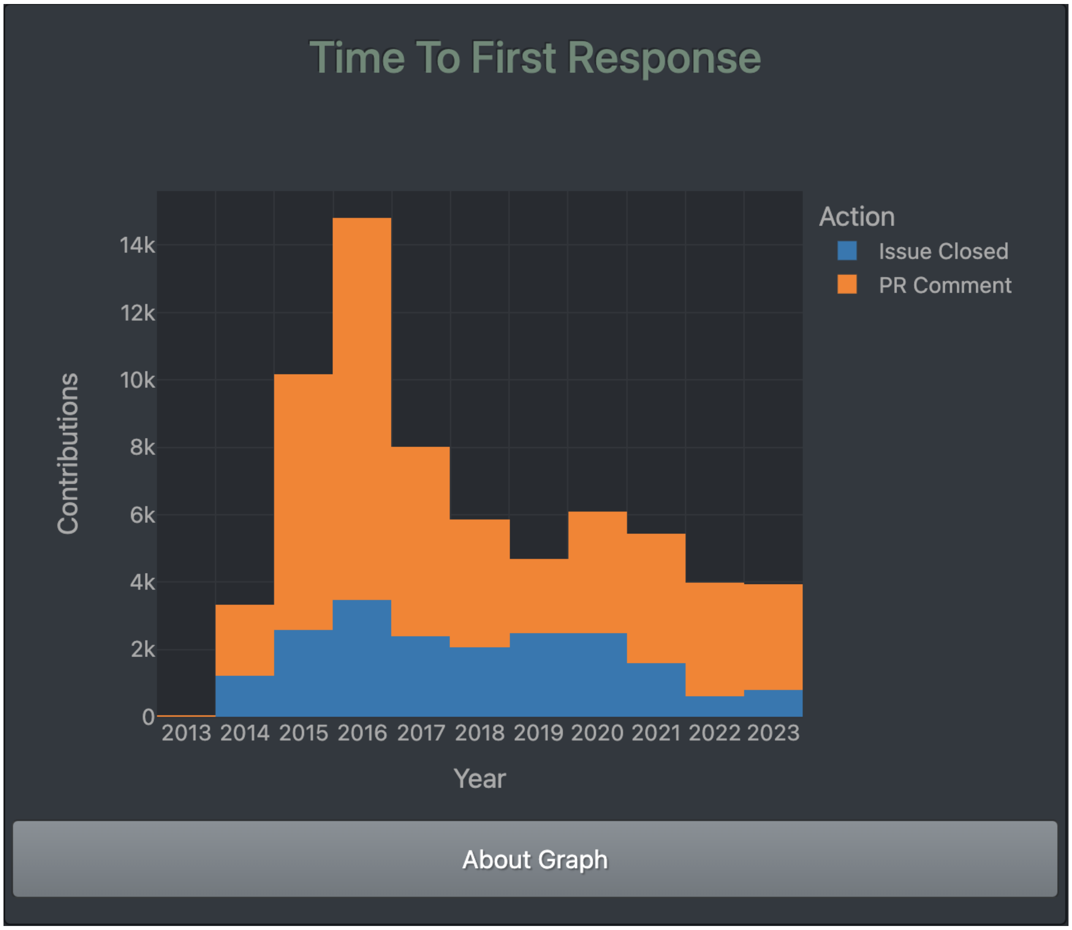
Metrics Model: Starter Project Health - Design

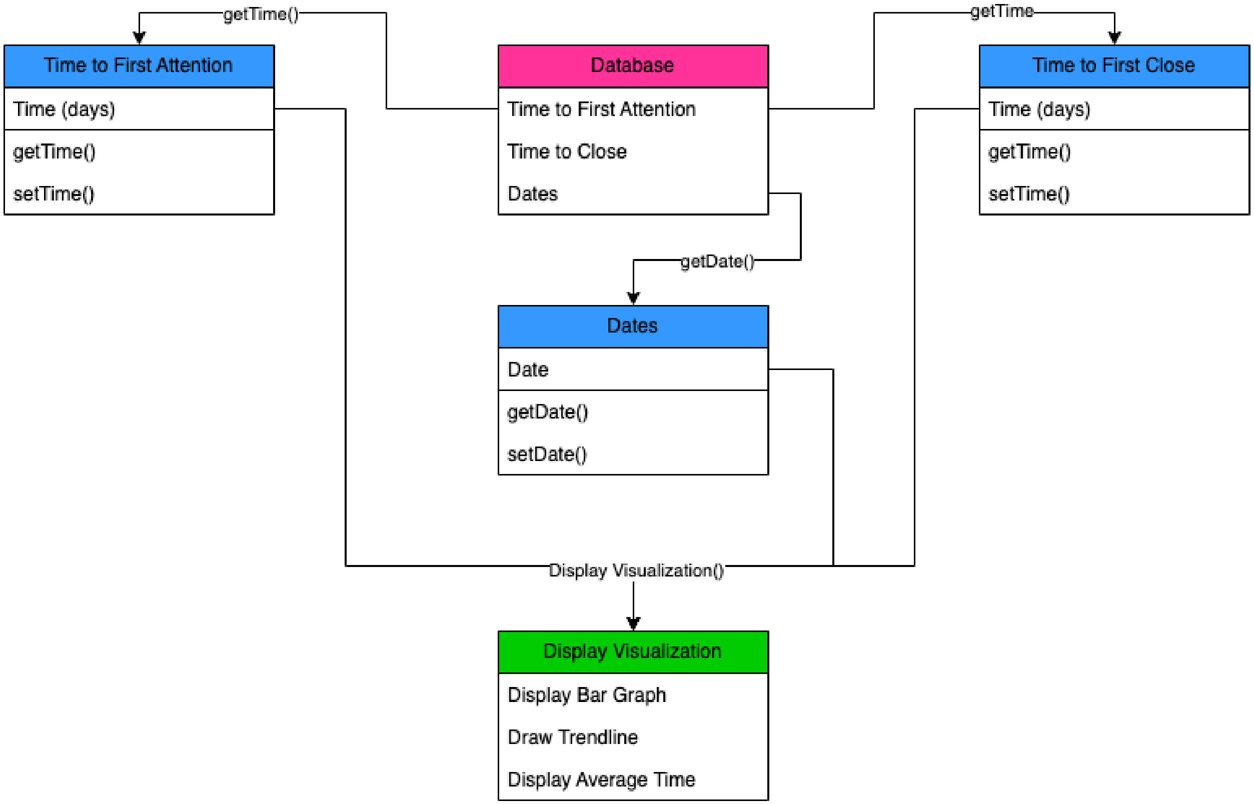
Time to First Response

https://chaoss.community/kb/metric-time-to-first-response/

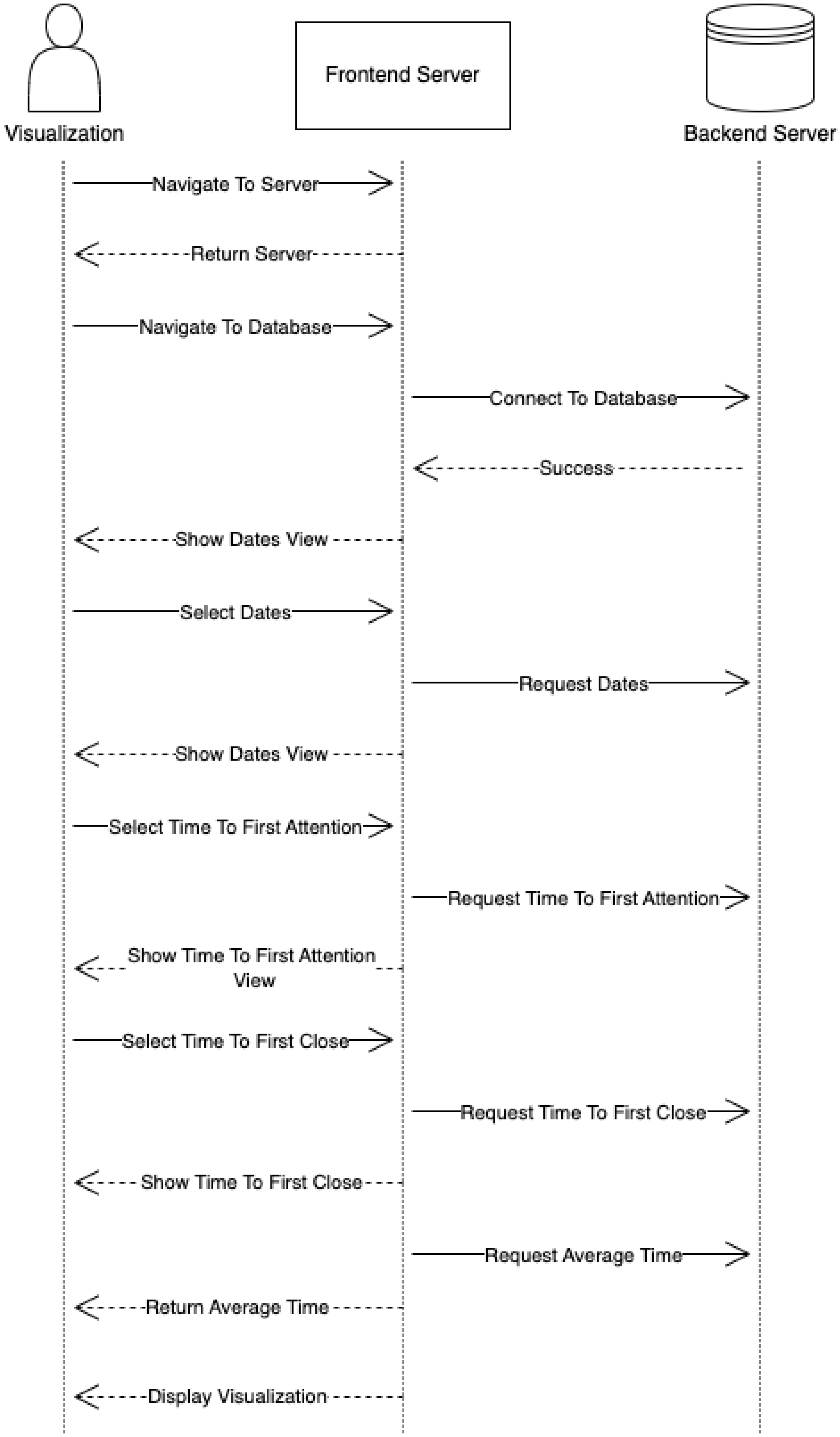
- Determine the amount of time between when an activity was opened (e.g. Issue or Change Request) and when it received the first response from a human.
  - The visualization will be a **bar graph**, x - axis will be the time goes by, the y axis will be the number contributions of first response(PR Comment) and issue close.
  - Description of the visualization
  - Display the average time to respond and close.



Class Diagram



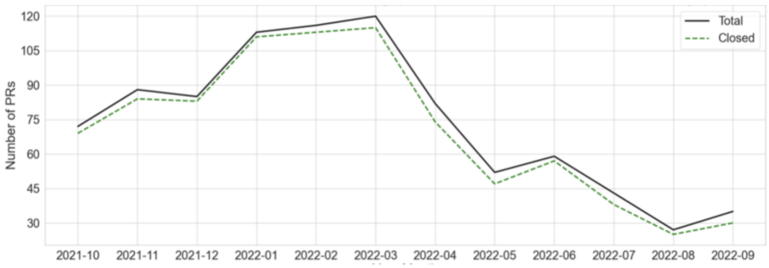
Sequence Diagram



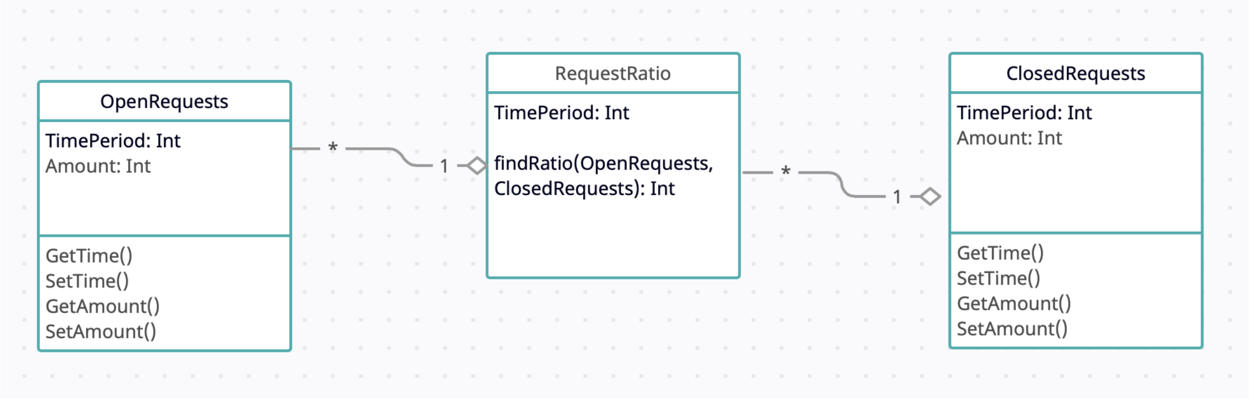
Change Request Closure Ratio

https://chaoss.community/kb/metric-change-request-closure-ratio/

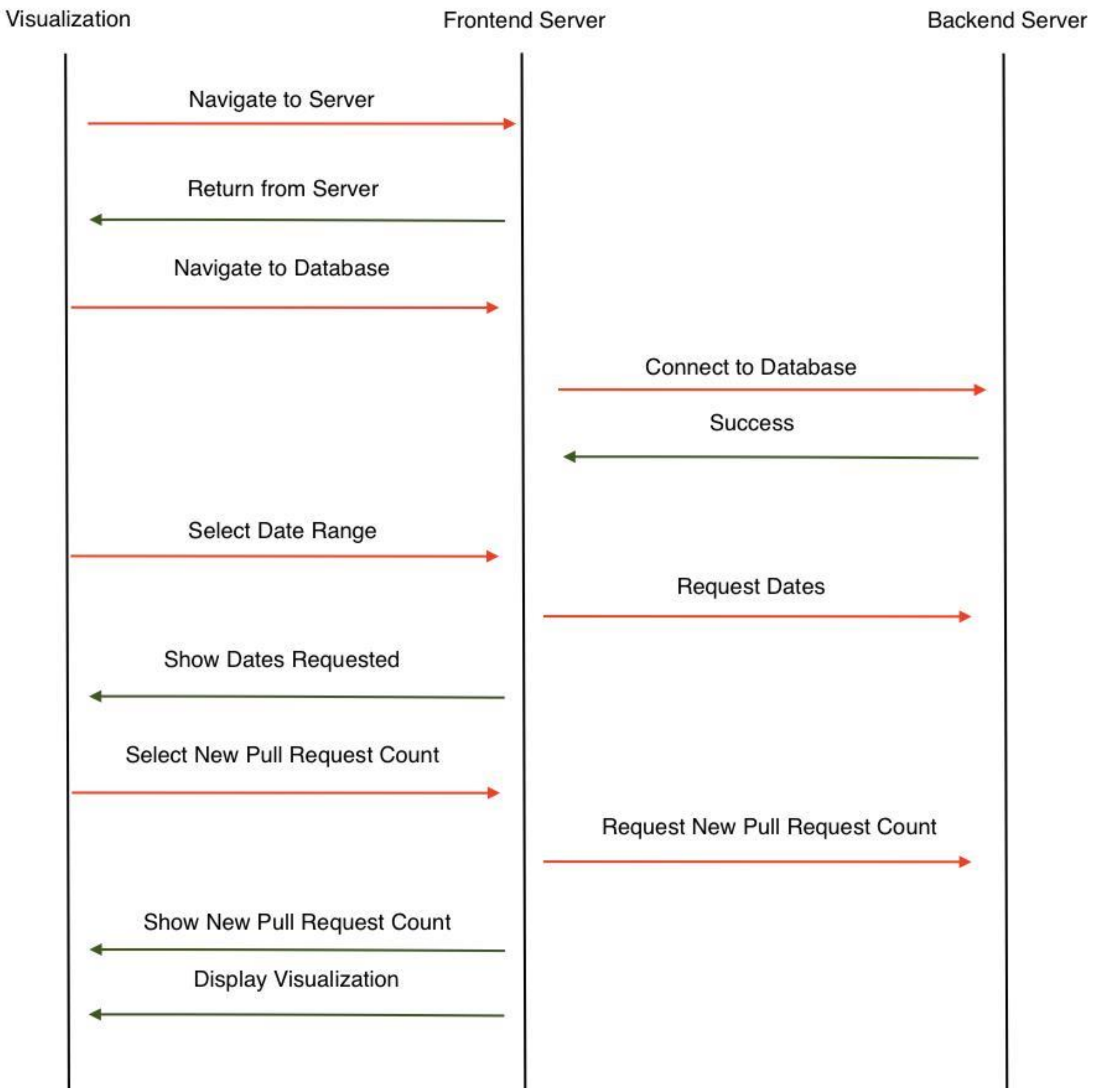
- Measure the ratio between the total number of open change requests during a time period versus the total number of change requests closed in that same period.
  - The visualization will be a **line graph**, x - axis will be the time goes by, the y axis will be changed (create, open, close, merged) over time.
  - Description of the visualization



Class Diagram



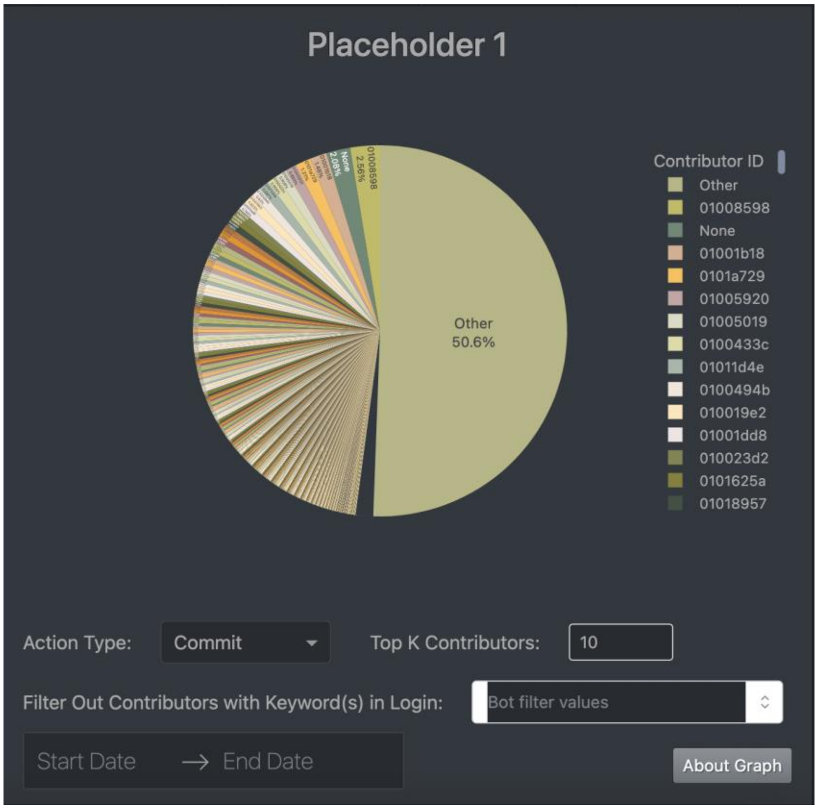
Sequence Diagram



Bus Factor

https://chaoss.community/kb/metric-bus-factor/

- Determine the smallest number of people that make 50% of contributions
    - The visualization will be a **pie chart**, including the numbers of people's contribution and how much are they in percentage.
    - Description of the visualization



Action Type: Commit

Top K Contributors: 10

Filter Out Contributors with Keyword(s) in Login: Bot filter values

Diagram 1

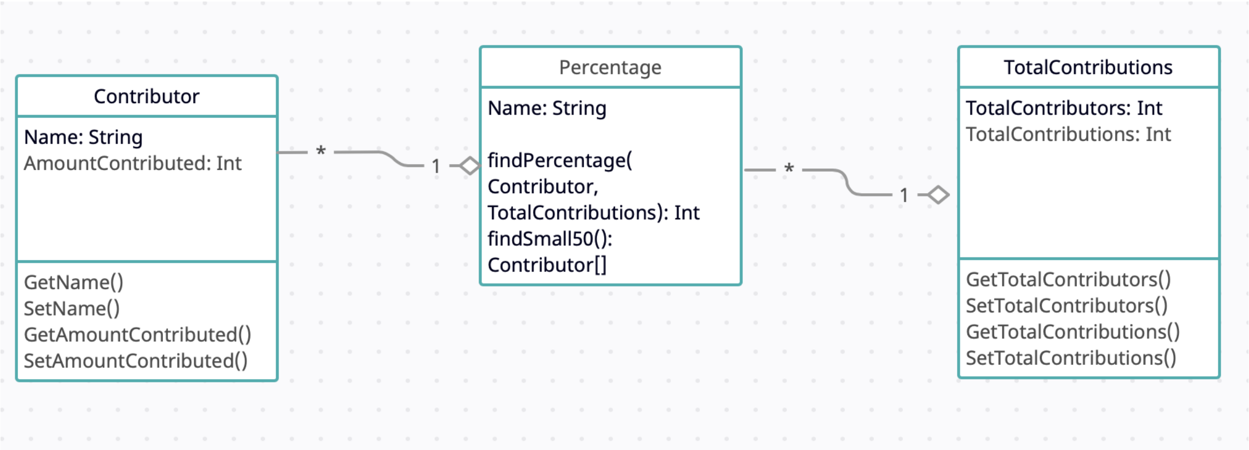
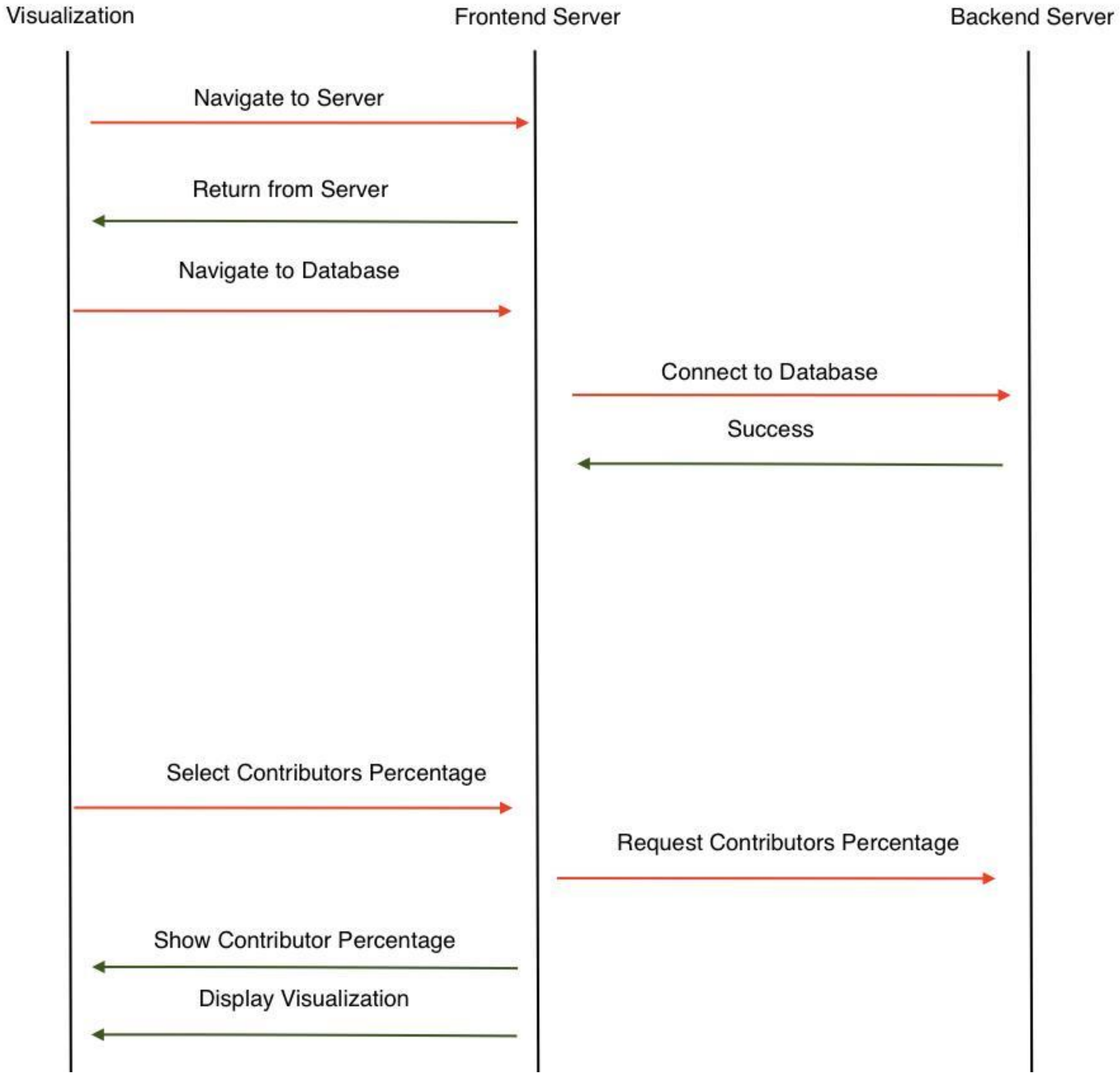


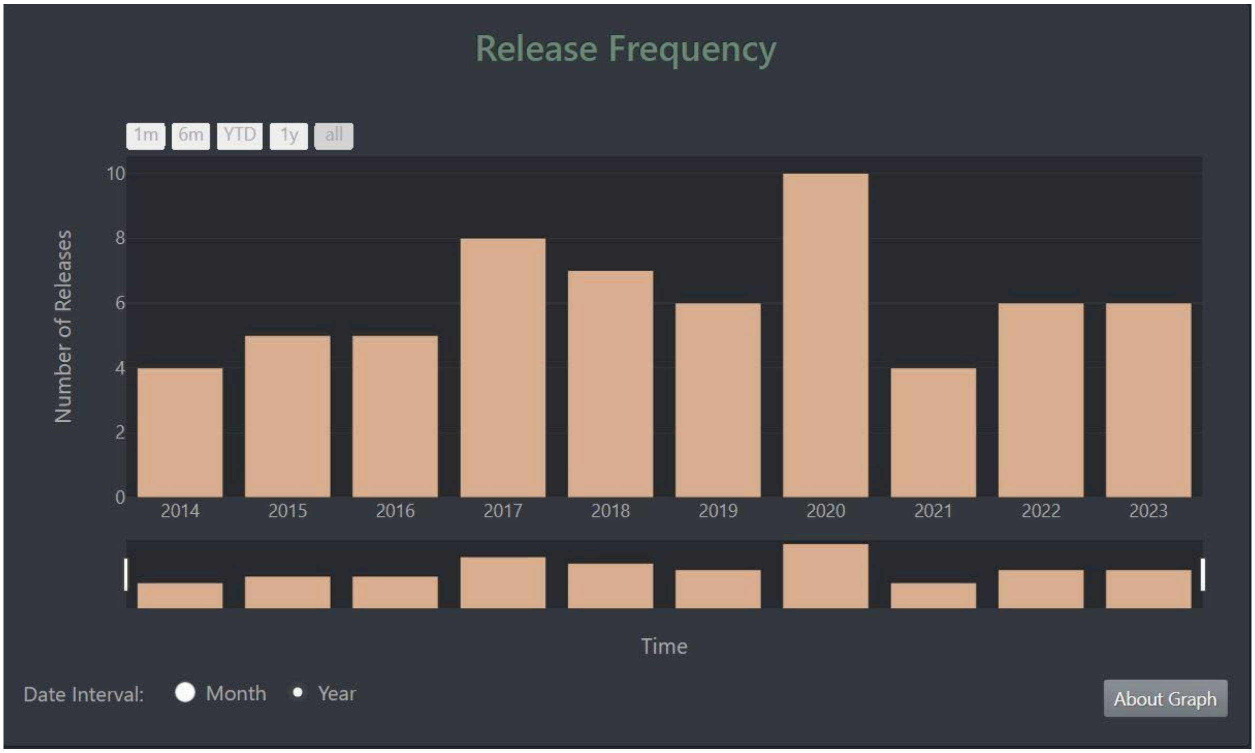
Diagram 2


Visualization
Frontend Server
Backend Server
Navigate to Server
Return from Server
Navigate to Database
Connect to Database
Success
Select Contributors Percentage
Request Contributors Percentage
Show Contributor Percentage
Display Visualization

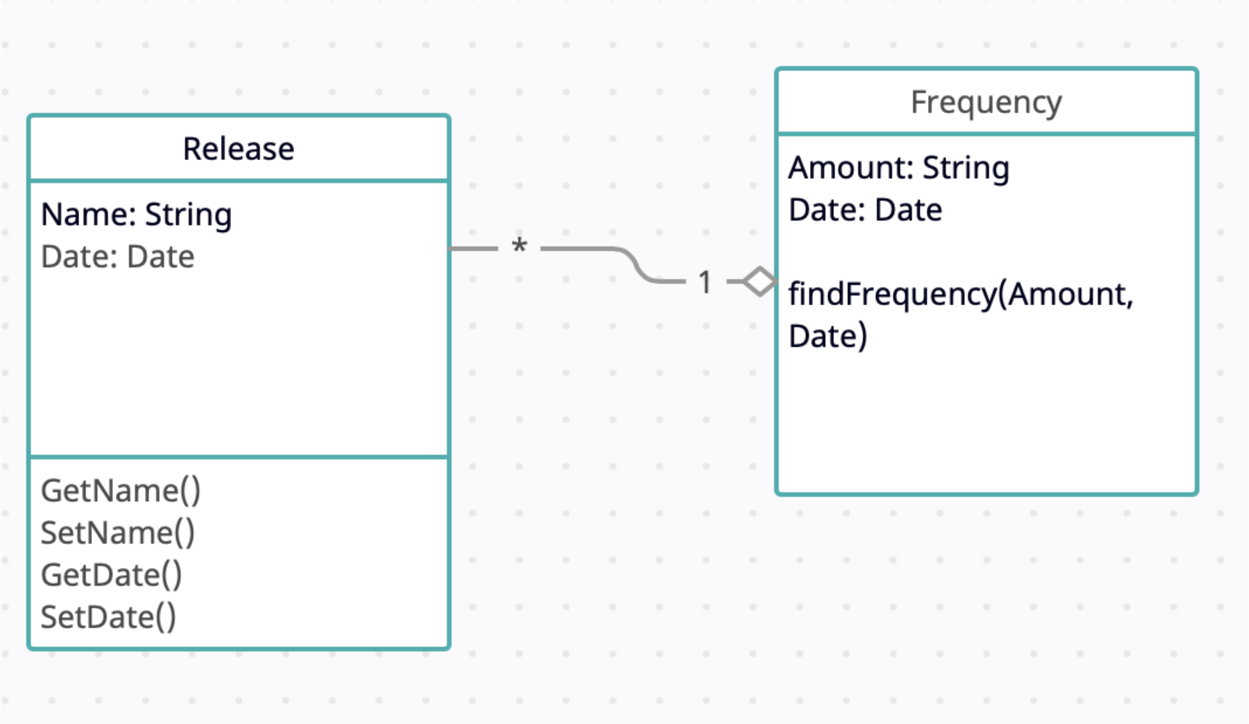
Release Frequency

https://chaoss.community/kb/metric-release-frequency/

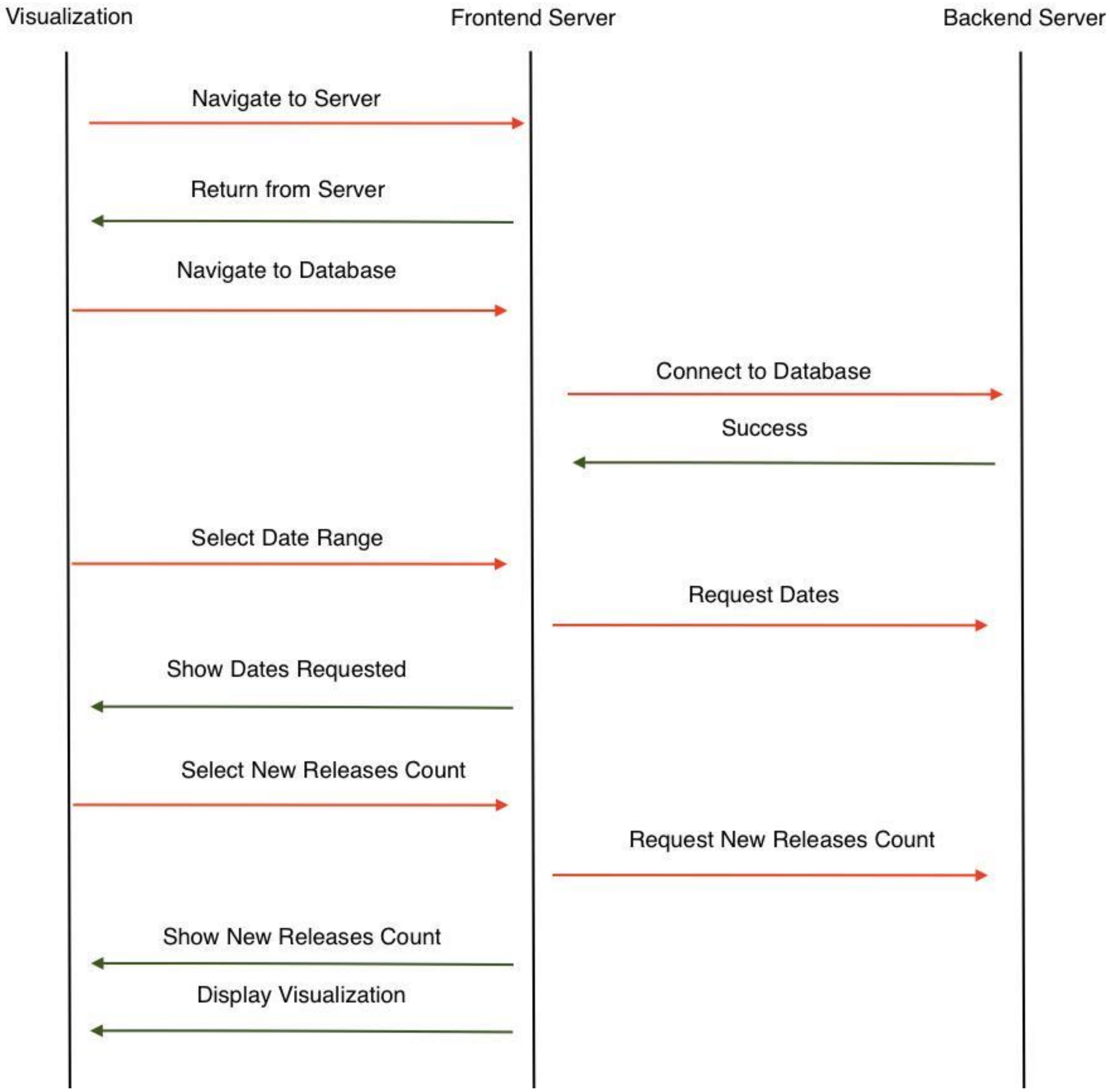
- Determine the frequency of project releases (including point releases with bug fixes)
    - The visualization will be a **Bar graph**, x - axis will be the time goes by, the y axis will be the numbers of project release, and the line graph will clearly show the frequency of the release over time.
    - Description of the visualization



Class Diagram



Sequence Diagram



Software Overview

In this model, we want to show the user how to help people get started with four key project health metrics that they can expand on and customize to meet their unique needs later.

System Requirements

- Software
  - python
  - 8Knot clone and install
  - docker Install
  - docker-compose install
  - git install
  - env.list at the top-level of the 8Knot directory that you clone
- Hardware
  - Not a specific requirement but must have enough memory space

Metrics Model: Community Activity - Design

Metrics (pick 5 from 6):

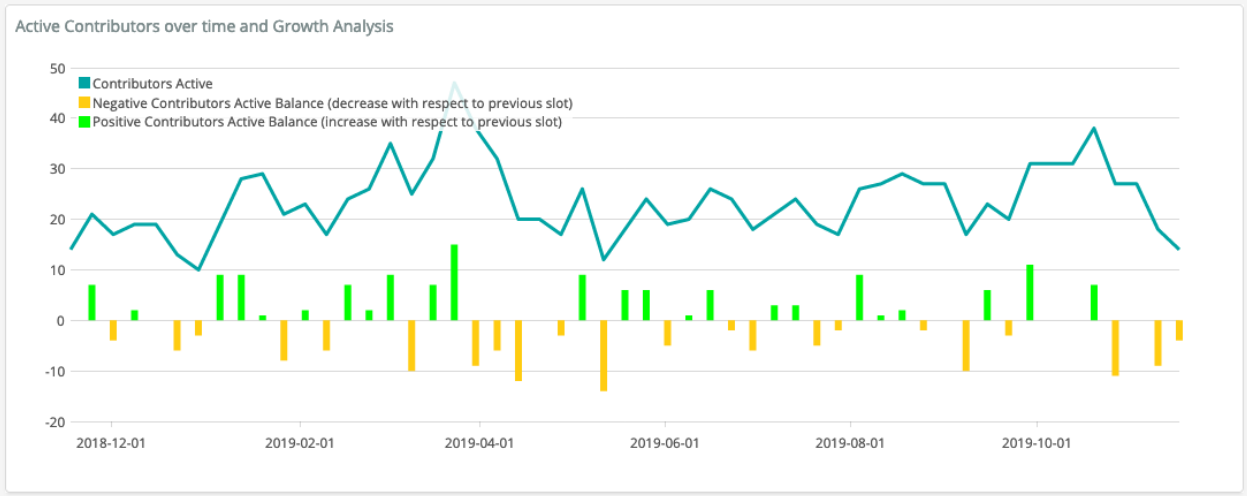
Contributors

- This visualization has many different visualizations, it contains a lot of information, but mostly uses pie charts or line graphs to show who are the contributors. The user can change the action type and top numbers of contributions. And the user can range the data on a date they choose.
  - contributor_count : Determine how many active commit authors, review participants, issue authors, and issue comments participants there are in the past 90 days.
  - maintainer_count : Determine the average number of maintainers per repository.

Basically, it's asking: Who are the contributors to a project? We just collect author names from collaboration tools a project uses. Or we can make a pie chart of top contributors by pie chart.

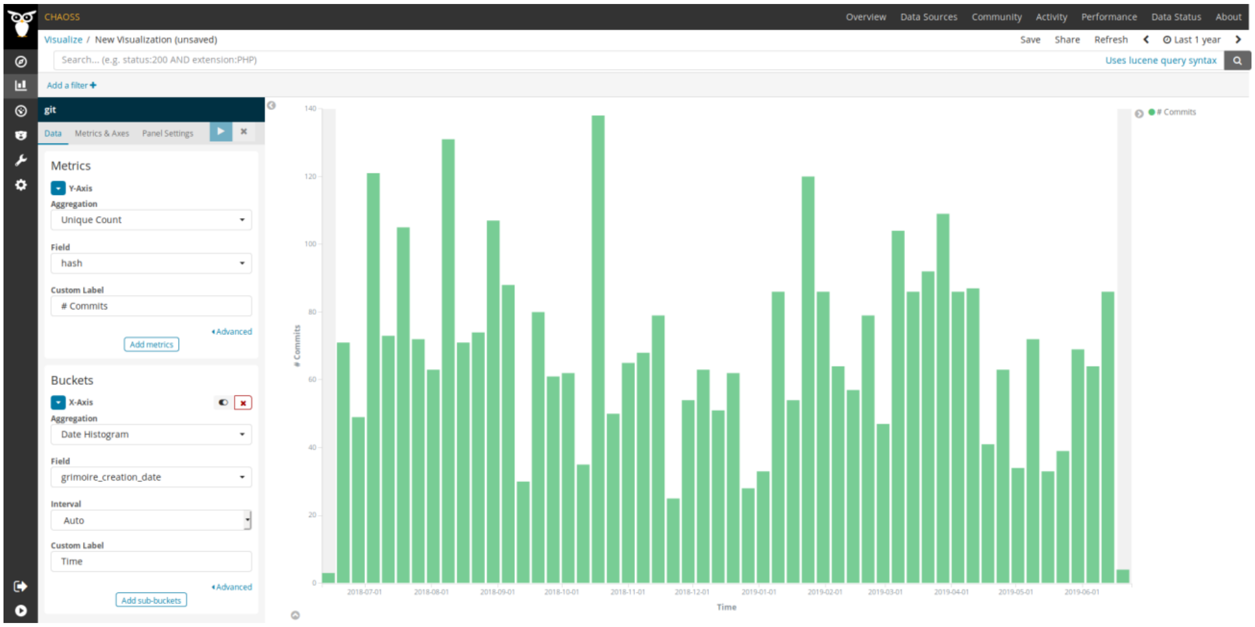
Visualizations:

- Change in the number of active contributors over time



Code Change Commits (N/A)

- This visualization will be a bar graph of the number of commits per week for the past year or more. Because the bar graph can easily show the difference between the number commits in an amount of time.
    - commit_frequency : Determine the number of commits in the past amount of time.


CHAOSS

Visualize / New Visualization (unsaved)

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

git

Metrics

Y-Axis

Aggregation: Unique Count

Field: hash

Custom Label: # Commits

Buckets

X-Axis

Aggregation: Date Histogram

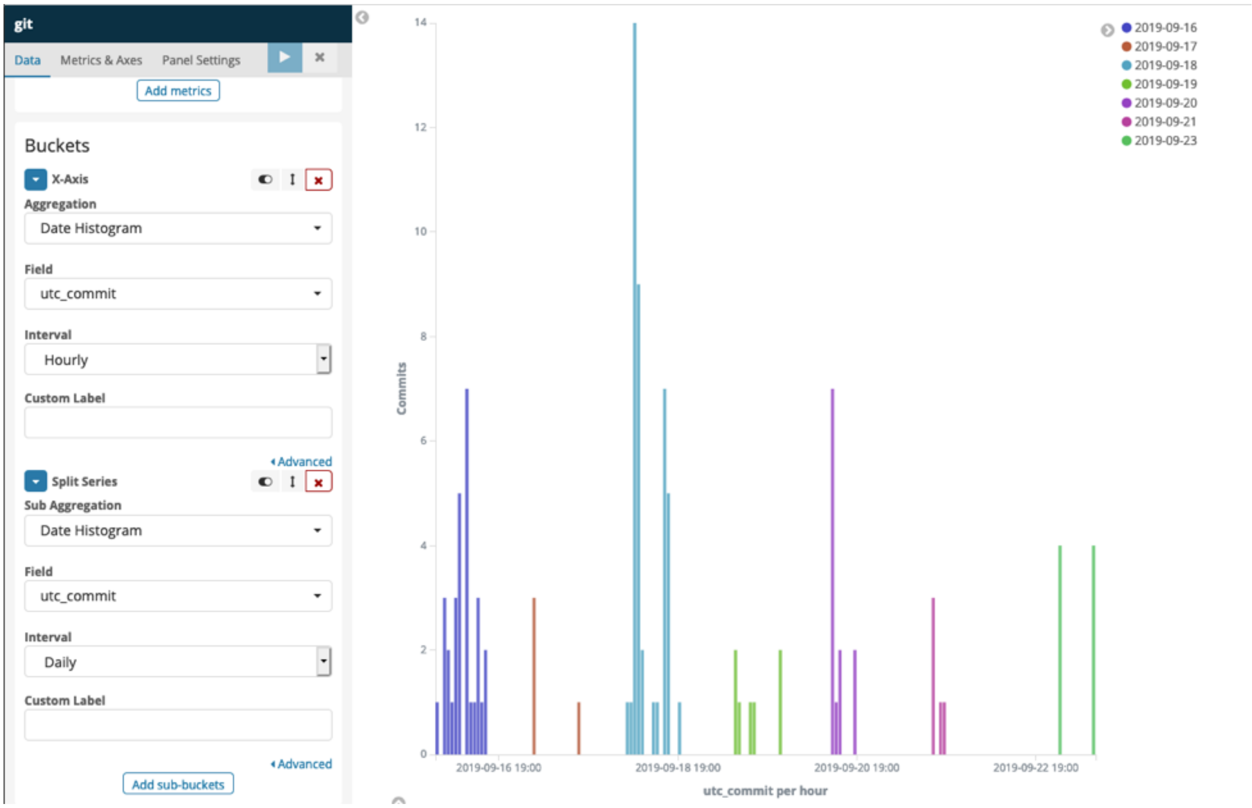
Field: grimoire_creation_date

Interval: Auto

Custom Label: Time

Activity Dates and Times

- This visualization will be a bar graph of the number of different types of activity over time. We are showing the dates and timestamps of when contributor activities occur to our viewer.
    - updated_since : Determine the average time per repository since the repository was last updated (in months).
    - created_since : Determine the average time per repository since a repository was created (in months).
    - comment_frequency : Determine the average number of comments per issue in the last 90 days.
    - updated_issues_count : Determine the number of issues updated in the last 90 days.
    - recent_releases_count : Determine the number of releases in the last year
    - meeting_count : Determine the number of meetings held in the last 90 days
    - meeting_attendee_count: Determine the average number of attendees per meeting



Buckets

X-Axis

Aggregation: Date Histogram

Field: utc_commit

Interval: Hourly

Custom Label:

Split Series

Sub Aggregation: Date Histogram

Field: utc_commit

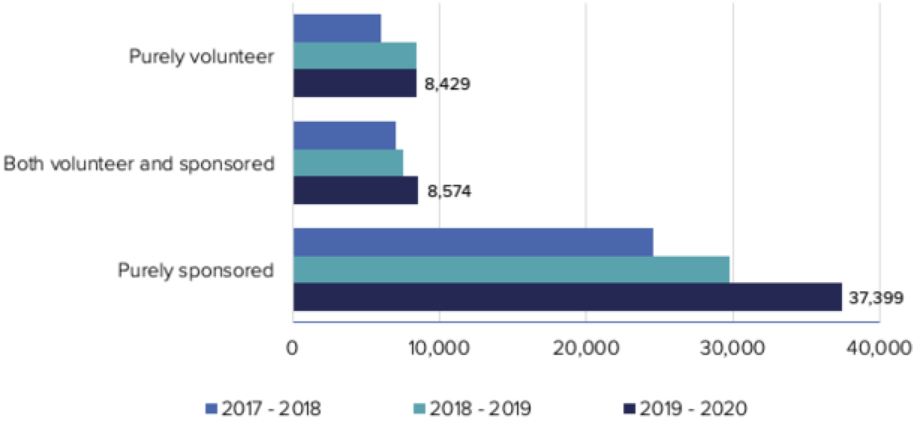
Interval: Daily

Custom Label:

X-axis will be the dates and timestamps, Y-axis will be the number of commits by contributors.

Contribution Attribution

- **Contribution Attribution** - This visualization will be a bar graph of contribution types over time and the details of contributors. We will show to our viewer who has contributed to the open source object, and information about people and organizations contributors belong to.
    - org_count : Determine the number of distinct organizations that contributors belong to

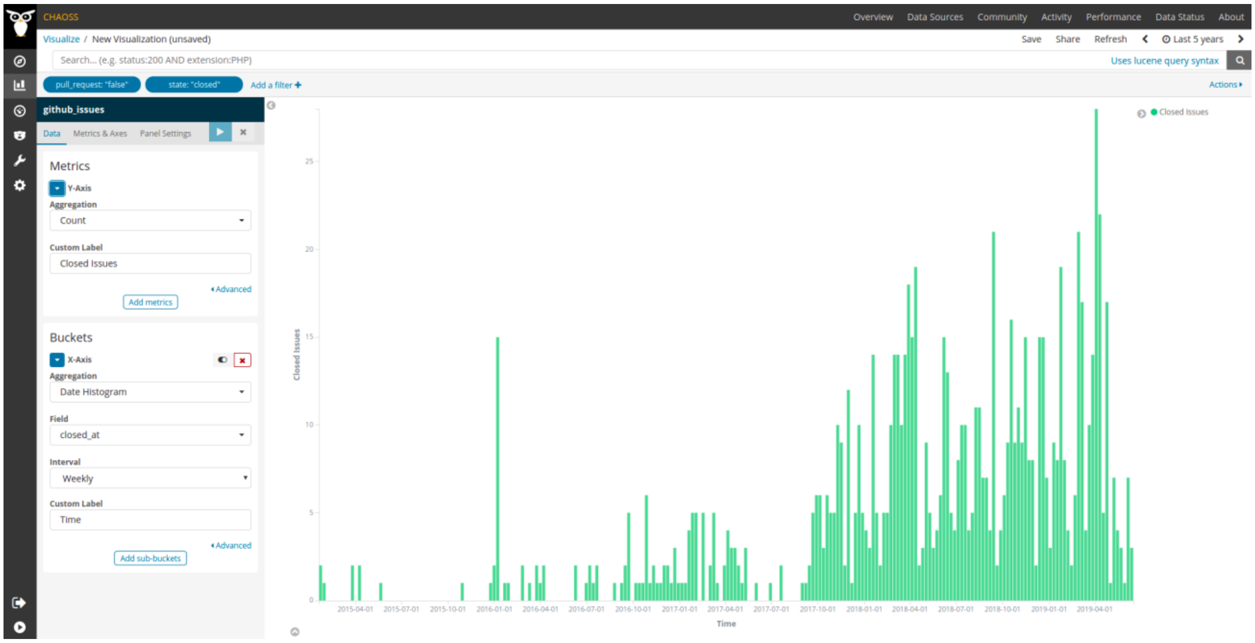


Change (Pull) Request Reviews (N/A)

- This visualization will be a bar graph of change requests over time. We just use a bar graph to show how many changes requests review (pull request review) were made in a certain period of time.
  - code_review_count : Determine the average number of review comments per pull request in the last 90 days

Issues Closed (Junshan)

- **Issues Closed** - This visualization will be a bar graph of issues closed over time. We just use a bar graph to show how many issues were closed in a certain period of time.
  - closed_issues_count : Determine the number of issues closed in the last 90 day



pull_request: "false"
state: "closed"

github_issues

Metrics

Y-Axis

Aggregation: Count

Custom Label: Closed Issues

Buckets

X-Axis

Aggregation: Date Histogram

Field: closed_at

Interval: Weekly

Custom Label: Time

Software Overview

In this model, we want to show the user how active the project is often used, the increase and decrease of the change in community activity, simple ways to quickly identify how the project is managed, and how to collaborate with different communities.

There are 6 different metrics, these visualizations will show the users different types of data of the software use.

System Requirements

- Software
  - python
  - 8Knot clone and install
  - docker Install
  - docker-compose install
  - git install
  - env.list at the top-level of the 8Knot directory that you clone
- Hardware
  - Not a specific requirement but must have enough memory space