

First Indico Workshop

29-27 May 2013 CERN

Conversion Server

Thomas
Baron

Service Description Architecture Conversion Alternatives Future Directions

Service description



Goal

Provide a PDF version of all textual documents uploaded to Indico

Long-term preservation

Multi-platform reading

Converted formats: .ppt, .pptx, .doc, .docx, .sxi, .odp

Interface

On user request only

AVC section meeting

Wednesday, 22 May 2013 from 11:00 to 12:46 (Europe/Zurich)
at CERN (513-R-055)

[Manage](#)

Participants Thomas Baron; Joao Correia Fernandes; Marek Domaracky; Jose Benito Gonzalez Lopez; Franck Jouberjean; Loic Lavrut

Material: [Slides](#)

Video Services Vidyo public room : AVC_section_meeting_indico_214156 [More Info](#) | [Join Now!](#) | [Connect 513-R-055](#)

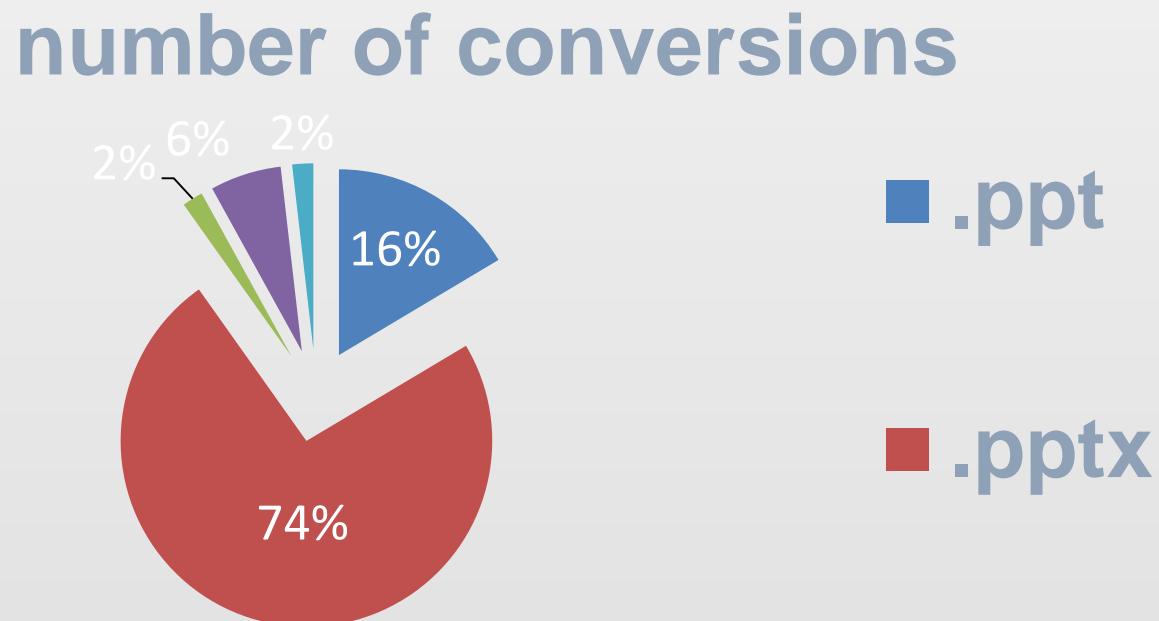
Wednesday, 22 May 2013	
11:00 - 11:05	General Section Information 5' Speaker: Thomas Baron (CERN) Material: Minutes
11:05 - 11:15	Conference Rooms Service 10' Material: Minutes
11:15 - 11:25	Videoconference Service 10' Material: Minutes
11:25 - 11:35	Indico 10' Material: Minutes
11:35 - 11:40	Public Screens 5' Material: Minutes
11:40 - 11:45	Webcast and recording 5' Material: Minutes
11:45 - 11:46	AOB 1'

Service parameters

Asynchronous

About 30 seconds in average

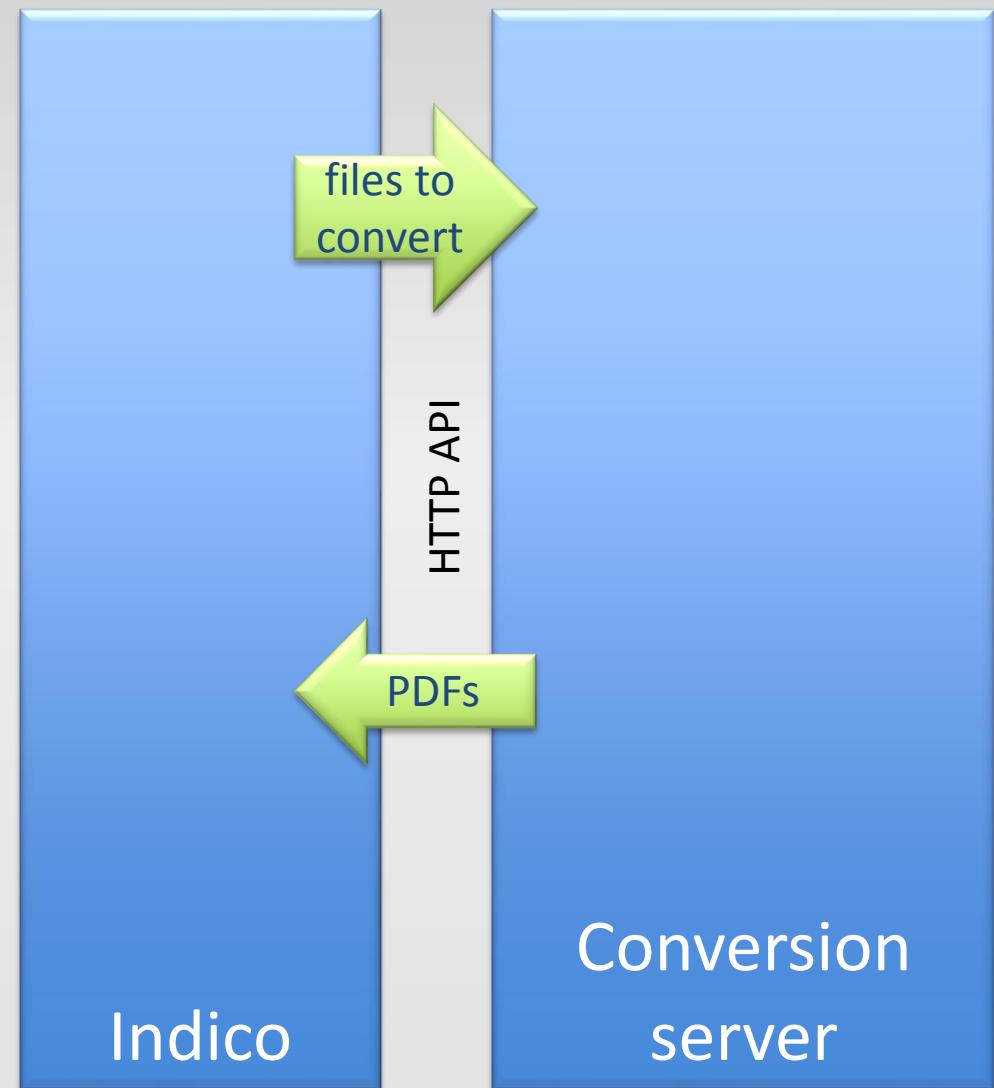
At CERN: An average of 165 conversions per day



architecture



General overview



Integration to indico

Currently all entangled
Indico's core

Configuration in `indico.conf`

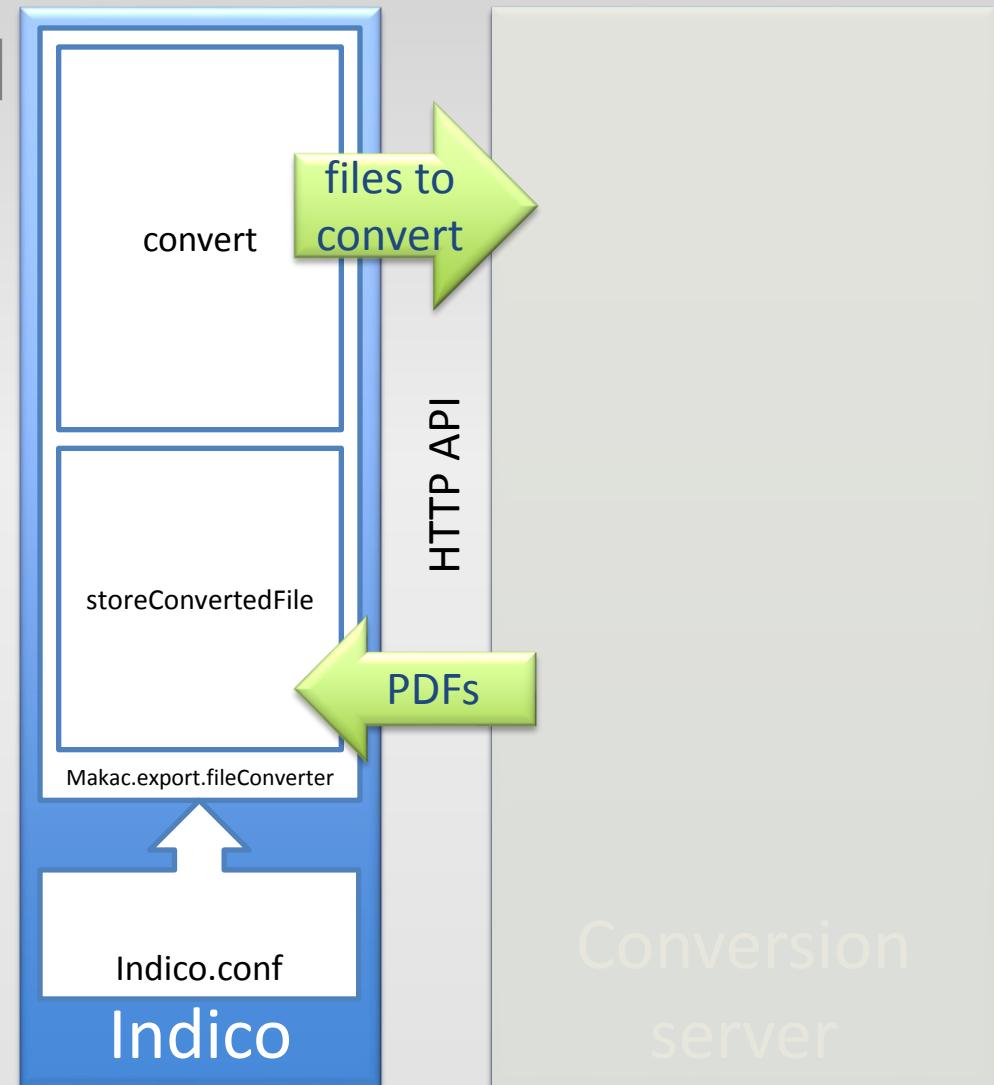
conversion server URL:
`FileConverter['conversion_server']`

callback URL:
`FileConverter['response_url']`

Conversion handled by the
`Makac.export.fileConverter`
class

`convert` function : sends the
file

`storeConvertedFile` function:
gets the converted file
back



Conversion server

side

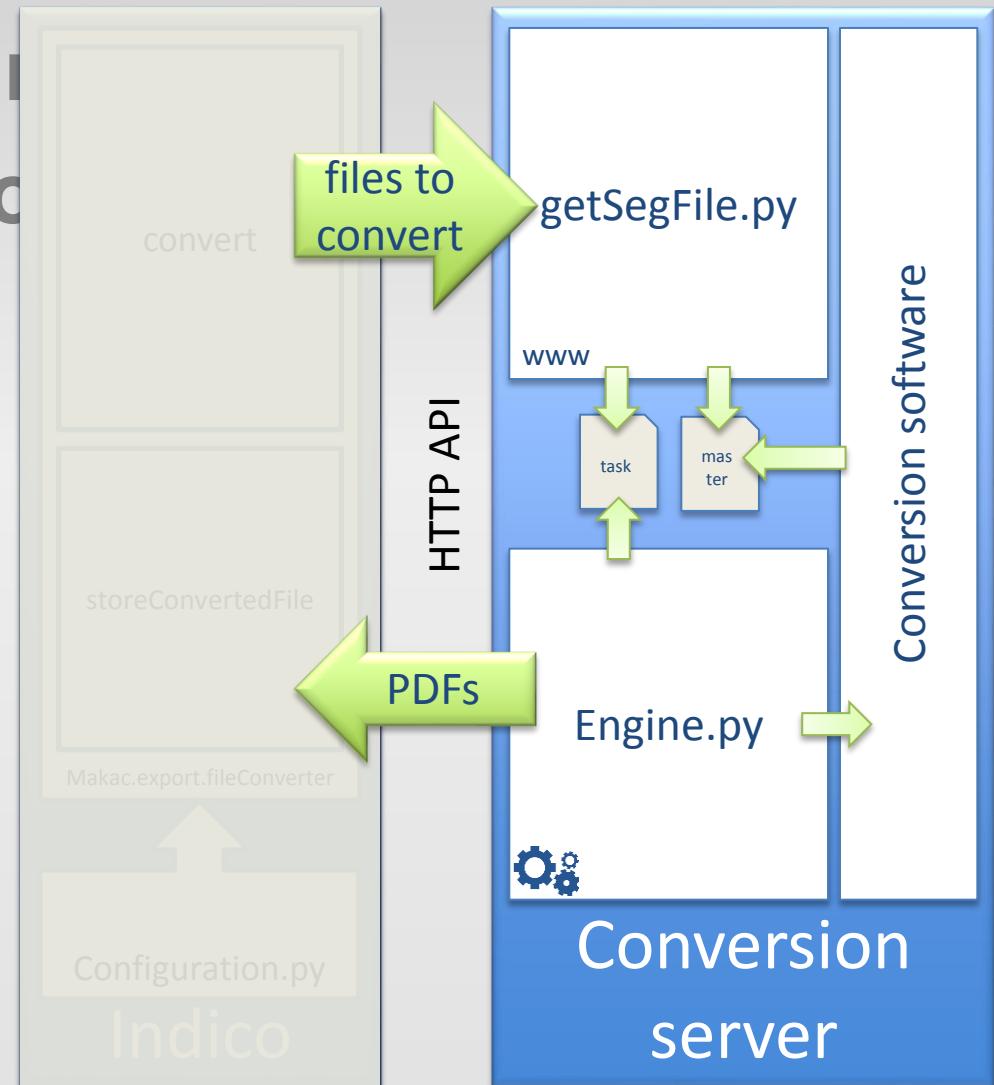
A dedicated server running

non-indico code and software

Web server: IIS (previously Apache)

Listener script: getSegFile.py ; python; receives the file, saves it locally, creates the conversion task (a text file)

Conversion Daemon: Engine.py ; python script in scheduled tasks; parses the conversion task files, and for each of them launch the conversion, wait for its completion and send the file back to the callback URL (to Indico)



Conversion alternatives



Fully home made

Was the case at CERN until 2009

Using direct OLE-automation of Microsoft Office applications

python scripts

Example

Using commercial products

Example at CERN: Neevia Document Converter Pro

OLE automation

Features:

- supports 300 file types
- Com, hot folder, email interfaces
- Watermark, stamping etc.
- **Convert to PDF, PostScript, TIFF (including Class F), BMP, PNG, PCX, JPEG**

Pros:

- More reliable
- Better error management
- Regular updates
- More extensible
- More formats
- Hot folders
- Can be used for other services

NEEVIA doc Converter Pro

Simplified automation code in python

```
NDocConverter =  
win32com.client.Dispatch("docConverter.docConverterClass")  
NDocConverter.DocumentOutputFormat = "PDF"  
NDocConverter.DocumentOutputFolder = output_dir  
NDocConverter.JobOption = "printer"  
  
rv = NDocConverter.SubmitFile( file_path , "")  
rv = NDocConverter.CheckStatus( file_path , "")
```

Future directions



What should be coming

Unfortunately the feature is not directly usable by external instances

Rewrite the conversion server-side code

Very old!

Improve the conversion server monitoring at CERN

Not planned yet

Replace current implementation with a plugin on the Indico side

v1.5 (2015?)



Thomas Baron

Thomas.baron@cern.ch