FOSSology - OSS for Open Source License Compliance

Anupam Ghosh (anupam.ghosh@siemens.com)

Overview: Contents

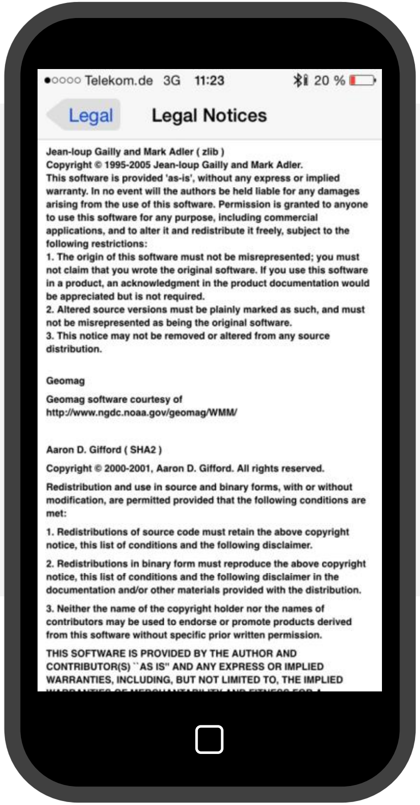
1. **Introduction FOSSology**
2. What is FOSSology
3. **Motivation**
4. What FOSSology needs
5. **Introduction Project Builder**
6. The ProjectBuilder Project
7. **Build Native Fossology Packages**
8. To get container running in the continuous build
9. **Conclusion**
10. Where to see it

The Problem Actually

You know these examples

Distributing open source software requires to

- Provide licenses of involved software
- Provide copyright statements of involved authors
- Provide disclaimers
- … and much more



Jean-loup Gailly and Mark Adler (zlib)
Copyright © 1995-2005 Jean-loup Gailly and Mark Adler.
This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:
1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Geomag

Geomag software courtesy of
http://www.ngdc.noaa.gov/geomag/WMM/

Aaron D. Gifford (SHA2)

Copyright © 2000-2001, Aaron D. Gifford. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

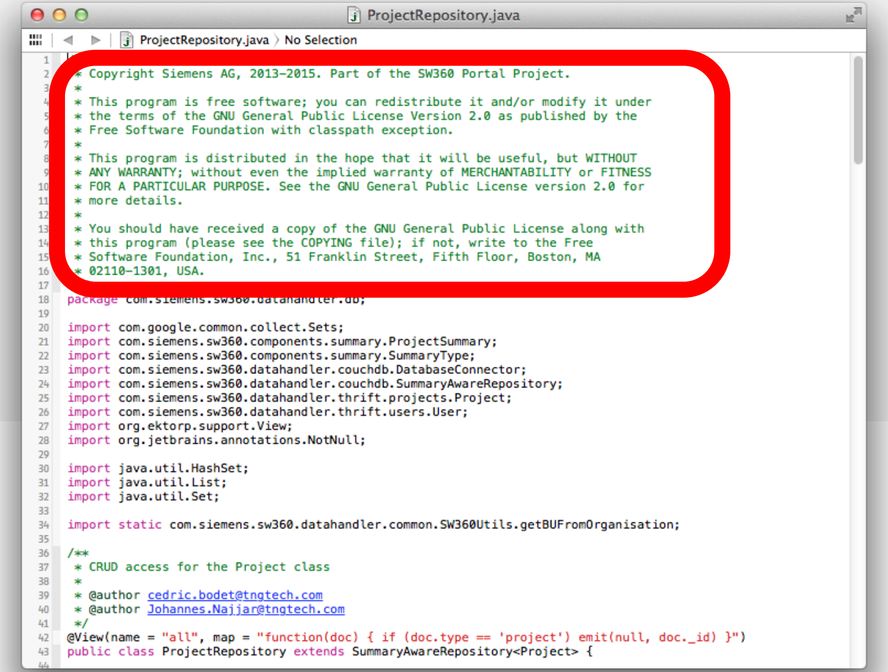
3. Neither the name of the copyright holder nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTOR(S) ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

It is about finding licenses

Finding Licenses

- License texts
- References to licenses
- Written texts explaining licensing
- License relevant statements

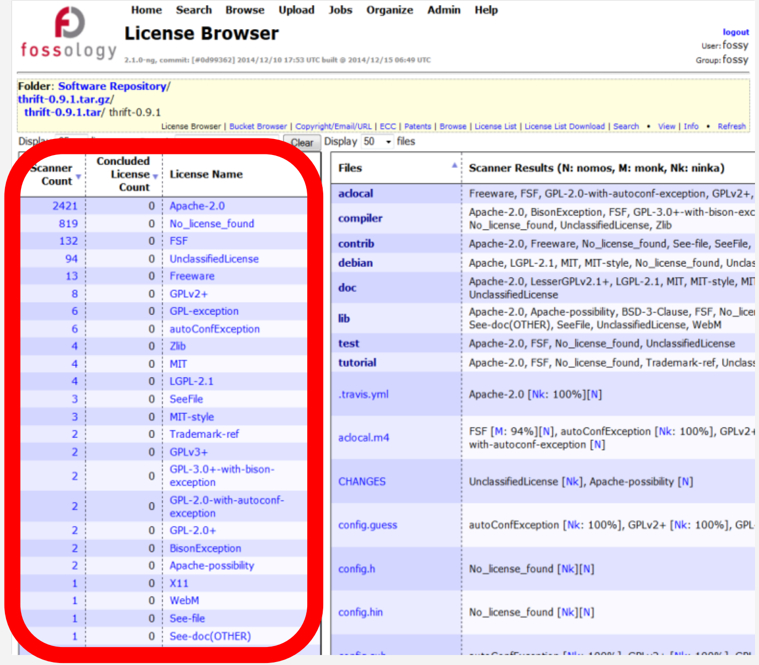


```
 * Copyright Siemens AG, 2013-2015. Part of the SW360 Portal Project.
 *
 * This program is free software; you can redistribute it and/or modify it under
 * the terms of the GNU General Public License Version 2.0 as published by the
 * Free Software Foundation with classpath exception.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
 * FOR A PARTICULAR PURPOSE. See the GNU General Public License version 2.0 for
 * more details.
 *
 * You should have received a copy of the GNU General Public License along with
 * this program (please see the COPYING file); if not, write to the Free
 * Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
 * 02110-1301, USA.

package com.siemens.sw360.datahandler.db;

import com.google.common.collect.Sets;
import com.siemens.sw360.components.summary.ProjectSummary;
import com.siemens.sw360.components.summary.SummaryType;
import com.siemens.sw360.datahandler.couchdb.DatabaseConnector;
import com.siemens.sw360.datahandler.couchdb.SummaryAwareRepository;
import com.siemens.sw360.datahandler.thrift.projects.Project;
import com.siemens.sw360.datahandler.thrift.users.User;
import org.ektorp.support.View;
import org.jetbrains.annotations.NotNull;

import java.util.HashSet;
import java.util.List;
import java.util.Set;

import static com.siemens.sw360.datahandler.common.SW360Utils.getBUFromOrganisation;

/**
 * CRUD access for the Project class
 *
 * @author cedric.bodet@tngtech.com
 * @author Johannes.Najjar@tngtech.com
 */
@View(name = "all", map = "function(doc) { if (doc.type == 'project') emit(null, doc._id) }")
public class ProjectRepository extends SummaryAwareRepository<Project> {
```

Problem of many Licenses (“Proliferation”)

Open Source and Reuse

- It is natural that an OSS project reuses available https://github.com/fossology/fossology
- Likely OSS from other projects is found
- For example, FOSSology will find 25 other licensing relevant text occurrences in Apache thrift


fossology
Home
Search
Browse
Upload
Jobs
Organize
Admin
Help
logout
License Browser | Bucket Browser | Copyright/Email/URL | ECC | Patents | Browse | License List | License List Download | Search • View | Info • Refresh
Clear
Display 50 files

License Browser

2.1.0-ng, commit: [#0d99362] 2014/12/10 17:53 UTC built @ 2014/12/15 06:49 UTC

User: fossy
Group: fossy

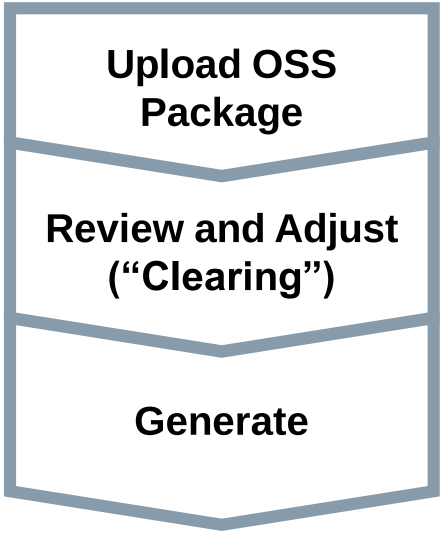
Folder: Software Repository/
thrift-0.9.1.tar.gz/
thrift-0.9.1.tar/ thrift-0.9.1

Scanner Count	Concluded License Count	License Name
2421	0	Apache-2.0
819	0	No_license_found
132	0	FSF
94	0	UnclassifiedLicense
13	0	Freeware
8	0	GPLv2+
6	0	GPL-exception
6	0	autoConfException
4	0	Zlib
4	0	MIT
4	0	LGPL-2.1
3	0	SeeFile
3	0	MIT-style
2	0	Trademark-ref
2	0	GPLv3+
2	0	GPL-3.0+-with-bison-exception
2	0	GPL-2.0-with-autoconf-exception
2	0	GPL-2.0+
2	0	BisonException
2	0	Apache-possibility
1	0	X11
1	0	WebM
1	0	See-file
1	0	See-doc(OTHER)

Files	Scanner Results (N: nomos, M: monk, Nk: ninka)
aclocal	Freeware, FSF, GPL-2.0-with-autoconf-exception, GPLv2+,
compiler	Apache-2.0, BisonException, FSF, GPL-3.0+-with-bison-exc No_license_found, UnclassifiedLicense, Zlib
contrib	Apache-2.0, Freeware, No_license_found, See-file, SeeFile,
debian	Apache, LGPL-2.1, MIT, MIT-style, No_license_found, Unclas
doc	Apache-2.0, LesserGPLv2.1+, LGPL-2.1, MIT, MIT-style, MI UnclassifiedLicense
lib	Apache-2.0, Apache-possibility, BSD-3-Clause, FSF, No_lice See-doc(OTHER), SeeFile, UnclassifiedLicense, WebM
test	Apache-2.0, FSF, No_license_found, UnclassifiedLicense
tutorial	Apache-2.0, FSF, No_license_found, Trademark-ref, Unclass
.travis.yml	Apache-2.0 [Nk: 100%][N]
aclocal.m4	FSF [M: 94%][N], autoConfException [Nk: 100%], GPLv2+ with-autoconf-exception [N]
CHANGES	UnclassifiedLicense [Nk], Apache-possibility [N]
config.guess	autoConfException [Nk: 100%], GPLv2+ [Nk: 100%], GPL
config.h	No_license_found [Nk][N]
config.hin	No_license_found [Nk][N]

How does FOSSology work?

See more details the Basic Workflow Description: https://www.fossology.org/get-started/basic-workflow



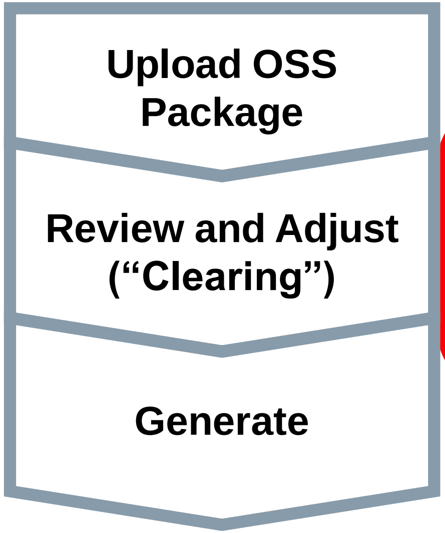
- Upload an open source package to the server
- Select scan agents that analyze the software

- Review what scanners have found
- Review license occurrences and correct findings if necessary

- Generate report output
- For example list of licenses or SPDX

What is the point of FOSSology?

See more details the Basic Workflow Description: https://www.fossology.org/get-started/basic-workflow


Upload OSS Package

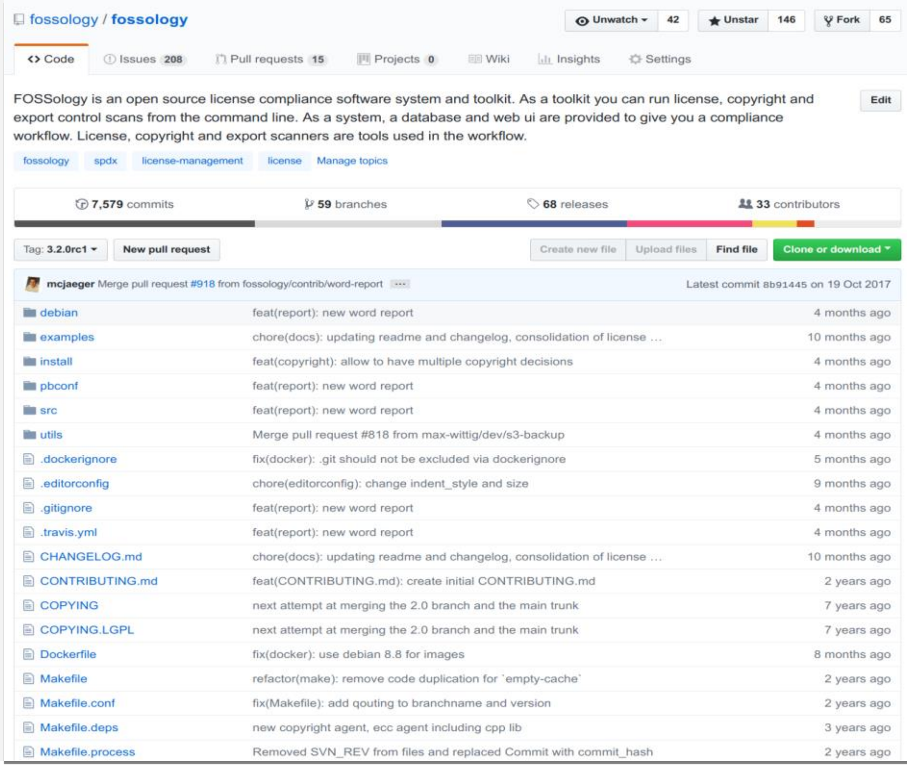
Review and Adjust (“Clearing”)

Generate

- Upload an open source package to the server
- Select scan agents that analyze the software
- Review what scanners have found
- Review license occurrences and correct findings if necessary
- Generate report output
- For example list of licenses or SPDX

Key facts

- Linux Foundation collaboration project
- GPLv2 licensed
- Linux, and only Linux application
- Mostly C/C++ and PHP
- Frontend runs on Apache httpd
- Provides also a Command Line Interface
- Backend schedules multiple Agents in parallel
- PostgreSQL as database
- Provides scripts for Docker and Vagrant



FOSSology is an open source license compliance software system and toolkit. As a toolkit you can run license, copyright and export control scans from the command line. As a system, a database and web ui are provided to give you a compliance workflow. License, copyright and export scanners are tools used in the workflow.

7,579 commits · 59 branches · 68 releases · 33 contributors

mcjaeger Merge pull request #918 from fossology/contrib/word-report … Latest commit 8b91445 on 19 Oct 2017

debian	feat(report): new word report	4 months ago
examples	chore(docs): updating readme and changelog, consolidation of license …	10 months ago
install	feat(copyright): allow to have multiple copyright decisions	4 months ago
pbconf	feat(report): new word report	4 months ago
src	feat(report): new word report	4 months ago
utils	Merge pull request #818 from max-wittig/dev/s3-backup	4 months ago
.dockerignore	fix(docker): .git should not be excluded via dockerignore	5 months ago
.editorconfig	chore(editorconfig): change indent_style and size	9 months ago
.gitignore	feat(report): new word report	4 months ago
.travis.yml	feat(report): new word report	4 months ago
CHANGELOG.md	chore(docs): updating readme and changelog, consolidation of license …	10 months ago
CONTRIBUTING.md	feat(CONTRIBUTING.md): create initial CONTRIBUTING.md	2 years ago
COPYING	next attempt at merging the 2.0 branch and the main trunk	7 years ago
COPYING.LGPL	next attempt at merging the 2.0 branch and the main trunk	7 years ago
Dockerfile	fix(docker): use debian 8.8 for images	8 months ago
Makefile	refactor(make): remove code duplication for `empty-cache`	2 years ago
Makefile.conf	fix(Makefile): add qouting to branchname and version	2 years ago
Makefile.deps	new copyright agent, ecc agent including cpp lib	3 years ago
Makefile.process	Removed SVN_REV from files and replaced Commit with commit_hash	2 years ago

New Features

Feature: SPDX Import

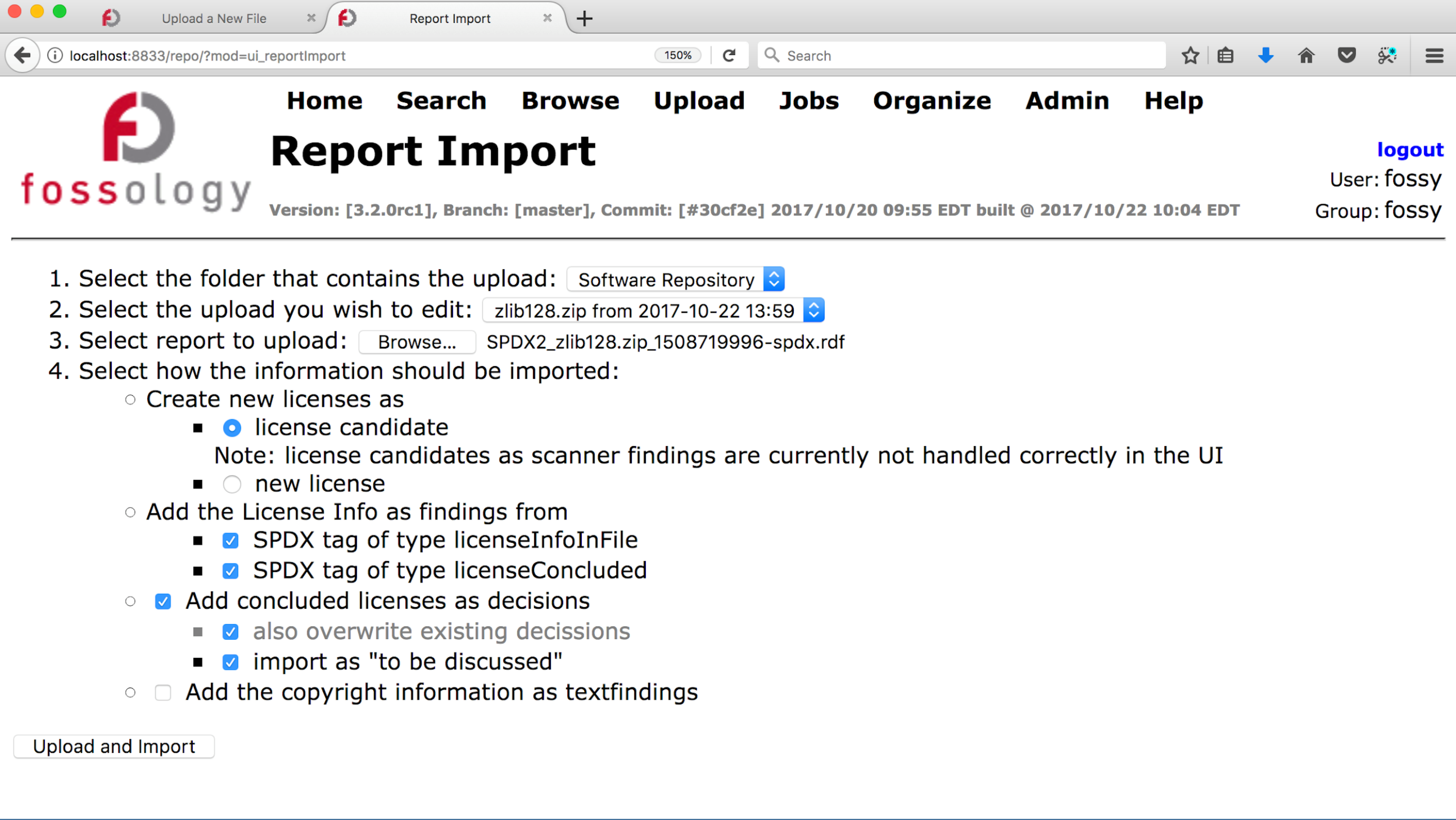
SPDX Import allows for applying SPDX license analysis information to uploaded source code packages

Use Case

- Licensing information in SPDX files require also to see the original source code
- If you receive an SPDX file from another (unknown) organisation, review is maybe necessary
- **How can I review SPDX license information from other source?**

Solution

- FOSSology allows for uploading SPDX files
- Select “Report import” from the “Upload menu”
- Select different options for importing
- Actually it is a little bit difficult
  - How to deal with found in file and concluded licenses?
  - How to deal with new licenses -> create candidates
  - How to deal with conclusions -> take over or stage them?



Report Import

Version: [3.2.0rc1], Branch: [master], Commit: [#30cf2e] 2017/10/20 09:55 EDT built @ 2017/10/22 10:04 EDT

User: fossy
Group: fossy

1. Select the folder that contains the upload: Software Repository
2. Select the upload you wish to edit: zlib128.zip from 2017-10-22 13:59
3. Select report to upload: Browse... SPDX2_zlib128.zip_1508719996-spdx.rdf
4. Select how the information should be imported:
   - Create new licenses as
     - ● license candidate
       Note: license candidates as scanner findings are currently not handled correctly in the UI
     - ○ new license
   - Add the License Info as findings from
     - ☑ SPDX tag of type licenseInfoInFile
     - ☑ SPDX tag of type licenseConcluded
   - ☑ Add concluded licenses as decisions
     - ☑ also overwrite existing decissions
     - ☑ import as "to be discussed"
   - □ Add the copyright information as textfindings

Upload and Import

(Note: Importing SPDX and reuse of licensing analysis information)

Feature: Analysis Documentation

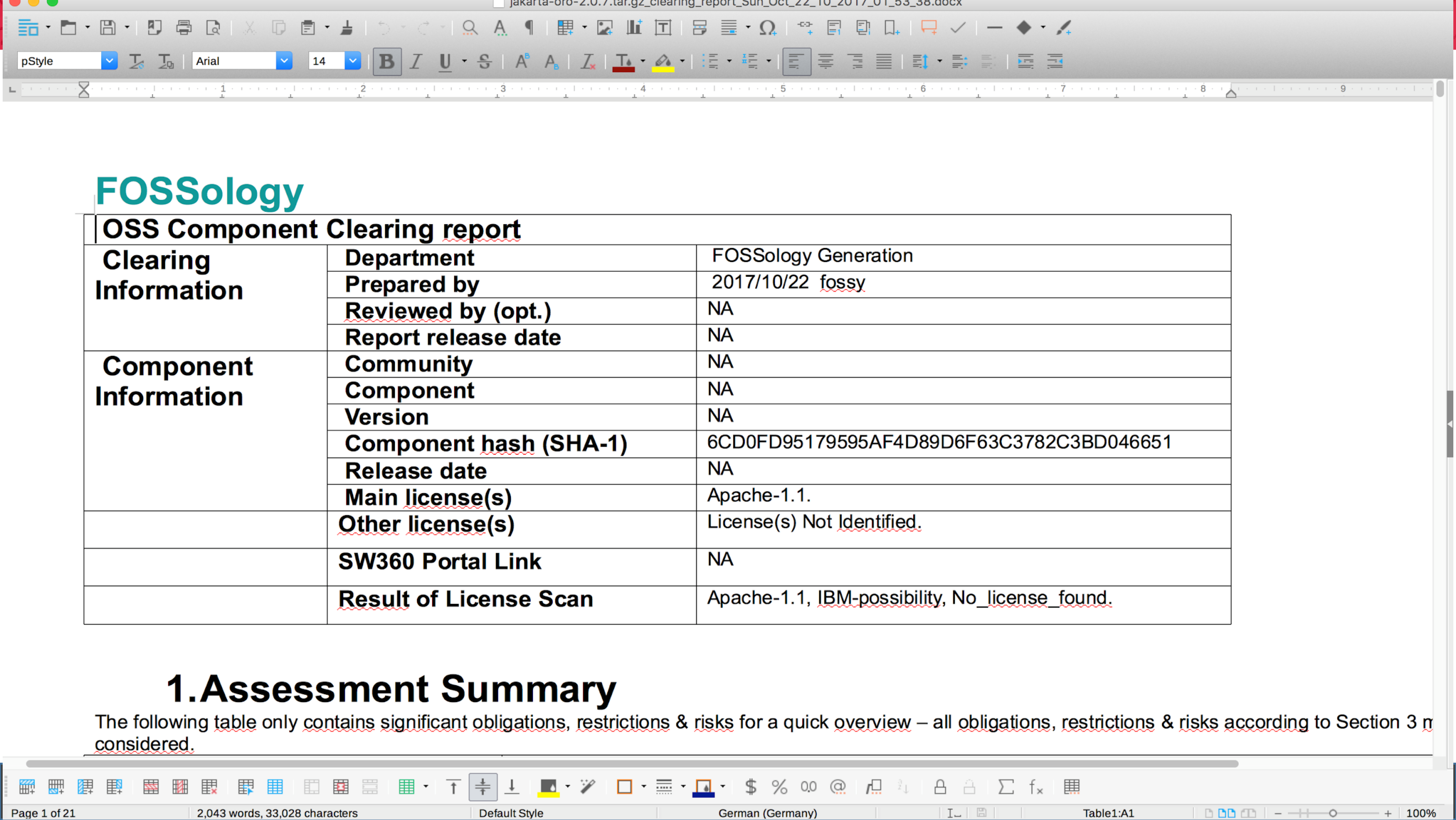
SPDX Import allows for applying SPDX license analysis information to uploaded source code packages

Use Case

- Exchanging licensing documentation with SPDX is fine, but ...
- Can I have documentation of my analysis?
- Can I provide comprehensive reporting what was analysed?
- **How can tell others what needs to be done?**

Solution

- Now generate a report
- Same as with SPDX output
- Contains rich set of elements
- License listings, copyright listings, ECC listing, Bulk phrase listing, ignored files listing, remarks listing
  - Trying to summarise all information out of FOSSology for a component


jakarta-oro-2.0.7.tar.gz_clearing_report_Sun_Oct_22_10_2017_01_53_38.docx
pStyle
Arial
14

FOSSology

OSS Component Clearing report		
Clearing Information	Department	FOSSology Generation
	Prepared by	2017/10/22 fossy
	Reviewed by (opt.)	NA
	Report release date	NA
Component Information	Community	NA
	Component	NA
	Version	NA
	Component hash (SHA-1)	6CD0FD95179595AF4D89D6F63C3782C3BD046651
	Release date	NA
	Main license(s)	Apache-1.1.
	Other license(s)	License(s) Not Identified.
	SW360 Portal Link	NA
	Result of License Scan	Apache-1.1, IBM-possibility, No_license_found.

1.Assessment Summary

The following table only contains significant obligations, restrictions & risks for a quick overview – all obligations, restrictions & risks according to Section 3 m considered.

Feature: Obligations / Policies Management

SPDX Import allows for applying SPDX license analysis information to uploaded source code packages

Use Case

- List of licenses is good, but ...
- … who understands what to do in your organisation?
- Obligations / Policies explain what to do , but how to get them in?
- **How can I have the involved obligations with the licenses?**

Solution

- Similar to licenses: obligations or policies
- Managed in the admin section
- But it is not so simple:
  - How to deal with candidate licenses?
  - How to deal with redundant obligations?
  - How to update the database?

License Scanning or Source Code Scanning?

Why Or?

License Scanning and Source Code Scanning

SPDX Import allows for applying SPDX license analysis information to uploaded source code packages

Use Case

- Searching for licensing information by license statements does not find the source code origin.
- Was source code copied?
- **How can I find the origin of Source code?**

Solution

- Well, there was no solution
- FOSSology can search for licensing statements, but it does not have a database for source code.
- A source code database would be needed to match source code and determine its origin (and more)
- **But FOSSology has a plug in architecture ...**

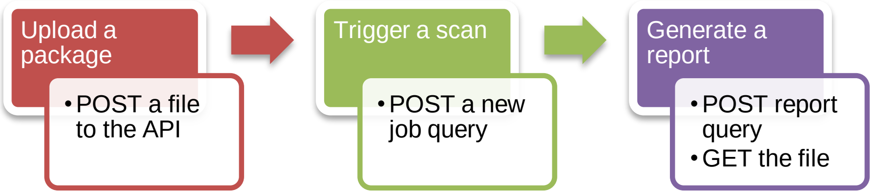
Fossology REST API

Feature: REST API – Basic functionality

After the release of version 3.4.0, the project has added a REST API to FOSSology

Current Endpoints :

- Uploads
- Folders
- Search
- Users
- Jobs
- Reports
- Tokens



Upload a package

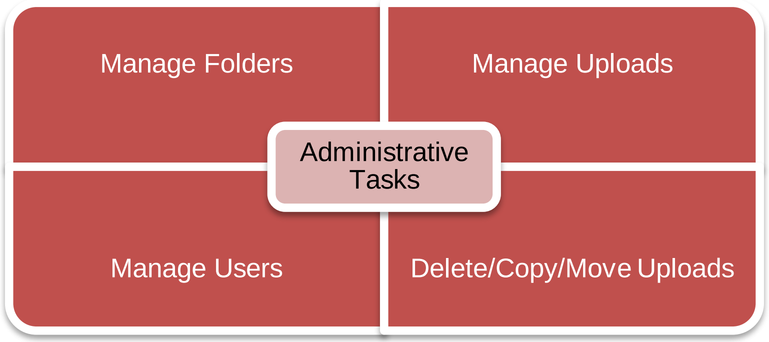
- POST a file to the API

Trigger a scan

- POST a new job query

Generate a report

- POST report query
- GET the file



Manage Folders

Manage Uploads

Administrative Tasks

Manage Users

Delete/Copy/Move Uploads

https://www.fossology.org/get-started/basic-rest-api-calls/

Atarashi

A Step towards non-rule based standalone command line scanner…

(https://github.com/fossology/atarashi)

Conclusion

Summary

1. **FOSSology for precise license analysis**
2. FOSSology is a mature framework and Web application for license analysis
3. **SPDX Import**
4. Finally: review of SPDX documents and … reuse of licensing info at new versions
5. **New Document Report**
6. Beyond exchange of license information: Complete documentation of analysis
7. **Obligations / Policies handling**
8. Organise obligations with the found licenses.
9. **Rest API**
10. To get container running in the continuous build

Thank you very much - … some links:

- https://github.com/fossology/fossology
- https://www.fossology.org/

Try it Yourself:

```
$ docker run -p 8081:80 fossology/fossology
```