
Perceval, Graal and Arthur

The Quest for Software Project Data

Open Source Summit North America,
Vancouver,
August 29th-31st, 2018

Jesus M. Gonzalez-Barahona, Santiago Dueñas, Valerio Cosentino
@jgbarah, @sduenasd, @_valcos_
[jgb, sduenas, valcos] at bitergia.com
<https://speakerdeck.com/bitergia>





Who are we?

CHAOSS is a Linux Foundation project focused on creating analytics and metrics to help define community health. Work in the CHAOSS Project community is organized in two committees and four workgroups:

Committees

- The [Metrics Committee](#) works on establishing implementation-agnostic metrics for measuring community activity, contributions, and health.
- The [Software Committee](#) works on producing integrated, open source software for analyzing software development, and definition of standards and models used in that software in specific use cases.

Workgroups

The goal of the workgroups is to refine the metrics and to work with software implementations. The workgroups are built around the four categories of metrics that CHAOSS has identified. The workgroups are:

- Diversity and Inclusion
 - Growth Maturity and Decline
 - Risk
 - Value
-

Software development analytics with

free, open source software

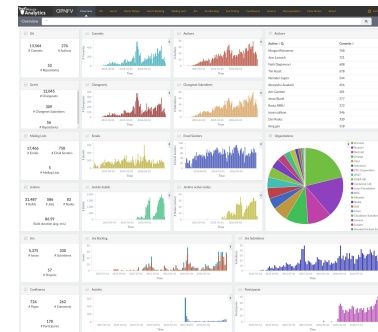
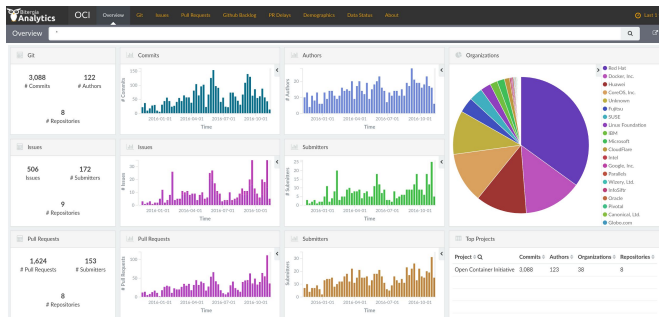
(a CHAOSS project)



- chaoss.github.io/grimoirelab
- chaoss.github.io/grimoirelab-tutorial

Software Development Analytics

for your peace of mind



Outline

Context

Perceval

Graal

Arthur

Exploitation

/context



 **Bitergia**

/context



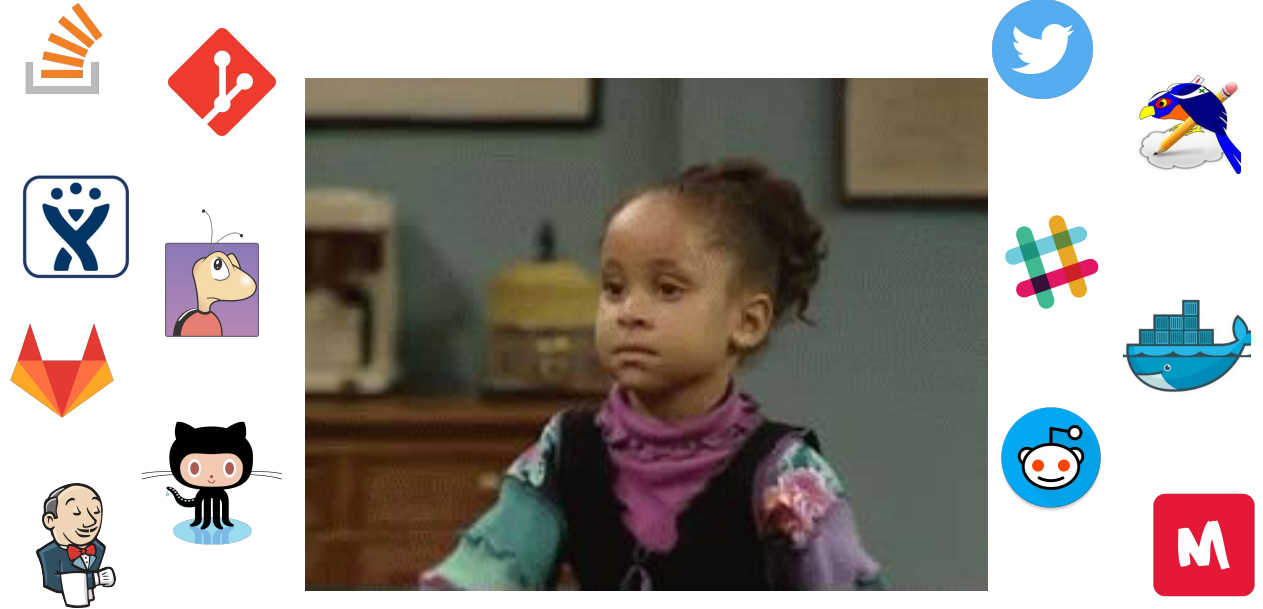
Are there new contributors?

/context



How many bugs were fixed past month?

/context



Has our gender diversity increased lately?

/context

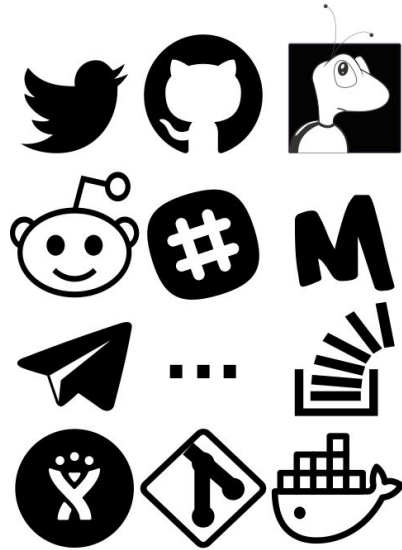


Perceval gathers data for you

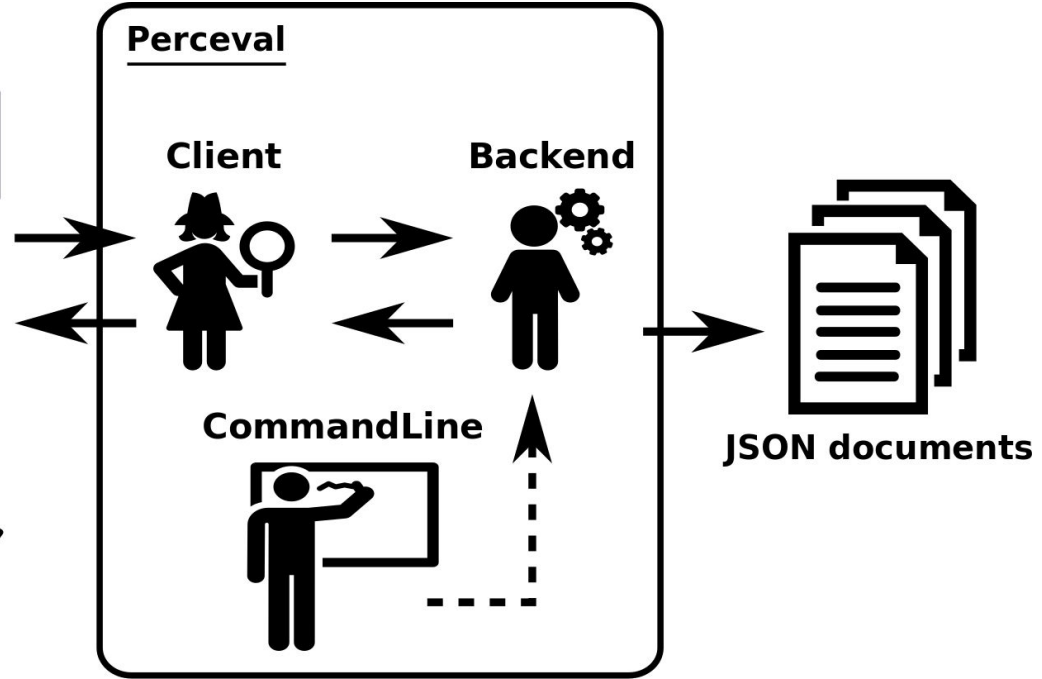


[chaoss/grimoirelab-perceval](https://chaoss.grimoirelab-perceval)

/perceval



Data sources





Backend gathering process for a specific data source, incremental and archiving mechanisms.

Client complexities to query the data source (e.g., pagination, OAuth access) handles connection problems.

CommandLine set up parameters to execute a backend. Optional arguments such as *help* and

debug.

As a program



```
$ pip3 install perceval  
$ perceval github chaoss grimoirelab-perceval  
--from-date=2017-03-01 --api-token=5...
```

```
[2018-08-01 12:31:10] - Sir Perceval is on his quest.
```

```
[2018-08-01 12:31:11] - Getting info for  
https://api.github.com/users/jgbarah
```

```
[2018-08-01 12:31:12] - Getting info for  
https://api.github.com/users/sduenas
```

```
... producing JSON documents ...
```

```
[2018-08-01 12:34:22] - Sir Perceval completed his quest
```



As a Python3 library

```
from perceval.backends.core.github import GitHub
from_date = datetime.datetime(2017, 3, 1)
github = GitHub("chaoss", "grimoirelab-perceval",
                api_token="5e3d7...")
for issue in github.fetch(from_date=from_date):
    print(issue['data'])
```

/perceval



```
{ "backend_name": "GitHub",  
  "backend_version": "0.17.2",  
  "category": "issue",  
  "data": {  
    "comments": 5,  
    "comments_data": [...],  
    "created_at": "2017-02-28T05:33:10Z",  
    .....,  
    "id": 210691361,  
    "state": "closed",  
    "title": "..",  
    "updated_at": "2017-03-02T09:51:49Z"  
  },  
  "origin": "https://github.com/chaoss/grimoirelab-perceval",  
  "perceval_version": "0.11.7",  
  "tag": "https://github.com/chaoss/grimoirelab-perceval",  
  "timestamp": 1517479766.878609,  
  "updated_on": 1488448309.0,  
  "uuid": "77a42463d5d0a34b1d58006263b85909a9788b52" }
```



Data source
data



Perceval
data



/graal



No code-based analysis with Perceval:

- Code complexity
- License evolution
- ...

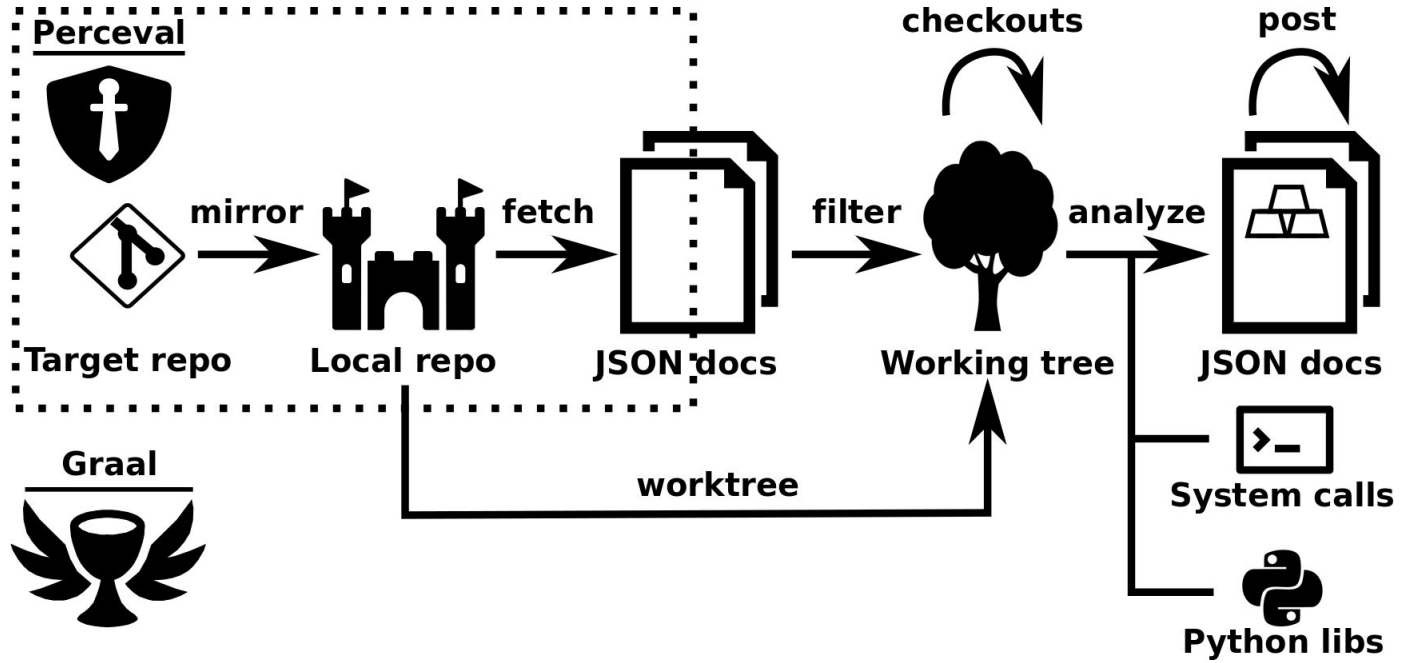
/graal



Graal does it!

It leverages on the incremental functionalities of Perceval and enhances the logic to handle Git repositories to process source code

/graal





Filter of commits based on Git JSON documents. For selected commits, checkouts it on the working tree.

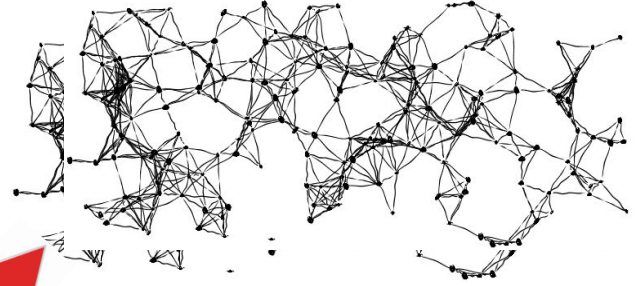
Analysis executes analysis tools in the working tree . Results of the analysis are automatically embedded in the JSON document.

Post-process alters the attributes of inflated JSON document, thus granting the user complete control over the output.

/graal



Code Complexity



Dependencies

Backends

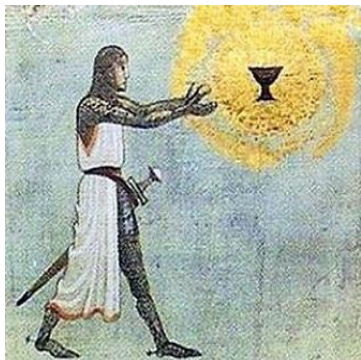


Quality checks



Security

As a program



```
$ pip install graal
$ sudo apt-install cloc
$ graal cocom
https://github.com/chaoss/grimoirelab-perceval
--git-path /tmp/graal-cocom
```

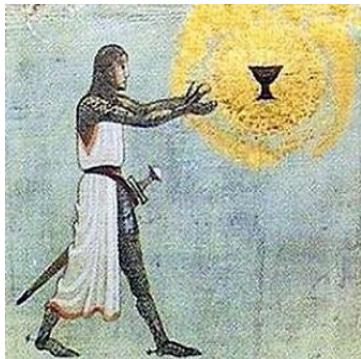
```
[2018-05-30 18:22:35,643] - Starting the quest for the
Gaal.
```

```
[2018-05-30 18:22:39,958] - Git worktree /tmp/...
created!
```

```
[2018-05-30 18:22:39,959] - Fetching commits
```

```
... producing JSON documents ...
```

```
[2018-05-31 04:51:56,112] - Quest completed.
```



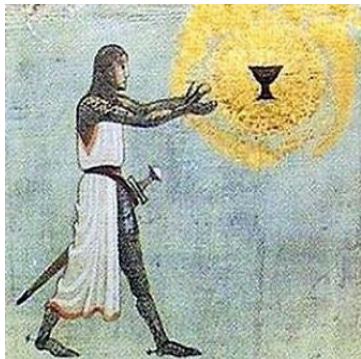
As a Python3 library

```
from graal.backends.core.cocom import CoCom

repo_uri =
"https://github.com/chaoss/grimoirelab-perceval"
repo_dir = "/tmp/graal-cocom"

cc = CoCom(uri=repo_uri, gitpath=repo_dir)

commits = [commit for commit in cc.fetch()]
```



```
{ "backend_name": "CoCom",  
  "backend_version": "0.2.1",  
  "category": "code_complexity",  
  "data": {  
    "AuthorDate": "Mon May 28 ...",  
    "CommitDate": "Mon May 28 ...",  
    "commit": "dc78c25...",  
    "message": "Increase coverage ...",  
    "analysis": [{  
      "ccn": 80, "num_funs": 33, ...,  
      "comments": "153", "loc": 341,  
      "file_path": "perceval/backend.py",  
    }], ...  
  },  
  "origin": "https://github.com/chaoss/grimoirelab-perceval",  
  "graal_version": "0.1.0",  
  "tag": "https://github.com/chaoss/grimoirelab-perceval",  
  "timestamp": 1534782753.878609,  
  "updated_on": 1392000436.0,  
  "uuid": "77a42463d5d0a34b1d58006263b85909a9788b52" }
```



Git metadata

Code data

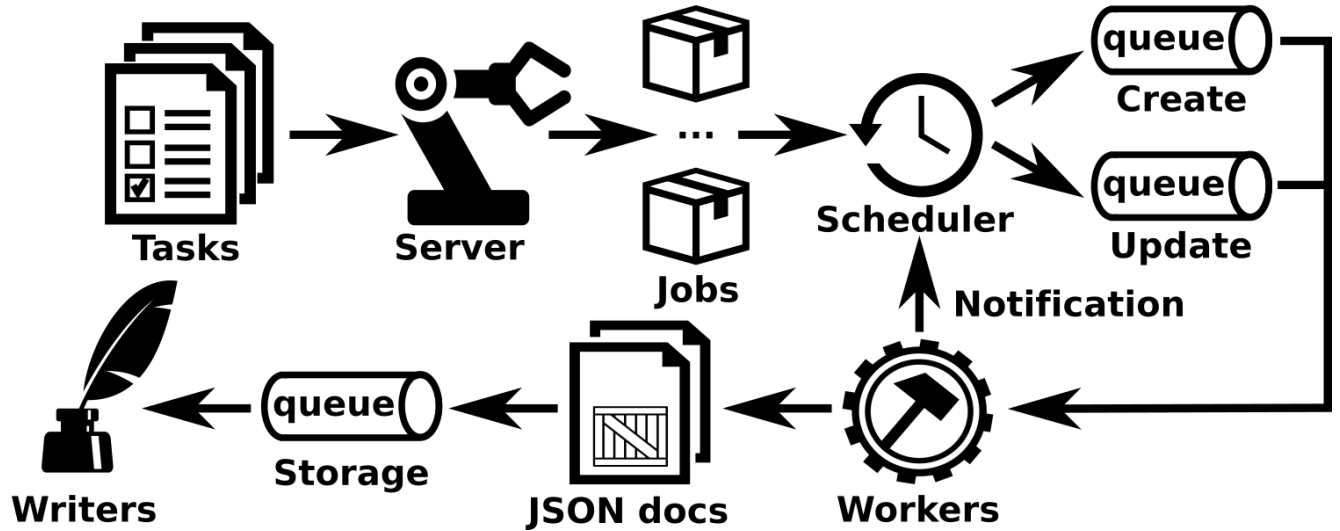


Graal data

/arthur

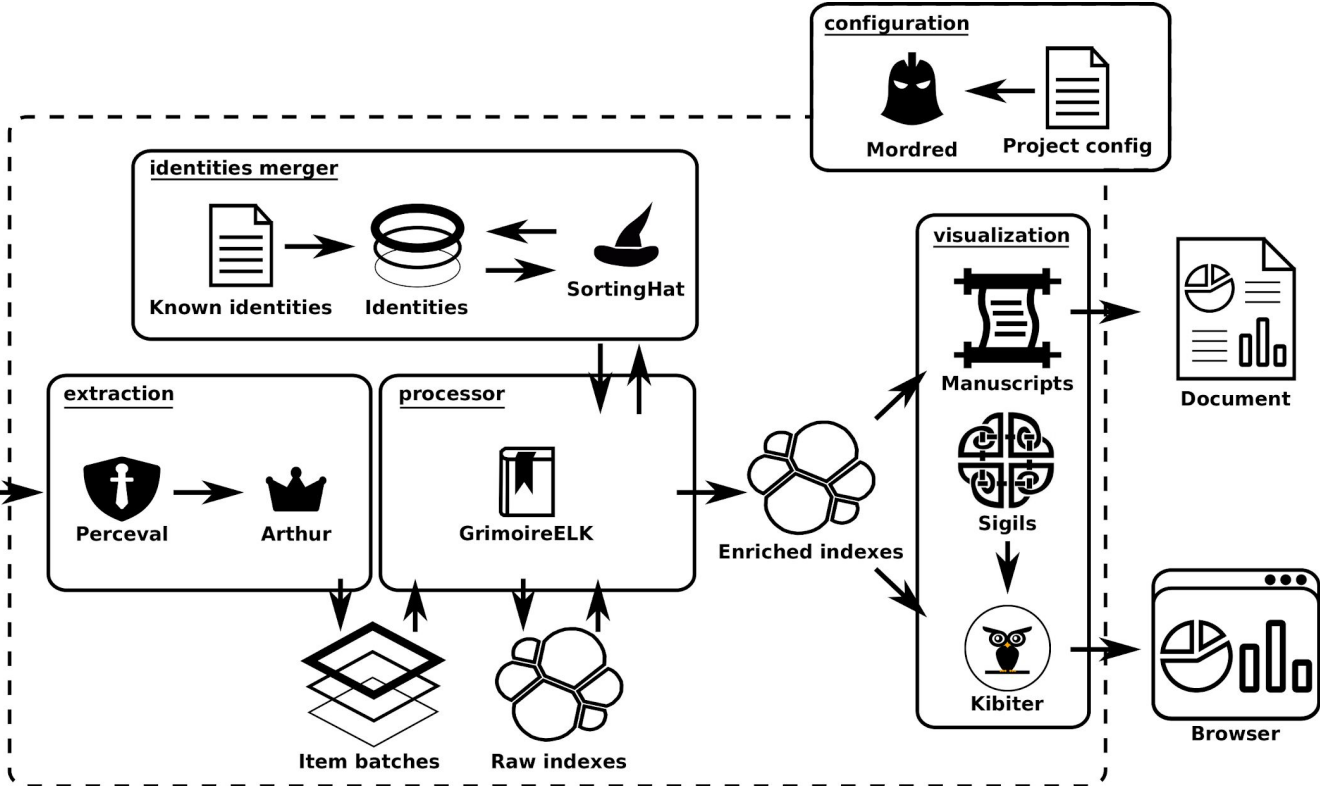
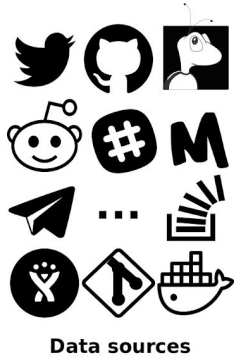


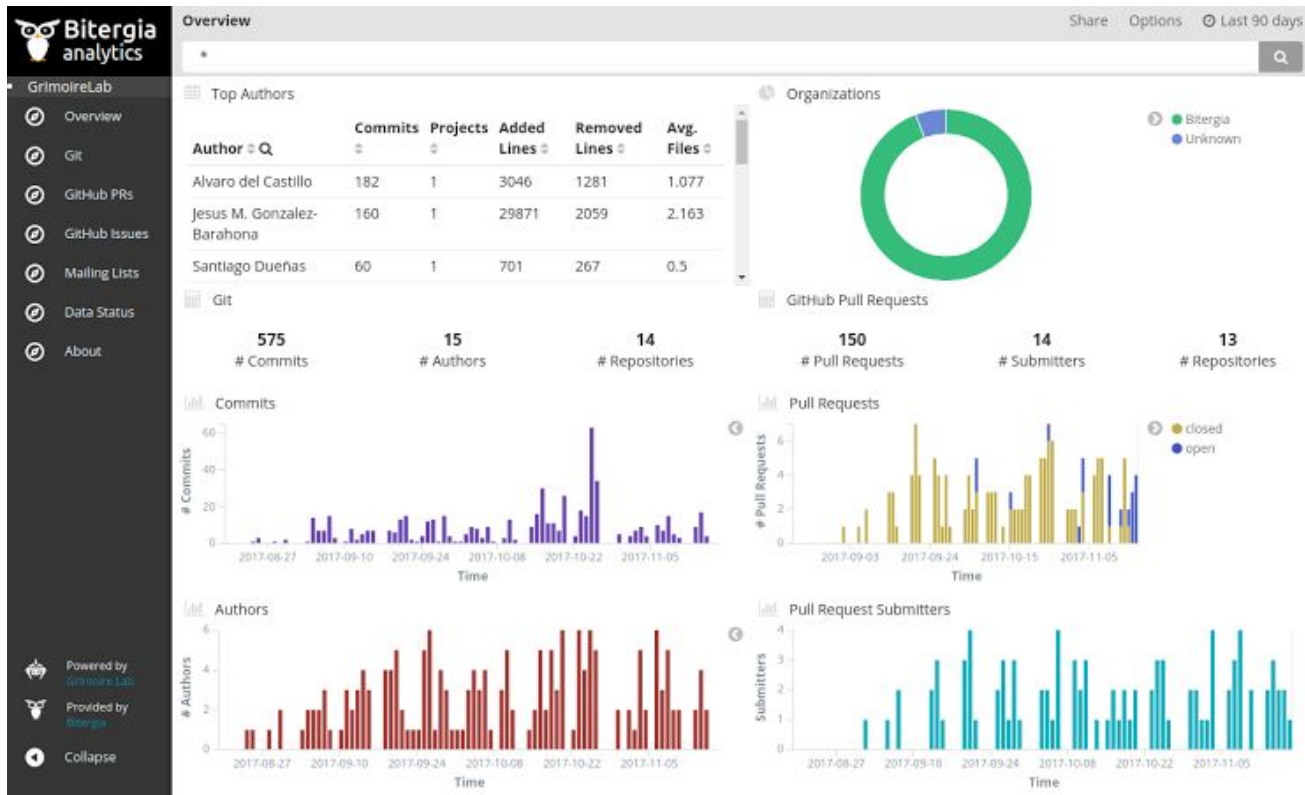
Arthur allows to schedule and run Perceval (and Graal) executions at scale through distributed *Redis* queues.



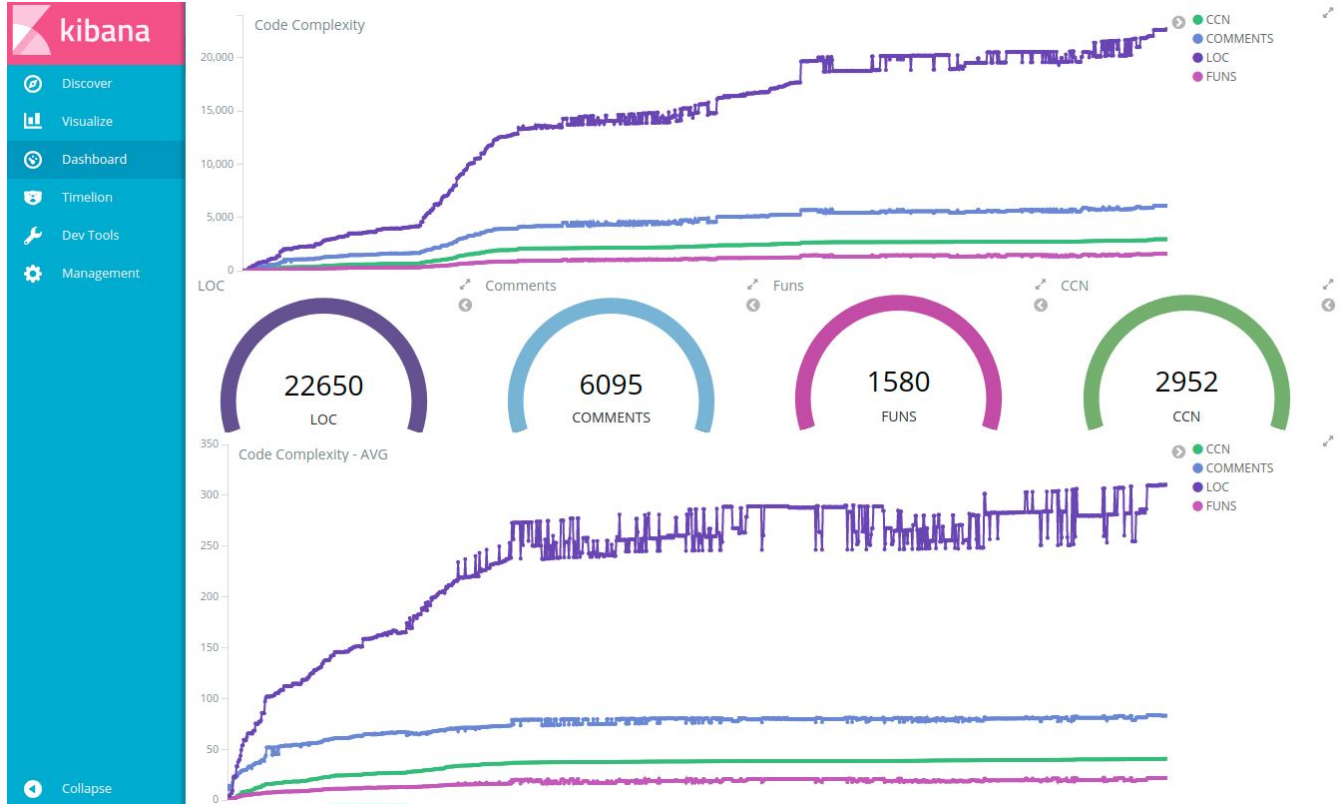
[chaoss/grimoirelab-kingarthur](https://chaoss.grimoirelab-kingarthur)







/explotation



/resources



[chaoss/grimoirelab-perceval](#)
[chaoss/grimoirelab-kingarthur](#)
[valerioscos/graal](#)



[grimoirelab.github.io](#)
[Perceval](#)
[Graal](#)



@grimoirelab
@bitergia
@CHAOSProj



Thanks