EMREX -> EWP migration

Things to do

The order of steps below should guarantee that all specification errors and misunderstandings will surface reasonable quickly. Since I wasn't able to spend so much time as I usually do for desigining this, these misunderstandings are *very* likely to occur :)

1. Create a draft version (`v0.x.x`) of *EMREX Gateway API* specification.

   This API will receive requests in similar format to existing EMREX NCPs. However, it will make use EWP's security methods. This means, among other things, that the client will be able to sign its requests with HTTP Signatures, and EMREX Gateway API will be able to verify exactly who's asking for what. Using EWP security methods allows NCPs to state that they allow anonymous access or not.

   To allow more efficient adoption, the specification of EWP manifest entries for this API should allow the servers to indicate two versions of the server implementation:

   - The server should be able to indicate that he implements the original `<emrex-ncp-direct>` API, exactly as in original NCP specification.
   - The server should be able to indicate that he implements `<emrex-ncp-gateway>` API, which adds another (optional) layer of client security.

   In practice, the servers will probably specify only the second option (`<emrex-ncp-gateway>` API), but we want to have the first one too, for backward compatibility.

2. Poland implements the first step of the EMREG proxy inside EWP Registry Service.

   The first version won't serve the JSON response yet. In fact, it will do the opposite. EWP Registry Service will periodically (e.g. every 5 minutes) fetch the current version of EMREG JSON response, and integrate it into its EWP Registry catalogue.

   It will dynamically map all EMREG entries to existing EWP institutions, and add `<emrex-ncp-direct>` manifest entries for each of them. If the institution doesn't exist, it will create it.

   The effect: EMREX clients will now be able to fetch all EMREG data directly from EWP catalogue. **They won't do that though** (because EMREG is still the only official catalogue for EMREX). We only need the *ability* to do so, in order to test the new clients in the next steps descibed below.

Note that it is perfectly safe to dispatch this new code directly to production EWP Registry server (because all clients, except our test clients, will ignore these new API entries). We trust that Norway won't put invalid HEIs in its EMREG file, so it's safe.

3. Norway implements *EMREX Gateway API* at its country-wide NCP server.

   The new version of Norwegian NCP server will publish its own EWP manifest, which will cover all Norwegian HEIs with *EMREX Gateway API* support.

   Possible implementation (it can be done in variety of ways): Gateway will process the request, save its parameters in a local database (along with its return URL), and generate a redirect URL with a short-lived OTP token (which will also be saved in the database, and will later allow the NCP to connect the user's browser session to this request). See *Brief design of EMREX Gateway API* below.

4. Poland implements *EMREX Gateway API* at its HEI-specific NCP servers.

   Similar as Norway, but each host will cover only a single Polish HEI.

   Norway and Poland are the first, because we should try to implement this API for both `singleFetch` and non-`singleFetch` NCPs first, and test it, before we release it and ask all other partners to implement it.

5. Both Poland and Norway implement alternative EMREX clients and test if everything works correctly.

   The alternative client will make use of EWP Registry's catalogue instead of EMREG. It will find both types of API entries (`<emrex-ncp-direct>` and `<emrex-ncp-gateway>`), and support both of them.

6. Poland and Norway unify their proxies and redirects.

   Variant A: Poland takes ownership of old EMREG entries.

   Up to this point, Norway was still responsible for keeping EMREG registry up-to-date. If I understood correctly, Norway doesn't want that responsibility anymore. If this is true, then we need additional changes:

   - Poland modifies the EWP Registry code so that it will now fetch the file from local filesystem, instead of fetching it from Norway's server. Let's call this internal underlying EMREG file `old-emreg-entries.json`. (If we go with *Variant B* described below, then Poland will still fetch this file from Norway's server, but from a different URL.)

   - Poland adds EMREG compatibility endpoint to EWP Registry implementation (not sure if we need it in EWP Registry API specification - probably not).

This new endpoint will act as simple filter and converter. It will fetch all NCP servers from EWP Registry Catalogue, find the ones compatible with older clients, and serve their data in EMREG's JSON format.

In more detail, it will:

- Extract all `<emrex-ncp-direct>` APIs from EWP catalogue. Some of these come (indirectly) from `old-emreg-entries.json`.
- Extract all `<emrex-gateway-api>` APIs from EWP catalogue. These ones will need to be filtered. We only want such entries, for which *EMREX Gateway API* states that it supports anonymous requests.
- Serve a "new EMREG file" somewhere in the EWP Registry's domain, e.g. http://registry.erasmuswithoutpaper.eu/emreg-compatible-response.json
- The new file contains all entries taken (indirectly) from `old-emreg-entries.json` *plus* new entries dynamically generated from `<emrex-gateway-api>` entries from partners' manifests.
- All entries will be served as `singleFetch` NCPs.
- The NCP URLs which came from `<emrex-gateway-api>` will be replaced by proxied URLs (because the old clients won't know how to handle `<redirect-url>` XML responses returned by *EMREX Gateway API*, the proxy will need to retrieve them, and then send an actual HTTP Redirect response directly to the user's browser). This proxy will be implemented directly in the EWP Registry.
- If a single institution turns out to be covered by both `<emrex-ncp-direct>` and `<emrex-gateway-api>` APIs, then `emreg-compatible-response.json` should contain only the former. Having the latter "override" the former doesn't serve much purpose, just generates traffic (one extra redirect). If we want to force such override, then the proper way do so is to remove all `<emrex-ncp-direct>` entries (this may require modifying the internal `old-emreg-entries.json` file, because some of these entries will be generated based on its contents).

- Once this is done, Norway sends their current EMREG file to Poland, and definitely stops adding any changes to it. (If we choose variant B described below, then Norway moves the EMREG file to a different URL instead of sending it to Poland.)

- Norway verifies that http://registry.erasmuswithoutpaper.eu/emreg-compatible-response.json looks fine for their needs.

- Norway configures EMREG to perform a redirect to http://registry.erasmuswithoutpaper.eu/emreg-compatible-response.json

- Norway and Poland remove their entries from `old-emreg-entries.json`, and remove the support for the original NCP protocol from their NCP servers. This step is needed to verify if proxy works correctly. Before we release the official version of the *EMREX Gateway API*, we should wait some time to verify that all EMREX communication works properly in this setup.

Variant B: Norway keeps ownership of old EMREG entries.

This alternative is similar to *Option A*, but `old-emreg-entries.json` stays on Norwegian servers. It will be served at a different URL though (because the original URL will now redirect to EWP's dynamically generated JSON). If we choose this option, then all steps decribed in *Option A* are still required - the only difference is that `old-emreg-entries.json` will be fetched dynamically every 5 minutes.

7. We release the official version (`v1.x.x`) of *EMREX Gateway API* specification.
8. All other partners implement the new versions of their NCP servers and EMREX clients.
    - If the partner doesn't have a need to verify the clients, then they only need to update their EMREX clients. There's no real need to update their NCP servers (because all new clients are required to work with old NCP servers).
    - It's always worth to implement a new EMREX client though. If partners fail to do that, then their clients won't see NCP servers which require the client to authenticate itself (the backward compatible EMREG response does not include such NCPs).

Brief design of EMREX Gateway API

This API will be called by the EWP client, *not* by the user's browser (as it was in NCP API). Both requests and responses can be signed (and/or encrypted) with EWP security methods. Clients and servers will negotiate supported security methods in the usual EWP fashion (with help of the EWP Registry). This takes care of problems such as certificate expiration, etc.

If the request is valid, and the gateway decides that the client can receive the results (i.e. he had either paid for the service, or is one of the clients which can get the results for free), then it will return an URL at which the EWP client should *redirect* the end user's browser. This extra step is required because user's browser is not able to safely execute some of the EWP's security methods. After the user is redirected, the process will work the same way as usual (user logs in, picks courses, the results are sent to the callback URL provided in the initial EMREX Gateway API request).

If the request is rejected, it will return a proper error message to the EWP client (which in turn might either alert the user, and/or an administrator).

Example success response:

```
<response xmlns="...">
    <redirect-url>https://usosweb.uw.edu.pl/ncp/?token=123456<redirect-url>
<response>
```