Erasmus Without Paper — from the technical perspective

Janina Mincer-Daszkiewicz (jmd@mimuw.edu.pl), Wojciech Rygielski (rygielski@mimuw.edu.pl), Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw, Banacha 2, 02-097 Warszawa, Poland

Keywords
Erasmus Without Paper, student mobility, Erasmus+, Student Information System, EWP Network, EWP Registry, connectors, GitHub, semantic versioning

1. INTRODUCTION

The **Erasmus Without Paper** (EWP) project [2, 3] aims to create a network supporting the electronic exchange of student data by interlinking the existing Erasmus student databases of Higher Education Institutions (HEIs) with the goal to permanently migrate from the paper world to the electronic world of student data. It is a response to the current needs of the information society, more specifically a large number of potential end-beneficiaries ranging from students, institutional Erasmus coordinators, IRO staff, HEIs at large, national agencies, or even the European Commission. The project addresses the Erasmus+ goal of increasing student mobility and recognition of student achievements (measured in recognized ECTS points). The EWP Network is the first attempt to standardize student data transfer on a European-wide scale. It is important to note that the transfer does not involve documents themselves (e.g. scanned copies) but the data that is contained in these documents, so that they can be used for the generation of various documents, processed automatically and stored in institutional databases. In particular, no data is held centrally.

There are 11 partners composed of public institutions, higher education organizations, and companies from 8 European countries with a dissemination potential to over 400 HEIs from 36 European countries. They are supported by 11 associate partners actively involved in the project.

The project started on November 1st, 2015 and will last for two years (grant number 562264-EPP-1-2015-1-BE-EPPKA3-PI-FORWARD).

In this paper we present the results of the first half year of the project, mostly from the technical perspective. We want to share some design and implementation decisions concerning the architecture, security, scalability of the EWP Network, data format, API, protocols supporting data exchange, and other technically oriented issues.

2. PROJECT GOALS

The project aims to design and work out a pilot for an integrated communication network supporting the exchange of student data in an electronic form, and to build connecting software modules (connectors) that will allow Student Information Systems (SISs) with built-in mobility modules (part of SIS that takes care of Bilateral Agreements, student applications, Learning Agreements, Transcript of Records and other documents) and/or stand-alone Mobility systems (like MoveON or Mobility-OnLine) to exchange data over the EWP Network. In more detail, the main project tasks are the following:

- Describe all possible mobility scenarios.
- Create common data models for the exchanged information and design appropriate document formats.
- Define the necessary transport protocols and standards.
- Take care of identity management (authentication and authorization methods).
- Solve the security and privacy issues.

- Build connector modules (also generic) that allow data handling software to send and receive data over the network.
- Include extra tools to increase performance and usability (e.g. grade conversion).

3. GENERAL PROJECT DEVELOPMENT DECISIONS

The final success of the project depends on the number of institutions joining the network. It is crucial that from the very beginning the software will be designed and developed using an open source approach with the active involvement of the academic community. We have created the official EWP organization on **GitHub** (https://github.com/erasmus-without-paper/). All technically biased project work will be reported on GitHub. We will post there specifications, source code, documentation etc. All developers interested in the work progress or wanted to get involved should subscribe to the chosen projects on GitHub. We have also created a web page for developers involved in the design and implementation of the EWP platform and connectors (http://developers. erasmuswithoutpaper.eu/) with the general overview of the documents and specifications, information on their state, links, tools, etc.

There is a set of repositories for various sections of documentation. Documents posted on GitHub are available for review, change notifications help to keep all informed, built-in issue trackers can be used for asking questions, reporting errors or posting suggestions. GitHub supports formal versioning of documents and source code (we will follow rules of **semantic versioning**). Official versions of data format and **Application Programming Interface** (API) will be approved by the project partners, and − once the APIs are released onto production servers − backward compatibility will be guaranteed. The data format will be defined officially by XSD and a validation tool will be provided to help with formal verification of the produced data files.

We plan to take into account data formats developed by other mobility-oriented European projects, like EMREX-ELMO worked out by the EMREX group [1, 4].

4. ARCHITECTURE AND SECURITY

EWP Network is composed of the **EWP Hosts** and the **Registry** (see Fig. 1). EWP Hosts are groups of HEIs. The EWP project aims on allowing communication between them. In order to join the Network, the Host should publish a valid **Discovery Manifest file** somewhere on its servers and send the URL of this file to the EWP Registry administrator. It will be stored in the Registry allowing partners to identify and verify future requests.
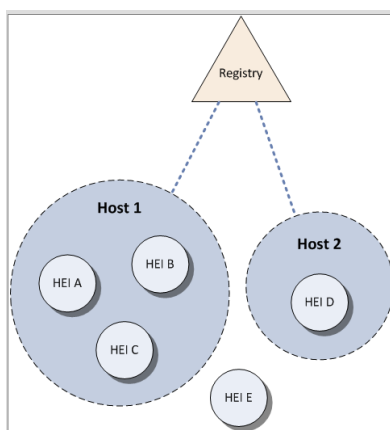

Fig. 1 Main components of the EWP Network

Registry

The Registry is the only centralized part of the EWP Network. It allows all EWP Hosts to access the list of other EWP Hosts. It may also be used for projects unrelated to EWP, as long as these projects have similar architecture. The Registry is being updated **automatically**. It periodically reads all the information which all EWP Hosts provide within their Discovery Manifest files, and these changes are reflected in the Registry responses.

The major advantage of such automatic updating is that the partners *do not need* to contact the Registry maintainer when they want to change some of their Registry entries. Most changes in the Registry can be performed simply by updating the manifest on the partner's server (and the Registry will fetch these changes automatically). This approach supports the scalability of the solution.

EWP Hosts are not required to implement *all* features of the EWP Network. In order to avoid unnecessary requests, the Registry also keeps track of all the implemented APIs.

Some of the APIs are **event listener APIs**. If Host 2 wants to receive change notifications from other hosts, it indicates in its Manifest file that it has implemented a specific event listener API. Now, if Host 1 is able to broadcast such notifications, then it asks the Registry which hosts have implemented the proper listeners, and posts the proper notifications at the listener URLs.

Security

There are two types of certificates involved in the communication:

- **Server certificates** are used by the Host when it **responds** to API requests.
- **Client certificates** are used to **issue requests** within the EWP Network.

Implementers may use the same certificate for both purposes.

API

In order to join the EWP Network, a new partner needs to implement a subset of its APIs. There are many APIs, and each partner will choose which of them he wants to implement. Each partner has to implement Discovery Manifest API:

- It serves to identify the partner, announce the list of HEIs covered by the partner's system, and used certificates while fetching the data from the EWP Network. It's like a simple "business card" of the EWP Host. Its role may be expanded along the project.

- It serves to inform all **other** members of the network about which features (APIs) **the EWP Host** has implemented. This list may include external APIs (unrelated to the EWP project).

The partner needs to host the Discovery Manifest file somewhere on its servers. The Discovery Manifest file is a simple XML file which must conform to the provided XML Schema.

The **Echo API** allows beginner EWP developers to test the security of their EWP Network connections. It doesn't "do" anything, but it requires the developer to implement the core security framework (which will be needed by all the other APIs later on).

The EWP Network specification posted on GitHub describes EWP Network components, establishes common security measures, features, data types and vocabulary, and explains how partners communicate between one another.

5. SUMMARY

The Erasmus Without Paper project has entered the design and implementation phase. The **pilot** we are working on aims to test out the feasibility of the proposed solutions for all HEIs in Europe. In the first step the general architecture of the EWP Network has been designed along with some general discovery services and security mechanisms. The API design is in progress. The development of connectors for local systems will follow. All the specifications, documents, and source code will be hosted on GitHub. We want to share them with the academic community and prospective developers of local connectors. This paper is a call for cooperation, especially directed toward other European projects/groups with the focus on student mobility and student data sharing across borders.

6. REFERENCES

[1] EMREX project. Retrieved in January 2016 from: http://www.emrex.eu.
[2] Erasmus Without Paper (EWP), an Update. Presentation by Valère Meus at 2015 Gröningen Declaration Network Annual Meeting, Málaga, 4-6 May, 2015. Retrieved in January 2016 from: http://www.groningendeclaration.org/erasmus-without-paper-ewp-update
[3] Erasmus Without Paper project. Retrieved in January 2016 from: http://www.erasmuswithoutpaper.eu.

[4] Lindstedt M., Hansen A.B., Ravaioli S., Vangen G., Sidselrud A., Mincer-Daszkiewicz J., Henriksson P. (2015). *Field trial on the impact of enabling easy mobility on recognition of external studies (EMREX)*. EUNIS 2015, Dundee. http://www.eunis.org/eunis2015/wp-content/uploads/sites/3/2015/04/EUNIS2015_submission_13_P1_TRACK_3_slot_3.pdf

7. AUTHORS' BIOGRAPHIES

Janina Mincer-Daszkiewicz graduated in computer science in the University of Warsaw, Poland, and obtained a Ph.D. degree in math from the same university. She is an associate professor in Computer Science at the Faculty of Mathematics, Informatics and Mechanics at the University of Warsaw specializing in operating systems, distributed systems, performance evaluation and software engineering. Since 1999, she leads a project for the development of a student management information system USOS, which is used in 47 Polish Higher Education Institutions, gathered in the MUCI consortium. Janina takes active part in many nation-wide projects in Poland. She has been involved in Egracons, EMREX and Erasmus Without Paper European projects.

Wojciech Rygielski graduated in computer science in the University of Warsaw, Poland. He is a senior programmer working on USOS.

He is one of the main developers from Poland in EMREX and Erasmus Without Paper European projects. He is responsible for keeping the EWP specifications clean, up-to-date and backward-compatible.