



---

## NGI\_DAPSI 2ND OPEN CALL

---

Acronym: **fedeproxy**

Title: A federation proxy for software development forges

19/01/2020

# 1 OVERVIEW OF THE PROPOSAL

## 1.1 EXECUTIVE SUMMARY

Subdomain	Justify your selection and briefly explain it.
<b>Service Portability</b>	<b>fedeproxy</b> facilitates the portability of software projects and helps them move freely from a software development hosting service (also known as forge) to another.

**fedeproxy** is an online service to federate forges. The software projects hosted on one forge are synchronized in real time with their counterparts on other forges, via the W3C ActivityPub protocol. Developers can freely use the forge of their choosing while contributing to the same project. It operates independently from the forges and serves as an incubator with rapid prototyping to research the best user experience. Initially focused on GitHub and GitLab, it can be extended to all forges in the future.

Most Free Software projects are hosted on proprietary online services (called forges) that do not provide an export/import feature flexible enough to allow them to move easily (GitHub, SourceForge, etc.). Although the code repository can conveniently be moved from one service to another, other essential components such as pull or merge requests, issues, developer accounts, continuous integration, etc. cannot and the project, as a whole, is trapped.

When two forges are federated via **fedeproxy**, the actions carried out by a developer on a forge are sent to the other, and vice versa. For instance:

- The **fedeproxy** service runs with a proxy for GitLab and a proxy for GitHub;
- The project Ceph exists on GitHub;
- I run a self-hosted GitLab instance;
- I ask the **fedeproxy** service to federate the Ceph project from GitHub with my GitLab instance;
- I browse the issues of the Ceph project on my GitLab instance;
- I comment on an issue on my GitLab instance;
- My comment is copied over to the GitHub Ceph project by **fedeproxy**;
- A GitHub user answers my comment which **fedeproxy** copies over to the Ceph project that resides on my GitLab instance;

Once all components of a software project are federated (issues, code, etc.) the project as a whole is effectively duplicated in real time on multiple forges, thus allowing the users to chose the one they prefer. **Service portability is achieved because federated forges continuously maintain identical copies of the software project although they are operated by independent actors and running different servers and user interfaces.**

## 2 EXCELLENCE/INNOVATION

---

### 2.1-Idea and Objectives

Software forges became popular in 2001 and service portability issues appeared at the same time. There are two ways to move a software project from one forge to another: (i) import/export: an “all or nothing” approach to move from one forge to another (ii) federation: a two way communication to synchronize software projects

The **import/export method did not succeed and there is no denying, twenty years later, that it may be time for another approach**. The number of forges publicly accessible was multiplied by at least two order of magnitude since 2001 and they would benefit from being federated. But no forge software supports it, although a data model since 2019 as well as the W3C ActivityPub protocol. Although a full featured implementation is a very ambitious undertaking, **it does not need to be complete to be usable**. For instance, the federation of issue comments could be the only feature available while others are implemented. In addition, it is practical to **experiment on federation with a proxy based on the forges API** (or even web scrapping) before attempting a native integration by modifying the code of a forge.

### 2.2-Technical challenges and barriers to be solved

Most Free Software projects are hosted on proprietary online services (called forges) that do not provide an export/import feature flexible enough to allow them to move easily. Although the code repository can easily be moved from one service to another, other essential components such as pull requests, issues, etc. cannot and the project, as a whole, is trapped. This has many negative consequences: **(I) Fragility**: When a service shutdown the development history is lost. **(II) Censorship**: A project may be unexpectedly blocked by an embargo. **(III) Lock-in**: It may take years for a projects to manually migrate to another forge. **(IV) Exclusion**: Users who do not agree to the TOS are effectively excluded. **(V) Vulnerability**: projects must trust provider with the integrity of the repository. **(VI) No interoperability**: the API and format are not based on open standards.

### 2.3-Free Software Licenses and/or open standards

**fedeproxy** will be released under the AGPLv3 license. The code contributed to existing code bases will be released under licenses compatible with their own licenses. The ActivityPub W3C protocol will be used to communicate between **fedeproxy** instances.

### 2.4-Project outcomes

- A software **library** is created, implementing federation proxy features for GitHub and GitLab based on the ActivityPub protocol;
- A software for running a **server** is created, implementing a self-hostable service (named **fedeproxy**) enabling all developers worldwide, including the 50 million GitHub users to: (I) Authenticate on a self-hosted GitLab instance and on GitHub (ii) Request the federation of a project residing on both GitHub and GitLab;
- One user research report based on at least 9 interviews of forge users focusing on their past experience regarding the portability of software project;
- A community of **fedeproxy** users is created during phase 1 and 2 of the project to ensure feedback in real-life situation of a development process: at least 2 of them use it on a daily basis, at least 10 of them use it occasionally;

- At least 3 reports explaining the data model and vocabulary used by **fedeproxy** and communicated to [Gitea](#) and [forgefed](#) with a request for comment;
- At least 3 merge requests to GitLab are submitted to implement features relevant to forge federation.

## 2.5-Added value

- The **fedeproxy server and libraries**: (i) bootstrap the work towards a full featured forge federation, (ii) facilitate the implementation of additional features in the future, (iii) provide a working implementation that allows developers to experiment
- The **community of users** of the **fedeproxy** service demonstrate the usage value of a partial implementation of forges federation and is therefore a practical method to incrementally make progress towards a full featured forge federation attractive for a very large user base, i.e. all developers working online
- The **user research report** shows what current forge users consider important when it comes to their ability to move from one forge to another
- The **reports explaining the data model and vocabulary** (i) provide real world data for forgefed to make progress toward a standard data model and vocabulary, (ii) provide a foundation for Gitea to use when implementing forge federation natively
- The **merge requests to GitLab**: (i) encourage the GitLab developers to make progress towards a native implementation of federation, (ii) demonstrates how a feature implemented in **fedeproxy** can be natively supported by a forge

## 2.6-Relevance, socio-economic impact and benefits

**Enabler for developers to move between forges:** The goal of **fedeproxy** is not to attract a large audience but to serve as a demonstration and experimentation ground. It is designed to be an enabler towards the generalization of forge federation. Every software developers (and not just the [50 million GitHub users](#)) should be able to freely move from one forge to another without friction. When forge federation becomes common place, **all developers will use the forge they prefer instead of the forge on which the project resides.**

**Promotes the concept of federated development:** Most developers work on Free Software in a centralized way and do not see the benefit for decentralization and federation. Implementing federation and **using it for practical purposes is a way to discover use cases and imagine new ones.**

**Improves the durability of software projects:** Organizations are routinely [impacted by the disappearance of forges](#) or by changes in TOS, which translates into a loss of value and money. **By continuously duplicating issues and pull / merge request on GitLab and GitHub (redundancy is another way of looking at federation) on two forges, fedeproxy improves the chances that they are recovered**, thus saving value and money for all organizations depending on the impacted projects. The same method can later be applied to all forges.

**Scale out forge federation development:** By providing a minimal infrastructure and a few features, **fedeproxy** demonstrates that the development towards a full featured forge federation scales out, The sum of all those efforts is much higher than what would be needed to implement an export/import feature but it is **distributed over a longer period of time and among many actors. It is therefore more likely to make progress and succeed.**

## 3 EXPERTISE AND EXCELLENCE OF THE TEAM

### 3.1 TEAM COMPOSITION

Name of the person	Role in the project	CV	Entity
Loïc Dachary	Developer, Coordinator	<a href="https://blog.dachary.org/cv/">https://blog.dachary.org/cv/</a>	Individual, France
Pierre-Louis Bonicoli	Developer	<a href="http://libregerbil.fr/CV_Bonicoli.pdf">http://libregerbil.fr/CV_Bonicoli.pdf</a> <a href="http://libregerbil.fr/Contributions.pdf">http://libregerbil.fr/Contributions.pdf</a>	Libregerbil, France

#### Loïc Dachary

In 2001 Dachary [raised concerns about centralized proprietary forges](#) and worked with the [Free Software Foundation](#) to setup, install and maintain the [Savannah forge](#). He also contributed to the [GNA!](#) forge, until 2017, when it shut down. In the recent years Dachary published software to migrate software projects from [GitHub to GitLab](#) and infrastructure as code including [GitLab deployment Ansible playbooks](#) as well as end to end integration tests for a [Django based API server](#) including the automated installation of a GitLab server for the duration of the test.

#### Pierre-Louis Bonicoli

Libregerbil is a french Free Software service provider founded in 2015 by Pierre-Louis Bonicoli, a Python developer with 10+ years of experience. In 2020, Libregerbil [improved the support of GitLab within Zuul](#), a continuous integration project and added [support for the Fuga OpenStack provider](#) to the Enough project. Libregerbil has a long track record of [contributions to the Ansible project](#). Bonicoli runs [a redmine instance](#) and made [some minor contributions](#) to the codebase. He also made contributions to a [bot interacting with GitHub](#).

### 3.2 TEAM MOTIVATION

In 2018 Dachary [closed his GitHub account](#) for ethical reasons and has since been unable to participate in Free Software projects hosted there. The federation of forges would allow him to reconnect with these projects.

Loïc and Pierre-Louis would both use **fedeproxy** and contribute to its development if it already existed, on a volunteer basis, because they need it for their day to day work. But the initial effort to create **fedeproxy** from scratch cannot conveniently be done on a volunteer basis and requires funding. In addition, because Libregerbil is a Free Software service provider, the expertise developed while creating **fedeproxy** may generate additional income in the future, if a market for the development of federation features emerges.

## 4 PROJECT PLANNING

### 4.1 MAIN ACTIVITIES OF THE PROJECT

**a-User research:** Forge federation is a new idea and its implementation would benefit from a user centered design that requires user research.

**b-Infrastructure and development:** (i) Create a self-hosted development environment for **fedeproxy** (with a GitLab forge and the associated CI) (ii) Create the **fedeproxy** software, a Django based server relying on ActivityPub and (iii) Define a data model and vocabulary (iv) Create **fedeproxy GitLab & GitHub**, two modules based on the [GitLab](#) & GitHub API for **creating and commenting on issues and merge requests** (v) Create a reference documentation including instructions for self hosting.

**c-User experience:** Define the user experience roadmap for interacting with the **fedeproxy** server, based on the user research report.

**d-Distributing and operating the service:** (i) Setup a **fedeproxy** server (ii) Publish **fedeproxy server**, **fedeproxy GitLab** and **fedeproxy GitHub** on the [Python Package Index](#)

**e-Advocacy and feedback:** Reach out to software developers, organizations and forge maintainers (Gitea & GitLab) to foster a community of **fedeproxy** users.

#### 4.1.1 Milestones (phase 1 has a green background, phase 2 is blue)

Legend U: User Research, I: Infrastructure, A: Advocacy, O: Operations.

N°	Milestone description	Date
U1	Prepare the user research, prepare the research sessions, create an intercept interview script, find participants	M1
I1	self-hosted development environment and website	M1
U2	Conduct interviews with the participants	M1,M2,M3
I2	Implement the <b>fedeproxy</b> GitLab and <b>fedeproxy</b> GitHub modules, the <b>fedeproxy</b> server, end-to-end integration tests, documentation	M1,M2,M3
A1	Reach out to GitLab implementors and submit merge requests to simplify the implementation of <b>fedeproxy</b> GitLab	M2 to M4
A2	Reach out to Gitea to implement a federation data model and vocabulary	M2 to M4
O1	Publish <b>fedeproxy</b> software on a monthly basis	M2 to M4
U3	Affinity mapping, result analysis, user research report, roadmap	M4
I3	Create the <b>fedeproxy</b> server user interface	M5-M7
O2	Create a production environment for running the <b>fedeproxy</b> server	M5
O3	Run the <b>fedeproxy</b> in production	M7
A3	Reach out to software developers, organizations and forge maintainers	M8,M9
A4	Seek feedback from users and modify <b>fedeproxy</b> accordingly	M8,M9
A5	Reach out to GitLab implementors and submit merge requests to simplify the implementation of <b>fedeproxy</b> GitLab	M5 to M9
A6	Reach out to Gitea to implement a federation data model and vocabulary	M5 to M9
O4	Publish <b>fedeproxy</b> software on a monthly basis	M5 to M9

## 4.2 VALUE FOR MONEY

TABLE 1 PERSON-MONTH

Entity	Name of the person	Person months
Libregerbil	Pierre-Louis Bonicoli	4.5
Individual	Loïc Dachary	4.5
	<b>Total</b>	<b>9</b>

Each participant must attend three face to face events for which the location will be precised later by the organisers: the travel budget includes them. In case the pandemic situation prevents these events to happen in a face to face format, this budget will be reallocated to Personnel costs to allow additional person months (0.66pm) for each partner.

TABLE 2 TOTAL BUDGET

Cost category	Libregerbil	Loïc Dachary	Total Amount (€)
Personnel	27,000	27,000	54,000
Travels	4,000	4,000	8,000
Overheads (20%)	6,200	6,200	12,400
<b>Total</b>	<b>37,200</b>	<b>37,200</b>	<b>74,400</b>