Vorteks

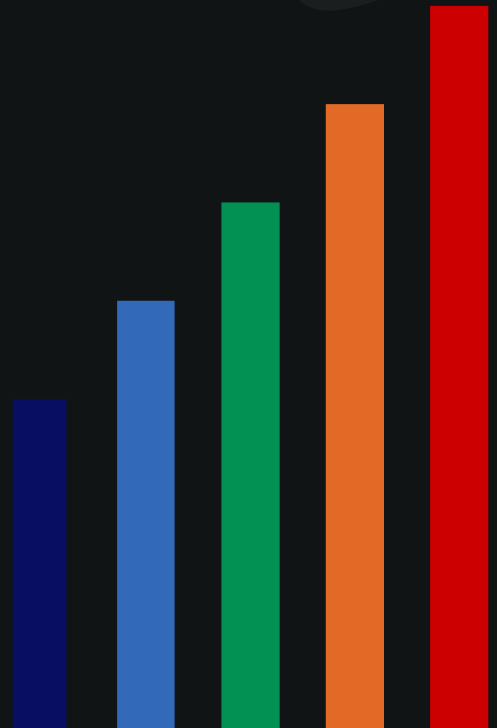
make IT work, make IT free

Open Source et développement: intégration aisée, ou pas?

OPEN SOURCE EXPERIENCE

11 décembre 2025
Paris

Sommaire



Worteks

Société d’expertise, d’édition et d’hébergement Open Source

Contribue activement à de nombreux logiciels libres comme LSC, LemonLDAP::NG, LDAP Tool Box et FusionIAM

Partenaires

Red Hat Advanced Business Partner Reseller

BlueMind

Collabora Online

ONLYOFFICE

Worteks

Une offre Open Source globale.

Solution de déploiement d'infrastructure complexe — V'Opla

Portail de travail collaboratif — V'Sweet

Intégration, support et expertise — V'ise

Hébergement souverain — V'aaS

Solution de gestion des identités et des accès — V'IDaaS

Introduction

Petit historique personnel

- Premiers programmes en 1982 (TI 57, TRS 80, TO7/MO5)
- Gnu CPP (1989)

```
/* Pre-C-Preprocessor to translate ANSI trigraph idiocy in BUF
   before main CCCP processing.  Name `pcp' is also in honor of the
   drugs the trigraph designers must have been on.
```

- Jess (2000), freeware
- Jikes (2001/2002), OSS JiJ (Jikes in Java)

Apache

- Committer Apache Directory (incubateur) 15 mars 2005
- Chairman Apache Directory, puis Apache MINA
- Démarrage de Apache Directory LDAP API en 2006
- Démarrage de Apache Mavibot en 2012
- Membre depuis 2007
- Mentor de 5 projets (dont Groovy, Syncope, Shiro)
- 13 000 commits en 20 ans
- Participant aux projets Apache Directory, LDAP API, Studio, Mavibot, Scimple, Fortress, Kerby, Mina, FtpServer

L’OSS en 2025

L’OSS est partout

(Jacks, 2022). Other recent studies have come to similar conclusions showing that open source software (OSS) – software whose source code is publicly available for inspection, use, and modification and is often created in a decentralized manner and distributed for free – appears in 96% of codebases (Synopsys 2023), and that some commercial software consists of up to 99.9%

Quelle est la valeur de l’OSS ?

Etude de 2024 par Manuel Hoffmann, Frank Nagle et Yanuo Zhou (Harvard Business School) [1]

- Coût d’écriture unitaire: 4.15G$
- Coût d’usage: 8.8T$
- Une économie de près de 75%

Comment le mesurer ?

- Valeur = prix * quantité vendue
- OSS: prix =0, quantité vendue non quantifiable…
- Une solution: valeur de remplacement
  - Combien coûterait l’écriture du code s’il était payé?
  - Combien coûterait ce même code si chaque entreprise utilisatrice payait le développement?
  - Combien coûterait l’achat d’un logiciel propriétaire?

Qui connaît et utilise ADS?

Total Physical Source Lines of Code (SLOC) = 263,642

Development Effort Estimate = 69.68

Estimated Average Number of Developers (Effort/Schedule) = 25.93

Total Estimated Cost to Develop
= $ 9,412,543

Studio → 5296 commits, 98% par 3 personnes

Les briques logicielles

Choix techniques

- Critères mesurables
  - L’adéquation au besoin
  - La compétence interne
  - Les performances
  - L’intégration avec les autres composants
  - Les limitations du composants
  - Administrabilité/Monitoring
- Critères non mesurables
  - La maintenance du projet
  - Possibilité de patcher
  - Qualité de la documentation
  - Support des CVEs

Choix organisationnels

- Disponibilité de resources externes
- Disponibilité de support
- Vitalité du projet
  - Fréquence des releases, nb de participants
- Etat de la communauté
  - ‘bus factor’, age, nombre d’acteurs, compétence des acteurs, financement
- Type de communauté (BDLF, fondation, ‘one man show’)

Choix économique

- Prix du support
- Coût des intervenants externes
- Prix de l’intégration en PaaS
- Disponibilité en mode cloud

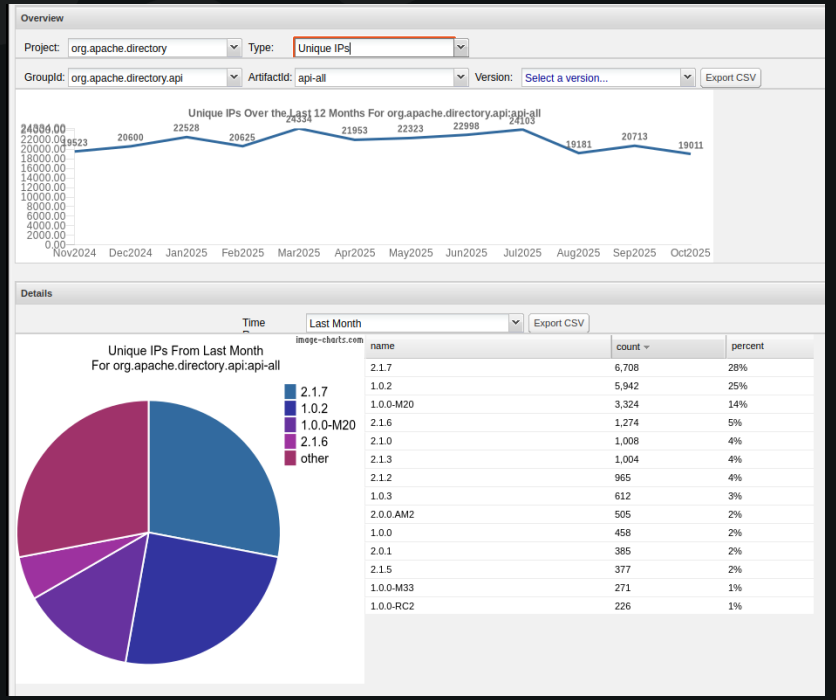
Choix juridique

- Compatibilité de la license avec le Business Model (GPL, ASF,
- Coût des intervenants externes
- Prix de l’intégration dans le Cloud
- Compliance [2]
- Limitation contractuelles
- Dépendences transitives

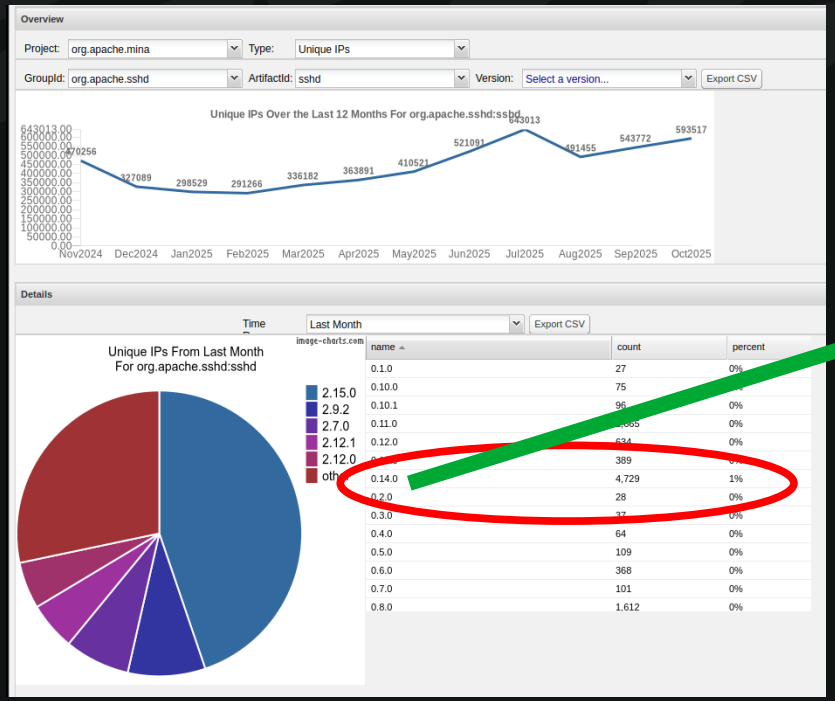
En pratique

Les versions

- Quelles version utiliser?
- Suivi des évolutions
- Contraintes liées à l'environnement?
- Rapidité d'évolution (versions de Java par exemple)



Les versions...



Les évolutions

- Le langage sous-jacent (dépréciation de composants, plsu de support)
- Les librairies et leur évolution (JS dans Java par exemple)
- Le problème des conflits de librairies (versions)
-

Dépendences transitives

- Quand vous avec une dépendence, vous héritez de ses propres dépendences
- Toujours vérifier l'ensemble des dépendences!
- mvn dependency:tree

```
[INFO] +- org.apache.directory.api:api-ldap-schema-converter:jar:2.1.8-SNAPSHOT:compile
[INFO] |  +- org.apache.directory.api:api-i18n:jar:2.1.8-SNAPSHOT:compile
[INFO] |  +- org.apache.directory.api:api-ldap-model:jar:2.1.8-SNAPSHOT:compile
[INFO] |  |  +- commons-codec:commons-codec:jar:1.19.0:compile
[INFO] |  |  \- com.github.ben-manes.caffeine:caffeine:jar:2.9.3:compile
[INFO] |  +- org.apache.directory.api:api-util:jar:2.1.8-SNAPSHOT:compile
[INFO] |  |  \- org.apache.commons:commons-text:jar:1.14.0:compile
[INFO] |  \- org.apache.servicemix.bundles:org.apache.servicemix.bundles.antlr:jar:2.7.7_5:compile
```

Vérifier les dépendances

- Toujours vérifier les signatures
- Un repository peut être attacké (NPM)

Les SBOMs, une solution ?

- SBOM: “Software Bill of Material”
- Outils de génération sur les dépendences transitives
- Contrôle des licences
- Contrôle des CVEs
- A l’ASF: https://cwiki.apache.org/confluence/display/SECURITY/SBOM+Software+Bill+of+Materials

Les licenses

- Quelle licence ?
  - https://choosealicense.com/
- Les contraintes
  - GPL like : distribution sous la même license
  - ASF/MIT like: mention obligatoire
  - Et le cloud?
- Les risques

Le support

Quoi ?

- Identifier les besoins
  - Le composant est-il critique?
  - Y-a-t-il des anomalies qu’il a été nécessaire de contourner?
  - Les performances sont-elles adaptées?
  - Le composants est-il supervisé? (logs, traces, statistiques)
- Est-il hébergé (PaaS/SaaS) ou intégré au produit?

Qui ?

- Un participant actif du projet/composant/librairie
  - A condition qu’il connaisse la partie qui pose souci
  - Les releases ne dépendent pas forcément aux besoins
- La communauté
  - Best effort, zéro garantie
- Un externe
  - Disponibilité?
  - Contrat de support avec une société de service

Comment ?

- Resources internes: formation, embauches
- Resources externes internalisées (prestataire, société de service): contrat de service
- Resources externes (SaaS/PaaS): contrat d’exploitation
- Communauté: Mailing list, tickets d’anomalie

Dans le monde réel

“Hello,

I am writing to inquire about the support and update policies for Apache Directory Studio.

In our company’s policy, we are only allowed to use applications that are regularly updated. However, I noticed that the last update for Apache Directory Studio was released in July 2021. This raises some concerns regarding the ongoing support and security of the application.

Could you please clarify how support is ensured for Apache Directory Studio? Specifically, I would like to know if security vulnerabilities are addressed promptly for all users, or if it is advisable to consider commercial support for timely updates and fixes. Additionally, I would like to know if security updates are provided in the open-source version and whether the community is still actively working on the project.

Thank you for your assistance. I look forward to your response.

Best regards”

Forker ou pas ?

- Une gouvernance inadaptée
  - BDFL (Benevolent Dictator For Life)
  - “One person project”
  - Communauté fermée
  - Manque de moyen
- Le changement de ‘propriétaire’:
  - Hudson (Jenkins), MySQL (MariaDB), OpenOffice (LibreOffice)…
- La fin de vie:
  - NCSA (Apache httpd)

Exemple : OpenSSL

- 2009: Agglomerated SSL (dead by 2015)
- 2014: LibreSSL (OpenSSD) (HeartBleed)
- 2014: BoringSSL (Google) → Tink [2]
- 2017: QuicTLS (Akamai/Microsoft, based on OpenSSL 3.3
- 2019: AWS LC (OpenSSL + BoringSSL)

La raison principale: Une équipe de volontaires sous-dimensionnée, sans moyens.

LTS ou pas

- Accélération des releases
  - Apache Httpd: de 1.3 à 2.4 en 30 ans
  - Firefox: De 1 à 5 en 7 ans, maintenant 1 release par mois
  - Chrome: de 1.0 à 141.x en 17 ans
- Besoin de versions stables: LTS
  - Maintenues sur plusieurs années
- Le coût de la maintenance d’anciennes releases est élevé!

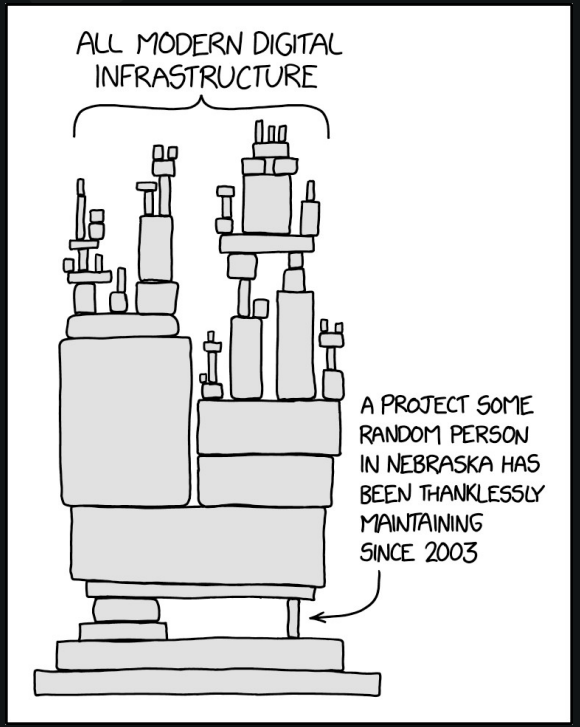
Retrouver de l’information

- Les versions évoluent (vite): Firefox 143…
- Les sociétés se font racheter
- Les liens sont cassés
- Le contenu disparait
- Même Internet Archives a du mal…
- Sans compter les IA qui pourrissent les résultats car entraînées sur les dernières informations disponibles :-/

« Open Source is one person »

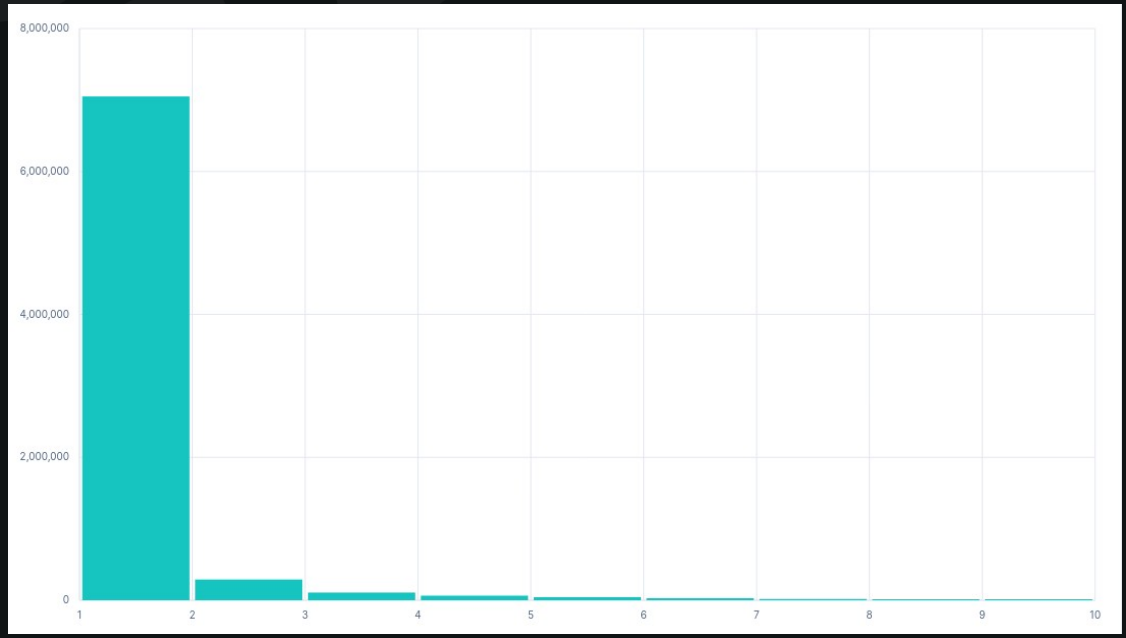
https://xkcd.com/2347/

- Log4J
- OpenSSL
- Tomcat
- Etc, etc.



«Open source is 1 person» [1]

- D’après ecosyste.ms:
  - 11.8M projets
  - Donc 7M avec 1 personne
  - 4M nombre inconnu...



Exemple : Apache Log4J2

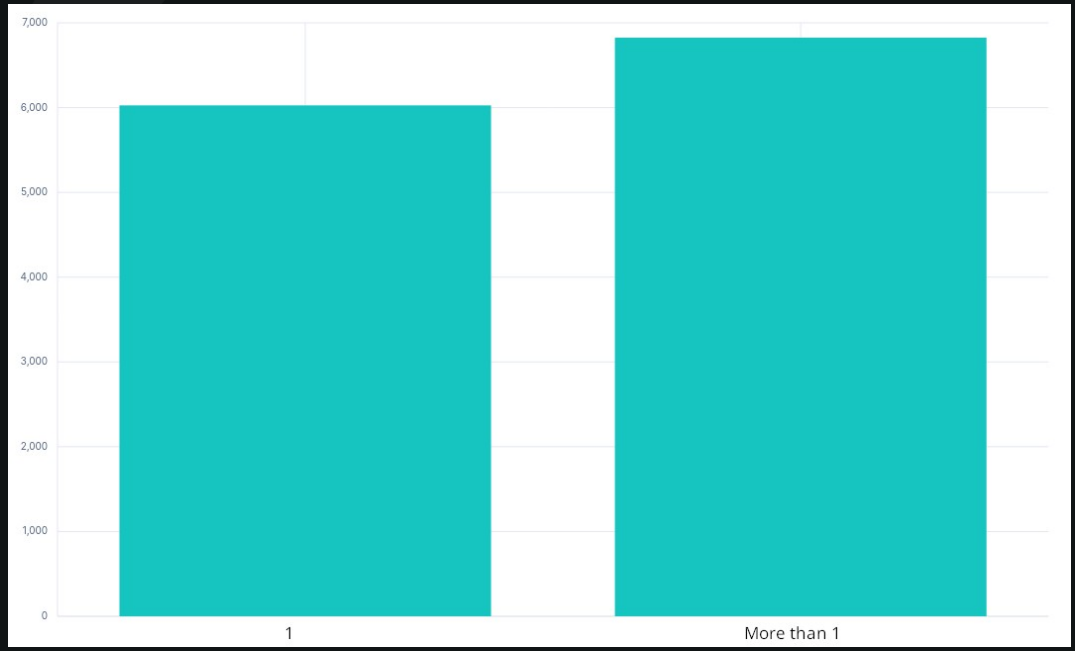
- Sur les 2 dernières années
- 1748 commits
- Donc 452 par des outils (ASF CI Release Manager, Dependabot), soit 26%
- Dev #1: 649 commits, 50% des commits
- Dev #2: 422 commits, 33% des commits
- Dev #3: 74 commits, 6% des commits
- Les 3 premiers: 88% des commits!

Exemple : Apache Tomcat

- 24 891 commits
- Dev #1: 19029 commits, 77% des commits
- Dev #2: 2594 commits, 10% des commits
- Dev #3: 1115 commits, 4% des commits
- Les 3 premiers: 91% des commits!

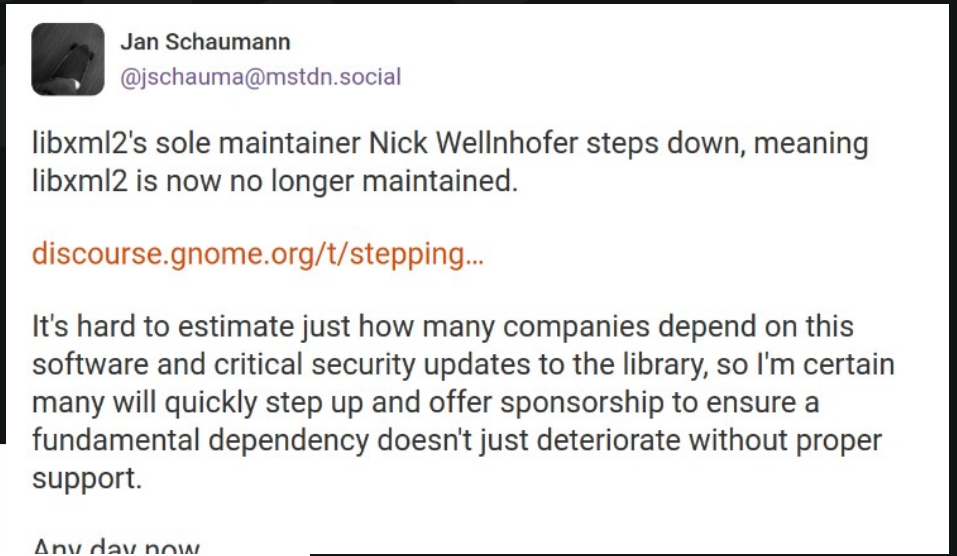
«Open source is 1 person»[2]

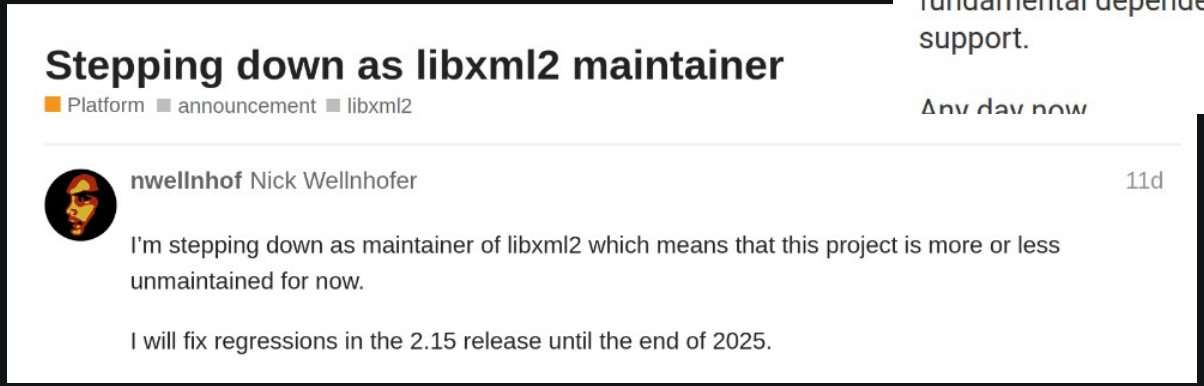
- NPM:
  - Pour les projets avec +1M de download/mois :



«Open source is 1 person»[3]

- Libxml2:





Questions ouvertes

Longévité ?

“This repository is now archived.

Thank you all for your excellent contributions.

May it continue to live in all your wonderful forks.”

OPENLDAP Exporter

Gestion des CVEs

- Comment informer les utilisateurs?
- Qui corrige?
- Combien de temps la faille est-elle accessible?
- Qui finance la correction

Que faire en cas de bug ?

- Attendre?
- Qui contacter pour avoir une correction?
- Comment intégrer une correction?
- Gérer un fork en attendant?

Et l’IA ?

- PR générés par des LLMs: qualité?
- Quid de la license?
- Rapports d’anomalie invalides
- Charge des serveurs constamment requêtés par des bots…
- Qualité des moteurs de recherche débatable

Qui finance ?

- Des sociétés privées (Google, IBM, etc)
- Des fondations (OpenSSL, TDF, The ASF ...)
- Des organisations gouvernementales (NGI, NLNET, ...)
- Des initiatives publiques ou privées (GH, bug bounties, ...)
- Beaucoup d’individus sur leur temps personnel

Parlons de souveraineté…

- De qui dépend-t-on?
- Des initiatives nationales ou européennes en ordre dispersées [5]
- Un effort gouvernemental certain, mais encore peu structuré et surtout auto-centré et national

Questions ?

Références

- [1] https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf
- [2] https://www.linuxfoundation.org/hubfs/LF%20Research/OpenSourceLicenseComplianceReport_010224.pdf?hsLang=en
- [3] https://opensourcesecurity.io/2025/08-oss-one-person/
- [4] https://medium.com/asecuritysite-when-bob-met-alice/goodbye-openssl-and-hello-to-google-tink-583163cfd76c
- [5] https://interoperable-europe.ec.europa.eu/collection/open-source-observatory-osor/funding-opportunities-open-source-software-projects-public-sector#fundingresources

www.worteks.com

info@worteks.com
+33 1 84 20 86 47
worteks_com
worteks