Erasmus Without Paper from the technical perspective

Janina Mincer-Daszkiewicz
Wojciech Rygielski
University of Warsaw

Agenda

- Project goals
- Development decisions
- Architecture
- Security
- Use cases
- API + flowcharts
- Summary

Erasmus Without Paper (EWP)

- EU project, 2015-2017.
- 11 partners from public institutions, higher education organizatons, and companies from 8 European countries.
- 11 associate partners.
- Dissemination potential to over 400 HEIs from 36 European countries.

Project goals

- Design and work out a pilot for an **integrated communication network** supporting the **exchange of student data[1] in an electronic form**.
- Build **connecting software modules** that will allow Student Information Systems (SISs) with built-in mobility modules[2] and/or stand-alone Mobility systems to exchange data over the EWP Network.

[1]data, not documents (eg. scanned copies), which can be processed automatically, stored in databases, used to create documents

[2]part of SIS that takes care of ***Bilateral Agreements***, ***student applications***, ***Learning Agreements***, ***Transcript of Records*** and other documents

Project goals – details

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

Project development decisions

- Open source approach.
- General overiew of documents and specifications on developers.erasmuswithoutpaper.eu.
- Design and implementation reported on GitHub.
- Specifications available publicly, everyone can contribute (not only EWP partners).
- Changes easy to follow (version numbers, release notes).
- Set of repositories for various sections of documentation and code.
- Backward compatibility.
- API and data formats defined formally by XSD.
- Tool for formal verification of the produced data files.

http://developers.erasmuswithoutpaper.eu/

Expand/Collapse All

- Chapter 1. Working with EWP Technical Documentation v1.1.0
- Chapter 2. Architecture and Security v1.1.0
- Chapter 3. Primary Network APIs
  - Chapter 3.1. Discovery Manifest API v3.0.0
  - Chapter 3.2. Echo API DRAFT
  - Chapter 3.3. Registry API DRAFT
- Chapter 4. EWP Mobility Process Explained

Versions of this document:

Version 0.2.0 - REVIEW RECOMMENDED	read it	download as ZIP	see a diff
Version 0.1.0	read it	download as ZIP	

Summary:

With help of some flowcharts, this document briefly describes **how the Student Mobility Business Process is modelled** within the EWP Network. It should be useful to get a quick grasp on which APIs are used by whom and when.

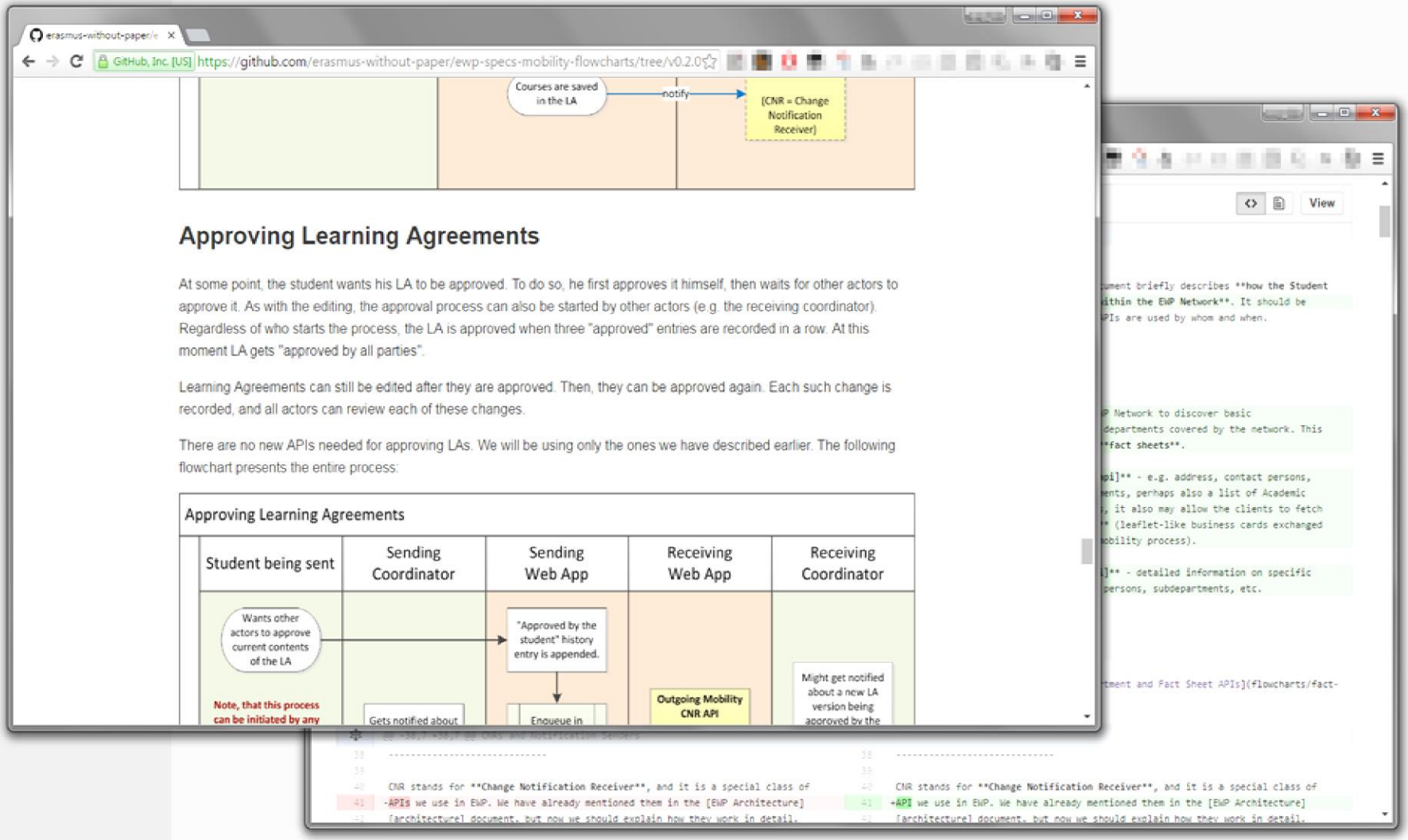
- Chapter 5. General Purpose APIs
  - Chapter 5.1. Institutions and Departments
    - Chapter 5.1.1. Institutions API DRAFT
    - Chapter 5.1.2. Departments API DRAFT
- Chapter 6. Erasmus Mobility APIs
  - Chapter 6.1. Interinstitutional Agreements (IIAs)
    - Chapter 6.1.1. Interinstitutional Agreements API DRAFT
    - Chapter 6.1.2. Interinstitutional Agreement CNR API DRAFT
    - Chapter 6.1.3. Interinstitutional Agreement Search API DRAFT
  - Chapter 6.2. Outgoing Mobilities
    - Chapter 6.2.1. Outgoing Mobilities API DRAFT
    - Chapter 6.2.2. Outgoing Mobility CNR API DRAFT
    - Chapter 6.2.3. Outgoing Mobility Search API DRAFT
    - Chapter 6.2.4. Outgoing Mobility Remote Update API DRAFT
  - Chapter 6.3. Transcripts of Records
    - Chapter 6.3.1. Transcripts of Records API DRAFT
- Chapter 7. Appendices
  - Chapter 7.1. `.ewpmobility` Exchange File

Versions of this document:

Latest draft version - DRAFT	read it

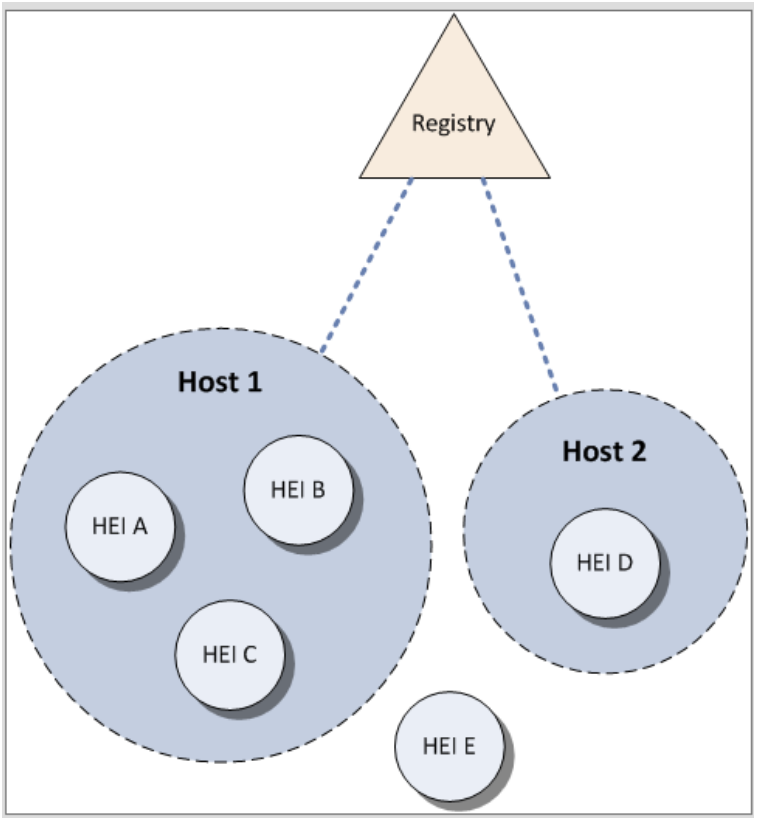
Summary:

http://developers.erasmuswithoutpaper.eu/



EWP Network

- EWP Network is composed of EWP Hosts and Registry.
- Each EWP Host may represent more than one HEI.
- EWP Host publishes **Discovery Manifest File**, somewhere on its servers. The manifest is fetched by the registry, information is extracted and propagated.

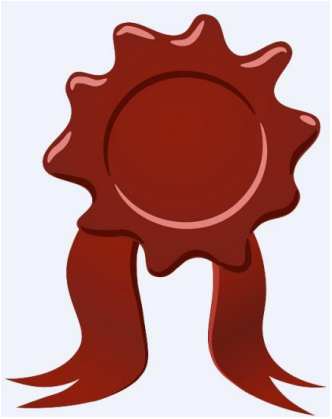


Registry

- The only central part of the EWP Network.
- Keeps track of the EWP Hosts and APIs they implement (possibly only subset).
- Updated automatically – periodically reads Discovery Manifest files (which are updated locally → **scalability**).
- API – Discovery (obligatory), Echo (for security testing), Event listener, other.

```
<?xml version="1.0"?>
- <manifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:ewp="https://github.com/erasmus-without-paper/ewp-specs-architecture/blob/stable-v1/common-types.xsd"
  xmlns="https://github.com/erasmus-without-paper/ewp-specs-api-discovery/tree/stable-v3">
      <ewp:dev-email>admin-or-developer@example.com</ewp:dev-email>
      <ewp:dev-notes>It's not a real manifest. Just an example.</ewp:dev-notes>
    - <apis-implemented>
        - <discovery version="3.0.0">
              <url>https://example.com/manifest.xml</url>
          </discovery>
        - <echo xmlns="https://github.com/erasmus-without-paper/ewp-specs-api-echo/blob/master/manifest-entry.xsd"
          version="0.0.0">
              <ewp:dev-email>usually-a-developer@example.com</ewp:dev-email>
              <ewp:dev-email>some-other-developer@example.com</ewp:dev-email>
              <ewp:dev-notes>Some notes which might useful for client developers.</ewp:dev-notes>
              <url>https://example.com/ewp/echo</url>
          </echo>
      </apis-implemented>
    - <institutions-covered>
        - <hei id="uw.edu.pl">
              <other-id type="pic">999572294</other-id>
              <other-id type="erasmus">PL WARSZAW01</other-id>
              <name xml:lang="en">University of Warsaw</name>
          </hei>
      </institutions-covered>
    - <client-credentials-in-use>
          <certificate> MIIB9TCCAV4CCQDzcL/uks2ttjANBgkqhkiG9w0BAQUFADA/MQswCQYDVQQGEwJOTzETMBEGA1UE
              CAwKU29tZS1TdGF0ZTENMAsGA1UEBwwET3NsbzEMMAoGA1UECgwDVWlPMB4XDTE1MDIwNTEzNDE1
              MloXDTE4MDUyMDEzNDE1MlowPzELMAkGA1UEBhMCTk8xEzARBgNVBAgMClNvbWUtU3RhdGUxDTAL
              BgNVBAcMBE9zbG8xDDAKBgNVBAoMA1VpTzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAsnI3
              V1JKOl6Ghgzu5AOSsUFcJfefKP3y4lkVzW18T8AXkLKjNWQxJpW7h7kqeEYpcijAIV/xRMu8vEeb
              roadh9joOOD+ePL95VnAxcMxy3SdrGxRJ5RZwNf1l5UBxm416acIR78frlFX2kKz8kz0gmOmqFFe
              y3zZm7ZdiKw9fTsCAwEAATANBgkqhkiG9w0BAQUFAAOBgQBn6MlnleclAG/BVF4irljHag7OOopx
              TlQzb1V/mR0kYtsr6PZzCTJdwwU5YYWPG2hdqYRzyydlEGid5OPeCVtNQNQG8yKDCYeCS571jSbQ
              7Q0n0IR4mv7SeLq537mVBaaftV3HdnyoSLcXQpDep/akUOa8TpPwVg/+2U7TWNGJeQ== </certificate>
      </client-credentials-in-use>
</manifest>
```

Security



Two types of certificates involved in the communication:

- **Server** certificates
  - used by the Host when it **responds** to API requests,
  - "regular" SSL certificates, bound to host's domain, signed by a trusted CA,
  - neither the clients nor the registry will be storing server certificates.
- **Client** certificates
  - used to **issue requests** within the EWP Network,
  - each Host (via its Manifest file) declares a list of certificates it will use for making requests to other hosts, list is fetched by registry, **fingerprints** of these certificates are served to the EWP Network,
  - **Extended Validation** (**EV**) certificates are recommended (but not required) for serving manifest files, because they allow the Registry Service administrators to vet new EWP partners more easily. They are DNS-spoofing-proof and Terena provides them for all HEIs **for free**.

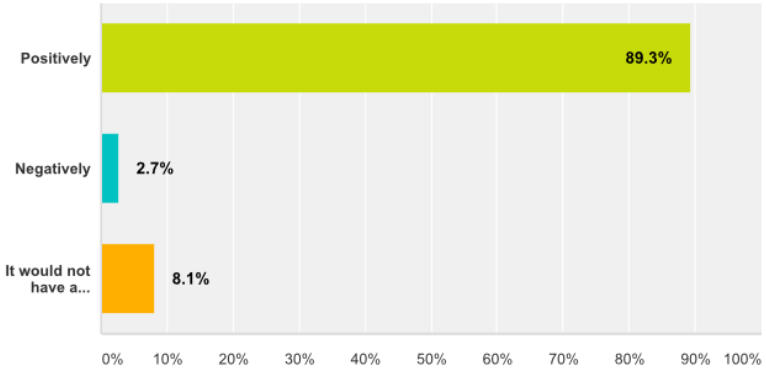
Security concerns – follow discusssion on GitHub: https://github.com/erasmus-without-paper/ewp-specs-architecture/issues/9

Use cases

- Based on a **survey – 1049 filled questionaries from 31 countries**.
- Use cases identified
  - **Interinstitutional Agreement**
  - **Nominations**
  - **Learning Agreement**
  - **Arrival & Departure**
  - **Transcript of Records**
  - **Grade Conversion**
- Summary
  - Very high interest in EWP
  - All steps of mobility are strong candidates for EWP integration
  - IT platforms less used for data exchange than ... snail mail
  - Local IT systems only modestly integrated
- Detailed analysis of use cases led to design of API

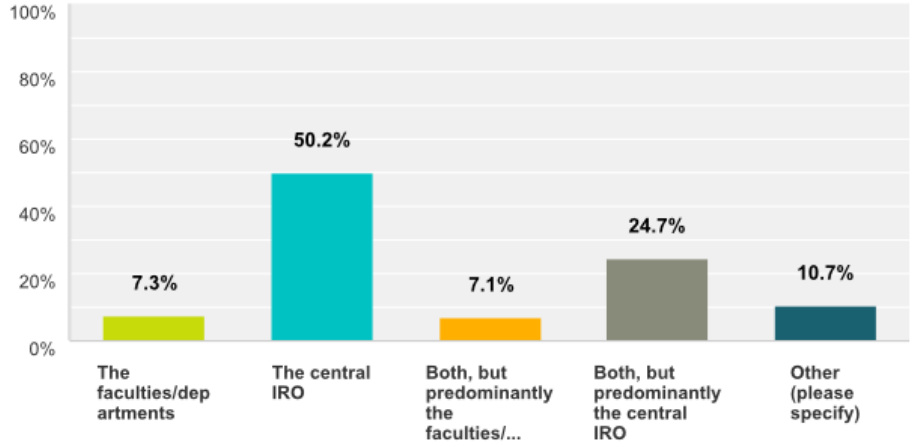
The goal of Erasmus Without Paper is to enable different IT systems to seamlessly exchange information and documents among Erasmus partner institutions. How do you think achieving this goal could impact your work?

Answered: 903 Skipped: 0



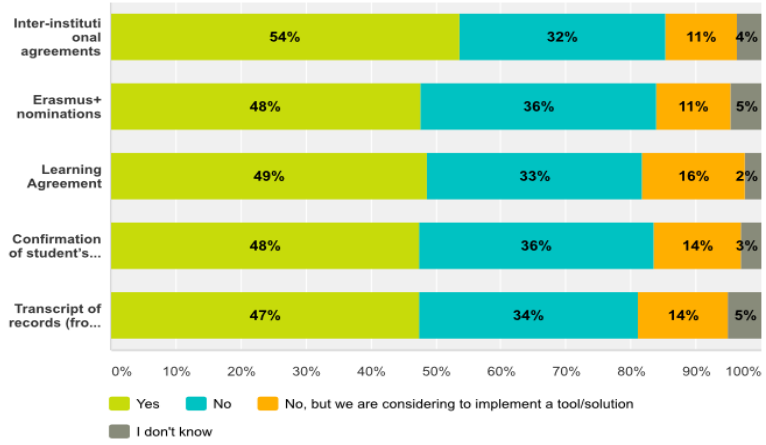
Is the management of Erasmus+ mobility the main responsibility of

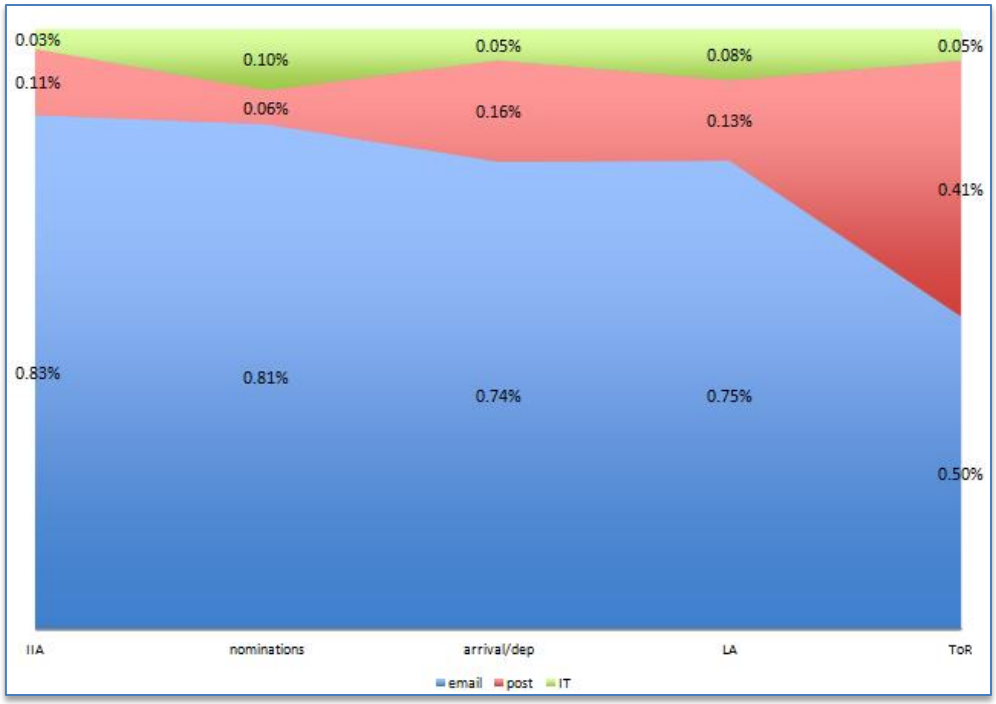
Answered: 903 Skipped: 0



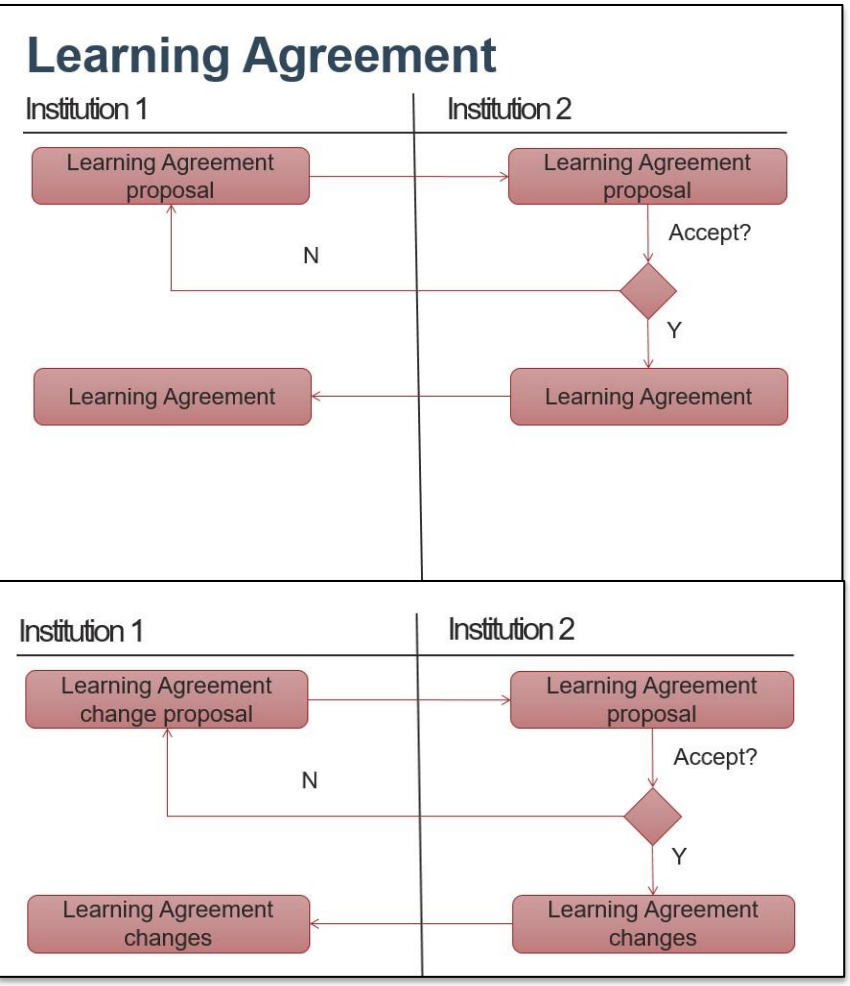
Is the data of outgoing Erasmus+ students for the following documents/steps stored in a local system in such a way that it can be reused for other purposes (e.g. reporting to the Mobility Tool+ or used for local documents like the diploma supplement)?

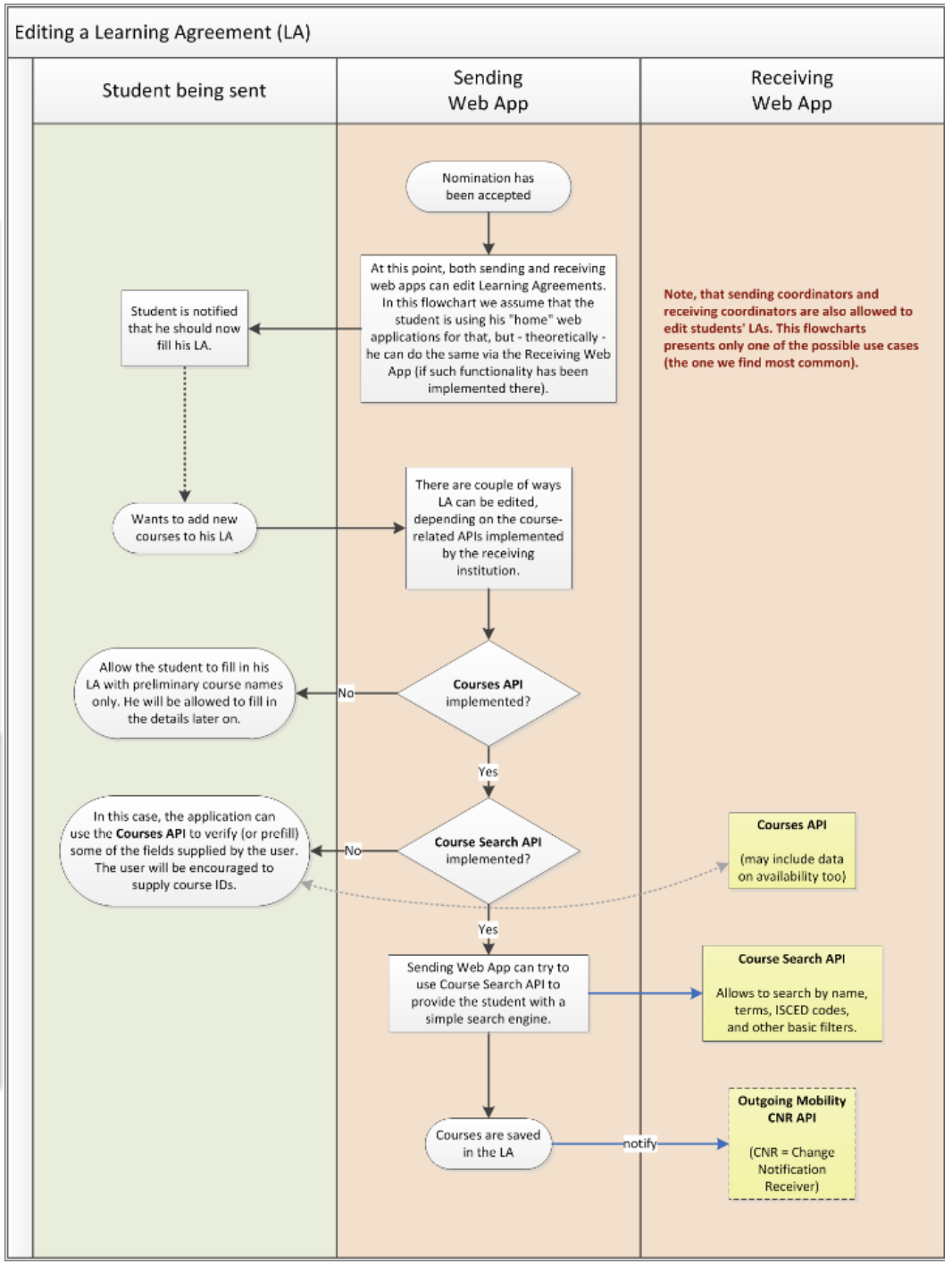
Answered: 903 Skipped: 0





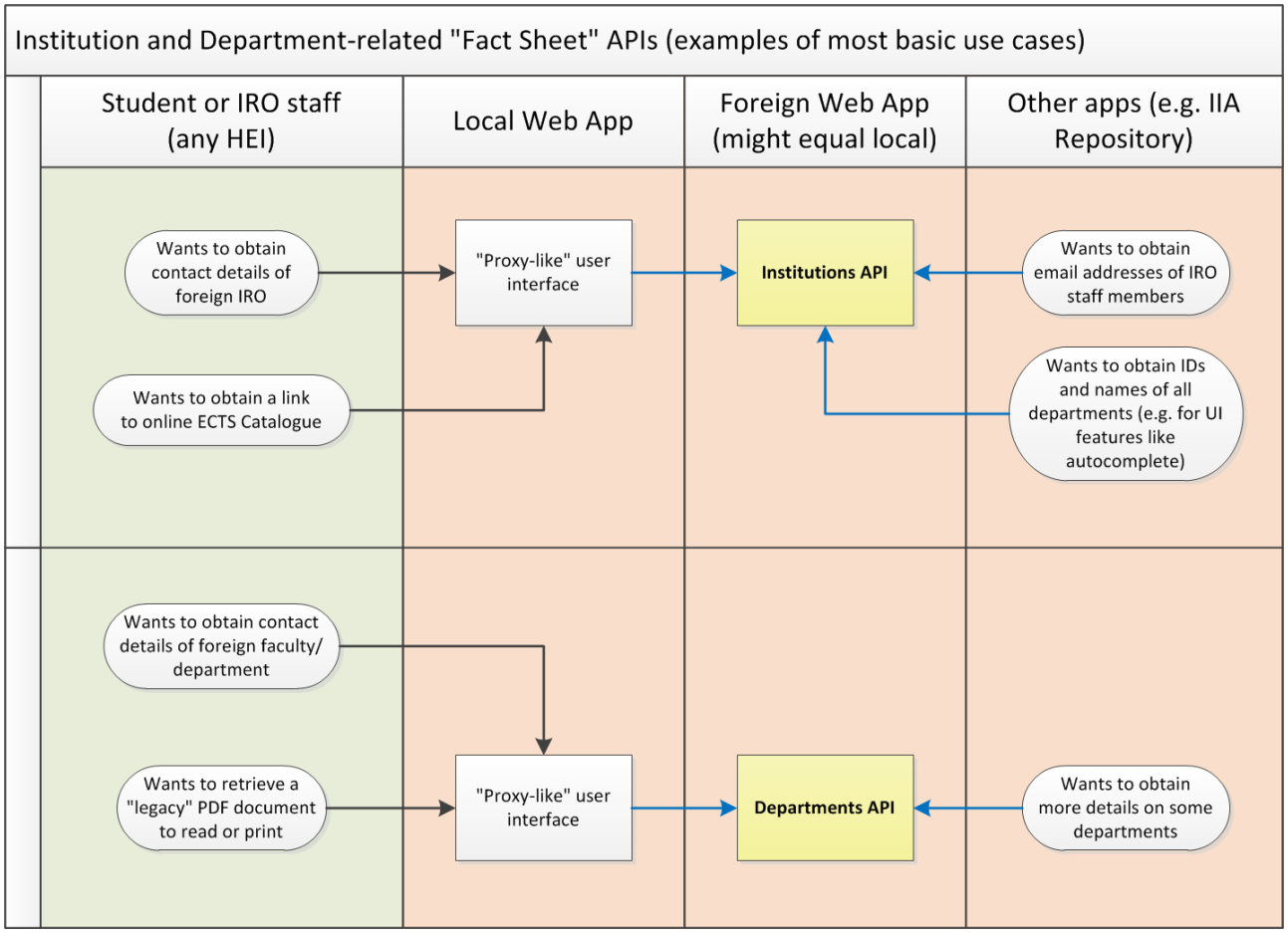
Use case leading to API





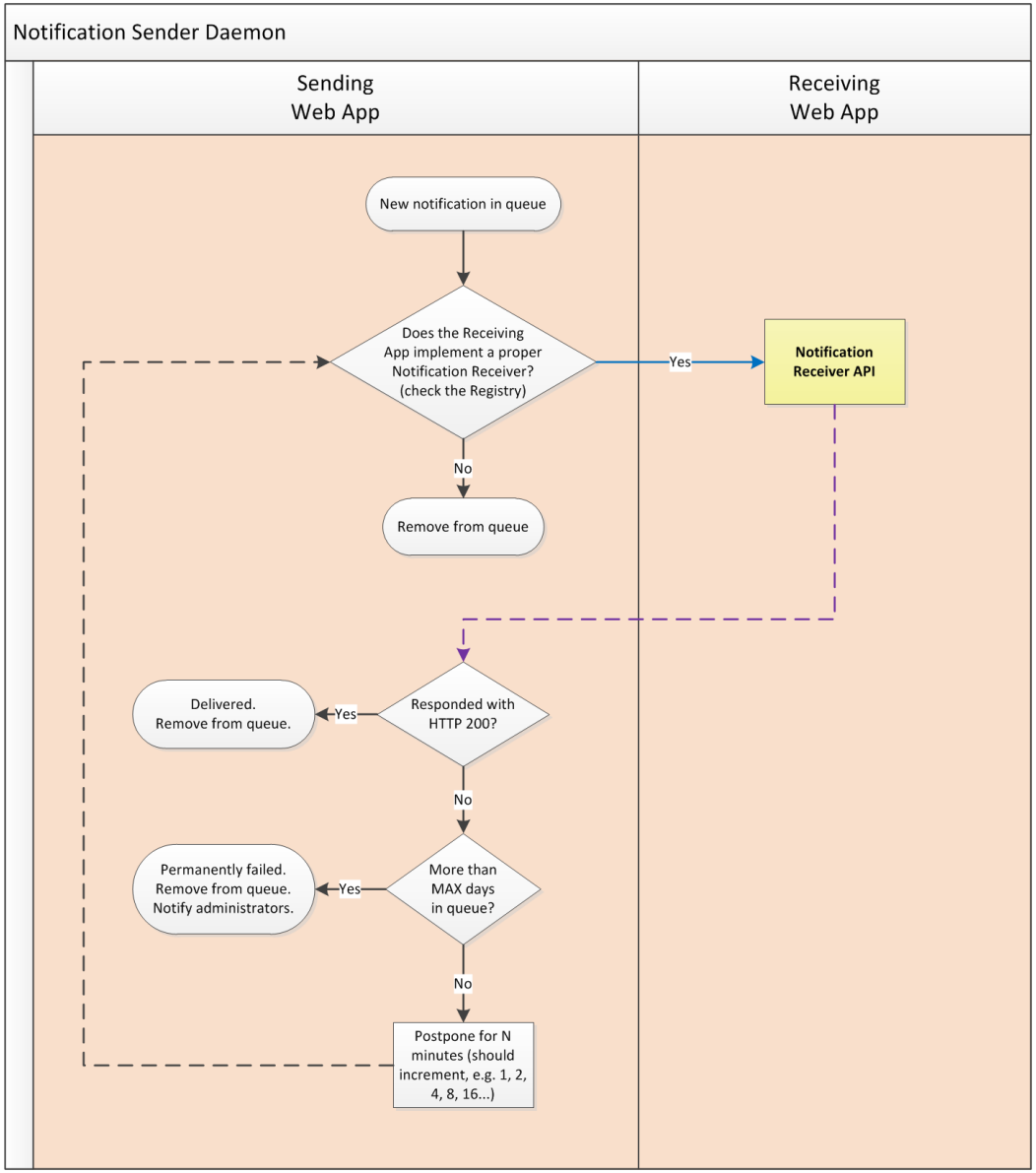
Accessing information on Institutions APIs

- Allow members of EWP Network to discover basic information on other institutions and departments covered by the network (**fact sheets**).
- **Institutions API** – e.g. address, contact persons, logo image, list of departments, list of academic terms used. May allow clients to fetch **PDF Fact Sheets** (nice, printable format, exchanged by IROs).
- **Departments API** – detailed information on specific departments, e.g. address, contact persons, institutes or other kinds of subunits.



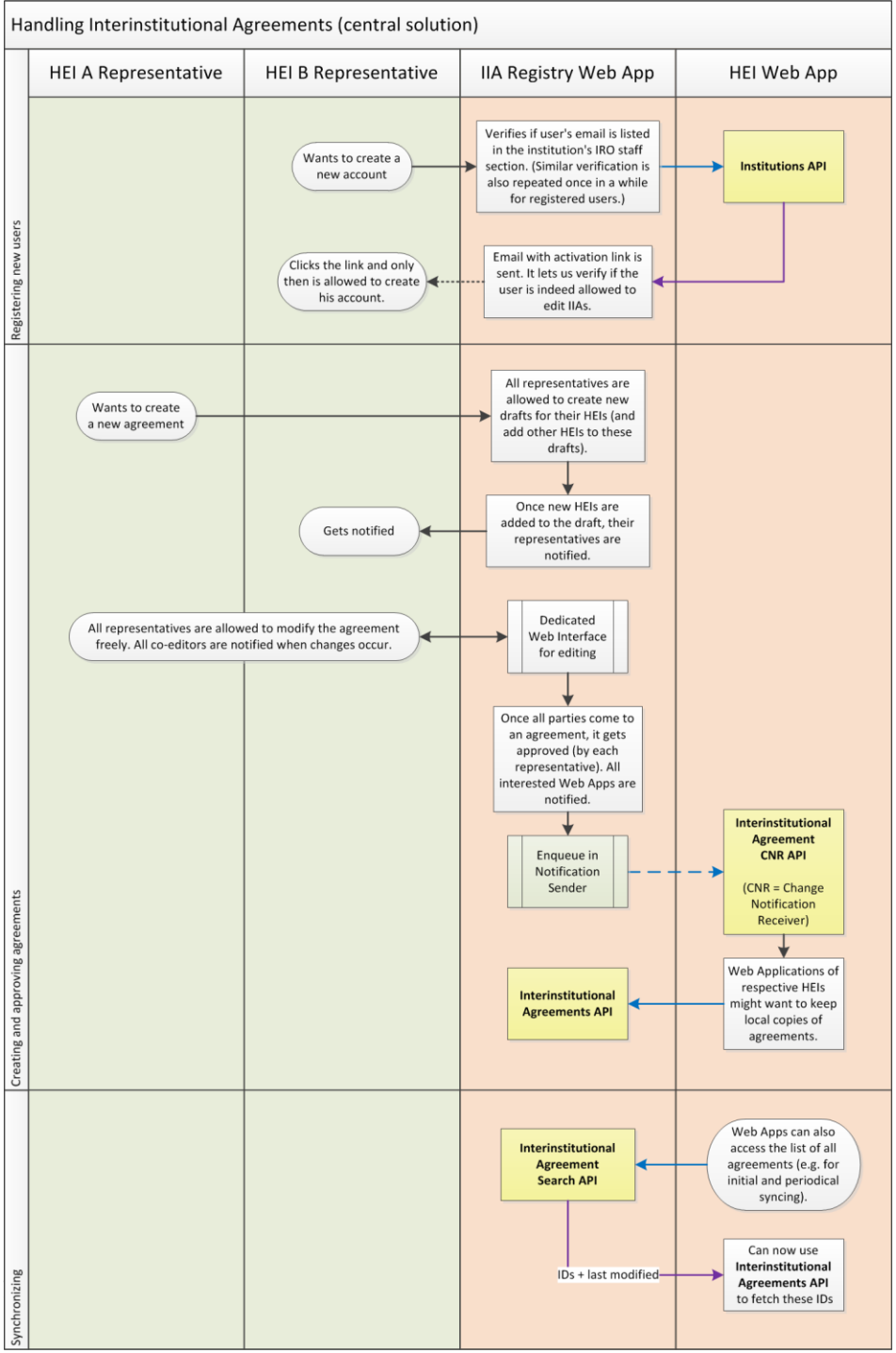
Change Notification Receiver (CNR) and Notification Senders API

- CNR is a callback URL for push notifications.
- Partners subscribe for notifications by implementing a chosen CNR API and publishing it in their manifest file.
- CNR URL is triggered whenever a related entity is updated. This allows the partners to keep fresh copies of data.
- Server responsible for the entity must be able to send such notifications (this ability is also published in the manifest files).



Interinstitutional Agreements (IIA) API

- Starting point of each mobility process.
- IIAs might be stored in a central **EWP IIA repository**.
- It might be a web application (with a user interface), which keeps track of all changes and provides the latest copy of the agreement to all partners.
- Final scenario to be supported still under discussion among partners (centralized vs decentralized approach)



EUNIS THESSALONIKI 2016

Summary

- State of work in June 2016
  - Use cases and recommendations for developers.
  - Design of architecture, security, data flow.
  - First versions of API specifications (under review by the project partners).
  - Data model and data format – work has started.
- Soon
  - Design of data formats (data types for API parameters).
  - Registry available for testing.
  - Libraries for connectors.

Summary

- Work in progress, but EWP partners are open for discussion.
- Call for cooperation.
- **Acknowledgments** – tribute to all project partners for their tremendous job which makes it all happen.

Additional information

- EWP website: www.erasmuswithoutpaper.eu
- GitHub: github.com/erasmus-without-paper
- EWP for developers: developers.erasmuswithoutpaper.eu

Register for EWP Newsletter to keep in touch!