Interoperability of FLOSS forges; lessons from the COCLICO project, implemented in FusionForge and other platforms

Olivier Berger <olivier.berger@telecom-sudparis.eu>,
Télécom SudParis
Christian Bayle <christian.bayle@orange.com>
Orange Labs

Thursday 11/10/2012
Open World Forum - Paris

Short bio

- Olivier BERGER
  <olivier.berger@telecom-sudparis.eu>
  Research Engineer at TELECOM SudParis, expert on software development forges, and interoperability in Libre Software development projects. Contributor to FusionForge, Debian, etc. Participated to COCLICO project (ended 10/2011).
- Christian BAYLE
  <christian.bayle@orange.com>
  Research engineer at Orange Labs, in Grenoble (France), in charge of deploying FLOSS software for developers, like Software Forges, Continuous integration as a service. Contributes to FusionForge and Tuleap forges, Debian maintainer, and participated in COCLICO project (FUI7) in the previous years.

FusionForge

https://fusionforge.org/

- Formerly known as GForge (Libre version)
  - inherited its codebase from the initial SourceForge (year 2000)
- Host your own forge (Linux, Postgres, PHP, etc.)
- FLOSS + Open Community (loose consensus)
- Used in many academic institutes (INRIA, SourceSup RENATER, IFREMER, ...), FLOSS projects (Debian's Alioth, ...)
- Boosted by project COCLICO in 2010-2011
- Christian and Olivier co-maintainers of the Debian packages
- FusionForge 5.2 released on 2012/09/28

Recap 2011

Last year at OWF 2011 (OSDCFr) :
Jailbreaking the Forges : project export/import efforts
Since then :

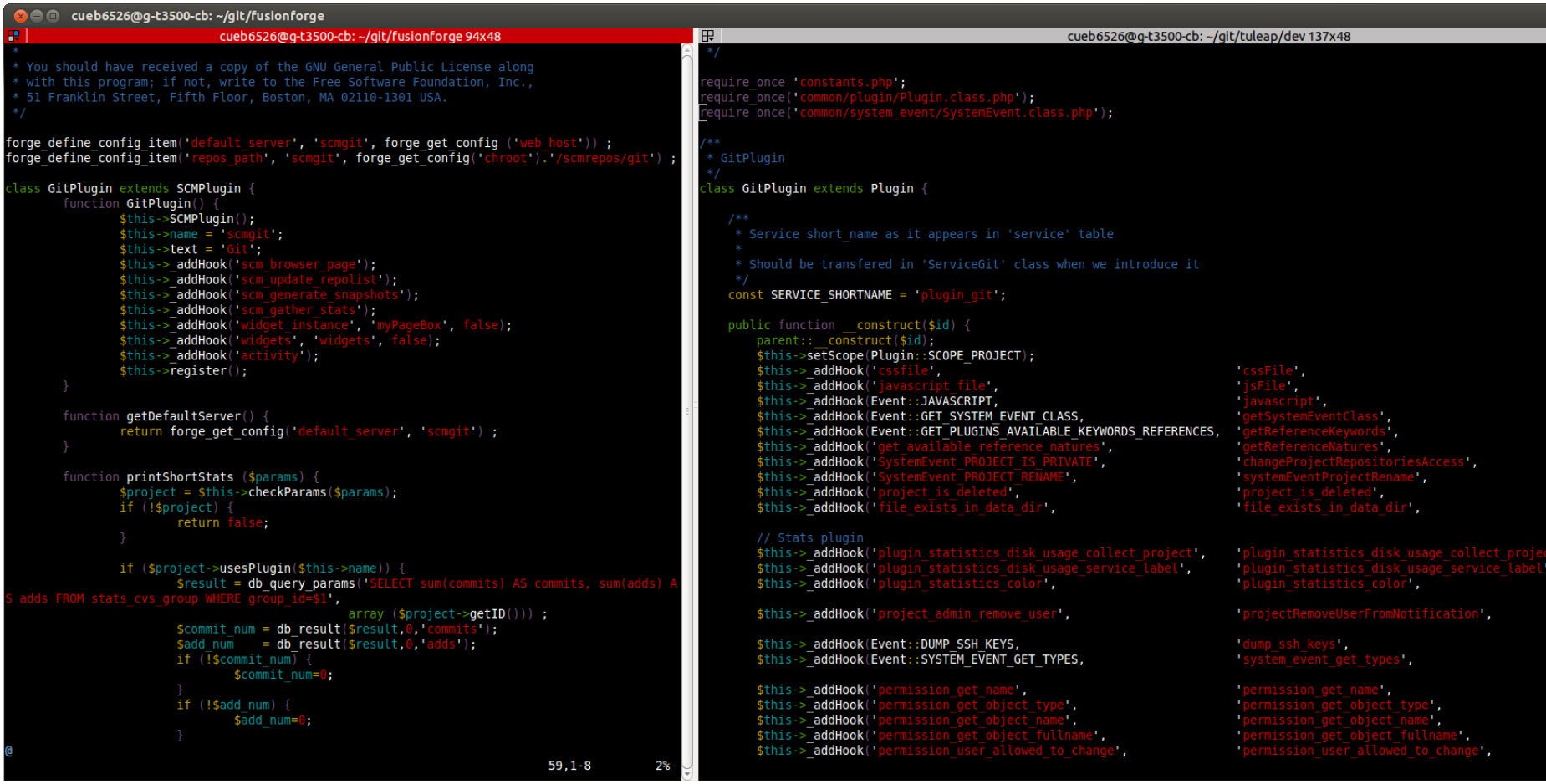
- COCLICO project finished
- Codendi -> Tuleap
- Some contributions to FusionForge in 5.2
- Some interest in OSLC
- ADMS.SW 1.0 specification released
- not much effort in ForgePlucker

Forge architecture
Integration strategies

Architectures & development models

- From scratch / comprehensive
  - Can you compete with GitHub or other proprietary tools ?
- Integration of other FLOSS projects
  - SSO, APIs, via Unix / Apache / PHP glue/hacks
- Services
  - Components / Web Services
- Web 2+ mashups
  - REST, Ajax
- Unhosted (.org) ?

Git SCM code


cueb6526@g-t3500-cb: ~/git/fusionforge
cueb6526@g-t3500-cb: ~/git/fusionforge 94x48
cueb6526@g-t3500-cb: ~/git/tuleap/dev 137x48
59,1-8 2%

Common plugins

- Common Sourceforge™ inheritance
  - Common Theme functions
  - Codendi™/Tuleap™ plugin system derivated from Gforge OSE/FusionForge
- They made it because they didn't know it was impossible
  - Common plugin were realized for Mailman/Forumml/Hudson/Mediawiki
  - Possible continuation with ADMS.SW, industrialisation of Mediawiki
- But hard to industrialize and keep synced
  - Leading to duplicated effort

Hosted, cloud. . .

Silos

Who owns your data/code

Can you escape ?

Data portability

- software is libre (well most of it ;-)
- (software development) data jailed in services (silos)
  - even worse at the Cloud scale ?
    - "Freedom in the Cloud" by Eben Moglen
    - "Franklin Street Statement on Freedom and Network Services"

(Semantic) Web Standards

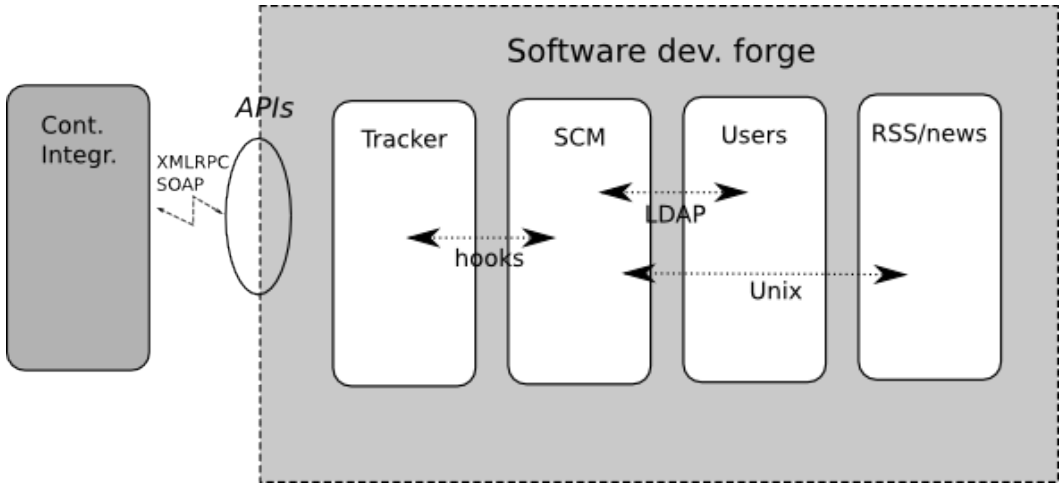
- REST (not yet so much of a standard... but *LDP* WG @ W3C)
- RDF (not necessarily XML)
- Linked (Open) Data

Some solutions

Software development tools integration issues

- Lots of different (legacy) systems in respective silos
- Integrated suites
  - Open Source tools or Proprietary Software alike
  - Software development *forges*
- Agile methodologies vs. monolithic suites
- Traceability
- Scalability
- Very poor interoperability
  - Syntactic
  - Semantic

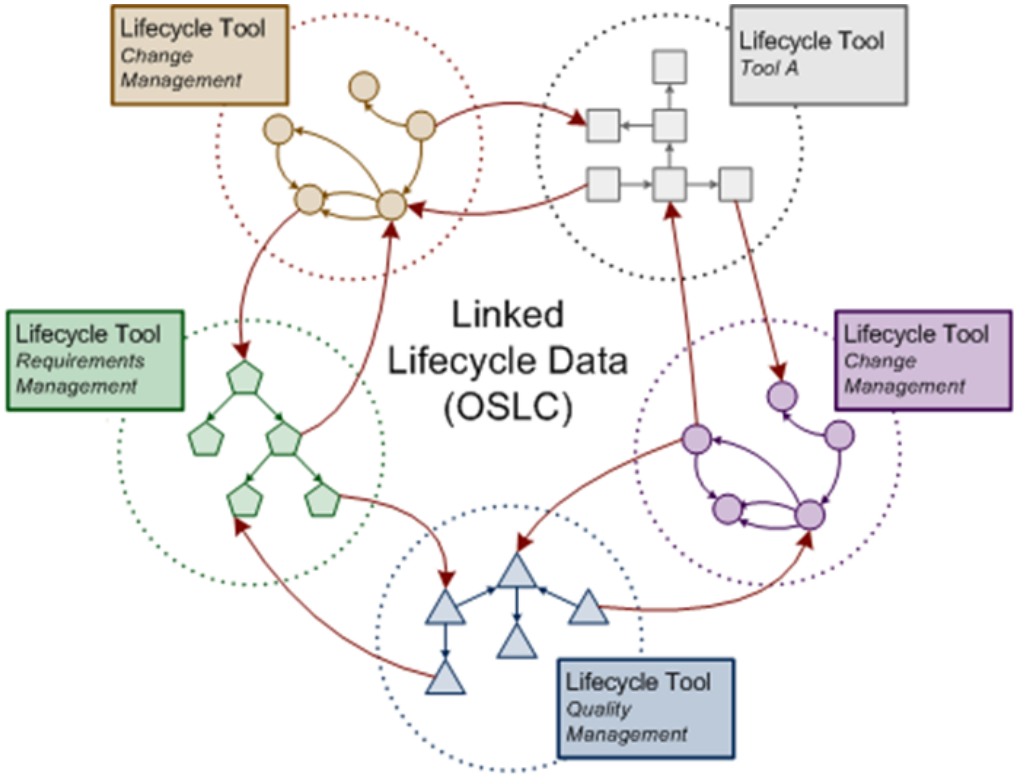
Integration with silos



- Tools are “natively” integrated in isolated suites
- Ad-hoc interfaces : no standardization

Linked lifecycle data

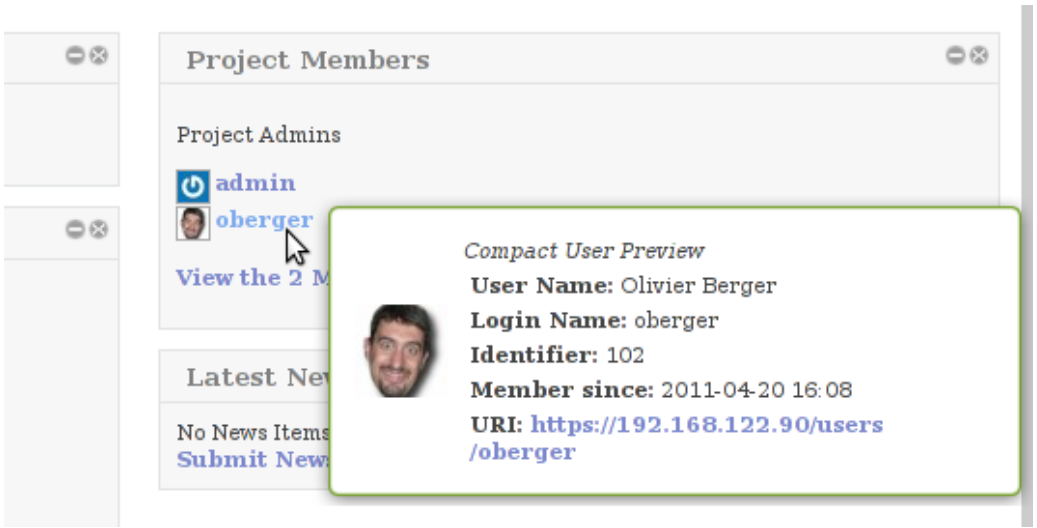
OSLC (Open Services for Lifecycle Collaboration) : http ://oslc.co/



Source : IBM/Rational

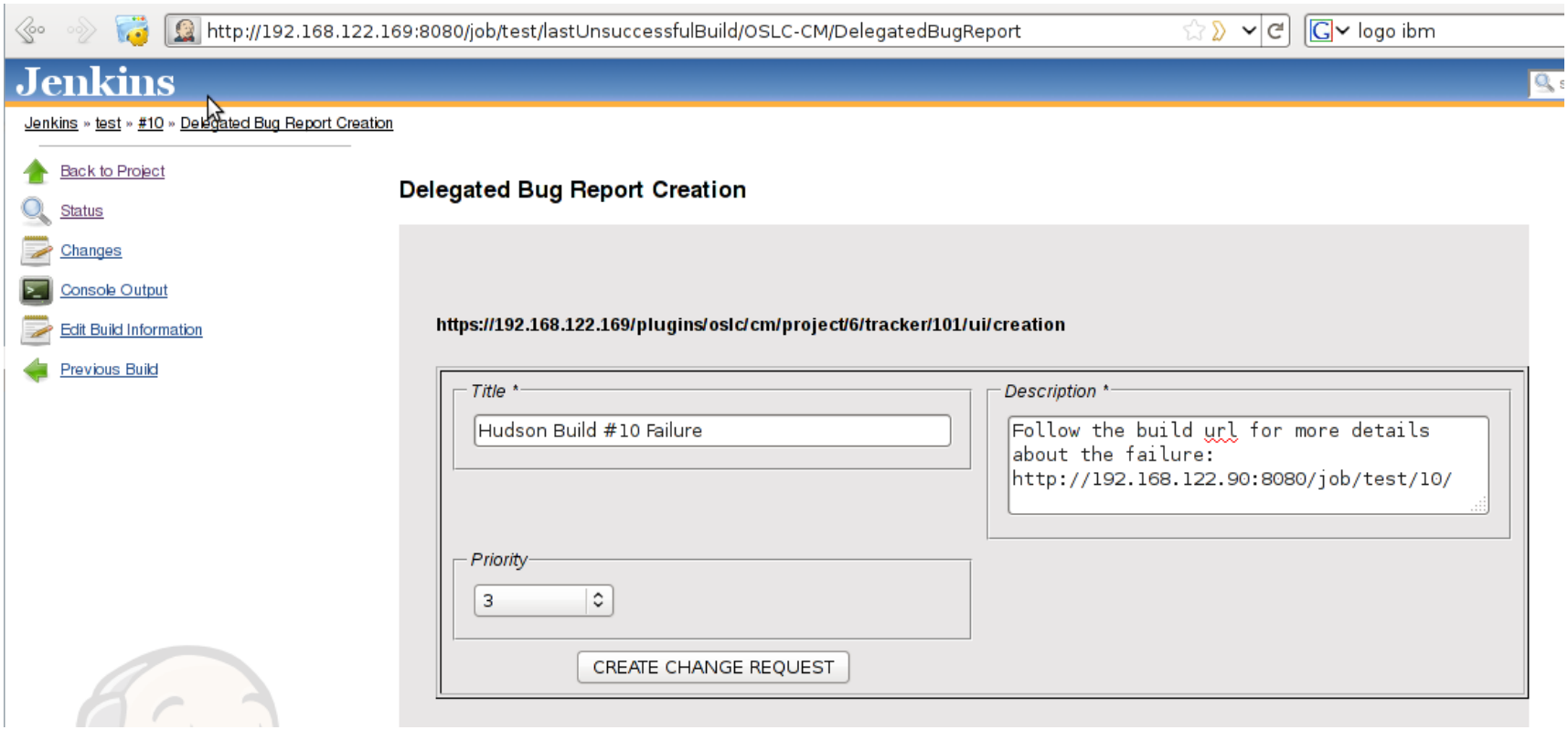
Example : OSLC *Compact preview* of FusionForge users

Resources linking and preview through standardized mechanism for URIs + AJAX preview



http://fusionforge.org/plugins/mediawiki/wiki/fusionforge/index.php/OslcCompactPreviewTooltips

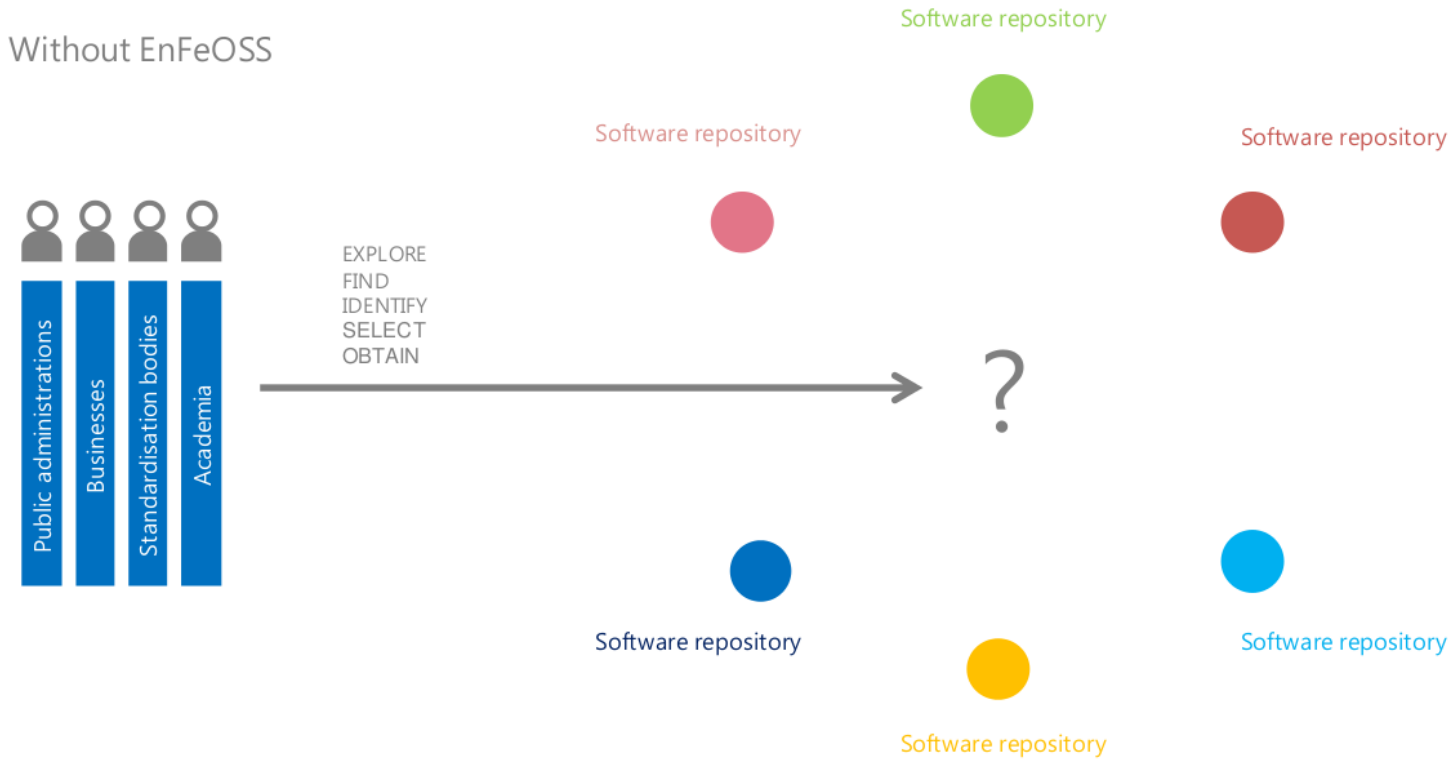
Continuous integration : FusionForge + JenkinsCI



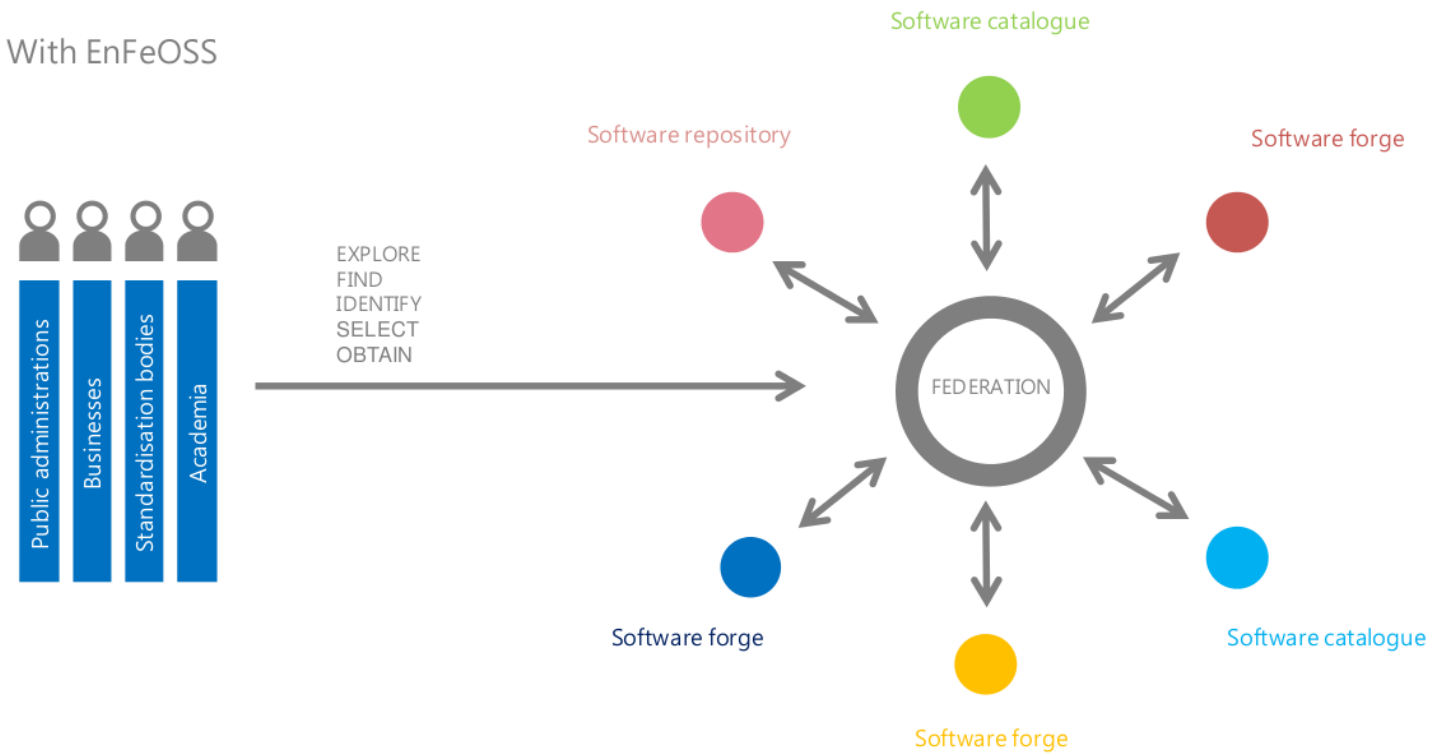
Demo : http://vimeo.com/23480321

Vision EC / ISA

Vision for an enhanced federation of software forges



Goal : Federations of forges / directories



Source : SEMIC.EU D5.1.1 – "Vision document : The vision for an enhanced software description metadata schema and federated software catalogue"

ADMS.SW

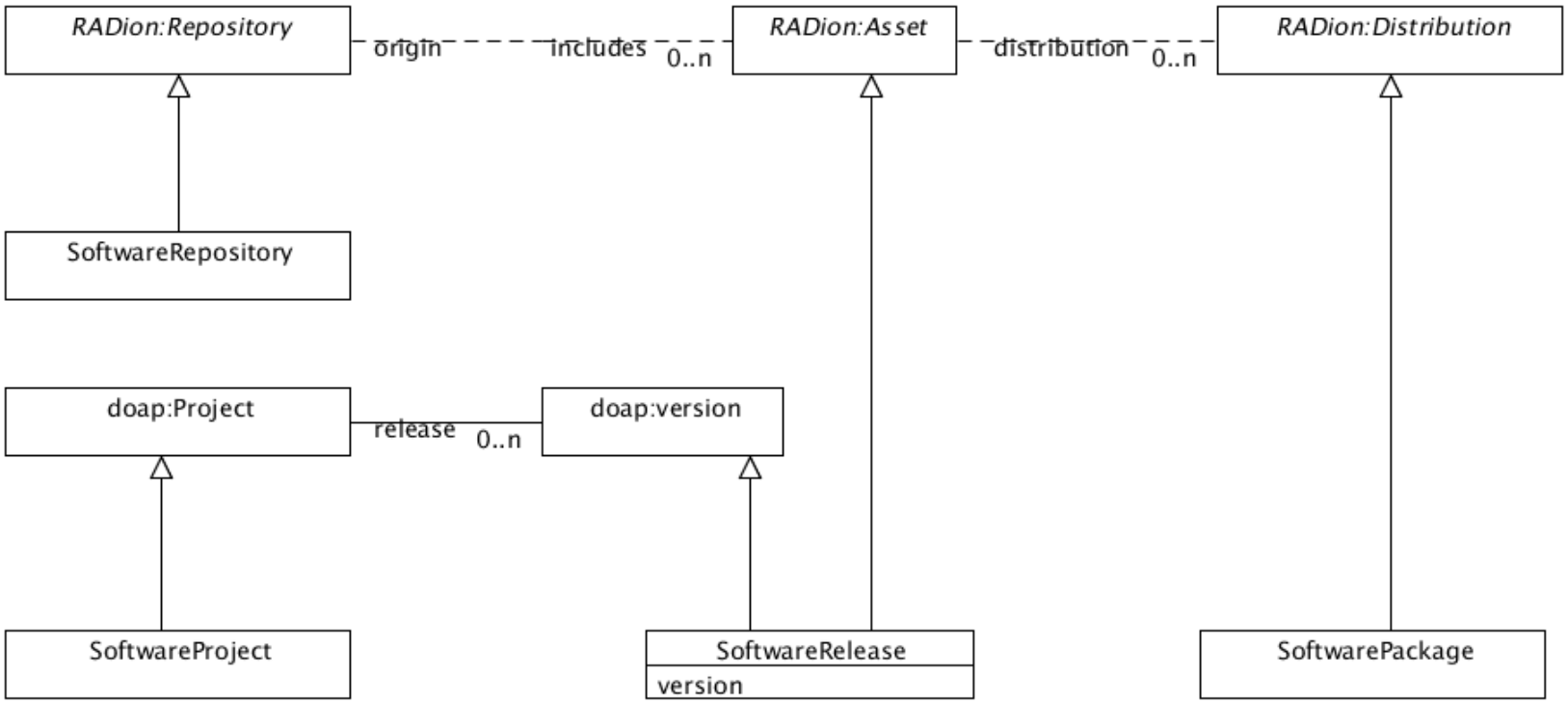
Asset Description Metadata Schema for Software (ADMS.SW)

- Pilot : EC / Interoperability Solutions for European Public Administrations (ISA) - cf. Joinup site
- Exchanging project / packages / releases descriptions across development platforms and directories

Specifications + RDF Schema

- Reuse :
  - ADMS / RADion (generic meta-data for semantic *assets* indexing)
  - DOAP (*Description of a project*)
  - SPDX™ (*Software Package Data Exchange* ®)
  - W3C Government Linked Data (GLD) Working Group
- version 1.0 issued 2012/06/29

ADMS.SW main concepts



Implementation

- FusionForge plugin for project meta-data (reuse of the DOAPRDF plugin developped in COCLICO)
- Debian PTS (Package Tracking System) : 1.5 M triples

```
PREFIX doap: <http://usefulinc.com/ns/doap>

SELECT * WHERE
{
  GRAPH <http://packages.qa.debian.org/>
  {
   ?dp doap:homepage ?h
  }
  GRAPH <http://projects.apache.org/>
  {
   ?ap doap:homepage ?h
  }
}
```

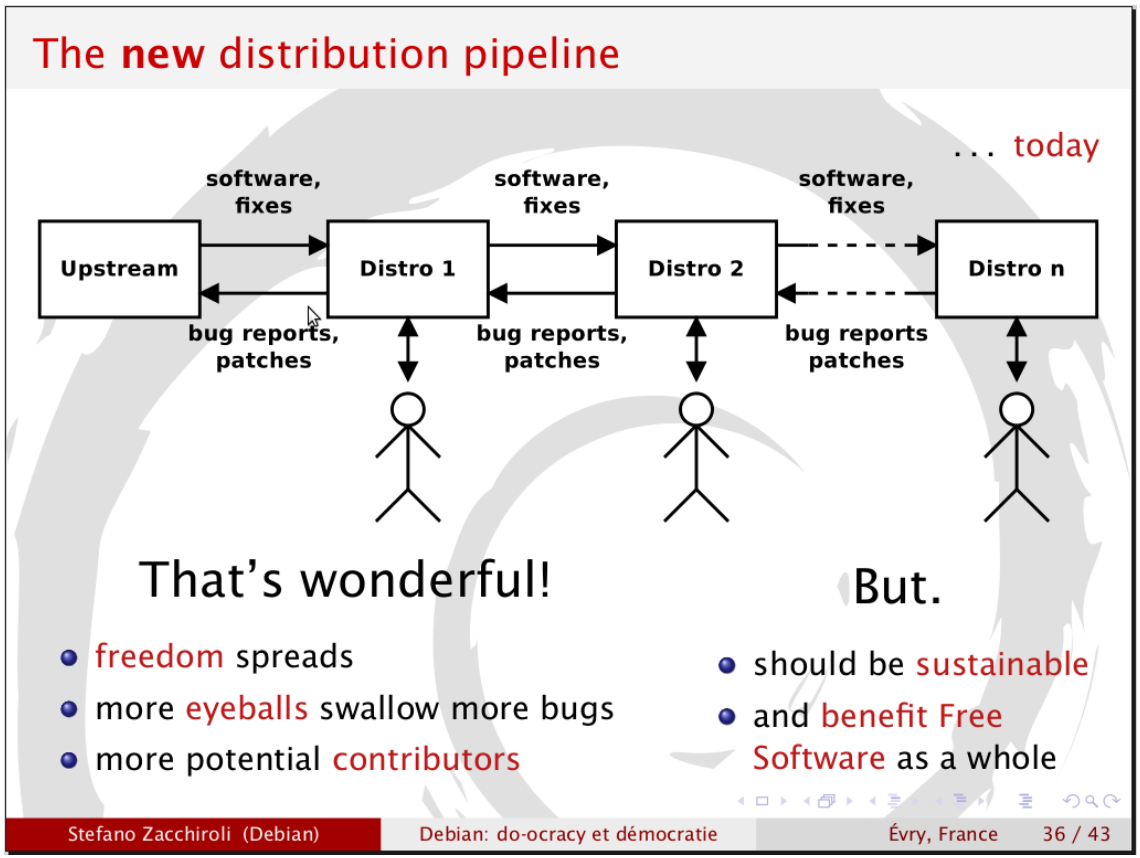
Matching packages

Example (Matching upstream project homepages with Debian source packages')

dp	h	ap
ivy	ant.a.o/ivy/	ant.a.o/ivy/
apr	apr.a.o/	apr.a.o/
apr-util	apr.a.o/	apr.a.o/
libcommons-cli-java	commons.a.o/cli/	commons.a.o/cli/
libcommons-codec-java	commons.a.o/codec/	commons.a.o/codec/
libcommons-collections3-java	commons.a.o/collections/	commons.a.o/collections/
libcommons-collections-java	commons.a.o/collections/	commons.a.o/collections/
commons-daemon	commons.a.o/daemon/	commons.a.o/daemon/
libcommons-discovery-java	commons.a.o/discovery/	commons.a.o/discovery/
libcommons-el-java	commons.a.o/el/	commons.a.o/el/
libcommons-fileupload-java	commons.a.o/fileupload/	commons.a.o/fileupload/
commons-io	commons.a.o/io/	commons.a.o/io/
commons-jci	commons.a.o/jci/	commons.a.o/jci/
libcommons-launcher-java	commons.a.o/launcher/	commons.a.o/launcher/
...	...	...

Perspectives
Conclusion

FLOSS distributions ecosystem



Source : http ://git.upsilon.cc/r/talks/20110224-evry.git

Large scale bug-tracking

- Lots of duplicate or related bug **reports**
- Not a single place where to monitor bugs
  - OK, *launchpad*, maybe... too much a silo anyway
- No interoperability of tools
- Manual work of maintainer / QA (bug triaging, etc.)

Bugtracker interoperability prerequisite

- No existing standard for bugtracker APIs
- . . . until *OSLC-CM* (2009-)

More interoperability in FLOSS tools

- Push standardization in tools development
  - Eclipse Mylyn support of OSLC (in the FLOSS parts), based on Eclipse Lyo roots ? . . . When ?
- New opportunities
  - Open Linked Data for direct "mining" of FLOSS projects development process metrics
- Large scale bug tracking and QA process in the FLOSS ecosystem
- *Qui vivra verra. . .*

Merci.
Questions ?