ORCID Hub Design Principles:

Design Principles

Useful and Usable

The user is happy to use the solution. The solution presents an interface that is appropriate for the user. The solution guides the user through designed functions with minimal need for human assistance or need to reference external documentation.

Scalability

Largest to the smallest, with some cautions around the largest using the small-scale-manual interfaces: aim for the highest common factor.

Globally Reusable

The solution shall be architected, designed, and developed for use in New Zealand and for use by the global ORCID community; flexibility is therefore crucial.

Appropriate Abstraction

An appropriate level of abstraction is needed; the major step here is that of abstracting from having only one authentication mechanism to two (once you've done that, you've done the abstraction, and this is an existing requirement for the NZOH). These three modes for authentication are essential: Federation; OAuth 2; LDAP/ADFS make the first drop here.

The ORCID Hub is a Transactional Engine

The ORCID Hub is not a content-management system. Separate the functional core of the NZOH that does the feature work from any communications and education part from any content management. The NZOH will be a small and independent piece of code, a nice and tight codebase and set of functions.

Contemporary and Consumable DevOps

The ORCID Hub must be easy to build and deploy using contemporary, consumable, relevant CI/CD techniques. The DevOps approach used here needs to be modern and state-of-the-art and aligned with openness:

Lightweight, Stateless Approach

The lighterweight the web framework and the supporting libraries, etc, the better. Where possible, the ORCID Hub will be stateless. Only as complicated as it needs to be (and no more) driven cleanly by the requirements.

Extendability

The transaction part is pretty similar from the perspective of the ORCID Hub regardless of what is being updated on the ORCID profile. The variability occurs at the front-end web assets and in the underlying payload that is communicated.

Singularity of Purpose

Do not duplicate functionality that exists elsewhere (e.g., in ORCID). The initial process is really focused on the onboarding (i.e., creating the link between researchers and their organisations on ORCID). In our system design, we will make as much as possible somebody else’s problem! That is, we delegate wherever possible, and we don’t store anything we don’t need to store: minimum-possible data footprint.

Security Primacy

The solution will be security-conscious and designed in accordance with OWASP design and development principles and guidelines. This is particularly relevant in that the ORCID Hub will be storing ORCID access tokens for all of a country’s researchers, and those tokens need to be safe both from attackers and from each other (i.e., each research organisation’s set of access tokens must be strongly compartmentalised from one another).

Audited

All actions taken through the ORCID Hub must be audited.

Open-Standards-Based

HTML5, RESTful, separates styling from content with CSS, must meet accessibility rules (e.g., WCAG2).