

---

# Soft Dive Into GrimoireLab

CHAOSCon North America - San Diego - Ago. 2019

Manrique Lopez / Santiago Dueñas  
@bitergia  
[jsmanrique@bitergia.com](mailto:jsmanrique@bitergia.com) / [sduenas@bitergia.com](mailto:sduenas@bitergia.com)



# GrimoireLab ... what is it?

The free, open source toolkit to answer your questions about the community and processes involved in software development

And, if it fails, or you know how to improve it, feel free to [open an issue](#), [send a pull request](#), or [discuss about it in the mailing list](#) ;-)



-

What are your  
questions?

*Tip:*

Don't start with the questions.

Start with the goals

**Think strategically!**

# Goals - Questions - Metrics

## Goal

We want these projects to be community driven and not ruled by a single company

## Questions

How many organizations are participating or have participated?

Is this a community driven project?

## Metrics

Organizational diversity (CHAOSS Metric)

Gmail Factor: % of commits, and authors, from gmail.com accounts

# CHAOSS Metrics

[chaoss.community/metrics](https://chaoss.community/metrics)



**COWABUNGA!**

# Our own questions

How many Bitergians have contributed to CHAOSS working groups?

What is the responsiveness to CHAOSS community pull requests in GrimoireLab?

Is this a contributor driven community or a company driven community?

# Workshop Materials

You need **git** and **Docker** installed

... and *some* GBs of RAM and HDD

Extra: Python might help if you wanna run some scripts locally

[gitlab.com/jsmanrique/grimoirelab-workshop](https://gitlab.com/jsmanrique/grimoirelab-workshop)

```
~/$ git clone https://gitlab.com/Bitergia/lab/analytics-demo  
~/$ cd analytics-demo  
~/analytics-demo $
```

-

What do you want to  
analyze?

# What's in a “project”?

Data sources

Repositories (*of data*)

Granted access to repositories

# projects.json

```
{  
    "grimoirelab": {  
        "meta": {  
            "title": "...",  
            "logo_url": "...",  
            "..."  
        },  
        "git": [...],  
        "github": [...],  
        ...  
    },  
    ...  
}
```

# setup.cfg

```
...  
[github]  
api-token = <YOUR_API_TOKEN_HERE>  
raw_index = github_demo_raw  
enriched_index = github_demo_enriched  
...
```

```
~/analytics-demo $ docker-compose up -d
```

1, 2, 3 ..., 100 (or *coffee time*)

**Visit** [http://<localhost\\_or\\_your-server-IP>:5601](http://<localhost_or_your-server-IP>:5601)





***What a piece of junk!***

GIFSec.com

What's going on?

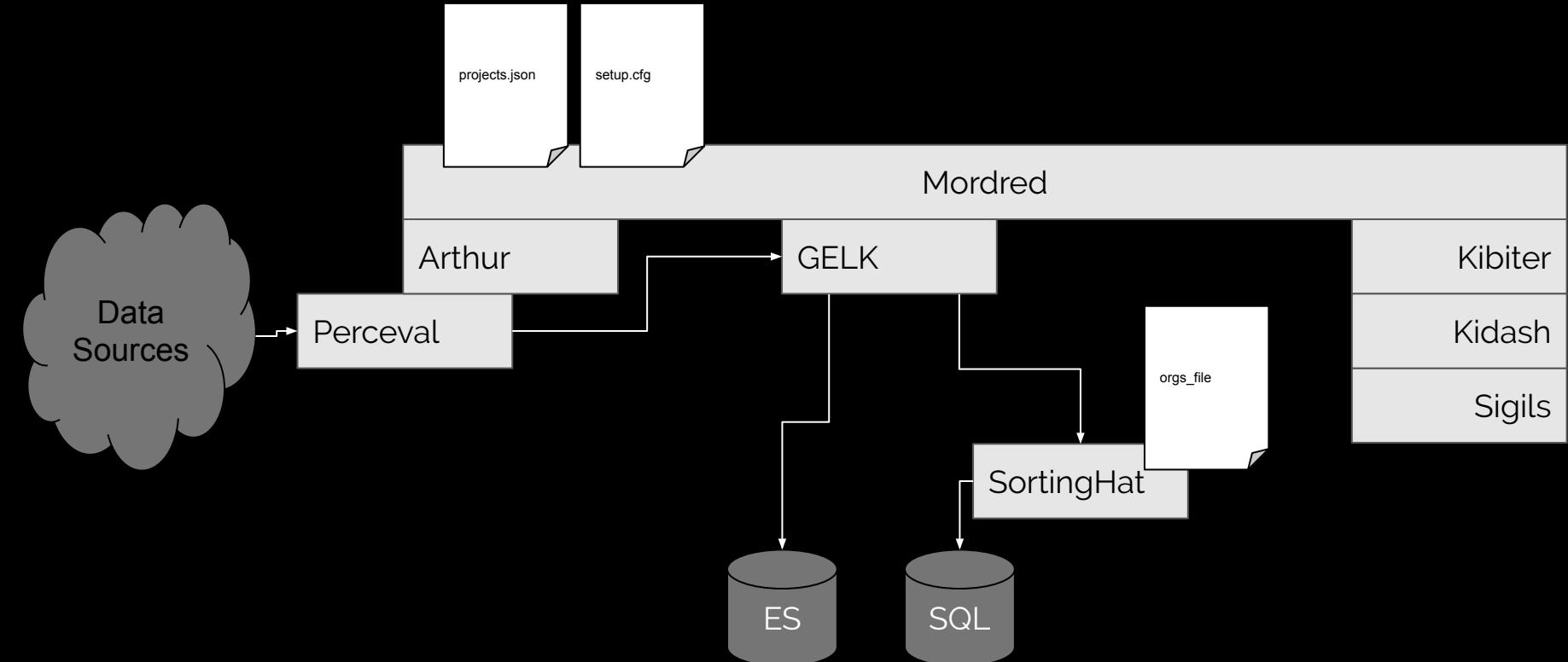
-

# GrimoireLab Data Workflow

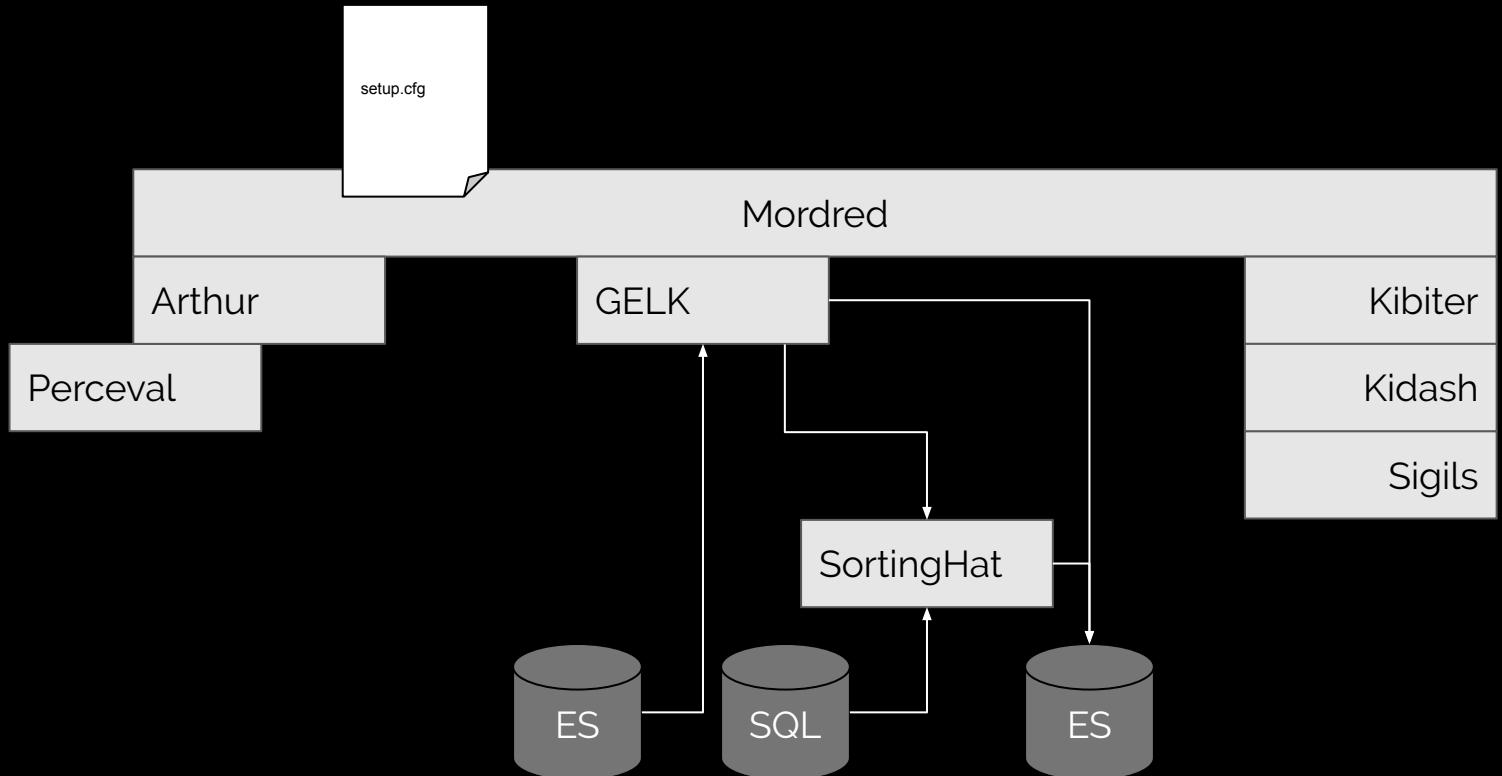
# What's inside GrimoireLab?

Mordred		
Arthur	GELK	Kibiter
Perceval	SortingHat	Kidash
Sigils		
Manuscripts		

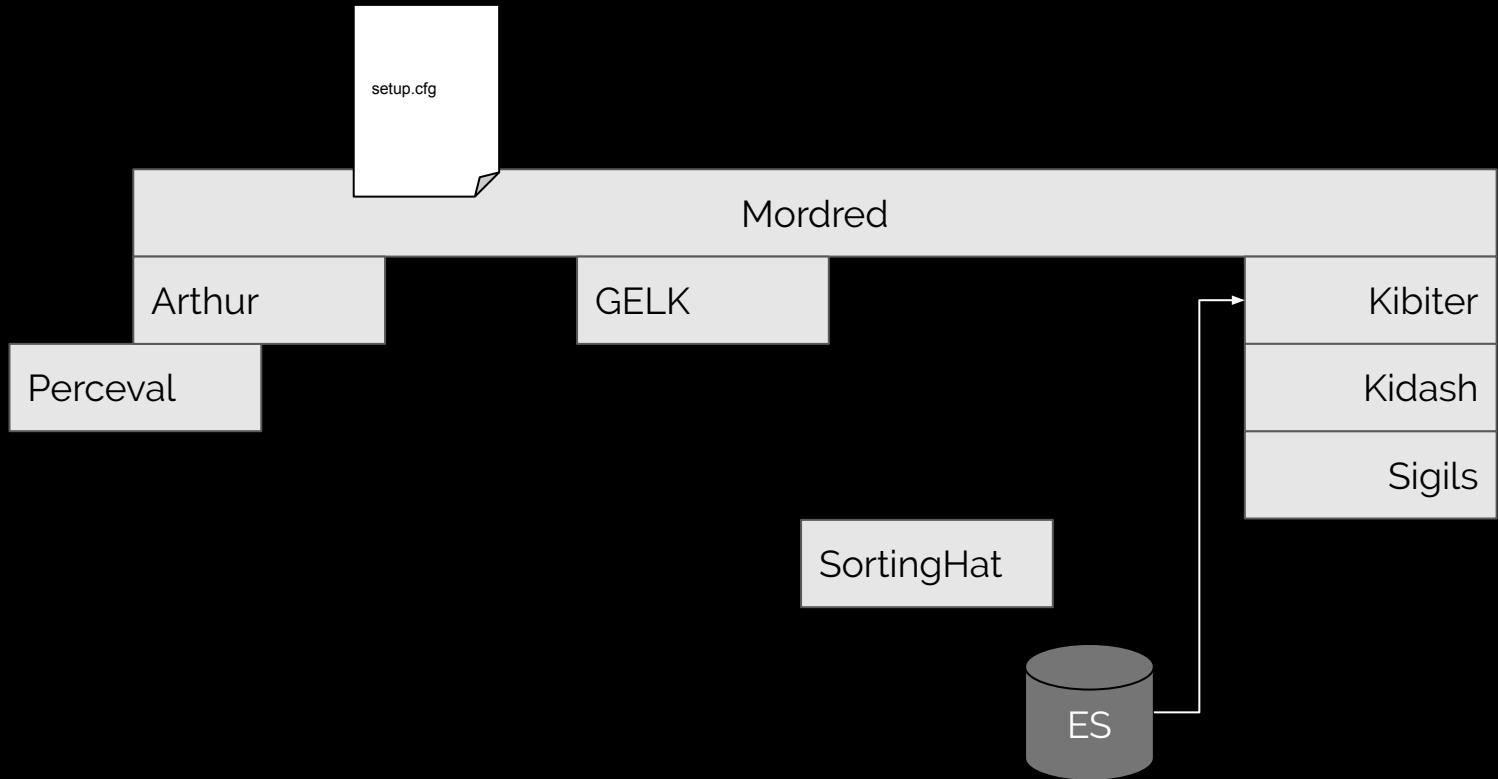
1<sup>st</sup> step: **Gathering data**



## 2<sup>nd</sup> step: Data Enrichment



## 3<sup>rd</sup> step: **Data Consumption**



**CHAOSS**

**Dashboard / Engagement: Contributions overview**

Search... (e.g. status:200 AND extension:PHP)

NOT RISK Add a filter +

Filters

Data Source Select... Organizations Projects Contributors Select... Main Numbers

**14,293** Contributions

**361** Contributors

Engagement: Contributions overview

- This panel focuses on **Engagement** in terms of **number of contributions**.
- Heat map allows to find most active organizations and which projects they are targeting their contributions to.
- Bar charts show evolution of contributions in total and stacked by data source and allows to identify the general trend and local trends based on moving averages.
- Panel also includes a section on **Contributor Activity** and **Contributor Growth**.

Organization Contributions to Projects

Contributions

Contributions by Data Source

**CHAOSS**

**Dashboard / Contributors Growth**

Search... (e.g. status:200 AND extension:PHP)

NOT RISK Add a filter +

Filters

Data Source Select... Organizations About

Total Contributors **361**

Contributors Growth by Data Source

Source 1 mail 138  
b\_issues 130  
b\_pull\_requests 126  
b\_pr\_requests 119

Contributors

Search

**CHAOSS**

**Dashboard / Efficiency: Timing overview**

Search... (e.g. status:200 AND extension:PHP)

Closed Tickets With Time to Close With Time to First Attention NOT RISK Add a filter +

Efficiency: Timing Overview

Data Source Select... Organizations Projects Select...

Avg. Time to Attend and Close Tickets

Avg. Time to First Attention

Avg. Time to Close

Median Time to First Attention

Median Time to Close

Efficiency: Timing overview

This panel focuses on time to first attention and time to close tickets.

Bar chart compares the difference in average between first attention and closing time of tickets created in a given time frame. Ideal case would be low values for both, which means contributors are attended soon and tickets are closed in a reasonable time. Worst case would have long bars for both, which means customers have to wait a long time to get our first attention and also closing time is of course longer than desired.

Repositories (Time to Close in Days)

Repository	Median	Avg.
https://github.com/chaoslab/treemap	51.76	51.76
https://github.com/chaoslab/metric	90.29	95.409
https://github.com/chaoslab/diversity-inclusion	33.95	40.753
https://github.com/chaoslab/web-site	10.84	18.073
https://github.com/chaoslab/prospector	6.65	6.65
https://github.com/chaoslab/metrics	35.09	35.245
https://github.com/chaoslab/gimme-a-tutorial	48.145	82.205
https://github.com/chaoslab/gimme-a-singraph	3.11	53.374
https://github.com/chaoslab/simordred	62.99	80.577
https://github.com/chaoslab/signts	20.97	79.281

Report: Raw ▲ Formatted ▲

1 2 3 \*

Active Contributors over time and Growth Analysis

361 Total Contributors

Active Contributors over time and Difference with the Average

# GrimoireLab Components

## Note:

This is not a microservices  
architecture, yet :-)

Have fun playing with them!

# Before starting

NOTE: To avoid installing components one by one, and to speed up training:

```
$ python3 -m venv /tmp/grimoirelab
```

```
$ source /tmp/grimoirelab/bin/activate
```

```
(grimoirelab) $ pip install --upgrade pip  
setuptools wheel
```

```
...
```

```
(grimoirelab) $ pip install grimoirelab
```

# Data Gathering

# Bestiary

Work in progress: [github.com/chaoss/grimoirelab-bestiary](https://github.com/chaoss/grimoirelab-bestiary)

The screenshot shows a web application titled "Bestiary Editor" running at [localhost:8000/editor\\_select\\_ecosystem](http://localhost:8000/editor_select_ecosystem). The interface includes a header with back, forward, and home buttons, a search bar, and a toolbar with import and download buttons. The main area has tabs for "Ecosystem:" (set to "GrimoireLab") and "Editor". Below these are sections for "Projects" and "Data Sources". A modal window titled "Data source for in GrimoireLab" is open, showing a dropdown menu for "Data source type" set to "URL", and a list of various data sources including askbot, bugzilla, bugzillarest, confluence, crates, discourse, dockerhub, functest, gerit, git, github, gmane, hyperkitty, jenkins, jira, kitsune, mbox, mediawiki, and meetup. At the bottom of the modal are three "Select" buttons and a "Remove" button. The footer of the page says "Powered by GrimoireLab".

# Perceval

PyPI package:

```
$ pip3 install perceval
```

From sources:

```
$ git clone https://github.com/chaoss/grimoire-perceval.git
```

```
$ pip3 install -r requirements.txt
```

```
$ python3 setup.py install
```

# Perceval

<b>askbot</b>	Fetch questions and answers from Askbot site
<b>bugzilla</b>	Fetch bugs from a Bugzilla server
<b>bugzillarest</b>	Fetch bugs from a Bugzilla server (>=5.0) using its REST API
<b>confluence</b>	Fetch contents from a Confluence server
<b>discourse</b>	Fetch posts from Discourse site
<b>dockerhub</b>	Fetch repository data from Docker Hub site
<b>gerrit</b>	Fetch reviews from a Gerrit server
<b>git</b>	Fetch commits from Git
<b>github</b>	Fetch issues, pull requests and repository information from GitHub
<b>gitlab</b>	Fetch issues, merge requests from GitLab
<b>googlehits</b>	Fetch hits from Google API
<b>groupsio</b>	Fetch messages from Groups.io
<b>hyperkitty</b>	Fetch messages from a HyperKitty archiver
<b>jenkins</b>	Fetch builds from a Jenkins server
<b>jira</b>	Fetch issues from JIRA issue tracker
<b>launchpad</b>	Fetch issues from Launchpad issue tracker
<b>mattermost</b>	Fetch posts from a Mattermost server
<b>mbox</b>	Fetch messages from MBox files
<b>mediawiki</b>	Fetch pages and revisions from a MediaWiki site
<b>meetup</b>	Fetch events from a Meetup group
<b>nntp</b>	Fetch articles from a NNTP news group
<b>phabricator</b>	Fetch tasks from a Phabricator site
<b>pipermail</b>	Fetch messages from a Pipermail archiver
<b>redmine</b>	Fetch issues from a Redmine server
<b>rss</b>	Fetch entries from a RSS feed server
<b>slack</b>	Fetch messages from a Slack channel
<b>stackexchange</b>	Fetch questions from StackExchange sites
<b>supybot</b>	Fetch messages from Supybot log files
<b>telegram</b>	Fetch messages from the Telegram server
<b>twitter</b>	Fetch tweets from the Twitter Search API

# Perceval

From command line:

```
(perceval) $ perceval [-c <file>] [-g] \
<backend> [<args>] | --help | --version
```

In your Python code:

```
...
from perceval.backends.core.<backend> import <Backend>
...
backend_repo = <Backend>(<params>)
for item in backend_repo.fetch():
...

```

# Perceval

Write your own backends!

[github.com/chaoss/grimoirelab-perceval/tree/master/perceval/backends/core](https://github.com/chaoss/grimoirelab-perceval/tree/master/perceval/backends/core)

```
...
class <Backend>(Backend) :
...
@metadata
def fetch(self):
...
...
class <Backend>Client:
...
class <Backend>Command(BackendCommand) :
...
```



# GSOC: CHAOSS Metrics with Perceval

Aniruddha Karajgi

[github.com/Polaris000/GSoC\\_19\\_Perceval\\_Implementations](https://github.com/Polaris000/GSoC_19_Perceval_Implementations)

# Arthur (AKA KingArthur)

Work in progress: [github.com/chaoss/grimoirelab-kingarthur](https://github.com/chaoss/grimoirelab-kingarthur)

Scheduler for Perceval

# Citadel

Work in progress: [github.com/Bitergia/citadel](https://github.com/Bitergia/citadel)

# GSOC: Graal integration w GrimoireLab

Nishchith Shetty

[github.com/inishchith/gsoc](https://github.com/inishchith/gsoc)

Graal != GraalVM by Oracle

Graal leverages on the Git backend of Perceval and enhances it to set up ad-hoc source code analysis (code complexity, licensing, vulnerabilities, etc.)

More about Graal:

[github.com/chaoss/grimoirelab-graal](https://github.com/chaoss/grimoirelab-graal)

[blog.bitergia.com/2018/07/24/graal-the-quest-for-source-code-knowledge](https://blog.bitergia.com/2018/07/24/graal-the-quest-for-source-code-knowledge)

# Data Enrichment

# Sorting Hat

Maintains an SQL database with identities that can be merged in the same unique identity.

For each unique identity, a profile can be defined: name, email, and other data.

Each unique identity can be related to one or more affiliations, for different time periods.

[github.com/chaoss/grimoirelab-sortinghat](https://github.com/chaoss/grimoirelab-sortinghat)

# Sorting Hat

From command line:

```
(grimoirelab) $ sortinghat --help
```

```
...
```

```
(grimoirelab) $ sortinghat --host <DATABASE_IP>\  
                      --user root \  
                      --database <DATABASE_NAME> \  
<COMMAND> <PARAMETERS>
```

# Sorting Hat

In your Python code:

```
from sortinghat.db.database import Database
import sortinghat.api
from sortinghat.db.model import MIN_PERIOD_DATE,
MAX_PERIOD_DATE, \
    UniqueIdentity, Identity, Profile, Organization, Domain, \
Country, Enrollment, MatchingBlacklist

sortingshat_db_connection = Database(user=<DB_USER>, \
                                      password=<DB_PASS>, \
                                      database=<DB_NAME>, \
                                      host=<DB_HOST>)

sortingshat.api.<COMMAND>(sortingshat_db_connection, <PARAMETERS>)
```



# Hatstall

Work in progress: [github.com/chaoss/grimoirelab-hatstall](https://github.com/chaoss/grimoirelab-hatstall)

The screenshot shows a web browser window titled "Hatstall Profile Information" at the URL [localhost:8001/profiles/cc1000a28d57ed5351bb8a62dff5e37569b2df8/](http://localhost:8001/profiles/cc1000a28d57ed5351bb8a62dff5e37569b2df8/). The page displays profile information for a user named "J. Manrique Lopez de la Fuente".

**Profile info / cc1000a28d57ed5351bb8a62dff5e37569b2df8**

**Name:** J. Manrique Lopez de la Fuente  
**e-mail:** jsmanrique@bitergia.com

**Bot?** July 2013 (calendar dropdown)

**Country:** Su Mo Tu We Th Fr Sa  
30 1 2 3 4 5 6  
7 8 9 10 11 12 13  
14 15 16 17 18 19 20  
21 22 23 24 25 26 27  
28 29 30 31 1 2 3  
Organization: Bitergia

**Enrollment:** 2013-07-01 to 2100-01-01 (Update un-enroll buttons)

**Add** button for enrollment

**Profile Identitites**

Name	email	Username	Source
Manrique Lopez	None	jsmanrique	github unmerge
J. Manrique Lopez de la Fuente	jsmanrique@bitergia.com	None	git

**Add** button for identities

**Bottom section:** A large blacked-out area containing sensitive information.



# SortingHat GraphQL API

Work in progress: [github.com/chaoss/grimoirelab-sortinghat](https://github.com/chaoss/grimoirelab-sortinghat)

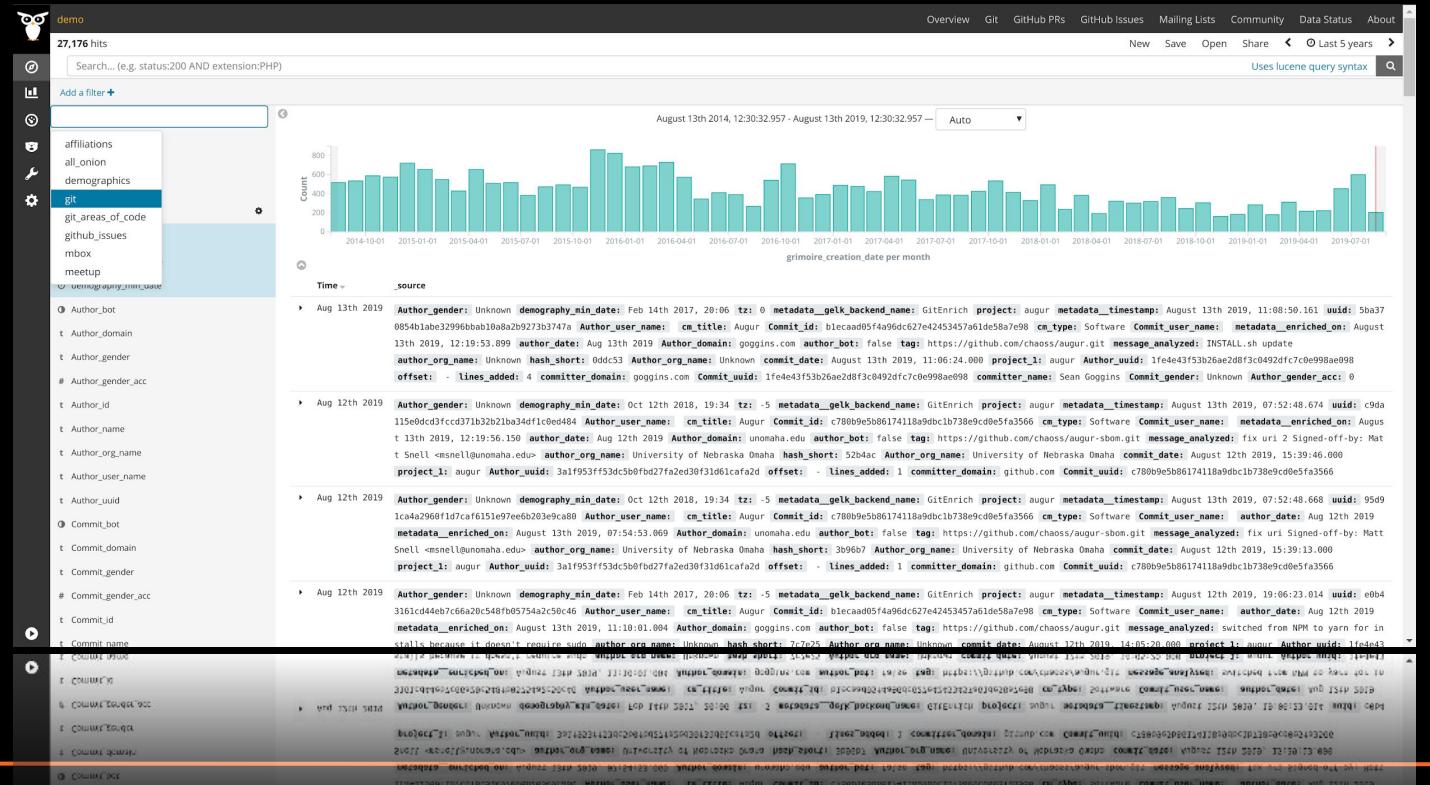
# Raistlin

Work in progress: [github.com/Bitergia/raistlin](https://github.com/Bitergia/raistlin)

# Data Consumption

# Data Schema

[github.com/chaoss/grimoirelab-elk/tree/master/schema](https://github.com/chaoss/grimoirelab-elk/tree/master/schema)



# Elasticsearch REST API

Python API

[elasticsearch-py.readthedocs.io](https://elasticsearch-py.readthedocs.io)

[elasticsearch-dsl.readthedocs.io](https://elasticsearch-dsl.readthedocs.io)

Javascript API

[www.elastic.co/guide/en/elasticsearch/client/javascript-api/current/index.html](https://www.elastic.co/guide/en/elasticsearch/client/javascript-api/current/index.html)



# Sigils

[chaoss.github.io/grimoirelab-sigils](https://chaoss.github.io/grimoirelab-sigils)

- **Lines of Code Changed:** panel focused on the number of lines of code changed.
- **Maintainer Response to Merge Request Duration:** panel focused on the time to response after a pull request took place.
- **Pull request merge duration:** This panel focuses on pull requests merge duration, defined by the time between code merge request and code commit.
- **Pull Requests Merged:** panel focused on the number of Pull Requests merged.

## GrimoireLab Sigils Panels

- **Community Structure by Organization:** view of a community grouping contributors by their activity, split by organizations.
- **Community Structure by Project:** view of a community grouping contributors by their activity, split by projects.
- **Data Status:** data freshness information.
- **Demographics:** attraction and retention of contributors.
- **Gerrit Review Efficiency:** efficiency closing reviews in Gerrit.
- **Git Demographics:** attraction and retention of developers specifically for Git.
- **Git:** metrics focused on Git commits.
- **GitHub Backlog:** focused on pending (open) tasks.
- **GitHub Issues Efficiency:** efficiency closing issues on GitHub.
- **GitHub Issues Timing:** metrics focused on how long issues remain open.
- **GitHub Issues:** activity and community metrics focused on Issues.
- **GitHub Pull Requests Efficiency:** efficiency closing Pull Requests.
- **GitHub Pull Requests Timing:** metrics focused on how long pull requests remain open.
- **GitHub Pull Requests:** activity and community metrics focused on Pull Requests.
- **GitHub Repositories:** metrics focused on repositories popularity.
- **GitLab Issues Efficiency:** efficiency closing issues on GitLab.
- **GitLab Merge Requests Efficiency:** efficiency solving (merging or closing) MRs on GitLab.
- **Jenkins Job Categories:** results of Jenkins job executions by category.
- **Jira:** metrics focused on Jira issues.
- **Lifecycle:** Level of activity in git repositories
- **MediaWiki:** contains metrics focused on reviews, including editions, revisions and editors.
- **Overall Community Structure:** overall view of a community grouping contributors by their activity.
- **Overview:** summary of basic metrics on all analyzed sources.



# Kidash

Tool to export & import GrimoireLab dashboards

```
(grimoirelab) $ kidash --help
```

```
...
```

```
(grimoirelab) $ kidash -e <Elasticsearch_IP> --list
```

# Archimedes

Work in progress: [github.com/Bitergia/archimedes](https://github.com/Bitergia/archimedes)

# Opendistro for Elasticsearch

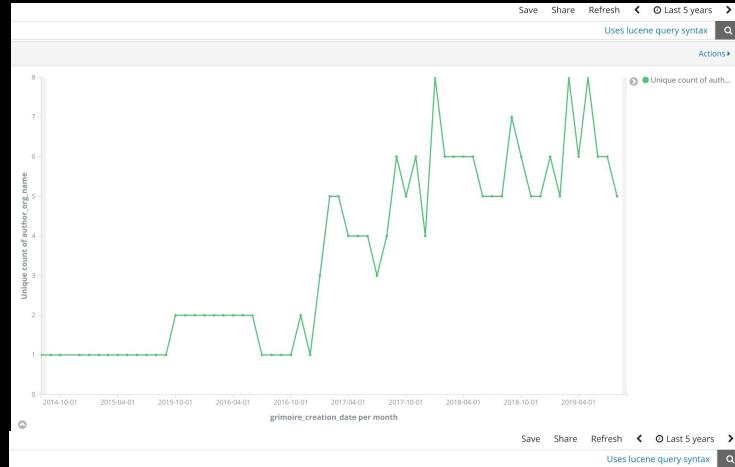
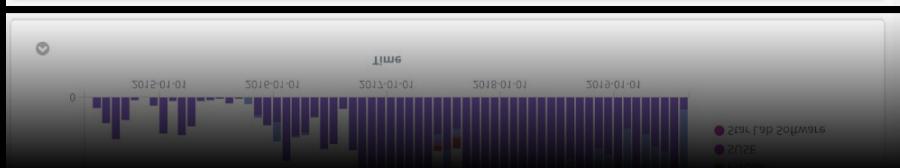
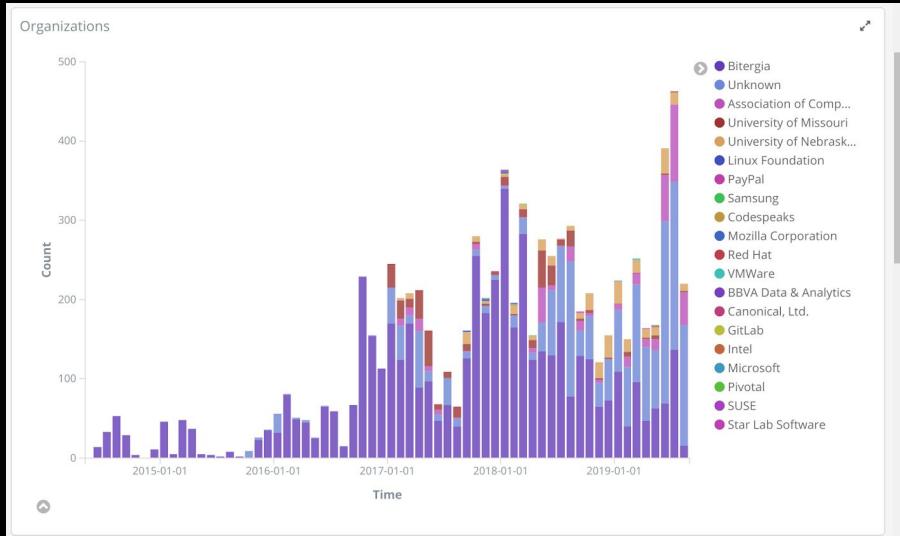
Being tested in [alpha.cauldron.io](https://alpha.cauldron.io)

More about that later ;-)

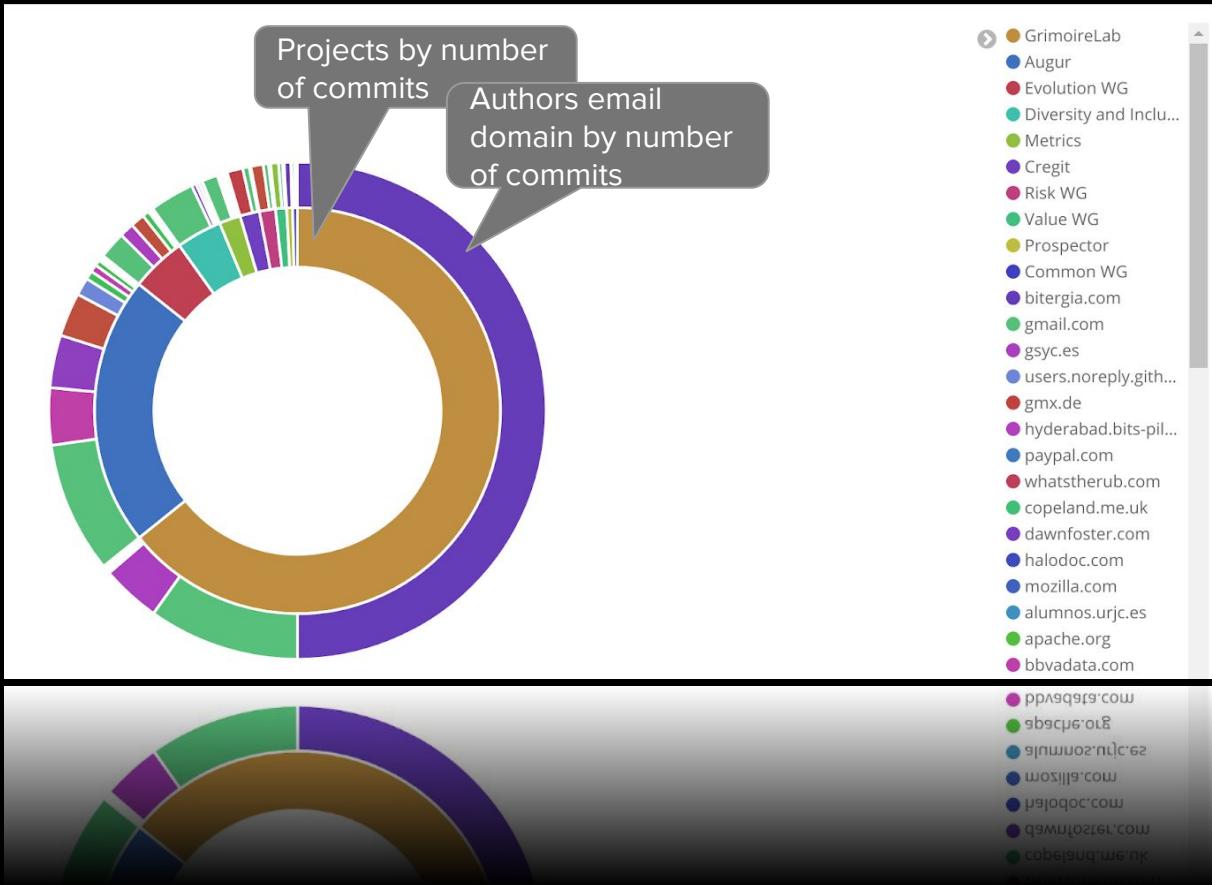
Let's answer  
questions

Is this a contributor driven community or a  
company driven community?

# Organizational Diversity



# GMail Factor



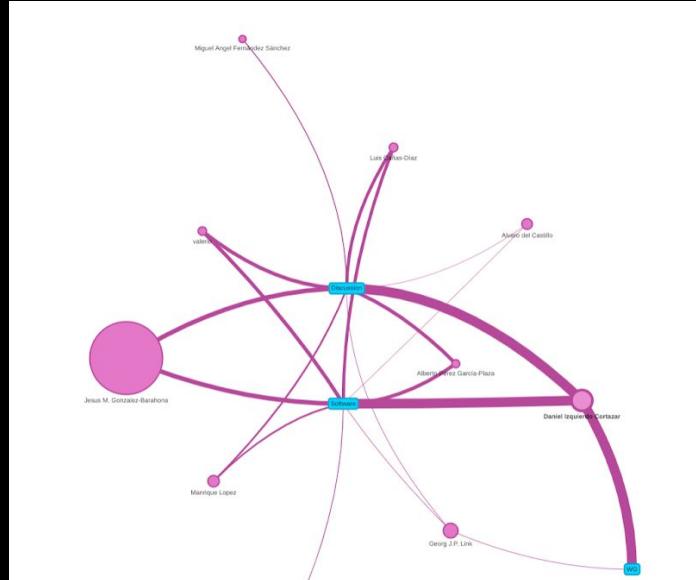
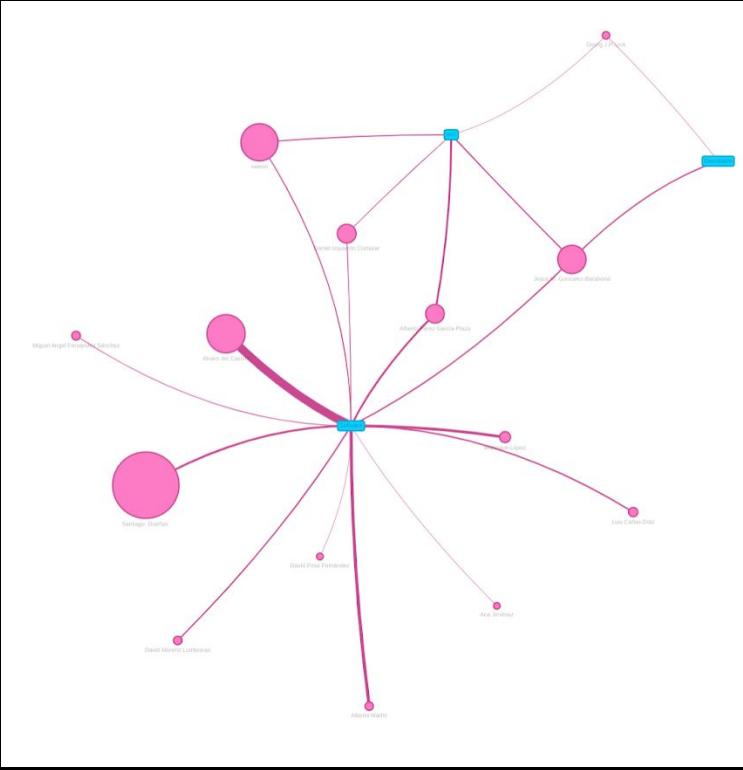
# CHAOSS Metrics

# Evolution - Code Development



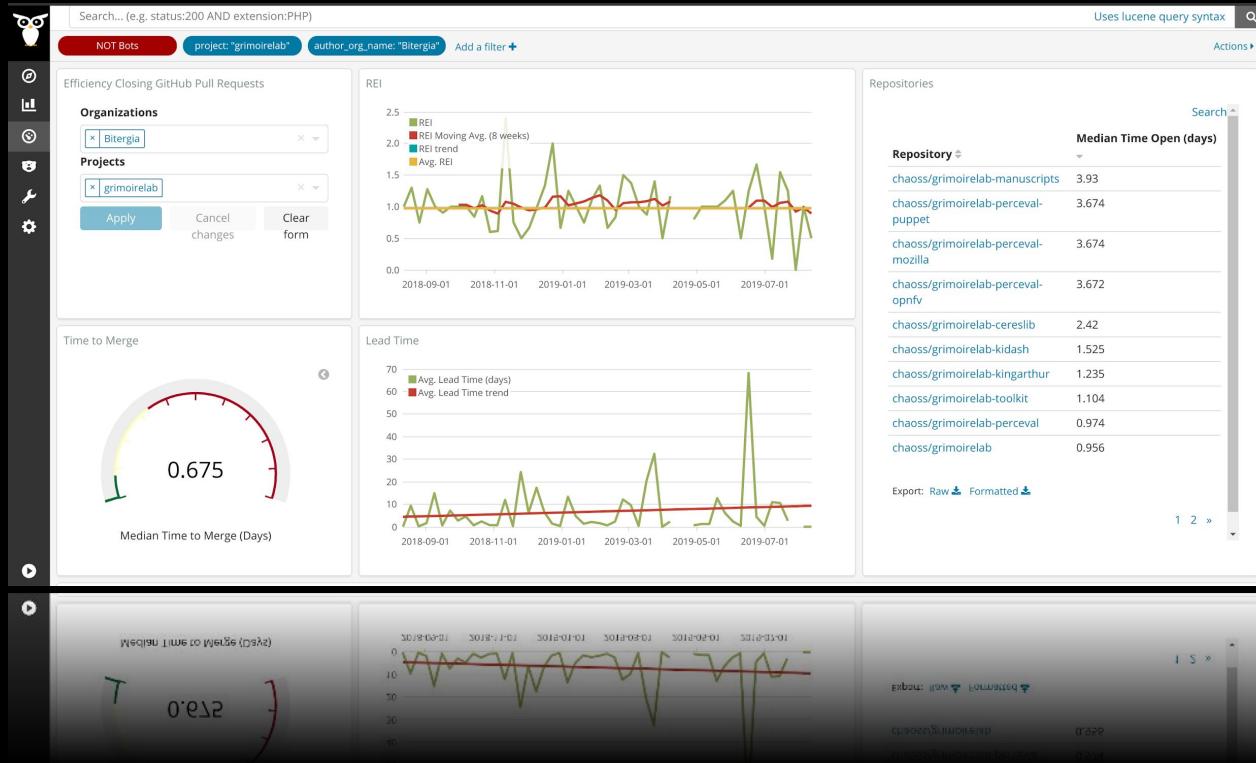
How many Bitergians have contributed to  
CHAOSS working groups?

# Network view

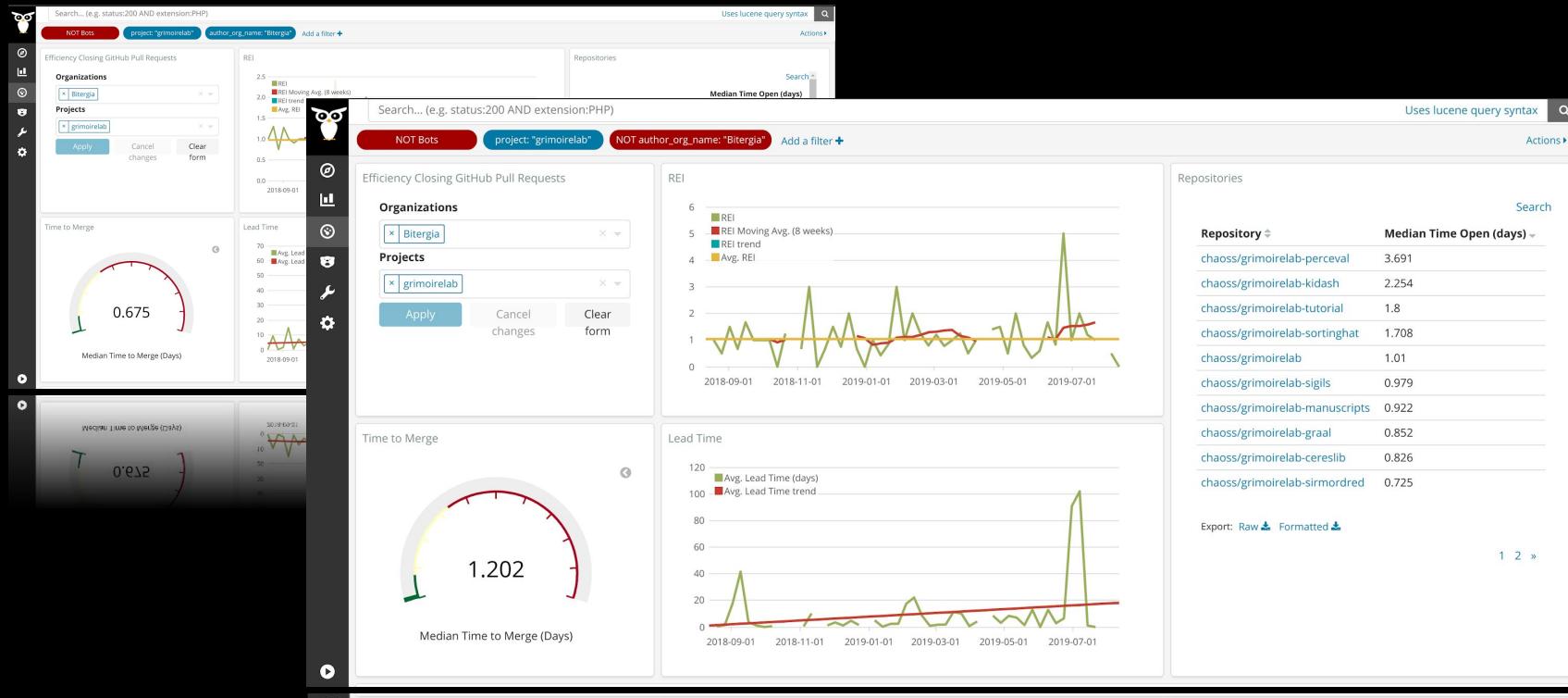


What is the responsiveness to CHAOSS  
community pull requests in GrimoireLab?

# Lead time, time to 1st response, etc.



# Lead time, time to 1st response, etc.



That is *almost* all ...

One more thing!

Cauldron (a version)

This is an alpha version of the Cauldron. All data can be removed without prior notice. Feedback is welcome!

## Welcome to Cauldron (a version)!

Create an analytics environment for the software development projects that matter to you!

The Cauldron is a PoC service developed by [Bitergia](#) to analyze community and processes in software development projects.

Analyze a project



Copyright © 2019 [Bitergium SLL](#), and its contributors.

Some Rights Reserved.

Unless otherwise stated, this work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 Unported License](#). Cauldron and the Cauldron logo are trademarks of [Bitergium SLL](#) in the European Union and/or other jurisdictions. All other trademarks are the property of their respective holders.

Made with  and free, open source software:



Software  
development  
environment  
for the  
analysis  
of open  
source  
communities



Software  
development  
environment  
for the  
analysis  
of open  
source  
communities



Software  
development  
environment  
for the  
analysis  
of open  
source  
communities

# Let's go for questions!

**Manrique López / Santiago Dueñas**

**CEO / CTO at Bitergia**

**[jsmanrique@bitergia.com](mailto:jsmanrique@bitergia.com) /  
[sduenas@bitergia.com](mailto:sduenas@bitergia.com)**



# About us

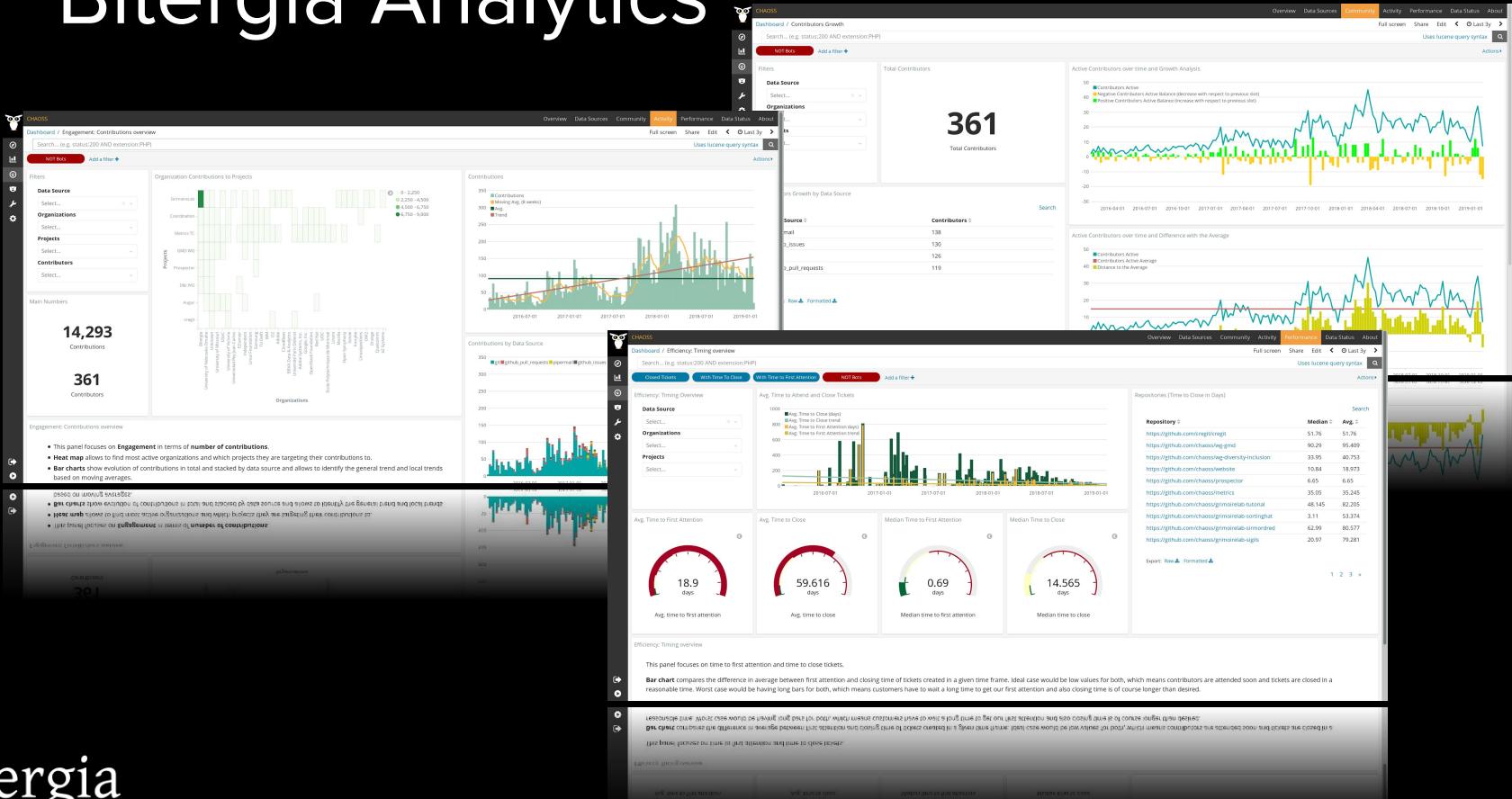
Bitergia

Software Development Analytics

For Your Peace of Mind

Bitergia helps companies and organizations with understanding and improving software development projects that matter to them

# Bitergia Analytics



## Community (who?)

- Who are the contributors to the analyzed projects?
- Where are my developers? Where do they come from?
- Who are my core, regular and casual developers?
- What's the talent rotation and retention level?

## Activity (what?)

- What is being done in the analyzed projects?
- How many active projects do I contribute to?
- What's developers engagement level?
- What is being modified and what's left untouched for too long?

## Performance (how?)

- How fast are projects analyzed performing?
- How are we dealing with issues and merge requests?
- Where are the bottlenecks?
- How are we dealing with the backlog?

Community  
(who?)

- Who are the contributors to the analyzed projects?
- Where are my developers? Where do they come from?
- Who are my core, regular and casual developers?
- What's the talent rotation and retention level?

Activity  
(what)

- Who are the active contributors to the analyzed projects?
- Who are the inactive projects do I contribute to?
- What's the developer engagement level?
- What's the backlog growth rate and what's left untouched for too long?

Performance  
(how?)

- How fast are projects analyzed performing?
- How are we dealing with issues and merge requests?
- Where are the bottlenecks?
- How are we dealing with the backlog?



—

# How we do it

## Bitergia Analytics Consultancy

## Bitergia Analytics Platform

### Strategy

### Analysis

### Customization

### Reporting

Bitergia outlines organization strategy around software development to achieve organization's business goals.

Bitergia defines the data sources, questions and associated metrics to measure that provide the insights about goals status.

Bitergia deploy and operates its analytics platform to gather the data needed to answer the questions and metrics defined.

Bitergia provides consistent reporting mechanisms, including dashboards, reports, and even data APIs for custom integrations.

## **Open Source Software Foundations**

Non-profit organizations managing open source projects.

## **Open Source Program Offices (OSPO)**

Managing their relation with the open source projects they depend on.

## **InnerSource Program Offices (ISPO)**

Adopting open source development practices internally.

## Open Source Software Foundations

Non-profit organizations managing open source projects.

- Transparency level up
- Organizational diversity
- Members engagement
- Fair play among *coope*titors
- Projects attraction and demographics
- Management board composition

## Open Source Program Offices (OSPO)

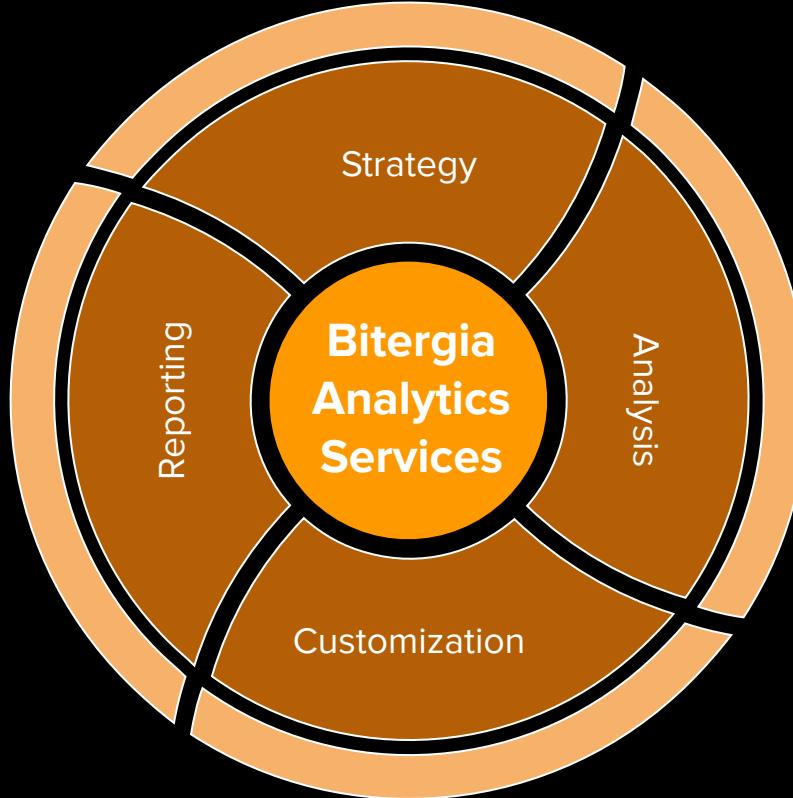
Managing their relation with the open source projects they depend on.

- Company OSS ecosystem
- Talent acquisition and retention
- Company footprint in OSS
- Consistent reporting mechanism

## InnerSource Program Offices (ISPO)

Adopting open source development practices internally.

- Developers engagement and talent retention
- Cross-Collaboration
- Onboarding mentoring
- Reuse and optimization



“To measure is to know”

“If you can not measure it,  
you cannot improve it”

Lord Kelvin

