

# Forges

## Usage

Some forges might be able to:

- Provide a way for users to create and manage repositories by themselves, without any administrator interventions.
- Enable users to fork repositories very efficiently (without having the users push all the repository contents over a slow Internet connection).
- Enable users not used to sending patches by mail to have the forge do it for them.

## Requirements

Several forges have issues that prevent its use and deployment by Replicant and/or other free software projects.

Here are some of the requirements we could have:

- Fully free software: we currently don't have the resources to maintain fully free forks of forges that aren't fully free software.
- Low resource usage: we have a VM with very low CPU and RAM.
- Javascript requirements: we can't ship Nonfree JavaScript and forcing users to run specific JavaScript code without being able to easily modify it is an issue, even if the code is free software. To work to avoid that we have the following requirements:
  - A Friendly upstream that accepts patches to make all features work without JavaScript.
  - A code base that don't depend a lot on JavaScript
  - If possible all features should be usable without JavaScript
  - If possible the JavaScript has to be compatible with LibreJS
- Workflow requirements:
  - Have the ability to customize the interface, for instance if the forge doesn't support a mail workflow we need to add a link from within the forge to a page that explains how to send patches to the mailing list. The same probably applies to bug tracker system as we already have one that is already handled by redmine.
  - Friendly upstream that can accept patches for integrating it with a mailing based workflow. We might need to customize it as we have many git repos.
  - Code base where integrating a mailing list based workflow is not too hard to do
- Have it packaged in a distribution we can use in the FSF VMs to reduce maintenance cost

## Forge evaluations done by other Free Software organizations

The FSF is currently evaluating forges to compliment their Savannah servers:

<https://www.fsf.org/blogs/sysadmin/coming-soon-a-new-site-for-fully-free-collaboration>

FSF 2020 forge evaluation page on the LibrePlanet wiki: [https://libreplanet.org/wiki/FSF\\_2020\\_forge\\_evaluation](https://libreplanet.org/wiki/FSF_2020_forge_evaluation)

December 4th, 2020 update: "The FSF forge is still a work in progress. Since our last update, summer intern Amin Bandali deployed a testing instance of SourceHut, one of many possible programs for the forge."

<https://www.fsf.org/bulletin/2020/fall/updates-from-the-fsf-tech-team>

## Possible forges for Replicant

### Gitlab

We used gitlab before and we had resource usage issues. The VM went out of RAM because of that.

It also extensively depends on JavaScript and it doesn't look easy at all to fix.

From our [FOSDEM 2020 report](#) :

"GitLab was first built upon a framework that did most of the work on the server side, but at some point they switched to a framework that does client-side rendering with JavaScript.

As we had an opportunity to talk with people from the GitLab team, we asked them whether they would be open to accept patches that fix this. They explained us that such would require to double the UI work for everything, but that it might be possible to do server side rendering with the same JavaScript that's used on the client. The issue is that it cannot make requests from buttons like that, so

in addition to the page rendering that could happen through server-side JavaScript, introspection could be used to rewrite the buttons."

It also hide buttons, which is awful for users as they don't understand what is going on.

Packages: ?

## Libregit.org (GOGS/Gitea)

**Website:** <https://libregit.org>

**Primary Developer:** SharePunks Collective - spks.cc

**Primary Maintainer:** Lauris BH

### Pros

- Based of Gitea and Gogs, so self-hosting is the focus.
- LibreJS complaint.
- Simple user export and import of data, including migration feature.
- Simple and easy to use interface, making for the easy integration for non-technical users.
- FSDG complaint distributions installations would both work with minimal effort.

### Cons

- MIT Licensed, not committed to free software philosophies given it is permissibly licensed.
- Email setup is not the most straight forward but it could be accomplished.
- The size of the project is growing exponentially given its upstream, which includes GOGS/Gitea.

Packages: ?

## Pagure

**Website:** <https://pagure.io/pagure>

**Command line tools :** <https://pagure.io/pag-off>

Features:

- [The buttons seem to be greyed out when they cannot be used](#) This is much better than Gitlab where the buttons like "Send pull request" completely disappear if the feature is deactivated. Having button like that disappearing is really bad as users struggle to understand what they are doing wrong and can't contribute, when in fact the feature is simply deactivated without any way that shows it.
- It's possible to create a theme that replaces the Pull Requests tab with a link (for instance the link could point to a page that explains how to send patches to the mailing list). src.fedoraproject.org uses that to replace the Issues tab with a link to Red Hat's Bugzilla
- It's compatible with LibreJS but still requires JavaScript for some functionalities. Upstream seem friendly and is [probably very interested in getting patches to make it possible to use all features without any JavaScript](#) . Few features seem to depend on JavaScript.
- There is some interest in bridging a mailing based workflow to pagure: <https://pagure.io/pagure/issue/15>

Packages: [currently packaged in Debian SID](#), in [a PPA](#), might be packaged in other distributions

Contacts: #pagure on Freenode and/orthe [pagure-devel@lists.pagure.io](mailto:pagure-devel@lists.pagure.io) mailing list

## Sourcehut

**Website:** <https://sourcehut.org>

**Primary Maintainer:** Drew Devault - [sir@cmpwn.com](mailto:sir@cmpwn.com)

**Person who wrote this section:** j3s - [j3s@c3f.net](mailto:j3s@c3f.net) (feel free to contact me with any questions!)

### Pros

- Completely integrated mailing list, build, todo, hub, and git repositories. Organizations/groups coming eventually.
- Only has 3 static js files. Extremely easy to support LibreJS. Requires no Javascript to function.
- Simple user export of data
- AGPLv3, committed to free software philosophies and open source communities <https://sourcehut.org/blog/2019-10-23-srht-puts-users-first/>
- Also has wiki/manpage support, a paste service, and a dispatch service.

- Trisquel/Parabola installations would both work with minimal fuss: <https://man.sr.ht/packages.md>

### Cons

- Alpine Linux is the most supported distribution, and doesn't meet RYF certification.
- Email setup can be a bit more complicated than alternatives, but it's fine if one is familiar with email concepts - SPF/DKIM/etc
- Drew maintains that sourcehut is "in alpha stage" - he goes over what that means here: <https://sourcehut.org/alpha-details>

### See Also

Relevant email from the libreplanet-dev email list about how the Sourcehut CI can optionally be adapted to provide CI support for other forges, like Pagure and Gitea: <https://lists.libreplanet.org/archive/html/libreplanet-dev/2021-01/msg00001.html>