Institute of Mathematics and Statistics – University of São Paulo

StarFLOSS

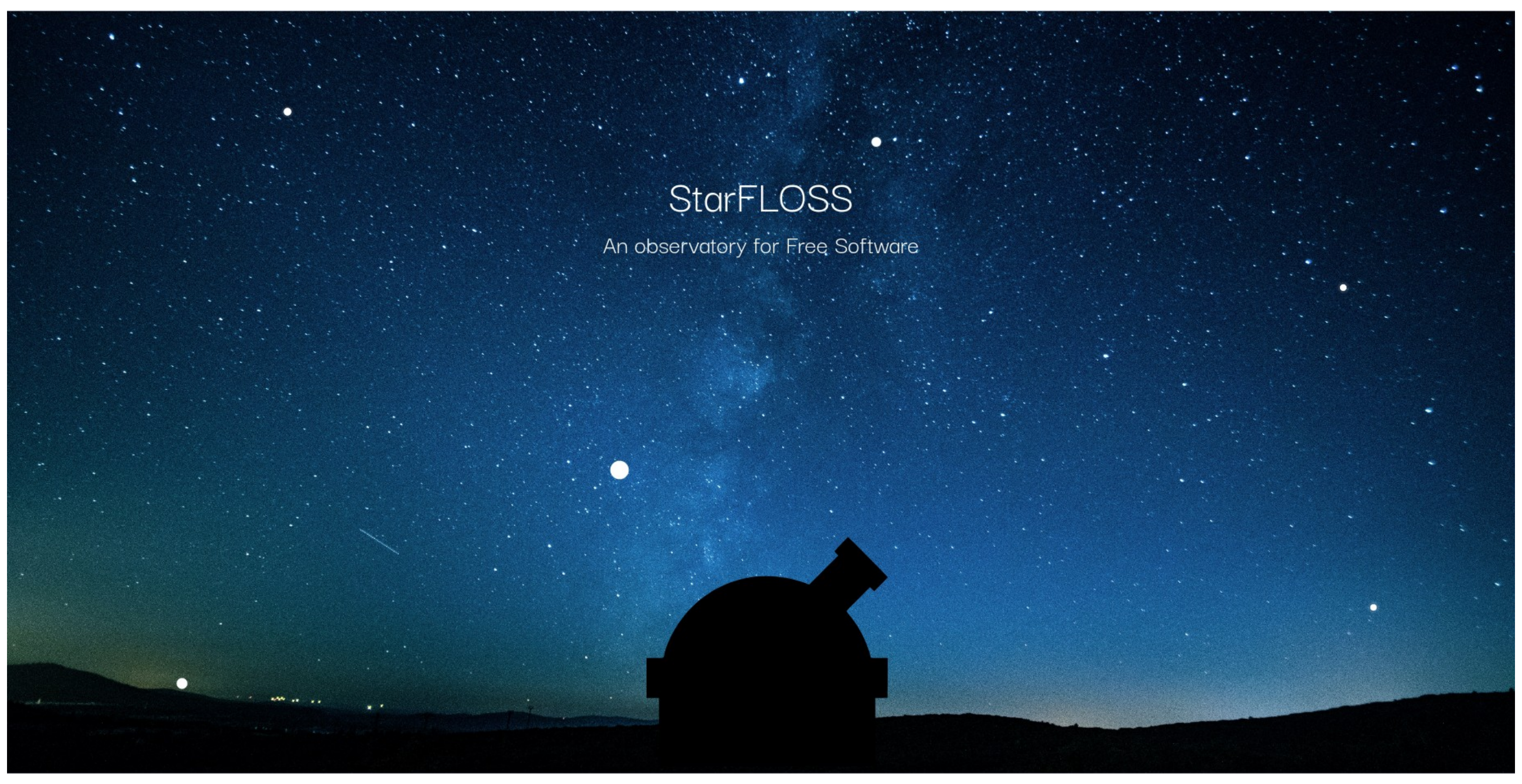
An Observatory of FLOSS Communities

Camila Naomi Kodaira, Felipe Caetano Silva
Advisor: Paulo Meirelles **Co-advisor:** Melissa Wen

"GIVEN ENOUGH BOTS, ALL BEHAVIOURS ARE DOCUMENTED AND VISUALIZED."

StarFLOSS is a website whose goal is to be an observatory of the Free/Libre/Open Source Software community. It allows users to register communication channels and visualize a dashboard with information about the status of their community.

https://starfloss.herokuapp.com/



StarFLOSS
An observatory for Free Software

INTRODUCTION

Free/Libre/Open Source Software (FLOSS) is an umbrella term for software in which users are free to inspect, use, change, and distribute it [1]. For over two decades, investigations have sought to describe better the practices surrounding a successful FLOSS project.

In 1999, Raymond published "The Cathedral and the Bazaar", defining the development model o free software as a Bazaar [2]. In this seminal essay, the author coined the following statement: "Given enough eyeballs, all bugs are shallow", which became known as the "Linus Law". A crucial aspect of FLOSS is how structured are their ways of deciding the north of the project. Many channels of communication are used, naming Mail lists, Version control Repositories, IRC channels, bug trackers, Websites, and Wikis [3].

In this context, the StarFLOSS project wants to observe how FLOSS communities work. By using some of their channels to gather information and cohesively present them, we want to create a tool that allows any person interested to understand the inner workings of any given community in a fun and easy way.

FLOSS essential freedoms:

> The freedom to run the program as you wish, for any purpose.

> The freedom to study how the program works, and change it so it does your computing as you wish.

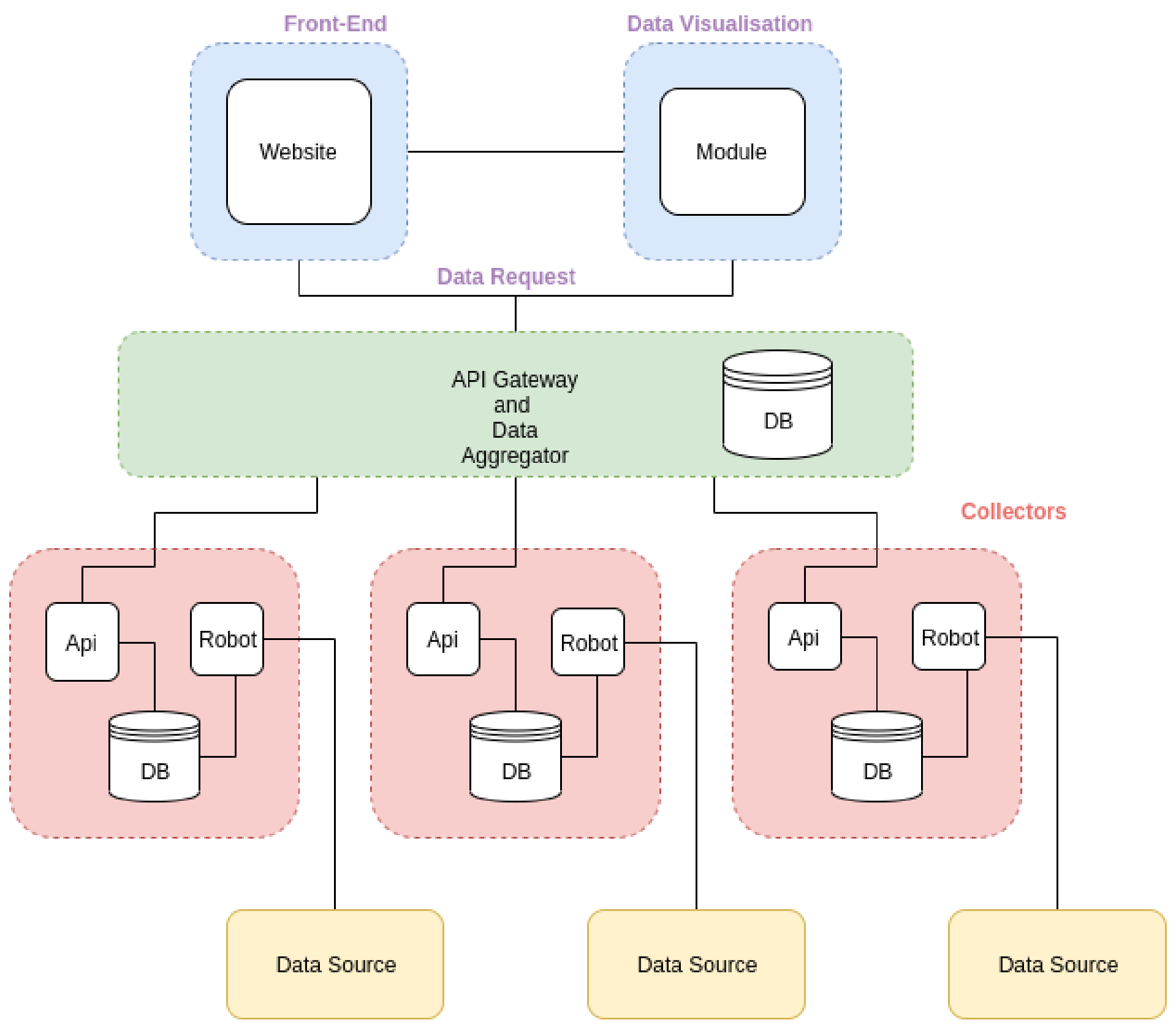
> The freedom to redistribute copies so you can help others.

> The freedom to distribute copies of your modified versions to others.

INTEGRATION LAYER

The Integration layer is represented by the API Gateway and Data Aggregator, which has the responsibility to handle the requests made to the services and filter the information according to the client. The Collectors in the picture represent the layer of micro-services used to collect its information from the Data sources.

This architecture was based mainly on Service-Oriented architecture. This style of software design and its central principles are independent of vendors, products, and technologies. A service is defined as a self-contained unit, it is a black box for the client, and it may serve other services.



StarFLOSS architecture

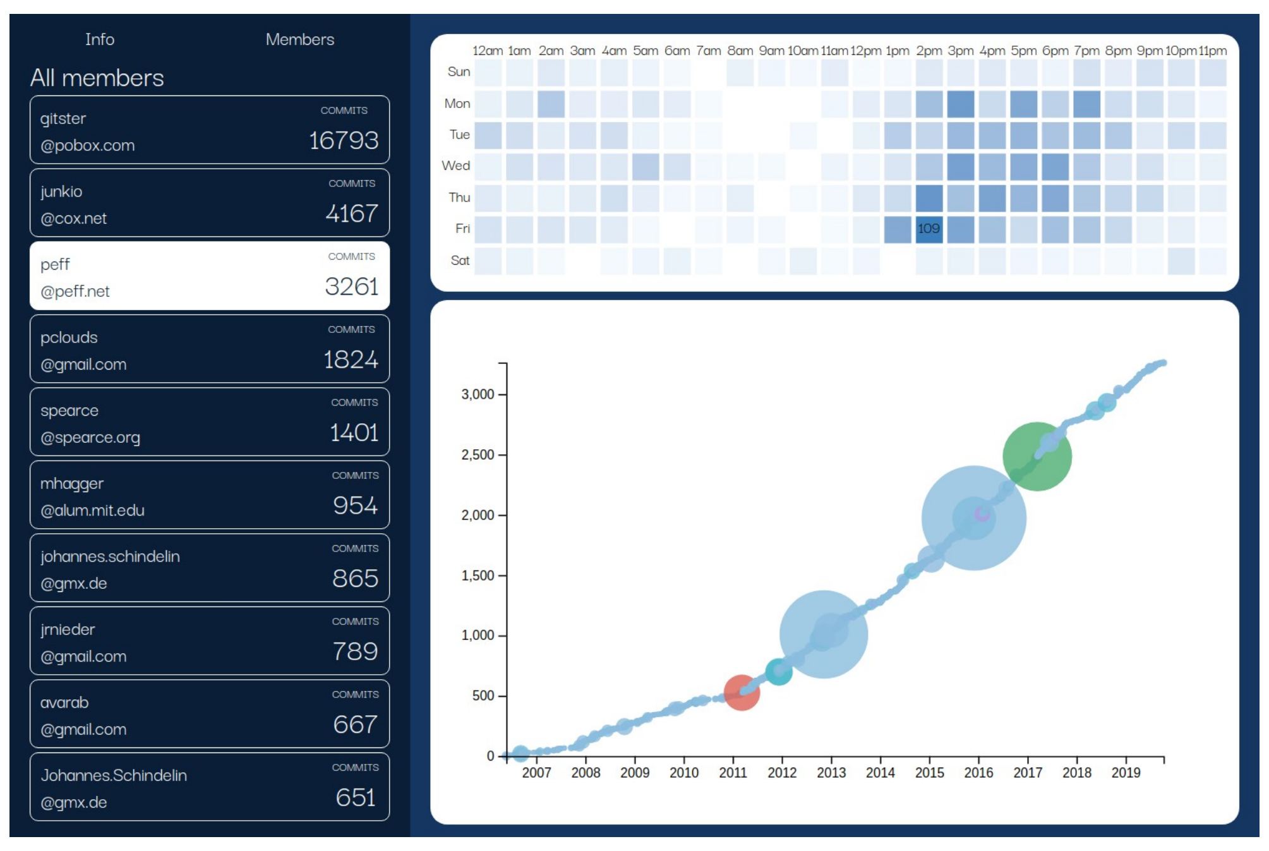
A good API must have a good documentation in order to make it easy for any user, even if it is not a contributor of the project, to consume its endpoints. Api pie-rails is described as a Rails engine for documentation of Restful API, without the usage of traditional comments in the code, this library allows documentation with ruby code.

WEBSITE

The StarFLOSS website is the first thing the user will see when they discover this project. It will inform their first impression, and as such, beautiful to view and use.

StarFLOSS design was carefully planned to be visually pleasing and easy to use. Every part of the dashboard is interactive; selecting any part of the charts and menus will filter the presented data to illustrate that selection.

It was created by using React as the framework for the website and D3 as the library to aid in drawing the graphs. All components were hand-made by using HTML and CSS.



Info Members

All members

gitster @pobox.com	COMMITS 16793
junkio @cox.net	4167
peff @peff.net	3261
pclouds @gmail.com	1824
spearce @spearce.org	1401
mhagger @alum.mit.edu	954
johannes.schindelin @gmx.de	865
jrnieder @gmail.com	789
avarab @gmail.com	667
Johannes.Schindelin @gmx.de	651

StarFLOSS dashboard. Peff's was selected so the commits shown on the graphs are only his.

CONSIDERATIONS

StarFLOSS started looking at a place very different from where it ended; it had to be rebooted after a failed attempt and had a tight production schedule. Throughout the year, we as developers grew a lot by learning new concepts and tools. As we become better, we can easily see the mistakes made on the course of this project.

[1] Crowston K., Wei K., Howison K., Wiggins A. "Free/Libre Open-source Software Development". ACM Computing Surveys 44.2, Feb. 2012.
[2] Raymond E. "The Cathedral and the Bazaar". Knowledge, Technology and Policy 12.3, Sept. 1999
[3] Fogel K. "Producing Open Source Software: How to Run a Successful Free Software Project". Second. O'Reilly Media, Jan. 2017.