Team: Ninka and Fosssology

Members: Hai Ninh Nguyen, Ryan Vanek, Jordan Rauscher

Milestone 2

Ninka and Fossology

1. **System Charter**
   - 1.1. **Problems/Opportunities**: None of the open source license scanning tools are perfect. Each scanning tool has strengths and weaknesses with regard to the types of licenses they detect. Combining the output of multiple license scanners could be useful in providing more reliability to automated license explication.
   - 1.2. **System Overview**: Our system will process a package or file with FOSSology and Ninka (two license scanning tools) to determine the open source licenses associated with the file or package. The outputs of these two programs will be merged into one meaningful SPDX document.
   - 1.3. **Objectives**
     - 1.3.1. Enhance the open source software license identification process
     - 1.3.2. Create SPDX documents that can be used in other processes
     - 1.3.3. Provide more choices for open source compliance organizations to identify licenses

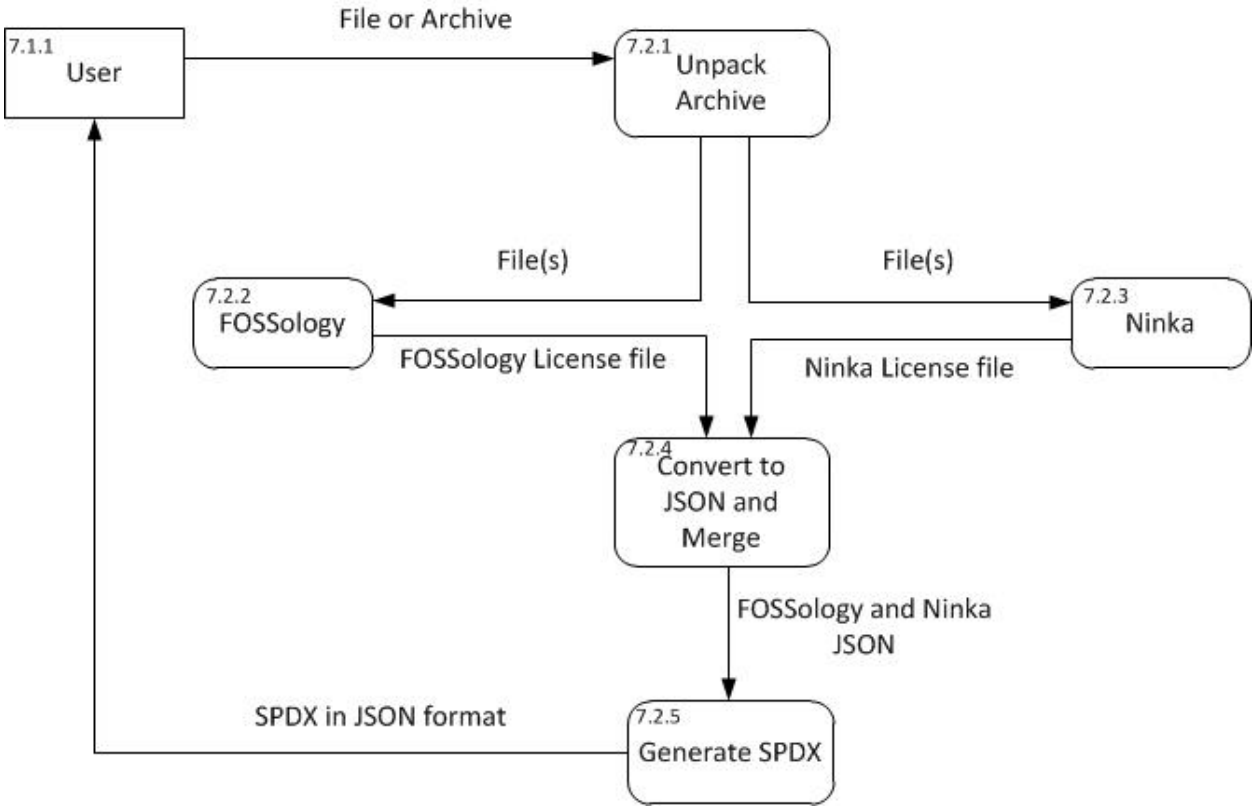
2. **System Service Request**
   - 2.1. A machine with Ubuntu 12.04 LTS operating system.
     - 2.1.1. Perl version 5.14.2
     - 2.1.2. PHP version 5.3.10-1ubuntu3.9
     - 2.1.3. Perl JSON module version 2.53 (http://search.cpan.org/~makamaka/JSON-2.53/lib/JSON.pm)
     - 2.1.4. mailutils, sharutils, ssmtp (for the mail server)
   - 2.2. Apache Web Server 2.4 for the Graphical User Interface to the system.
   - 2.3. FOSSology and Ninka must be downloaded and installed to run the tool. In addition, FOSSology requires PHP (prefer 5.6 version) and PostgreSQL (prefer 9.1 version)
   - 2.4. Recommended minimum hardware: 2.5GHz processor, 4GB RAM, 500GB HDD.

3. **Stakeholders/Community**
   - 3.1. Kate Stewart (SPDX)
   - 3.2. Bob Gobeille (HP) - a manager and developer of FOSSology
   - 3.3. The FOSSology community
   - 3.4. Daniel German (Ninka)
   - 3.5. Liang Cao: database owner and manager
   - 3.6. 4900 Class: YOCTO+SPDX, Dashboard, the other Ninka/FOSSology group

4. **Communication Management Plan**
   - 4.1. The team meets inside of class on Monday and Wednesday from 1:30 PM to 2:45 PM at the PKI building at University of Nebraska at Omaha. The team will also meet outside of class every Friday at 1:00 PM at the PKI building.
   - 4.2. If any communication is necessary with the SPDX or FOSSology communities, we will use their respective mailing lists. We are currently subscribing to FOSSology mailing list. The team will email the FOSSology community for any questions regarding the FOSSology scanning tool.

5. **Distribution System**
   - 5.1. Github code repository.
   - 5.2. The Fossology and Ninka repositories contain code that the team will use, but we will not be contributing any code to these repositories.

6. **Code Contribution Management**
   - 6.1. Up to May 2014, all contributions will be made through one of the three team members (Ryan Vanek (vanek909@gmail.com), Jordan Rauscher (jtrtoday@gmail.com), Hai Nguyen (hnguyen@unomaha.edu). Anybody outside the team who wishes to contribute may e-mail one of us. Final decisions will be made by simple majority among the group.
   - 6.2. As of May 2014, we will hand over the repository to Matt Germonprez or any party who is interested in taking over control of the project. At this point, we will no longer manage any new contributions to the project.



7. DataFlow Diagram

7.1. Entities

7.1.1. User

Description: The end-user of the system. This could be a human or another outside process.

7.2. Processes

7.2.1. Unpack Archive

Description: Sends the file to Ninka and FOSSology. If the file is an archive, unpacks the archive and sends the files to Ninka and FOSSology to be scanned.

Decomposition:

```
IF file is an archive
THEN
{
        Unpack archive
        Scan files with Ninka and FOSSology
}
ELSE
        Scan file with Ninka and FOSSology
```

7.2.2. **Ninka [third-party software]**

Description: Ninka is an open source license scanning tool

Decomposition:

Scans the file(s) to determine which, if any, licenses it contains. Sends a license file containing one line for each file and its license(s) to the “Convert to JSON” process.

7.2.3. **FOSSology [third-party software]**

Description: FOSSology is an open source license scanning tool

Decomposition:

Scans the file(s) to determine which, if any, licenses it contains. Sends a license file containing one line for each file and its license(s) to the “Convert to JSON” process.

7.2.4. **Convert to JSON and Merge**

Description: Converts the license files to JSON and merges the result into one JSON file.

Decomposition

```
FOR each license in both license files
DO
{
        Convert file/license pairs to JSON
        Merge JSON pairs (ignore license conflicts)
        Add to output
}
ELSE
        Send output to Generate SPDX (7.2.5)
```

7.2.5. **Generate SPDX**

Description: Generates the final SPDX (in JSON format) from the merged license file.

Decomposition:

Generate final SPDX JSON file from merged license JSON

8. **Data Base / Data Store Structures of the System**

8.1. There is no database used directly by our system. Our output will formatted in such a way that it will be easily inserted into the “SPDX Tools” database.

8.2. This is the shema of the “SPDX Tools” database:

	standard_license_header	varchar(255)	Not Null
	license_cross_reference	varchar(255)	Not Null
	created_at	datetime	Not Null
	updated_at	datetime	
doc_license_associations	id	int(11)	Auto Increment, Not Null, unique
	spdx_doc_id	int(11)	Not Null, FK(spdx_docs(id))
	license_id	int(11)	Not Null, FK(licenses(id))
	license_identifer	varchar(255)	Not Null
	license_name	varchar(255)	Not Null
	license_comment	text	Not Null
	created_at	datetime	Not Null
	updated_at	datetime	
licensings	id	int(11)	Auto Increment, Not Null, unique
	package_file_id	int(11)	Not Null,FK(packages_files(id))
	juncture	varchar(255)	Not Null
	doc_license_association_id	int(11)	Not Null,FK(doc_license_associations(id))
	created_at	datetime	Not Null
	updated_at	datetime	
doc_file_package_associations	id	int(11)	Auto Increment, Not Null, unique
	spdx_doc_id	int(11)	Not Null,FK(spdx_docs(id))
	package_id	int(11)	Not Null,FK(packages(id))
	package_file_id	int(11)	Not Null,FK(package_files(id))
	created_at	datetime	Not Null
	updated_at	datetime	
creators	id	int(11)	Auto Increment, Not Null, unique
	generated_at	datetime	Not Null
	creator_comments	text	
	license_list_version	varchar(255)	Not Null
	spdx_doc_id	int(11)	Not Null,FK(spdx_docs(id))
	creator	varchar(255)	Not Null
	created_at	datetime	Not Null
	updated_at	datetime	
reviewers	id	int(11)	Auto Increment, Not Null, unique
	reviewer_date	datetime	Not Null
	reviewer_comment	text	
	spdx_doc_id	int(11)	Not Null,FK(spdx_docs(id))
	reviewer	varchar(255)	Not Null
	created_at	datetime	Not Null
	updated_at	datetime	
products	id	int(11)	Auto Increment, Not Null, unique
	product_name	varchar(255)	Not Null
	product_type	varchar(255)	Not Null
	product_description	text	
	created_at	datetime	Not Null
	updated_at	datetime	
	parent_product_id	int(11)	FK(products(id))
product_software	software_id	int(11)	Not Null,FK(software(id))
	product_id	int(11)	Not Null,FK(products(id))
	package_id	int(11)	Not Null,FK(packages(id))
software	id	int(11)(11l)	Auto Increment, Not Null, unique
	software_name	varchar(255)	Not Null
	software_version	varchar(255)	Not Null
	software_description	text	
	created_at	datetime	Not Null
	updated_at	datetime	

9. **Copyright Declarations and License Choice**
   We are using the CC by-SA-3.0 license (https://creativecommons.org/licenses/by-sa/3.0/us/). Our software may be used or modified in whole or in part, including for commercial purposes, as long as the we are given credit as the original creators and it is indicated if changes are made. The resulting products must be released under the same license.

10. **Change Log**

Change Log

Version	Revision Date	Revision Comment
1.0	01/24/14	Creation of Change Log
1.1	01/31/14	Updating of Change Log
2.0	02/17/14	Updating of Change Log
2.1	02/18/14	Updating of Change Log
2.2	02/21/14	Updating of Change Log
2.3	02/24/14	Updating of Change Log
2.4	02/26/14	Updating of Change Log

Communication Management Plan

Version	Revision Date	Revision Comment
1.0	01/24/14	Creation of Communication Management Plan
1.1	01/31/14	Updating of Communication Management Plan
2.0	02/18/14	Formatting of Communication Management Plan
2.1	02/21/14	Rearrangement of items
2.2	02/24/14	Addition of specific meeting time and locations

Code Contribution Management Plan

Version	Revision Date	Revision Comment
2.0	02/21/14	Creation of Code Contribution Management Plan, including contact information for the group for people who want to contribute to the system and the deadline for distributing the codes.
2.1	02/24/14	Insertion of decision rules in case of anybody want to contribute to the code

Copyright Declarations and License Choices

Version	Revision Date	Revision Comment
1.0	01/26/14	Creation of Copyright Declarations and License Choices (CC BY-SA-3.0 license)
2.0	02/21/14	Definition of CC BY-SA-3.0 license

Data Flow Diagram

Version	Revision Date	Revision Comment
1.0	01/26/14	Creation of Data Flow Diagram
2.0	02/17/14	Deletion of Web Interface entity and editing of data flow items
2.1	02/21/14	Editing of data flow items, separation of “Combine&Compare” process into two processes (“Covert to JSON” and “Generate SPDX”) Decomposition of Data Flow Diagram (entities, processes, data flow items)
2.2	02/24/14	Renaming process “Convert to JSON” into “Convert to JSON and Merge” Rewriting process decomposition in IF - ELSE statements

Database / Data Store Structures of the System

Version	Revision Date	Revision Comment
1.0	01/26/14	Creation of Data Storage Structure of the System
2.0	02/21/14	Attachment of the database schema

Distribution System

Version	Revision Date	Revision Comment
1.0	01/26/14	Creation of Distribution System (using Github as the code repository)

System Charter

Version	Revision Date	Revision Comment
1.0	01/24/14	Creation of System Charter
1.1	01/31//14	Formating of System Charter
2.0	02/18/14	Formating of System Charter, adding description
2.1	02/23/14	Addition of problems/opportunities and objectives section

System Service Request

Version	Revision Date	Revision Comment
1.0	01/24/14	Creation of System Service Request
1.1	01/31/14	Formating of System Service Request
2.0	02/18/14	Formating of System Charter, adding description
2.1	02/21/14	Addition of the server's hardware specifications
2.2	02/24/14	Addition of software version Added the mail server requirements and the Perl JSON module requirement

Stakeholder/Community

Version	Revision Date	Revision Comment
1.0	01/24/14	Creation of Stakeholder Register
1.1	01/31/14	Formating of Stakeholder Register
2.0	02/18/14	Formating of Stakeholder Register, including the SPDX group in the Stakeholder Register
2.1	02/25/14	Addition of Daniel German to the Ninka community

Use Cases

Version	Revision Date	Revision Comment
2.0	02/24/14	Creation of Use Case for installation, web interface and command line
2.1	02/24/14	Addition of detailed instruction on how to install Ninka, FOSSology, JSON, Perl Module and configure mail server