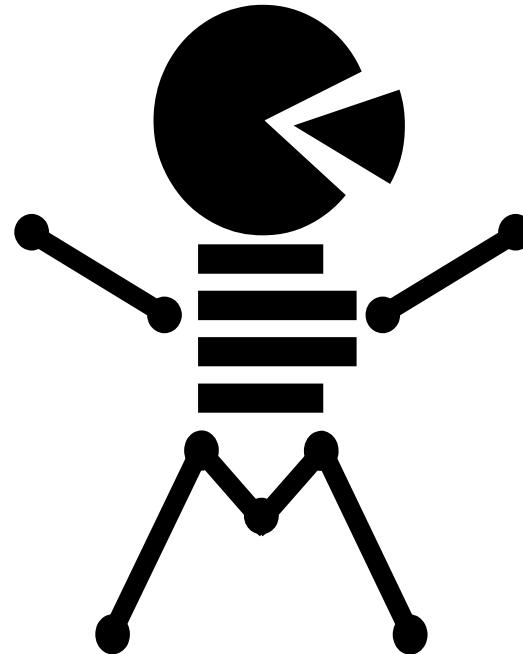


Q-DashMan

Data Visualization of Software Development



Created by Quan Zhou

Course: 2018-2019

Tutor: Dr. Jesús María González-Barahona

Main Technologies



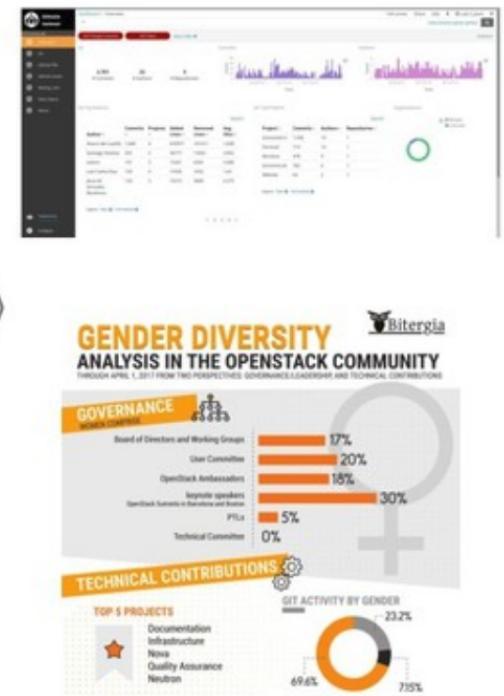
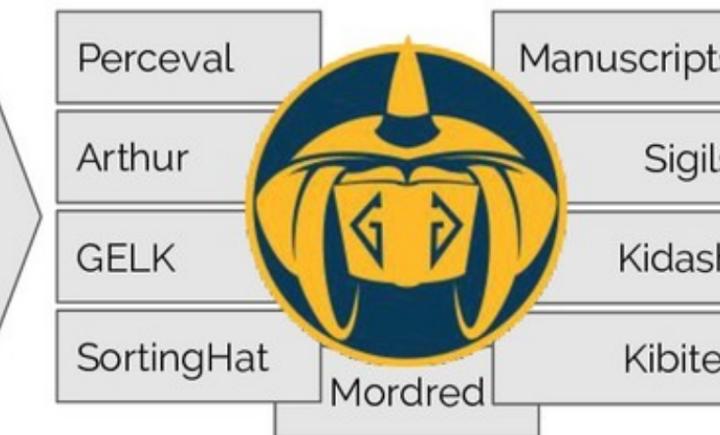
Data Sources



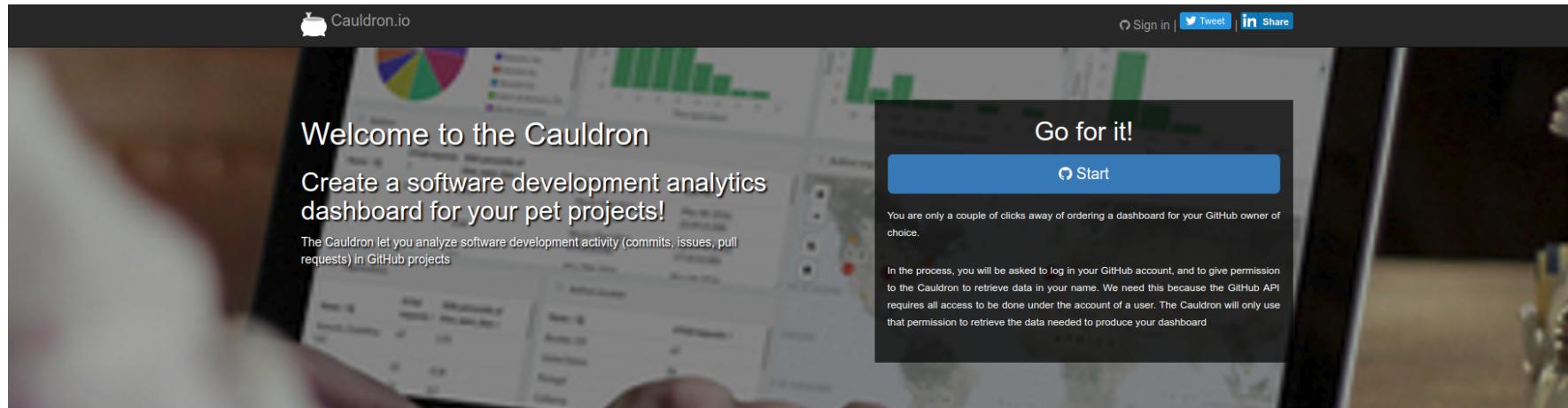
Metrics



GrimoireLab



Cauldron



About



The *Cauldron* is a proof of concept service for analyzing activity (git commits, issues, and pull requests) in GitHub projects.

Each GitHub user may ask the Cauldron for up to five dashboards, each corresponding to a GitHub owner (user or organization).



For each of GitHub owner, the Cauldron will retrieve data from up to 30 of the most recently active projects, and produce an interactive dashboard showing overall summaries, and providing capabilities for filtering and drilling down in the details.



The Cauldron is based on the [GrimoireLab](#) toolset. All the software used to produce it is free / open source software.



The Cauldron uses the [GitHub API](#) to retrieve information about issues and pull requests. Since the access to this API is [rate-limited](#), the retrieval of the data for your dashboard may take a while. Please, be patient.



If you want to browse more complete dashboards produced by [Bitergia](#), visit:

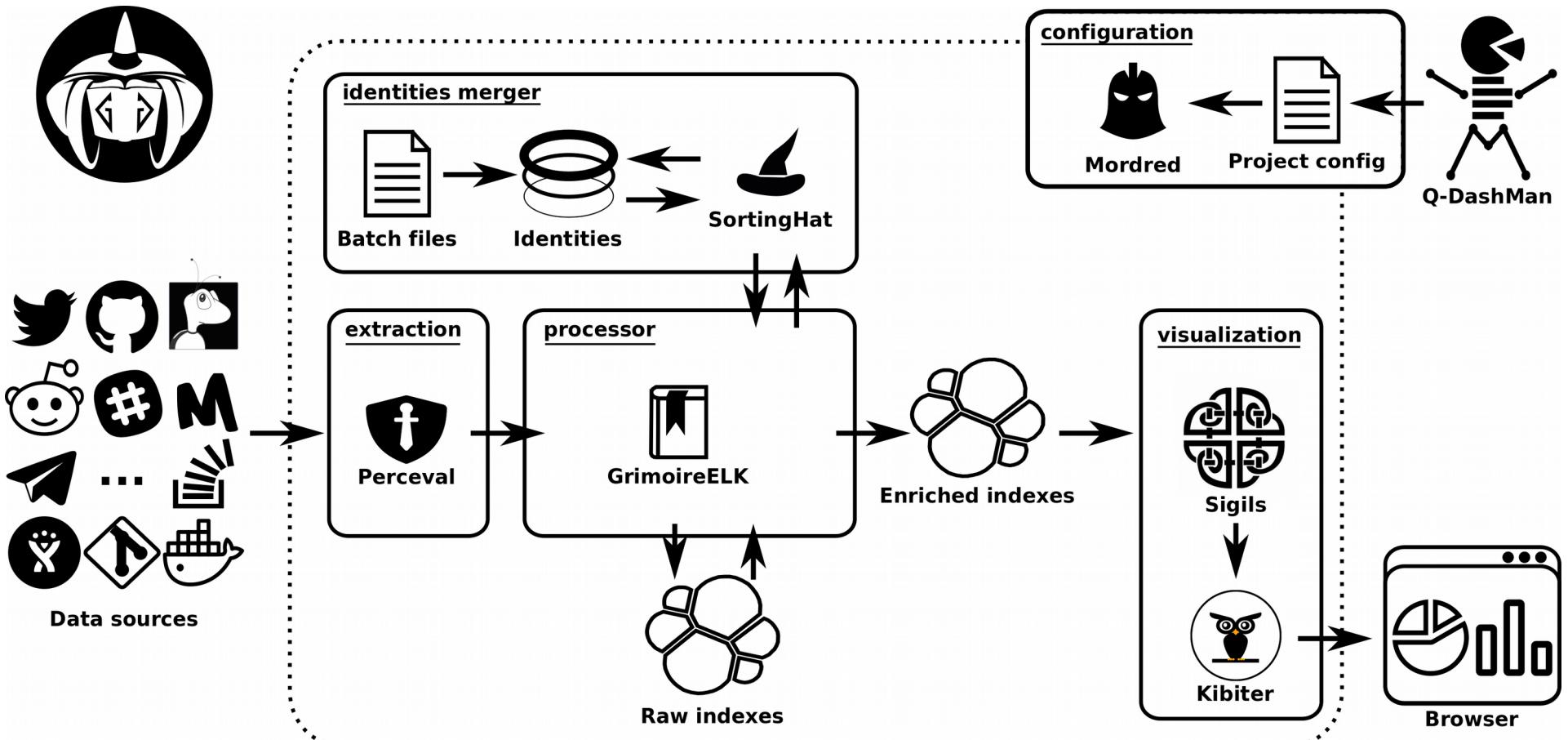
Search dashboards

Search for a project

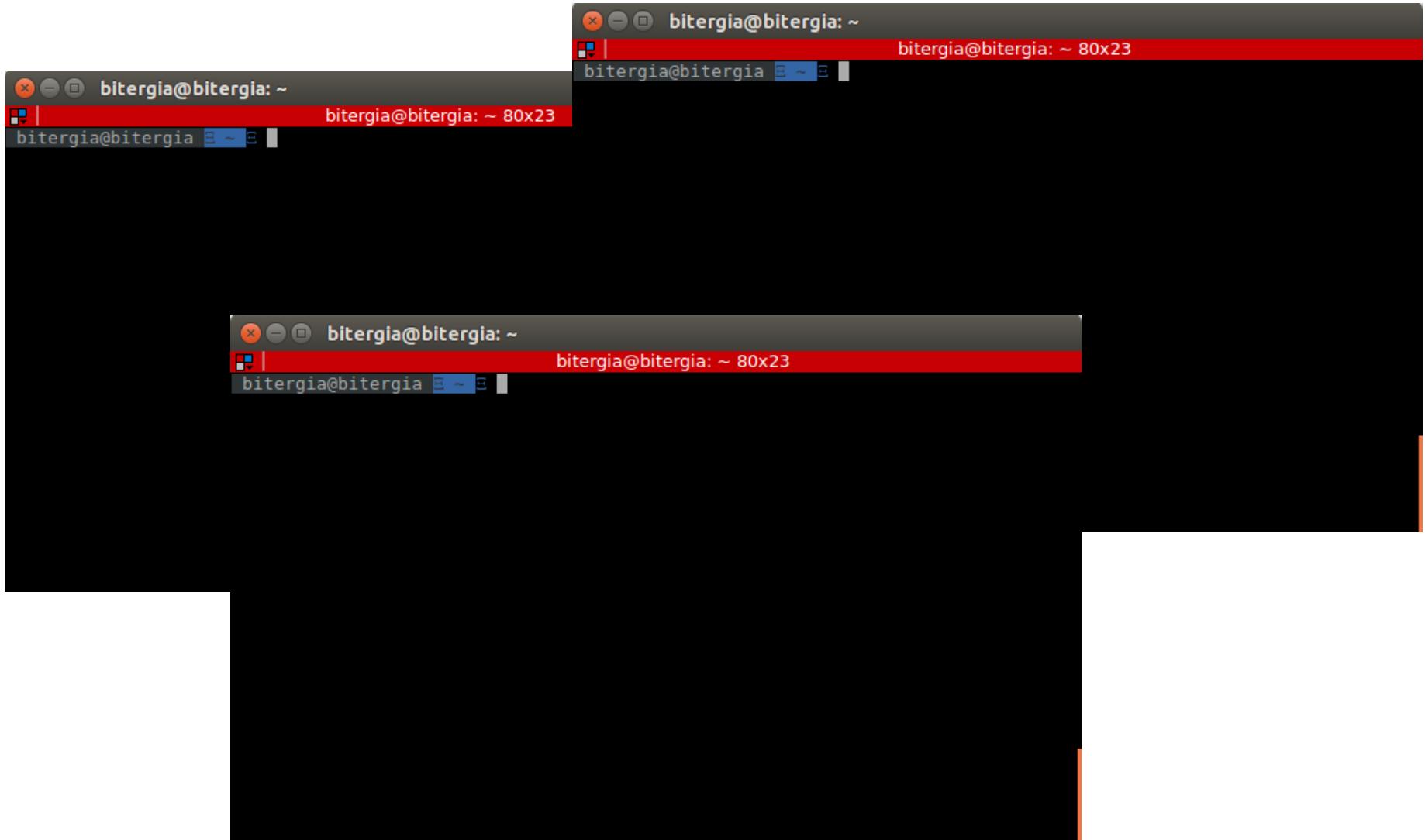
Popular Latest All dashboards

angular	
docker	
elastic	
facebook	
microsoft	

Q-DashMan



Shell



Projects File

Add projects.json

GitHub

Organization *

Token

Add

Slack

Project *

Token *

Add

Meetup

Project *

Topic *

Add

Manually

Project *

Data Source *

Repository *

Add

Close

This screenshot shows a modal window titled "Add projects.json". It contains four sections, each representing a different way to add projects:

- GitHub:** Fields for "Organization *" and "Token", with an "Add" button.
- Slack:** Fields for "Project *" and "Token *", with an "Add" button.
- Meetup:** Fields for "Project *" and "Topic *", with an "Add" button.
- Manually:** Fields for "Project *", "Data Source *", and "Repository *", with an "Add" button.

Each section has a "Close" button at the bottom right.

Import projects.json

Choose File No file chosen

Close Upload

This screenshot shows a modal window titled "Import projects.json". It contains a "Choose File" button with the message "No file chosen" next to it. At the bottom right are "Close" and "Upload" buttons.

Set Up File

General Settings

```
[general]
short_name:
    quan
update:
    true
min_update_delay:
    720
debug:
    true
logs_dir:
    /home/bitergia/logs

[projects]
projects_file:
    /home/bitergia/conf/sources/projects.json

[es_collection]
url:
    http://elasticsearch:9200

[es_enrichment]
url:
    http://elasticsearch:9200

autorefresh:
```

Add Data Source

-- select a Data Source--

- Raw Index *
- Enriched Index *
- No Archive
- Api Token
- Sleep Time
- Sleep For Rate
- Studies
- Latest Items

Import setup.cfg

Choose File No file chosen

The image shows three overlapping modal windows. The 'General Settings' window is at the bottom left, containing configuration options for [general], [projects], [es_collection], [es_enrichment], and autorefresh. The 'Add Data Source' window is in the center, listing various data source types like Raw Index, Enriched Index, and Sleep Time. The 'Import setup.cfg' window is at the bottom right, showing a file selection input field. All windows have standard UI elements like 'Close' and 'Save changes' buttons.

Show

projects.json

```
{  
  "zhquan": [  
    "github": [  
      "https://github.com/zhquan/DAT-Mashup",  
      "https://github.com/zhquan/DAT-PrincesGame",  
      "https://github.com/zhquan/ISI-Practica-7",  
      "https://github.com/zhquan/ISI-Practica-ChatAlien",  
      "https://github.com/zhquan/ISI-Practica-leaderboard",  
      "https://github.com/zhquan/ISI-Practica8",  
      "https://github.com/zhquan/owl",  
      "https://github.com/zhquan/PFG",  
      "https://github.com/zhquan/Practica-3",  
      "https://github.com/zhquan/PST-PeerChat",  
      "https://github.com/zhquan/qz-dashboard",  
      "https://github.com/zhquan/SAT-MiRevista",  
      "https://github.com/zhquan/TFG",  
      "https://github.com/zhquan/TFM",  
      "https://github.com/zhquan/videochat"  
    ],  
    "git": [  
      "https://github.com/zhquan/DAT-Mashup.git",  
      "https://github.com/zhquan/DAT-PrincesGame.git",  
      "https://github.com/zhquan/ISI-Practica-7.git",  
      "https://github.com/zhquan/ISI-Practica-ChatAlien.git",  
      "https://github.com/zhquan/ISI-Practica-leaderboard.git",  
      "https://github.com/zhquan/ISI-Practica8.git",  
      "https://github.com/zhquan/owl.git",  
      "https://github.com/zhquan/PFG.git",  
      "https://github.com/zhquan/Practica-3.git",  
      "https://github.com/zhquan/PST-PeerChat.git",  
      "https://github.com/zhquan/qz-dashboard.git",  
      "https://github.com/zhquan/SAT-MiRevista.git",  
      "https://github.com/zhquan/TFG.git",  
      "https://github.com/zhquan/TFM.git",  
      "https://github.com/zhquan/videochat.git"  
    ]  
  ]  
}
```

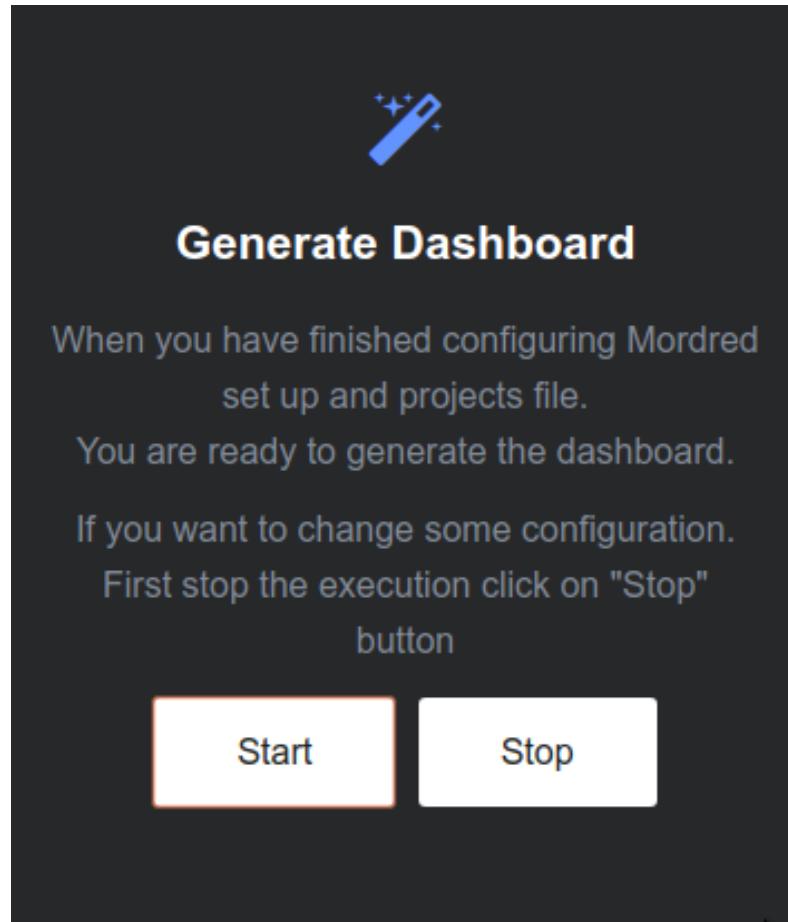
setup.cfg

```
[general]  
short_name = quan  
update = true  
min_update_delay = 720  
debug = true  
logs_dir = /home/bitergia/logs  
  
[projects]  
projects_file = /home/bitergia/conf/sources/projects.json  
  
[es_collection]  
url = http://elasticsearch:9200  
  
[es_enrichment]  
url = http://elasticsearch:9200  
autorefresh = true  
  
[sortinghat]  
host = mariadb  
user = root  
password =  
database = quan_sh  
load_orgs = false  
orgs_file = /home/bitergia/conf/orgs_file  
auto_profile = []  
matching = [email]  
sleep_for = 50  
bots_names = []  
unaffiliated_group = Unknown  
affiliate = true  
  
[phases]  
collection = true  
identities = true  
enrichment = true  
panels = true
```

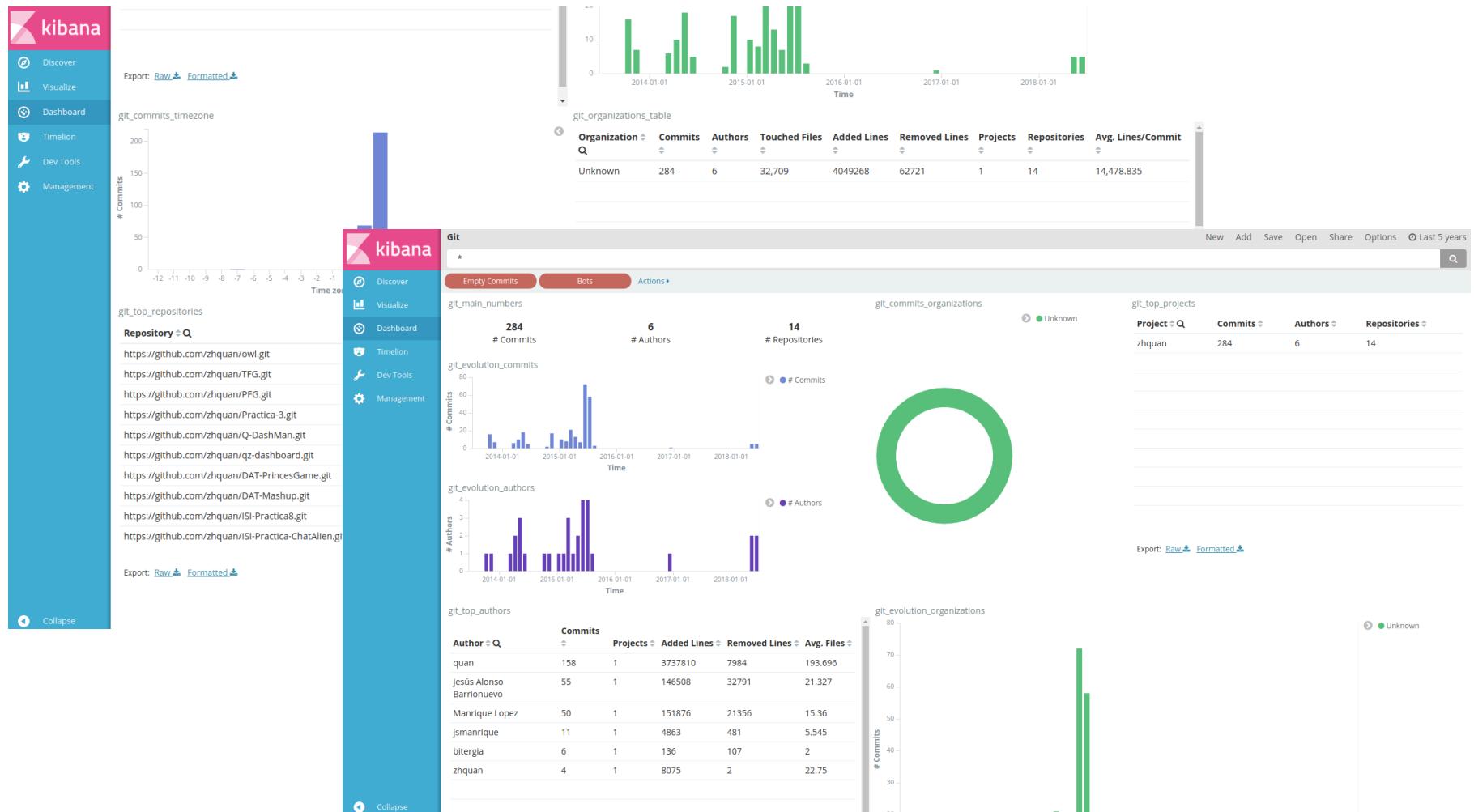
Close

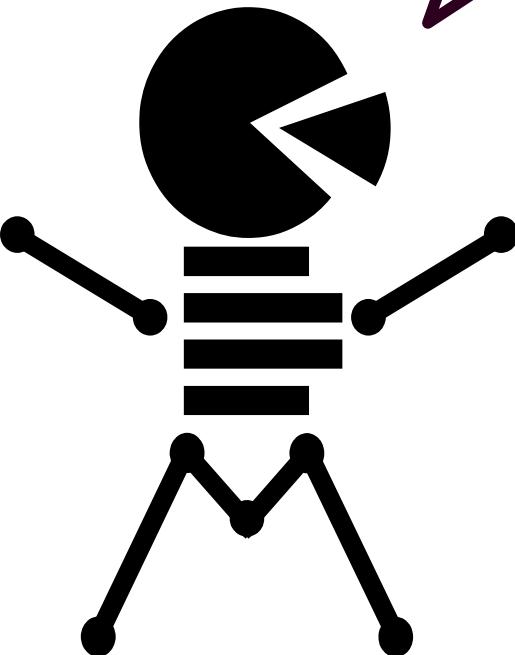
Close

Generate Dashboard



Dashboard





Let's do a demo



Thank you for your attention!!!

