D6.1 Requirements and system architecture

Deliverable Co-ordinator: France Lasfargues
Deliverable Co-ordinating Institution: Internet Memory Foundation

Other Authors: Chato Castillo, Efstratios Tzoannos, Florent Carpentier

Other Authors: Chato Castillo, Efstratios Tzoannos This is an abstract of generic template for the ARCOMEM project, which shall contain all the necessary styles to help you produce a reasonably and consistently looking deliverable.
### ARCOMEM Consortium

This document is a part of the ARCOMEM research project funded by the ICT Programme of the Commission of the European Communities by the grant number ICT-2009-270239. The following partners are involved in the project:

<table>
<thead>
<tr>
<th>The University of Sheffield (USFD) – Coordinator</th>
<th>Leibniz Universität Hannover (LUH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department of Computer Science</td>
<td>Forschungszentrum L3S</td>
</tr>
<tr>
<td>Regent Court</td>
<td>Appelstrasse 9a</td>
</tr>
<tr>
<td>211 Portobello</td>
<td>30169 Hannover</td>
</tr>
<tr>
<td>Sheffield, S1 4DP</td>
<td>Germany</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Contact person: Thomas Risse</td>
</tr>
<tr>
<td>Contact person: Hamish Cunningham, Wim Peters</td>
<td>E-mail address: <a href="mailto:risse@L3S.de">risse@L3S.de</a></td>
</tr>
<tr>
<td>E-mail address: <a href="mailto:arcomem-coord@lists.dcs.shef.ac.uk">arcomem-coord@lists.dcs.shef.ac.uk</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Yahoo Iberia SLU (YIS)</th>
<th>Internet Memory Foundation (EA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avinguda Diagonal 177, 8th floor, Barcelona,</td>
<td>45 ter rue de la revolution</td>
</tr>
<tr>
<td>08018, CAT, Spain</td>
<td>93100 Montreuil</td>
</tr>
<tr>
<td>Contact person: Alejandro Jaimes</td>
<td>France</td>
</tr>
<tr>
<td>E-mail address: <a href="mailto:ajaimes@yahoo-inc.com">ajaimes@yahoo-inc.com</a></td>
<td>Contact person: Julien Masanes</td>
</tr>
<tr>
<td></td>
<td>E-mail address: <a href="mailto:julien@internetmemory.org">julien@internetmemory.org</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>University of Southampton (SOTON)</th>
<th>Athens Technology Center (ATC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Room 4011, Building 32</td>
<td>10, Rizarioi Street</td>
</tr>
<tr>
<td>Highfield campus</td>
<td>15233, Halandri</td>
</tr>
<tr>
<td>University of Southampton</td>
<td>Athens, Greece</td>
</tr>
<tr>
<td>SO17 1BJ</td>
<td>Contact person: Dimitris Spiliotopoulos</td>
</tr>
<tr>
<td>Contact person: Paul Lewis</td>
<td>E-mail address: <a href="mailto:d.spiliotopoulos@atc.gr">d.spiliotopoulos@atc.gr</a></td>
</tr>
<tr>
<td>E-mail address: <a href="mailto:phil@ecs.soton.ac.uk">phil@ecs.soton.ac.uk</a></td>
<td></td>
</tr>
</tbody>
</table>

| ATHENA Research and Innovation Center in        | Telecom ParisTech (IT)            |
| Information Communication & Knowledge           | 46 rue Barrault                   |
| Technologies (ATHENA)                           | 75634 Paris Cedex 13              |
| National Technical University of Athens          | France                            |
| School of Electrical and Computer Engineering   | Contact person: Pierre Senellart  |
| Division of Computer Science                    | E-mail address: pierre.senellart@telecom-paristech.fr |
| Iroon Polytechniou 9                            |                                   |
| Athens, 15780                                   |                                   |
| Greece                                          |                                   |
| Contact person: Nectarios Koziris               |                                   |
| E-mail address: nkoziris@imis.athena-innovation.gr |                               |

| Deutsch Welle (DW)                              | SUDWESTRUNDFUNK (SWR)             |
|                                                 | Hans-Bredow-Strasse,              |
| Neue Medien / Distribution                      | D-76522 Baden-Baden               |
| Voltastr. 6                                     | Germany                           |
| 13355 Berlin, Germany                           | Contact person: Robert Fischer    |
| Germany                                         | E-mail address: robert.fischer@swr.de |
| Contact person: Birgit Gray                     |                                   |
| E-mail address: birgit.gray@dw-world.de         |                                   |

<table>
<thead>
<tr>
<th>HELLENIC PARLIAMENT (HEP)</th>
<th>PARLAMENTSDIREKTION (AUP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amalias 14, 10557, Athens, Greece</td>
<td>Dr. Karl Renner-Ring 3</td>
</tr>
<tr>
<td>Contact person: Dimitris Koryzis</td>
<td>A-1017 Vienna</td>
</tr>
<tr>
<td>E-mail address: <a href="mailto:dkoryzis@parliament.gr">dkoryzis@parliament.gr</a></td>
<td>Contact person: Guenther Schefbeck</td>
</tr>
<tr>
<td></td>
<td>E-mail address: <a href="mailto:guenther.schefbeck@parlament.gv.at">guenther.schefbeck@parlament.gv.at</a></td>
</tr>
</tbody>
</table>
Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts:

Internet Memory Foundation (IM)
Yahoo Iberia SL (YIS)
Athens technology center SA (ATC)

Change Log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Amended by</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>01-07-2011</td>
<td>France Lasfargues</td>
<td>Initial detail outline</td>
</tr>
<tr>
<td>0.2</td>
<td>18-07-2011</td>
<td>France Lasfargues</td>
<td>Update User requirements</td>
</tr>
<tr>
<td>0.3</td>
<td>19-07-2011</td>
<td>Chato Castillo</td>
<td>Update executive summary and mock-ups</td>
</tr>
<tr>
<td>0.4</td>
<td>10-08-2011</td>
<td>Dimitris Spiliotopoulos, EfstratiosTzoannos,</td>
<td>Update Technical assessments</td>
</tr>
<tr>
<td>0.5</td>
<td>16-08-2011</td>
<td>France Lasfargues, Julien Masanès</td>
<td>General review</td>
</tr>
<tr>
<td>V1.0</td>
<td>17-08-2011</td>
<td>France Lasfargues</td>
<td>Final release for Review</td>
</tr>
<tr>
<td>V1.0</td>
<td>28/08/2011</td>
<td>Katerina Doka, Yannis Stavrakas</td>
<td>Internal review</td>
</tr>
<tr>
<td>V1.0</td>
<td>30/08/2011</td>
<td>Wilfried Runde</td>
<td>Internal review</td>
</tr>
<tr>
<td>V1.1</td>
<td>27/09/2011</td>
<td>France Lasfargues, Chato Castillo, Florent Carpentier</td>
<td>Final version</td>
</tr>
</tbody>
</table>

Executive Summary

This document outlines the open preservation platform that we plan to build during the project. We start by discussing the limitations of automatic crawling, particularly with respect to unexpected events of broad social significance. In these cases, automatic software may be insufficient for the task of identifying relevant content items. The open preservation platform we propose builds upon two existing paradigms for online collaboration: social bookmarking systems and publicly-editable pages. In a sense, the platform is a social bookmarking system as it allows users to add and tag resources available on the web; however, it departs from traditional systems of this type as it allows multiple views of the data to coexist and allows everyone to edit these views. We present a general description of how this system may be implemented on top of a free/open software platform for social networking and mock-ups to depict the functionality provided to users.
# Table of Contents

ARCOMEM Consortium ................................................................................................................. 2

1. Role of the OPP (Open Preservation Platform) ......................................................................................... 5
   1.1 Selecting Sources .......................................................................................................................... 5
   1.2 OPP Reputation and acquisition ...................................................................................................... 5

2. OPP Requirements ................................................................................................................................. 7
   2.1 General Interaction requirement ........................................................................................................ 7
       2.1.1 Platform collaboration model overview .................................................................................... 7
       2.1.2 Interaction of users and contents ............................................................................................... 7
       2.1.3 Interactions among users ............................................................................................................. 8
       2.1.4 Community .................................................................................................................................. 8
   2.2 Interaction functionalities ................................................................................................................... 9
       2.2.2 Resource: Validated view ........................................................................................................... 9
       2.2.3 Resource: Community view ....................................................................................................... 10
       2.2.4 Resource: Personal view ............................................................................................................ 11
       2.2.6 Tag: Community view ............................................................................................................... 12

3. State of the art ........................................................................................................................................... 14
   3.1 Elgg: An open Open Preservation Platform ....................................................................................... 14
       3.1.1 Access controls .......................................................................................................................... 14
   3.2 Current functionalities of Elgg used in OPP ....................................................................................... 15
   3.4 Technical assessments ...................................................................................................................... 17
       3.4.1 Installation Requirements ......................................................................................................... 17
       3.4.2 Performance and Stability ........................................................................................................ 17
       3.4.3 Scalability .................................................................................................................................. 17
       3.4.4 Data Structures .......................................................................................................................... 18
       3.4.5 Plugin development ................................................................................................................... 18
       3.4.6 RESTful API .............................................................................................................................. 18
       3.4.7 Security .................................................................................................................................... 18

4. Architecture and development ............................................................................................................... 20
   4.1 OPP in Arcomem Architecture ......................................................................................................... 20
   4.2 Plugins development ....................................................................................................................... 20
   4.3 Plan ................................................................................................................................................ 21

## List of figures

Figure 1: Mock-up: resource .................................................................................................................... 10
Figure 2: Mock-up - Resource - Community view ................................................................................. 10
Figure 3: Mock-up - Resource - Personal View ....................................................................................... 11
Figure 4: Mock-up – Tag - Validated view ................................................................................................. 12
Figure 5: Mock-up - Tag - Community view ........................................................................................... 13
1. Role of the OPP (Open Preservation Platform)

1.1 Selecting Sources

Capturing an event raises several challenges when it comes to creating a high quality and representative collection. Multiplication of sources, actors, and producers makes the selection of sources a difficult task. At the same time, the issue of information relevance has to be taken into account. Facing content diversity and heterogeneity of sources, how can we define a website reputation? Finally, the level of stability (reputation and update) has to be considered and may influence acquisition and collection decisions.

The experience of Institutions in charge of digital preservation points out the need for a new tool, which would help them fulfil their selection and preservation mission. In this context, the idea of a Open Preservation Platform emerges as a new paradigm for archivists to select and observe online events related to a specific topic.

The Open Preservation Platform gathers a community composed of experts or non-experts, sharing the objective of digital preservation. The goal of this platform is to collaborate in the selection and preservation of digital content by crowdsourcing discovery and appraisal of content to preserve, and/or to create corpora on specific contents.

From the users' perspective, the platform will start from two well-known paradigms: social bookmarking and wikis. Social bookmarking services such as Bibsonomy or Delicious, allow users to bookmark, annotate, and share content of interest. Wikis, such as the Wikipedia, allow users to collaboratively and concurrently edit content.

The platform will be supported by a rich interaction model and by technical components to help a community collaborate with an archivist in the creation of collections. These components include features for tracking reputation of individuals, advanced concurrent versioning for communities, and automatic conflict resolution based on reputation.

These technologies support the archivist in arriving to a final choice on which objects should be preserved and how they should be annotated, and allow to launch crawls with the ARCOMEM crawler starting from such lists of selected URLs.

1.2 OPP Reputation and acquisition

The main capabilities that the OPP will contribute to Arcomem are:

1. The capacity to find more resources about a topic than what an automatic crawler or a single archivist can find.
2. The capacity to get a manual appraisal of web sites.
3. The capacity of involving a large community in the annotation and rating of such resources.

On the web, any publicly editable area is exposed to abuse, including vandalism (e.g. replacing a publicly writeable page with empty or meaningless content). Most current systems attempt to reduce the negative effects of abuse by using a combination of community guidelines/policies as well as technological measures.
As described in the “Community” section below, we expect that OPP contributors will go from casual visitors (less trusted) to regular community members (trusted) and archivists (fully trusted). Each user will be assigned a trust level that will be a function of several factors, including:

- The time passed elapsed since the users' registration or first login in the system.
- The number of new items contributed.
- The number of existing items annotated with tags or descriptions.
- The quality score computed for contributed items by the Social Web Analysis module (see the “OPP in Arcomem Architecture” section below).
- The degree of similarity of this user's annotation with those of other users.
- The degree of similarity of this user's annotation with those validated by the archivist.
- The number of contributions of this user that have been flagged by other users for moderation (e.g. flagged as spam, or duplicate).

This type of trust score is commonly referred as "karma" in popular community web sites such as Reddit and Slashdot, among others.

The score will be computed periodically by an offline process that will also compute the trust of content items by a mutual-reinforcement process ("good" information bits are those provided by "good" users, and vice-versa).

The trust level of content items will be used to produce the “community view” of a resource. User contributions that do not have a high level of trust will only be visible in the “individual view” of a resource.

As a way of encouraging high-quality contributions, some of the aspects used in the computation of the users' trust score will be publicly-visible in a sort of leader board (e.g. number of contributed items, or number of contributed items accepted for inclusion in an archive).
2. OPP Requirements

This part presents the interaction requirements; the end-user requirements will be developed in section 3.

Definitions

Resource: anything on the Web that can be uniquely identified by a URL.

View: the state of a resource as seen at one of the following levels: personal, community, and public. Each level may have different views over the same resource: user sees only her own annotations, in the personal view community sees aggregated annotations based on the users’ input, and the public level views the resources with annotations validated by an authorized user (e.g., an archivist).

Collection: a set of resources sharing a common characteristic. In most cases given ARCOMEM’s application scenario, the characteristic in common is pertaining to a specific event.

Archive: a Collection in which the View of each resource is fixed.

Validation: the process by which an archivist determines which Resources are members of the Collection.

2.1 General Interaction requirement

2.1.1 Platform collaboration model overview

The platform is based on collaborative actions. Policies and guidelines are developed by the community to describe best practices and clarify principles.

We will start by describing the basic interaction between a single user and a collection, and then describe the interactions among users.

2.1.2 Interaction of users and contents

The basic operations for users are to create and annotate URLs, put in her/his personal view.

Annotations may include a title and a short description of the resource. Particular kinds of annotations are tags: labels associated to resources.

The ARCOMEM database is comprised of two main parts: the object store (content fetched from the web and extracted objects, along with some basic meta data like MIME type) and knowledge base (meta data derived from analysis modules associated to URLs and more semantic information).

Using OPP, the archivist will be able to refer to a collection from the ARCOMEM database, in order to add annotations. When reading data about a resource from ARCOMEM, the following
information will be used, if available for that resource: the title, description, annotations (including social web analysis annotations) and any numeric score(s) associated to the resource, e.g. crawling priorities.

OPP will allow to perform the following updates on the database:

- Annotate existing items by using the archivist-validated, community-provided tags and descriptions.
- Add new items that are validated by the archivist.
- Remove items whose deletion has been validated by the archivist.

2.1.3 Interactions among users

As in a user-editable wiki, resources are not “owned” by those users who posted them, but can be edited by other users. Any user can edit any resource.

Unlike current wikis, however, in our platform there may be multiple views over a resource. In particular, the archivist may choose to validate a resource, indicating agreement with its current annotations. The archivist may also choose to validate a tag, indicating that the current resources annotated with a tag form a valuable collection.

The archivist may repeat this validation afterwards, being assisted by the platform in order to accept selectively the changes made by the community to the state of the resource or the tags.

The appraisal of the work done by the user is supported by technological mechanisms including:

1. Mechanisms that keep track of the reputation of individuals

2. Advanced concurrent versioning for communities:

   a. Multiple annotations of the same resource can be described as “branches”.
   
   b. Users by default “merge” automatically changes done by other users or by archivists.

3. The process of validation is akin to updating the “trunk” version; the process of merging the branches from users can be done in a semi-automatic way by having reputation-aware conflict-resolution mechanisms.

After this process, the validated URLs can be submitted to the crawler for preservation.

2.1.4 Community

As in many open Open Preservation Platforms, different levels of participation are not only expected but they are fundamental to the sustainability of the platform. We will distinguish three levels of users:

- Casual visitors: level 1 – passers by
- Community – Level 2 – partially trusted
• Archivist – level 3 – trusted

Participation and reputation levels are transparent to users. Reputation is acquired by successfully contributing resources and annotations that are validated by archivists.

Finally, in addition to the tools for working with resources, each user has her own space in which she can describe herself: localization, ages, skills, and interests. Users can have friends/followers in order to be aware of their activities in the community and be informed of what happens in the community. Users can also discuss about resources and communicate with other users.

2.2 Interaction functionalities

2.2.1 Open Preservation Platform functionalities (mock-ups)

The objective of this section is to describe some functionalities of the online preservation platform. The two key functionalities beyond what a social bookmarking tool provides are those based on reputation and reputation-aware versioning. In this section we present several mock-ups of functionalities that the platform will provide.

In addition to the functionalities of describing an item through annotations, including title, description, and tags, as presented in the following mock-ups, we will also define a special vocabulary of special tags to be used to “rate” an item. These ratings could be used for instance to express a positive or negative opinion about a resource, e.g.: “important”, “interesting”, “insightful”, etc. as positive aspects and “duplicate”, “redundant”, “biased”, “spam”, etc. as negative aspects.

A simplified version of these rating tags may have only two levels, as commonly used in several online social media sites: a yellow “star” to indicate that a resource is particularly important, and a red “flag” to indicate that there is some issue with the resource.

2.2.2 Resource: Validated view

When examining a resource, the default view is the validated view. An archivist that reviews a resource does the validation.

In the mock-up (Figure 1), a non-archivist is looking at a resource that was edited by 6 users before being validated by Joe Doe, an archivist.
2.2.3 Resource: Community view

The **community view** represents the consensus of the community. It is created by showing the resource as the result of the edits by community members, discarding automatically edits with very low-quality, selecting among conflicting titles and descriptions the ones to show (but allowing users to see alternatives), and showing all the tags except those contributed by untrusted users or found to be spam.

In the mock-up, (Figure 2) a user has switched to the community view. There are 12 users who contributed to this view (e.g. 6 that contributed before the validation, and 6 that contributed afterwards). There are a number of tags, some of them contributed by more than one user.
2.2.4 Resource: Personal view

The personal view of a resource represents the edits of the user that is looking at the resource. The user may decide to either (1) automatically accept changes (e.g. new tags) from the validated or community versions -or- (2) keep a personal view detached from those changes. This is done by the process of "syncing" the personal view with either the validated or the community view.

In the mock-up (Figure 3), the personal view of the article is not synchronized with that of the community, so it will stay "static" unless the user choose to "sync" this view.

![Mock-up - Resource - Personal View](image)

2.2.5 Tag: Validated view

The validated view contains the resources attached to a tag according to an archivist.

In the mockup (Figure 4), a tag for the "spanish elections 2010" (which defines a collection of the same name) has been reviewed by Jane Doe on 4/June. The collection contains 2 newspapers article, 1 blog and 1 video.
2.2.6 Tag: Community view

The community view contains all the resources that have been tagged by the community as belonging to a collection. The order could be based on the number of visits of each resource or on the reputation of those who created each resource.

In the mockup (Figure 5), a tag for the “spanish elections 2010” (which defines a collection of the same name) has been reviewed by Jane Doe on 4/June. The top elements are 3 videos and 1 blog post. Several tags that have not been validated by the archivist (yet?) appear.
## Tag "spanish elections 2012"

Validated view | Community view | Personal view
---|---|---

Built from community edits. Not reviewed by an archivist. Based on contributions by 21 users.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Video: Demonstrators in Valencia [edit]</td>
<td>Demonstrators &quot;receive&quot; the visit of the candidate with loud protests.</td>
<td>twitter video rajoy spain protest (4)</td>
</tr>
<tr>
<td>Antena 3: Meeting of PSOE in Valencia [edit]</td>
<td>Video from congress of PSOE party in Valencia</td>
<td>video valencia spain tv conservative-tv</td>
</tr>
<tr>
<td>El País: Technically tied candidates [edit]</td>
<td>Rajoy is ahead of Rubalcaba for a 0.2% margin</td>
<td>progressive-newspaper poll spain</td>
</tr>
<tr>
<td>Jose Cárcamo: Rubalcaba in Valencia [edit]</td>
<td>Rubalcaba meets with supporters in the city of Valencia, promises to create jobs</td>
<td>blog jobs economic economy spain</td>
</tr>
</tbody>
</table>

**Figure 5: Mock-up - Tag - Community view**
3. State of the art

According to users needs a first benchmark of existing open source community management platform has been done. The Elgg platform was selected as it provides the functionalities and modularity required to power generic social network functions. As it is open-source and pretty mature, we can safely build on top of it the specific modules required for the purpose of Arcomem.

Institution, Universities and companies in various domains as cultural, education, art and market place already use Elgg. The development stage as well as the community around Elgg is the most mature we have found when doing the research.

Example:
- Courseware: Stanford University\(^1\) uses it to develop a Courseware project with a hint of social networking built -in.
- A living archive East London Lives 2011\(^2\) - East London lives 2012 is a digital archive project which aims to document some aspects of change in the lives of East Londoners towards the hosting of the 2012 Olympic and Paralympic Games.

Elgg comes with advanced user management and administration and social networking. Most of the end user functionality in Elgg comes from plugins. Main features are: dashboard, friends, groups, Profile and Widget (section 3.2.). Some optional plugins can be added: bookmarks repository, message board, activity status and micro blogging…

Elgg platform encourages developers to extend the existing functionality by implementing new Plugins. Add new plugin can be done in a easy way (section 3.4.5)

First part of this section (section 3.1) describes the technical aspects of Elgg. Current features are following (section 3.2). At least results (section 3.3) from a technical assessment are exposed. Then, section 3.3 and 3.4 provides some technical guidelines in terms of development.

3.1 Elgg: An open Open Preservation Platform

Elgg provides a powerful data model making the creation of different entities simple, yet flexible. The granular activity stream API ensures that plugins push the required content to users.

3.1.1 Access controls

All objects in Elgg can have an access control level applied making granular access permissions possible. Web Services API exposes functionality through the REST API by building a plugin and then either publishes the API for other developers to build clients or provide your own.

---

\(^1\) https://courseware.stanford.edu/
\(^2\) http://eastlondonlives2012.net/
3.2 Current functionalities of Elgg used in OPP

An installation of Elgg is available on [http://ia200107.eu.archive.org/eggl/](http://ia200107.eu.archive.org/eggl/)

Release - 1.7.8, Version - 2010071002

Elgg has a flexible architecture in which most of the functionality is served by plugins. It will be used as the base for the open preservation platform. A current set (table 1) of plugins included in a default installation proposes a lot of functionalities. According to ARCOMEM needs, all of them are not useful for the OPP.

**Table 1: List of default plug-in in Elgg**

<table>
<thead>
<tr>
<th>Messageboard:</th>
<th>This plugin allows users to put a message board on their profile for other users to post comments.</th>
</tr>
</thead>
<tbody>
<tr>
<td>messages:</td>
<td>Elgg internal messages plugin. This plugin lets user send each other messages.</td>
</tr>
<tr>
<td>Tinymce:</td>
<td>TinyMCE plugin.</td>
</tr>
<tr>
<td>Zaudio:</td>
<td>This simple plugin lets users play mp3's in the page.</td>
</tr>
<tr>
<td>Defaultwidgets:</td>
<td>This plugin enables an administrator to define a default set of widgets for new users</td>
</tr>
<tr>
<td>Crontrigger:</td>
<td>Provides a poor man's cron trigger to trigger cron scripts based on site visits.</td>
</tr>
<tr>
<td>Custom_index:</td>
<td>A simple plugin to replace the site index pages with boxes containing the latest</td>
</tr>
<tr>
<td>Friends:</td>
<td>provides friend widget to user profile</td>
</tr>
<tr>
<td>Twitterservice:</td>
<td>enable Twitter integration in supported plugins</td>
</tr>
<tr>
<td>Captcha:</td>
<td>Provides captcha support</td>
</tr>
<tr>
<td>Reported content:</td>
<td>adds the option for users to report content and for admins to check it out</td>
</tr>
<tr>
<td>Externalpages:</td>
<td>plugin let site admin populate an about pagen terms, privacy and contact</td>
</tr>
<tr>
<td>Groups:</td>
<td>provides group support for elgg</td>
</tr>
<tr>
<td>Members:</td>
<td>a simple plugin to let users browse other member</td>
</tr>
<tr>
<td>Lobrowser:</td>
<td>browse the system event log</td>
</tr>
<tr>
<td>File:</td>
<td>file browser plugin</td>
</tr>
<tr>
<td>Bookmarks:</td>
<td>elgg bookmarks plugin</td>
</tr>
<tr>
<td>Garbagecollector:</td>
<td>perform some database cleanup tasks</td>
</tr>
<tr>
<td>Embed:</td>
<td>allows users to easily upload and embed media into text areas</td>
</tr>
<tr>
<td>Invitefriends:</td>
<td>friend invite page</td>
</tr>
<tr>
<td>Search:</td>
<td>allow search across entities of the site</td>
</tr>
<tr>
<td>Notifications:</td>
<td>Elgg notification plugin</td>
</tr>
<tr>
<td>Uservalidationbyemail:</td>
<td>simple user account validation via email</td>
</tr>
<tr>
<td>Thewire:</td>
<td>provides a microblogging service for your Elgg site</td>
</tr>
<tr>
<td>Profile:</td>
<td>Elgg profile plugin</td>
</tr>
<tr>
<td>logrotate:</td>
<td>rotate the system log at specific</td>
</tr>
</tbody>
</table>

---

3 Messageboard, Custom_index, Twitterservice, Riverdashboard, Groups, Thewire, Twitther, Blog. We also will also not use the Bookmarks, plugin, but develop our own plugin for handling resources plugins will be not used on OPP
Elgg provides a powerful data model making the creation of different entities simple, yet flexible. The granular activity stream API ensures that plugins can push the required content to users.

This section provides the set of plug-in will be use in OPP

**User and profiles**

A user of the OPP is any Internet user who has interest on digital preservation. Among these users we distinguish archivist who play a special role (in addition to the standard user role): they can validate the state of a resource to put it in the validated view.

To be a member of the community, user has to register. Login and password let him to get into the OPP. Each user has a unique profile. Each profile field has its own access restriction, so users can choose who can see each of his field.

**Dashboard**

The main dashboard provides the functionalities of the OPP. The user dashboard allows the access at all the functionalities: Activity, Wire, Bookmarks, Groups.

**Activity** gives an overview of all the last changes done by the community.

**On the Wire**, user can post news and comments. User can interact with other users by posting comment on Wire.

**Group**: A group gathers users, which share similar interest in specific topics. Each group has its own URL and Profile and comes with a file repository, forum, pages, bookmarks and messageboard. All these functionalities make easier the communication between users.

Currently in Elgg, users can create and annotate bookmarks. By default, bookmarks in Elgg may include: Title, URL of the resource to bookmark, Description, and Tags. Currently, they also include a Share with and Right access functionalities.
3.4 Technical assessments

3.4.1 Installation Requirements

The Elgg platform is developed in PHP, which means that the application is cross-platform and not highly demanding in terms of hardware specifications. The prerequisites for the installation are the following as found in the installation guidelines of Elgg community website:

- MySQL 5 or greater
- PHP 5.2 or greater with the following extensions:
  - GD (for graphics processing: user and group profile icons plus many plugins)
  - Multibyte String support (for internationalisation)
  - Proper configuration and ability to send email through an MTA
- Web server with support for URL rewriting (e.g. Apache, IIS)

For testing and evaluation purposes, the installation took place in Windows environment, using Apache Web Server, PHP 5.2 and MySQL 5. The installation process has been performed following the steps described in the installation guidelines. It is a relatively easy process, although it requires some basic knowledge of the underlying technologies. Therefore, some basic knowledge about MySQL, Apache and PHP installation and configuration is necessary.

3.4.2 Performance and Stability

Although we have not performed intensive load and stress tests, the first impression about the platform is that it is relatively fast and stable. This means that we did not observe any major delays in either navigation between pages nor in the internal functionality of the modules-plugins. Furthermore, the system did not produce any error or exception after some basic testing.

3.4.3 Scalability

The current version of Elgg platform makes use of a number of techniques to improve code efficiency and avoid bottlenecks. Those are:

- Memcache - Elgg can use Memcache for storing objects, metadata and some common settings. This removes the need to retrieve this data from the database and so reduces the load on the server.
- Query caching - For the lifetime of a given page’s execution a cache of all select queries is kept. This means that for a given page load a given select query will only ever go out to the database once, even if it is executed multiple times.
- Simple cache - By default views are cached in the Elgg data directory for a given period of time. This removes the need for a view to be regenerated on every page load.
- View path cache - As well as the simple cache documented above, Elgg also takes advantage of a view path cache. The location of views are cached so that they do not have to be discovered (profiling indicated that page load took a non-linear amount of time the more plugins were enabled due to view discovery).
- .htaccess - Elgg 1.5 and SVN trunk also switches on a number of server side technologies (expires headers, etags and gzip compression to name a few).

As stated in Elgg community wiki: “Initial reports and profiling data give a very positive indication of future performance”. Some more hints are provided for further optimisation of the performance.
3.4.4 Data Structures

Each data model residing in Elgg platform consists of atomic units of data named “entities”. Every entity in the system inherits the ElggEntity class. This class controls access permissions, ownership and so on. The encapsulation of database logic is a goal of Elgg platform, which encourages developers to ignore database operations and develop the desired functionality on upper layers. This makes the system more powerful and stable by unifying all the plugin functionality in a consistent way, by means of using common generic principles.

The entities are extended with extra information with the usage of:

- **Metadata** – Additional details of the object. For example, tags, an ISBN number, file location or language information.

- **Annotations** - Pieces of data attached to an entity, generally added by third parties, for instance, comments and ratings. Annotations are stored as instances of the ElggAnnotation class. Each annotation has:
  - An internal annotation type (like comment)
  - A value (which can be a string or integer)
  - An access permission distinct from the entity it's attached to
  - An owner

3.4.5 Plugin development

Elgg platform encourages developers to extend the existing functionality by implementing new Plugins. Plugins are modules that could be integrated into Elgg platform which provide additional features. This could extend from a simple page to a fully functional module like a custom blog engine or a photo gallery. The support provided for such a task is appropriate as there is a whole set of information and guidelines about feature development.

3.4.6 RESTful API

Elgg exposes an HTTP API by means of RESTful web services. This enables developers to reuse features in external applications as well as integrate the functionality of third party applications into Elgg platform. More precisely, the API provided is a REST/RPC hybrid similar to the APIs provided by sites like Flickr and Twitter.

3.4.7 Security

- **Cross-Site Request Forgery protection**: The Elgg platform is using an anti-CSRF mechanism implemented by using tokens in its forms. These unique private tokens are generated with each session and are stored in the session data. Every time a form is created another token is created based on that private token, the session identifier, the user agent, a site secret, and the timestamp. This token and the timestamp are embedded in the form as hidden input. When the form is submitted, the submitted token is checked against what was in the form and the timestamp must be less than one hour old.

- **RESTful API authentication**: In order to control the access to the exposed RESTful methods, Elgg offers an authentication API. Two built-in methods to perform API authentication are provided: key based and HMAC signature based. The key based approach is very similar to what Google, Flickr, or Twitter. Developers can request a key (a random string) and pass that key with all calls that require API authentication. The keys are stored in the database and if an API call is made without a key or a bad key, the call is denied and an error message is returned. The HMAC Signature
Authentication [http://docs.elgg.org/wiki/HMAC_Authentication](http://docs.elgg.org/wiki/HMAC_Authentication) is similar to what is used with OAuth or Amazon’s S3 service. This involves both the public and private key. If you want to be very sure that the API calls are coming from the developer you think they are coming from and you want to make sure the data is not being tampered with during transmission, you would use this authentication method. Be aware that it is much more involved and could turn off developers when there are other sites out there with key-based authentication.

- **User Authentication**: Elgg provides a token-based approach for user authentication. It allows a user to submit their username and password in exchange for a token using the method `auth.gettoken`. This token can then be used for some amount of time to authenticate all calls to the API before it expires by passing it as the `auth_token`. If you do not want to have your users trusting their passwords to 3rd-party applications, you can also extend the current capability to use an approach like OAuth.
4. Architecture and development

4.1 OPP in Arcomem Architecture

The OPP will interact with the following ARCOMEM components:

**Intelligent Crawl Definition module**: The OPP launch crawls about specific topics. Communication with the crawler will be required: specifying seeds and other crawl parameters for the Intelligent Crawl Definition module (WP5). Fetching resources to get basic meta data in order to assign automatically some tags to a URL will require a basic download feature. This could be implemented with the WP5 crawler, or with an internal fetching module, depending on the delay requirements for OPP and the capabilities of the WP5 crawler.

**Social Web Analysis module**: The OPP will interact with the annotations provided by the Social Web Analysis module (WP2) and incorporated in the ARCOMEM database. There should be an API that allows to get these annotations and refer to them. This will be used in particular as part of the process that determines an overall quality score (e.g. number of times this URL is mentioned in other social media platforms).

**ARCOMEM knowledge base**: meta data such as MIME type, HTML page title, incoming links, analysis modules annotations will be read from this database, and all the annotations produced with OPP will be written to it.

4.2 Plugins development

The functionalities of the OPP will be provided by the plugins implemented for Elgg.

**OPP ARCOMEM DB communication plug-in**: if Elgg allow us, we will replace its MySQL interface with an **interface to the ARCOMEM knowledge base** for all ARCOMEM-specific data. This would enable direct and “real-time” communication. If Elgg’s dependency to MySQL, or relational databases in general, is too strong, we will implement an **import/export plug-in** that will be activated when an archivist decides to export or import a collection from the ARCOMEM database into OPP's local tables, and then export the collection back to the ARCOMEM database.

**OPP content score import plug-in**: this plug-in will get scores from the Social Web Analysis module in order to use them in the OPP logics (e.g., user scoring, and URL annotations).

**OPP crawler communication plug-in**: this plug-in will submit URLs to the Intelligent Crawl Definition module, to launch crawls, and possibly to fetch URLs to get meta data.

**OPP reputation plug-in**: this plug-in will re-compute the trust score of users and content items based on the information available. It will also display some of the components of this score through the user interface to users. This module will also keep track of user levels: who is an archivist, who is a community member, and who is a passer-by.

**OPP collections plug-in**: this plug-in will provide the main enhanced bookmarks functionalities with reputation-aware branching/merging, in order to provide the individual's view, a community view, and an archivist view.
4.3 Plan

We will start with the development of the most independent modules: reputation and collections. The development of the interface modules and their integration tests will be started as soon as an initial version of the interfaces to the knowledge base and the object store as well as sample content for these systems are available.

The development of crawl launching (and the interface with the crawler), being the conclusion of the process, can be scheduled for later, and can start when an initial version of the intelligent crawl definition database is agreed upon.