indico

Version 2.0

Under the Hood

Adrian Mönnich

Indico Workshop 2.0
October 2017

The last years

2004-2014 – ZODB

- One decade on ZODB
- Massive growth

2014-2017 – SQLAlchemy/Postgres rewrite

- Frontend changes are nice and people see them
- Backend changes are the big thing “under the hood”

The Problems

The Database

ZODB

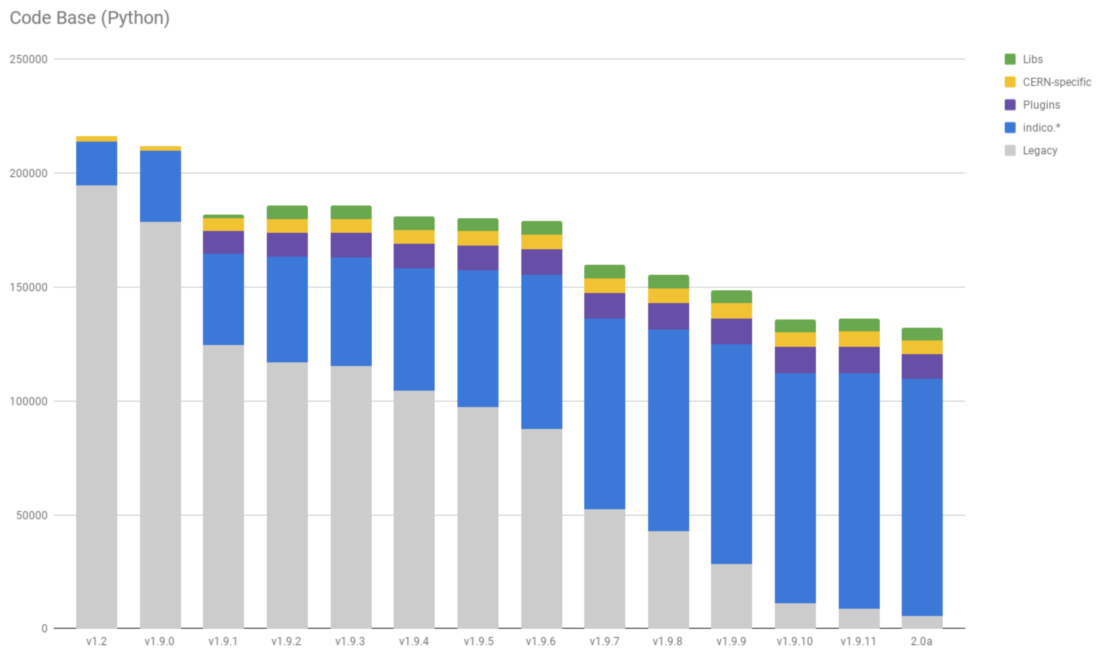
- Poor maintainability (no DB-level indexes)
- Not scalable
- Niche product
- Lack of active community
- New developers are unlikely to have used it before

The Code

- Legacy codebase
  - Grown over 10 years
  - Less thorough code reviewing
  - Unused code
  - Duplicate code
  - ➔ Hard to maintain

The Rewrite

Code Base Evolution



97%

Amount of legacy code removed

39%

Total codebase size reduction

Quality

Modern Python

Getters/Setters.. Anyone likes Java?

```
def getdetailPayment(self):
     return self._detailPayment
def setdetailPayment(self, detailPayment):
     self._detailPayment= detailPayment
```

What about HTML in Python files?

```
tmp = """%s<textarea id="%s" name="%s" cols="%s" rows="%s" %s >%s</textarea>%s""" % (
           desc, htmlName, htmlName, cols, rows, disable, v, param)
tmp = """ <td>%s</td><td align="right" align="bottom">""" % tmp
tmp = """%s </td> """ % tmp
```

Structure

Legacy – no real structure in many places

- 22% of total LOC in top-level package
- `MaKaC/rb_*.py`
- `MaKaC/conference.py` (12.5k LOC)
- `MaKaC/registration.py` (6k LOC)
- `MaKaC/review.py` (4k LOC)

Structure

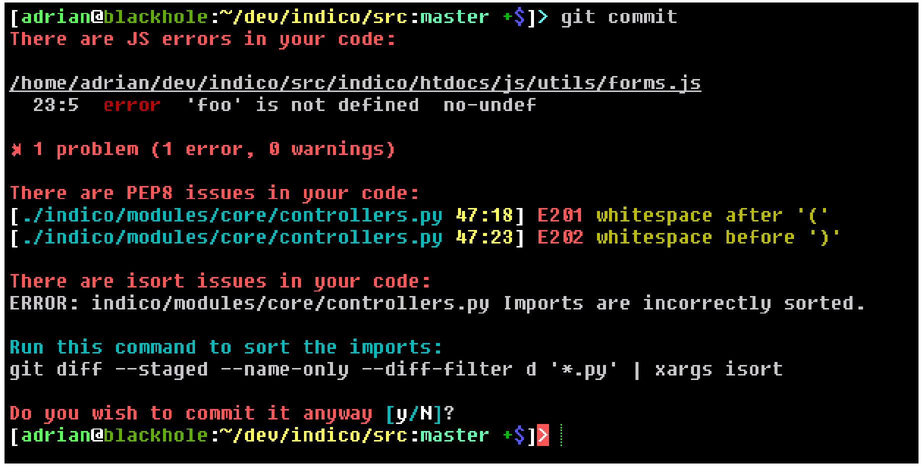
2.0 – separate modules

```
indico/core/{auth,config,logger,…}.py
indico/web/{breadcrumbs,menu,…}.py
indico/web/{flask,forms,…}/
indico/modules/{admin,categories,users,…}/
indico/modules/events/{abstracts,reminders,timetable,…}/
indico/modules/events/reminders/{blueprint,controllers,views,…}.py
```

Code Style

Enforced style

- pep8 / pycodestyle
- isort
- ESLint
- sass-lint
- pre-commit git hook & Travis CI



```
[adrian@blackhole:~/dev/indico/src:master +$]> git commit
There are JS errors in your code:

/home/adrian/dev/indico/src/indico/htdocs/js/utils/forms.js
  23:5  error  'foo' is not defined  no-undef

✘ 1 problem (1 error, 0 warnings)

There are PEP8 issues in your code:
[./indico/modules/core/controllers.py 47:18] E201 whitespace after '('
[./indico/modules/core/controllers.py 47:23] E202 whitespace before ')'

There are isort issues in your code:
ERROR: indico/modules/core/controllers.py Imports are incorrectly sorted.

Run this command to sort the imports:
git diff --staged --name-only --diff-filter d '*.py' | xargs isort

Do you wish to commit it anyway [y/N]?
[adrian@blackhole:~/dev/indico/src:master +$]>
```

Backend

PostgreSQL
the world's most advanced open source database

SQLAlchemy

Flask
web development,
one drop at a time

Jinja

WERKZEUG
LIGHTWEIGHT WSGI TOOLKIT

WTForms

Database Benefits

Consistency

- Enforced using FKs, CHECKs, unique indexes, triggers

Soft deletion

- Things people delete ≠ Things people want deleted
- Easy to undelete e.g. an event

Structured data

- Recovering specific data from a backup is not too hard

Re-using Code

Existing tools & libraries

Celery	Background tasks, Cronjobs
Bleach	HTML sanitization
Marshmallow	JSON serialization
WTForms	HTML forms

Custom libraries

Developed by us
Usable outside Indico
Flask-PluginEngine, Flask-Multipass

Storage Abstraction

Where to store event materials?

- Local filesystem?
- EOS? (at CERN)
- S3?
- Another cloud provider?
- ➔ Simple storage abstraction layer

```
STORAGE_BACKENDS = {
    'afs-prod': 'fs-readonly:/afs/cern.ch/project/indico',
    'local': 'fs:/home/adrian/dev/indico/data/archive',
    'eos-test': 'eos:host=eospublic.cern.ch,root=/eos/workspace/i/indico/test,fuse=true,datestamp=true'
}
```

Unit Tests

- Every application should have tests
  - Ideal world: 100% test coverage
  - Reality: hard to cover everything in a big app
- What do we focus on?
  - Critical functions
  - Edge cases
  - Utils (usually self-contained functions)

Configuration

How to know where the config is

- ~~Patch the code at install time~~ (legacy) 😿
- `export INDICO_CONFIG=/opt/indico/etc/indico.conf`

For your convenience…

- `/etc/indico.conf` or `~/.indico.conf` symlink
- Used if env var is not set

Decoupling

- Avoid hard dependencies between modules
  - “Link registrations by email when a user is created”
  - ~~Call function in *registration* module from *users* module~~
  - Notify subscribers of “user registered” event
- Use dependencies where it makes sense
  - “Create contribution for accepted abstract”
  - Call function provided by the *contributions* module

➔ Calling code from more generic modules is OK, not vice versa!

Frontend

Base Technologies

HTML

CSS

JS

jQuery
write less, do more.

Sass

Choices…

Legacy framework needs to go away!

Our new “framework”?
- Using jQuery & friends
- Lightweight declarative framework (data attributes)
- No real documentation (yet?)

2018: Look out for nice frameworks which…
- …are developer-friendly
- …are lightweight and not too intrusive
- …do not target mainly single-page apps

indico