Requirements for a Collaboration Database

Authors: Renee Fatemi, Jerome Lauret, Maxim Potekhin, Peter Steinberg, Thomas Ullrich

ePIC Executive Summary

The ePIC Collaboration needs a **Collaboration Database** to carry out its mission. Broadly speaking, ePIC is looking for the core functionality that exists in applications already maintained by the BNL SDCC in the form of the “phone book” Web applications used by EICUG, STAR, and sPHENIX. The Collaboration also needs additional/revised functionality as detailed below.

The most basic functionality of the Collaboration Database is to maintain contact information about the Collaboration members, institutions, and member affiliation with institutions. A member of the ePIC Collaboration can be *affiliated with more than one institution* and this affiliation may evolve over time.

For the most part, data entry and modification will be the responsibility of the ePIC Collaboration Council via its designated members.

ePIC requirements

1. **Extensibility of attributes:** there needs to be the possibility of adding attributes to objects (e.g. members) as needed, without disruptive or

extensive changes to the database schemas and with minimal changes to the interfaces. Equally, there must be a way to *remove* attributes no longer required by the Collaboration. Examples:

a. ePIC needs information on whether a member belongs to a specific working group (or multiple working groups) at a specific point in time, hence group membership needs to be formalized as an attribute.

b. It must be possible to access whether an individual has a specially defined status e.g. “early career” status, and others

c. ePIC shall maintain its official ePIC *Author List* which will change over time. The list is generated by the *Collaboration Council*, and information contained in it should be reflected in the Collaboration Database. The Collaboration must be able to keep track of who is included in the list, for a specific period in time. Such information may not (and likely should not) be reflected in the Web UI itself and only available through data export (see a note on CLI below)

d. Members in Good Standing and Institutions in Good Standing: as is common in many experiments, there shall be criteria defined for declaring individuals and their home institutions as having the “Good Standing” in ePIC, which can be a prerequisite for inclusion in the Author List.

e. Foreign keys: in order to assure interoperability of the “phone book” with other databases and systems used in ePIC and elsewhere, there must be a possibility to add “foreign keys” for querying external systems. An example of a foreign key could be the use of an ORCID (or other unique identifiers) which would then be used to query additional community databases.

2. **History:** The Collaboration Database must be capable of producing a “history report” for any of the attributes assigned to entities stored, e.g.

    a. It must be possible to trace affiliation of a member with his/hers institution(s) over time. History should be preserved - and example of use is that it should be possible to generate author lists from past years or periods.

    b. Same for the Author List and “Member in Good Standing” data (see above)

    c. For accountability, the history should identify who/whom modified the record and when.

3. **Protection of PII:** ePIC shall abide by all applicable rules and policies regarding the protection of the Personally Identifiable Information (PII). In following this principle, information that is, or can be construed as PII must not be accessible via any open (public) interface.

    a. Access to attributes should keep privacy in mind. Not considering service sysadmin access (by default seeing everything) or aggregate information (global pie chart distribution), some attributes may be viewable and accessible by the user only. Attributes may also be made public at the discretion of the member. An example of the former may be non-binary gender information (useful for a demographic / DEI statistics but may not be appropriate for access by any roles). An example of the second may be a phone number, not made public.

    b. To achieve this goal, attributes may be private (no access other than the individual), public or have multiple restricted properties (visible to a community manager, visible by the institution's representative, etc …). In general, the relation between “roles” and access to attributes need to be flexible (to preserve fields considered PII, community administrators may see some but not all). The tool should be flexible enough to accommodate for the suggested use-cases.

    c. ePIC requires a reliable and secure way of performing auth/auth to the managerial functions that give personnel access to the PII. The “authorization” part of auth/auth should include “roles” assigned to specific authorized members of the Collaboration

4. **Collaboration Council (CC):** the ePIC organization differs from that of other entities, in that there is no *Institutional Board* where a single person represents an institution. Instead there is a Collaboration Council, where larger groups can have multiple representatives. The database needs to allow for such a possibility (e.g. keep record of multiple people with the Collaboration Council role, being associated with a single institution).

5. **The Command-Line Interface:** in addition to the Web UI, the Collaboration Database must be equipped with a *Command Line Interface (CLI)* incorporating most of the query and management functions. This CLI must be well-documented. Examples of queries that the CLI will provide:
    a. List of members for particular institutions, exported in CSV or a comparable format (YAML etc)
    b. History of member affiliation with different ePIC institutions
    c. A valid Author List for specific dates in the past
    d. Any type of queries based on additional object attributes (see bullet point “1”) which are not necessarily represented in the Web UI but can still be supported in the CLI
    e. Based on a list of members assigned to a specific service task (which can be defined via an attribute), the CLI will be used to determine the level of support to the task provided by the ePIC institutions
6. **Data Import/Export:** ePIC will need methods to import and export all data as needed, into and from the database. Some of these import/export operations will be limited to the CLI (see above), and some will exist in the GUI. Examples:
    a. Bulk ingestion of membership and institutions data into the database. A primary key should be used for disambiguation (for example, an ORCID). There should be a way to handle disambiguation (name change / what name to use?).
    b. Export (e.g. as a file) of the member data based on certain criteria, or simply bulk export

Additional request for new functionalities for the EIC-UG phonebook

The EICUG has recently migrated their primary email platform from the previous bluehost server to a BNL mailman list. During this migration a significant number of email addresses that were found to have “bounced” were culled from the list. Some of the users associated with those emails have left the field and their emails should simply be removed, others have moved to another institution and we are in the process of updating those addresses. In both cases it would be very helpful to be able to provide a list of emails that need to be discarded and, if appropriate, updated. Finally, it would be helpful if entries for new users and their information could be submitted in list form.

To summarize - these are the new functionalities we are asking to be considered:

1) Ability to discard a list of existing email addresses, without removing the user
2) Ability to replace a list of existing email addresses with updated addresses. This could be done using an `OldEmail;NewEmail` listing.
3) Ability to create new users and enter all appropriate information (institute, email1, email2, ….) with a single list. In this case, disambiguation may use a unique identifier (such as an ORCID).