Indico 2.0

The whole Iceberg

Adrian Mönnich  
CHEP 2016, San Francisco
The last year

2015 – Migration of Indico’s core
   Paving the way for a clean codebase
   Lots of frontend changes

2016 – Almost three years into the migration
   Frontend changes are nice (tip of the iceberg)
   Backend changes are the big thing (hidden parts)
Backend
The Database Problem

ZODB $\rightarrow$ PostgreSQL

Maintainability
Performance
Well-known, easy to use
Base Technologies

- PostgreSQL
- SQLAlchemy
- Flask
- Jinja
- WTForms
The Plan

Migrate Indico to the new stack

Rewrite Indico using the new stack

As clean and extensible as possible

Plugins

Signals/Hooks

RESTful (where applicable)
Results

Complex, not complicated
- ~125 tables in 11 schemas
- ~275 relationships (foreign keys)
- ~80 signal hooks
- ~400 templates

Lots of tables compared to other webapps/sites
- Stack Overflow: ~30 tables (from public DB dump)
- GitLab: 80 tables, 12 FK relationships
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
Benefits

Speed
  Fast queries
  Many statistics can be calculated on the fly

Easy migrations
  Versioned schema (Alembic)
Frontend
Base Technologies

HTML 5

CSS

JavaScript

jQuery
write less, do more.

Sass
Main Principles

Consistent, lightweight UI
  Reusable widgets/components
  Dialogs
  AJAX
The Project
Timeline

Started 2014

- Started with room booking (almost standalone)
- Plugins
- Users / Groups (first “intrusive” change)

Initially scheduled to finish 2015

- Almost finished (finally!)
78%

Amount of legacy code removed (up from 36% at the time of CHEP 2015)
28%

Total codebase size reduction
(up from 16% at the time of CHEP 2015)
Lessons Learned

Dealing with legacy software is hard
  Code full of surprises
  Old functionality not always clear
  Legacy data is not always rarely consistent

Doing the Right Thing™ is hard (and takes time)
  Easy to bike-shed on small things
Current Status

All the core (i.e. intrusive) parts done
  Users, Materials, Contributions, Categories
  Most parts of events

Not much left on the road to 2.0
  Call for Abstracts & Paper Reviewing
  Remaining event parts (e.g. badges & posters)
  Housekeeping
Conclusions
Conclusions

Rewrites are sometimes necessary
  Dealing with legacy code is harder than throwing it away
Users don’t always rarely like changes
  But end up appreciating them once they get used to them
One year of development is not much
  ...and easily becomes two (or three!) years
Indico is huge (~150k LOC, initially over 200k)

Building a strong team is essential
  Processes are important (see our poster), people even more!
http://indico-software.org
Some Eye Candy