# ANNEX X1 – EUROPASS SERVICES

## Technical description of the Europass web-based resources

## 1. GENERAL

The Europass multilingual website (http://europass.cedefop.europa.eu) was launched in February 2005; it is now available in 27 languages. It is the main resource for the implementation of the Europass initiative and gives citizens direct access to Europass instruments, the Europass CV, the Europass Language Passport, the Europass Cover Letter and the Europass Skills Passport, which can be filled in by citizens, either offline or online using a user-friendly interface; completed CVs or ELP can be saved in various formats (Microsoft Word/OpenOffice/PDF, HTML and XML) and later on uploaded (in XML format) for update.

A custom Content Management System (CMS) has also been developed by Cedefop, through which all content in the Europass platform and the Europass Mobility Management Tool is managed (accessed, edited, added, manipulated). The CMS also serves the purpose of the batch creation of a set of Excel files with all content, in a special format facilitating an easy translation, and also the extraction of any part of the content, in any language, in XML format. These extracts will be used for import to Drupal. See Annex X3 for an example.

The Europass platform is currently developed, maintained and hosted by Cedefop in Thessaloniki.

## 2. TECHNICAL ARCHITECTURE

From the user perspective, the Europass editor is presented as a Thin Client, Single Page Application (SPA), i.e. HTML & JavaScript, and it can be accessed by a common Internet browser. Most operations are done locally in the browser. Few operations, such as document generation, are done by calling a RESTful API on the server side. This server is currently implemented with a thread-based model using Apache Tomcat application server.
The language of presentation of the content has to be any of the official languages of EU, and the site gives the ability to a user (who browses through its contents) to change the presentation language at any point of the site.

Thus, Europass is multilingual (UTF-8) in all interfaces presented to the user or administrator. Also its internal data transfer and storage mechanisms employ UTF-8.
From the functional perspective, Europass can be divided in the following three subsystems:

1. **The Europass Site**, which contains all the static and informative content of the site. This content is managed by the Content Management System (CMS).
2. **Europass Web Apps (EWA) on-line editor**. This is an integrated application with main purpose to help a user fill them in, and generate a document in the appropriate format and language to be downloaded by the user. CMS is used on this subsystem to manage the static textual content (help, tooltips, labels & instructions) fully multilingual as it does for the rest of Europass site. In order to allow the later editing of a document without database storage, each document can be serialized and unserialized in an XML file. This XML file holds all the data of the instrument, along with user preferences specifying part of the appearance.
   The Europass documents are generated by the direct, programmatic synthesis of the data supplied in the on-line HTML forms, to the XML-based OpenOffice (odt) format. After the ODT file is created, if requested, it is transformed to the appropriate format (PDF, Microsoft Word), through the LibreOffice software running in server mode.
   See section 3 below for a more technical description.
3. **REST Web Services**: This subsystem is a standard REST Access point to EWA. It relies on the EWA APIs to provide online generation of documents to external systems.

## 2.1. Technologies

The development is platform neutral. It does not require the existence of a particular Database server or Web server. For the implementation of this project, the following are used:

- Web Server: Apache
- Servlet Container: Apache Tomcat
- Database Server: MS SQL Server 2012

The architecture is database independent and the possibility to use other available databases (such as Oracle) is given.

### 2.1.1. LibreOffice

The LibreOffice platform was chosen for the development of those subsystems of EWA requiring the generation of files in formats that are able to be processed by Office suites (MS DOC & RTF), or printed in a local to the user printer (PDF). The LibreOffice server is a part of the LibreOffice suite and can be deployed simply by installing the LibreOffice suite. The recommended type of installation is Network Installation, comprising the following parts:

- A centralized installation, keeping the necessary program files.
- Multiple workstation installations on users that exist on the machine. This installation installs only setting and configuration files.

This installation schema favours the usage of the Open Office server in a multiple listener mode, one listener for each user, enabling load-balancing features.

#### 2.1.2. Java 3rd Party libraries

All the libraries in this project are Open Source libraries, licensed under the Apache or Apache like License. The following are of major importance for the project:

- Backbone, fancySelect, jcrop, jsonpath, scrollbar, underscore, backbone-nested, fileupload, jquery, select2, xdate, jquery-ui, redactor (licensed version), touche, handlebars, jsonjs, require, typeahead.

## 3. EWA TECHNICAL DESCRIPTION

### 3.1. Modules

EWA (Europass Web Apps) is a distributed Java-based web system which consists of the following modules:

1. **Editors**: This is the application which is called first when a user requests https://europass.cedefop.europa.eu/editors. It is responsible for assembling the front-end part of EWA (a JavaScript-based Single Page Application) by detecting the User Agent, Locale, etc. and populating the initial HTML page that the user sees with suitable variables.
2. **API**: This is the heart of EWA where most of the processing and business logic takes place: data model, marshalling/unmarshalling of XML/JSON to Java objects and vice versa, file type detection, image processing, error handling and messages, session management, population of ODT template with user-entered data, downloads, emails, export to cloud services, database interactions, etc.
3. **Office**: The module responsible for converting ODT templates to PDF and DOC as well as for managing the lifecycle of the underlying LibreOffice binaries.

In addition, there's an independent fourth module for the Europass **REST API**. This is also a Java application which runs in isolation from the other modules, but nevertheless shares some common libraries (JARs) with the API module.

### 3.2. Server Topology

For load-balancing and high-availability purposes, multiple redundant instances of each one of the aforementioned EWA modules are deployed in the production environment.

The distribution of requests is handled via Round-Robin DNS and four Apache HTTP Server instances in the front-end. Each EWA Java app runs on its own Tomcat instance (except from the two REST APIs which run on the same Tomcat with the API module). See table below for details.

| Host | OS | VM Specs |
|---|---|---|
| **Apache HTTP Server – 4 instances** | | |
| 194.26.23.51 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 8Gb RAM, 240Gb HDR |

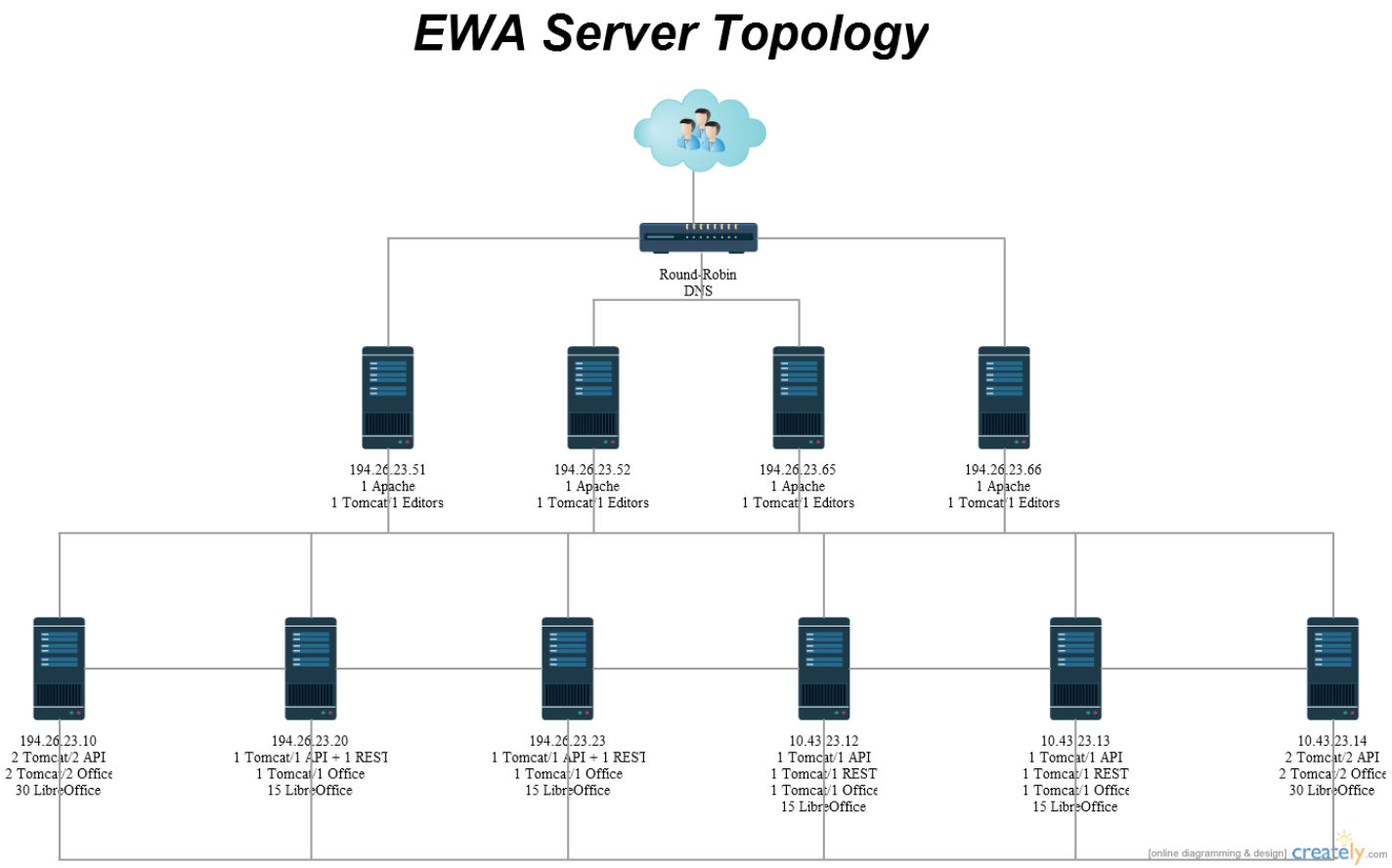
| | | |
|---|---|---|
| 194.26.23.52 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 8Gb RAM, 240Gb HDR |
| 194.26.23.65 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 50Gb HDR |
| 194.26.23.66 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 50Gb HDR |
| **EWA Editors – 4 instances** | | |
| 194.26.23.51 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 8Gb RAM, 240Gb HDR |
| 194.26.23.52 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 8Gb RAM, 240Gb HDR |
| 194.26.23.65 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 50Gb HDR |
| 194.26.23.66 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 50Gb HDR |
| **EWA API – 8 instances** | | |
| 194.26.23.10 (2 inst.) | Ubuntu 12.04.5 LTS | Intel Xeon@2GHz, 8Gb RAM, 170Gb HDR |
| 194.26.23.20 | Ubuntu 12.04.5 LTS | Intel Xeon@2GHz, 6Gb RAM, 140Gb HDR |
| 194.26.23.23 | Ubuntu 12.04.5 LTS | Intel Xeon@2GHz, 4Gb RAM, 140Gb HDR |
| 10.43.23.12 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 100Gb HDR |
| 10.43.23.13 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 100Gb HDR |
| 10.43.23.14 (2 inst.) | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 150Gb HDR |
| **EWA Office – 8 instances** | | |
| 194.26.23.10 (2 inst.) | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 8Gb RAM, 170Gb HDR |
| 194.26.23.20 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 6Gb RAM, 140Gb HDR |
| 194.26.23.23 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 4Gb RAM, 140Gb HDR |
| 10.43.23.12 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 100Gb HDR |
| 10.43.23.13 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 100Gb HDR |
| 10.43.23.14 (2 inst.) | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 150Gb HDR |
| **EWA REST API – 4 instances** | | |
| 194.26.23.20 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 6Gb RAM, 140Gb HDR |
| 194.26.23.23 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 4Gb RAM, 140Gb HDR |
| 10.43.23.12 | Ubuntu 12.04.5 LTS | Intel Xeon@2GHz, 5Gb RAM, |

| | | |
|---|---|---|
| | 64bit | 100Gb HDR |
| 10.43.23.13 | Ubuntu 12.04.5 LTS 64bit | Intel Xeon@2GHz, 5Gb RAM, 100Gb HDR |

Some other points of interest:

- All static resources (CSS, JavaScript files, etc.) are served directly by Apache, not Tomcat.
- All Editors instances are connected with all API instances via Apache (i.e. requests to APIs are sent using https://europass.cedefop.europa.eu/api).
- All API instances are connected with all Office instances directly, not via Apache (i.e. load-balancing is handled by the API app).
- Each Office instance is connected with 15 LibreOffice binaries running on localhost. In total, 120 LibreOffice binaries are used (8 Office instances * 15 binaries).

Consult the schematic below for graphical overview of the topology.




### 3.3. Error management and logserver

A log aggregator is in place, gathering all error logs files from all servers and inserting them into an indexed and searchable system (Logstash and ElasticSearch). This makes the pinpointing of the problem reported by users and its troubleshooting easier.

## 4. PROBLEMS WITH CURRENT ARCHITECTURE

The following need to be improved in the current architecture and are identified as next steps:

4.1. The load balancing of incoming requests to the 4 Apache web servers is done using Round-Robin DNS, which is somewhat cumbersome (if a server is down, it needs to be removed from the DNS and the zones be updated throughout the Internet), creates delays to the experience of the user (until the browser detects the unavailable instance) and does not guarantee preservation of sessions. A more robust solution should be applied, e.g. by introducing a layer of a hardware or specialised software load balancing unit on the front end.

4.2. The primary resource-intensive activity of the backend EWA service is the conversion of the ODT documents to other format, i.e. PDF and Microsoft© Word. This is currently done with the use of LibreOffice in server mode. This has advantages such as e.g. that there is a single template for each document in ODT format (e.g. for the Europass CV), and the file is just converted to other two formats via the LibreOffice server. Some problems currently existing are:

4.2.1. The distribution of requests to the LibreOffice binaries is threads-based and is missing a proper load balancing layer. This results in it becoming a bottleneck and starting getting clogged with the current pattern of traffic. A proper load balancing layer (e.g. even using Apache web server) and an asynchronous model should be introduced (e.g. node.js or a message queuing solution like Apache Camel or RabbitMQ).

4.2.2. The LibreOffice binaries themselves are not stable enough. The LibreOffice processes just die occasionally. An infinite loop has been introduced to restart them as soon as they die.

4.3. There are severe problems with the caching mechanism, which is currently based on OSCache. De-caching is cumbersome and it is difficult to get to see soon the updated content online, to production. A propose caching layer should be introduced, e.g. using Varnish Cache.

## 5. EUROPASS STATISTICS TECHNICAL ARCHITECTURE

### 5.1. General information

The following three types of Statistical Data are stored in Europass:

1. **Visits** of the Europass Portal (http://europass.cedefop.europa.eu/en/home) and of the Europass editor (https://europass.cedefop.europa.eu/editors/en/cv/compose). For each visit, an entry is recorded in Apache Web Server logs.

2. **Documents downloaded offline** from the Europass Portal. Each time a document (ECV|ELP|EDS|ECS|EM Templates|Instructions|Examples) is downloaded from the Europass Portal, an entry is recorded in Apache Web Server logs.

3. **Documents completed online** using the Europass online editor. Each time a document (ECV/ELP/ESP/ECL) is generated online, the web application records an entry to the Europass database.

### 5.2. AWStats

In the beginning of each month, AWStats Tool runs on the Europass Statistics server in order to analyse the Apache web server logs for Visits and Documents downloaded offline. Then the result of the analysis is copied using a custom Java application to the database "europass_statistics" of the Europass Statistics Database.

### 5.3. Google analytics

There is a plan to replace AWStats with Google Analytics as regards the analysis of visits and documents downloaded offline. In particular, visits and documents downloaded offline will get recorded in Google analytics and then transferred to a local database using a dedicated Java Tool.

### 5.4. Oracle BI Tool

In order to *visualise* Europass Statistics, Oracle Business Intelligence has been installed. Hardware and Storage characteristics of this VM can be found below.

Statistical data are retrieved from the Europass Statistics Database and are visualised as Dashboards in the Oracle Business Intelligence Enterprise Edition. They are also automatically pushed as PDF Reports to Apache Web servers using Scheduled Tasks of the Oracle BI Tool in order to be available online in a dedicated section of the Europass Portal. Currently, Oracle BI pushes PDFs to the four production Apache Web Servers, i.e. 194.26.23.24, 194.26.23.25, 194.26.23.65 and 194.26.23.66.

### 5.5. Statistics API AND query interface

There is a dedicated Statistics API deployed as a Java Web application under a Tomcat 8 servlet container which fetches statistical data filtered/combined by multiple dimensions. A prototype of a custom query UI written in JavaScript calls the aforementioned API so as to enable users make to custom queries in the statistics database.

The Statistics API query will be performed following the syntax below:

GET /stats/to/<format>/<query>

where <format> is the response's type and can be either json or csv, if no format is added the response type is JSON by default.

| ENDPOINT | RESPONSE TYPE |
|---|---|
| /stats/to/json | JSON |
| /stats/to/csv | CSV |
| /stats | JSON (default) |

<query> follows the syntax:

<query> : <prefix>;<arg1>=<assign1>;<arg2>=<assign2>; …
;<argN>=<assignN>

- <prefix> : “param”, “visits” or “downloads”
- <argX> : parameter, special parameter or query shortcut
- <assignX> : a value, a condition, a range or a group.

Below there are some example queries, the request endpoints and the responses in JSON format.

**Example 1:**
Get all CV, ESP, ELP documents that were generated by residents of Greece for February, May and September of 2013

**Request**:
GET /stats/generated;document-type=ECV|ESP|ELP;country=el;date=201302|201305|201311;orderby=date,DESC

**Response in JSON format:**

```
{
	{ doctype : ecv,
		{ country: el,
			{ year:  2013,
				{ month: 2, value: <number> }
				{ month: 5, value: <number> }
				{ month: 9, value: <number> }
			}
		}
	},
	{ doctype : esp,
		{ country: el,
			{ year:  2013,
				{ month: 2, value: <number> }
				{ month: 5, value: <number> }
				{ month: 9, value: <number> }
			}
		}
	},
```

```
        { doctype : elp,
                { country: el,
                        { year:  2013,
                                { month: 2, value: <number> }
                                { month: 5, value: <number> }
                                { month: 9, value: <number> }
                        }
                }
        }
}
```

**Example 2:**
Get all CV/ESP documents that were generated by residents of Greece aged between 22 and 34 in groups of age with difference 2 years

**Request:**
GET /stats/generated;document-type=ECV&ESP;country=el;age=22-34;groupby=age;step=2

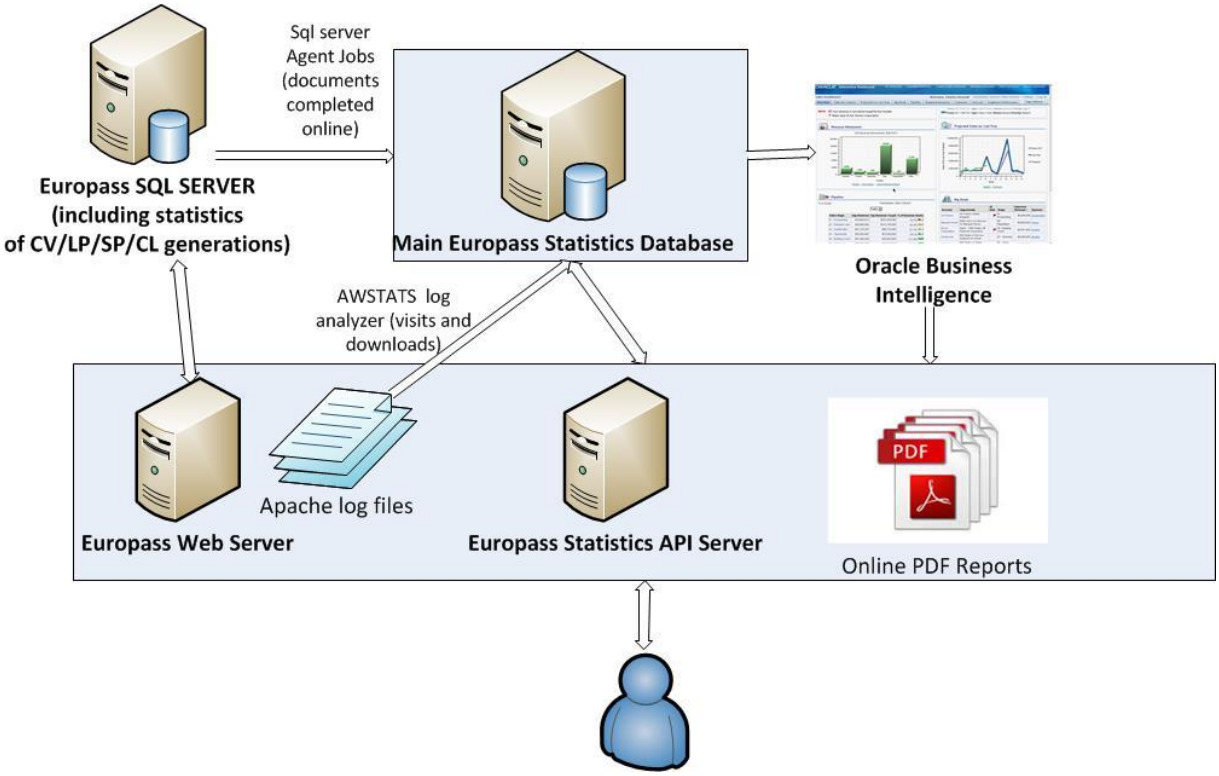
**Response in JSON format:**

```
{
        { doctype: ecv-esp,
                { country: el,
                        { from : 22, to : 24 , value : <number> }
                        { from : 24, to : 26 , value: <number> }
                        …
                        { from : 32, to : 34 , value: <number> }
                }
        }
}
```

### 5.6. Appendix – Technical characteristics of Server and VMs

| Host name / IP | OS | | | Specs |
|---|---|---|---|---|
| Europass Statistics Server | Ubuntu 64bit | 12.04.5 | LTS | Intel(R) Xeon(R) CPU E7 - 2820 @2.00GHz / 6.00 GB RAM |

## 5.7. Statistics architecture diagram




## 5.8. Abbreviations used

| | |
|---|---|
| **ECV** | Europass Curriculum Vitae |
| **ELP** | Europass Language Passport |
| **ESP** | European Skills Passport |
| **ECL** | Europass Cover Letter |
| **EDS** | Europass Diploma Supplement |
| **ECS** | Europass Certificate Supplement |
| **EM** | Europass Mobility |
| **BI** | Business Intelligence |
| **VM** | Virtual Machine |
| **UI** | User Interface |

# 6. CONNECTION WITH CLOUD SERVICES

Specific modules provide for connection with Authentication providers, to allow users save their documents to Cloud storage. Currently, the connection is established for Google drive, Onedrive and Dropbox. Additionally, an import from LinkedIn allows for the population of the EWA editor with the CV data existing there.