



**Identify (and find ways to help fix)  
critical open source software  
used by European Public Services**

**June 2022**

**Deloitte.**

## Study Authors

This study was carried out for the European Commission in 2022 by [Deloitte Consulting](#) supported by [InnoCube \(or Inno<sup>3</sup>\)](#) for the open source research and analysis.

Contract: Framework Contract n° DI/07624 - ABCIV Lot3

Request: Request No 623, DG DIGIT B.3

## European Commission

The study was managed by Saranjit Arora (external PM & member of OSPO) and Miguel Diez-Blanco (Commission PM & OSPO Lead) from DIGIT B.3.002. The project saw major contributions from Gijs Hillenius (EC OSPO) ranging from open source perspectives to contacts. Evangelos Tsavalopoulos (Head of Sector) and others within DIGIT B.3 reviewed the project.

## Disclaimer

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the Commission. The Commission does not guarantee the accuracy of the data included in this document. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use which may be made of the information contained therein.

More information on the European Union is available on the Internet (<http://www.europa.eu>).

## © European Union, 2021-2022



The Commission's reuse policy is implemented by [Commission Decision 2011/833/EU of 12 December 2011 on the reuse of Commission documents](#).

Unless otherwise noted (e.g. in individual copyright notices), the reuse of the editorial content on this website owned by the EU is authorized under the [Creative Commons Attribution 4.0 International \(CC BY 4.0\) licence](#). This means that reuse is allowed, provided appropriate credit is given and any changes are indicated.

You may be required to clear additional rights if a specific content depicts identifiable private individuals or includes third-party works. To use or reproduce content that is not owned by the EU, you may need to seek permission directly from the respective right holders. Software or documents covered by industrial property rights, such as patents, trademarks, registered designs, logos, and names, are excluded from the Commission's reuse policy and are not licensed to you.

## EUROPEAN COMMISSION

Directorate-General for Informatics

Directorate DIGIT

Unit B3 — DIGIT.B3 Reusable Solutions

E-mail: [DIGIT-OSPO@ec.europa.eu](mailto:DIGIT-OSPO@ec.europa.eu)

## Table of Contents

Executive Summary .....	5
1. Introduction.....	7
1.1 Context and scope .....	7
1.2 Document Structure.....	8
2. Identifying target information sources.....	9
2.1 Already known critical software .....	9
2.2 EU-FOSSA 2 identified Critical Software.....	11
2.3 ISA2 Sharing and Re-use action (2016.31) identified Critical Software ...	12
2.4 Identify Information Sources.....	14
2.5 Open source expert groups .....	16
2.6 Concluding remarks .....	17
3. Defining “Critical Software” .....	19
3.1 Creating a definition of Critical open source Software .....	19
3.2 Suggested process to identify critical software .....	20
3.3 Concluding Remarks .....	24
4. Preparing for information collection.....	25
4.1 Contacting European Public Services .....	25
4.2 Contacting open source experts.....	28
5. Analysing received information.....	30
5.1 Results from the European Public Services Survey .....	30
5.2 Results of surveying open source experts .....	36
5.3 Analysing critical open source Projects .....	40
6. Exploring and proposing solutions.....	48
6.1 Project Needs.....	48
6.2 Proposed solutions.....	49
6.3 Summary .....	53
7. Conclusion.....	54
7.1 Process related.....	54
7.2 Open source experts .....	54
7.3 The Study.....	55
7.4 Specific recommendations .....	56
8. APPENDIX A List of Files .....	57
9. APPENDIX B Projects by status.....	58

## List of figures

Figure 1: Identifying critical software.....	21
Figure 2: Depiction of a Dependency Tree .....	22
Figure 3: Survey Guide extract - Suggested process to identify critical software.....	22
Figure 4: Survey Guide extract - Steps to identify critical software (1).....	23
Figure 5: Survey Guide extract - Steps to identify critical software (2).....	23
Figure 6: Survey fields - critical software template .....	26
Figure 7: Survey fields - dependencies template .....	26
Figure 8: Contacting Process - FOSSEPS Survey.....	27
Figure 9: Contacting Process - open source Experts Survey .....	29
Figure 10: Project Health according to European public services.....	32
Figure 11: Type of open source policy.....	32
Figure 12: Open source representatives within European public services .....	33
Figure 13: Rules covered through European public services open source policies .....	33
Figure 14: Awareness of security issues within FOSSEPS.....	34
Figure 15: Do FOSSEPSs take actions for dependencies security?.....	34
Figure 16: Type of contribution .....	35
Figure 17: Support to Projects .....	36
Figure 18: Volume of contributions to Apache Tomcat by the main committers .....	41
Figure 19: Volume of contributions to Libxml2 by the main committers.....	42
Figure 20: Bus Factor for libXML2.....	42
Figure 21: Volume of contributions to Curl by the main committers .....	43
Figure 22: Bus Factor for Curl .....	44
Figure 23: Volume of contributions to LibreOffice by the main committers.....	45
Figure 24: Bus Factor for LibreOffice.....	45
Figure 25: Volume of contributions to M2Crypto by the main committers .....	46
Figure 26: Bus Factor for M2Crypto .....	47

## Executive Summary

### Background

With its growing use across and reliance upon by European public services, the need to manage and protect open source as a shared and valuable European public asset is becoming clear. This is the *raison-d'être* of the FOSSEPS<sup>1</sup> Pilot project.

With thousands of open source software, tools and libraries in use, it is challenging to know the health of each, and to know whether it will last the course. If we have any doubts, it becomes a *critical software* for us. This is why the FOSSEPS project commissioned this study to “*create an inventory of Europe’s most critical open source software used by European Services, and to identify solutions to the causes.*”

### What is critical software?

The study team formally defines critical software as

*An open source software is considered as Critical Software, if it is important to an organisation (i.e., it is needed to deliver key services) and is at risk due to a lack of support from an external/external support organisation, or from an inability of its parent software community to guarantee its long-term upkeep, evolution, and maintenance.*

Right from the outset, the study team knew that there were some already known critical software within the industry. After speaking with experts, the study already had a list of 30 critical software. This was a great start to additional information gathering.

### Information gathering

For additional data gathering, the study compiled a list of 191 open source stakeholders from European Public Services and around 20 open source experts in the field of sustainability and those that directly maintained potentially critical software. Public administrations were contacted using surveys, and experts via individually crafted emails. For credibility, we used a Commission email id to communicate with public administrations.

### Responses

The study team received 21 responses from the 191 European Public Services contacted - an 11% response rate. This is clearly a low response. However, this response rate reflects the complexity of the subject, rather than a lack of effort or enthusiasm, as several respondents sent emails appreciating this initiative from the European Commission. It also emerged that they did not have adequate technology tools to establish open source software dependencies. Fortunately, the 21 responding organisations were spread across 14 countries, reflecting widespread open source use across European Public Services.

Open source experts however, were much more forthcoming, with 13 responding from the 20 contacted. Furthermore, this small group provided quality responses and valuable suggestions on how to solve the critical software problem.

### Key findings from European Public Services

Analysis of the data gathered revealed several points. Firstly, the gathered data sample was small, and only indicated that LibreOffice was to be considered a critical software. This

---

<sup>1</sup> <https://joinup.ec.europa.eu/collection/fosseps>

is clearly not the case in the real world. Further, it showed that public administrations have a high awareness of security issues, and now, sustainability. They also recognise the need to contribute more towards the open source in general and projects they use. There was very little formalised open source policy, but there was a desire to work with other European Public Services to contribute and share ideas.

### **Key findings from open source experts (sustainability and project maintainers)**

Open source experts provided keen insights on the problems, their causes and suggested potential solutions. They agreed with the difficulties in identifying critical software, and agreed that the recent Executive Order from the US mandating a software bill of materials (SBOM) for every software component, would help.

As expected, the most important areas for sustaining projects and moving projects from their critical status, were the need for additional funding, attracting additional contributors (tied to funding in many ways, but there are other factors such as awareness and attractiveness of the project), retaining talent, and increasing diversity. They also suggested that better software governance would improve sustainability.

### **Recommendations for the European Commission**

1. Further studies should focus on specific types/groups of software, and only contact software and sustainability experts, rather than European Public Services.
2. Financially and technically support the use of open source tools that show transitive dependencies and generate reliable SBOMs
3. Support *community initiatives* to define metrics and methodologies e.g., Bus Factor, to identify open source projects in need of external help for maintenance.
4. Encourage European Public Service to consider financial contribution to open source, in line with the EC's open source Software Strategy 2020 – 2023.
5. Raise awareness of the concept of the *open source virtuous circle* in European public services, encouraging them to invest in support services directly with the producer communities, thereby increasing the sustainability of these open source projects.
6. Help create European public services specific guidelines in the best practices for OSPO regarding contribution (e.g. Good Governance Initiative of [www.ospo.zone](http://www.ospo.zone)).
7. Encourage European public services and European companies update the open source they use and initiate and fund specific Long Time Support versions to accommodate contexts incompatible with the release cycles of the projects.
8. Initiate connections with EU organisations dedicated to cybersecurity to work with them on the topic of Software Supply Chain security.

### **In conclusion**

The study succeeded in identifying over 30 critical software, understanding what caused them to be in this state, and suggestions for a way forward to eliminate these reasons. It concludes that European Public Services can contribute more towards open source projects, with many experts suggesting the use of legislation to pass business to smaller projects. The study team also spoke directly with maintainers of some critical software projects, and documented their comments. This is invaluable information to help identify, help and keep safe critical software and ensure long-term sustainability of this valuable open source resource. It is no wonder that open source is often referred to as, *digital infrastructure*.

**END**

## 1. Introduction

As the use of open source increases across European institutions and European public administrations<sup>2</sup>, so does the need to manage and protect it, perhaps by treating it as a collective, shared, and valuable European public asset.

Building on the success of earlier initiatives such as OSOR, ISA<sup>2</sup> and the EU-FOSSA initiative, the European Parliament asked the European Commission to conduct the FOSSEPS<sup>3</sup> Pilot Project to catalyse and establish this cooperation.

This study relates to one of the FOSSEPS pilot project's key objectives, namely, to *create an inventory of Europe's most critical open source software used by European Services*, and to identify solutions to the causes.

This report describes the process, mechanisms used, results obtained, and conclusions from the study.

### 1.1 Context and scope

This study started a few weeks before the *Log4shell*<sup>4</sup> zero-day security vulnerability hit millions of IT systems worldwide. It reminded the world that security vulnerabilities that first reached a mainstream audience back in 2014 with Heartbleed, were still very much present.

A few months later, the less publicized FakerJS/ColorJS incident<sup>5</sup> was a clear reminder that security is *only one part of the more complex issue of free and open source*<sup>6</sup> software sustainability.

Even if the issue is still not resolved, substantive community-driven initiatives, like SustainOSS<sup>7</sup>, have been active for a few years. The US President's subsequent executive order on improving the nation's cybersecurity<sup>8</sup> mandating the use of a SBOM (software bill of material) for every software component, shows interest at the highest political levels.

---

<sup>2</sup> European Public Services and European Public Administrations are used interchangeably within this document and mean "national, regional and local government bodies/administrations within a country".

<sup>3</sup> <https://joinup.ec.europa.eu/collection/fosseps>

<sup>4</sup> <https://en.wikipedia.org/wiki/Log4Shell>

<sup>5</sup> [https://www.theregister.com/2022/01/10/npm\\_fakerjs\\_colorsjs/](https://www.theregister.com/2022/01/10/npm_fakerjs_colorsjs/)

<sup>6</sup> Throughout this document we use the words "open source" to imply Free and Open Source Software.

<sup>7</sup> <https://sustainoss.org/>

<sup>8</sup> <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>

The objective of this pilot project was to identify *critical*<sup>9</sup> open source projects used by European Public Services. The scope covers any type of software, as long as another software uses or relies on it, so it includes libraries and frameworks, infrastructure, but also applications that could serve as platforms.

The study targeted EU27 national, regional, and local administrations.

## 1.2 Document Structure

The document is structured as follows:

Chapter 1: Introduction

Chapter 2: Identifying target information sources

Chapter 3: Defining "critical software"

Chapter 4: Preparing for information collection

Chapter 5: Analysing received information

Chapter 6: Exploring and proposing solutions

Chapter 7: Conclusions

Appendix A: List of Files

Appendix B: Projects by status

---

<sup>9</sup> A definition for "critical software" is defined in Chapter 3



## 2. Identifying target information sources

Prior to the start of this study, the European Commission and study team knew that there was already some knowledge/awareness of open source projects that could be considered critical software. It made sense therefore to collect that information right at the outset. So, the team was asked to do that, alongside, identifying target organisations and open source experts who would be contacted by the study team. This is summarised as:

- A. Already known critical software: Make a list of already known critical software.
- B. Identify information sources: Identify and examine open source experts, groups, and other sources from whom we can obtain information about critical open source software.
- C. Identify key open source experts: From the list established in (B), identify a small group of open source elders/advisors who can guide the project on specific open source issues and review the interim/final list of software identified as part of this project.

As input, the study team found or added to information received from the EC:

- Lists of critical software identified by earlier EC projects, specifically the EU-FOSSA 2 project and the OSS inventory study carried out as part of the ISA2 Sharing and Re-use action (2016.31).
- Individuals concerned with the topic of open source sustainability, falling into three groups:
  - Experts in developer/packager communities
  - Experts in open source sustainability
  - People specialised in open source in public services in different EU countries
- A comparison and combination of the starting critical software list and sustainability involved open source experts, should allow the creation of the first list of critical software used by European Public Services.

### 2.1 Already known critical software

Inno<sup>3</sup> contacted several individuals involved in various open source projects asking them for examples of libraries/infrastructure components<sup>10</sup>. These file named "Task\_2\_A already\_known\_critical\_FOSS.ods" (uploaded on Joinup) contains that first list of software. The table below shows those which could have a significant impact if they fail.

---

<sup>10</sup> More than 20 communities were contacted in November/December 2021. Contact persons were at least significant committers in these communities, many of them being maintainers/founders.

No.	Component	Description	Reason/s for Criticality
1	Liftweb	A Scala library used, among others, by the Rudder application, which was listed by the open source software inventory study as part of the ISA2 Sharing and Re-use action (2016.31).	The Rudder core team has indicated that the project faces maintenance problems.
2	Hackney	The most depended upon package in Erlang. It is maintained by a single individual (Benoît Chesnau), who also maintains other immensely popular Erlang packages. He is currently trying to find ways to improve the situation from a sustainability point of view.	The Erlang community has identified it as being potentially at risk.
3	FFmpeg	A ubiquitous multimedia framework used in reference applications like VLC. Its maintenance problems were brought to our attention by VLC creator and maintainer Jean-Baptiste Kempf.	VLC's maintainer noted that its continued maintenance should not be taken for granted.
4	CouchDB	A document-oriented NoSQL database perceived to have maintenance problems by people in the Erlang ecosystems, such as the delay of integration of fixes. Also of the same understanding are users in other ecosystems, such as web development, who are unable to contribute to it due to the complexity of its language.	Its maintenance is perceived as problematic by people within and outside the Erlang ecosystems.
5	Hunspell	One of the most used open source spell checkers. It is used by reference applications like Firefox and LibreOffice. Caolán McNamara, who is running for the board of The Document Foundation (TDF) this year and is especially aware of the question of dependencies sustainability, cited it along with the next two items.	Caolán McNamara (core developer for LibreOffice) noted that it faces potential maintenance problems.
6	Redland, Raptor and Rasqal	Popular RDF libraries.	Caolán McNamara noted that the project faces potential maintenance problems.
7	Libeot	A library for parsing Embedded OpenType files (Microsoft embedded font "standard").	Caolán McNamara noted that this project faces potential maintenance problems.
8	Doorkeeper-i18n	Handles the i18n capabilities for the popular Doorkeeper ruby gem, which is used, among others, by the iconic Decidim project. The core team of Decidim has an open issue related to the management of dependencies at risk.	The project is mentioned in the issue opened by the core team of Decidim to list its dependencies at risk.
9	Fog: Local	Another Decidim dependency whose sustainability is problematic.	The project is mentioned in the issue open by the core team of Decidim to list its dependencies at risk.

## 2.2 EU-FOSSA 2 identified Critical Software

The goal here was to leverage information from the EU-FOSSA 2 project inventory study to identify critical Open Source. The document EU-FOSSA 2 - SR80 - Deliverable D07.02 provides a list of 56 projects with a high Business Criticality Index (BCI). These projects are categorized according to their technical nature (Library, Infrastructure, Tool, Application).

User-facing applications have been excluded, except those with exceptional dissemination or those that could be used as a platform for other applications (noted "key applications"). Following this, the ones considered to have a solid sustainability model (e.g., Git) were excluded, as well as the items that would form part of any standard GNU/Linux distribution (e.g., glibc) that could be studied as part of Debian.

Component	BCI	Category	URL
Qt	0,8	Library	<a href="https://www.qt.io/">https://www.qt.io/</a>
Firefox	0,6	Key application	<a href="https://www.mozilla.org">https://www.mozilla.org</a>
Nspr & nss	0,5390131281122	Library	<a href="https://n-2.org/">https://n-2.org/</a>
Apache Tomcat	0,4260208239022	Infrastructure	<a href="https://tomcat.apache.org/">https://tomcat.apache.org/</a>
VLC	0,3796825700353	Key application	<a href="https://www.videolan.org/">https://www.videolan.org/</a>
XULRUNNER	0,3580361411224	Library	<a href="https://www.mozilla.org">https://www.mozilla.org</a>
OpenLDAP	0,3565595291987	Infrastructure	<a href="https://www.openldap.org/">https://www.openldap.org/</a>
OpenSSH	0,3551652331371	Infrastructure	<a href="https://www.openssh.com/">https://www.openssh.com/</a>
curl	0,3510909913988	Tool	<a href="https://curl.se/">https://curl.se/</a>
cyrus-sasl	0,3505839746491	Library	<a href="https://www.cyrusimap.org/sasl/">https://www.cyrusimap.org/sasl/</a>
SVN	0,3504028972385	Tool	<a href="https://subversion.apache.org/">https://subversion.apache.org/</a>
libgcrypt	0,3495518334087	Library	<a href="https://www.gnupg.org/related_software/libgcrypt/">https://www.gnupg.org/related_software/libgcrypt/</a>
m2crypto	0,3485377999094	Library	<a href="https://gitlab.com/m2crypto/m2crypto">https://gitlab.com/m2crypto/m2crypto</a>
GnuPG	0,2004206903145	Tool	<a href="https://www.gnupg.org/">https://www.gnupg.org/</a>
bzip2	0,1724400181077	Library	<a href="http://www.bzip.org/">http://www.bzip.org/</a>
Gecko SDK	0,1579787742614	Library	<a href="https://www.mozilla.org">https://www.mozilla.org</a>
libxml2	0,1526120416478	Library	<a href="http://www.xmlsoft.org/">http://www.xmlsoft.org/</a>

## 2.3 ISA2 Sharing and Re-use action (2016.31) identified Critical Software

The file lists 30 open source projects and ranks them by sustainability. This consolidated metric does not draw a clear distribution between projects. Their impact has been manually ranked on the global open source ecosystem (Structural, Moderate, Low). The ones for which a reasonable degree of confidence in the solidity of their sustainability model is held have not been considered (e.g., Ubuntu).

Table 1: ISA2 Sharing and Re-use action identified critical software

Component	URL	Tags	Category
<b>LibreOffice</b>	<a href="https://libreoffice.org">https://libreoffice.org</a>	C++, Desktop, Office Suite	Key application
<b>Firefox</b>	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	C++, Rust, Mozilla	Key application
<b>Apache Tomcat</b>	<a href="https://tomcat.apache.org/">https://tomcat.apache.org/</a>	Java	Infrastructure
<b>Debian</b>	<a href="https://www.debian.org/">https://www.debian.org/</a>	Operating System	Infrastructure
<b>Thunderbird</b>	<a href="https://www.thunderbird.net">https://www.thunderbird.net</a>	C++, email	Key application
<b>PostgreSQL</b>	<a href="https://www.postgresql.org/">https://www.postgresql.org/</a>	Database	Infrastructure

The complete list of the EU-FOSSA<sup>2</sup>, ISA2 and already known critical software is presented in the below combined list.

Table 2: List of the EU-FOSSA 2, ISA2 and already known critical software

Component	URL	Origin	Tags	Category	Governance Type
<b>libgcrypt</b>	<a href="https://www.gnupg.org/related_software/libgcrypt/">https://www.gnupg.org/related_software/libgcrypt/</a>	EU-FOSSA 2	Security	Low level library	Community driven – informal
<b>GnuPG</b>	<a href="https://www.gnupg.org/">https://www.gnupg.org/</a>	EU-FOSSA 2	Security	Tool	Community driven – informal
<b>cyrus-sasl</b>	<a href="https://www.cyrusimap.org/sasl/">https://www.cyrusimap.org/sasl/</a>	EU-FOSSA 2	C, Identity	High level Library	Community driven – informal
<b>Thunderbird</b>	<a href="https://www.thunderbird.net">https://www.thunderbird.net</a>	ISA <sup>2</sup>	C++, email	Key application	Community driven
<b>CouchDB</b>	<a href="https://couchdb.apache.org/">https://couchdb.apache.org/</a>	Already known critical software	Erlang	Infrastructure	Community driven – Foundation
<b>hunspell</b>	<a href="https://github.com/hunspell/hunspell/">https://github.com/hunspell/hunspell/</a>	Already known critical software	C, Mozilla, Libreoffice	High level Library	Community driven – informal
<b>Apache Tomcat</b>	<a href="https://tomcat.apache.org/">https://tomcat.apache.org/</a>	ISA <sup>2</sup>	Java	Infrastructure	Community driven – Foundation
<b>Curl</b>	<a href="https://curl.se/">https://curl.se/</a>	EU-FOSSA	C	Low level	Community driven

		2		library	- informal
<b>libxml2</b>	<a href="http://www.xmlsoft.org/">http://www.xmlsoft.org/</a>	EU-FOSSA 2	Gnome	Low level library	Community driven - Foundation
<b>m2crypto</b>	<a href="https://gitlab.com/m2crypto/m2crypto">https://gitlab.com/m2crypto/m2crypto</a>	EU-FOSSA 2	Python, Security	High level Library	Community driven - informal
<b>XULRUNNER</b>	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	EU-FOSSA 2	C++, Mozilla	High level Library	Community driven - Foundation
<b>Nspr &amp; nss</b>	<a href="https://n-2.org/">https://n-2.org/</a>	EU-FOSSA 2	Mozilla	High level Library	Community driven - informal
<b>Liftweb</b>	<a href="https://www.liftweb.net/">https://www.liftweb.net/</a>	Already known critical software	Java, Scala, Rudder	High level Library	Community driven - informal
<b>SVN</b>	<a href="https://subversion.apache.org/">https://subversion.apache.org/</a>	EU-FOSSA 2	Deprecat ed	Tool	Community driven - Foundation
<b>Gecko SDK</b>	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	EU-FOSSA 2	Mozilla	High level Library	Community driven - Foundation
<b>Qt</b>	<a href="https://www.qt.io/">https://www.qt.io/</a>	EU-FOSSA 2	C++, UI	Low level library	Vendor driven
<b>OpenLDAP</b>	<a href="https://www.openldap.org/">https://www.openldap.org/</a>	EU-FOSSA 2	C, Identity	Infrastruct ure	Community driven - Foundation
<b>PostgreSQL</b>	<a href="https://www.postgresql.org/">https://www.postgresql.org/</a>	ISA <sup>2</sup>	Database	Infrastruct ure	Community driven - informal
<b>doorkeeper-i18n</b>	<a href="https://github.com/doorkeeper-gem/doorkeeper-i18n">https://github.com/doorkeeper-gem/doorkeeper-i18n</a>	Already known critical software	Ruby, Decidim	High level Library	Community driven - informal
<b>Fog::Local</b>	<a href="https://github.com/fog/fog-local">https://github.com/fog/fog-local</a>	Already known critical software	Ruby, Decidim	High level Library	Community driven - informal
<b>bzip2</b>	<a href="https://sourceware.org/bzip2/bzip2-howto/using-bzip2.html">https://sourceware.org/bzip2/bzip2-howto/using-bzip2.html</a>	EU-FOSSA 2	C	Low level library	Community driven - informal
<b>Debian</b>	<a href="https://www.debian.org/">https://www.debian.org/</a>	ISA <sup>2</sup>	Operatin g System	Infrastruct ure	Community driven - Foundation
<b>Firefox</b>	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	ISA <sup>2</sup>	C++, Rust, Mozilla	Key application	Community driven - Foundation
<b>OpenSSH</b>	<a href="https://www.openssh.com/">https://www.openssh.com/</a>	EU-FOSSA 2	C, Security	Infrastruct ure	Community driven - Foundation
<b>Libeot</b>	<a href="https://github.com/umanwalt/libeot">https://github.com/umanwalt/libeot</a>	Already known critical software	C, Fonts, Desktop, Libreoffice	High level Library	Community driven - informal
<b>FFmpeg</b>	<a href="https://ffmpeg.org/">https://ffmpeg.org/</a>	Already known critical	C, multimedia	Low level library	Community driven - informal

		software			
<b>redland, raptor and rasqal</b>	<a href="https://librdf.org/">https://librdf.org/</a>	Already known critical software	C, RDF, LibreOffice	High level Library	Community driven - informal
<b>Hackney</b>	<a href="https://github.com/benoitc/hackney">https://github.com/benoitc/hackney</a>	Already known critical software	Erlang	High level Library	Community driven - informal
<b>LibreOffice</b>	<a href="https://libreoffice.org">https://libreoffice.org</a>	ISA <sup>2</sup>	C++, Desktop, Office Suite	Key application	Community driven - Foundation
<b>VLC</b>	<a href="https://www.videolan.org/">https://www.videolan.org/</a>	EU-FOSSA 2	C, Multimedia	Key application	Community driven - Foundation
<b>libgcrypt</b>	<a href="https://www.gnupg.org/related_software/libgcrypt/">https://www.gnupg.org/related_software/libgcrypt/</a>	EU-FOSSA 2	Security	Low level library	Community driven - informal
<b>GnuPG</b>	<a href="https://www.gnupg.org/">https://www.gnupg.org/</a>	EU-FOSSA 2	Security	Tool	Community driven - informal

## 2.4 Identify Information Sources

### 2.4.1 Consolidate EU 27 contacts

This first part aims at consolidating a list of contacts in the European Public Services (FOSSEPS) from the 27 different EU countries, at three different levels:

- National
- Regional
- Local

This has been achieved mainly by consolidating data provided by the European Commission.

The sheet "FOSSEPS 2" from the file "1 - Initial FOSSEPS and other contacts provided.xlsx" has been structured into a proper row sequence and imported into a sqlite database.

In the file "1 - FOSSEPS Contacted.xlsx", the sheet "Trasys Team" is a sanitized and consolidated version of the sheet "EC Team". DIGIT has checked the data to remove organisations not considered to be European Public Services.

There are 15 Country vs Scope (Level) combinations for which no contact has been identified, with most of them being at local level.

Table 3: Country vs Scope (Level) - No contacts identified

No	Country	Scope
1	Belgium	Small/local council
2	Bulgaria	Regional Body
3	Bulgaria	Small/local council
4	Croatia	Small/local council
5	Cyprus	Regional Body
6	Cyprus	Small/local council
7	Czech Republic	Small/local council
8	Estonia	Small/local council
9	Lithuania	Small/local council
10	Luxembourg	Small/local council
11	Malta	Regional Body
12	Malta	Small/local council
13	Slovakia	Regional Body
14	Slovakia	Small/local council
15	Spain	Small/local council

The comprehensive list of contacts can be found in the following files (same information in different formats):

- *Task\_2\_B\_1\_FOSSEPS\_contacts.csv*
- *Task\_2\_B\_1\_FOSSEPS\_contacts.sqlite*
- *Task\_2\_B\_1\_FOSSEPS\_contacts.ods*

## 2.4.2 Useful open source organisations and experts

This second part is aimed to identify open source contacts, both at organisational and individual level, who could be relevant to the study.

They were initially classified into seven categories:

1. European, National and Regional opensource bodies/associations
2. Recognised open source industry experts across Europe
3. Key open source developers
4. open source associations
5. European open source funding organisations
6. open source influencers/groups
7. open source experts/organisations

During the workshop<sup>11</sup> with the European Commission, another classification was identified for which Inno<sup>3</sup> proposes the following values:

1. Open source companies' associations
2. Foundation for corporate projects
3. Foundations for community projects
4. Language specific foundations
5. Open source projects
6. Open source sustainers
7. Package Management
8. Open source civil society
9. Funding organisations

The category "Sustainers" refers to organisations and individuals with a dedicated focus on open source sustainability, such as:

1. The SustainOSS Community
2. The ChaOSS Community
3. The Libraries.io Community
4. Open Collective
5. TideLift
6. The open source Security Foundation

The complete list of contacts is available in the file "*Task\_2\_B\_2-8\_contacts.ods*".

## 2.5 Open source expert groups

The suggested list of open source Software Experts Group (EC-OSS-EG) to be discussed with the European Commission is currently composed of the following.

Contact	Organisation
Brian Behlendorf	<ul style="list-style-type: none"><li>• Open Source Security Foundation</li></ul>
Andrew Nesbitt	<ul style="list-style-type: none"><li>• Creator of Libraries.io</li></ul>
Ben Nickols	<ul style="list-style-type: none"><li>• Open Collective</li><li>• SustainOSS community</li></ul>
Caolán McNamara	<ul style="list-style-type: none"><li>• LibreOffice</li></ul>
Chad Whitacre	<ul style="list-style-type: none"><li>• Creator of Gratipay</li><li>• Co-founder of SustainOSS</li></ul>
Chris Lamb	<ul style="list-style-type: none"><li>• Director at Software in the Public interest</li><li>• Contributor to Debian LTE</li></ul>
Daniel Izquierdo Cortázar	<ul style="list-style-type: none"><li>• ChaOSS Community</li><li>• CEO of Bitergia</li></ul>
Dirk-Willem van Gulik	<ul style="list-style-type: none"><li>• Apache Foundation</li></ul>

---

<sup>11</sup> The FOSSFOSSEPS workshop was held on the 6th and 7th December 2021.



Gael Blondelle	<ul style="list-style-type: none"> <li>Eclipse Foundation Europe</li> </ul>
Karen M. Sandler	<ul style="list-style-type: none"> <li>Software Freedom Conservancy (home of the git project etc.)</li> </ul>
Luis Villa	<ul style="list-style-type: none"> <li>CEO of Tidelift</li> </ul>

These are indicated in the global list and flagged as "Suggested by Inno<sup>3</sup>" in the "Core Group" column of the file named "Task\_2\_B\_2-8\_contacts.ods".

## 2.6 Concluding remarks

The study team was successful in this part of the project. It was able to identify people and organisations who could be contacted, and compile a list of already known critical software. This list of software is shown the table below and the file named "Task\_2\_A already\_known\_critical\_FOSS.ods".

Origin	Component	URL	Tags	Category
Already known critical software	CouchDB	<a href="https://couchdb.apache.org/">https://couchdb.apache.org/</a>	Erlang	Infrastructure
Already known critical software	doorkeeper-i18n	<a href="https://github.com/doorkeeper-gem/doorkeeper-i18n">https://github.com/doorkeeper-gem/doorkeeper-i18n</a>	Ruby, Decidim	Library
Already known critical software	FFmpeg	<a href="https://ffmpeg.org/">https://ffmpeg.org/</a>	C, multimedia	Library
Already known critical software	Hackney	<a href="https://github.com/benoitc/hackney">https://github.com/benoitc/hackney</a>	Erlang	Library
Already known critical software	hunspell	<a href="https://github.com/hunspell/hunspell/">https://github.com/hunspell/hunspell/</a>	C, Mozilla, Libreoffice	Library
Already known critical software	libeot	<a href="https://github.com/umanwizard/libeot">https://github.com/umanwizard/libeot</a>	C, Fonts, Desktop, Libreoffice	Library
Already known critical software	liftweb	<a href="https://www.liftweb.net/">https://www.liftweb.net/</a>	Java, Scala, Rudder	Library
Already known critical software	redland, raptor and rasqal	<a href="https://librdf.org/">https://librdf.org/</a>	C, RDF, LibreOffice	Library
EU-FOSSA 2	bzip2	<a href="http://www.bzip.org/">http://www.bzip.org/</a>	C	Library
EU-FOSSA 2	Curl	<a href="https://curl.se/">https://curl.se/</a>	C	Library
EU-FOSSA 2	cyrus-sasl	<a href="https://www.cyrusimap.org/sasl/">https://www.cyrusimap.org/sasl/</a>	C, Identity	Library
EU-FOSSA 2	Gecko SDK	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	Mozilla	Library
EU-FOSSA 2	GnuPG	<a href="https://www.gnupg.org/">https://www.gnupg.org/</a>	Security	Tool
EU-FOSSA 2	libgcrypt	<a href="https://www.gnupg.org/related_software/libgcrypt/">https://www.gnupg.org/related_software/libgcrypt/</a>	Security	Library
EU-FOSSA 2	libxml2	<a href="http://www.xmlsoft.org/">http://www.xmlsoft.org/</a>	Gnome	Library
EU-FOSSA 2	m2crypto	<a href="https://gitlab.com/m2crypto/m2crypto">https://gitlab.com/m2crypto/m2crypto</a>	Python, Security	Library
EU-FOSSA 2	Nspr & nss	<a href="https://n-2.org/">https://n-2.org/</a>	Mozilla	Library

Origin	Component	URL	Tags	Category
EU-FOSSA 2	OpenLDAP	<a href="https://www.openldap.org/">https://www.openldap.org/</a>	C, Identity	Infrastructure
EU-FOSSA 2	OpenSSH	<a href="https://www.openssh.com/">https://www.openssh.com/</a>	C, Security	Infrastructure
EU-FOSSA 2	qt	<a href="https://www.qt.io/">https://www.qt.io/</a>	C++, UI	Library
EU-FOSSA 2	SVN	<a href="https://subversion.apache.org/">https://subversion.apache.org/</a>	Deprecated	Tool
EU-FOSSA 2	VLC	<a href="https://www.videolan.org/">https://www.videolan.org/</a>	C, Multimedia	Key application
EU-FOSSA 2	XULRUNNER	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	C++, Mozilla	Library
ISA <sup>2</sup>	ApacheTomcat	<a href="https://tomcat.apache.org/">https://tomcat.apache.org/</a>	Java	Infrastructure
ISA <sup>2</sup>	Debian	<a href="https://www.debian.org/">https://www.debian.org/</a>	Operating System	Infrastructure
ISA <sup>2</sup>	Firefox	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	C++, Rust, Mozilla	Key application
ISA <sup>2</sup>	LibreOffice	<a href="https://libreoffice.org">https://libreoffice.org</a>	C++, Desktop, Office Suite	Key application
ISA <sup>2</sup>	PostgreSQL	<a href="https://www.postgresql.org/">https://www.postgresql.org/</a>	Database	Infrastructure
ISA <sup>2</sup>	Thunderbird	<a href="https://www.thunderbird.net">https://www.thunderbird.net</a>	C++, email	Key application
Already known critical software	Fog::Local	<a href="https://github.com/fog/fog-local">https://github.com/fog/fog-local</a>	Ruby, Decidim	Library

### 3. Defining “Critical Software”

This chapter sets out to define what is meant by Critical Software in the context of this project, and to ensure that the definition is in line with the understanding within the open source industry, as validated by some of the open source experts the study team contacted.

#### 3.1 Creating a definition of Critical open source Software

To define the term Critical Open Source Software, it is necessary to define its three sub-components, namely, Open Source, Software and Critical, separately within the context of the current study.

The definition of open source Software (OSS), which is also called Free and open source Software (FOSS or FLOSS) is software that meets the criteria proposed by the Free Software Foundation<sup>12</sup> and the open source Initiative<sup>13</sup>.

Defining “Software” in the context of the present study: In the context of this project, the word Software means any open source software, library, toolkit, or application development framework.

##### 3.1.1 Defining “Critical”

The meaning of “critical” in this context encompasses two dimensions.

- The importance of the software to the organisation
- Its state of health or sustainability

###### **Importance of the considered software**

The importance of the software to a specific organisation, itself has two sub-components:

- How much it is used, or relied upon
- The impact on the organisation if it should fail

The impact in the case of the loss or dysfunction of the concerned software, even though it may not reach the threshold of vitality that is implied by “critical,” is a measure of the ability for a public service to deliver its core services in case the concerned software would stop working.

---

<sup>12</sup> <https://www.fsf.org/>

<sup>13</sup> <https://opensource.org/osd>

### **Risks associated with the use of the software**

A *Critical software* is also understood as being “in a critical state of health.” This of course presents a risk to the organisation. This risk can be either *extrinsic* or *intrinsic*.

**Extrinsic risks** are related to the **context** in which the software is used, while **intrinsic risks** are related to the **software project itself**. The two dimensions are strongly related.

Example of Extrinsic risks:

***Un-Supported:*** *The software is used, but there is no support contract for it.*

Example of Intrinsic risks are related to the software project itself.

***Sustainability:*** *Only a few open source experts maintain this software and its long-term sustainability is at risk. Therefore, the software becomes critical for the organisation / Europe.*

Due to these two dimensions being strongly related, to mitigate the risks, the measures adopted need to act both on the extrinsic cause (the context) and on the intrinsic cause (the project itself) in parallel.

### **3.1.2 Critical Software – Definition**

In summary, we can define Critical Software as follows:

“An open source software is considered as Critical Software, if it is important to an organisation (i.e., it is needed to deliver key services) and is at risk due to a lack of support from an external/external support organisation, or from an inability of its parent software community to guarantee its long-term upkeep, evolution, and maintenance.”

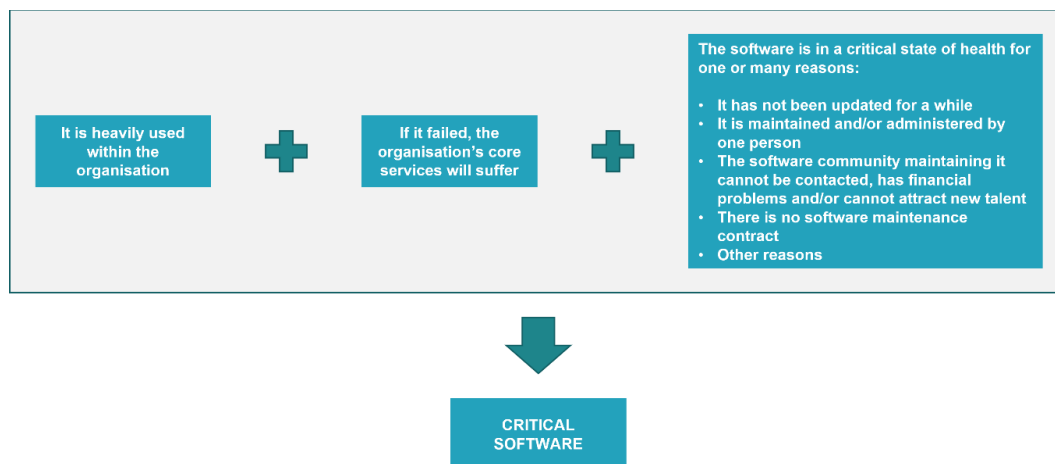
### **3.2 Suggested process to identify critical software**

The suggested process to identify critical open source software in an organisation requires us to follow sub-components identified above. Given that we are only dealing with Open Source, we are left with three aspects to study:

1. the importance of the software to the organisation (based on usage and reliance upon it)
2. the impact on the organisation caused by the risk of the software failing, and
3. its state of health, from a sustainability/longevity perspective.

The schematic below captures this.

Figure 1: Identifying critical software



### 3.2.1 Establishing Importance

The study team used both quantitative and qualitative approaches to assess the importance of a specific open source software in use at the public administrations.

For the degree of usage, we can look at:

- The cumulative number of instances across all departments in the organisation.
- The relative usage for this type of software (e.g., 10 PostgreSQL servers out of 50 database servers).

Adding a qualitative qualification highlighting the importance of the software would be interesting, e.g., we use only 1 PostgreSQL server, but it is used by the accounting department.

The methods for identifying open source software will also vary based on their nature. There is a fundamental difference between the identification of development dependencies<sup>14</sup> and the identification of infrastructure software.

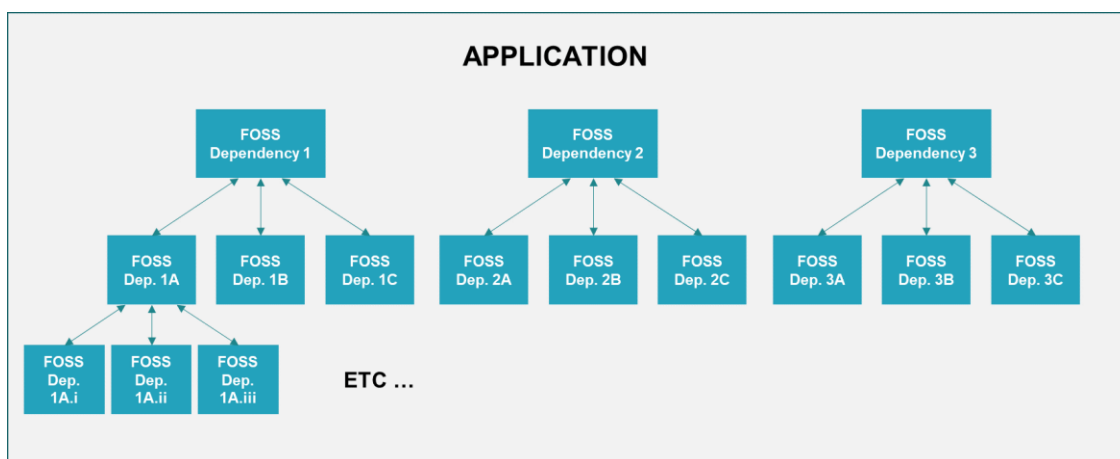
Whereas infrastructure items are at a higher level and therefore easily identifiable, development dependencies can be more difficult to identify.

Development dependencies often rely on other dependencies, creating a "dependency tree". It is hard to have visibility of all dependencies, especially at the bottom part of the "dependency tree". Software tools such as OSS Review Toolkit<sup>15</sup> can help in this regard.

<sup>14</sup> Dependencies are software components used alone or embedded within software libraries of open source applications or products e.g., application development frameworks, also referred to as SDKs.

<sup>15</sup> <https://github.com/oss-review-toolkit/ort>

Figure 2: Depiction of a Dependency Tree



### 3.2.2 Establishing Criticality

This section combines the other two components, (ii) assessing the impact on the organisation caused by the risk of the software failing, and (iii) assessing its state of health, from a sustainability/longevity perspective.

In a sense, the two look highly qualitative, but there are metrics available that assist in forming a view. The medium to larger public administrations are encouraged to look at the European Commission’s 2021 open source software inventory methodology<sup>16</sup>, which explains a step by step process for creating an inventory and establishing criticality.

In the absence of carrying out a thorough inventory study, the FOSSEPS Pilot project created a survey and a supporting Survey Guide, to help European public services to identify critical software and software dependencies. The following extracts from the Survey Guide provide an indication of the suggested process. The complete Survey Guide can be found in the appendices.

Figure 3: Survey Guide extract - Suggested process to identify critical software

Survey questions	We have purposely designed a short survey, and added help text to make the questions self-explanatory.
Identifying my critical software	This is the primary objective. Right now, European Public Services do not know their critical software. So your answers will help us create a Europe-wide picture.
Simplest way	Get a group of people round a table, brainstorm, and write some potential candidates, and fill in the spreadsheets.
Small effort	Getting a list of key software your organisation uses, look for the most used, and compare support contracts. What’s missing? Ask your developers what they think is really crucial for your organisations. Ask, if this software vanishes, can we survive?
Medium effort	Look at each data source – applications, infrastructure, cloud, virtual machines. Try and create extracts, and do some number crunching. Follow the guidelines below.
Large effort	Create a complete open source software inventory and follow the steps suggested below. Look at what the European Commission did in its own inventory process. Download the recently improved <a href="#">inventory methodology</a> .
Filling in the spreadsheets	Fill in the spreadsheets as best you can. If in doubt, email us and we can set up a call.

<sup>16</sup> <https://joinup.ec.europa.eu/collection/fosseps/news/funding-sustainability>

### 3.2.3 Identifying the most critical open source Applications, Infrastructure and Dependencies Software

The below illustrates the suggested steps included in the Survey Guide to identify the most critical open source Applications, Infrastructure and Dependencies Software (the Survey Guide suggests a separate process for the dependencies, but these have been combined for this report).

Figure 4: Survey Guide extract - Steps to identify critical software (1)

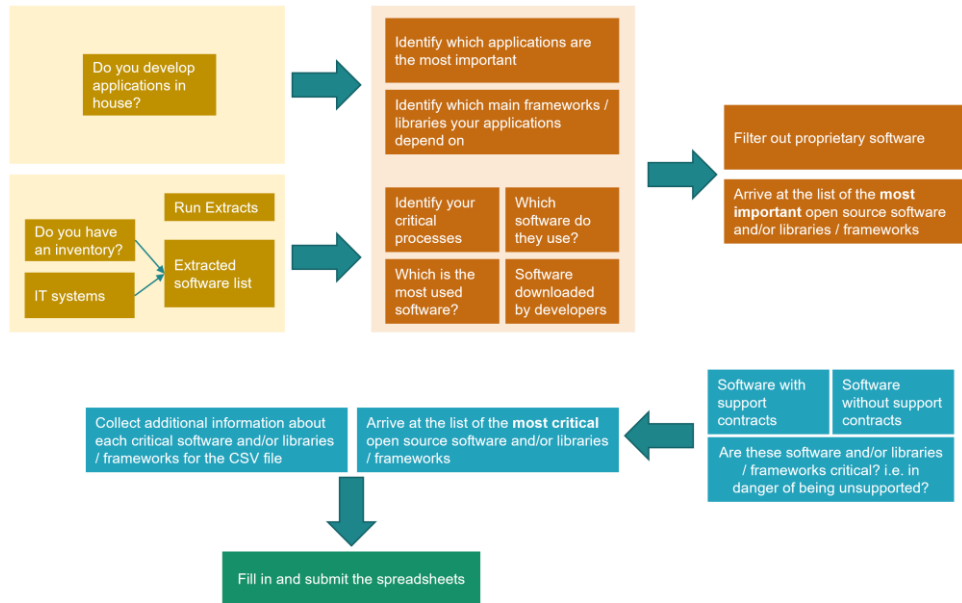


Figure 5: Survey Guide extract - Steps to identify critical software (2)

	Medium - Large organisation	Small organisation
Step 1: Get a list of in-house applications and software items	For applications, contact your Software Asset Manager, they may have an inventory already. For libraries and frameworks, contact your Software Asset Manager, or the CIO/Head of IT to identify the groups with significant software development activity.	For applications, contact the head of IT, who can facilitate an extraction of the list of software used. For libraries and frameworks, contact the person of the IT team in charge of Software Development.
Step 1: Where to look for open source data	Open source software data can be found in the following places: data centres, departmental systems/servers, virtual machines, end user and developer pcs, cloud systems, mobile devices, network switches, routers etc. Data can be equally widespread to include: operating systems such as gnu/Linux (various distributions), applications running on servers for performance, messaging, email and connectivity, software development tools and frameworks and user desktop tools such as web browsers, utilities, office suites, password managers; etc.	
Step 2: Identify the most important ones and their dependencies	Select the most deployed software and the ones which support the most important services provided by your organisation. The product owner / product manager of each application and/or critical service will know about the software and main libraries and frameworks they depend on. Please note that with respect to libraries and frameworks our focus is on first level dependencies and not the whole tree of recursive dependencies.	
Step 3: Filter out the proprietary ones	The person who has provided you with the list of elements in step 2 should be able to tell you which software is free and Open Source. The goal is to only retain the 10 to 20 of the most important applications and dependencies across all your applications.	
Step 4: Evaluate the potential critical state	Identify if you have a support contract for each piece of software and/or dependency, from the software asset management, the IT department or the procurement department. Then identify the critical state of the open source project by asking technical teams in contact with it.	
Step 5: Collect additional information about the software	Additional information will likely be available from the technical teams who use and operate the concerned open source software.	

### **3.3 Concluding Remarks**

The primary objective of this chapter was to define “Critical Software” for the purposes of this project, whilst ensuring that the definition is in line with the understanding within the open source industry.

This report also summarises the processes suggested to organisations assisting them to identify “Critical Software” within their organisations. Such processes cater for different types of organisations (small/medium) and suggest ways depending on the effort such contacts afford to dedicate to this exercise.



## 4. Preparing for information collection

Central to the project is the collection of data and the way the data was collected. This chapter outlines the process/tools used to collect data from the target groups identified and the process followed to reach out to them and to obtain their feedback.

Information regarding critical software was collected from two target groups: (i) European public services, and (ii) select open source experts.

### 4.1 Contacting European Public Services

The European Commission provided an initial list of contacts which were added to later. The idea was to contact at least 3 contacts from each of the 27 member states, making a minimum target of 81 contacts. These contacts were to be drawn from national, regional, and local governmental bodies. In the end, 191 European public services were contacted<sup>17</sup>.

#### Communication mechanism and Survey

The Commission elected to send out an email to all participants, some of whom had been contacted via an earlier critical software information collection exercise. Email templates can be found in the folder *Data Collection - Media Used.zip*.

The email contained a link to **an easy to fill in survey, with spreadsheets** to add names and details of critical software. In addition, to help participants, a **Survey Guide** was also created.

The survey also included questions on how open source is handled in the responding public administration. Two spreadsheets required to be filled in and uploaded – one for open source applications, the other for dependencies.

The Survey Guide was designed to help European Public Services of varying sizes, and provided an explanation of what critical software is, whom to contact within their organisation and how to identify critical software. It also contained a section on FAQs and guidance on filling in the survey spreadsheets.

Both the Survey and the Survey Guide can be found in the *Data Collection - Media Used.zip* file.

---

<sup>17</sup> Appendix FOSSEPS - [7] - Task\_4\_2\_Master\_Contacts\_Tracker\_20220404 – Consolidated shows a table of countries and contact types.

Figure 6: Survey fields - critical software template

Critical Free/Open Source applications													
List of the most important FOSS applications/infrastructure elements for your Organisation and their criticality													
Name of the FOSS project [Mandatory]	URL of the FOSS project	Category of the FOSS project [Mandatory]	Number of instances [Mandatory]	Total number of instances of this type of software (both FOSS and proprietary)	Maximum degree of importance of this FOSS project for your org [Mandatory]	Explanation of the degree of importance	The governance of the project is	Your assessment of the health of this project [Mandatory]	Explanation of your assessment	Type of support contract your organisation has for this FOSS project [Mandatory]	Details on your support contract	Your contribution to this FOSS project	Details on your contribution to this FOSS project

Figure 7: Survey fields - dependencies template

Critical FOSS dependencies												
List of the most important FOSS framework/libraries used in your software developments												
Name of the FOSS framework/library [Mandatory]	URL of the FOSS project	Number of applications this dependency is used in (integer number)	Maximum degree of importance of the applications using this FOSS library/framework	Explanation of the degree of importance	The governance of the project is	Your assessment of the health of this project [Mandatory]	Explanation of your assessment	Type of support contract your organisation has for this FOSS project [Mandatory]	Details on your support contract	Your contribution to this FOSS project	Details on your contribution to this FOSS project	

Two follow up emails were sent by the Commission.

### Target contacts

The table below shows the spread of European public services contacted across the 27 EU member states.

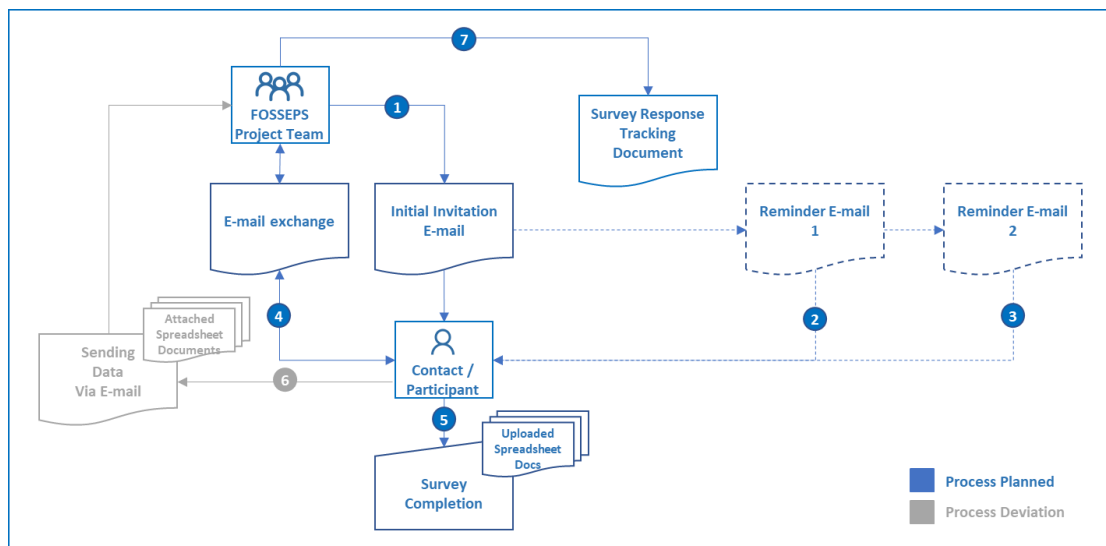
Table 4: Count of FOSSEPS contacted per Country

Country	Scope Category					
	National Government Body	Regional Body	Small / local council	SME Organisation	Supranational	University / Academic
Austria	2	3	2			
Belgium	7	1				
Bulgaria	2					
Croatia	8	1				
Cyprus	2	1				
Czech Republic	1	2	1			
Denmark	1	2	2			
Estonia	4	1				
Finland	3	2	3		2	1
France	3	8	2			
Germany	3	7	4			
Greece	5	1	2			
Hungary	2	3	3			1
Ireland	3	2	2			
Italy	20	1	2			
Latvia	5	1	5			1
Lithuania	6	1				
Luxembourg	3	2				
Malta	3	1			1	
Netherlands	2	5	1			1

Country	Scope Category					
	National Government Body	Regional Body	Small / local council	SME Organisation	Supranational	University / Academic
Poland	1	2	1			
Portugal	6	2	3	1		1
Romania	7	1	1			
Slovakia	3					
Slovenia	6					
Spain	3	9				
Sweden	9	2	4			
Switzerland					1	

\* A full list of organisations can be found in the Response Tracker.

Figure 8: Contacting Process - FOSSEPS Survey



### Tracking responses

A dedicated email ID was created to receive responses – fosseps@inno3.eu, which forwarded emails to all team members. Responses were tracked in detail on a spreadsheet (see media zip file).

## Conclusion

The table below shows the response summary.

This table shows that European public services did spend time in gathering data, and this raised awareness within the organisations. However, the lack of critical software identified shows the difficulty for non-open source experts in doing so.

# Contacted	# Responded	# Replied to the survey	# Sent filled in spreadsheets	# Critical software identified
191	39	23	20	1

## 4.2 Contacting open source experts

To strengthen the identification of critical software, a select group of open source experts concerned with open source sustainability were contacted. In addition, foundations hosting or funding open source projects were also contacted.

In all, 12 carefully selected individuals were contacted, with 8 of them getting back to us with their feedback.

As this was a small group, rather than a survey, the study team decided to write personalised emails with questions specific to the area of interest/expertise of the expert. Several emails led to phone/web meetings and a large amount of useful information was gathered by the study team's open source expert, Inno<sup>3</sup>.

Two sample e-mails sent to the Apache Foundation (Foundation) and APELL (Sustainers) can be found in the folder Data Collection - Media Used.zip file. Questions covered topics covering:

- What critical open source software do you know?
- The main challenges or problems related to open source maintenance/sustainability
- The most promising initiatives finding solutions to those problems/challenges today?
- Should governments/public bodies take specific actions on this topic? For example, the US President's Executive Order on improving cybersecurity by mandating software bill of materials (SBOMs).
- How can we assess the health of open source projects using publicly available resources? What additional information needs to be created/improved/made publicly available?

A response tracker was created along with an online [NocoDB dashboard](#).

## Target contacts

To obtain qualitative information on critical software a small number of open source experts were contacted. These experts were either stakeholders in the business of sustainability (sustainers) or involved within open source foundations (foundations).

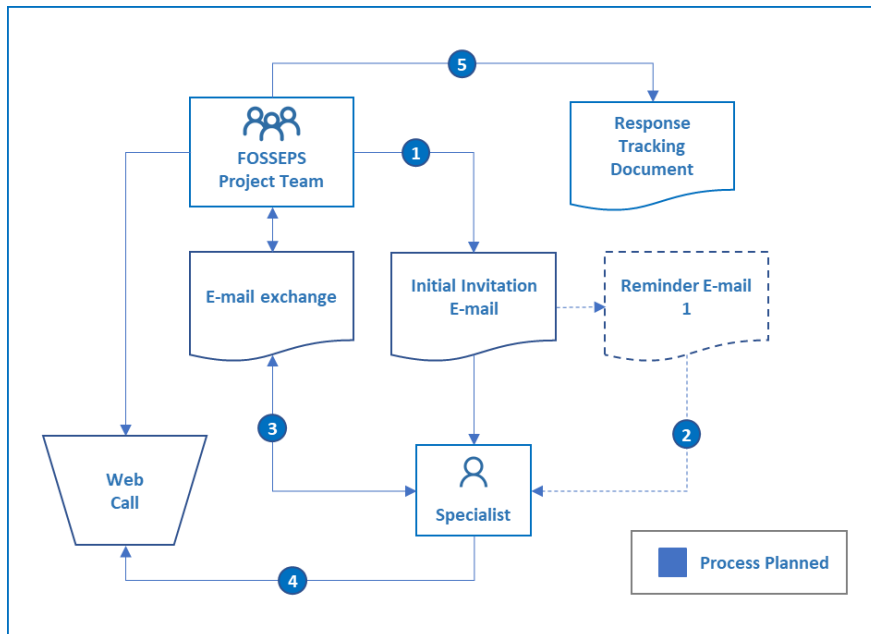
### Sustainers<sup>18</sup>

- OpenSSF
- GratiPay / SustainOSS / Sentry
- OpenCollective
- TideLift
- Librairies.io
- ChaOSS
- FSFE
- APELL

### Foundations

- Apache Foundation
- Eclipse Foundation
- Software Freedom conservancy
- Debian
- Foundation for Public Code

Figure 9: Contacting Process - open source Experts Survey



<sup>18</sup> <https://sustainoss.org/blog/design-is-your-friend/>

## 5. Analysing received information

The information received from European public services and the various open source or open source experts contacted was analysed keeping in mind their expected potential for contribution:

### 1 European Public Services

- 1.a Identify their top critical software
- 1.b How FOSSEPSs handle open source policies, security, technical and financial contributions

### 2 Open source experts (related to Sustainability)

- 2.a Identifying already known and as yet unknown critical software
- 2.b Issues within the open source landscape which lead to criticality
- 2.c Suggestions for improving the sustainability of open source software

### 3 Open source experts (Project Maintainers)

- 3.a Their specific issues, problems, concerns
- 3.b What they need as solutions/help to fix this situation

### 4 Overall

- 4.a Identify the top 30 most critical software
- 4.b Finalise the list of critical software, with clearly identified reasons for the software project's criticality
- 4.c Identify solution

## 5.1 Results from the European Public Services Survey

### 5.1.1 Survey Response

A survey on critical open source software was sent to 191 European public services, 21 returned completed replies. This represents an 11% response rate, which is rather low, given three reminders were sent. Furthermore, responses varied in terms of completeness and validity, with only nine organisations providing complete, high-quality information regarding their infrastructure software, and seven regarding their software dependencies.

We feel that this response rate reflects the complexity of the subject, rather than a lack of effort or enthusiasm, as several respondents sent emails appreciating this initiative from the European Commission. Fortunately, the 21 responding organisations were spread across 14 countries, reflecting a widespread open source use across European Public Services.

Table 5: Respondents per Country

Country	Organisations Responded
Belgium	2
Croatia	2
Denmark	2
Finland	2
Greece	1
Hungary	1
Ireland	2
Italy	2
Latvia	1
Netherlands	2
Slovakia	1
Slovenia	1
Spain	1
Sweden	1
<b>Total</b>	<b>21</b>

### 5.1.2 Assessment of the Health of open source projects

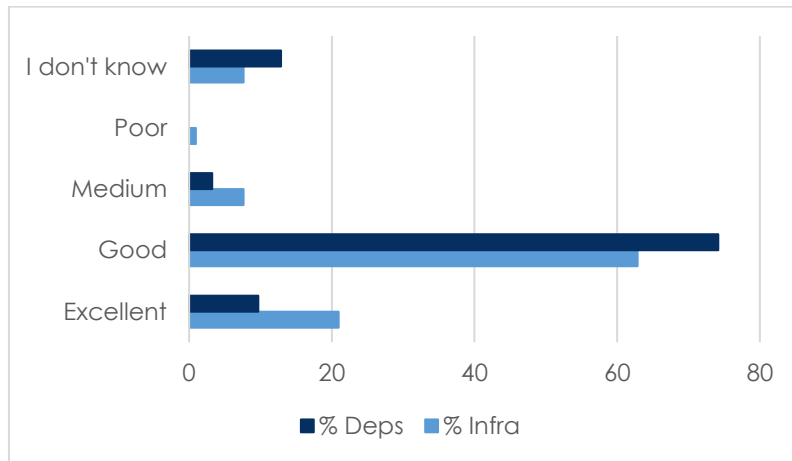
From the responses, 10 public administrations provided information relating to their infrastructure software, and another 10 about software dependencies. The table below shows this information.

Table 6: Responses received from European Public Services

Project Health	Infrastructure software projects	% of Infrastructure software	Software Dependencies	% of Software Dependencies
Excellent	22	21%	3	5%
Good	66	63%	60	89%
Medium	8	8%	1	2%
Poor	1	1%	0	0
I don't know	8	8%	4	6%

Interestingly, across the two categories (infrastructure software and software dependencies), only one open source project was identified as being in a poor state of health, namely LibreOffice. This project, however, was already identified as being critical as mentioned in Section 2.3.

Figure 10: Project Health according to European public services



### 5.1.3 Open source policies

The survey results indicate that open source policies within European Public Services are not currently formalised. Of the 21 respondents, four (25%) have an open source policy, and two organisations said they were working to create a dedicated open source strategy. If we take six, then the figure increases to 29%. The vast majority, 71%, do not have open source policies or a strategy, and open source is handled by individuals with a personal interest in the subject.

Figure 11: Type of open source policy

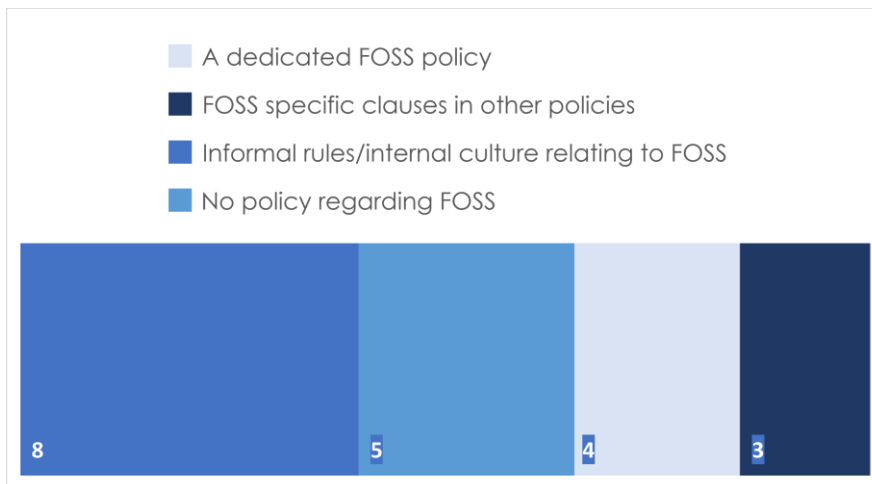
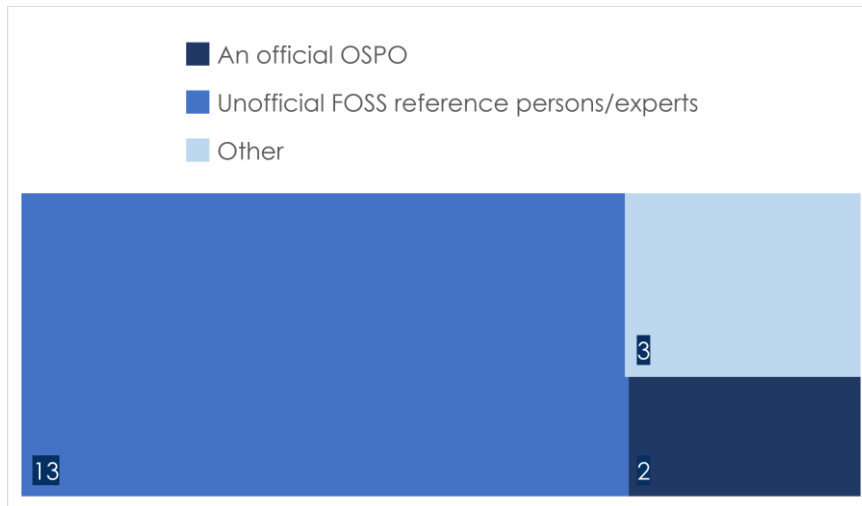


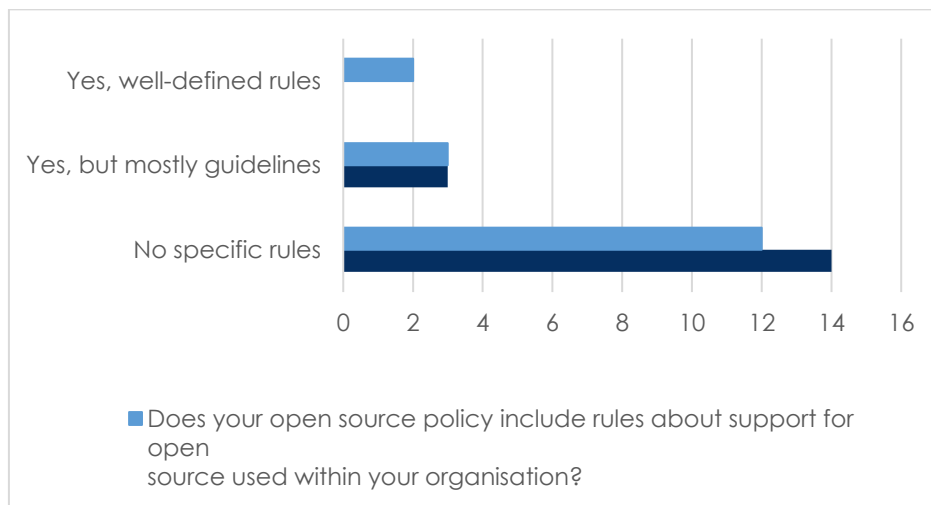


Figure 12: Open source representatives within European public services



Even when open source policies exist, they seem to have loose rules regarding security and open source contributions, as the figures below show.

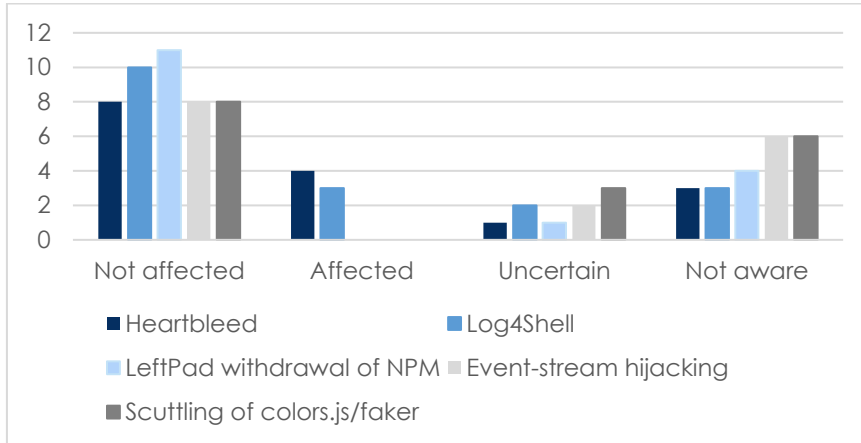
Figure 13: Rules covered through European public services open source policies



### 5.1.4 Open Source and Security

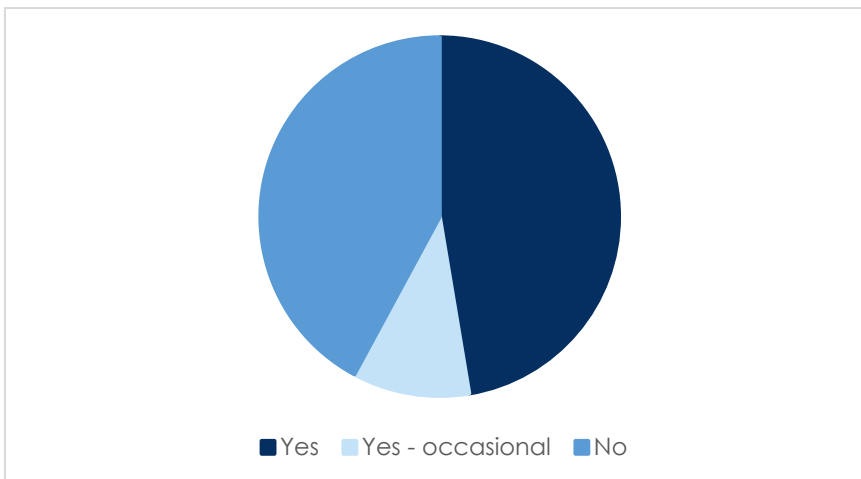
There is a high level of awareness of security issues among respondents, with 11 organisations considering themselves as “very aware”, while 5 “only vaguely” aware of security issues.

Figure 14: Awareness of security issues within FOSSEPS



The survey also found that awareness does not always translate into concrete actions to tackle security problems that may affect the tree of transitive open source dependencies used in their applications. One reason for this is the lack of freely available tools to easily compile a complete list of dependencies (SBOM).

Figure 15: Do FOSSEPSs take actions for dependencies security?



Note: as this was just a survey, we only asked whether you did or not, now how you did each action. So, we do not know some public administrations managed security of dependencies.

### 5.1.5 Support and contribution to open source projects

Given the limited response to the survey, the data gathered does not provide a complete picture when taken at the EU level. Therefore, it is not possible to get a true picture of contributing organisations.

However, the figures do provide some insights:

- Six organisations contribute to open source infrastructure projects they use
- Four contribute to open source dependencies
- Seven contribute to a least one category

Whilst there is still an important room for improvement, this does show that there are no fundamental difficulties for European Public Services to contribute to open source projects.

Regarding the types of contributions, technical (code or non-code contributions) are more frequent than financial. This may be due to different factors, like a common cultural confusion between open source and gratis, but difficulties at procurement level may also be part of the explanation.

Indirect contributions such as through support contracts allocated to individual project contributors, also appear to be rare, especially when it comes to software dependencies.

European Public Services usually obtain Software Support from larger established organisations such as GNU/Linux distributions subscriptions purchased from vendors like Red Hat or Suse.

Figure 16: Type of contribution

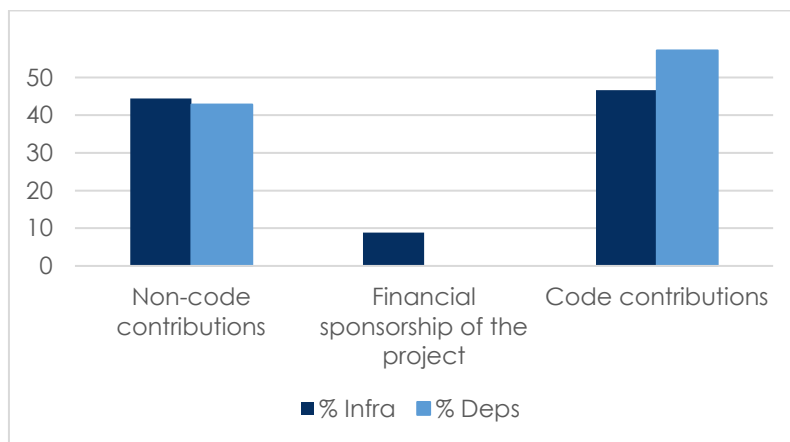
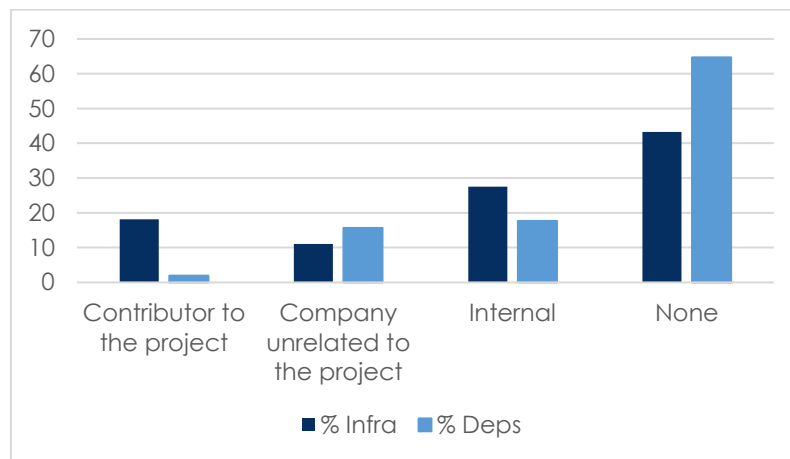


Figure 17: Support to Projects



## 5.2 Results of surveying open source experts

Thirteen open source experts, from two groups (those related to sustainability and project maintainers) were contacted using personalised emails. Their email responses do not lend themselves to tables, but to golden snippets of first-hand, invaluable experiences and wisdom. The topics related to asking about:

- Identifying already known and as yet unknown critical software
- Issues within the open source landscape which lead to criticality
- Suggestions for improving the sustainability of open source software
- Specific issues, problems, concerns of project maintainers
- What project maintainers need as solutions/help to fix this situation

Experts interviewed:

1. Andrew Nesbitt, founder of the Libraries.io and 24 Pull Requests projects
2. APELL (European Open Source Software Business Association), whose members include the German OSBA, the French CNLL and industry associations from Finland, The Netherlands and Portugal amongst others.)
3. Benjamin Nickolls, Executive Director at Open Source Collective and Chief Product Officer at Open Collective, co-founder of SustainOSS
4. Brian Behlendorf, General Manager of the Open Source Security Foundation, and one of the original authors of the Apache HTTP Server
5. Caolán McNamara, Deputy Chairperson on The Document Foundation board and one of the maintainers of LibreOffice
6. Chad Whitacre, Head of Open Source at Sentry, previously founder of Gratipay and co-founder of SustainOSS
7. Daniel Stenberg, founder and maintainer of Curl
8. Dirk-Willem van Gulik, co-founder of the Apache Foundation
9. Gaël Blondelle, VP, Ecosystem Development at the Eclipse Foundation

10. Luis Villa, co-founder of Tidelift
11. Mark Thomasz, core PMC member and one of the maintainers of Apache Tomcat
12. Mark Wielaard, maintainer of bzip2
13. Nick Wellnhofer, maintainer of libxml2

The European Commission is pleased to accurately document these observations in the paragraphs below.

## 5.2.2 Difficulties in identifying critical software

Open source is the foundation and bedrock of all technology. It is no wonder that open source is often referred to as, digital infrastructure<sup>19</sup>.

However, the task of identifying all critical software remains a significant challenge. Especially at the lower levels, e.g., indirect dependencies at the bottom of the dependency tree of software applications, but also system libraries that are outside of these dependency trees<sup>20</sup>. Identification must occur at different levels – publicly available software repository, internal software, and third-party proprietary software.

Regarding internal software, evidently from survey feedback from European Public Services, there is currently a lack of freely available tooling to generate reliable Software Bills of Materials (SBOMs). Some open source tools exist and are making steady progress, like OSS Review Toolkit<sup>21</sup>, but they would need external help in terms of both contributions and visibility.

The main initiative to give a general overview on this topic was the Census II by Harvard Laboratory for Innovation Science (LISH) and the open source Security Foundation (OpenSSF). It relied on data from partner Software Composition Analysis (SCA) companies including Snyk, the Synopsys Cybersecurity Research Center (CyRC), and open source A<sup>22</sup>, a Software Composition Analysis (SCA) company.

To be able to identify the Open Source libraries embedded by proprietary software vendors, it is possible that an “exchange of value would be necessary to allow this identification to happen” as suggests Benjamin Nickolls. A legal incentive such as the Executive Order on Cybersecurity, which mandates the use of SBOMS, may play a positive role in this regard, however, it should be designed in a way not to increase the burden for maintainers, as Luis Villa explains it<sup>23</sup>.

---

<sup>19</sup> <https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/>

<sup>20</sup> See Armin Ronacher “Dependency Risk and Funding”: <https://lucumr.pocoo.org/2022/1/10/dependency-risk-and-funding/>

<sup>21</sup> <https://github.com/oss-review-toolkit/ort>

<sup>22</sup> <https://fossa.com/>

<sup>23</sup> <https://blog.tidelift.com/sboms-are-important-but-they-wont-work-if-we-dont-pay-the-maintainers>

The metrics, methodologies and consolidated publicly available data to identify open source projects with maintenance problems are another area that needs attention too. Many community initiatives like the ChaOSS community, the WG Securing Critical Projects of the OpenSSF, Libraries.io or the more recent ecosystems are working on the topic.

### 5.2.3 Better governance improves sustainability

The open source experts all agreed that good governance leads to increased sustainability of open source software. They also agreed that this area needs strengthening.

#### Governance of Community led vs Vendor led projects

Vendor-driven (or led) open source projects have specific needs that need to be considered. Members of APELL and France's CNLL have been communicating on this topic extensively.<sup>24</sup>

A project when supported by a foundation, by varying degrees, leads to improved governance and sustainability. Although foundations provide support to manage project lifecycles including adequately retiring or transitioning projects, they are not a panacea. Joining a foundation is not always the optimal solution as it involves considerable effort, which may not be worthwhile for very stable projects. Mark Wielaard, maintainer of bzip2, explains: "the overhead of (picking) a "foundation" is probably more than the current energy needed for maintenance of the project."

Open Collective<sup>25</sup> can be a lighter solution in some cases and has shown to bring an efficient solution to some projects such as the Curl project. Curl's maintainer said "Yes, [the creation of an OpenCollective account] did make a significant impact in the way companies can now sponsor the project easily and the level of donations we get. Since we started the OpenCollective it has skyrocketed in comparison to what we got before this was started." He adds however, "Still, all our donations and sponsorships are far from enough to pay even a single developer full-time."

Beyond governance, Brian Behlendorf, General Manager of the open source Security Foundation<sup>26</sup> says that a cultural change within organisations could help open source projects achieve better sustainability. He quotes, "Many open source projects are open, but are actually the work of a single person". This is especially noticeable in the NodeJS ecosystem and its micro-dependency culture. Focusing from the start of a project on having a shared maintenance could help prevent some burnout situations for the maintainers.

### 5.2.4 Providing resources to open source projects

"If you look at the significant security vulnerabilities associated with open source software over recent years, the issue isn't that there were vulnerabilities (there are always going to be vulnerabilities), neither was it the response of the open source project (the vulnerabilities were responded to in a timely manner with fixed versions available very quickly). The issue is the speed at which downstream users

---

<sup>24</sup> <https://cnll.fr/media/Enquete-marches-support-resultats-V5.pdf>

<sup>25</sup> <https://opencollective.com/>

<sup>26</sup> <https://openssf.org/>

are willing or able to update their systems to pick up fixed versions once a vulnerability has been announced. Equifax being a case in point.

[...]

As a Tomcat committer, I regularly see questions about old versions that have not been supported for years. Just this week we received a report of multiple public-facing websites using a version of Tomcat that is at least a decade past its publicly announced end-of-life date and not only that, the sites were configured with an admin interface that used a very weak password. This sort of behaviour is endemic and represents a much bigger security risk to society as a whole.”

Table 7: Table of Recommendations

Area	Issue	Recommendations
Identification of open source projects in need of support	European Public Services don't have a visible and easy to use tool to generate complete SBOMs.	The European Commission, European public services and private companies should contribute financially and technically to the open source tools that allow organisations to have better visibility on their transitive dependencies and generate reliable SBOMs (e.g., OSS Review Toolkit). The European Commission should synchronise efforts among European Public Services on this topic.
Identification of open source projects in need of support	There are no generally accepted metrics and methodology to identify projects in need of help for their maintenance.	The European Commission, European Public Services and private companies should support the community initiatives to define metrics and methodologies to identify open source projects in need of external help for maintenance.
Bringing more resources to open source projects	Many European Public Services have no open source policy and no dedicated process and (group of) persons to implement it.	The European Commission should encourage the other European Public Service to adopt a strategic approach to contribution to open source, in line with the European Commission's open source Software Strategy 2020 – 2023.
Bringing more resources to open source projects	Many European Public Services pay for support to organisations with no direct involvement in the open source projects.	The European Commission should raise awareness on the concept of open source virtuous circle among European Public Services so that when European Public Services invest in support services for open source; this contributes efficiently to the maintenance of the open source projects.
Bringing more resources to open source projects	Existing guidelines for creating an OSPO may not be fully suitable for European Public Services.	The European Commission and other European Public Services should work on including European Public Services specific guidelines in the best practices for OSPO regarding contribution (e.g., Good Governance Initiative of OSPO.zone).
Bringing more resources to open source projects	Many European Public Services use older versions or deprecated features of open source components, creating	The European Commission, European Public Services and European companies should better update the open source they use and initiate and fund specific Long Time Support versions to

Area	Issue	Recommendations
source projects	extra maintenance work for the projects.	accommodate contexts incompatible with the release cycles of the projects.
Bridging the topic of sustainability and cybersecurity	Cybersecurity and open source sustainability are closely related but require different expertise.	The European Commission should initiate connections with EU organisations dedicated to cybersecurity to work with them on the topic of Software Supply Chain security.
Studying open source	Software development dependencies, infrastructure software and user facing applications require specific methodologies to be identified and analysed.	For its further studies, the European Commission should focus on a specific type of software.  Note to Deloitte: Related to chapter 2
Studying open source	There are many different types of organisations involved in the present and future of open source in FOSSEPS, that may not bring the same type of help during studies or implementations of solutions.	Create a structured landscape of open source-related organisations at EU and global level, considering their specific natures. That would include organisations which pool software development for European Public Services or national agencies for digitalisation of public services.

### 5.3 Analysing critical open source Projects

The timeframe of this study allowed a detailed analysis of only 5 of the 30 initially identified critical software projects. For each of these five projects the study team calculated the "Bus Factor"<sup>27</sup> according to the method presented by the ChaOSS community<sup>28</sup> (which defines the Bus Factor as the smallest number of people that make 50% of contributions).

However, the study team does not say that its calculations are definitive, or that the right method was used. The European Commission does not wish to create alarm on any project. Any analysis here is of the study team only and not a view of the European Commission.

All the projects analysed below but one, have a Bus Factor of 1, which is the lowest possible for a not-quite-dead or barely-alive project.

Following this, we have asked the maintainers of each of these five respective projects about the needs of the project.

- Apache Tomcat
- Libxml2
- Curl
- LibreOffice
- M2Crypto

<sup>27</sup> [https://en.wikipedia.org/wiki/Bus\\_factor](https://en.wikipedia.org/wiki/Bus_factor) The bus factor is a measurement of the risk resulting from information and capabilities not being shared among team members, derived from the phrase "in case they get hit by a bus"

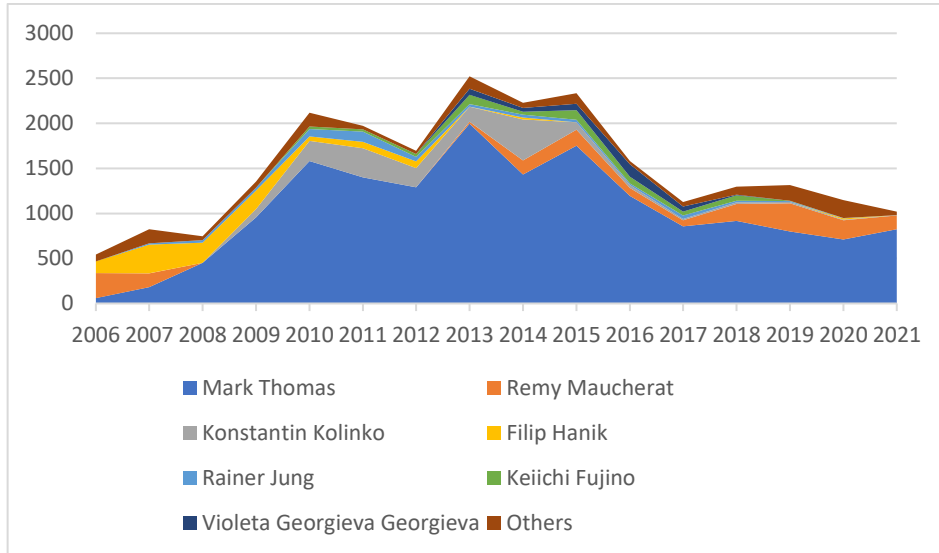
<sup>28</sup> <https://chaoss.community/metric-bus-factor/>



### 5.3.1 Apache Tomcat

Despite its ubiquity, the Apache Tomcat project has relied on a small team of core committers for more than 10 years.

Figure 18: Volume of contributions to Apache Tomcat by the main committers



Upon being asked about the project’s maintenance needs, Apache Tomcat’s maintainer underlined the following aspects.

There are no specific gaps in Tomcat’s maintenance, even if additional contributions are always welcome. Regarding security, a review/audit would be the most productive only if it is a manual audit by one or more skilled security researchers. A simple output from a static analysis tool or similar would not be useful.

**Study team observation:** Apache Tomcat’s needs resonate with those of other mature software projects. They have a small but stable and highly committed core team, and so incredible work, to keep Tomcat running. Of course, they would like more contributors, because this project is likely to continue to be relied upon by being needed by the world for a long time. There is obviously a training period before someone can become as experienced as the core team. This can present a hurdle for newcomers and the project team.

### 5.3.2 Libxml2

Libxml2 is widely used. However, since it is a low-level library, it can often be overlooked by inventories based on metadata of package managers for higher level libraries.

Figure 19: Volume of contributions to Libxml2 by the main committers

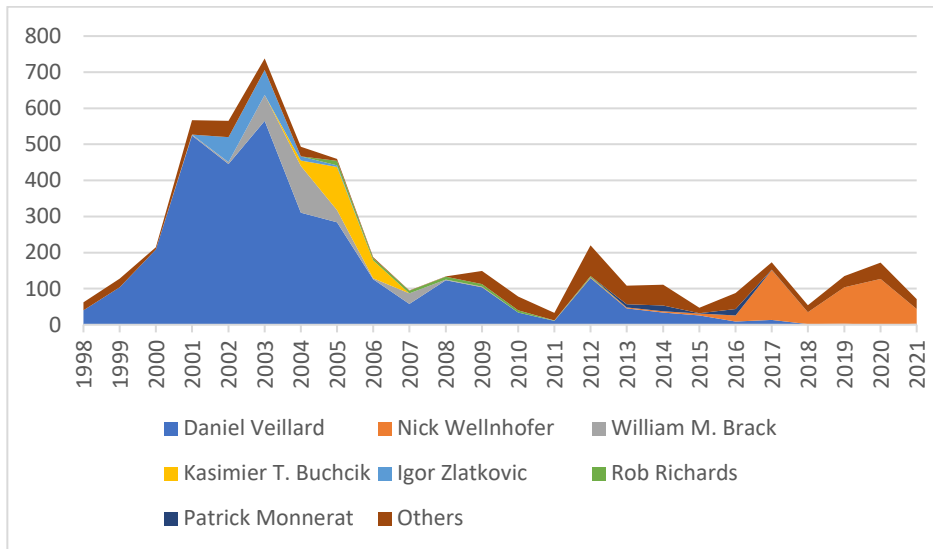
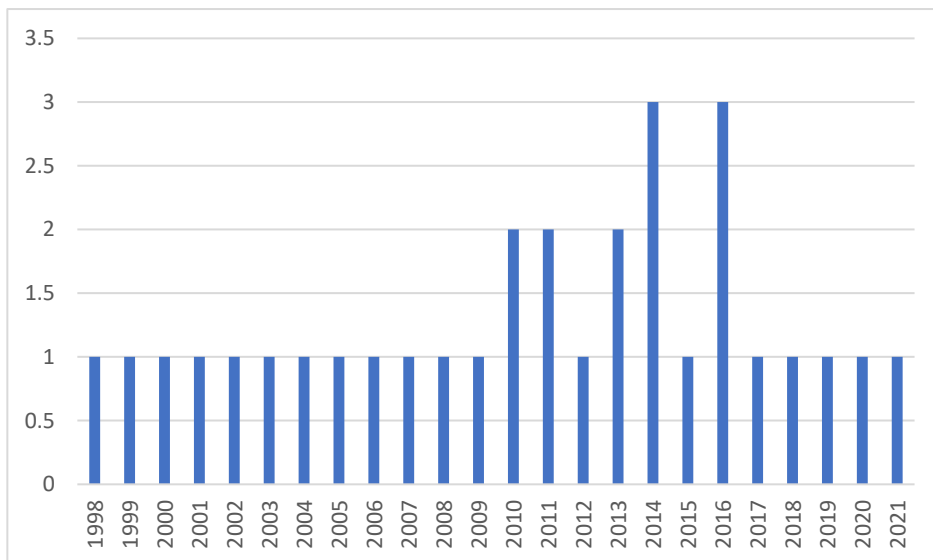


Figure 20: Bus Factor for libXML2



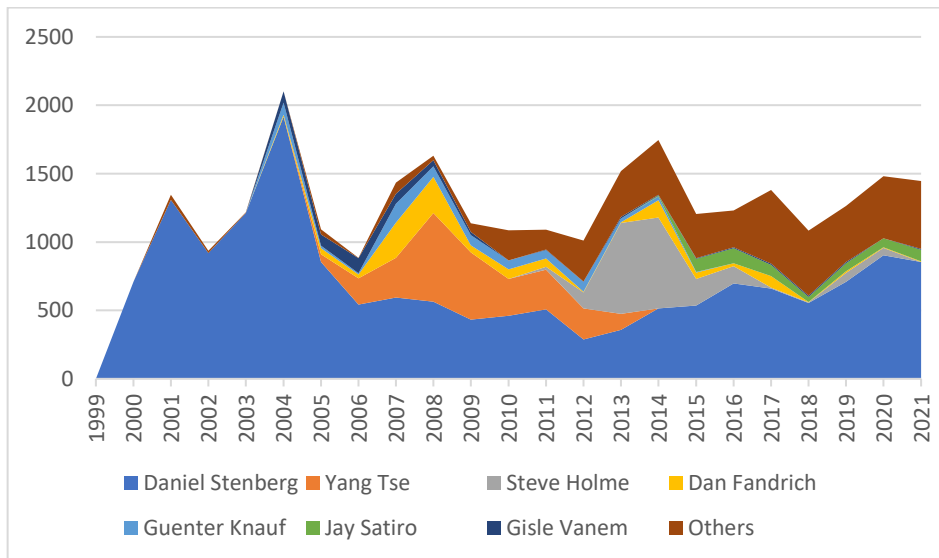
Libxml2's current maintainer has outlined the below needs in relation to the project's maintenance:

- 1 **Minimal team:** A core group of active maintainers are required, ideally three or more people.
- 2 **Technical contributions:** Specially to fix all issues that emerge when fuzzing (refer to point 3 below).
- 3 **Security:** Regarding security, fuzz testing is required as this would be more effective as opposed to code audits. The project does have 'fuzzers' focusing on some core features; however code coverage is still low.
- 4 **Finance:** To accomplish these goals, funding, and more direct involvement of larger organizations, both public and private, would be essential.

### 5.3.3 Curl

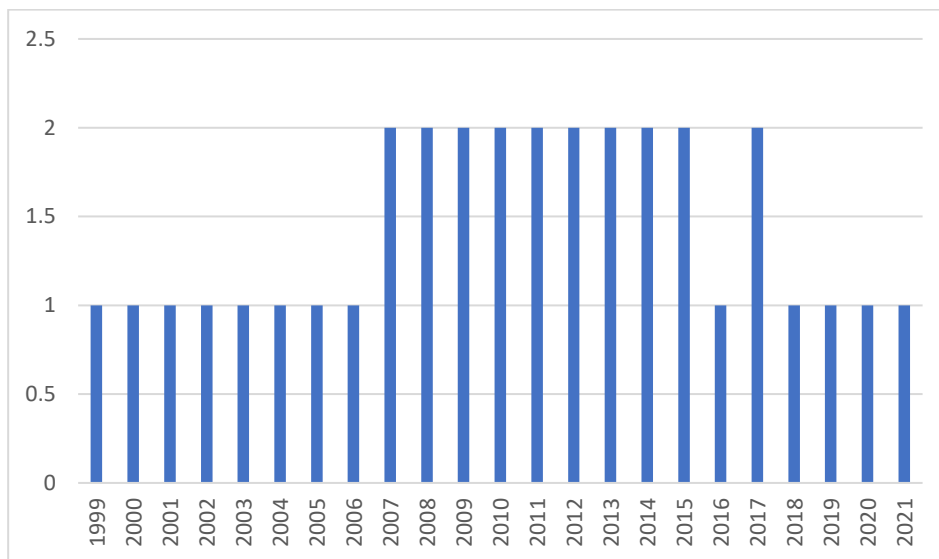
Despite having a maintainer who is very involved in the topic of open source sustainability<sup>29</sup>, Curl is still in a relatively vulnerable position with a Bus Factor of 1 for the past four years.

Figure 21: Volume of contributions to Curl by the main committers



<sup>29</sup> See "Enforcing the pyramid of Open Source" <https://daniel.haxx.se/blog/2022/01/17/enforcing-the-pyramid-of-open-source/> and "Free Apple support" <https://daniel.haxx.se/blog/2021/11/18/free-apple-support/>

Figure 22: Bus Factor for Curl



Curl's maintainer has identified the following needs for the project:

1. **Finance:** The project needs to acquire a certain level of paying customers for his employer to be able to sustain his salary to work on the project on a full-time basis.
2. **Attract and retain talent:** In general, the project needs to attract and retain contributors to support developing new features, expanding the range of testing execution and completing the documentation.
3. **Improving the diversity:** of the contributor base to benefit the project and its users by having a more complete and varied participation to the project.

### 5.3.4 LibreOffice

LibreOffice is one of the most popular and complex projects with a code base in its core repository of more than 80,000 files and more than 7 million of lines of code. In this regard, a Bus Factor of 7 is not as high as might initially seem.

Figure 23: Volume of contributions to LibreOffice by the main committers

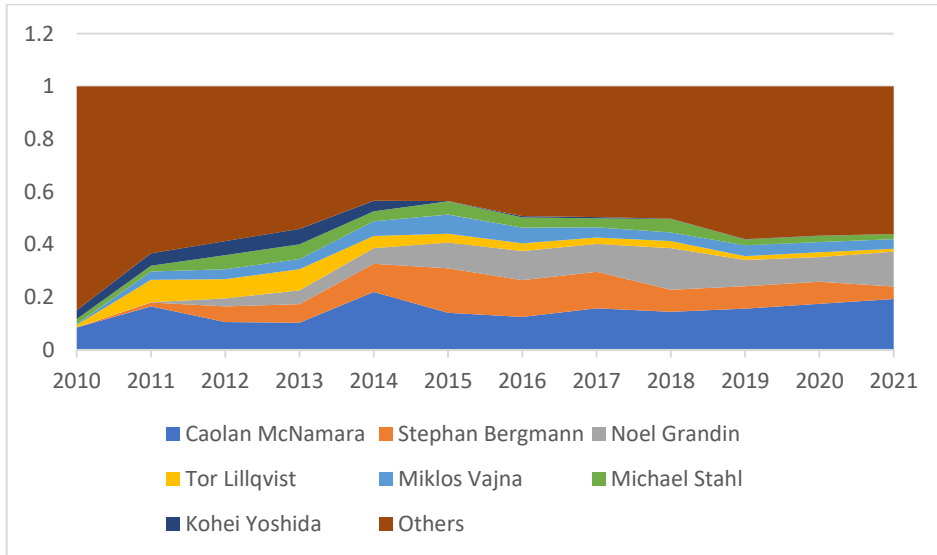
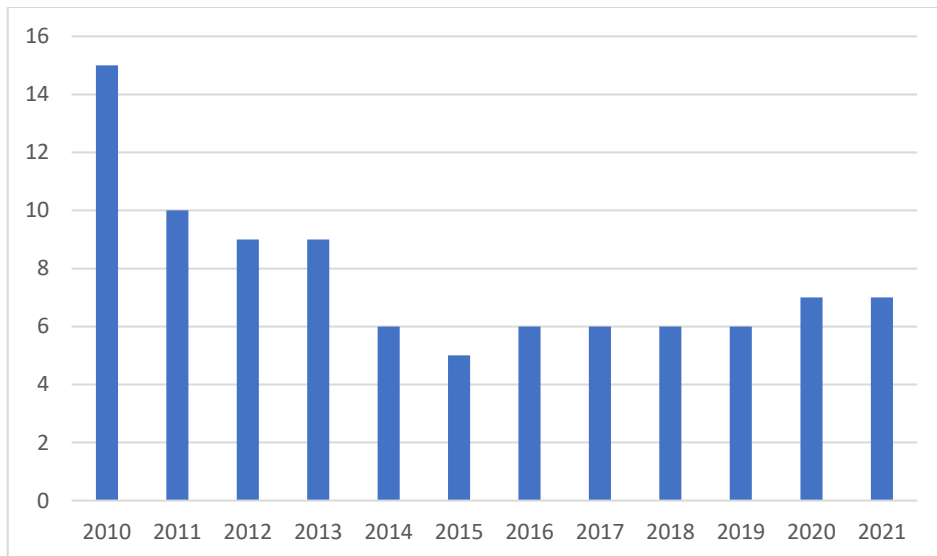


Figure 24: Bus Factor for LibreOffice



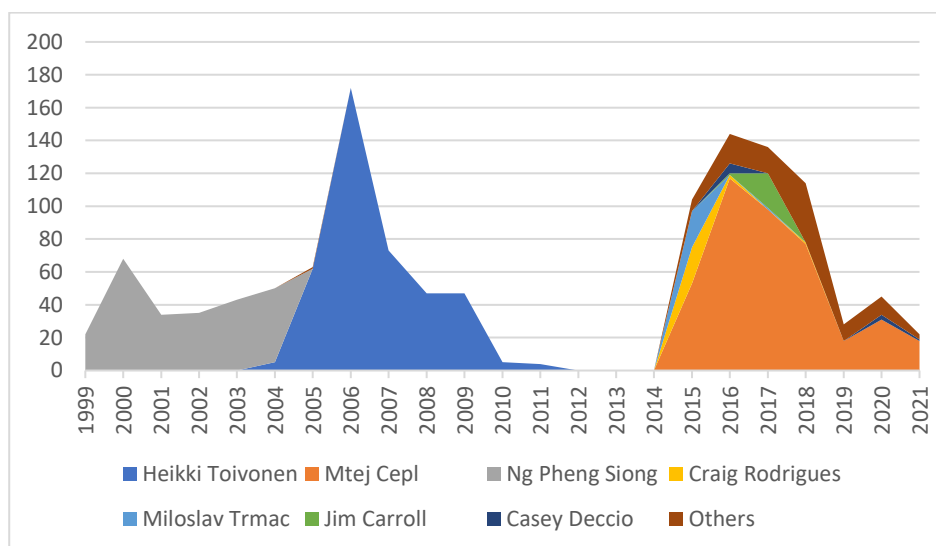
According to one of its core developers, the following are the needs of LibreOffice:

1. **Attract new Talent:** To attract new contributors in the medium and longer term.
2. **Increased Funds:** Having public bodies getting more involved, whether it is through direct involvement or through pooling organisations, would lead to greater financial contribution.
3. **Security:** Regarding security, whereas it has been acknowledged that notifications from national security agencies are useful and appreciated, contributions to fix detected problems would be even more beneficial.

### 5.3.5 M2Crypto

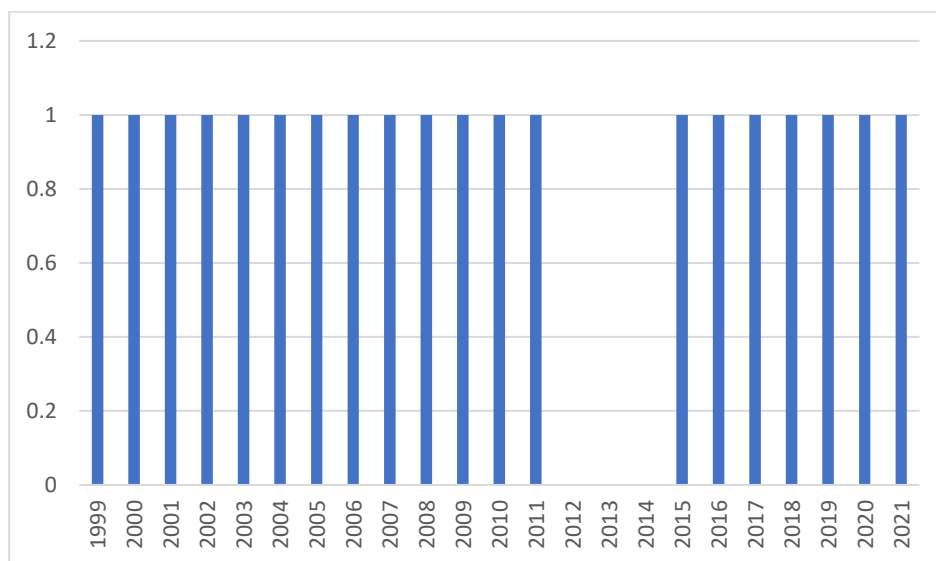
The M2Crypto project had been abandoned for several years before being revived by a single maintainer to support its significant number of users. Attesting to its popularity are the current monthly download figures of approximately 246,000, making it one of the top 5% not-quite-dead or barely-alive packages on PyPI. Additionally, there are 116 other PyPI packages depending on it.<sup>30</sup>

Figure 25: Volume of contributions to M2Crypto by the main committers



<sup>30</sup> <https://deps.dev/py/m2crypto/0.38.0/dependents>

Figure 26: Bus Factor for M2Crypto



Although its maintenance needs have not been clearly defined yet, M2Crypto’s maintainer has already identified the **contribution of a cryptography-engineer as desirable**. The migration to a better maintained alternative Python package would not be straightforward.

### 5.3.6 Summary

The following table is a summary of needs as declared by the projects with a rating of 0 to 5, with 0 being the least important and 5 being the most important to that particular project.

Table 8: Summary of Project Needs

Project Need	Apache Tomcat	Curl	Libxml2	M2Crypto	LibreOffice	Totals
Funding	0	5	5	0	5	15
Security – Audit	3	0	0	0	5	8
Security – Fuzzing	0	0	5	0	0	5
Contributors	2	3	5	0	5	15
Crypto expertise	0	0	0	5	0	5
Clients for service company	0	5	0	0	0	5
Total scores	5	13	15	5	15	

### Conclusions

- Libxml2 and LibreOffice need the most assistance followed by Curl.
- The biggest needs are funding and contributors, followed by a security audit

## 6. Exploring and proposing solutions

This chapter explores and proposes solutions to shift identified critical open source software to a non-critical stage.

The proposed solutions arise out of individual discussions with five selected project communities, mentioned in Chapter 5. There are several solutions, which can be extrapolated to create a set of common solutions for similarly vulnerable communities. These are shown later in the document. An attempt has also been made (with necessary assumptions) to estimate the cost of implementing such solutions. This could be further investigated at a later stage.

### 6.1 Project Needs

The table below shows the five software communities the project team spoke with, their expressed needs and the solutions proposed (by the study team) to satisfy the project's needs.

*Table 9: Project Needs and Proposed Solutions*

Software project	Needs Expressed	Proposed solutions
Apache Tomcat	<ul style="list-style-type: none"> <li>Manual security review/audit by one or more skilled security researchers. A simple output from a static analysis tool or similar would not be useful.</li> <li>We have funds and contributors, but additional funds and contributors are always welcome.</li> </ul>	<ul style="list-style-type: none"> <li>Workshop in partnership with ENISA.</li> <li>Specific assistance by skilled security expert.</li> <li>Internship with EU programs.</li> <li>Specific assistance by open source sustainability expert.</li> </ul>
LibXML2	<ul style="list-style-type: none"> <li>Increasing the number of maintainers.</li> <li>Technical contributions, especially to fix all issues that emerge when fuzzing. Improving fuzzing coverage.</li> <li>Increasing funding and more direct involvement of larger organizations, both public and private, would be essential.</li> </ul>	<ul style="list-style-type: none"> <li>Workshop in partnership with ENISA.</li> <li>Specific assistance by skilled security expert.</li> <li>Award a grant for the core maintainer.</li> </ul>
Curl	<ul style="list-style-type: none"> <li>Acquire a certain level of paying customers for his employer to be able to sustain the salary of the maintainer.</li> <li>Attract and retain contributors to support developing new features, expanding the range of</li> </ul>	<ul style="list-style-type: none"> <li>Internship with EU programs.</li> <li>Award a grant for the core maintainer.</li> <li>Legislative evolutions in favour of SMEs.</li> </ul>



Software project	Needs Expressed	Proposed solutions
	testing execution and completing the documentation. <ul style="list-style-type: none"> <li>Improving the diversity of the contributor base (not a need but a strong will).</li> </ul>	
LibreOffice	<ul style="list-style-type: none"> <li>New contributors in the medium and longer term.</li> <li>Having public bodies getting more involved, whether it is through direct involvement or through pooling organisations.</li> <li>Security contributions from national security agencies including contributions to fix detected problems.</li> </ul>	<ul style="list-style-type: none"> <li>Workshop in partnership with ENISA.</li> <li>Contribution workshop with European Public Services.</li> </ul>
M2Crypto	<ul style="list-style-type: none"> <li>Finding a solution for the mid/long term maintenance.</li> <li>Technical expertise in cryptography.</li> </ul>	<ul style="list-style-type: none"> <li>Specific assistance by skilled security expert.</li> <li>Workshop in partnership with ENISA.</li> <li>Award a grant for the core maintainer.</li> </ul>

## 6.2 Proposed solutions

This section describes the study team’s proposed solutions in further detail.

### 6.2.1 Security workshop in partnership with ENISA

#### Description

The study team feels that many security issues could be resolved by a workshop organised in partnership with ENISA<sup>31</sup> involving security experts from European National Security Agencies on the topic of Software Supply Chain (in the broader sense) security and applied to the example of the selected open source project.

The precise topic of the workshop would depend on the needs of the selected open source project. The expected output would be three-fold:

1. Have a concrete, actionable plan for the specific project to improve the security of its millions of users.
2. Provide the ENISA and the different European National Security Agencies with a concrete opportunity to coordinate and advance their work on the topic of software supply chain security.

---

<sup>31</sup> <https://www.enisa.europa.eu/>

3. Create a connection between the stakeholders of the two key aspects of maintaining open source: security and sustainability.

#### **Associated costs**

- Logistic costs for the organisation and coordination of the workshop.
- Travel expense for the participants.
- Financial compensation for the time spent by the maintainers of the open source project.

#### **Projects which could benefit**

- Apache Tomcat: The workshop should focus on code auditing.
- Libxml2: The workshop should focus on fuzzing.
- M2Crypto: The workshop should focus on cryptography.
- LibreOffice: The workshop should focus on fixing, not just detecting security vulnerabilities.

### **6.2.2 Specific assistance by a skilled security expert**

#### **Description**

In the case that coordinating a workshop is not possible or not sufficient, funding a skilled security researcher to assist the project would help the selected open source project to cover its security needs.

#### **Associated costs**

- Cost of the security expert's services.
- Compensation for the time spent by the maintainers of the open source project.

#### **Projects which could benefit**

- Apache Tomcat: The expert's assistance should focus on code auditing.
- Libxml2: The expert's assistance should focus on fuzzing.
- M2Crypto: The expert's assistance should focus on cryptography.

### 6.2.3 Specific assistance by open source sustainability expert

#### Description

Open source projects have different levels of maturity in terms of organisation and governance, which directly impacts their ability to receive the contribution they would need. An open source sustainability expert could help them evolve the organisation of the project to make the project more accessible to new contributors and share the work of existing contributors.

#### Associated costs

- Cost of the open source sustainability expert's services.
- Compensation for the time spent by the maintainers of the open source project.

#### Projects which could benefit

- Libxml2
- M2Crypto
- Apache Tomcat

### 6.2.4 Internship with EU Diversity programs

#### Description

There are several opportunities to engage interns in open source. Financing such internships in partnership with EU programs dedicated to diversity in technical contexts<sup>32</sup> would both serve the purpose of such programs and help the open source projects get more contributions and more diversity in their contributor base.

#### Associated costs

- Costs of coordination between the different organisations.
- Stipends for interns.
- Compensation for the time spent by the maintainers of the open source project for mentoring the intern.
- Compensation for expenses.

---

<sup>32</sup> For example, programs similar to [https://ec.europa.eu/info/news/eu-launch-new-support-scheme-women-deep-tech-and-call-mentors-2021-mar-08\\_en](https://ec.europa.eu/info/news/eu-launch-new-support-scheme-women-deep-tech-and-call-mentors-2021-mar-08_en)

### **Projects which could benefit**

- cURL: Daniel Stenberg is willing to be a mentor.
- Apache Tomcat: If a committer is willing to be a mentor.

## **6.2.5 Contribution workshop with European Public Services**

### **Description**

In collaboration with other FOSSEPS initiatives a workshop could be organised with representatives of European Public Services having decision-making authority, to discuss how they could potentially contribute better to open source projects, for a specific critical open source project.

### **Associated costs**

- Logistic costs for the organisation and coordination of the workshop.
- Compensation for the time spent by the maintainers of the open source project.

### **Projects which could benefit**

- LibreOffice

## **6.2.6 Award a grant to the core maintainer**

### **Description**

When the maintenance of the open source project relies on the personal financial investment of the maintainer, a grant can help alleviate the pressure. The grant could be funded directly or in collaboration with one of the funding organisations identified during a previous study by the European Commission.

### **Associated costs**

- Amount of the grant.

### **Projects which could benefit**

- Curl
- Libxml2
- M2Crypto

## 6.3 Summary

The primary needs and related solutions can be listed as follows:

- Funding
- Security audits
- Additional contributors
- Diversity

Funding remains the primary need for most open source projects. One of the projects (Curl) has described its first need as being the ability to find sufficient paying users to sustain the salary of its maintainer. This echoes the requests of many open source associations of SMEs in Europe, asking for an evolution of procurement policies in both public and private sectors that would facilitate these companies to directly access categories of customers which are currently reserved to larger organisations.

It is worth noting that many of the proposed solutions above, are related to security. This shows how important security is to the ongoing sustainability of an open source project.

Beyond these, it is essential to encourage broader evolutions that allow better collaboration between European public services and the open source projects they use, as detailed in the recommendations in the next chapter.

## 7. Conclusion

This has been a challenging, but very important study for the long-term sustainability of critical open source software. The study team was able to contact a large number of open source stakeholders in European public services and relevant experts from the open source industry.

Amongst other conclusions, one is crystal clear – which is, that despite its complexity, the need to identify critical software is crucial, not only for European public services, the specific open source projects, but also for the wider world, because the global digital infrastructure depends on relatively unknown and unacknowledged open source software project.

This section aims to group suggestions, lessons, and conclusions in four areas:

1. Process related
2. From open source experts
3. The study itself
4. Specific recommendations

### 7.1 Process related

Although a standard and high-level process can be defined to identify software considered “critical”, the study shows that the process to evaluate the “criticality” of a software, is complex. It is not a one-size-fits-all solution, but depends on the type of software in question.

The feedback gathered from European Public Services shows that some organisations put in significant effort to gather data with respect to open source software used and their level of criticality. Others have without doubt, benefitted by increased awareness of the whole issue of critical software.

Equally, the lack of critical software identified by European public services demonstrates the challenges those who are not experts, in this arena. In the future. We would need to find different methods, as surveys will not elicit previously unknown critical software.

Due to this and owing to the various types of problems different projects can face, the study team suggest that future studies could focus on a *specific type of software*. This could allow a deeper analysis of the problems impacting these specific types of projects.

### 7.2 Open source experts

Discussions with select open source experts and communities focused on the main needs of open source projects. As can be expected, funding is the most important need highlighted by most experts, followed by the need for further contributors. Less important but still needed is support in Security (Audit and Fuzzing), Crypto expertise, and attracting more clients towards service companies.

The experts also shared their views and proposed a number of solutions to address the study’s main objective, i.e., shift identified critical open source software to a non-critical stage. Various proposals, many of which are related to security, qualify

as a common solution for a group of similarly exposed communities. Most proposed solutions are of a one-time nature; therefore, these should be sustained through broader and longer-term initiatives, allowing for better collaboration between European Public Services and open source projects they use.

### **7.3 The Study**

There is a large body of open source projects with various maintenance problems, both from a technical and legal perspective. This is a reality faced by open source projects irrespective of their popularity and/or contributor base.

The findings from the study show there is increasing awareness of the *concept of critical software*. However, although there is a certain level of contribution to open source by European public services, it is evident that more is required.

An analysis of the study highlights a number of key issues.

1. Identification of open source projects in need of support
2. Bringing more resources to open source projects
3. Bridging the topic of sustainability and cybersecurity
4. Studying open source

Most of these recommendations include initiatives which can be initiated by the European Commission in collaboration with European Public Services and/or private companies to help start addressing these issues. Recommended collaborations range from financial and technical contributions, encouraging the adoption of strategic contribution approaches and having updated versions of the open source they use, to liaising with them on the topic of Software Supply Chain security.

## 7.4 Specific recommendations

This section ends with recommendations for the European Commission (EC).

1. Further studies should focus on specific types/groups of software, and only contact software and sustainability experts, rather than European Public Services.
2. The EC should contribute financially and technically to the open source tools that allow organisations to have better visibility on their transitive dependencies and generate reliable SBOMs (e.g., OSS Review Toolkit). The EC should also encourage European public services to use and contribute to these open source tools.
3. The EC should support *community initiatives* to define metrics and methodologies e.g., Bus Factor, to identify open source projects in need of external help for maintenance.
4. The EC should encourage the other European Public Service to adopt a strategic approach to contribution to open source, in line with the EC's open source Software Strategy 2020 – 2023. The EC should work on including European public services specific guidelines in the best practices for OSPO regarding contribution (e.g. Good Governance Initiative of [www.ospo.zone](http://www.ospo.zone)).
5. The EC should raise awareness on the concept of open source virtuous circle among European public services so that when public administrations invest in support services for open source; this contributes efficiently to the maintenance of the open source projects. The EC should encourage European public services and European companies to better update the open source they use and initiate and fund specific Long Time Support versions to accommodate contexts incompatible with the release cycles of the projects.
6. Initiate connections with EU organisations dedicated to cybersecurity to work with them on the topic of Software Supply Chain security.



## 8. APPENDIX A List of Files

### Section 2

#### Task 2 A - open source projects

- *Task\_2\_A\_open source\_projects.ods*

#### Task 2 B1 - FOSSEPS contacts

- *Task\_2\_B\_1\_FOSSEPS\_contacts.csv*
- *Task\_2\_B\_1\_FOSSEPS\_contacts.sqlite*
- *Task\_2\_B\_1\_FOSSEPS\_contacts.ods*

#### Task 2 B2 to B8 - contacts

- *Task\_2\_B\_2-8\_contacts.ods*

### Section 3

A compressed file on Joinup contains the following files.

- *FOSSEPS Survey - [1] - Email template (Not contacted).pdf*
- *FOSSEPS Survey - [2] - Email template (Contacted - No Data).pdf*
- *FOSSEPS Survey - [3] - Survey Specimen.pdf*
- *FOSSEPS Survey - [4] - FOSSEPS\_Survey\_Critical\_Apps\_infra\_template.ods*
- *FOSSEPS Survey - [5] - FOSSEPS\_Survey\_Critical\_dependencies\_template.ods*
- *FOSSEPS Survey - [6] - FOSSEPS Survey Guide V1.0.pdf*
- *FOSSEPS Survey - [7] - Task\_4\_2\_Master\_Contacts\_Tracker\_20220404 - Consolidated*
- *FOSS Experts Survey - [1] - E-mail\_Apache.pdf*
- *FOSS Experts Survey - [2] - E-mail\_APELL.pdf*

## 9. APPENDIX B Projects by status

Table 10: Projects by Status

Component	Status	URL	Category	Governance Type	Bus Factor in 2021
libxml2	Data received	<a href="http://www.xmlsoft.org/">http://www.xmlsoft.org/</a>	Low level library	Community driven – Foundation	1
m2crypto	Data received	<a href="https://gitlab.com/m2crypto/m2crypto">https://gitlab.com/m2crypto/m2crypto</a>	High level Library	Community driven – informal	1
LibreOffice	Data received	<a href="https://libreoffice.org">https://libreoffice.org</a>	Key application	Community driven – Foundation	7
Apache Tomcat	Data received	<a href="https://tomcat.apache.org/">https://tomcat.apache.org/</a>	Infrastructure	Community driven – Foundation	1
Curl	Data received	<a href="https://curl.se/">https://curl.se/</a>	Low level library	Community driven – informal	1
Thunderbird	Contact to identify	<a href="https://www.thunderbird.net">https://www.thunderbird.net</a>	Key application	Community driven – Foundation	5
CouchDB	Contact to identify	<a href="https://couchdb.apache.org/">https://couchdb.apache.org/</a>	Infrastructure	Community driven – Foundation	3
cyrus-sasl	Contact to identify	<a href="https://www.cyrusimap.org/sasl/">https://www.cyrusimap.org/sasl/</a>	High level Library	Community driven – informal	
VLC	Contact to identify	<a href="https://www.videolan.org/">https://www.videolan.org/</a>	Key application	Community driven – Foundation	
libgcrypt	Contact to identify	<a href="https://www.gnupg.org/related_software/libgcrypt/">https://www.gnupg.org/related_software/libgcrypt/</a>	Low level library	Community driven – informal	
GnuPG	Contact to identify	<a href="https://www.gnupg.org/">https://www.gnupg.org/</a>	Tool	Community driven – informal	
hunspell	Contact to identify	<a href="https://github.com/hunspell/hunspell/">https://github.com/hunspell/hunspell/</a>	High level Library	Community driven – informal	
SVN	Deprecated	<a href="https://subversion.apache.org/">https://subversion.apache.org/</a>	Tool	Community driven – Foundation	
Nspr & nss	Deprecated	<a href="https://n-2.org/">https://n-2.org/</a>	High level Library	Community driven – informal	
Gecko SDK	Deprecated	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	High level Library	Community driven –	

Component	Status	URL	Category	Governance Type	Bus Factor in 2021
				Foundation	
Liftweb	Deprecated	<a href="https://www.liftweb.net/">https://www.liftweb.net/</a>	High level Library	Community driven – informal	
XULRUNNER	Deprecated	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	High level Library	Community driven – Foundation	
PostgreSQL	No Health problem	<a href="https://www.postgresql.org/">https://www.postgresql.org/</a>	Infrastructure	Community driven – informal	
OpenLDAP	No Health problem	<a href="https://www.openldap.org/">https://www.openldap.org/</a>	Infrastructure	Community driven – Foundation	2
doorkeeper-i18n	No Health problem	<a href="https://github.com/doorkeeper-gem/doorkeeper-i18n">https://github.com/doorkeeper-gem/doorkeeper-i18n</a>	High level Library	Community driven – informal	
Qt	No Health problem	<a href="https://www.qt.io/">https://www.qt.io/</a>	Low level library	Vendor driven	
Fog::Local	No Health problem	<a href="https://github.com/fog/fog-local">https://github.com/fog/fog-local</a>	High level Library	Community driven – informal	
Libeot	Waiting for reply	<a href="https://github.com/umawizard/libeot">https://github.com/umawizard/libeot</a>	High level Library	Community driven – informal	
Debian	Waiting for reply	<a href="https://www.debian.org/">https://www.debian.org/</a>	Infrastructure	Community driven – Foundation	
Firefox	Waiting for reply	<a href="https://www.mozilla.org">https://www.mozilla.org</a>	Key application	Community driven – Foundation	
OpenSSH	Waiting for reply	<a href="https://www.openssh.com/">https://www.openssh.com/</a>	Infrastructure	Community driven – Foundation	2
FFmpeg	Waiting for reply	<a href="https://ffmpeg.org/">https://ffmpeg.org/</a>	Low level library	Community driven – informal	2
Hackney	Waiting for reply	<a href="https://github.com/benoitc/hackney">https://github.com/benoitc/hackney</a>	High level Library	Community driven – informal	1
redland, raptor and rasqal	Waiting for reply	<a href="https://librdf.org/">https://librdf.org/</a>	High level Library	Community driven – informal	1
bzip2	Waiting for reply	<a href="https://sourceware.org/bzip2/bzip2-howto/using-bzip2.html">https://sourceware.org/bzip2/bzip2-howto/using-bzip2.html</a>	Low level library	Community driven – informal	0

Table 11: Open source Infrastructure Health Evaluation

Name	Evaluation	URL
.NET framework	Excellent: 1	<a href="https://dotnet.microsoft.com">https://dotnet.microsoft.com</a>
AbuseIO	Excellent: 1	<a href="https://abuse.io">https://abuse.io</a>
Adempiere/Idempiere ERP	Medium: 1	<a href="https://www.idempiere.org">https://www.idempiere.org</a>
Ansible	Good: 1	<a href="https://www.ansible.com">https://www.ansible.com</a>
Apache ActiveMQ	Good: 1	<a href="https://activemq.apache.org">https://activemq.apache.org</a>
Apache HTTPD	Good: 2 Excellent: 1	<a href="https://httpd.apache.org">https://httpd.apache.org</a>
Apache Tomcat	Excellent: 1	<a href="https://tomcat.apache.org">https://tomcat.apache.org</a>
BIND	Good: 1	<a href="https://www.isc.org/bind">https://www.isc.org/bind</a>
Cacti	Good: 1	<a href="https://www.cacti.net">https://www.cacti.net</a>
Centos	Good: 1	<a href="https://www.centos.org">https://www.centos.org</a>
DIGGSweden/dataportal-dev	Good: 1	<a href="https://github.com/DIGGSweden/dataportal-dev">https://github.com/DIGGSweden/dataportal-dev</a>
DIGGSweden/dataportal-web	Good: 1	<a href="https://github.com/DIGGSweden/dataportal-web">https://github.com/DIGGSweden/dataportal-web</a>
Debian	Excellent: 1 Good: 1	<a href="https://www.debian.org">https://www.debian.org</a>
Docker	Good: 2	<a href="https://www.docker.com">https://www.docker.com</a>
Drupal	Good: 1	<a href="https://www.drupal.org">https://www.drupal.org</a>
Dspace	Excellent: 1	<a href="https://duraspace.org/dspace">https://duraspace.org/dspace</a>
EDB Database	Excellent: 1	<a href="https://www.enterprisedb.com">https://www.enterprisedb.com</a>
ElasticSearch	Good: 1	<a href="https://www.elastic.co">https://www.elastic.co</a>
Fedora Commons	Good: 1	<a href="https://www.lyrasis.org/programs/Pages/Fedora.aspx">https://www.lyrasis.org/programs/Pages/Fedora.aspx</a>
Free Radius	Good: 1	<a href="https://freeradius.org">https://freeradius.org</a>
Freepbx / asterisk	Good: 1	<a href="https://www.asterisk.org">https://www.asterisk.org</a>
Fuseki	I don't know: 1	<a href="https://jena.apache.org/documentation/fuseki2">https://jena.apache.org/documentation/fuseki2</a>
GeoServer	Good: 3	<a href="https://geoserver.org">https://geoserver.org</a>
GlobaLeaks	Medium: 1	<a href="https://www.globaleaks.org/it">https://www.globaleaks.org/it</a>
Gluster	Good: 1	<a href="https://www.gluster.org">https://www.gluster.org</a>
HAProxy	I don't know: 1	<a href="https://www.haproxy.org">https://www.haproxy.org</a>
Ice Tea Java	Good: 1	<a href="https://openjdk.java.net/projects/icedtea">https://openjdk.java.net/projects/icedtea</a>
Icinga	Good: 1	<a href="https://icinga.com">https://icinga.com</a>
JBoss 7.1.1	Good: 1	<a href="http://www.jboss.org">http://www.jboss.org</a>
Jenkins	Excellent: 1	<a href="https://www.jenkins.io">https://www.jenkins.io</a>
Keycloak	Good: 1	<a href="https://www.keycloak.org">https://www.keycloak.org</a>
Koha	Good: 1	<a href="https://koha-community.org">https://koha-community.org</a>
Kubernetes	Excellent: 1 Good: 1	<a href="https://kubernetes.io">https://kubernetes.io</a>
Libreoffice	Poor: 1	<a href="https://www.libreoffice.org">https://www.libreoffice.org</a>
Linux Kernel	Excellent: 1	<a href="https://www.kernel.org">https://www.kernel.org</a>
MapServer	Medium: 1	<a href="https://mapserver.org">https://mapserver.org</a>
Matomo	Good: 1	<a href="https://matomo.org">https://matomo.org</a>
Matrix	Good: 1	<a href="https://matrix.org">https://matrix.org</a>
Memcached	I don't know: 1	<a href="https://github.com/memcached/memcached">https://github.com/memcached/memcached</a>
ModSecurity	Excellent: 1	<a href="https://github.com/SpiderLabs/ModSecurity">https://github.com/SpiderLabs/ModSecurity</a>
Mysql	Good: 2	<a href="https://www.mysql.com">https://www.mysql.com</a>

Name	Evaluation	URL
Nextcloud	Good: 2	<a href="https://nextcloud.com">https://nextcloud.com</a>
Nginx	Good: 1 Excellent: 2	<a href="https://www.nginx.com">https://www.nginx.com</a>
Node.js	Excellent: 1	<a href="https://nodejs.org">https://nodejs.org</a>
NodeBB	Good: 1	<a href="https://nodebb.org">https://nodebb.org</a>
OAW	I don't know: 1	<a href="https://github.com/ctt-gob-es/oaw">https://github.com/ctt-gob-es/oaw</a>
Open Shift	Good: 1	<a href="https://access.redhat.com/products/openshift">https://access.redhat.com/products/openshift</a>
OpenAPI Generator	Good: 1	<a href="https://github.com/OpenAPITools/openapi-generator">https://github.com/OpenAPITools/openapi-generator</a>
OpenJDK	Good: 1	<a href="https://openjdk.java.net">https://openjdk.java.net</a>
OpenSUSE	Good: 1	<a href="https://www.opensuse.org">https://www.opensuse.org</a>
Oskari.org	Excellent: 1	<a href="https://oskari.org">https://oskari.org</a>
Outbackcdx	I don't know: 1	<a href="https://github.com/nla/outbackcdx">https://github.com/nla/outbackcdx</a>
PDFsam	Medium: 1	<a href="https://pdfsam.org">https://pdfsam.org</a>
PHP	Good: 1	<a href="https://www.php.net">https://www.php.net</a>
PostGIS	Excellent: 1 Good: 1	<a href="https://postgis.net">https://postgis.net</a>
PostgreSQL	Excellent: 2 Good: 3	<a href="https://www.postgresql.org">https://www.postgresql.org</a>
QGIS	Excellent: 1 Good: 1	<a href="https://qgis.org">https://qgis.org</a>
RecordManager	Good: 1	<a href="https://github.com/NatLibFi/RecordManager">https://github.com/NatLibFi/RecordManager</a>
Redis	Good: 2	<a href="https://redis.io">https://redis.io</a>
Request Tracker	Excellent: 1	<a href="https://bestpractical.com/request-tracker">https://bestpractical.com/request-tracker</a>
RockyLinux	Good: 1	<a href="https://rockylinux.org">https://rockylinux.org</a>
Ruby	Good: 1	<a href="https://www.ruby-lang.org/en">https://www.ruby-lang.org/en</a>
Samvera	Good: 1	<a href="https://samvera.org">https://samvera.org</a>
Shibboleth	Good: 3	<a href="https://www.shibboleth.net">https://www.shibboleth.net</a>
Skosmos	Excellent: 1	<a href="https://skosmos.org">https://skosmos.org</a>
Smokeping	Good: 1	<a href="https://oss.oetiker.ch/smokeping">https://oss.oetiker.ch/smokeping</a>
Solr	Good: 1	<a href="https://github.com/apache/lucene-solr">https://github.com/apache/lucene-solr</a>
SQLite	Good: 1	<a href="https://sqlite.org">https://sqlite.org</a>
Strapi	Good: 1	<a href="https://strapi.io">https://strapi.io</a>
SuiteCRM	Medium: 1	<a href="https://suitecrm.com">https://suitecrm.com</a>
Traefik	Good: 1	<a href="https://doc.traefik.io/traefik">https://doc.traefik.io/traefik</a>
Typo3	Good: 1	<a href="https://typo3.org">https://typo3.org</a>
Unbound	Good: 1	<a href="https://www.nlnetlabs.nl/projects/unbound/about">https://www.nlnetlabs.nl/projects/unbound/about</a>
Varnish	I don't know: 1	<a href="https://varnish-cache.org">https://varnish-cache.org</a>
VuFind	Excellent: 1 Good: 1	<a href="https://vufind.org">https://vufind.org</a>
WSO2 API gateway for service REST API services	Good: 1	<a href="https://wso2.com">https://wso2.com</a>
Wordpress	Good: 1	<a href="https://wordpress.org">https://wordpress.org</a>
Znuny Helpdesk (former OTRS)	Medium: 1	<a href="https://www.znuny.org">https://www.znuny.org</a>
mariaDB	Good: 1	<a href="https://github.com/MariaDB/server">https://github.com/MariaDB/server</a>
nginx	I don't know: 1	<a href="https://www.nginx.org">https://www.nginx.org</a>
python language	Medium: 1	<a href="https://www.python.org">https://www.python.org</a>
pywb	I don't know: 1	<a href="https://pypi.org/project/pywb">https://pypi.org/project/pywb</a>

Name	Evaluation	URL
sanic framework	Medium: 1	<a href="https://sanic.dev/en">https://sanic.dev/en</a>

Table 12: Open source Dependencies Health Evaluation

Name	Evaluation	URL
Angular	Good: 1	<a href="https://www.npmjs.com/package/@angular/core">https://www.npmjs.com/package/@angular/core</a>
AngularJS	Good: 1	<a href="https://angularjs.org">https://angularjs.org</a>
Apache FreeMarker	Good: 1	<a href="https://freemarker.apache.org">https://freemarker.apache.org</a>
Apache Log4j	Good: 1	<a href="https://logging.apache.org/log4j">https://logging.apache.org/log4j</a>
Apache POI	Good: 1	<a href="https://poi.apache.org">https://poi.apache.org</a>
Apollo gateway	Good: 1	<a href="https://github.com/apollographql/federation">https://github.com/apollographql/federation</a>
Bootstrap	Good: 1	<a href="https://getbootstrap.com">https://getbootstrap.com</a>
Codeigniter	Good: 1	<a href="https://codeigniter.com">https://codeigniter.com</a>
Connexion	I don't know: 1	<a href="https://github.com/zalando/connexion">https://github.com/zalando/connexion</a>
Django	Excellent: 1, Good: 3	<a href="https://pypi.org/project/Django">https://pypi.org/project/Django</a>
Flask	I don't know: 1	<a href="https://pypi.org/project/Flask">https://pypi.org/project/Flask</a>
GDAL	Excellent: 1, Good: 1	<a href="https://gdal.org">https://gdal.org</a>
GeoTools	Good: 1	<a href="https://geotools.org">https://geotools.org</a>
Hibernate	Good: 1	<a href="https://hibernate.org">https://hibernate.org</a>
Laminas	I don't know: 1	<a href="https://getlaminas.org">https://getlaminas.org</a>
Next.js	Good: 1	<a href="https://github.com/vercel/next.js">https://github.com/vercel/next.js</a>
OpenSSL	Excellent: 1	<a href="https://www.openssl.org">https://www.openssl.org</a>
Quarkus	Good: 1	<a href="https://quarkus.io">https://quarkus.io</a>
React	Good: 3, Medium: 1	<a href="https://www.npmjs.com/package/react">https://www.npmjs.com/package/react</a>
Ruby on Rails	Good: 1	<a href="https://rubyonrails.org">https://rubyonrails.org</a>
Chalk	Good: 1	<a href="https://github.com/chalk/chalk">https://github.com/chalk/chalk</a>
Inherits	Good: 1	<a href="https://github.com/isaacs/inherits">https://github.com/isaacs/inherits</a>
jQuery	Good: 1	<a href="https://jquery.com">https://jquery.com</a>
vue-keycloak-ts	I don't know: 1	<a href="https://www.npmjs.com/package/@caassis/vue-keycloak-ts">https://www.npmjs.com/package/@caassis/vue-keycloak-ts</a>