

OSPO Plans: Software

OSPO Launch Event day 2 – Nov 29th, 2023

Giacomo Tenaglia for the OSPO team
Open.Source@cern.ch - <https://opensource.cern>



Introduction

- Free and Open Source Software ([definition](#)):
 - Uses © to permit the 4 freedoms: use, study, share, improve.
- [Academic Training Open Source 2023](#).

Outbound FOSS

```
double leading_cut = 0.0001;
double subleading_cut = 0.0001;
if (leptons[0] == e || leptons[0] == mu)
{
    countCutflowEvent("CutFlow_21_3_3th_1_1")
    if (leptons[0] -> Charge == leptons[1] -> Charge)
    {
        countCutflowEvent("CutFlow_21_3_3th_1_2")
        bool CA_3th_1 = false;
        if (leptons.size() > 2)
        {
            CA_3th_1 = leptons[2] -> PT > ((leptons[1] -> PT) * 0.5);
        }
        if (taus.size() > 0)
        {
            CA_3th_1 = taus[0] -> PT > 20;
        }
        if (!CA_3th_1)
        {
            countCutflowEvent("CutFlow_21_3_3th_1_3")
        }
    }
}
```



Outbound FOSS: introduction

- Outbound FOSS:
 - CERN role: author/contributor.
- “Due diligence”:
 - Copyright: who owns the IP?
 - (Spoiler for MPEs: it’s CERN.)
 - License choice:
 - Original work vs combined/derivative work;
 - License compatibility.

Outbound FOSS: where we are

- Open Source License Task Force report (2012)
- Policy on SW Dissemination (2017)

```
#####  
# (c) Copyright 2021 CERN for the benefit of the LHCb Collaboration      #  
#                                                                    #  
# This software is distributed under the terms of the GNU General Public  #  
# Licence version 3 (GPL Version 3), copied verbatim in the file "COPYING". #  
#                                                                    #  
# In applying this licence, CERN does not waive the privileges and immunities #  
# granted to it by virtue of its status as an Intergovernmental Organization #  
# or submit itself to any jurisdiction.                                  #  
#####  
""Wrappers around Th0r selection framework configurables.""  
from PyConf import configurable  
from PyConf.Algorithms import (  
    ChargedBasicsFilter, ParticleRangeFilter, PrFilter_PV, TwoBodyCombiner,  
    ThreeBodyCombiner, FourBodyCombiner, FiveBodyCombiner, SixBodyCombiner,  
    ParticleContainerMerger)  
from Moore.persistence.particle_moving import is_particle_producer
```

Outbound FOSS: where we are

- **Gitlab**: 138k repos (19k archived), 18k users, 120-270 req/sec (~1/3rd of all Openshift).
- GitHub: 10s of orgs (incl. “**CERN**”), ~1.5k repos mentioning “CERN”, ...
- “FOSS by default”.
- Ecosystem-specific licenses / © statements.

$$W = \lambda_k D D H_1 + \mu H_1 H_2$$

$$D = 0 \quad I = \alpha D D H_2 + \beta H_2 L \Rightarrow$$

$$A_+ = A - \frac{(m_0^2 + m_2^2)}{a^2} \left(\frac{2m_0^2}{2} + \frac{12m_2^2}{2} \right)$$

$$m_{233}^2 + m_{u33}^2 + m_{22}^2 + \dots$$



$$A_4 = -0.6 M_2$$

$$A_i = A \quad m_{2i}^2 + m_{22}^2 > 0$$

$$D_R = (r, z_i, K_i + \dots)$$

$$\lambda H_1 L E R + \mu H_1 H_2$$

$$0 = M_p(t+L)$$

$$\left(\frac{r_i}{t-1} \frac{P_i}{H} + 2 \right) \left(K_i - \frac{W}{t-1} \right)$$

$$m_{jk}^d \frac{1}{v_1} \left(M_1(a)^2 + M_2(a)^2 \right)$$

$$W = \langle H Z_i \rangle = M_W \quad -2a (T, T - Z_i) \langle O V \rangle$$

$$H_1 L_j E_k + \dots$$

$$H_2 L_j N_k$$

$$H_2 H_1 N_k$$

$$\epsilon^{l_i - h_i} \dots$$

Outbound FOSS: standards

- Machine and human readable.
- On-demand reports extraction.
- Software Bill of Materials: “SW ingredients list”.
- Competing standards:
 - [SPDX](#) (LF 2011): license compliance;
 - [CycloneDX](#) (OWASP 2017): security.

Outbound FOSS: tools

“Automatic open source license compliance tools for developers” (G. De Bonis 2022)

	Gitlab	GitHub	Speed	Completeness
FOSSology	Yes	Yes	Fast	Medium
FOSSology wrapper	Yes	No	Very slow	High
Scancode	Yes	Yes	Slow	High
License Finder	Yes	Yes	Fast	Low
Licensee	Yes	No	Fast	Low

Outbound FOSS: REUSE framework

- [FSFE](#) initiative;
- Bundling standards and tools together;
- Requires changes to the repository structure;
- Nicely integrate in Gitlab/GitHub pipelines;
- Example: [Allpix Squared](#).

Outbound FOSS: plans/ideas

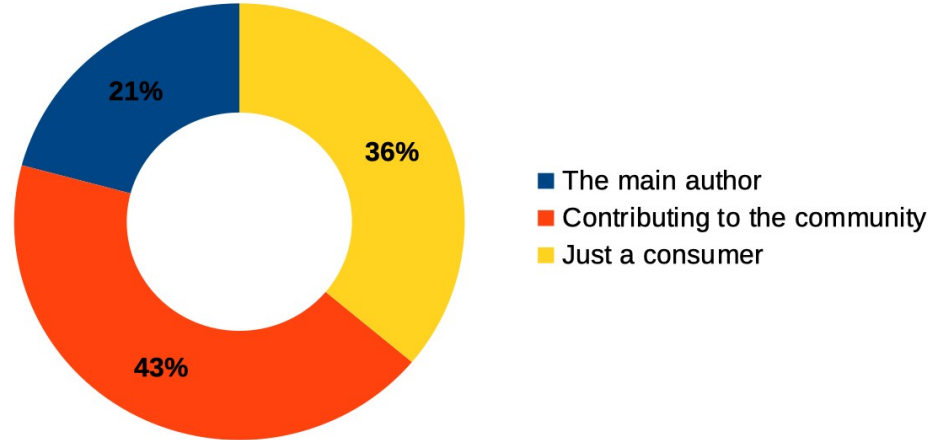
- “How do I license my code?” page;
- Promote [REUSE](#) / [SPDX](#) identifiers;
- Self-service scanning recipes/services (FOSSology);
- Rationalise GitHub presence;
- Catalogue/showcase of CERN FOSS;
- CLA process/inventory;
- Hosted [project metrics dashboards](#)?

Inbound FOSS



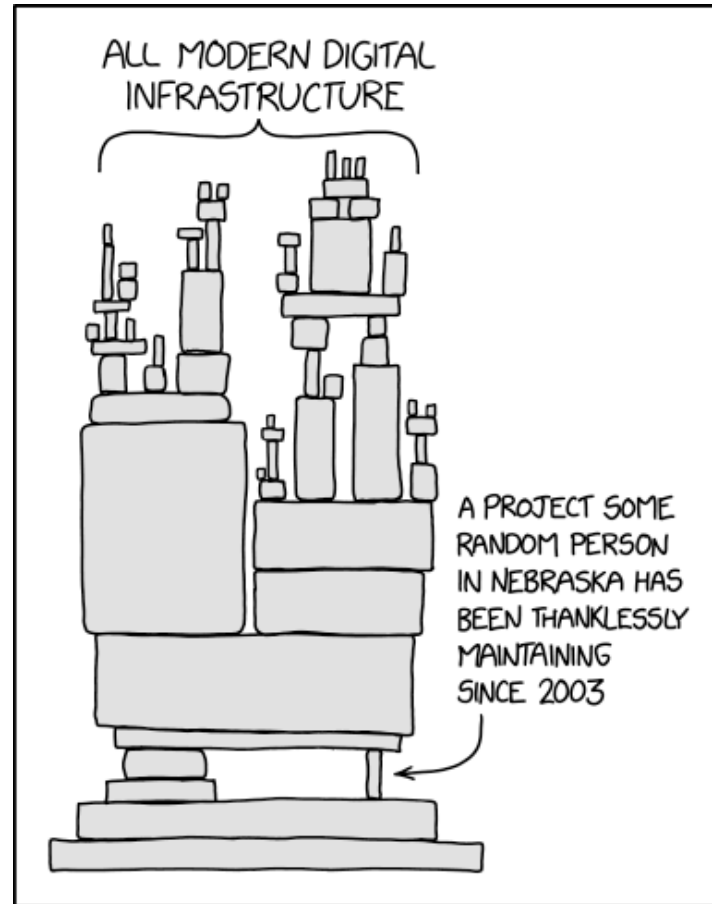
Inbound FOSS: introduction

- 70% of CERN IT services rely on a main FOSS component, of which CERN is:



- Main author of dozens of “long tail” contributions (utilities, libraries, patches, ...).

Inbound FOSS: the dependency problem



Inbound FOSS: audit requirements

- Implement cross-CERN governance processes for FOSS projects.
- Report to the directorate on an annual basis.
- Track FOSS usage across the Organization and provide training and guidance.

Inbound FOSS: plans/ideas

- FOSS dependency tracking process/system;
 - Initial use case: IT services / SNOW SEs;
 - Automation (w/ periodic validation):
 - Puppet / kubernetes / Openshift;
 - Synergy with Enterprise Architecture initiative.
- “How do I purchase FOSS / consultancies?”
- “How to sponsor a FOSS development?”

Q&A

- Outbound FOSS:
 - Is REUSE too much?
 - Automatic SBOM generation / DTrack?
 - Catalogue: opt-in/best practices?
- Inbound FOSS:
 - What are your worries as service managers?