Where is Invenio 3 going?

Conclusions of the *Invenio v3 Bootcamp*
CERN, 19-21 March 2019

Ferran Jorba, UAB, 3 April 2019

Sumari

- How was the meeting at CERN?
- Personnel changes continue
- Invenio 3 objectives
- Computer requirements
- Pending issues in Invenio 3
- Conclusions
- Future proposals.

The current team and the old school

Lars Holm Nielsen, Nicola Tarocco, Alexandros Ioannidis, Diego Rodríguez Rodríguez, Zacharias Zacharodimos i Karolina Przerwa, Invenio 3 team.

Alexander Wagner (DESY, Deutsches Elektronen-Synchrotron) and Tibor Šimko (CERN, autor d'Invenio 0.x i 1.x), our Marc21 partners.

Work and social life

35 people, 35 laptops, 35 smartphones, and a pile of cables and mice...

Makoto Sumiyoshi (or maybe Masaharu Hayashi?), Lars Holm Nielsen and José Benito González López (Lars boss)

Personnel changes continue at CERN

- There is nobody left from the original team. Nobody.
- José Benito González López is still the coordinator.
- Lars Holm Nielsen is the leader for Zenodo and Invenio.
- Presentations where made by a dane (Lars Holm Nielsen), two spaniards (José Benito González López and Diego Rodríguez Rodríguez) two greeks (Alexandros Ioannidis and Zacharias Zacharodimos), an italian (Nicola Tarocco) and a pole (Karolina Przerwa).
- All them very young and enthusiastic about the newest computing trends.
- In the first Invenio meetings there were a few CERN librarians. Now there is none. That must have some consequence...

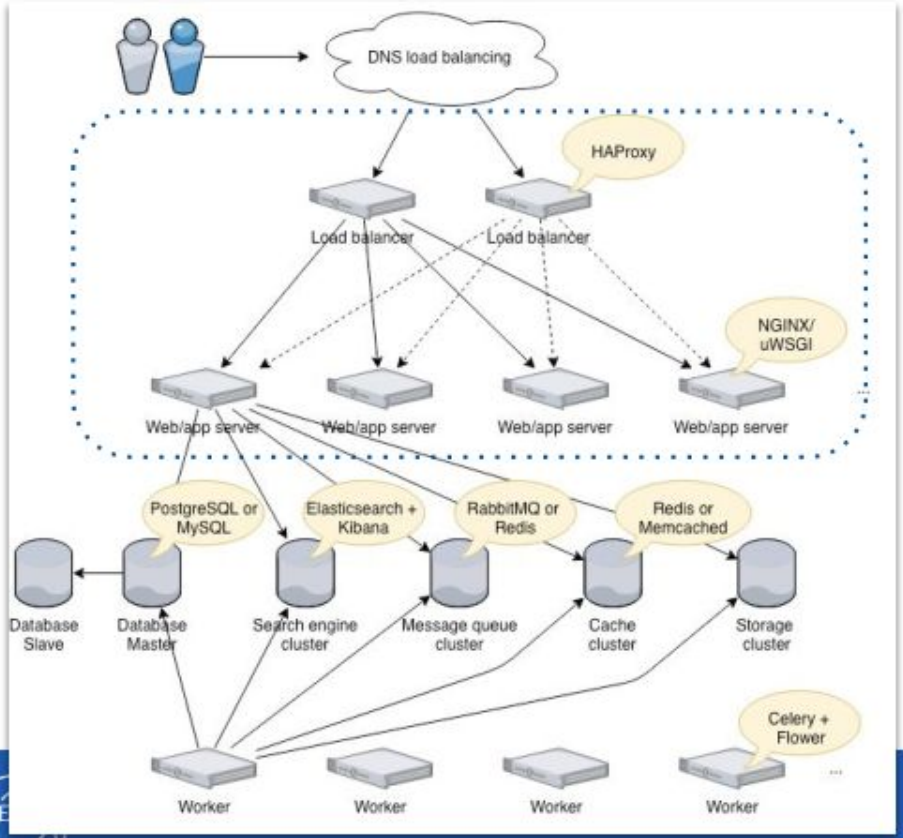
What is (now) Invenio 3?

- A framework to develop repositories.
- It is no longer an installable software. You need to develop Python software using the Flask conventions to do something useful.
- Invenio 3 provides some installable so-called *flavours*: Zenodo (although it doesn't make much sense out of CERN),a future RDM (Research Data Management), etc.
- Software license has changed from GPL 2.0 (a viral license that makes all derivative works also free) to MIT (that allows propietary and closed works).

Invenio 3 computer requirements

- Required:
  - Python, with Flask, pyenv and uWSGI, with a bunch of other Python packages.
  - Web server: Nginx or Apache.
  - SQL server: PostgreSQL or MariaDB (and SQLite for testing).
  - Indexer: ElasticSearch v5 or v6 (incompatibles with the previous ones).
  - Task scheduler: Celery (using RabbitMQ or Redis as storage).
- Optionals:
  - Cache and session handler: Redis or Memchached.
  - Load balancer or high availabilty: HAproxy.
  - Application packager: Docker.

Infrastructure overview



- *Load balancers:* HAProxy, Nginx or others.
- *Web servers:* Nginx, Apache or others.
- *Application servers:* uWSGI, Gunicorn or mod_wsgi.
- *Distributed task queue:* Celery
- *Database:* PostgreSQL, MySQL or SQLite.
- *Search engine:* Elasticsearch (v5 and v6).
- *Message queue:* RabbitMQ, Redis or Amazon SQS.
- *Cache system:* Redis or Memcache.
- *Storage system:* Local, S3, XRootD, WebDAV and more.

Invenio 3, missing (I): bibliographic records

- Marc21 (with the mantra: 'We do not recommend it')
  - They have no interest: they leave it in our hands.
- Forms
  - They recognize that it is very immature.
- Bibliographic records editor
  - Officially it depends on the JSON schema of the bibliographic record.
- Records import
  - Now they call it *loaders.*
- Output formats
  - Now they call it *serializers.*
- Indexing
  - ElasticSearch must be configured for Marc21.

Invenio 3, missing (II)

- Collections
  - They have no interest: they leave it in our hands.
- File handling
  - Not implemented yet.
  - ElasticSearch is not yet indexing full text files.
- Users, roles and permissions
  - It is not clear that we may still have two authentication types: LDAP for internal users and email for external ones.
  - What we saw seemed quite embryonic.
- OAI, server and client
  - They don't seem integrated.

So, what have they been doing, meanwhile?

Good question.

If we have interpret what they have been more interested in during their presentations ...

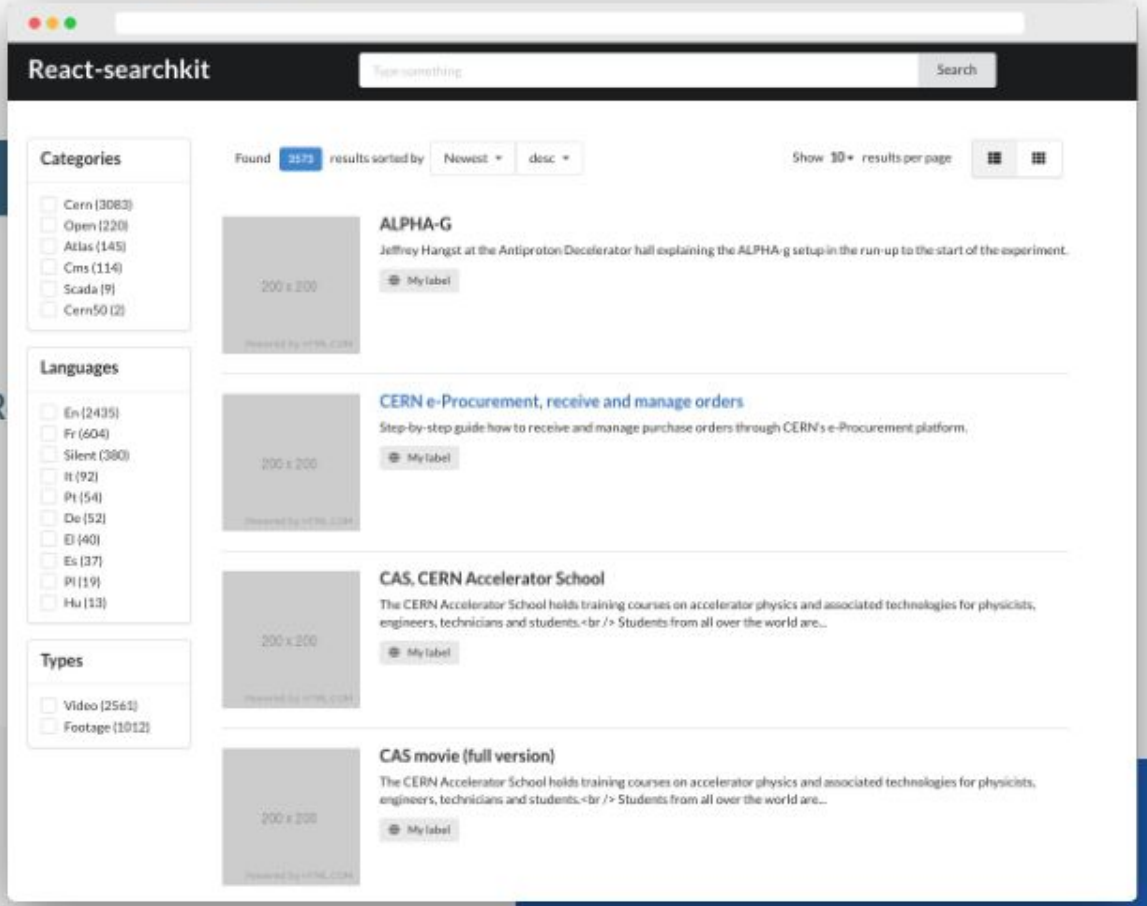
AngularJS to React

React-SearchKit

A simple yet powerful UI search kit built with R

GET STARTED | DISCOVER COMPONENTS

★ Star 12



Walk-through Invenio headers

- Secure defaults
- Mainly set by Flask-Talisman
- Still configurable in a per route basis
- Some more considerations...

```
HTTP/1.0 200 OK
Content-Length: 293
Content-Security-Policy: default-src 'self'; object-src 'none'
Content-Type: application/json
Date: Wed, 13 Mar 2019 21:38:36 GMT
Link: <https://127.0.0.1:5000/api/records/?pag...
Referrer-Policy: strict-origin-when-cross-origin
Retry-After: 3582
Server: Werkzeug/0.14.1 Python/3.6.7
Strict-Transport-Security: max-age=31556926; includeSubDomains
X-Content-Security-Policy: default-src 'self'; object-src 'none'
X-Content-Type-Options: nosniff
X-Frame-Options: sameorigin
X-RateLimit-Limit: 5000
X-RateLimit-Remaining: 4997
X-RateLimit-Reset: 1552516699
X-XSS-Protection: 1; mode=block
```

Relevant presentations (I)

- José Benito González López (CERN), *Welcome*
  - https://indico.cern.ch/event/773969/contributions/3351793/
  - General presentation, Invenio history and Invenio projects at CERN.
- Karolina Przerwa, *Anatomy of a repository*
  - https://indico.cern.ch/event/773969/contributions/3351830/
  - A taste about what it means to install Invenio 3
- Zacharias Zacharodimos, *Data models: Add a new field*
  - https://indico.cern.ch/event/773969/contributions/3351834/
  - Terminology, concept and fear...
- Nicola Tarocco, *Build a simple deposit form*
  - https://indico.cern.ch/event/773969/contributions/3351843/
  - Awe and fear...

Presentacions rellevants (II)

- Karolina Przerwa, *Managing access to records*
  - https://indico.cern.ch/event/773969/contributions/3351844/
  - 100% Python.
- Diego Rodríguez Rodríguez, *Securing your Invenio instance*
  - https://indico.cern.ch/event/773969/contributions/3351845/
  - How to defend from attacks via http headers, tokens and cookies.
- Nicola Tarocco, *Deployment and monitoring*
  - https://indico.cern.ch/event/773969/contributions/3351846/
  - The most complete one from a computer point of view.
- Lars Holm Nielsen, *Roadmap and Invenio development @ CERN*
  - https://indico.cern.ch/event/773969/contributions/3351852/
  - What is missing, from the computer point or view (only).

Invenio 3, conclusions

- We are still far from Invenio 1.x provides.
- Invenio 3 strategy was to take existing pieces to advance faster, but it seems that they haven't succeeded, so far.
- TIND, in hibernate state?
- Translations in stand-by.
- A lot of JavaScript in the user interface. I asked Lars this affects people with visual disabilities, that use browsers without JavaScript, like Lynx. Or home-made robots. He showed surprise and said that they didn't have time to tackle that.

Invenio 3, two personal opinions

- Computer guys in self-contemplation mood:
  - Authors, titles or serial publications didn't appear in the presentations.
  - Not even a pulldown menu to select the search field was accepted, as it depends, according to them, to the data model.
  - Much attention to Python and Javascript frameworks, http headers, high availability, containers, distributed load,…
  - We: 'You are risking to be left alone'. CERN: 'If it has to be that way, ok'.
- Invenio 1.x to 3.x evolution like Omeka Classic to Omeka S:
  - Omeka Classic can be understood, but not Omeka S.
  - Most obvious concepts and known ideas are forgotten and enter into a world of semantic and abstract ideas

So what? What should we do at UAB?

- Follow the Invenio 3 path?
  - We are in the same boat that the germans (contact: Alexander Wagner).
  - They proposed us to join the Join2 consortium (https://github.com/join2) to avaluate it together.
  - There is a proposal to meet in Hamburg, along with the OpenRepositories meeting.
  - The goal would be to create an Invenio 3 flavour that mimmicks Invenio 1.x, installable and configurable.
- Is there an alternative?
  - Should we take a look to what is available?

I have a proposal

May I explain it?

The Muscat alternative (I)

A Ruby on Rails free software that handles:

- Marc21 records, including bibliographic, authority and holdings.
- Multilingual (English, French, Italian and German, with Spanish and Portuguese on the way).
- Full text indexing (via Solr), including attached documents.
- Integrated bibliographic records version control.
- Users, roles, permissions, alerts, etc.
- Multiple forms per database, with autocomplete fields.
- With integrated geovisualization
- Specialized in music cataloguing.
- http://muscat-project.org/

The Muscat alternative (II)

- Public since 2016.
- Development: https://github.com/rism-ch/muscat, now at 5.1
- With collections, and hierarchical? http://muscat-project.org/model.html
- Cataloguers can create folders to organize workflow.
- Comments system in the bibliographic and authority records to review internally.
- Intuitive management of digital objects (?)
- Link to VIAF to import personal name authority files (maybe also to Orcid?).
- Specific music fields (ex: incipits)
- Free software (license no explicitly stated, though).

Muscat: tutorials, presentations and videos

- http://www.rism.info/community/muscat.html
  - General information, mainly oriented to music cataloguers.
- http://www.rism.info/en/publications/iaml-congresses/2018.html
  - Some interesting presentations.
- https://github.com/rism-ch/muscat/wiki
  - Development notes.
- https://www.youtube.com/watch?v=ncnQ-TD9dGM&list=PL9SyOIE9iSYLnyhJz3fGPkI8Xaf_ikDe4&index=3
  - Four videos (start with number 3) that explain internal works, specially how authorities work.
  - As seen on screen, version 3.5.2 is shown. As now the current version is 5.1, probably there have been some improvements.

Muscat examples

- Real example: http://rism-ch.org/. 84.386 records.
  - Record with attached files: http://www.rism-ch.org/catalog/402004752
- The other: https://www.canons.org.au/, 2.216 records.
- Demo server: http://demo.muscat-project.org.
  - 1,125,000 records (*musical sources*), 111,700 personal authority records, 70,400 institutional authority records, and 34,500 secoundary literature records: 1,248,000 total records.
- Training server: https://muscat-training.rism.info
  - Apparently with the same records than above.
  - We can log in in the internal zone:
    - Users: training01@rism.info to training99@rism.info
    - Password: password

Muscat is also based on preexisting pieces (I)

- Ruby on Rails, web development platform (https://rubyonrails.org/)
  - Quite popular. We know Redmine.
- Solr, the indexer (http://lucene.apache.org/solr/)
  - Later we'll discuss the relation between Solr and ElasticSearch.
- Ruby Marc (https://github.com/ruby-marc/ruby-marc)
  - In continuous development, it seems very complete.
- Blacklight (http://projectblacklight.org/)
  - “An open source Solr user interface discovery platform”. Used in different library applications.
- Blacklight Marc (https://github.com/projectblacklight/blacklight-marc)
  - Blacklight specific improvement for Marc records.

Muscat is also based on preexisting pieces (II)

Users and permission (http://muscat-project.org/users-and-feedback.html):

- Autentication based on Devise (https://github.com/plataformatec/devise)
  - It appears to allow different authentication types.
- Roles based on Rolify (http://rolifycommunity.github.io/rolify/).
- Autorizations based on CanCan (https://github.com/ryanb/cancan).

Geovisualitzation:

- Dariah GeoBrowser (https://geobrowser.de.dariah.eu/)

That is, they seem to have succeded where Invenio 3 hasn't.

Lucene, Solr, i ElasticSearch

Which relation there is among them? (things I've learned, without practical knowledge)

- Lucene (1999) is a Java based Api. Praised by everybody.
- Solr (2004) is a Lucene based indexer, now integrated in the project.
- ElasticSearch (2010) is another Lucene based indexer, more JSON specific and with a more commercial bias.
- There seems to be a consensus that they are quite comparable (ex: http://solr-vs-elasticsearch.com/), in functionality, performance and ambition.
- Both coexist and Internet is full of arguments favouring one or the other.

Invenio 3 o Muscat 5?

Provisional evaluation about how they fit and how much effort is needed:

- IFMuC
  - Invenio 3: 10% (too large, complex and immature).
  - Muscat: 90% (100% Marc21 and 100% musica, adapt user interface).
- Traces
  - Invenio 3: 20% (also too large, complex and immature).
  - Muscat: 80% (100% Marc21, also adapt user interface and form).
- DDD
  - Invenio 3: 30% (still too large, complex and immature).
  - Muscat: 70% (100% Marc21, also adapt user interface, forms and permissions).

What is missing in Muscat?

Things we should resolve if we want to adopt it:

- Catalan translation (examples of the other translations: https://github.com/rism-ch/muscat/tree/master/config/locales).
- OAI server (but: https://github.com/code4lib/ruby-oai).
- (OAI client, although it hasn't to be integrated, and: https://github.com/fieldhand/fieldhand).
- Collections, or Invenio-compatible collection URLs.
- More community, more installations? Although it may be an advantage.

That is, those exist also in Invenio 3.

Strategic proposal

1. Evaluate first with more detail how Muscat fits IFMuC or Traces (volunteers?).
2. Let's install it, let's test it and learn it.
3. Let's design, translate and adapt.
4. If it works, move it to production.
5. Repeat it for the other (Traces or IFMuC).
6. Review how Invenio 3 has evolved since.
7. Evaluate Muscat for DDD.
8. Let's decide.
9. Let's do it.