A new Information Architecture, Website and Services for the CMS Experiment

Lucas Taylor, Fermilab

Vidmantas Zemleris, Vilnius Univ. & EPFL

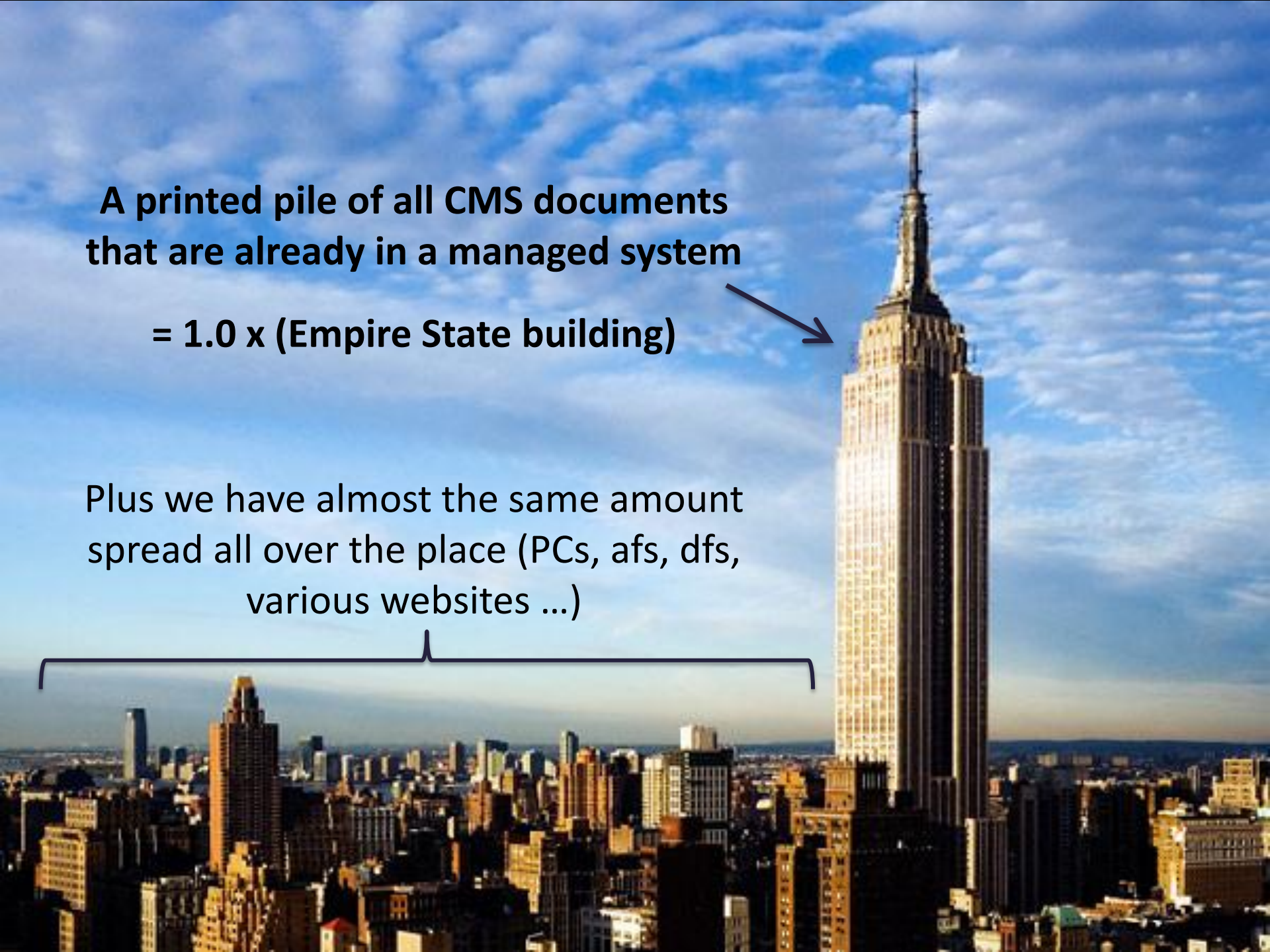CHEP 2012, New York

x 4000 people ... for many decades

A printed pile of all CMS documents that are already in a managed system

= 1.0 x (Empire State building)

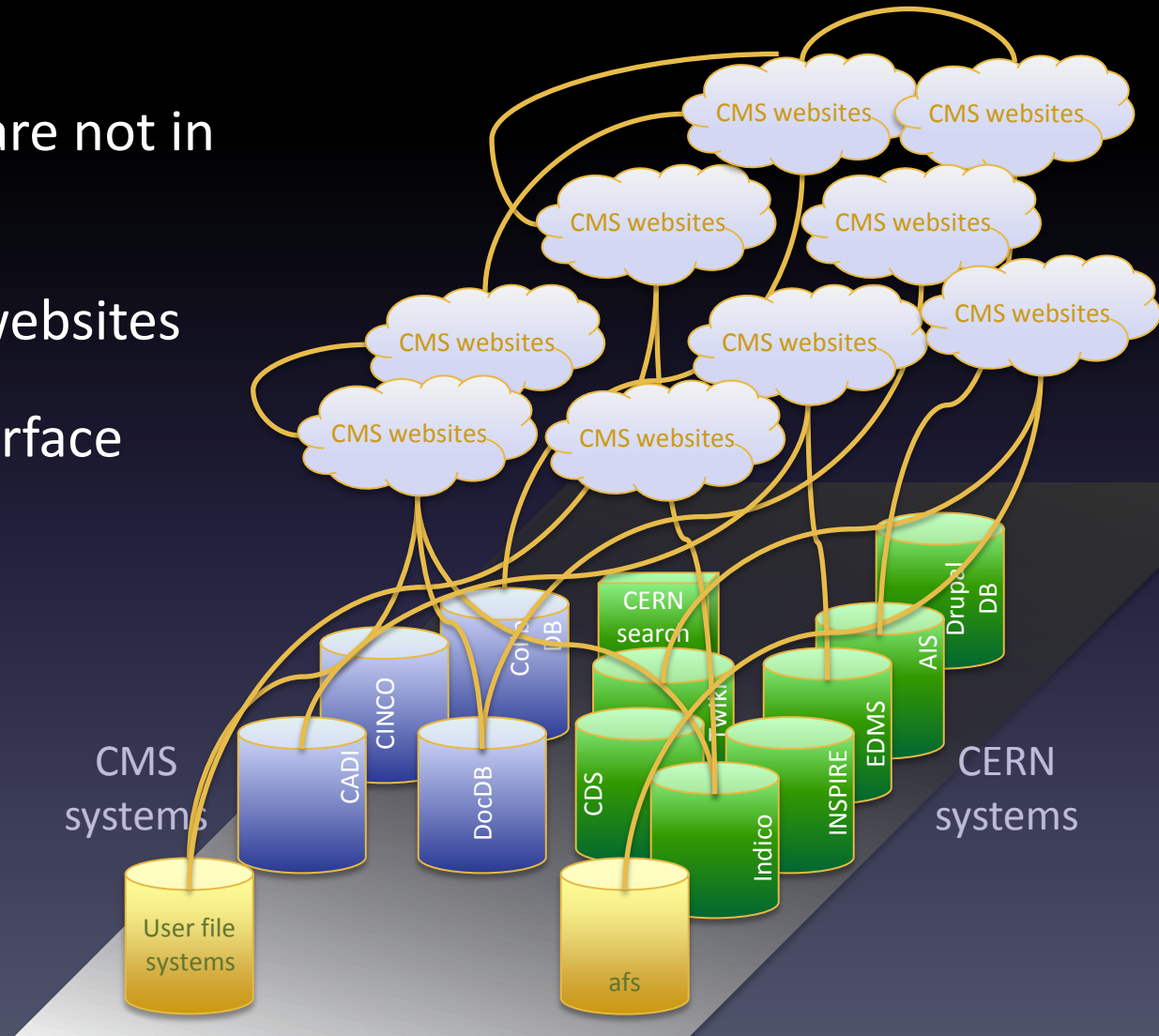Plus we have almost the same amount spread all over the place (PCs, afs, dfs, various websites …)

No. of CMS Documents, May 2012

Document database	**379,219**
Indico (Meetings)	205,206
DocDB (Miscellaneous documents)	128,948
CDS (Photographs)	12,252
CDS (Papers, reports, notes)	11,720
EDMS (Engineering drawings & data)	8,105
CADI (Papers & notes in preparation)	7,513
CINCO (Conference contributions)	5,475
Wiki or file system	**325,957**
Twiki (Web attachments)	118,644
Twiki (Web pages)	25,820
User disks (Misc. PDF, tex, MS office)	100,000
Group afs (Misc. PDF, tex, MS office)	81,493
Grand Total	**705,176**

Information Architecture Challenges (Jan 2010)

- > 50% of documents are not in a document database

- > 500 "official" CMS websites

- No coherent user interface

- No overall navigation

- No overall search

CMS websites

CMS websites

CMS websites

CMS websites

CMS websites

CMS websites

CMS websites

CMS websites

CMS websites

CMS websites

CMS systems

CERN systems

CADI

CINCO

Co?DB

DocDB

CERN search

Twiki

CDS

Indico

INSPIRE

EDMS

AIS

Drupal DB

User file systems

afs

Goal 1:

Manage all CMS documents in a document database

Goal 2:

Create a coherent web interface to all CMS information

Goal 1: Manage all CMS documents in a document database

- CERN systems
  - CDS: papers, notes, photos
  - EDMS for engineering
  - Indico for meetings ...

- CMS systems
  - CADI: physics analysis
  - CINCO: conferences

- Fermilab's DocDB
  - For everything else

DocDB:
Perl, mySQL
Linux file
system

Goal 1:   Manage all CMS documents in a document database

Created a "bulk document upload" tool to find documents
on file systems and websites and import into DocDB

- Scripts search for documents

    (helped by web crawlers)

- Try to auto-extract metadata

    (title, author, date, etc.)

- Make thumbnail images
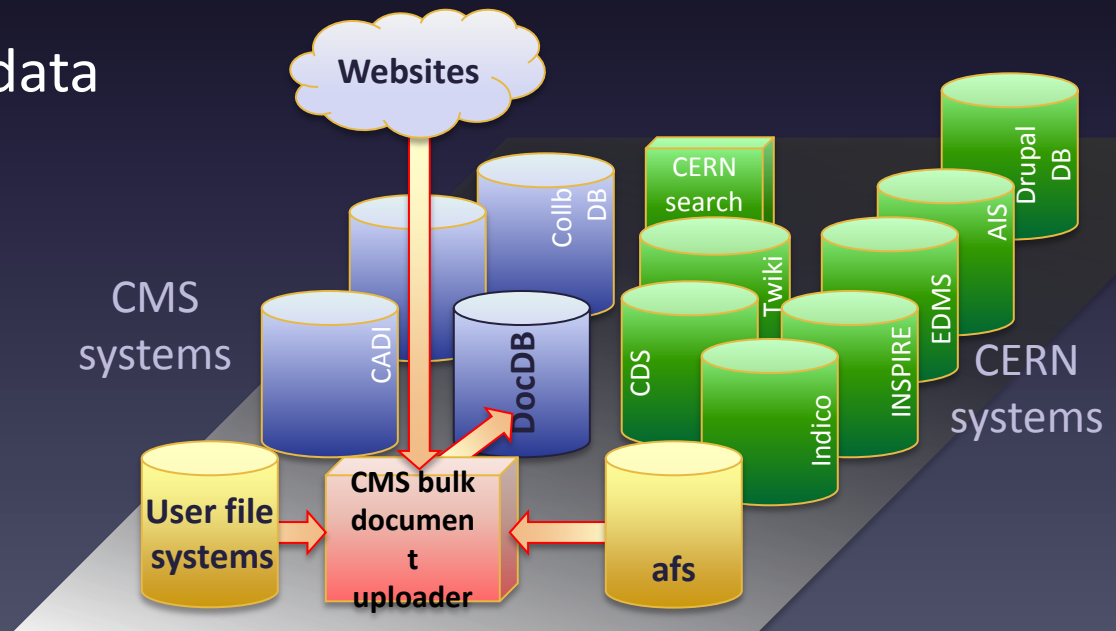
Goal 1: Manage all CMS documents in a document database

- **User web interface**
  - Choose documents to keep
  - Add metadata

- **Script does bulk import to DocDB**
  - XML import interface
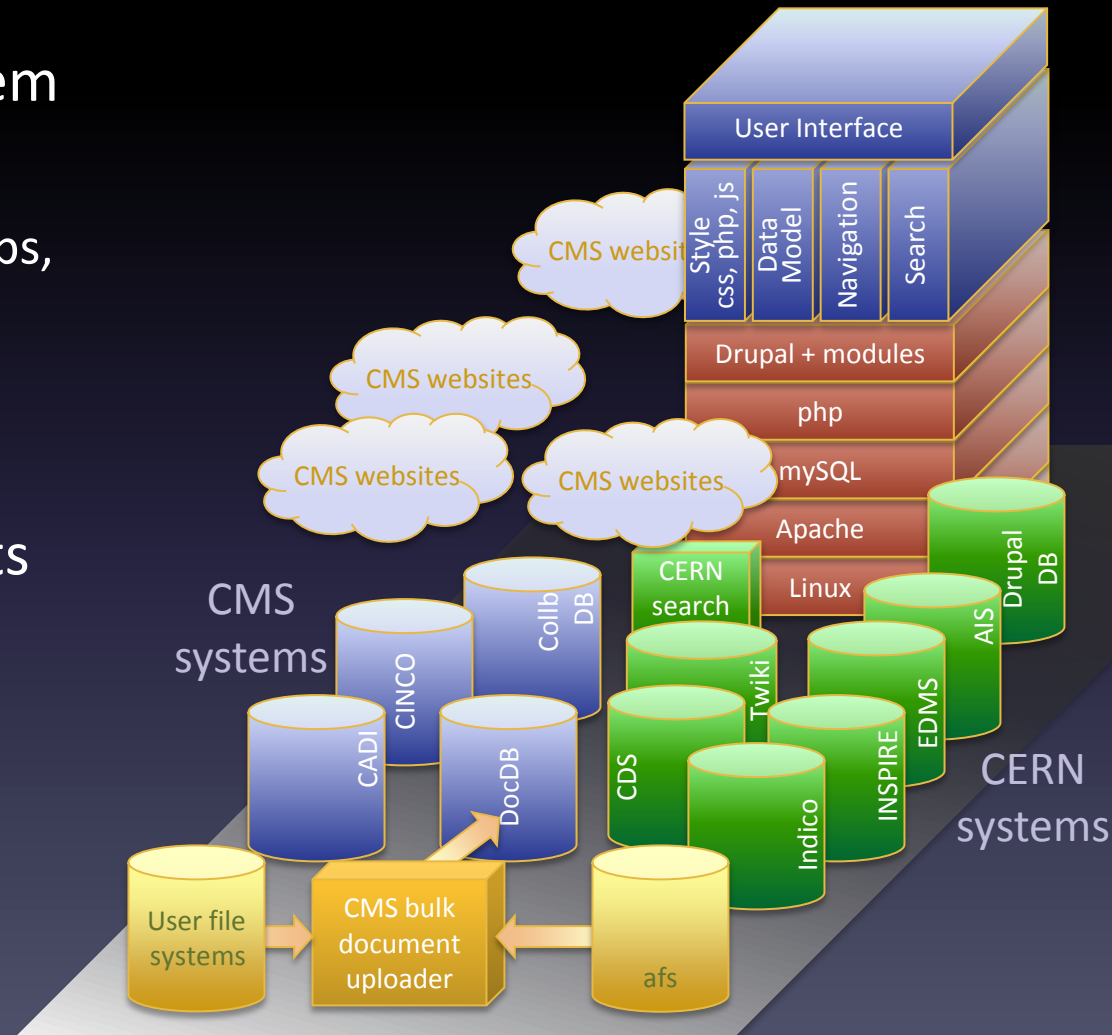
bash,
Python
django,
mySQL,

Goal 1:

Manage all CMS documents in a document database

Goal 2:

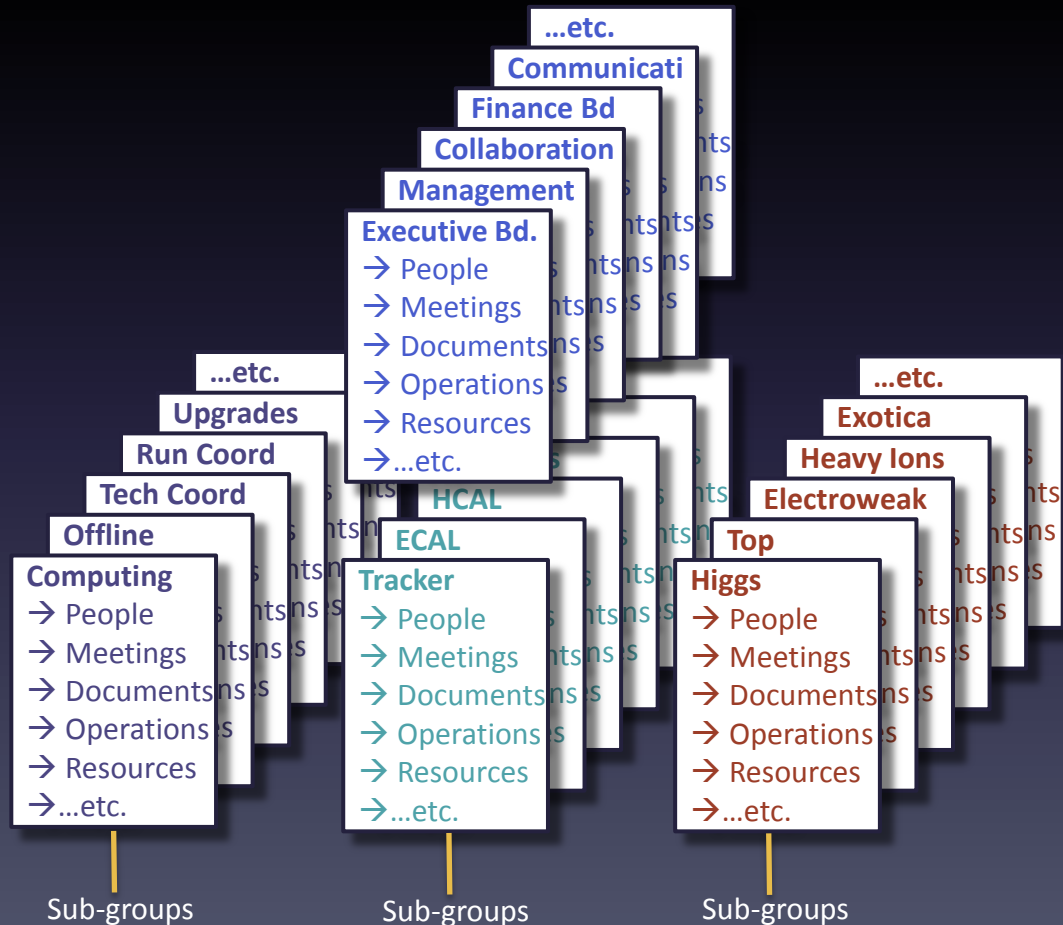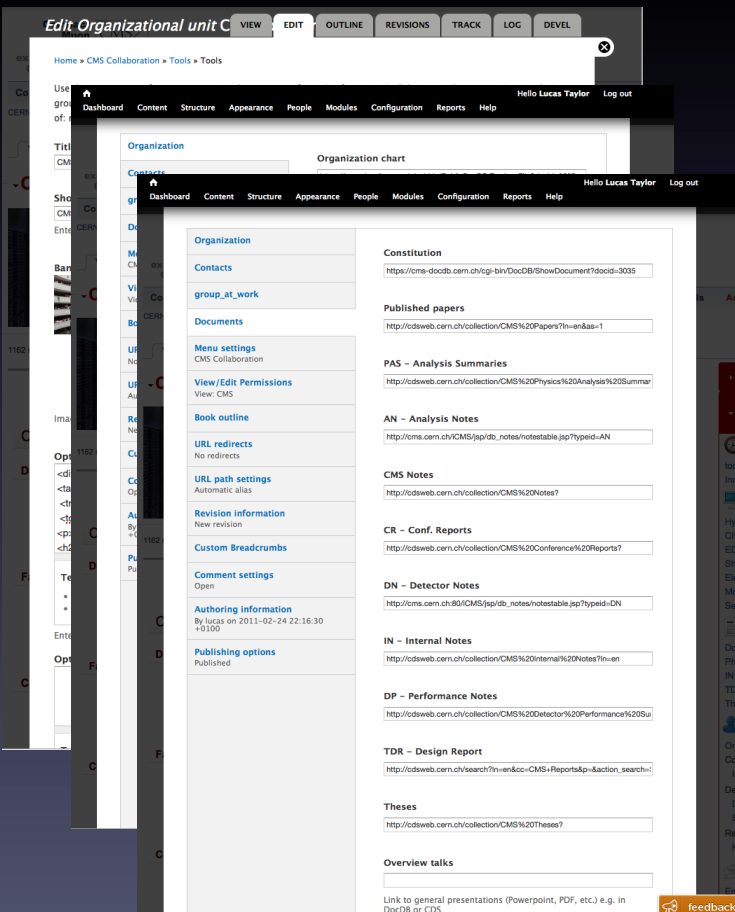Create a coherent web interface to all CMS information

Goal 2: Create a coherent web interface to all CMS information

- Use open-source Drupal Content Management System
  - Now a CERN service
  - Integrated with SSO, e-groups, DB and web services, etc.

- CMS layer with distinct parts
  - Style customizations
  - Data model
  - Navigation
  - Search

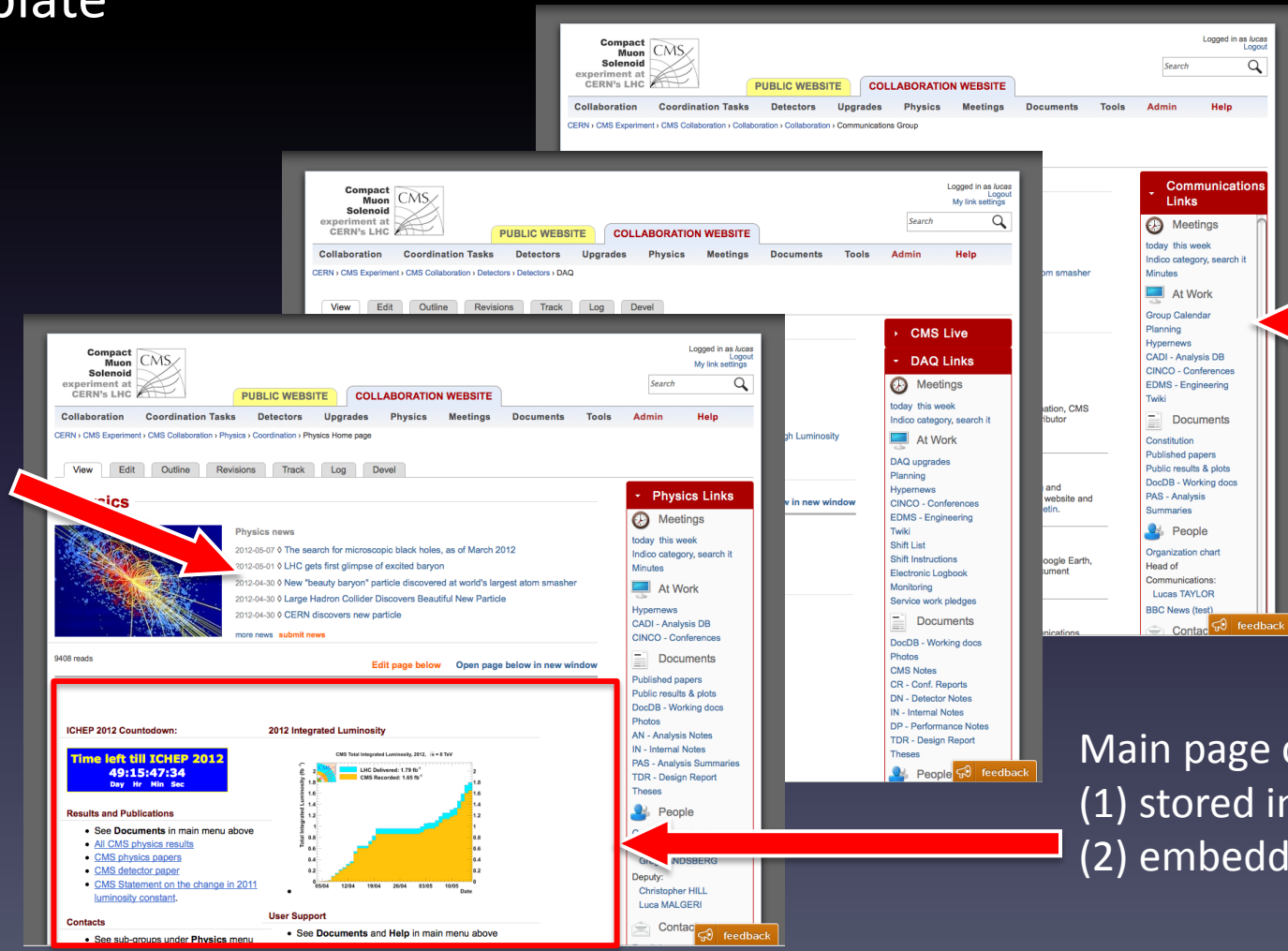# Goal 2: Create a coherent web interface to all CMS information

- Data model based on CMS "organisational units"
  - Each with similar set of data and links

...etc.

Communicati

Finance Bd

Collaboration

Management

Executive Bd.
→ People
→ Meetings
→ Documents
→ Operations
→ Resources
→ ...etc.

...etc.

Upgrades

Run Coord

Tech Coord

Offline

Computing
→ People
→ Meetings
→ Documents
→ Operations
→ Resources
→ ...etc.

HCAL

ECAL

Tracker
→ People
→ Meetings
→ Documents
→ Operations
→ Resources
→ ...etc.

...etc.

Exotica

Heavy Ions

Electroweak

Top

Higgs
→ People
→ Meetings
→ Documents
→ Operations
→ Resources
→ ...etc.

Sub-groups   Sub-groups   Sub-groups

Goal 2: Create a coherent web interface to all CMS information

- Build web pages for each organisational unit from a standard template

Sidebars show metadata specific to each org. unit

News, if tagged to be of interest to org. unit

Main page content is either
(1) stored in Drupal or
(2) embedded via an iframe

Goal 2: Create a coherent web interface to all CMS information

- Build site-wide menus from "organisational unit" structure

Goal 2: Create a coherent web interface to all CMS information

- Create different views from the same Drupal metadata

Goal 2: Create a coherent web interface to all CMS information

- Single CMS search bar

- Search done by underlying systems

- Results shown in tabs

- Access control by target systems

Goal 2: Create a coherent web interface to all CMS information

- Public site focus is fresh news
  - Text, images, links, etc.
  - Keywords, groups/topics
  - Author / editor workflow

- Email notifications, RSS, Twitter, FB, G+ …

- Promote to home page or features slider

- Push to selected CMS groups

Goal 2: Create a coherent web interface to all CMS information

- Interactive Google maps view of CMS

- Interactive view of CMS papers

- Embedded video and images

Migration to new CMS public site was smooth and rapid
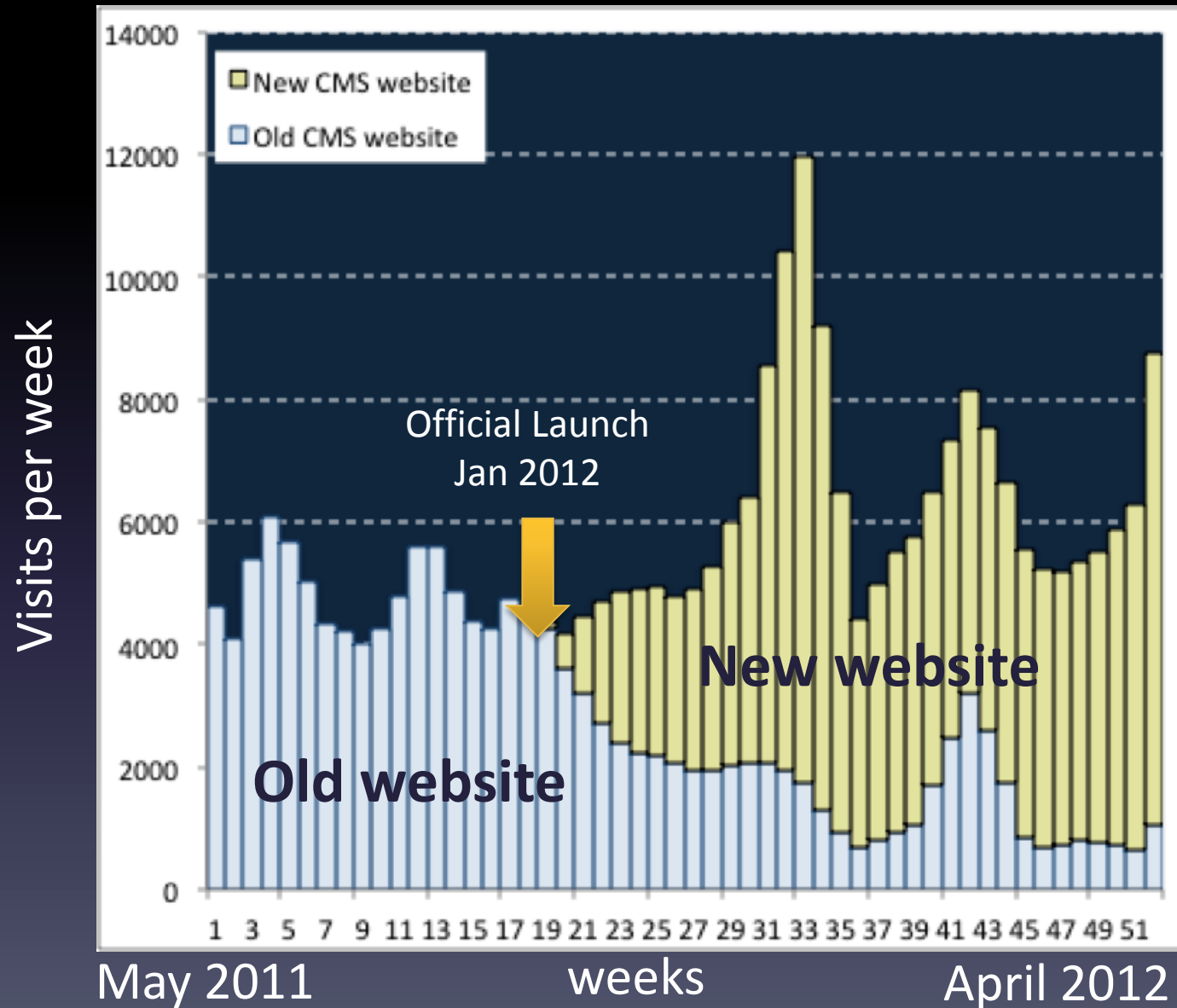
What next ?

- Indico feeds for meetings and Google calendars

- Live LHC and CMS operations displays

- Find and import about 180k existing documents

- Make more coherent interfaces to document systems

- Allow users to manage more of the content
  - Without compromising data model, style, and navigation

- Develop strategy to manage 144,000 Twiki pages & files

Do the benefits justify the costs?

1 FTE
CMS effort working to improve CMS information systems

Linux, html, css, php, js, persistence
Skills required

18 months
Time to achieve what was described in this talk

2 FTE
CMS-wide effort saved by reducing time looking for information by 1 min per person per day

1 FTE
Effort saved per 1% gain in CMS meeting efficiency (15,000 meetings, 50,000 docs per year)

1 M$
Rough cost of 1% of data-taking inefficiency (CMS ave. eff. is 91%)

Conclusions

- Goal 1: Manage all documents in a document database
  - Ensure that crucial CMS information is not lost
  - Users will help ... if easy-to-use tools are provided

- Goal 2: Create coherent web interface to all information
  - Use a Content Management System – Drupal is good enough
  - Keep the data model simple and the navigation coherent
  - To make an effective user transition, embed or link to external content if you have to

- Resources needed are modest relative to the benefits