

DELIVERABLE

Project Acronym: EAGLE
Grant Agreement number: 325122
Project Title: Europeana network of Ancient Greek and Latin Epigraphy

The EAGLE Flagship Storytelling Application

D5.4.1

version: N° 1.0

Revision: final

Authors:

Wolfgang Schmidle (DAI)
Francesco Mambrini (DAI)

Contributors:

Vittore Casarosa (CNR-Isti)
Pietro Liuzzo (University of Heidelberg)
Claudio Prandoni (Promoter srl)

Reviewers:

László Borhy (ELTE)
Rada Varga (UBB)
Anja Ragolič (ZRC SAZU)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	x
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
0.1	21/11/2014	Francesco Mambrini, Wolfgang Schmidle	DAI	First schema of the deliverable
0.2	12/12/2014	Francesco Mambrini, Wolfgang Schmidle	DAI	Draft of the introduction
0.3	09/02/2015	Wolfgang Schmidle, Francesco Mambrini	DAI	First complete draft
0.4	23/02/2015	Wolfgang Schmidle	DAI	Minor changes, added Screenshots
1.0	04/03/2015	Wolfgang Schmidle	DAI	Last improvements, integration of the peer reviewers' comments

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS	5
EXECUTIVE SUMMARY	6
1. INTRODUCTION	7
1.1. DOCUMENT PURPOSE.....	7
1.2. DESIGN AND RATIONALE	7
1.2.1. <i>Purpose of the FSA</i>	7
1.2.2. <i>Intended audience</i>	7
1.2.3. <i>Users and stories: read, write and publish</i>	8
1.2.4. <i>Development cycle and functional requirements</i>	8
2. USER MANUAL	10
2.1. THE FSA ACCESS PAGE.....	10
2.1.1. <i>The main column</i>	10
2.1.2. <i>The sidebar</i>	11
2.1.3. <i>Searching and filtering the collection of stories</i>	12
2.2. THE STORY PAGES.....	13
2.3. THE STORY EDITOR.....	15
2.3.1. <i>The list view</i>	15
2.3.2. <i>The edit view</i>	15
2.3.3. <i>The WYSIWYG editor</i>	15
2.3.4. <i>The sidebar</i>	16
2.3.5. <i>Publishing a story</i>	17
2.3.6. <i>Searching inscriptions and other data sources</i>	17
3. DEVELOPER'S GUIDE	20
3.1. INTEGRATION WITH THE EAGLE PORTAL	20
3.1.1. <i>The "Eagle Storytelling Application" plugin</i>	20
3.1.2. <i>The "story" post type</i>	21
3.2. THE FSA ACCESS PAGE	21
3.2.1. <i>Design rationale</i>	21
3.2.2. <i>The sidebar</i>	21
3.2.3. <i>Searching and filtering stories</i>	22
3.3. THE STORY PAGES.....	22
3.4. THE STORY EDITOR.....	23
3.5. SEARCHING INSCRIPTIONS AND OTHER DATA SOURCES	23
4. CONCLUSIONS AND FUTURE PLANS.....	24
4.1. THE INTERMEDIATE VERSION 1.1	24
4.2. REQUIREMENTS FOR VERSION 2.0	24
4.3. ADDITIONAL PLANS.....	25
4.3.1. <i>Redesigning the edit page</i>	25
4.3.2. <i>Shortcodes for inscriptions</i>	25
4.3.3. <i>Where the stories are stored</i>	25
4.3.4. <i>Miscellanea</i>	26
5. REFERENCES	27



LIST OF ABBREVIATIONS

- ISE EAGLE Inscription Search Engine
FMA EAGLE Flagship Mobile Application
FSA EAGLE Flagship Storytelling Application

For the deliverables D5.1, D5.2, D5.4, D5.5 and D6.1 see the References

EXECUTIVE SUMMARY

The Deliverable 5.4.1 introduces the first release of the EAGLE Flagship Storytelling Application (FSA) to the partners and the general public.

The Introduction (Section 1) summarizes some aspects of the design and development cycle of this and other EAGLE applications that are useful to contextualize the work on the FSA.

Section 2 illustrates the three main interfaces of the FSA: the first interface allows users to access the collection of the published stories, the second interface allows users to read each story in full; the third interface provides registered users with a unique storytelling writing environment, where they can compose epigraphy-based narratives and enrich them with context from the EAGLE portal and other online repositories. Section 2 serves also as a user reference manual.

The developer's guide (section 3) details the specifics for the implementation of the FSA services and its interfaces. The FSA is implemented as a plugin of the EAGLE portal to ensure a tight integration of services.

Conclusions and future plans are discussed in Section 4.

1. INTRODUCTION

1.1. DOCUMENT PURPOSE

This document describes version 1.0 of the EAGLE Flagship Storytelling Application (**FSA**). It provides both a **user manual** (section 2) and a **developer's guide** (section 3).

In the following sections of this introduction we summarize some of the preliminary design aspects (target users, functional requirements, development and evaluation cycle) that are discussed elsewhere in detail (see especially D5.1, D5.2, D5.3.1, D5.5). Such details are presented here because they provide an indispensable context for understanding the place of the FSA within the development process of the EAGLE user-dedicated services, as well as important background information for evaluating this first release.

1.2. DESIGN AND RATIONALE

1.2.1. Purpose of the FSA

The FSA is a web-based tool designed to allow users to create multimedia narratives on epigraphic content and share them on the web. It is one of the components of the EAGLE portal (see deliverable D5.2). As such, its interfaces are designed to be integrated within the general architecture of the EAGLE portal, and they are visually coherent with the graphic identity of the EAGLE portal (see deliverable D6.1).

With the help of a friendly interface, users will be able to write texts and include content in different format into their narratives. In particular, the FSA is designed so as to help authors with special interest in epigraphy and in ancient history or archaeology to get easy access to the main resources for multimedia content on the Ancient World. With the FSA we intend to create a tool that will ultimately enable users to include images and videos from popular hosts (such as YouTube or Europeana), excerpts from Ancient texts, maps locating ancient places and, especially, the vast collection of inscriptions from the EAGLE network.

The FSA is thus a useful resource to spread the knowledge about monuments and inscriptions in the collections of the EAGLE network, or to suggest unexplored paths through themes and objects. All the users of the EAGLE services, the members of the EAGLE Best Practices Network and the whole community of persons interested in cultural heritage and history, whether specialists or amateurs, are encouraged to unleash their creativity and experiment with the potential of this tool.

1.2.2. Intended audience

The FSA is designed to assist the creativity of our public. Everyone that has a story to tell about inscriptions in the EAGLE collections is a potential user of the application; everyone is encouraged to experiment with this new form of scientific communication. In particular, however, we have in mind two areas where professionals can profit from the tool.

In education, the FSA may especially suit different professional figures and contexts. Stories featuring textual as well as visual content can be used as a support to teaching activities of lecturers and professors. For students, epigraphy-based stories may be an innovative format for essays, assignments and presentations. Scholars and researchers, on the other hand, can use the FSA as a “blogging” platform to disseminate preliminary results or, more generally, to inform the public about ongoing research activities.

Cultural heritage professionals are also strategic figures to engage. The FSA is potentially an excellent

medium that curators of museums and exhibitions can use to promote permanent collections and special events. Narratives based on single objects can leverage the multimedia capabilities of the FSA to provide information on the geographical and the historical context of each monument. Maps, photographs and passages from ancient texts (either in the original and/or in translation), and videos can be interwoven with textual explanations, so that the public can get an easier and visually stimulating access to information about the objects on display.

On the other hand, stories built as “galleries” of different monuments or inscriptions can be an attractive way to promote, advertise or introduce the public to new initiatives and exhibitions. Narratives that are easily accessible on the web and include written descriptions as well as visual content are the ideal medium to suggest possible thematic paths through collections to the attention of the audience.

1.2.3. Users and stories: read, write and publish

Examples of the possible stories that can be shared thanks to the FSA include notes centered on one specific monument, where the multimedia content can be effectively used to contextualize the object. Or stories can be used to guide the readers on new thematic journeys, by collecting and discussing all the inscriptions from a certain area, or dated to a specific era, or important for specific topics, like e.g. slavery, and so on.

On a dedicated page on the EAGLE portal, the public will have access to the stories written and published using the FSA. In addition to reading what others have shared, users can easily start to write and publish their own content. Every user will be able to register a personal account on the EAGLE portal that will grant access to the unique features of all the EAGLE User Dedicated Services (including also dedicated features of the EAGLE Portal and the Flagship Mobile Application), as discussed in the Deliverables D5.1, D5.2 and D5.3.1.

Users with registered EAGLE accounts can immediately create their own stories. The interface is designed so as to make the process of writing, searching and including multimedia content as fluent and intuitive as possible. Authors can move the retrieved objects (text excerpts, videos, images and inscriptions) by dragging and dropping them to the intended location within the story.

1.2.4. Development cycle and functional requirements

The development cycle of the user-dedicated services is outlined in D5.1. For the two flagship applications (the mobile app, on which see D5.3.1, and the FSA) partners and developers of the EAGLE consortium have scheduled two main releases:

- **First release** (version 1.0): to be completed before month 21 of the project
- **Second release** (version 2.0): to be completed before month 30 of the project

Between the two releases, the process of functional validation will take place, as discussed in the D5.5; the validation will help the developers to address the bugs that are left in the first published version of the applications in time for the second release. This two-step process is also reflected on the priority scale assigned to the functional requirements, as discussed in D5.1: 28, and D5.5: 7.

Table 1 reports the functional requirements of the FSA as defined in the D5.1 as well as their implementation status in version 1.0.

ID	Sec.	User	Requirement	Priority	Status
STE01	4.4.1	Generic	Perform full text search on stories	High	implemented
STE02	4.4.1	Generic	Browse stories with a keyword word-cloud	Medium	implemented
STE03	4.4.1	Generic	List of recently published stories	High	implemented
STE04	4.4.1	Generic	Share stories on social networks	Low	planned for v2.0 or later
STE05	4.4.2	Registered	Create new stories	High	implemented
STE06	4.4.2	Registered	Create paragraph in WYSIWYG editor	High	implemented
STE07	4.4.2	Registered	Use title and description paragraph	High	implemented
STE08	4.4.2	Registered	Retrieve and add content from the EAGLE database and from selected external repositories	High	part of v1.1
STE09	4.4.2	Registered	Retrieve and add content from the Perseus Project	Medium	planned for v2.0
STE10	4.4.2	Registered	Format of the search results	High	part of v1.1
STE11	4.4.2	Registered	Single window for the story editor interface	High	part of v1.1
STE12	4.4.2	Registered	Drag and drop content	High	part of v1.1
STE13	4.4.2	Registered	Save drafts	High	implemented
STE14	4.4.2	Registered	Publish a story	High	implemented
STE15	4.4.2	Registered	Edit, unpublish, delete published stories	High	implemented
STE16	4.4.2	Registered	Access and include saved items	Medium	planned for v2.0
STE17	4.4.2	Registered	Generic EpiDoc XML reader	Medium	planned for v2.0
STE18	4.4.2	Registered	Add tags to a story	Medium	implemented

Table 1: EAGLE Flagship Storytelling Application requirements and implementation status

Table 1 refers to an intermediate version 1.1 which will add search functionalities directly in the story editor (see the discussion in section 2.3.6). Version 1.1 will be available before the functional validation for the first release and will implement all requirements with priority “high”. Apart from the search functionalities all these requirements are already implemented in the current version 1.0. In addition, the requirements STE02 and STE18 that were originally planned only for version 2.0 are already implemented in version 1.0.

2. USER MANUAL

The EAGLE Flagship Storytelling Application (FSA) includes three interfaces:

1. The **FSA access page**, where all users (registered and unregistered) can browse and search the collection of published stories
2. The **story pages** for displaying single stories
3. The **story editor**, where authors (after logging in) can start composing their own story and edit existing stories

All three interfaces are part of the EAGLE website. The FSA will be accessible by an entry in the main menu of the EAGLE website, which will lead to the FSA access page. The FSA access page in turn contains links that lead to the story pages and to the story editor. The story pages and the story editor can also be accessed directly if the user knows the direct link.

In the following sections the functionalities of the three interfaces are described in detail. The description will refer to the requirements STE01 to STE18 that are listed in table 1 and laid out in more detail in section 4.4 of the Deliverable D5.1.

2.1. THE FSA ACCESS PAGE

The FSA access page with the title “STORIES” is the first interface of the FSA that the user will see. It has the same look as most other pages in the EAGLE website. In particular, it uses the standard header and footer for EAGLE pages, there is a main column for the content, and when the browser window is wide enough there is an additional sidebar on the right for so-called widgets that can perform functions like searching stories etc.

The display of the story list in the main column resembles the display of news items. The selection of widgets is specific to the FSA, however. The sidebar contains a general “LOGIN” widget, but it includes also some story-centric widgets.

2.1.1. The main column

The main column shows a list of all published stories, starting with the most recent ones. When the user has performed a story search (see section 2.1.3), the main column displays the search results instead.

All stories are listed in the same fashion as the news items, i.e. in a paginated list where each story is represented by its title and a short excerpt consisting of the first few lines of text with all formatting removed. Each story has a “Read more” link that leads to the story page for this story (see section 2.2). In addition, if any keywords are associated with a story they are displayed below the excerpt.

Together with the “LATEST STORIES” widget in the sidebar (see the following section) this implements requirement STE03, which states that the list of most recently published stories should include (1) a limited number of stories, e.g. between 5 and 10, and for each story the list should display (2) the title, (3) the author’s user name and (4) the text of the description paragraph. One notable difference between requirement and implementation is that the list is implemented in two versions on the FSA access page rather than in one version as it was originally envisaged in D5.1: The list in the main column implements the requirements (2), (3) and (4), while the “LATEST STORIES” widget implements the requirements (1) and (2). This allows for more consistency with the overall design of the EAGLE website, especially the News page.

STORIES

Verb ellipsis in Greek inscriptions: a research project?

Posted by Francesco Mambrini on Mar 3, 2015

In this “story” I would like to discuss a research idea that came to me while I was attending the EAGLE International Conference in Paris. I am not a specialist on the field, and what follows is just a series of notions and ideas I have developed ever since. I have always suspected that the EAGLE storytelling app could be a great environment for presenting ongoing research topics or questions; therefore, I will keep you posted about my ideas on the subject! Nominal sentences In my previous work on the syntax of ancient Greek...

[READ MORE »](#)

Inscribed Monuments in Arachne: The Votive Monument of Atarbos

Posted by Francesco Mambrini on Mar 2, 2015

One of the most fascinating (and celebrated) inscribed monument in the Arachne collection is the base of a votive offering from the Acropolis of Athens. In 1852 a basis of a group of bronze statues was discovered on the west end of the Acropolis in Athens, incorporated in the so called Beulé Gate. The basis consists of two blocks that were once clamped together. The bronze group that stood on top of it has disappeared but on the upper side of the basis it is still possible to view the traces of the footprints of the statues, both on the left...

[READ MORE »](#)

PAGE 1 OF 2 **1 2** »

Login

LATEST STORIES

Verb ellipsis in Greek inscriptions: a research project?

Inscribed Monuments in Arachne: The Votive Monument of Atarbos

Funerary Inscriptions in the current EAGLE collection

A case of cow disease in 175 BCE

SEARCH STORIES

FILTERS

Filter by keywords

Filter by author

Filter by TM ID

Figure 1: The FSA access page

If logged in, the author of the story will see an additional „Edit this story“ link next to the „Read more“ link. This link leads to the story editor for the selected story (see section 2.3.2).

2.1.2. The sidebar

The sidebar contains a general widget, namely “LOGIN”, as well as the story-specific widgets “LATEST STORIES”, “SEARCH STORIES” and “FILTERS”.

In the “LOGIN” widget the user can log in using their normal EAGLE user account. In order to save some screen real estate, this is actually a link to dedicated login page. After login, the user is automatically redirected back to the FSA access page and greeted with “Hello” and their user name. In addition, they will now see a link “Create new story”, which leads to the edit view of the story editor, and a link “Edit my stories”, which leads to the list view of the story editor where the user can see their published and

unpublished stories (see section 2.3).



Hello wolfgang!
Create new story
Edit existing stories
Logout

Figure 2: “LOGIN” widget when logged in

The “LATEST STORIES” widget displays the titles of the last five stories. The titles are links and lead to the story pages of these stories (see section 2.2). As mentioned above, this widget is part of the implementation of requirement STE03.

All widgets that are related to searching the stories and filtering results are described in the next section.

2.1.3. Searching and filtering the collection of stories

The story-specific widgets “SEARCH STORIES” and “FILTERS” help the user navigate through the collection of published stories.

The widget “SEARCH STORIES” is dedicated to searching the full text of all published stories. If the user types in one or more words in the search slit, the search result (i.e. all stories that fit this request) will be displayed in the main column. For example, a search for the German word “Grabinschrift” will find all stories that include this word. However, it will not find stories that only include the English counterpart “funerary”.

The second widget is called “FILTERS”. Regardless of the actual content of the main column (all stories or a search result), the user can restrict the displayed list to stories that satisfy some additional filter criteria. The widget is simply a wrapper for a group of filter widgets, namely “FILTER BY KEYWORDS”, “FILTER BY AUTHOR” and “FILTER BY TM ID”, and doesn’t do anything in itself. In the following we will refer to the “FILTER BY ...” widgets either as filter widgets or as sub-widgets of “FILTERS”.

With the sub-widget “FILTER BY KEYWORDS”, the user can choose to see only stories that contain one or more given keywords by typing them in the text field. For example, the author of a story may have assigned the keyword “funerary” to it. Thus, the story can be found even if the full text of the story is written in German and does not contain the word “funerary”. Keywords can be separated by commas or spaces.

Below the text field there is a word cloud of the most keywords that have been assigned most often by the story authors. If a user clicks on a keyword, the keyword replaces the list of keywords in the text field and the results in the main column are updated accordingly. The same happens if the user clicks on a keyword associated to a story.

In the sub-widget “FILTER BY AUTHOR” the user can specify an author, or more precisely the user name of the author, and then will see only stories by this author. Only one author name can be specified. If the user is logged in, they will have an additional option “show only my stories”. This will have the same effect as if the user had specified their own user name in the text field. The main difference to the “edit my stories” link in the “LOGIN” widget is that the user only sees their published stories, but not their story drafts.

In the sub-widget “FILTER BY TM ID” the user can filter the search result for stories that contain inscriptions with a given Trismegistos ID.

Together the “SEARCH STORIES” and “FILTERS” widgets implement the requirement STE01, which states that users should be able to (1) perform full-text searches of all published stories and to search for (2) keywords set by the author, (3) the user name of the author and (4) the Trismegistos IDs of the EAGLE inscriptions included in the story. In addition, the word cloud already implements the requirement STE02, which is only of medium priority and thus was only due in the second release of the FSA.

FILTERS

Filter by keywords
 

Filter by author
 

Filter by TM ID
 

Figure 3: The filter widgets

2.2. THE STORY PAGES

Each story has its own story page, just as each entry in a blog can appear in the list of all blog posts as well as on its own page. A story page displays a single story. Each story page has a different title, namely the title of the story it displays.

In the main column the story is shown in its entirety with all its text formatting and all images. Below the title there is the excerpt (if there is one) in italics but without any other formatting, as well as a list of keywords (if there are any.) There is no need for a “Read more” link. However, as on the FSA access page there will be a link “Edit this story” if the author of the story is logged in. In addition, there is a link “Back to all stories” that leads back to the FSA access page.

The sidebar is the same as on the FSA access page. “LOGIN” and “LATEST STORIES” behave exactly the same as on the FSA access page, with the obvious difference that the user is now redirected back to the story page after logging in. The “SEARCH STORIES” and “FILTERS” lead back to the FSA access page.

FUNERARY INSCRIPTIONS IN THE CURRENT EAGLE COLLECTION

Posted in

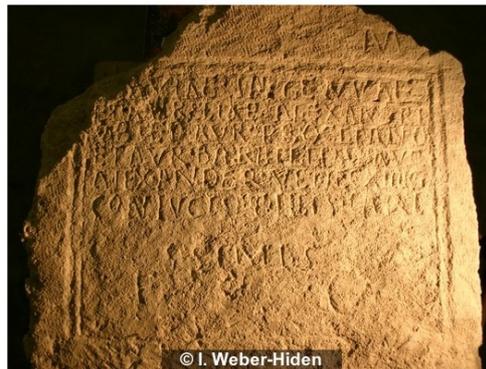
SHARE »

To get a synoptic view of all the inscriptions with funerary destination that are currently available in the EAGLE collection we will use the *Type of Inscription* field of the Advanced Search. This field actually triggers a menu where the preferred labels in the EAGLE controlled vocabulary are reported; the user has only to pick the desired value in order to perform the query.

In fact, this is not an easy task. The preferred labels are not in a single language, and the value that one looks for (e.g. the equivalent of type "funerary") can be expressed in any of the languages in use of the content providers of EAGLE. In our specific case, for example, the language is Latin: if we pick the type "Sepulcralis" from the Advanced Search options we can extract all the inscription that are described as e.g. "funerary" or "Grabinschriften".

At the moment, such a query returns 56220 results. It might be interesting to compare this datum with the numbers of all other types attested (or at least the major ones, like e.g. "votive", "honorary inscriptions" etc.). We could display a pie chart that can easily be inserted into this story!

Among the results that are returned, the inscription dedicated by the veteran Marcus Aurelius Alexander to his wife Cavia Ingenua and sons stands out immediately as a particularly striking example.



The text, as edited in the Epigraphic Database of Heidelberg, reads:

```
[D(is)] M(anibus) / [C]aviae Ingenuae / et Aureliae Alexandri/ae et Aur(elio) Procliano /  
et Aur(eliae) Barbillae M(arcus) Aur(elius) / Alexander vet(eranus) leg(ionis) XIII /  
coniugi et filis caris/simis
```

It may be noted that the text reported in the database *Ubi erat lupa* (which is itself a content provider of EAGLE, though this inscription is at the moment either not included in the searched collection or not recognized as an alternative edition of the same text) reads

```
[D(is)] M(anibus) // [F]aviae Ingenuae / et Aureliae Alexandri/ae et Aur(elio) Procliano  
/ et Aur(elio) Barbillo L(ucius?) Aur(elius) / Alexander vet(eranus) leg(ionis) XIII /  
coniugis (!) et fill(i)s cari/ssimis / f(aciendum) c(uravit)
```

A few important things stand out by comparing the two versions:

- in the Lupa's edition the word for wife, is written *coniugis*, followed by an exclamation mark that marks it as an error; the correct form of the dative singular would have been *coniugi*, which is in fact the text read by the editors of Heidelberg
- note that the editors of Lupa and Heidelberg give to the protagonists of this inscription quite different names!

Why does this happen? What are the reasons? Who is right and why? In order to solve these interesting puzzle, we must turn to our friends the epigraphists...

[Edit this page](#)

Figure 4: Display of a single story

2.3. THE STORY EDITOR

The story editor is accessible only to users that are logged in. It has two views, namely the **list view** and the **edit view**. In the first release of the FSA, both views are based on the corresponding views of the blog post editing functionality in WordPress¹. However, the views are specifically adapted for editing stories.

2.3.1. The list view

In the list view, the user sees a list of all stories they have created. This includes not only published stories, but also unpublished story drafts. The stories are listed in a more concise way than on the FSA access page. While the FSA access page is made for browsing stories and finding stories that are interesting, the list view should make it easy to locate any particular story the user wants to edit. For this reason the story titles are listed without any keywords and without any excerpt of the main text.

Each story has a link “Edit this story”, and there is one link “Create new story”. These links lead to the edit view. In the case of “Create new story”, the user starts with an empty story. For each story there is also a link “Delete this story”.

In the list view there are no widgets in the sidebar.

2.3.2. The edit view

In the edit view, the user can create a new story or edit existing ones, regardless of their publication status. The publication status can be changed in the “PUBLISH” widget in the sidebar (see section 2.3.5).

Together, the two views implement requirement STE05, which states that (1) logged in users can create new stories, and that (2) the stories will be visible to other users only after they are published.

The following sections describe aspects of the edit view in more detail: The WYSIWYG editor, the widgets in the sidebar and the modal window in which one can search different data sources.

2.3.3. The WYSIWYG editor

The centrepiece of the edit view is the WYSIWYG (“what you see is what you get”) editor. WordPress already contains a WYSIWYG editor that can be leveraged for the creation of stories.

The WYSIWYG editor provides all the standard features that one would expect. The user must specify the title of the story in a slot at the obvious position above the main text entry field. In the main text entry field the user can structure the main text of the story into paragraphs and add section headings at the appropriate places. The editor supports text formatting features such as bold, italic and underlined.

A widget “Excerpt” below the main text entry field allows the user to add a description paragraph about the story. If an excerpt is added, the abbreviated view on the FSA access page will automatically display exactly this excerpt instead of cutting off the text at an arbitrary point. Strictly speaking, the “Excerpt” widget is not part of the WYSIWYG editor. As in the title, the formatting features of the WYSIWYG editor can’t be used in the excerpt, and the description paragraph will automatically be formatted in italics.

The built-in WordPress WYSIWYG editor provides all the requirements in STE06, which states that there should be (1) a WYSIWYG editor that can create (2) text paragraphs and (3) section headers and can (4) mark text as bold, italic and underlined. It also provides the requirements in STE07, which adds that stories should have (5) a title and (6) a description paragraph.

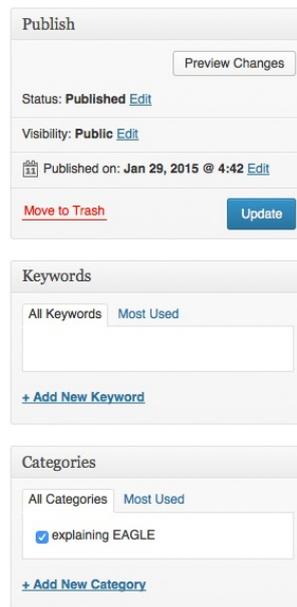
¹ <https://wordpress.org/>

Funerary Inscriptions in the current EAGLE collection

Permalink: <http://www.eagle-network.eu/stories/funerary-inscr...gle-collection/> Edit View Story

Add Media Visual Text

B *I* ABC   “      ABC                          



The screenshot shows three sections in the edit view sidebar:

- Publish:** Contains a "Preview Changes" button, "Status: **Published** [Edit](#)", "Visibility: **Public** [Edit](#)", "Published on: Jan 29, 2015 @ 4:42 [Edit](#)", a "Move to Trash" link, and an "Update" button.
- Keywords:** Has tabs for "All Keywords" and "Most Used", an empty text input field, and a "+ Add New Keyword" link.
- Categories:** Has tabs for "All Categories" and "Most Used", a list with a checked item "explaining EAGLE", and a "+ Add New Category" link.

Figure 6: The edit view sidebar

2.3.5. Publishing a story

In the sidebar users will also find the “PUBLISH” widget. Here they can save draft versions, publish the story, change the publication status back to draft, and even delete the story. Publishing a story adds it to the list of published stories in the FSA access page (see section 2.1).

When the story has been published or saved as draft, the user can always resume working on the story simply by going to the story editor page for this story again. If they work on an already published story, the “Publish” button changes to “Update”. If they do not update, the story editor will save the published version as well as the draft.

This implements the requirements STE13, STE14 and STE15. Requirement STE13 states that users should be able to (1) save drafts of their stories and to (2) resume their work and edit the drafts at any time. STE14 adds that (3) users should be able to publish their stories. STE15 further adds that users should be able to (4) edit, (5) unpublish, or (6) delete the stories they have published.

2.3.6. Searching inscriptions and other data sources

The search functionality in the story editor is missing in the first release (version 1.0) of the FSA. Although it has high priority, the actual implementation turned out to be more difficult than anticipated (see the discussion below). However, we are actively working on implementing this feature, and it will be part of an intermediate release 1.1 of the FSA. Version 1.1 will be available in time before the validation round for the first release. In the meantime, stories can be and have been created with the FSA. In version 1.0 one can use the normal EAGLE inscription search functionality as well as searches in other data sources and manually copy the results into the story text.

The inscription search will be implemented as a modal window. When the user clicks on a button “Search

inscriptions in EAGLE”, the modal window will appear. In the modal window the user will have all the functionalities the inscription search has to offer, i.e. basic search, advanced search, filtering of the results and the possibility to save searches and search results to the user’s personal space (for details see D5.2). Likewise, searches in data sources other than EAGLE will take place in a modal window. This may or may not be the same modal window as for the EAGLE inscription search.

This is a change from requirement STE11, which states that users should be able to write a story, search for objects and see the result list on a single window, as seen in services such as Storify. However, there is a good reason for this change. It doesn’t make much sense to offer a boiled-down version of the EAGLE search interface in the story editor where for example the advanced search or the possibility to filter the results is missing. If the story editor did this, the users would have to go to “real” EAGLE search page in order to fully leverage the inscription search. This is exactly what the requirement tries to prevent. On the other hand, in the standard page layout of the EAGLE portal there is simply not enough screen real estate to provide the whole inscription search functionality within the story editor page. Thus, the modal window is a compromise that implements not the letter but the spirit of the requirement. Besides, the same solution is already used for other data sources in the “WordPress Media Explorer” plugin and its “Add media” button. This plugin has chosen a modal window approach even for data sources such as YouTube, whose simpler search interfaces would have allowed it be implemented in the sidebar as well.

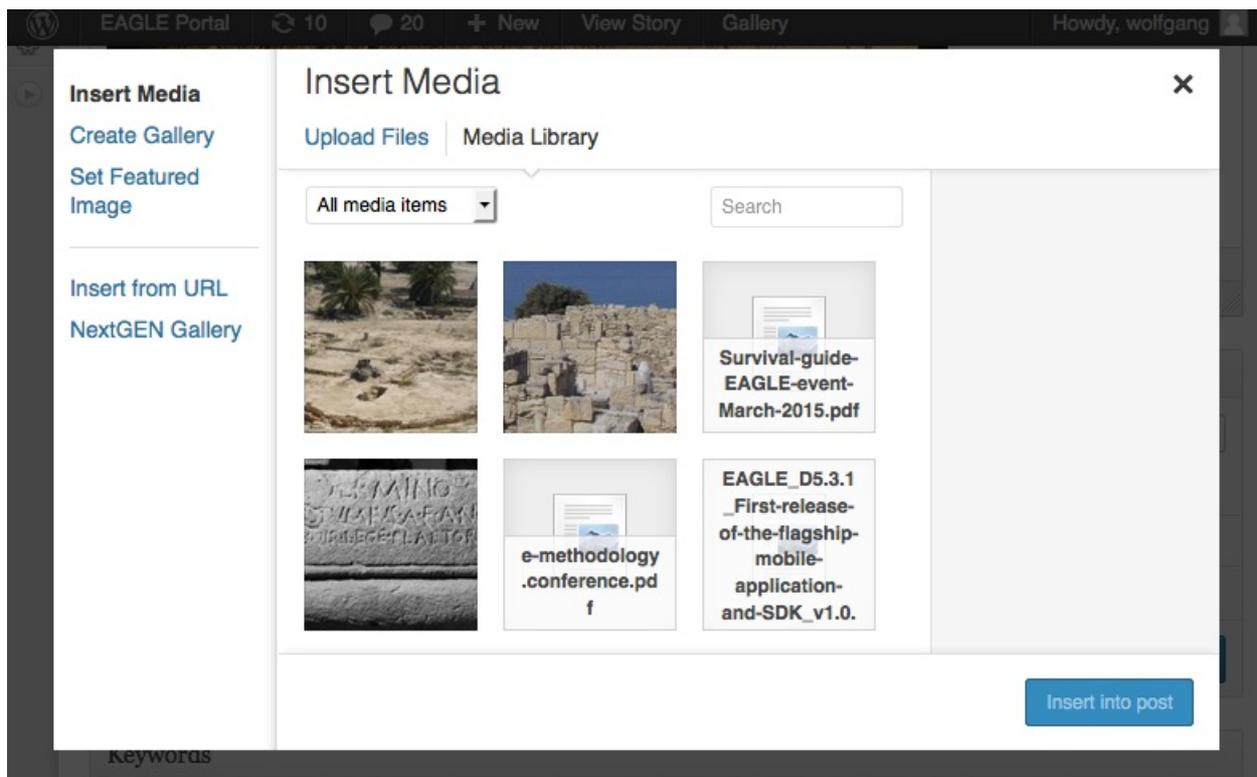


Figure 7: The intended modal window for searches

When the search functionality in the story editor is implemented, it will satisfy the following requirements:

- STE08: The editor interface should allow users to search for objects from the EAGLE collection and external repositories, in particular from Europeanana, Wikipedia, YouTube, Flickr and Pelagios.

- STE10: Search results should be formatted as: (1) a thumbnail image, (2) a title, positioned on the top-right side of the image, (3) basic information about the object below the title. Since the requirements were formulated the EAGLE search has been implemented, and after some further discussions the decision was made that the EAGLE search results should be presented in exactly same way as on the dedicated EAGLE search page, i.e. the same information about the objects should be displayed. This supersedes the original requirement at least in the case of the EAGLE inscription search.
- STE11: Users should be able to write a story, search for objects and see the result list on a single window. (However, see the discussion above.)
- STE12: The editor interface should have a drag&drop functionality for objects and text paragraphs in the search results.

When creating a new story, inserting references to already existing stories is not among the requirements for the FSA, and indeed there appears to be no pressing need for this feature. However, in section 4.3.2 we discuss the possibility to add this feature in the same way as the inscription search functionality.

3. DEVELOPER'S GUIDE

In a nutshell, the Flagship Storytelling Application (FSA) is a plugin in the WordPress installation that the EAGLE portal is based on. Stories are blog posts with a specific post type called “story”. The FSA access page is a WordPress page with its own template file. The story pages are comparable to single blog post pages and also have their own template file. The story editor with its list view and edit view is a modified clone of the WordPress blog post editor. All of this is described in detail below.

3.1. INTEGRATION WITH THE EAGLE PORTAL

The EAGLE portal is a WordPress installation. It has an EAGLE-specific theme called “eaglenetwork” based on the Flexible theme by elegantthemes². The WordPress installation includes a number of open source plugins and a few plugins that were written for EAGLE, most importantly the “Eagle Search Inscriptions” plugin. (For details about the EAGLE portal, the Eagle Search Inscriptions plugin and where to find the source code see D5.2.)

The Flagship Storytelling Application (FSA) is also implemented as a plugin in the EAGLE WordPress installation. Several standalone alternatives, including a TiddlyWiki³ and basing it on the storytelling application of the MOVIO system for virtual exhibitions⁴, were discussed and rejected. The decision to implement the FSA as a WordPress plugin reflects the goal of sharing the EAGLE corporate design as well as fully integrating the FSA with the other functionalities of the EAGLE portal. For example, the plugin can directly access the user accounts that are used throughout the EAGLE portal. The integration will be even tighter in version 2.0 where the FSA will be connected with the personal spaces of the users (see section 4.2).

3.1.1. The “Eagle Storytelling Application” plugin

The name of the plugin is “Eagle Storytelling Application”. It is located in a subfolder “eagle-storytelling” of the standard “wp-content/plugins” folder. In fact, all the files of the FSA plugin are in this folder. Even the template files for the FSA access page and the story pages are in the FSA plugin folder instead of the theme folder.

While the abbreviation FSA for the Flagship Storytelling Application was chosen in analogy to the acronym FMA of the Flagship Mobile Application, the namespace of the WordPress plugin is “esa”, in analogy to the namespaces “et” of the EAGLE theme and “esi” of the EAGLE plugin for searching inscriptions. It is unlikely that this will lead to any confusion.

The FSA plugin inherits the following things from the EAGLE portal and its underlying WordPress installation:

- The EAGLE corporate design (via some modified template files)
- Access to the EAGLE user accounts
- The WordPress editor

² <http://www.elegantthemes.com/gallery/flexible/>

³ <http://tiddlywiki.com/>

⁴ <http://www.movio.beniculturali.it/>

3.1.2. The “story” post type

WordPress distinguishes between blog posts and so-called pages. Displaying a page is similar to displaying a single blog post. However, pages are normally limited in number and tend to include fixed information such as an “about” page. In an unmodified WordPress installation the start page includes links to all pages and shows a paginated list of all blog posts.

The backend of a WordPress installation is a MySQL database. Both blog posts and pages are stored in a table called “wp_posts”. In the data field “post_type” they have the values “post” and “page”, respectively. Blog posts and pages are created on different but very similar internal “edit pages”. The main difference is that the edit pages can include different sets of widgets for adding tags etc. The relevant URL parts are “/wp-admin/edit.php” for blog posts (the default post_type “post” is not spelled out) and “/wp-admin/edit.php?post_type=page” for pages.

The EAGLE portal does not display blog posts on the start page (“Home”), but as items on the News page. Links to the News page as well as other pages are in the header area of the normal page template. The EAGLE portal also introduces a new page type called “project” with a new edit page (/wp-admin/edit.php?post_type=project) based on the page edit page. Project pages are also stored in the “wp_posts” table. The comment functionality is generally turned off in the EAGLE portal.

The FSA introduces another blog post type called “story”. Stories are created in yet another edit page /wp-admin/edit.php?post_type=story (for more details see section 3.4), and are also stored in “wp_posts”. The story edit page is modified to accommodate the needs of the FSA with a different set of widgets. In accordance with the policy of the EAGLE portal, it is not possible to comment on stories.

3.2. THE FSA ACCESS PAGE

The FSA access page is implemented as a page in the WordPress sense. It is comparable to the start page displaying a paginated list of blog posts, or in the case of the EAGLE portal the News page.

The URL path of the FSA access page is /stories/. It has its own associated template page “page-stories.php”.

3.2.1. Design rationale

The FAS access page was designed to visually resemble the News page. The FSA access page shares the idea of searching and filtering the results with the Inscription Search page (with the obvious difference that the FSA access page searches stories and the Inscription Search page searches inscriptions), but since stories resemble news items more closely than they resemble inscriptions, the design of the News page is more appropriate for the FSA access page than the design of the Inscription Search page.

There are, however, some differences in the set of widgets in the sidebar between the News page and the FSA access page that arise from the specific needs of the FSA. For example, while the News page has a News archive sorted by months, this does not make much sense for user-generated stories. On the other hand, the FSA access page has an arsenal of filters for restricting search results that wouldn’t make sense for the small number of news items.

3.2.2. The sidebar

The sidebar starts with the “LOGIN” widget. The login itself is not specific to the FSA, and technical details about the login process are described in deliverable D5.2 about the EAGLE portal. In addition, users that are logged in will also see a link “create new story” that leads to the edit view of the story editor. Another link “edit my stories” leads to the list view of the story where the user can see their

published and unpublished stories. Thus, the FSA plugin adds these links to the normal view when the user is logged in.

The next widget is “LATEST STORIES”. This is comparable to “LATEST NEWS” on the News page. It is based on the same search as for the list of all published stories, but only the titles of the five most recent stories are displayed.

The last two widgets, “SEARCH STORIES” and “FILTERS”, are described in the next section.

3.2.3. Searching and filtering stories

The “SEARCH STORIES” widget searches the full text of all published EAGLE stories. This functionality is provided by the WordPress engine. Normally it searches all blog posts, i.e. all entries in the table “wp_posts” with the normal post_type “post” and the post_status “publish”. Here it searches the entries with post_type “story” instead. If a search was performed, the list of all published stories in the main column is replaced by the search result.

The second widget is called “FILTERS”, which is actually a group of filter widgets called “FILTER BY KEYWORDS”, “FILTER BY AUTHOR” and “FILTER BY TM ID”. All of these filters take the search result (or all published stories if the user did not perform a search) and filter this list for additional criteria. That is to say, only the stories that also fulfil the additional criteria will be displayed.

In the first of these filters, “FILTER BY KEYWORDS”, the user can enter one or more keywords. If they enter more than one keyword, the keywords are connected by AND, i.e. all of the keywords must be present in the list of keywords of a story. This filter is comparable to the “TOPICS” filter on the News page and it is implemented in the same way.

The second filter is “FILTER BY AUTHOR”, where author means user name. It adds the criteria that the data field “post_author” in the table “wp_posts” should match the name that has been entered by the user.

The third filter “FILTER BY TM ID”, where the user can filter the search result for stories that contain inscriptions with a given Trismegistos ID, is implemented differently as there is no corresponding data field that lists the Trismegistos IDs of these inscriptions. At the moment this is implemented as a simple full-text search for the ID. In other words, this filter has the same effect as adding the ID as a search term in the “SEARCH STORIES” field. However, in section 4.3.2 we discuss a more elegant way of searching for Trismegistos IDs.

3.3. THE STORY PAGES

The story pages are comparable to pages displaying a single blog post. When the user publishes a story, it gets a unique URL (a “pretty URL”, see below) and the corresponding story page is created automatically by the WordPress engine, or more precisely, the page will be created on-the-fly when the URL is requested. The URL path of a single story page is /stories/ plus a sanitized version of title of the story that is stored in the data field “post_name” in “wp_posts”. For example, the story with the title “Funerary Inscriptions in the current EAGLE collection” has the URL path

`/stories/funerary-inscriptions-in-the-current-eagle-collection/`

Since stories are blog posts with a different post_type, there is no explicit page counterpart of the FSA access page. However, the FSA plugin makes sure that the story pages use the dedicated template file “page-stories-single.php” by changing the default template file for the post_type “story”. The FSA access page is not affected by this.

The design of the main column is similar to the template for news items, again with the difference that only posts with the `post_type` “story” are displayed. Another difference is that the excerpt of a story, if there is one, will be displayed in italics below the title and before the main text of the story.

The sidebar comprises the “LOGIN” widget and the “LATEST STORIES” widget, both of which are implemented in the same way as on the FSA access page.

3.4. THE STORY EDITOR

The story editor with its list view and edit view is a clone of the normal WordPress edit page, but dedicated to stories. The list view has the URL path

```
/wp-admin/edit.php?post_type=story
```

The edit view has the URL path

```
/wp-admin/post-new.php?post_type=story (new story)
```

```
/wp-admin/post.php?post=...&action=edit (existing story or draft)
```

where “...” denotes the story ID. Thus, stories are referred to by their IDs rather than their “pretty URL” as on the story pages. This makes sense especially for new stories that do not yet have a title to generate a pretty URL from. On the other hand, the URL path “/wp-admin/post.php?post=...” without the “&action=edit” will be automatically replaced by the pretty URL of the corresponding story page.

As with a normal blog post, the user can enter keywords and an excerpt for the story, save drafts and publish the story. A story draft normally has the `post_status` “inherit”, a published story has the `post_status` “publish”. All of this is provided by the WordPress engine “for free” as soon as the clone of the normal edit page has been created. Thus, at the moment, the requirements STE05, STE06 and STE07 that deal with the behaviour of the story editor are automatically provided by WordPress (see section 2.3.2 and 2.3.3 for a discussion of these requirements). However, in a later version of the FSA the design of the story editor and especially of the edit view will change in order to unify the appearance of all interfaces of the FSA (see section 4.3.1 for details).

The sidebar of the edit view contains a “KEYWORDS” widget, which is the counterpart of the “FILTER BY KEYWORDS” widget on the FSA access page. The implementation is again provided by the WordPress engine. In addition, the FSA plugin separates the story keywords from the keywords for news items.

There is no counterpart for the “FILTER BY AUTHOR” widget as the author cannot be changed by the user. The user who creates the story is automatically the author. The “FILTER BY TM ID” widget has no counterpart in the edit view either since the filter is at the moment implemented as a full-text search for the Trismegistos ID. And even when the implementation of the filter changes as discussed in section 4.3.2, there will be no need for the user to enter the Trismegistos IDs explicitly.

3.5. SEARCHING INSCRIPTIONS AND OTHER DATA SOURCES

The search functionality in the story editor for EAGLE inscriptions and other data sources has not been implemented yet. See the discussion in section 2.3.6 for the reasons. It will be added in an intermediate version 1.1 of the FSA.

4. CONCLUSIONS AND FUTURE PLANS

With version 1.0, the Flagship Storytelling Application (FSA) is ready to use. The stories that have already been created with it can be seen on the FSA access page. The decision to implement the FSA as a plugin of the EAGLE portal was an important step towards the goal of a tight integration with the EAGLE infrastructure. We are excited to see what stories will be created by members of the EAGLE community and the interested public.

The missing search functionalities within the story editor will be implemented in version 1.1. The remaining medium priority requirements are slated for version 2.0. For this version we also intend to give the story editor a clearer design. These improvements are described in more detail in the following sections.

4.1. THE INTERMEDIATE VERSION 1.1

The next step after version 1.0 is the intermediate version 1.1, which will be available in time before the first evaluation round for the FSA. This version will focus on adding the search functionalities in the story editor. The plans for implementing the search functionalities are discussed in section 2.3.6. Although these functionalities are not a prerequisite for creating stories with the FSA, they will nonetheless make creating stories much more convenient.

With version 1.1, all requirements with high priority will be implemented.

4.2. REQUIREMENTS FOR VERSION 2.0

After the release of version 1.1 the outstanding requirements of medium priority will be STE09, STE16 and STE17.

STE09 states that users should be able to search and include passages of Ancient Greek and Roman texts from the Perseus Digital Library. We assume that implementing this requirement will take a lot of effort, but since Perseus is undoubtedly a major provider of Ancient Greek and Roman texts, we anticipate that it will make the FSA more attractive.

STE16 states that users should be able to access items stored in their personal space and include them into their stories. In their private space, a user can save search results and individual items from the EAGLE collection (for more details see D5.2). A possible implementation could be another modal window or as part of the modal window for the inscription search functionality (see section 2.3.6). The corresponding requirement for the EAGLE portal is PSE28, which states that users should be able to view the list of the stories created by them in their personal space. In other words, the user will be able to access the list view of the story editor in their personal space. As soon as this requirement is implemented, the list view of the story editor might disappear completely if it is no longer needed. STE16 and its counterpart PSE28 are an example of the envisaged tight integration of the FSA with the EAGLE portal.

STE17 states that the story editor should provide a generic EpiDoc XML reader that allows users to include any EpiDoc-compliant inscription as object into their stories. In other words, not only EAGLE inscriptions, but any inscription that is given in a standard format can be included in the story.

In addition, the requirement STE04 states that users should be able to share their stories on social networks. Since this requirement has a low priority, it may or may not be implemented in version 2.0. However, this requirement is rather straightforward to implement, so we expect it to find its way into

version 2.0.

Version 2.0 is to be completed before month 30 of the project, i.e. by September 2015.

4.3. ADDITIONAL PLANS

In this section we discuss some plans for improving the FSA that are not part of the requirements.

4.3.1. Redesigning the edit page

The most important improvement of the story creation process will be a clearer design for the story editor and especially its edit view. The new design of the edit view will make it look like other pages in the EAGLE portal and especially the FSA access page. We expect that this unified design will make the FSA more easily accessible especially for the interested public. In the redesigned version the “Description” field will be located where it appears in the published text, i.e. after the title and before the main text. The “KEYWORDS” widget may also be moved to a position below the “Description” field.

An alternative would be to introduce a new user role that allows the user to edit stories on the dedicated story editor page, but not news items or pages.

As discussed in section 4.2, the list view may become part of the normal view of the user’s personal space, although an alternative could be to merge the list view into the FSA access page. This could be achieved by adding button “also show my drafts” and a possibility to show just the titles without excerpts on the FSA access page.

4.3.2. Shortcodes for inscriptions

WordPress has the concept of so-called shortcodes such as “[inscription]12345[/inscription]”. We will probably implement such a shortcode that will display a box with the inscription dataset. If the user transfers the inscription from a search result into the story via drag&drop, it may internally be represented as such a shortcode instead of the full text and image of the inscription. The details need to be clarified. For example, there may be a group of similar shortcodes for (1) images, (2) text and (3) image plus text, all of which would be displayed within a box, or (4) just the link to a dataset without a box. As a consequence, EAGLE inscriptions would be quoted in a more standardized way in the stories. Stories could thus contribute to the EAGLE “corporate identity”.

Another consequence would be that the “FILTER BY TM ID” widget could be implemented in a more clever way. All stories would be found that feature an [inscription] that is associated with the desired Trismegistos ID. The Trismegistos ID would no longer have to appear in the full text of the story.

Another application would be “stories for tourists”, i.e. a list of all stories that include inscriptions from their current location. This could be implemented similarly to “FILTER BY TM ID” by searching for [inscription], only that the filter would not look for the Trismegistos ID of the inscription but the place information associated with the inscription.

Stories themselves could also have shortcodes. This would make it easier in a story to refer to another story without resorting to the URL of the other story. There is no way yet to search for other stories within the story editor, however, and again the problems with screen real estate would have to be solved.

4.3.3. Where the stories are stored

Stories are at the moment stored in the table “wp_posts”, together with all news items, pages and projects. At the moment, entries in “wp_posts” are created by a small number of EAGLE affiliates. As soon as the FSA is accessible to the interested public, it may make sense to move stories to a new

database table.

This would also make it easier to introduce a moderation system for stories. (The discussion whether such a system is needed at all is ongoing.)

As a continuation of the discussion in section 4.2 about requirement PSE28 and the integration of the FSA with the users' personal spaces, it may also make sense to move stories to the table where the users' personal spaces are kept. However, the table for the personal spaces might turn out to be ill-equipped for accommodating stories.

4.3.4. Miscellanea

Multiple authors: It remains to be seen whether stories need to be able to have more than one author. This would have consequences for the visibility of stories via the "FILTER BY AUTHOR" widget, but more importantly it would enable author teams to work on the same story, albeit not at the same time.

Dropdown list of authors: In the "FILTER BY AUTHOR" widget it may be helpful to have a dropdown list of the most active authors. Alternatively, the same functionality could be implemented with a word cloud for authors, similarly to the word cloud for keywords.

Timelines of events: The story editor may support creating timelines of events. The details would need to be discussed.

REST interface: e.g. `/stories/?storiesByUser=wolfgang`. Ideally this would be coordinated with a similar REST interface for the Inscription Search.

New URL scheme: A new scheme that is more independent of the Wordpress URL scheme and would replace especially the URLs of the type `/wp-admin/post.php?post=...&action=edit` when editing stories.

Different languages: Connect keyword search with EAGLE's controlled multilingual vocabularies. The user interface may be also translated into other languages.

5. REFERENCES

[D5.1]

F. Mambrini. EAGLE portal and services requirements and specifications. EAGLE Deliverable 5.1. Available online at:

http://www.eagle-network.eu/wp-content/uploads/2013/06/EAGLE_D5.1_-_Portal-and-services-requirements-and-specifications.pdf

[D5.2]

C. Prandoni, N. Alfarano, V. Casarosa. EAGLE Portal. EAGLE Deliverable 5.2. Available online at:

http://www.eagle-network.eu/wp-content/uploads/2013/06/EAGLE_D5.2_EAGLE-Portal_v1.0.pdf

[D5.4]

X. Martin. First Release of the Flagship Mobile Application and SDK. EAGLE Deliverable 5.3.1. Available online at:

http://www.eagle-network.eu/wp-content/uploads/2013/06/EAGLE_D5.3.1_First-release-of-the-flagship-mobile-application-and-SDK_v1.0.pdf

[D5.5]

F. Mambrini. Validation Plan. EAGLE Deliverable 5.5. Available online at:

http://www.eagle-network.eu/wp-content/uploads/2013/06/EAGLE_D5.5_Validation-Plan_v1.0.pdf

[D6.1]

C. Prandoni, R. Santucci, N. Clonini. EAGLE Project Website. EAGLE Deliverable 6.1. Available online at:

http://www.eagle-network.eu/wp-content/uploads/2013/06/EAGLE_D6.1_EAGLE-Project-Website_v1.0.pdf