is provided in appendix of the v1 of this D76 but it is not repeated here. It will be available on the project wiki very soon.

The SOCIETIES Platform aims at providing individual and community experiences while realising the concept of Ambient Intelligent (AmI) Community. The Platform is providing core services described in this document. It is supporting third party services to be developed by WP6 and integrated in WP7 in the other deliverables.

[End of abstract]
Disclaimer

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

In case of Public (PU):
All SOCIETIES consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP):
All SOCIETIES consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE):
All SOCIETIES consortium parties have agreed to full publication of this document. However this document is written for being used by <organisation / other project / company etc.> as <a contribution to standardisation / material for consideration in product development etc.>.

In case of Consortium confidential (CO):
The information contained in this document is the proprietary confidential information of the SOCIETIES consortium and may not be disclosed except in accordance with the consortium agreement.

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

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

Impressum

[Full project title] Self Orchestrating Community Ambient Intelligence Spaces
[Short project title] SOCIETIES
[Number and title of work-package]: WP7: Integration and testing (Internal Validation)
[Editors: Name, company] Bruno Jean-Bart (TRIALOG)
[Work-package leader: Name, company] Bruno Jean-Bart (TRIALOG)
[Estimation of PM spent on the Deliverable] 30

Copyright notice

© 2012 Participants in project SOCIETIES

Optionally list of organisations jointly holding the Copyright on this document
Executive summary

This deliverable aims at presenting the release 0.4.1 of the Societies Platform and the report of the tests performed until now.

The SOCIETIES project aims at developing a concept of Ambient Intelligent (AmI) Communities that extend ambient intelligent or pervasive systems beyond the individual to dynamic communities of users. The Platform implements Community Interaction Space (CIS) to intelligently connect people & things and enable them to communicate, share and consume services. SOCIETIES will embrace online community services, such as Social Networking, thus offering new and powerful ways of working, communicating and socialising.

A CIS is a group of, two or more, individuals who have agreed to share some, but not necessarily all, of their pervasive resources – personal information, physical context information, services, and devices – with other members of that community. These pervasive resources are defined within a CSS (a Co-Operating Smart Space) each user of the Platform should possess. A CSS is the user virtual space created via one or several devices (e.g. his smart phone) or nodes (e.g. Cloud Node). Therefore CIS aims to completely transform traditional online social networks, freeing users from web-applications and letting them loose in the real physical world. SOCIETIES Platform supports the creation of Communities by discovering, connecting and organising relevant people and things from both physical and digital environments. SOCIETIES will use pervasive technologies to form adaptive communities, while leveraging social networks and crowd computing techniques.

The D7.6 v2 is a software deliverable. It provides the release 0.4.1 of the SOCIETIES Platform. This document gives the description of the features of the platform with developers as the targeted users. This deliverable is the result of the integration and test work performed by WP7 from the outcome of the development of core components by WP4 (see D4.1) and WP5 (see D5.6). The integration process has been described in the previous deliverable D7.1.

The document is structured into two main parts:

- Section 2 to 4 provides a description of the platform and its distribution.
- Section 4 to 9 provides a description of the test results.
# List of authors

<table>
<thead>
<tr>
<th>Company</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIALOG</td>
<td>Bruno Jean-Bart, Rafik Said Mansour, Olivier Maridat</td>
</tr>
</tbody>
</table>
# Table of Contents

Executive summary ........................................................................................................... 3  
List of authors .................................................................................................................. 4  
Table of Contents ............................................................................................................... 5  
1  Introduction .................................................................................................................. 6  
2  Description of the Integrated Platform ........................................................................ 6  
   2.1  Deployment Architecture ....................................................................................... 6  
   2.2  Description of the Cloud Node .............................................................................. 7  
   2.2.1  Integration Diagram (v2) .................................................................................. 8  
   2.3  Description of the Domain Authority Node ........................................................ 10  
   2.3.1  Integration Diagram ......................................................................................... 11  
3  Description of the Light Node ...................................................................................... 12  
4  Distribution of the release 0.4.1 .................................................................................. 13  
   4.1  Cloud Node on Virgo ............................................................................................ 13  
      4.1.1  External Software .......................................................................................... 13  
   4.2  The Light Node on Android .................................................................................. 13  
5  Tests Organisation ........................................................................................................ 14  
6  Unit Tests Report ......................................................................................................... 15  
   6.1  Tests Coverage ...................................................................................................... 15  
   6.2  Unit Test Report per module ............................................................................... 18  
7  Integration Tests Report .............................................................................................. 19  
   7.1  Integration Tests Environment ............................................................................. 19  
   7.2  Test Results .......................................................................................................... 19  
8  Functional and Behavioral Tests Report ..................................................................... 23  
   8.1  Organisation of the Tests ...................................................................................... 23  
      8.1.1  Scenario .......................................................................................................... 23  
      8.1.2  Basic scenario .................................................................................................. 24  
      8.1.3  3P services scenario ....................................................................................... 24  
      8.1.4  Advanced scenario and other tests. ................................................................. 25  
      8.1.5  Process ............................................................................................................ 25  
   8.2  GUI used during the tests ...................................................................................... 27  
   8.3  Test Results .......................................................................................................... 31  
9  Conclusion .................................................................................................................... 34
1 Introduction

The current document describes the release 0.4.1 which is the first prototype of the SOCIETIES Platform. This release is the result of the work performed both by the developer of the platform (WP4, WP5) and the integrators (WP7) who have created then executed the test plan defined in the D7.1.

This release includes Cloud Node and Light Node on Android.

2 Description of the Integrated Platform

2.1 Deployment Architecture

Same as D76 (July 2012).
2.2 Description of the Cloud Node

The Cloud Node software consists of the components listed in table below. The technical description of each this component is provided in section 4.2.

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Description</th>
<th>Virgo Plan for deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Framework</td>
<td>This component provides the communication mechanism throughout the SOCIETIES platform.</td>
<td>org.societies.css.commfrwk</td>
</tr>
<tr>
<td>Common Modules</td>
<td>This component provides common features as activity feeds and persistence.</td>
<td>org.societies.common.modules</td>
</tr>
<tr>
<td>Security</td>
<td>This component groups the bundles required for secure policy negotiation, digital signatures management and access control that is related to policy negotiation and service sharing.</td>
<td>org.societies.security</td>
</tr>
<tr>
<td>Privacy &amp; Trust</td>
<td>This component manages the privacy and trust features as privacy policy negotiation when installing a 3P service or joining a CIS</td>
<td>org.societies.privacy.and.trust</td>
</tr>
<tr>
<td>Device Management</td>
<td>This component aims at managing external devices (sensors / actuators) connected to the node.</td>
<td>org.societies.css.devicemgmt</td>
</tr>
<tr>
<td>Context Management</td>
<td>This component provides the functionality regarding the representation, collection, management and sharing of context information and the inference of higher level context attributes, in order to support the provision of context-aware experience to the CSS users</td>
<td>org.societies.context.management</td>
</tr>
<tr>
<td>CSS Management</td>
<td>This component aims at managing CSS Nodes. By adding, deleting and update their configuration.</td>
<td>org.societies.css.management</td>
</tr>
<tr>
<td>CIS Management</td>
<td>This component aims at managing the CIS. By creating, deleting, update configuration, and manage the service and resource sharing.</td>
<td>org.societies.cis.management</td>
</tr>
<tr>
<td>Service Lifecycle</td>
<td>This component manages the 3P service lifecycle features as installing, uninstalling, starting, stopping and updating 3P services. It maintains a registry of services used by CIS.</td>
<td>org.societies.service-lifecycle</td>
</tr>
<tr>
<td>SNS</td>
<td>This component groups bundles needed to manage the social data connectors used to import information from social networks as facebook, twitter...etc</td>
<td>org.societies.sns</td>
</tr>
<tr>
<td>User Agent</td>
<td>This component groups bundles related to the user. It provides personalisation-related features such as decision-making algorithms, conflict resolution, user feedback (managing all notification to the user).</td>
<td>org.societies.user.agent</td>
</tr>
<tr>
<td>Personalisation</td>
<td>The personalization component enables other platform services to customise their behaviour for individual users to provide a better end user experience.</td>
<td>org.societies.personalisation</td>
</tr>
<tr>
<td>WebApp</td>
<td>This component provides a GUI for platform developer to run the core components of the platform.</td>
<td>org.societies.webapp</td>
</tr>
</tbody>
</table>

Table 1: Components of the Cloud Node
2.2.1 Integration Diagram (v2)

The diagram of the figures below depicts the interactions between components of the Cloud Node. Each square represents a component (i.e. a Virgo sub-plan grouping the bundles to form the component). The interfaces (provided and required) are represented by the circle and semi-circle respectively.

Some links between the components are not represented to avoid overloading the picture. In particular links to the Communication Framework are not represented as all other components are using it. Similar is the case of the Common Module component. But the principle is that when a service “provided” is available and the same service “required” exists, a link could be added between them.

Apart from the above cases, several provided interfaces are not used by another component. Two cases can explain this:

1) the interface is used only by the component itself and therefore will be removed from the org.societies.api in the next release.

2) the feature associated to this interface is not yet used by another component but this would be the case in the near future (i.e. next release).

The main additions compared to the v1 of this Deliverable are the integration performed between CIS, Service Life Cycle and Privacy Protection through the IPrivacyDataManager and IPrivacyPolicyManager interfaces. In this process, it is worth to mention also the use of INegotiation interface by the CIS Management. An important added feature is also the use of the IUserFeeback by CIS Management and Service Life Cycle to interact with the user during the sharing service process.

Figure 1: Diagram of Cloud Node Components and their Links (part 1).
Figure 2: Diagram of Cloud Node Components and their links (part 2).
2.3 Description of the Domain Authority Node

The Domain Authority Node software consists of the components listed in table below.

<table>
<thead>
<tr>
<th>Component Name (bundle)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DACommsBundle</td>
<td>This component acts as a proxy to communicate with the domain authority node</td>
</tr>
<tr>
<td>CssDirectory</td>
<td>This component is responsible for the management of the public information relating to the CSS</td>
</tr>
<tr>
<td>CisDirectory</td>
<td>This component is responsible for the management of the public information relating to the CIS</td>
</tr>
<tr>
<td>DomainAuthorityGroupManager</td>
<td>This component handles the remote APIs management through the communication framework</td>
</tr>
<tr>
<td>DomainAuthorityRegistry</td>
<td>This component handles the storing and persistence of the societies user accounts information</td>
</tr>
<tr>
<td>Security.secur-services</td>
<td>This bundle provides some of the low level security services, namely digital signatures support.</td>
</tr>
<tr>
<td>DomainAuthorityRest</td>
<td>RESTful server for sharing any files related to 3rd party services. The server has access control and allows download only to clients that present a valid proof of authorization from the service provider.</td>
</tr>
<tr>
<td>DomainAuthorityWebApp</td>
<td>This component provides a GUI for platform user to run and configure the Domain Authority Node.</td>
</tr>
<tr>
<td></td>
<td><strong>The two components below are already in the Cloud node deployment. They should be set in the DA plan only if the DA node is deployed separately</strong></td>
</tr>
<tr>
<td>XCCommunicationMgr</td>
<td>Same as for the Cloud Node (see table 1)</td>
</tr>
<tr>
<td>data-source</td>
<td>Same as for the Cloud Node (see table 1)</td>
</tr>
</tbody>
</table>

Table 2: Components of the Domain Authority Node
2.3.1 Integration Diagram

Figure 3: Diagram of Domain Authority Node Components and their links.
3 Description of the Light Node

The Light Node software on Android consists of the components listed in table below. Just note that the apklib is the name of the packaging format for Android and corresponds to jar file.

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Description</th>
<th>Name of the apklib</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIS Manager</td>
<td>This component aims at managing the CIS. By creating, deleting, update configuration, and manage the service and resource sharing</td>
<td>cis_CommunityManagerService_apklib_0.4</td>
</tr>
<tr>
<td>CSS Manager</td>
<td>This component aims at managing CSS Nodes. By adding, deleting and update their configuration.</td>
<td>cssmanager_SocietiesCSSManager_apklib_0.4</td>
</tr>
<tr>
<td>Service Life cycle</td>
<td>This component manages the 3P service lifecycle features as discovery installing, uninstalling, starting, stopping and updating 3P services on Android. This version is limited today to some key features (see section 8.3)</td>
<td>SocietiesServiceMonitor_apklib_0.4</td>
</tr>
<tr>
<td>Social Network System</td>
<td>This component groups bundles needed to manage the social data connectors used to import information from social networks as facebook, twitter...etc</td>
<td>sns_doConnectAPP_apklib_0.4 SocietiesAndroidSocialData_apklib_0.4</td>
</tr>
<tr>
<td>Privacy and Trust</td>
<td>Available but not used yet in R0.4.1.</td>
<td>privacytrust_apklib_0.4</td>
</tr>
<tr>
<td>Societies Content</td>
<td>Managing the local database.</td>
<td>content_SocietiesContent_apklib_0.4</td>
</tr>
<tr>
<td>LocalDeviceStatus</td>
<td>Not used.</td>
<td>LocalDeviceStatus_apklib_0.4</td>
</tr>
<tr>
<td>Android Agent</td>
<td>Implements the Comm Framework features to communicate with the other CSS Node and DA Node and XMPP server</td>
<td>AndroidAgent_apklib_0.4</td>
</tr>
<tr>
<td>Android Pub Sub</td>
<td>Implements the Eventing mechanism used by the Comm Framework</td>
<td>AndroidPubsub_apklib_0.4</td>
</tr>
<tr>
<td>utils</td>
<td>Some utilities features</td>
<td>SocietiesAndroidUtilities_apklib_0.4</td>
</tr>
<tr>
<td>Master GUI</td>
<td>This includes the html, CSS and Javascript code of the Android GUI. This code is using the PhoneGap framework</td>
<td>MasterGUI_apklib_0.4</td>
</tr>
</tbody>
</table>

Table 3: Components of the Light Node
4 Distribution of the release 0.4.1

The software distribution consists of the SOCIETIES platform for Domain Authority and Cloud Nodes (J2EE software running on Virgo container) and for the Light Node (Android version).

Instructions, manuals, how to use the software can be found on the redmine wiki of the project. In order to get these instructions, register first to the redmine societies server:

https://redmine.ict-societies.eu/account/register

then go to the following wiki page.


4.1 Cloud Node on Virgo

The distribution can be found in the following archive
SOCIETIES-Virgo-0.4.1.RELEASE.zip

- to be downloaded from here:

http://svn.ict-societies.eu/svn/ict-societies.eu/External_Access/Project_Deliverables/D76v2/

4.1.1 External Software

The following 3P software distributions are mandatory to run the release 0.4. Virgo is included in the Societies Platform distribution to ease first installation.

- **Application Server**: Virgo Tomcat Server 3.0.2
  http://www.eclipse.org/virgo/download/

- **XMPP server**: Openfire 3.7.1
  http://www.igniterealtime.org/downloads/index.jsp#openfire

- **Java**: Sun/Oracle Java - JDK 6 Update 29 (not Java 7)

- **Build System**: Maven Version 3.0.3 (only useful for build the platform from the source code)
  http://maven.apache.org/download.html

- **Developer IDE**: Eclipse 3.7 Indigo comes with the required eGit
  http://www.eclipse.org/downloads

4.2 The Light Node on Android

The distribution can be found in the following archive
SOCIETIES-Android-0.4.1.RELEASE.zip

- to be downloaded from here:

http://svn.ict-societies.eu/svn/ict-societies.eu/External_Access/Project_Deliverables/D76v2/
5 Tests Organisation

The tests of the platform have followed the principles of deliverable D71: unit testing, integration tests, functional and behavioural tests, then at last the robustness, performance and scalability tests. The following figure describes the four steps and the test tools used for each of them. The next sections will provide explanations and the test results for each of these steps.

This process has demonstrated its force and efficiency. As a majority of potential issues have been solved by developers during unit and integration testing, the functional testing phase has been the time to cover real functional integration problems. Among the most complex issue, we can quote the sharing service feature where not less than 7 components is involved (Service Life Cycle, CIS management, Domain Authority Rest, Security, Privacy, Comm Framwork).

Figure 4: The four steps of integration and testing
6 Unit Tests Report

6.1 Tests Coverage

The unit testing phase has been a continuous work performed by developers during their code development. JUnit library and Mockito package have been used by all to create powerful unit tests. Mockito package has demonstrated its power by avoiding the manual creation of stubs.

Using the cobertura to identify which parts of the Java code are lacking test coverage, the night build process provides a permanent coherent code, not only in terms of compilation, but in terms of unit tests.

The table below provides the coverage breakdown per package for the platform. In average, 62% for the classes are covered by unit tests.

<table>
<thead>
<tr>
<th>Name (org.societies.Name)</th>
<th>Classes</th>
<th>Files</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>utilities.DBC</td>
<td>100%</td>
<td>100%</td>
<td>89%</td>
</tr>
<tr>
<td>useragent.decisionmaking</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>useragent.conflict</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>slm.servicediscovery</td>
<td>50%</td>
<td>50%</td>
<td>33%</td>
</tr>
<tr>
<td>slm.servicecontrol</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>slm.commsmanager</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>security.storage</td>
<td>50%</td>
<td>50%</td>
<td>21%</td>
</tr>
<tr>
<td>security.policynegotiator.xml</td>
<td>100%</td>
<td>100%</td>
<td>18%</td>
</tr>
<tr>
<td>security.policynegotiator.sla</td>
<td>60%</td>
<td>100%</td>
<td>35%</td>
</tr>
<tr>
<td>security.policynegotiator.requester</td>
<td>20%</td>
<td>25%</td>
<td>32%</td>
</tr>
<tr>
<td>security.policynegotiator.provider</td>
<td>40%</td>
<td>40%</td>
<td>36%</td>
</tr>
<tr>
<td>security.digsig.util</td>
<td>33%</td>
<td>33%</td>
<td>13%</td>
</tr>
<tr>
<td>security.digsig.main</td>
<td>33%</td>
<td>33%</td>
<td>34%</td>
</tr>
<tr>
<td>security.comms.policynegotiator</td>
<td>60%</td>
<td>75%</td>
<td>41%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.repo.model.hibernate</td>
<td>100%</td>
<td>100%</td>
<td>63%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.repo.model</td>
<td>80%</td>
<td>80%</td>
<td>86%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.repo</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.evidence.repo.model.hibernate</td>
<td>100%</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.evidence.repo.model</td>
<td>75%</td>
<td>75%</td>
<td>81%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.evidence.repo</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.evidence.monitor</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.evidence</td>
<td>67%</td>
<td>100%</td>
<td>53%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.event</td>
<td>100%</td>
<td>100%</td>
<td>64%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.engine.util</td>
<td>100%</td>
<td>100%</td>
<td>67%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.engine</td>
<td>54%</td>
<td>100%</td>
<td>42%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.common.hibernate.event</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>privacytrust.trust.impl.common.hibernate</td>
<td>100%</td>
<td>100%</td>
<td>59%</td>
</tr>
<tr>
<td>privacytrust.trust.impl</td>
<td>67%</td>
<td>100%</td>
<td>33%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager.monitoring.ppn</td>
<td>100%</td>
<td>100%</td>
<td>67%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager.monitoring.ids</td>
<td>100%</td>
<td>100%</td>
<td>43%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager.monitoring.dobf</td>
<td>100%</td>
<td>100%</td>
<td>14%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager.monitoring</td>
<td>100%</td>
<td>100%</td>
<td>60%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager.merging</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Module</td>
<td>Coverage 1</td>
<td>Coverage 2</td>
<td>Score</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager.management</td>
<td>100%</td>
<td>100%</td>
<td>36%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager.evaluation</td>
<td>57%</td>
<td>80%</td>
<td>53%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypreferencemanager</td>
<td>50%</td>
<td>67%</td>
<td>31%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypolicy.registry</td>
<td>100%</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypolicy.reader</td>
<td>100%</td>
<td>100%</td>
<td>55%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacynegotiation.policyGeneration.provider</td>
<td>80%</td>
<td>100%</td>
<td>83%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacynegotiation.policyGeneration.client</td>
<td>33%</td>
<td>33%</td>
<td>7%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacynegotiation.negotiation.client.data</td>
<td>83%</td>
<td>100%</td>
<td>58%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacynegotiation.negotiation.client</td>
<td>67%</td>
<td>67%</td>
<td>73%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacynegotiation</td>
<td>50%</td>
<td>50%</td>
<td>73%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.privacypolicy.model</td>
<td>100%</td>
<td>100%</td>
<td>38%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.identity</td>
<td>67%</td>
<td>67%</td>
<td>40%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.dataobfuscation.obfuscator.util</td>
<td>75%</td>
<td>75%</td>
<td>44%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.dataobfuscation.obfuscator</td>
<td>43%</td>
<td>43%</td>
<td>55%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.dataobfuscation</td>
<td>100%</td>
<td>100%</td>
<td>75%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.datamanagement</td>
<td>100%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.assessment.util</td>
<td>100%</td>
<td>100%</td>
<td>72%</td>
</tr>
<tr>
<td>privacytrust.privacyprotection.assessment.log</td>
<td>100%</td>
<td>100%</td>
<td>78%</td>
</tr>
<tr>
<td>platform.TwitterConnector.model</td>
<td>100%</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>platform.TwitterConnector.impl</td>
<td>50%</td>
<td>50%</td>
<td>16%</td>
</tr>
<tr>
<td>platform.socialdata.service</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>platform.socialdata.converters</td>
<td>33%</td>
<td>35%</td>
<td>22%</td>
</tr>
<tr>
<td>platform.socialdata</td>
<td>33%</td>
<td>50%</td>
<td>47%</td>
</tr>
<tr>
<td>platform.sns.connector.googleplus</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>platform.servicelifecycle.serviceRegistry.model</td>
<td>100%</td>
<td>100%</td>
<td>97%</td>
</tr>
<tr>
<td>platform.servicelifecycle.serviceRegistry</td>
<td>100%</td>
<td>100%</td>
<td>95%</td>
</tr>
<tr>
<td>platform.FoursquareConnector.impl</td>
<td>50%</td>
<td>50%</td>
<td>21%</td>
</tr>
<tr>
<td>platform.FacebookConn.impl</td>
<td>50%</td>
<td>50%</td>
<td>31%</td>
</tr>
<tr>
<td>personalisation.UserPreferenceManagement.impl.monitoring</td>
<td>100%</td>
<td>100%</td>
<td>53%</td>
</tr>
<tr>
<td>personalisation.UserPreferenceManagement.impl.merging</td>
<td>10%</td>
<td>13%</td>
<td>3%</td>
</tr>
<tr>
<td>personalisation.UserPreferenceManagement.impl.management</td>
<td>80%</td>
<td>80%</td>
<td>53%</td>
</tr>
<tr>
<td>personalisation.UserPreferenceManagement.impl.evaluation</td>
<td>71%</td>
<td>100%</td>
<td>56%</td>
</tr>
<tr>
<td>personalisation.UserPreferenceManagement.impl</td>
<td>50%</td>
<td>50%</td>
<td>21%</td>
</tr>
<tr>
<td>personalisation.management.impl</td>
<td>50%</td>
<td>33%</td>
<td>35%</td>
</tr>
<tr>
<td>personalisation.CRISTUserIntentTaskManager.impl</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>personalisation.CRISTUserIntentPrediction.impl</td>
<td>100%</td>
<td>100%</td>
<td>40%</td>
</tr>
<tr>
<td>personalisation.CRISTUserIntentDiscovery.impl</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>personalisation.CAUITaskManager.impl</td>
<td>33%</td>
<td>50%</td>
<td>35%</td>
</tr>
<tr>
<td>personalisation.CAUIPrediction.impl</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>personalisation.CAUIDiscovery.impl</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>orchestration.cpa.impl.comparison</td>
<td>100%</td>
<td>100%</td>
<td>67%</td>
</tr>
<tr>
<td>orchestration.cpa.impl</td>
<td>38%</td>
<td>38%</td>
<td>5%</td>
</tr>
<tr>
<td>orchestration.collector</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>domainauthority.registry</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Package</td>
<td>Coverage 1</td>
<td>Coverage 2</td>
<td>Coverage 3</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>domainauthority.groupmanager</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>css.directory.model</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>css.directory.client</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>css.directory</td>
<td>100%</td>
<td>100%</td>
<td>60%</td>
</tr>
<tr>
<td>css.devicemgmt.deviceregistry</td>
<td>100%</td>
<td>100%</td>
<td>78%</td>
</tr>
<tr>
<td>css.devicemgmt.devicemanager.impl</td>
<td>100%</td>
<td>100%</td>
<td>54%</td>
</tr>
<tr>
<td>css.cssRegistry</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>cis.manager</td>
<td>55%</td>
<td>63%</td>
<td>42%</td>
</tr>
<tr>
<td>cis.directory.model</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>cis.directory.client</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>cis.directory</td>
<td>100%</td>
<td>100%</td>
<td>39%</td>
</tr>
<tr>
<td>activity.model</td>
<td>100%</td>
<td>100%</td>
<td>60%</td>
</tr>
<tr>
<td>activity</td>
<td>40%</td>
<td>50%</td>
<td>19%</td>
</tr>
</tbody>
</table>

Average coverage

<table>
<thead>
<tr>
<th>Coverage 1</th>
<th>Coverage 2</th>
<th>Coverage 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>62%</td>
<td>65%</td>
<td>42%</td>
</tr>
</tbody>
</table>

Table 4: Coverage Breakdown by Package
6.2 Unit Test Report per module

This table is generated by the Test Bed server Jenkins for the 456th build of the Societies platform by Oct 26th. It provides the number of unit test cases per module (column Total) and the number of failures.

<table>
<thead>
<tr>
<th>Module</th>
<th>Failure</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>api.internal:societies-api-internal</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>cis:cis-manager</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>cis:CisDirectory</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>context-management:community-context-db-mgmt</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>context-management:context-broker</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>context-management:context-source-mgmt</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>context-management:user-context-db-mgmt</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>context-management:user-context-history-mgmt</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>context-management:user-context-inference-mgmt</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>css.devicemgmt:devicemanager</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>css.devicemgmt:deviceregistry</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>css:CssDirectory</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>domain-authority:DomainAuthorityRest</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>orchestration:cpa</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>personalisation:CAUDDiscovery</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>personalisation:CAUIPrediction</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>personalisation:CAUITaskManager</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>personalisation:CRISTUserIntentPrediction</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>personalisation:dianne</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>personalisation:personalisation_Manager</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>personalisation:UserPreferenceLearning</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>personalisation:UserPreferenceManagement</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>platform.servicelifecycle:serviceManagement</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>platform.servicelifecycle:serviceRegistry</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>platform:facebook_connector</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>platform:foursquare_connector</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>platform:GooglePlusConnector</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>platform:socialdata</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>platform:twitter_connector</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>privacytrust.privacyprotection:assessment</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>privacytrust.privacyprotection:identityselection</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>privacytrust.privacyprotection:privadatamanagement</td>
<td>0</td>
<td>39</td>
</tr>
<tr>
<td>privacytrust.privacyprotection:privnegotiationmanagement</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>privacytrust.privacyprotection:privacymanagememnt</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>privacytrust.privacyprotection:privpreference</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>privacytrust:trust</td>
<td>0</td>
<td>38</td>
</tr>
<tr>
<td>security:policy-negotiator</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>security:secur-commsmgr</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>security:secur-services</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>useragent:feedback</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>useragent:monitoring</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>utilities:DBC</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>org.societies:activityfeed</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

| Table 5: Number of unit tests by Module |
7 Integration Tests Report

7.1 Integration Tests Environment

A simple but efficient test environment has been created. It is based on the principles described below. The Test Environment is a Java code packaged as a bundle. The Upper and Lower tester are two part of the testing bundle to manage the access to the APIs (internal or external or remote) of the platform under tests. Differences between Upper and Lower part are derived from a OSI Layer tests where the upper part is more oriented towards application / service while the lower part is used for low-level access (driver, communication protocol).

The Core Service Bundles under Test consists in bundles working together to perform a use case of the SOCIETIES platform. These use cases should be described in terms of functional chains as described below.

Definition: A functional chain (FC) is associated to a Use Case (e.g. CreateCIS). It consists of the software bundles which are involved in the use case. We use the term bundle here intentionally. At the level of platform integration, only the high-level features of the system shall be visible and considered. For instance, the functional chain attached to the use case CreateCIS will include the bundles CIS Management, Comm Framework, Privacy Protection.

![Figure 5: The integration test environment](image)

7.2 Test Results

The following test cases have been created during the first part of the integration phase. They have been used by integrators to test their module in the context of the Virgo container when several other components have been started and installed.

This integration tests have been created using the JUnit library to reuse the comparison function and the @test annotation. The outputs of these tests are either Success, Fail, generated by the JUnit comparison function. The Test Bed environment will support the environment soon so that a majority of these test cases could be performed automatically and continuously after each nightly build.
<table>
<thead>
<tr>
<th>Test Case Name - number is a Redmine issue.</th>
<th>Test Description</th>
<th>Involved Bundles</th>
<th>Test State</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIS</td>
<td>This task aims to test the creation of a CIS. The tester is creating new CIS and check that the Id of the created CIS is not null and as well as the Privacy Policy attached to the CIS.</td>
<td>CIS Management common-data-source, Comm Framework Privacy Protection</td>
<td>success</td>
</tr>
<tr>
<td>create_cis#958</td>
<td>Same as the 958 but create six CISs. Check that the CISs have each a distinct Id and a distinct policy. Then delete 1 CIS. Check that there are still 5 active CIS.</td>
<td>CIS Management common-data-source, Comm Framework Privacy Protection</td>
<td>success</td>
</tr>
<tr>
<td>stress_test_cis</td>
<td>Test the Eventing mechanism of the communication Framework between two nodes.</td>
<td>Comm Framework PubSub Client PubSub Server</td>
<td>success</td>
</tr>
<tr>
<td>PubSub-774</td>
<td>Test the asynchronous Message mechanism of the communication Framework between two nodes.</td>
<td>Comm Framework</td>
<td>success</td>
</tr>
<tr>
<td>remote_api_calls#771</td>
<td>Test the remote APIs call mechanism of the communication Framework between two nodes.</td>
<td>Comm Framework</td>
<td>success</td>
</tr>
<tr>
<td>Context Management</td>
<td>Test the Context Management system estimates the community context for a set of CSSs. The test is creating a community Entity with various individual entities containing ctxAttributes. CtxBroker and CommunityCtxInference components trigger the community context estimation process.</td>
<td>CtxBroker, CtxInferenceManager, CommunityContextEstimation, IdentityManager</td>
<td>success</td>
</tr>
<tr>
<td>comm-ctx-estimation#1108</td>
<td>The test aims at retrieving and updating context</td>
<td>Context Manager, Comm Framework</td>
<td>success</td>
</tr>
<tr>
<td>communication-ctx-frwk#1064</td>
<td>The test aims at retrieving and updating context</td>
<td>Context Manager, Comm Framework</td>
<td>success</td>
</tr>
<tr>
<td>ctx-3pBroker#1337</td>
<td>This test will verify the functionality of the 3p CtxBroker interface. CtxEntities, CtxAttributes, and CtxAssociations will be created, updated, lookuped and retrieved. Additionally context access control rules will be evaluated.</td>
<td>Context Manager, Comm Framework</td>
<td>success</td>
</tr>
<tr>
<td>ctx-identity#1083</td>
<td>The test should verify that context model objects maintain a valid identity.</td>
<td>Context Manager, Comm Framework</td>
<td>success</td>
</tr>
<tr>
<td>RetrieveContextAs3PService#861</td>
<td>To test if a 3rd party service succeed to get context data. This test includes a privacy</td>
<td>Context Manager, Privacy Protection</td>
<td>success</td>
</tr>
</tbody>
</table>
In this test we try to get location data of the user. In the normal process, the context bundle has to contact privacy protection bundle to check permission. Then if permission is available to get location information, context has to ask privacy protection bundle to obfuscate this data.

### Service Life Cycle

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Description</th>
<th>Involved Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>cis-list-services-964</td>
<td>This test case aims at testing how to install, uninstall services.</td>
<td>Service Life Cycle, Comm Framework</td>
</tr>
<tr>
<td>cis-share-service-962</td>
<td>This test case aims at testing how to sharing services though CIS.</td>
<td>CIS Management, Service Life Cycle</td>
</tr>
<tr>
<td>install-service-713</td>
<td>This test case aims at testing installing a service including the privacy Protection check.</td>
<td>CIS Management, Service Life Cycle</td>
</tr>
<tr>
<td>start-stop-714</td>
<td>To test if a service can be started from the CSS Gui. A started service runs in the execution framework and can be used by the CSS owner. To test if a service can be stopped from the CSS Gui. A stopped service does not anymore and cannot be used by the CSS owner.</td>
<td>CIS Management, Service Life Cycle</td>
</tr>
<tr>
<td>use-service-759</td>
<td>To test if a 3P service can be used. In this test case, an example of the &quot;Calculator&quot; service usage will be done programmatically. A Web app (#760) is used to test other services more specifically.</td>
<td>CIS Management, Service Life Cycle</td>
</tr>
<tr>
<td>cis-share-service-739</td>
<td>Same as 962 but using two nodes. Two CSSs are created (CSS1 and CSS2). CSS1 creates a CIS then shares a list of services through it. CSS 2 (member of the CIS) can browser the list of services</td>
<td>CIS Management, Service Life Cycle</td>
</tr>
<tr>
<td>discover-services-728</td>
<td>To test if services of a CSS can be discovered from another CSS (Capability Tests). Involved Modules: ServiceManagement (Service Discovery), Communication Framework</td>
<td></td>
</tr>
</tbody>
</table>

### CSS Management and Device Mgt

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Description</th>
<th>Involved Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateCSSCloudNode-#755</td>
<td>To test the cloud node creation and registration to the DA node (CSS Registry).</td>
<td>CSS Management, Comm Framework, CSS Registry</td>
</tr>
<tr>
<td>get_data_from_device#899</td>
<td>To test if devices connected to the CSS can be used by a service of the CSS.</td>
<td>Device Management, CommFramework (osgi_eventing)</td>
</tr>
</tbody>
</table>

### Privacy Management

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Description</th>
<th>Involved Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>data-management-1266</td>
<td>Check the Privacy Data Manager function “check permission” and “Obfuscate Data”</td>
<td>Privacy Protection, Comm Framework, Context Mgt</td>
</tr>
<tr>
<td>negotiation-management-1264</td>
<td>This test case will test the usage of the privacy negotiation manager. In the test preamble, the tester adds a Privacy Policy. Then a negotiation is started where the user is prompted to ask his consent. After user acceptance, the Privacy Aggrement can be</td>
<td>Privacy Protection, Comm Framework, Context Mgt</td>
</tr>
<tr>
<td>Requirement</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>policy-management-1244</td>
<td>This test case will test the usage of the privacy policy and agreement manager, by checking all the APIs of the Privacy Policy Manager and the Privacy Agreement Manager.</td>
<td>Privacy Protection, Comm Framework, Context Mgt</td>
</tr>
<tr>
<td>policy-management-remote-1267</td>
<td>This test case will test the usage of the privacy policy manager remotely through the Privacy Comm group Manager.</td>
<td>Privacy Protection, Comm Framework, Context Mgt</td>
</tr>
<tr>
<td>Assessment</td>
<td>To test if Privacy Assessment logs and correlates events.</td>
<td>Privacy Protection, Comm Framework, Context Mgt</td>
</tr>
<tr>
<td>Security</td>
<td>To test if Secure Policy Negotiator completes negotiation using comms fw on local node. To test if Domain Authority REST bundle successfully adds the new service and allows downloading the service client only if properly signed by the Secure Policy Negotiator.</td>
<td>Security Policy Negotiator, DomainAuthorityRest Comm Framework, Context Mgt</td>
</tr>
<tr>
<td>Personalisation</td>
<td>To test that the Personalisation system processes a context event, evaluating models and predicting actions to be implemented by the User Agent.</td>
<td>Personalisation, User Agent, Context Mgt</td>
</tr>
<tr>
<td>Ctx-Personalisation#751</td>
<td>To test that the User Agent and appropriately stored in context history</td>
<td>User Agent, Context Mgt</td>
</tr>
<tr>
<td>monitor-services#747</td>
<td>To test the that user actions (from services) are captured by the User Agent and appropriately stored in context history database.</td>
<td>User Agent, Context Mgt</td>
</tr>
<tr>
<td>trigger-caui-prediction#1109</td>
<td>A set of user actions are monitored and stored in context history database. When the training data set reaches the predefined limit the learning process is triggered. The outcome of this process is the discovery of a user intent model which stored in context database. The user intent model is then used in order to perform prediction of future user actions.</td>
<td>Personalisation Context Mgt</td>
</tr>
<tr>
<td>trigger-user-intent-learning#749</td>
<td>To test that the initiation of a context aware user intent model learning is triggered when certain circumstances occur. A set of user actions are monitored and stored in context history database. When the training data set reaches the predefined limit the learning process is triggered. The outcome of this process is the discovery of a user intent model which stored in context database.</td>
<td>Personalisation Context Mgt</td>
</tr>
<tr>
<td>trigger-user-preference-learning#748</td>
<td>Test preference learning process properly triggered and complete correctly</td>
<td>User Agent, Context Mgt, Personalisation</td>
</tr>
</tbody>
</table>
8 Functional and Behavioral Tests Report

8.1 Organisation of the Tests

The functional and behavioural tests have been performed on site (in HWU premises) in order to be able to
test the platform with devices and equipements mandatory for the trials. A majority of tests can be
reproduced in any partners labs however. The tests process follows a scenario-based approach where several
uses cases are used to reproduce a real environment.

8.1.1 Scenario

The scenario has been created around 4 CSSs. One CSS is for the University cloud node, the 2 others are
users, one professor Emma, two students, Arthur and Mario. The Domain Authority node is separated as well
as the OpenFire XMPP server.

The picture below depicts the deployment architecture for the test scenario. Several devices are connected to
the University CSS Cloud Node. Each user has also a Light Node embedded in an Android smart phone.

Figure 6: Functional Test Deployment Architecture
8.1.2 Basic scenario

The basic scenario provides the main set of tests for the core platform components. It is described in the picture below. Only the second level of the mind map is depicted to see the tests cases. Otherwise, the tree would be too large. The last level of the tree is open only for the first branches. The leaves contain the instructions to pass the tests. The results of the tests are available in section 8.3.

![Mind Map of the functional test scenario](image)

**Figure 7: the Mind Map of the functional test scenario**

8.1.3 3P services scenario

During the two integration and testing weeks, one team was assigned to test some 3P services in parallel of the other tests. Testing applications was also an important step in term of integration as these services are using the APIs provided by the Societies platform. Moreover several platform features are only testable via 3P services, especially the personalization and user agent components.

The following picture shows the 4 services selected for this Integration Period and the Architecture (Virgo / Android) used to run them. At last, for the Virgo node, the main platform features used by these services are listed. In the next phases, the 3P developers (WP6) will use more platform features in their new releases.
Figure 8: the Mind Map of the functional test scenario

8.1.4 Advanced scenario and other tests.
As shown in the figure above, there were plans to perform specific tests of advanced features such as Personalisation, Community context, User Agent, Learning, etc. The development and integration teams have been occupied on the tests described above to be able to perform these advanced scenario tests. These tests are planned to be done in the next weeks after this deliverable. Results of these tests will be reported in the second phase.

8.1.5 Process
This sections aims at reporting about the process followed during the integration weeks. During these testing weeks, bugs discovered by the testing teams has been assigned to a developer using the redmine issues facilities with the objective to fix them before the R0.4.1 release. If the bug cannot be fixed in time, common decision has been to postpone it to the next release. So in parallel of the R0.4.1 generation, a new list of bugs or features to be implemented has been created.

Three teams have been defined in order to cover the three types of tests: Team 1 on basic scenario tests, Team 2 on 3P services tests, Team 3 on Android integration tests. The same test environment, described in figure 5, was used. To be able to have these tests performed in parallel, specific Virgo Node has been assigned for each team. In total, the tests were performed on ten user CSS nodes. Clear identified test objectives have been assigned per day for each team. A progress per day has been reported by each testing team.
Figure 9: the Definition of the functional tests process
8.2 GUI used during the tests.

The basic test scenario has been performed using the “test WebAPP GUI” on Virgo and the Android GUI on the smart phone depicted in the next figures. Note that the “test WebAPP GUI” is not the GUI to be used in the trial and designed by the T6.5. The latter GUI was not ready at time for this test period. It has been decided to continue to work the integration tests with the so-called “test WebAPP”.

The screen dumps below intend only to show some of the pages used during the tests.

Figure 10: The welcome screen on Virgo WebAPP
Figure 11: The Suggested Friend Page.

Figure 12: The Privacy Policy Page
Figure 13: The Context Management Page

Figure 14: Smart Phone, the Login page and the Welcome page
Figure 15: Smart Phone, the Suggested Friend page and the Community page
## 8.3 Test Results

The following table provides a detailed report per features on Virgo side and on Android side.

Some features have been successfully tested on Virgo (noted OK) and not on Android (NOK) or vice-versa. The reasons of failure are more due to the level of GUI integration rather than a problem in the component itself. This explains why some features are available on Virgo and not on Android, or ready / accessible via Android and not on Virgo.

Some expected features are not ready in this release R0.4.1. They are reported here as N/A and commented as ‘next release’.

<table>
<thead>
<tr>
<th>Use case</th>
<th>Sub Use Case</th>
<th>S/Use Case virgo Status</th>
<th>Comment</th>
<th>Android</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login / Configuration</td>
<td>Start DA container</td>
<td>OK</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Register Users (On DA Node and Openfire)</td>
<td>OK</td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configure Users Accounts</td>
<td>OK</td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Start Users Container</td>
<td>OK</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Login To User Account</td>
<td>OK</td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>CSS</td>
<td>Consult My CSS Profile</td>
<td>OK</td>
<td></td>
<td>N/A</td>
<td>On-going issue</td>
</tr>
<tr>
<td></td>
<td>Update My public CSS Profile</td>
<td>OK</td>
<td></td>
<td>NOK</td>
<td>On-going issue</td>
</tr>
<tr>
<td></td>
<td>Update My private CSS Profile</td>
<td>OK</td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manage My CSS Privacy Preferences</td>
<td>N/A</td>
<td>Next release</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Friends</td>
<td>Search For a Friend</td>
<td>NOK</td>
<td>Next release</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>List all non-friends from CSS Directory</td>
<td>OK</td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show a Suggested Friends</td>
<td>OK</td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show CSS Details</td>
<td>N/A</td>
<td>Next release</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show Trust Level Of a CSS</td>
<td>N/A</td>
<td>Next release</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show My CSS Friends Details</td>
<td>N/A</td>
<td>Next release</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show CSS Activity Feeds</td>
<td>N/A</td>
<td>Next release</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show Shared services List of the non Friend CSS</td>
<td>N/A</td>
<td>Next release</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show Shared services List of the Friend CSS</td>
<td>N/A</td>
<td>Next release</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Add Friends</td>
<td>Invite a CSS To Be My Friend from a Searched CSS list</td>
<td>N/A</td>
<td>Next release</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Invite a CSS To Be My Friend from a Suggested CSS list</td>
<td>OK</td>
<td></td>
<td>N/A</td>
<td>Next release</td>
</tr>
<tr>
<td></td>
<td>Cancel a Friend's invitation sent to a CSS</td>
<td>Ok</td>
<td></td>
<td>N/A</td>
<td>Next release</td>
</tr>
<tr>
<td></td>
<td>Accept Friend's Invitation</td>
<td>Ok</td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Decline Friend's Invitation</td>
<td>Ok</td>
<td></td>
<td>N/A</td>
<td>Next release</td>
</tr>
<tr>
<td></td>
<td>Remove a friend</td>
<td>Ok</td>
<td></td>
<td>N/A</td>
<td>Next release</td>
</tr>
<tr>
<td>Community</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add membership Criteria to The CIS</td>
<td>Next release</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
</tr>
<tr>
<td>Add Privacy Policy To The CIS</td>
<td>Next release</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
</tr>
<tr>
<td>Attribute a Category To a CIS</td>
<td>NOK</td>
<td>OK</td>
<td>On-going issue</td>
<td>N/A</td>
<td>Next release</td>
</tr>
<tr>
<td>Browse CIS Lists</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search For a CIS</td>
<td>Next release</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
</tr>
<tr>
<td>Show Suggested CISs</td>
<td>Next release</td>
<td>NOK</td>
<td>On-going issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show CIS Directory</td>
<td>OK</td>
<td>OK</td>
<td>On-going issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show my CISs</td>
<td>OK</td>
<td>OK</td>
<td>On-going issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show My CIS (admin) members list</td>
<td>Ok</td>
<td>OK</td>
<td>On-going issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show CIS members list (for members)</td>
<td>OK</td>
<td>NOK</td>
<td>On-going issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show CIS members list (for not members)</td>
<td>Next release</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
</tr>
<tr>
<td>Show My CIS (admin) Privacy Policy</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show CIS Privacy Policy (Member)</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show CIS Privacy Policy (not Member)</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show CIS Activity Feeds</td>
<td>Next release</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
</tr>
<tr>
<td>Add data to CIS Activity Feeds</td>
<td>Next release</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
</tr>
<tr>
<td>Show CIS Trust Level</td>
<td>Next release</td>
<td>OK</td>
<td>N/A</td>
<td>Next release</td>
<td></td>
</tr>
</tbody>
</table>

**Join a CIS**

| Join CIS | OK | NOK | On-going issue |
| Add a member to a CIS | OK | N/A | Next release |
| Invite To Join CIS | Next release | OK | N/A | Next release |
| Accept Invitation To Join CIS | Next release | OK | N/A | Next release |
| Decline Invitation to Join CIS | Next release | OK | N/A | Next release |
| Reject Request to Join a CIS (By CIS Admin) | Next release | OK | N/A | Next release |
| Reject Request to Join a CIS (membership criteria doesn’t match) | Next release | OK | N/A | Next release |
| Reject Request to join a CIS (Privacy negotiation stop the process) | OK | NOK | Next release |
| Receive Notifications | Next release | OK | N/A | Next release |

**Leave a CIS**

| Notifications? | Next release | OK | N/A | Next release |
| Banish A Member From a CIS | Next release | OK | N/A | Next release |
| Attach SN | Next release | OK | N/A | Next release |

| Connect a Facebook account to a CSS | OK | OK | N/A | Next release |
| Connect a Twitter account to a CSS | OK | OK | N/A | Next release |
| Connect a FourSquare account to | OK | OK | N/A | Next release |
|.apps| get and display facebook profile | ok | nok |
|apps| get and display twitter profile | nok | on-going issue | nok |
|apps| get and display foursquare profile | nok | on-going issue | nok |
|apps| install a 3p service | ok | manual install | n/a |
|apps| install a 3p service using gui | ok | next release | n/a |
|apps| stop 3p service | ok | n/a | next release |
|apps| start a 3p service | ok | n/a | next release |
|apps| uninstall a 3p service | ok | next release | n/a |
|apps| show "my" 3p services list | ok | ok |
|apps| share a 3p service through a cis | ok | ok |
|apps| notify that a new service is shared in one of my cis | ok | n/a | next release |
|apps| show shared 3p services list for a given cis | ok | ok |
|apps| install a shared 3p service | ok | n/a | next release |
|apps| stop a shared 3p service | ok | n/a | next release |
|apps| start a shared 3p service | ok | n/a | next release |
|apps| uninstall a shared 3p service | ok | next release | n/a |
|apps| unshare a shared 3p service | ok | next release | n/a |
|apps| show privacy policy of a 3p service (from my apps) | ok | n/a | next release |
|apps| show privacy policy of my 3p service (from > my apps or my communities > my services or my communities > shared services) | ok | n/a | next release |
|apps| show privacy policy of a 3p service (from > my communities > my services)(not mine) | ok | n/a | next release |
|apps| show privacy policy of a 3p service (from > my communities > shared services)(not mine) | ok | n/a | next release |
|use 3p services| use near me service | ok |
|use 3p services| use context-aware wall service | ok |
|use 3p services| use my tv service | ok |
|use 3p services| use cobrowse service | ok |
|manage devices| connect devices | ok | n/a | next release |
|manage devices| disconnect devices | ok | n/a | next release |
|manage devices| show devices list | ok | ok |
|manage devices| share a device | ok | n/a | next release |
|manage devices| unshare a device | ok | n/a | next release |
|manage devices| install a shared device | ok | next release | n/a |
|manage devices| uninstall a shared device | ok | next release | n/a |
|use activity feeds| available to 3p services | ok |
9 Conclusion

This deliverable has outlined the description of the first integrated prototype of the SOCIETIES platform and the tests performed on it within work package 7. The aim of the SOCIETIES platform integration was to manage the roadmap of integrating components. After components development and unit testing, integration testing took place driven by four milestones from April to October 2012.

The outcome of this process is a release of the platform including all main components of the platform running on Virgo, which is the container on which the Cloud nodes will run. The client part for Android is now released also in this version 2 of the deliverable.

The design of the Societies Platform performed by WP4 and WP5 enabled the team to perform the integration task in a managed way. Virgo using Spring and OSGi framework proposed a very powerful tool approach for integration of a large and complex system.

The testing process (unit test, integration, functional and behavioural tests) has demonstrated its force and efficiency. As a majority of potential issues have been solved by developers during unit and integration testing, the functional testing phase has been the time to cover real functional integration problems.

The incremental test approach defined and used by WP7 should enable to start each of the trials with controlled integration deployment pilots. The WP7 team has already make tests in situ for the student trial, this should limit the troubles inherent to pervasive systems.

The next phase in T7.1 will be pursue the integration of the new features and to perform further testing cases in order to address the scalability, performance and robustness tests.