Improvements to the Login Flow

**Global Participation Fund (GPF)** has provided fund for the implementation of this feature in response to the TUHH's Project "ORCID Login improvement for DSpace-CRIS"

TUHH has entrusted the implementation to 4Science. The implementation is expected to be completed by Oct 2023.

One time again, we will be happy to share the outcome of the project with the community, the improvement will be available out-of-box in DSpace-CRIS since 2023.03.00 (maybe in 2023.02.00) and offered to DSpace for a wide adoption.

4Science would be happy to discuss with Institutions that aims to anticipate the porting of this work to plain DSpace but it is unlikely that 4Science would be able to donate it for DSpace 8 as already involved, on a volunteer basis on many other features for DSpace 8. In the worst scenario 4Science expects to be able to donate that for DSpace 9.0 if the idea is supported / liked by the DSpace community

**Problem statement**

The login via ORCID process is unfriendly for new users and existing users that have not yet connected their existing account to ORCID. When a user login via ORCID for the first time there are two scenarios that lead to error or suboptimal results:

1. the loggedin user has set the privacy flag of their email address on the ORCID side as private, this would prevent DSpace from getting the email address and the login process cannot be completed as the email is required to create a new user or identify an existing dspace account (eperson) to link with ORCID
2. the loggedin user has an email address on their ORCID profile that differs from the one used in their DSpace account. This lead to the creation of a new (duplicate) user account or to a login failure if the autoregister feature is disabled

**Overall idea of the solution**

For the problematic scenarios we want to inform the user about the specific issue: missing or unknown email address offering them the chance to manually fix the issue providing the right email address or login in DSpace to allow us to see which is the existing account that should be linked with ORCID.

**Technical proposal**

The solution is not trivial to be implemented, as DSpace-CRIS would need to store somewhere the information received from ORCID to be enriched with the email manual input by the user, then to be retrieved in a subsequent step to complete the login process or the registration process. Moreover, we need to validate the email provided by the user as otherwise it could become a security risk allowing users to gain access to existing account just knowing their email address.

To design the solution we have produced the following flow chart
and we have prepared the following wireframes for the new UI involved in the flow

**Page shown to users that have the email address private on ORCID**

Information needed to complete the login process

<table>
<thead>
<tr>
<th>ORCID</th>
<th>XXXX-XXXX-XXXX-XXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last name</td>
<td>Smith</td>
</tr>
<tr>
<td>First name</td>
<td>John</td>
</tr>
</tbody>
</table>

The information received from ORCID are not sufficient to complete the login process.
Please provide the missing information below, or login via a different method to associate your ORCID to an existing account.

Provide email

[Input box]

or

[Connect to an existing user]

[Send email verification link]
Page shown to users that release their email address but no account exists yet on DSpace with the same email (to avoid duplication)

Page shown after that the email has been provided
Page shown after that the email has been validated

Email validated
Your email address has been validated. You can now log in to the system with your preferred authentication method.

Please log in to DSpace

Email address
Password
Log in
Log in via Shibboleth
Log in via ORCID

Page shown if the information in the registration data conflict with information in the existing account
Expected changes

registrationdata should be extended to include

- NetID filled with the ORCID by the ORCIDAuthentication if the login process require the user to provide/confirm the email

- a list of metadata (so a new registrationdata_metadata table with foreign key to the metadatafieldregistry and a text_value for the metadata). These metadata will be copied over the new EPerson or merged in the existing eperson eventually overriding existing information if any (a parameter must list which metadata are going to be overridden)

- when exposing the registrationdata metadata we should also include an override attribute with the eventual value already set in the eventual eperson of the current user (if loggedin) or corresponding to the email address specified in the registrationdata. We cannot rely on a link to the eperson resource because the user that is retrieving the registrationdata via token could be not loggedin

- to merge a registrationdata over an existing eperson we plan to use a POST request to `/api/eperson/epersons/:uuid?token=<token>`. Only user that are logged-in as the corresponding uuid can perform such POST. This would replace the existing POST to `/groups?token=xxxx` currently used in dspace-cris to accept invitation to join groups. An override parameter should be used to list the metadata that eventually can be overridden otherwise only new value present in the registrationdata will be set

- registrationdata should expire automatically as it is possible that multiple attempt of connect different accounts leave lot of stale data

- registrationdata should be NOT constraint to be unique by email or netid. as we could have multiple invitation at the same time. Forgot password should cleanup previous forgot password request, maybe this imply that the registrationdata should have a type to identify them. The orcid type will be used when the registrationdata are created as part of an orcid login, self, invitation and forgot will be other two types

The response example in the search method findByToken of the registrationdata endpoint will be updated to include the extra information above.

The rest contract for the new POST to `/api/eperson/epersons/:uuid?token=<token>` will be fully described

The rest contract for the PATCH of the registrationdata to change the email address will be provided. We will clarify in the contract that the associated token will be automatically updated when the email is changed. The response should be 204 as we cannot return the updated registrationdata as this would unveil the new token. The token must be provided as parameter for the patch to validate the request so that attempt to patch a registrationdata without knowing its token would fail with a 401

Other consideration

The same problem affect the shibboleth login but also any other login that via autoregister could lead to duplication of email address due to discrepancy among the IdP.

For shibboleth we expect that applying the change assumed to be necessary in the ORCIDAuthentication to create a registrationdata instead than a new EPerson would be sufficient. For other authentication method more work is probably needed as they don't have a dedicated controller/filter that would deal with the redirection to the registration page on the UI.

In any case to get the perfect solution that could work across all kind of IdP we will also need to perform larger refactoring, not part of this project and suitable for a next step. Indeed, we should extract the credentials from the eperson object creating a dedicated object related many to one EPerson so that email, password and netid would be not just single attribute of the EPerson but moved to this dedicated entity/table and associated with the IdP that use such information as credentials.