Flipkart

GRiD 4.0

2022 Campus Challenge

Infosec Challenge

Open Source Software (OSS) Security Inspector

Team Name: Boolean Nincompoops
Institute Name: Sardar Patel Institute of Technology (SPIT)

Team members details

Team Name	Boolean Nincompoops		
Institute Name	Sardar Patel Institute of Technology (SPIT)		
Team Members	**1 (Leader)**	**2**	**3**
Name	Mirat Shah	Amal Thundiyil	Aaditya Mehar
Batch	2020-2024	2020-2024	2020-2024

Deliverables and Expectations

The solution should :

- Be able to analyse github, pypi, npm and other repos as well
- Should be able to perform scan of repo with link
- Criteria to say a repo/package is vulnerable
- Should provide rating of repo

Other Tips:

- Avoid giving suggestions or just plain solutions. Your idea should be backed by a solid approach and logic.
- You need to submit the code along with the idea proposal deck.
- Feel free to add links to any demo video as well

Glossary

- Hosts: Source code hosting platforms like GitHub, GitLab, Bitbucket, etc.
- Managers: Backend to clean and ingest data from package management tools/package indexes like NPM, PyPI, etc.
- Backend: Collection of hosts and managers.
- Processor: Performs data processing tasks and scores calculations based on security, popularity, community, and other metrics.
- Shift-Left Security: It is the practice of moving security checks as early and often in the SDLC as possible as part of a DevSecOps shift.
- Elasticsearch: It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

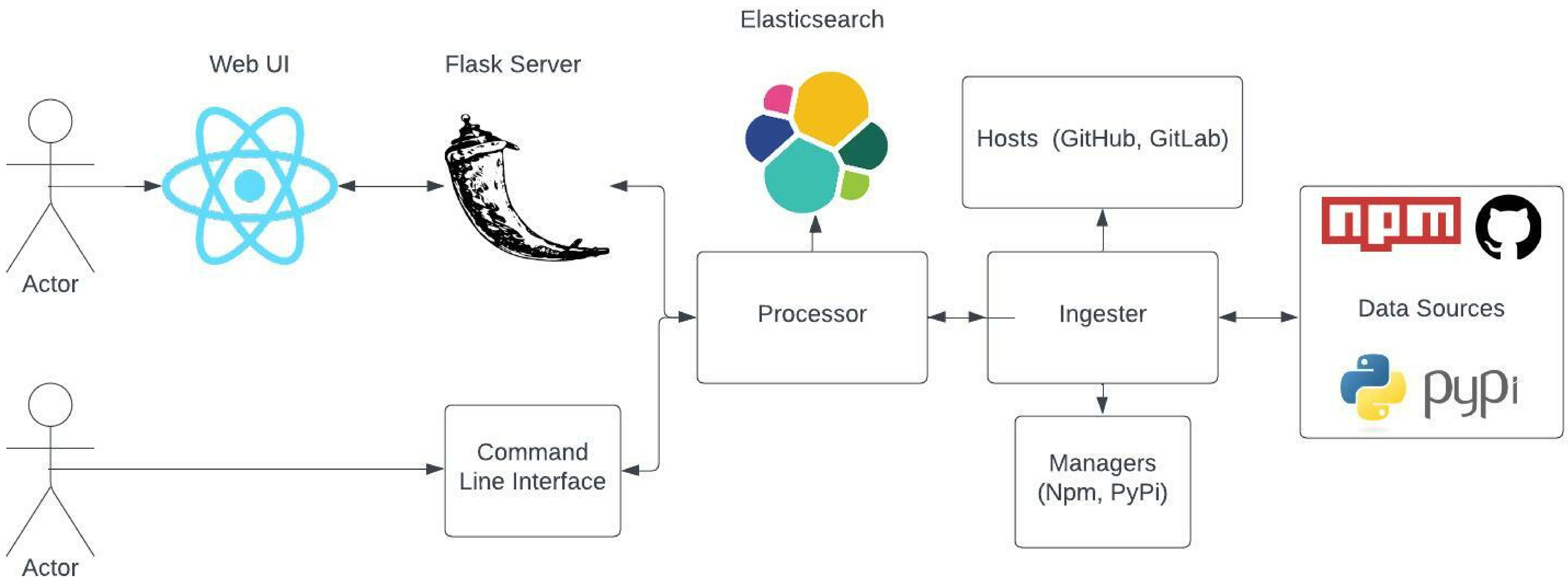
Use cases

- Sauron makes it easy for teams to find, prioritize, and fix security vulnerabilities in code hosting and package management websites like Github, NPM, PyPI, etc.
- Track open source security metrics in the libraries you consume.
- Understand how open source software impacts your business, and build a business case for managing open source software in a systematic way at the same time being proactive with open source packages you use.
- Get a comprehensive view of packages and vulnerabilities affecting them.

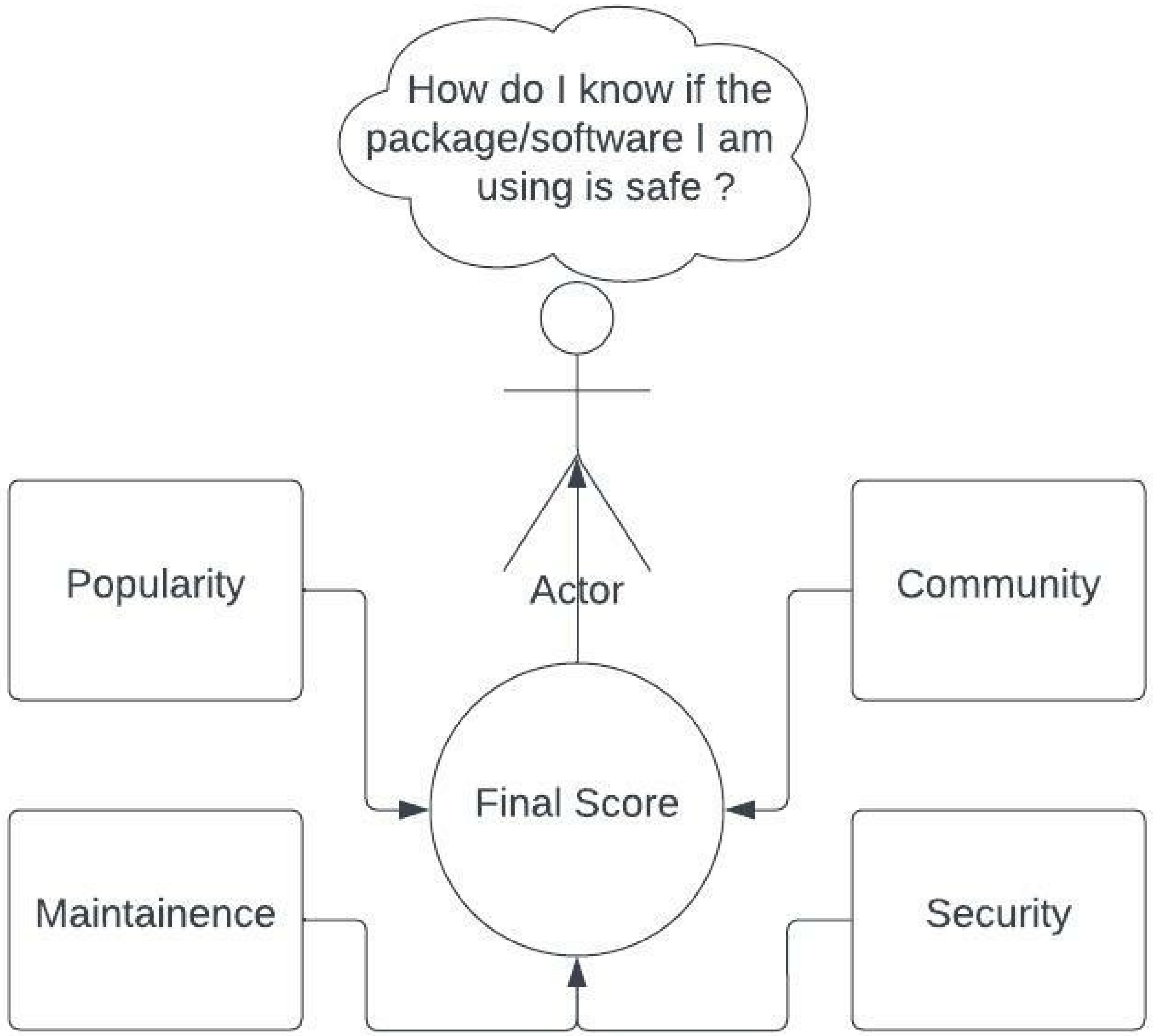
Solution Statement

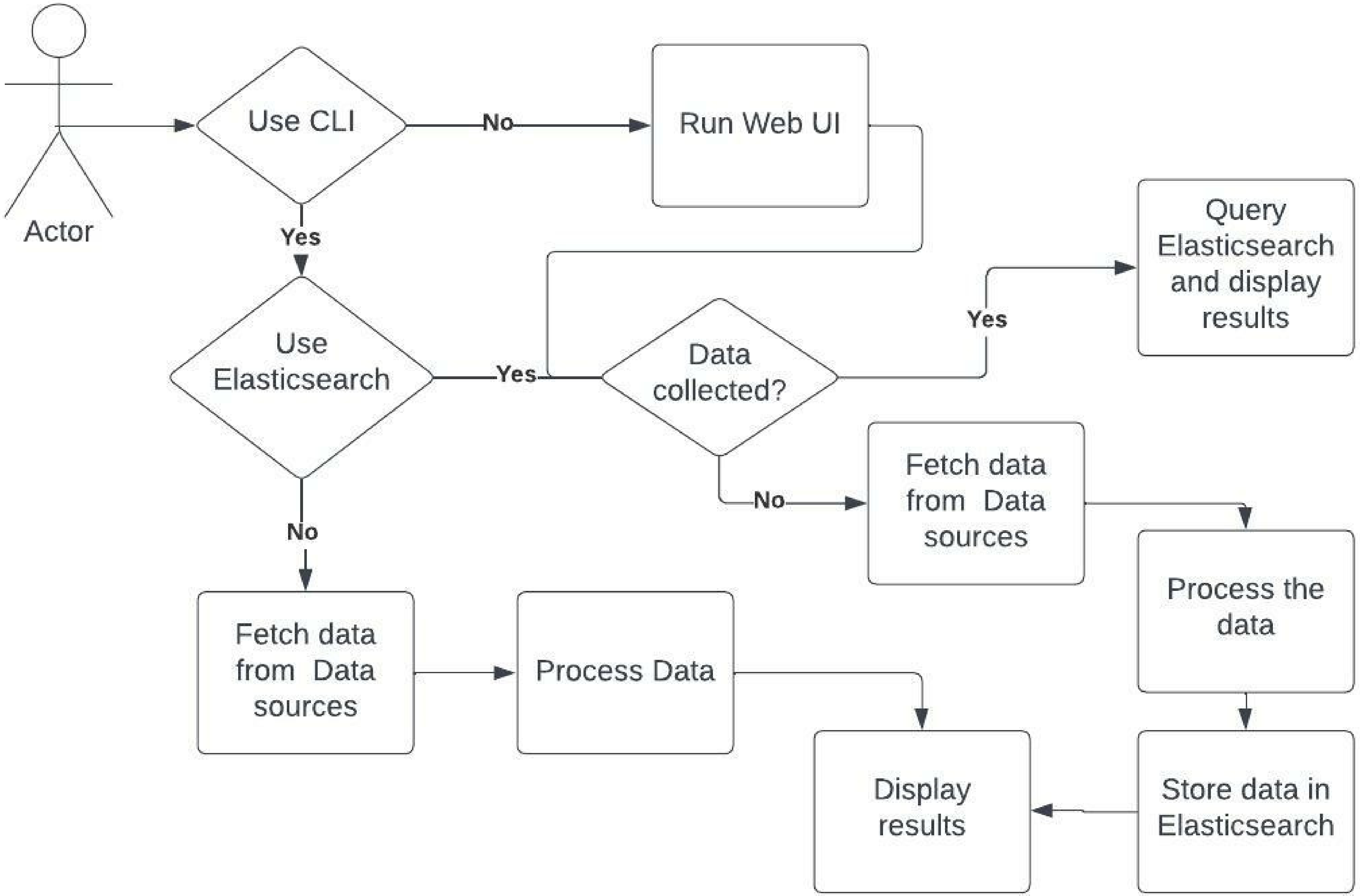
- User Story: How do I know if the package/repository I am using is safe?
- Solution: Ingest, clean, and processes the data available on the code hosting and package manager platforms to derive meaningful insights.
- Better security posture with cross-platform CLI tool and adoption of Shift-Left Security to incorporate security and testing into the development phase as early as possible.
- Assesses a number of important heuristics associated with software security and assigns each check a score.
- Sauron tracks four major metrics as shown in the diagram to produce a final score out of 10.

Architecture Diagram



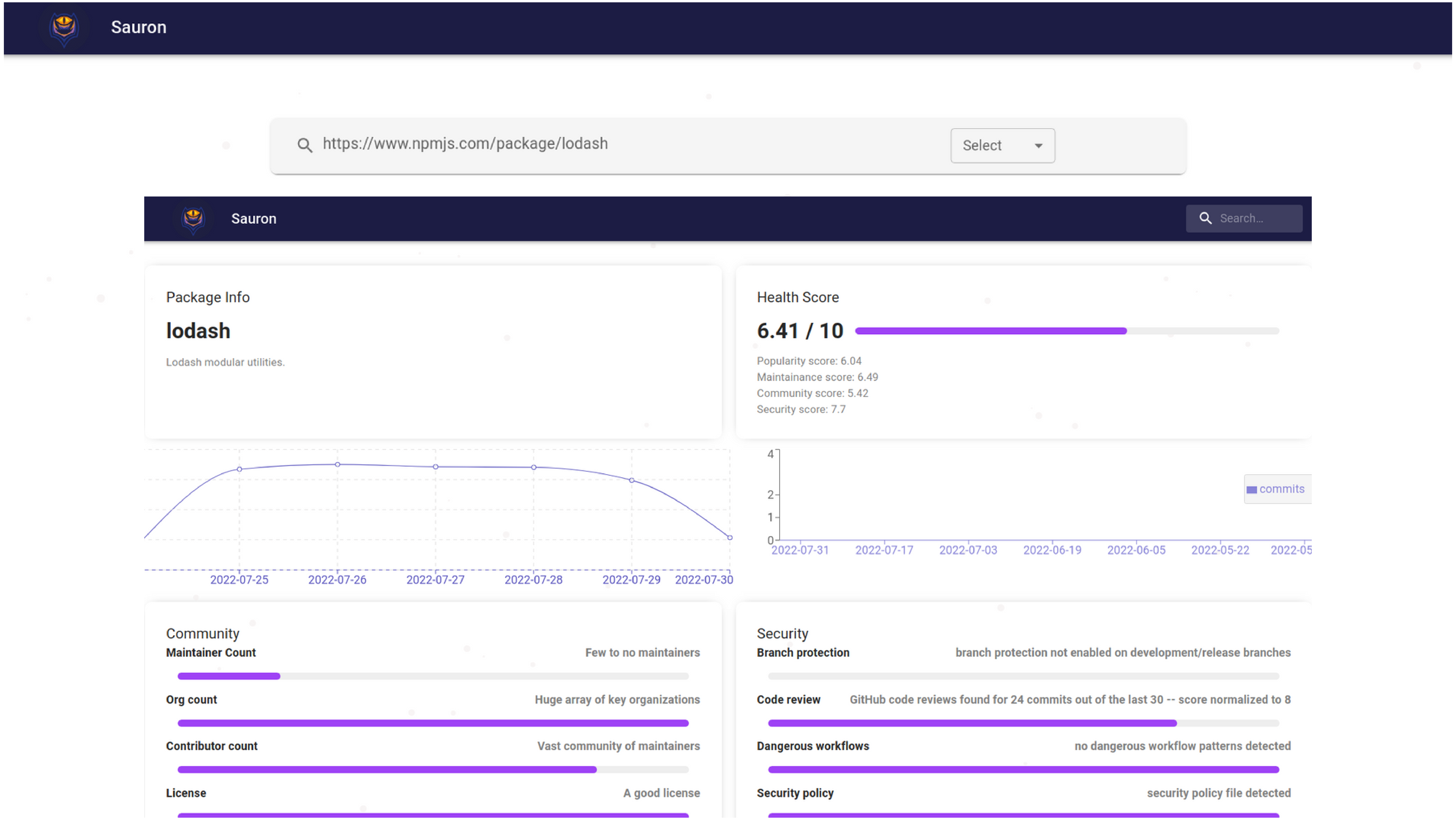
Workflow




Actor
Use CLI
No
Run Web UI
Yes
Use Elasticsearch
Yes
Data collected?
Yes
Query Elasticsearch and display results
No
Fetch data from Data sources
No
Fetch data from Data sources
Process Data
Process the data
Display results
Store data in Elasticsearch

Data processing is done by giving weights and thresholds to different parameters, tuned according to popular repositories and publicly defined metrics. More info in docs/metrics.md

Workflow – Web UI



Sauron

https://www.npmjs.com/package/lodash

Select

Sauron

Search...

Package Info

lodash

Lodash modular utilities.

Health Score

6.41 / 10

Popularity score: 6.04
Maintainance score: 6.49
Community score: 5.42
Security score: 7.7

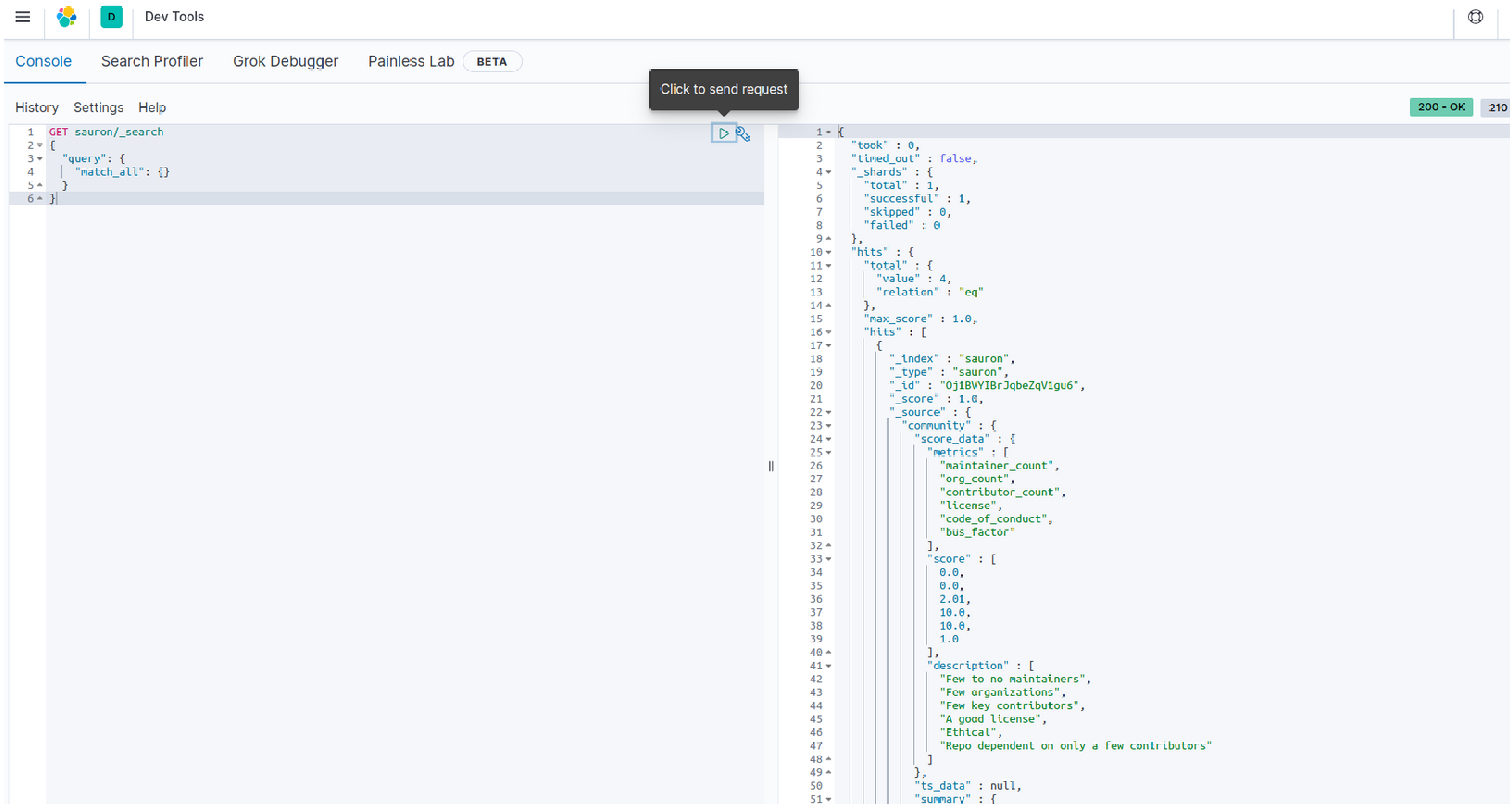
Community

Maintainer Count	Few to no maintainers
Org count	Huge array of key organizations
Contributor count	Vast community of maintainers
License	A good license

Security

Branch protection	branch protection not enabled on development/release branches
Code review	GitHub code reviews found for 24 commits out of the last 30 -- score normalized to 8
Dangerous workflows	no dangerous workflow patterns detected
Security policy	security policy file detected

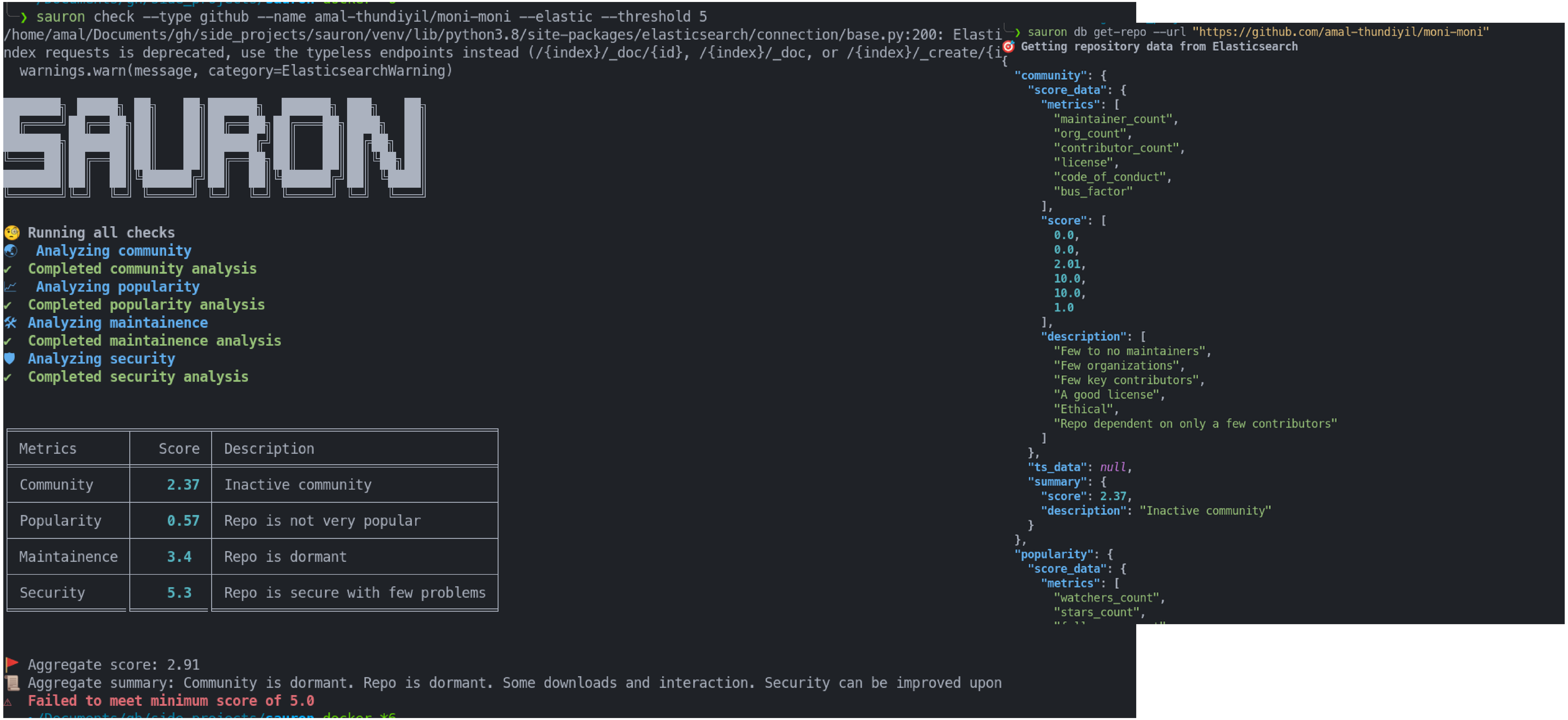
Workflow – Web UI



```
GET sauron/_search
{
  "query": {
    "match_all": {}
  }
}
```

```
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "sauron",
        "_type" : "sauron",
        "_id" : "Oj1BVYIBrJqbeZqV1gu6",
        "_score" : 1.0,
        "_source" : {
          "community" : {
            "score_data" : {
              "metrics" : [
                "maintainer_count",
                "org_count",
                "contributor_count",
                "license",
                "code_of_conduct",
                "bus_factor"
              ],
              "score" : [
                0.0,
                0.0,
                2.01,
                10.0,
                10.0,
                1.0
              ],
              "description" : [
                "Few to no maintainers",
                "Few organizations",
                "Few key contributors",
                "A good license",
                "Ethical",
                "Repo dependent on only a few contributors"
              ]
            },
            "ts_data" : null,
            "summary" : {
```

Workflow – CLI



```
❯ sauron check --type github --name amal-thundiyil/moni-moni --elastic --threshold 5
/home/amal/Documents/gh/side_projects/sauron/venv/lib/python3.8/site-packages/elasticsearch/connection/base.py:200: Elasti
ndex requests is deprecated, use the typeless endpoints instead (/{index}/_doc/{id}, /{index}/_doc, or /{index}/_create/{i
  warnings.warn(message, category=ElasticsearchWarning)

SAURON

🧐 Running all checks
🌏  Analyzing community
✓ Completed community analysis
📈  Analyzing popularity
✓ Completed popularity analysis
🛠 Analyzing maintainence
✓ Completed maintainence analysis
🛡 Analyzing security
✓ Completed security analysis

┌───────────────┬────────┬──────────────────────────────┐
│ Metrics       │  Score │ Description                  │
├───────────────┼────────┼──────────────────────────────┤
│ Community     │   2.37 │ Inactive community           │
├───────────────┼────────┼──────────────────────────────┤
│ Popularity    │   0.57 │ Repo is not very popular     │
├───────────────┼────────┼──────────────────────────────┤
│ Maintainence  │    3.4 │ Repo is dormant              │
├───────────────┼────────┼──────────────────────────────┤
│ Security      │    5.3 │ Repo is secure with few problems │
└───────────────┴────────┴──────────────────────────────┘

🚩 Aggregate score: 2.91
📜 Aggregate summary: Community is dormant. Repo is dormant. Some downloads and interaction. Security can be improved upon
⚠ Failed to meet minimum score of 5.0
```

```
❯ sauron db get-repo --url "https://github.com/amal-thundiyil/moni-moni"
🎯 Getting repository data from Elasticsearch
```

```
{
  "community": {
    "score_data": {
      "metrics": [
        "maintainer_count",
        "org_count",
        "contributor_count",
        "license",
        "code_of_conduct",
        "bus_factor"
      ],
      "score": [
        0.0,
        0.0,
        2.01,
        10.0,
        10.0,
        1.0
      ],
      "description": [
        "Few to no maintainers",
        "Few organizations",
        "Few key contributors",
        "A good license",
        "Ethical",
        "Repo dependent on only a few contributors"
      ]
    },
    "ts_data": null,
    "summary": {
      "score": 2.37,
      "description": "Inactive community"
    }
  },
  "popularity": {
    "score_data": {
      "metrics": [
        "watchers_count",
        "stars_count",
```

Limitations

- Takes time to process and present data on the first try.
- Rate limit and speed of vendor APIs to source the data are a concern.
- Limited to few popular platforms which contribute to the majority of publically available repositories.

Future Scope

- Adding monetization in the form of actionable steps to improve package health.
- Could be expanded to work on the security of cloud infrastructure, other package managers, and source code hosting platforms.
- Seamless integrations into developer tooling, workflows, and automation pipelines.