

ORCID in Invenio 1

Torsten Bronger
Central library, Research Centre Jülich
t.bronger@fz-juelich.de

1. ORCiD

- every author/researcher gets a unique number, the ORCiD ID
- thus, a reliable primary key for every author across DB boundaries → easy data transfers
- ORCiD maintains profiles (publications lists, affiliations etc)
- ORCiD is non-profit, and is supported by many institutions

2. Goals

- collect ORCiD IDs of as many colleagues as possible at Jülich, store them in our Invenio instance
- allow colleagues to upload their Invenio publication list to their ORCiD profiles
- no manual intervention of us necessary

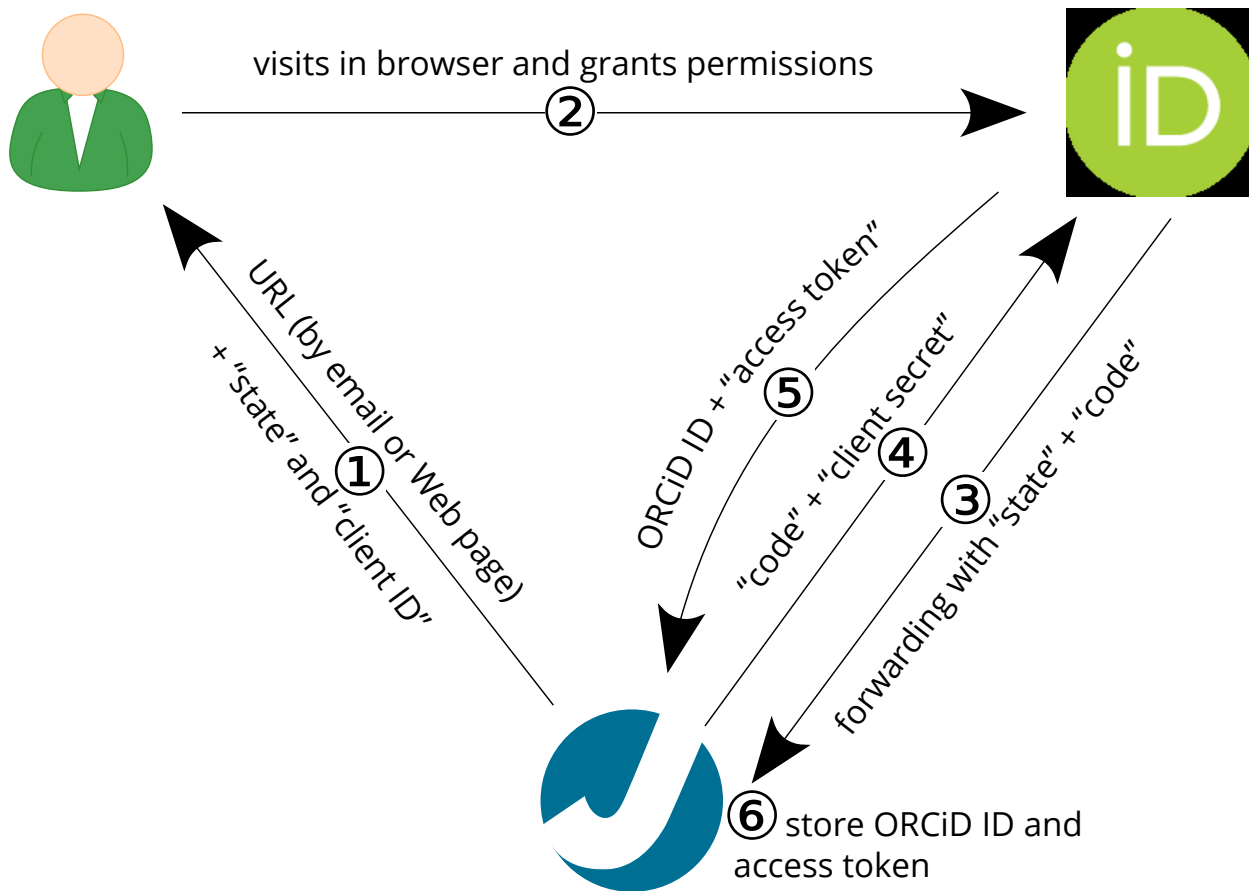
3. Constraints & peculiarities

- slightly lazy: own webserver (Flask-based) for simple implementation
- rather selfish: standalone publication uploader in Go
- absolutely hilarious: Python 2.4 for Invenio, Python 2.6 for ORCiD bindings

4. The webserver

- standalone, i.e. not part of the Invenio WSGI process, own port
- uses Invenio modules to access the Invenio DB directly
- caching of ORCiD IDs & access tokens in SQLite in /tmp

5. OAuth 2



6. URL endpoints

`/orcid`

with "code":

1. store ORCID and access token
2. redirect to `/orcid`

`/orcid/submit`

1. show publications list
2. let user select publications
3. upload to ORCID
4. redirect to `/orcid`

ORCID available: offer upload

else: offer connection with ORCID profile

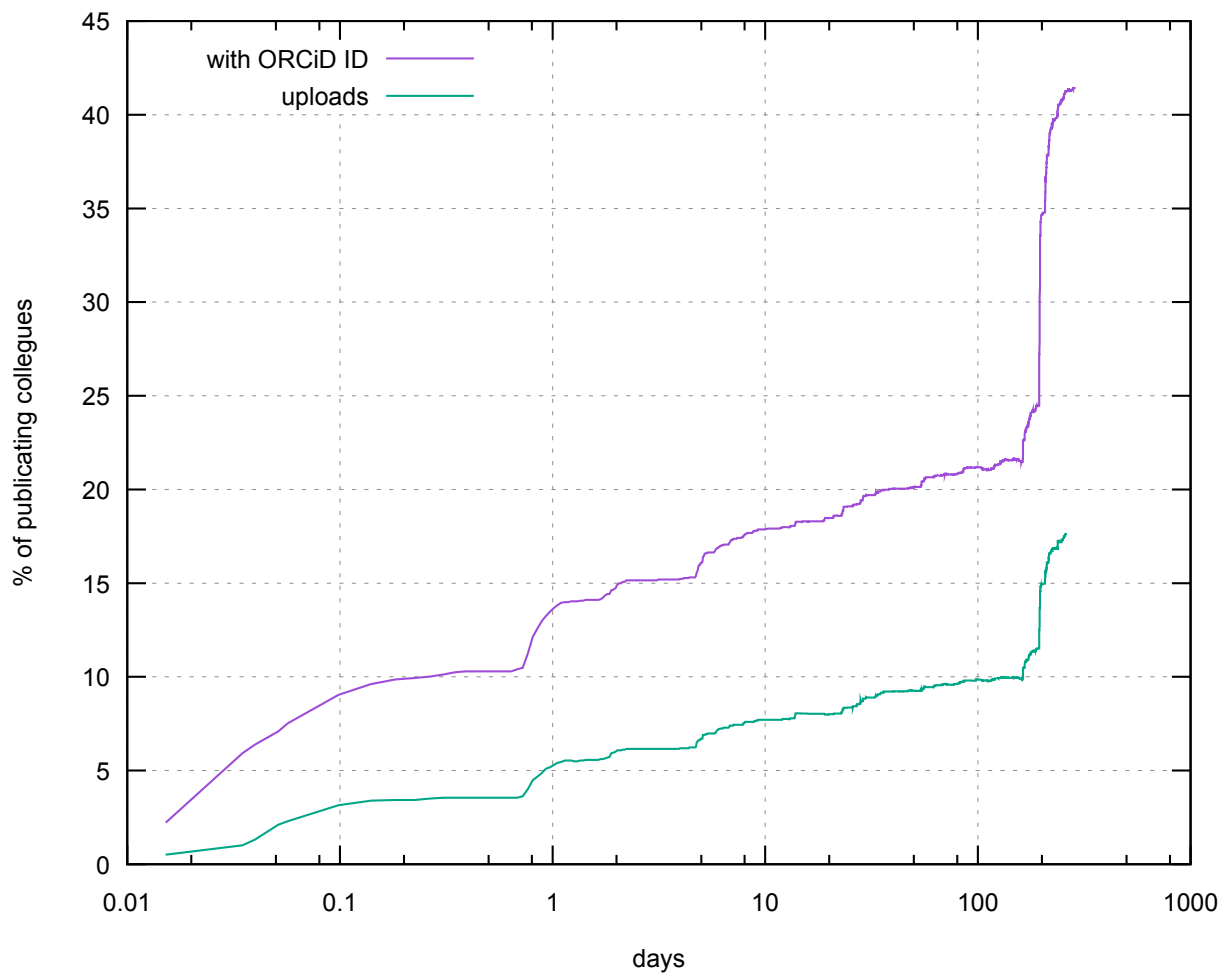
7. Publication upload

- ORCID is sloooow
- publication upload is independent background process
- upload must be throttled
- until recently, one had to upload one-by-one when using ORCID API 2.0
- still not possible to mass-upload all publications in one request

8. Technology is not everything!

- consider legal and regulatory implications
- plan thoroughly how to convince your scientists of ORCID
- the Git repository contains a mass mailer, which can run as a cronjob (use it carefully); and it contains a success measuring program

9. ORCID rate in Jülich



10. Future work

- merge Python 2.4 and 2.6 code into one 2.6 module!
- make use of recently added mass upload feature of the ORCID 2.0 API
- if concurrency is not needed anymore, re-implement Go code in Python
- filter publications eligible for upload (e.g. by year)

11. Check it out!

<https://git.io/orcid>