EMREX -> EWP migration

Things to do

1. Create a new *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.

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

   Norway's 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.

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

   This will work the same way, but each host will cover only a single Polish HEI. We should try to implement this API for both `singleFetch` and non-`singleFetch` NCPs first, and test it, before we ask all other partners to implement it.

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

5. Optional: If we value backward-compatibility with the previous EMREG, then we need some more actions.

   - (a) Rewrite EMREG.

     The new EMREG will act as a proxy. It will serve the data from both (a) old EMREG file, and (b) new EWP gateway APIs extracted from EWP Registry, in a JSON format compatible with old EMREX clients. In short, it will:

     - Take both old EMREG file, and EWP catalogue.
     - Serve a new EMREG file, with all entries from old EMREG file *plus* new entries dynamically generated from EWP catalogue (possibly replacing the old EMREG entries which seem to be redundant).
     - New entries will be created only for such EWP catalogue entries for which *EMREX Gateway API*states that it supports anonymous requests. All such new entries will be served as `singleFetch` NCPs. The NCP URLs will be replaced by proxied URLs (because the old clients won't know how to handle redirect-urls 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).

   - (b) New EMREX clients (the ones which make use of EMREX Gateway APIs instead of EMREG) should *also* make use of EMREG, and support both flows.

If we don't want backward-compatibility, then we can skip both (a) and (b) above. However, this will mean temporary downtime for EMREX:

- New clients won't see old NCP servers. They will only see EMREX Gateways.
- Old clients won't see new EMREX Gateways. They will only see old EMREG-based NCP servers.

6. Those of the EMREX partners which are also partners of EWP, implement gateway APIs, and new EMREX clients.

   If we go with the backward-compatible approach in the previous step, then they will also have the choice to *not* implement anything new:

   - Their NCP API will still work for all newer clients, because of the proxy implemented in EMREG.
   - Their SMP clients will still be able to fetch ToRs from all NCPs which have decided to allow anonymous clients to access them (because EMREG will still publish such NCPs, even if they came from EWP Registry).
   - Their SMP clients won't see NCP servers which decided to *not* serve students' ToRs to anonymous clients (because such clients won't be published by EMREG, they will be only published in EWP Registry).
   - It will also be possible for them to implement only new EMREX clients, but skip the gateway APIs. As long as EMREG lives, and as long as they want to allow anonymous clients, such EMREX implementation will be fully functional.

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 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>
```