A Federated Source Code Quality Query and Analysis Platform

Tugkan Tuglular

Onur Leblebici

Emre Baran Karaca

Naşit Uygun

Osman Anıl Hiçyılmaz

SOFTENG 2023

Outline

- Problem definition
- Solution proposal
- Benefits of the solution
- The FSCQQA platform
- Conclusion

Problem Description

Each site/partner does not want to share all of its source code but needs to be queried whether holding a pre-determined minimum source code quality level so that a certain level across the consortium is achieved and maintained.

Consortium Case

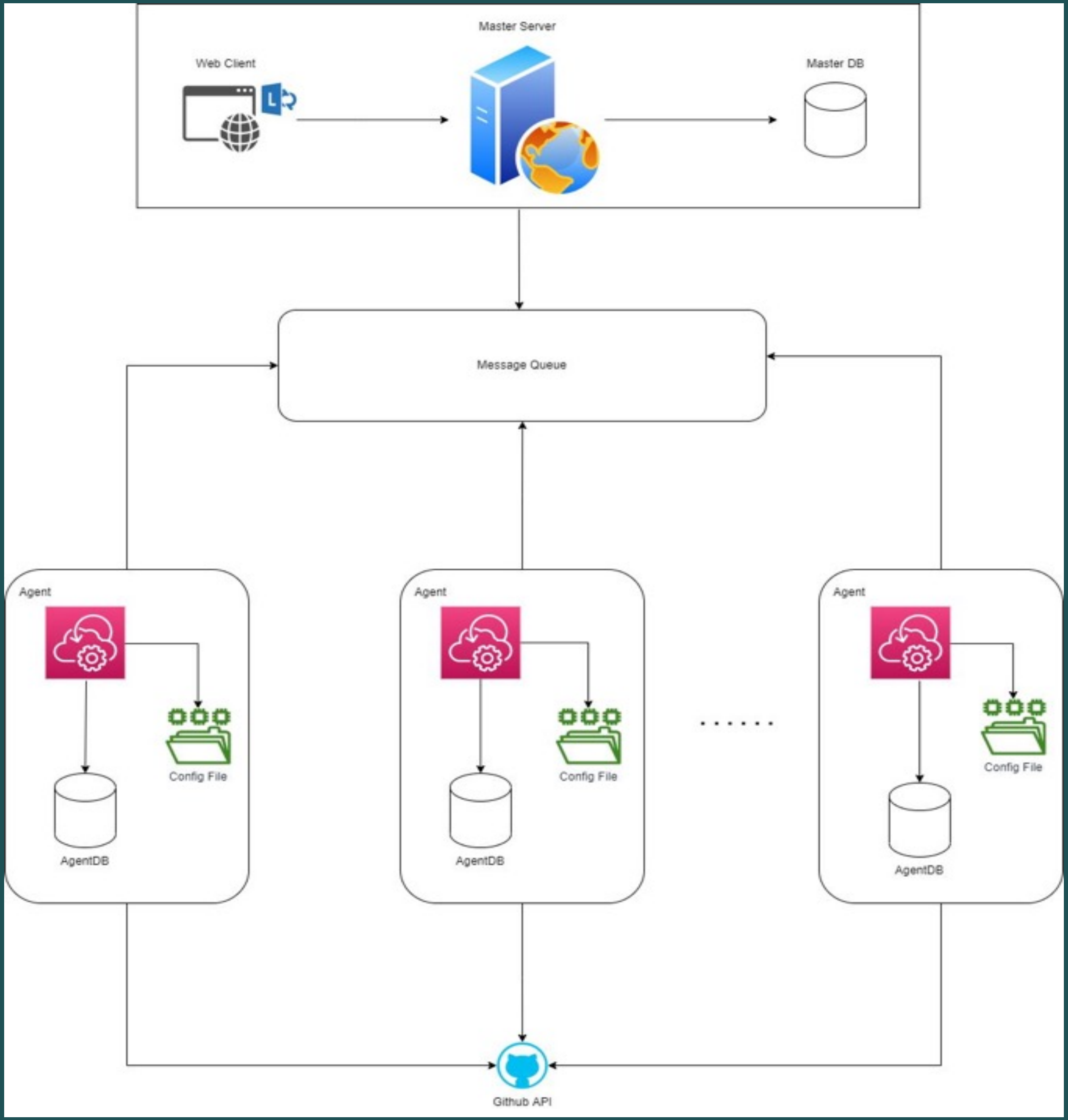
Large Company case

Solution

To build a federated network so that each node in this network has its privacy, but shares required quality information.

This paper considers this setting for source code quality and proposes a Federated Source Code Quality Query and Analysis (FSCQQA) platform.

The FSCQQA platform overview



The FSCQQA platform

consists of a central site and multiple sites, which are peers.

is a kind of peer-to-peer network, where the peers accept and follow a general policy and corresponding rules, but they can also enforce their own policies and rules.

The central site is responsible for inclusion and removal of peer sites with respect to the general policy.

The FSCQQA platform

This platform offers opportunities for querying and monitoring source code quality across a consortium.

This platform can facilitate analyzing how source code improvements are performed and how defect numbers are minimized.

Features:

- Analyze software quality with defect and source code metrics.
- Share defect and source code metrics with peers and consortium management.
- Follow trends and improve.
- Compile federated historical data on defects and source code quality.

Benefit

- By using federated approach, companies can collaborate and share data without actually sharing their code repository, which can help protect the privacy and intellectual property of the organizations involved.
- This can be particularly important for companies that rely on proprietary code and algorithms for their business.

Future Benefit

By using federated learning to predict bugs, companies can improve their code quality and reduce the number of bugs in their software.

This can lead to cost savings, improved customer satisfaction, and increased efficiency.

Future Benefit

Federated learning can also help reduce the environmental impact of training machine learning models.

By training models on data that does not need to be transmitted between server-client, there is less energy consumption and fewer carbon emissions associated with the process.

Fundamentals

Bug Datasets

Several research works have produced and utilized bug datasets to develop and evaluate novel bug prediction methods

The objective of their study is to collect and combine current public source code metrics-based bug databases.

They evaluated the abundance of gathered metrics and the bug prediction skills of the unified bug dataset.

One research direction in this field moves toward combining bug datasets with software code quality metrics for better prediction.

Source Code Quality Metrics

Nuñez-Varela et al.[1] did a comprehensive mapping investigation on 226 articles that were published between 2010 and 2015 and discovered nearly 300 source code metrics.

Even though object-oriented metrics have received a great deal of attention, there is a need for greater research on aspect and feature-oriented measurements.

Prediction of software faults, complexity, and quality evaluation were recurring themes in these investigations.

Currently, there are separate tools as well as tools embedded into platforms, which not only produce source code quality metrics but also calculate technical debt.

The next step for these tools seems to be towards predictions and suggestions for better code quality.

[1] A. S. Nuñez-Varela, H. G. Pérez-Gonzalez, F. E. Martínez-Perez, and C. Soubervielle-Montalvo, “Source code metrics: A systematic mapping study,” Journal of Systems and Software, vol. 128, pp. 164–197, 2017.

The FSCQQA platform

Design Goals

- Authentication & Authorization
- Access Control
- Secure Communication
- Logging and Monitoring
- Standard APIs
- Source Code Repositories
- Management of Federated Platform

The FSCQQA Standard APIs

provide the services of the FSCQQA platform with respect to Open-API specifications [14]. The services are grouped as follows:

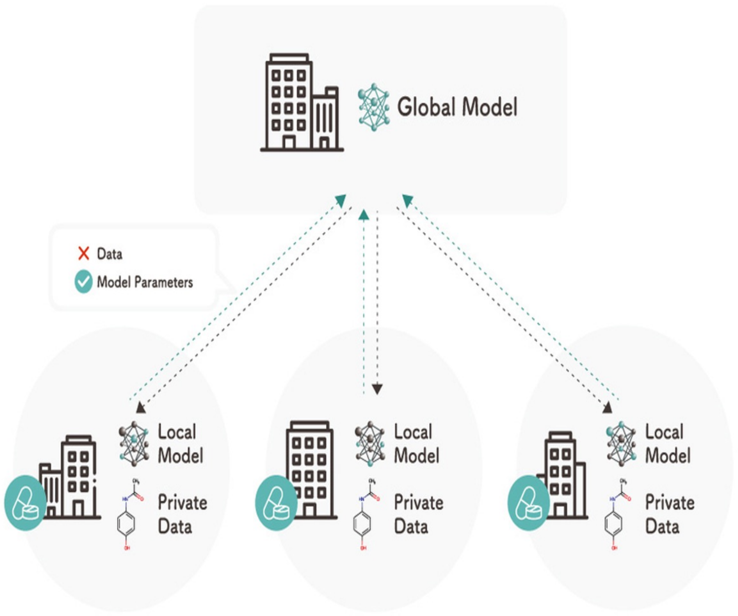
- ✓ Defect related metrics: number of existing (active) defects, defect density, defect resolve velocity, longest unresolved defect.
- ✓ Source code related metrics: class metrics, method metrics, coupling metrics, cohesion metrics, cyclomatic complexity metrics.

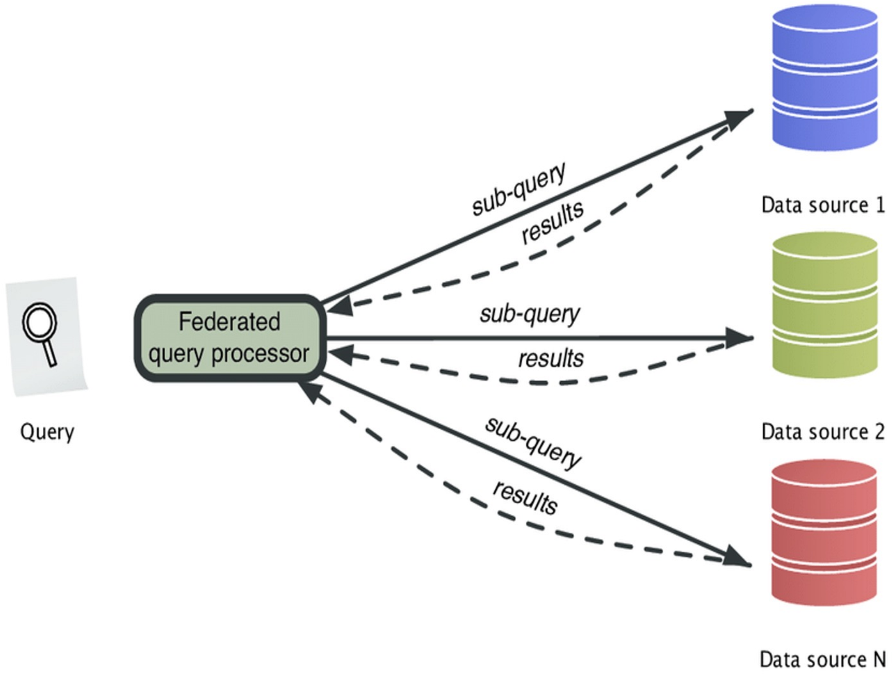
To mitigate security concerns related to standard APIs, their source code should be open.

The FSCQQA Standard APIs

The service calls can be for a specific metric or a set of metrics from a specific site or the whole network.

If the whole network is queried, the query site requests all alive sites from the central site and queries each one individually then accumulates the results.



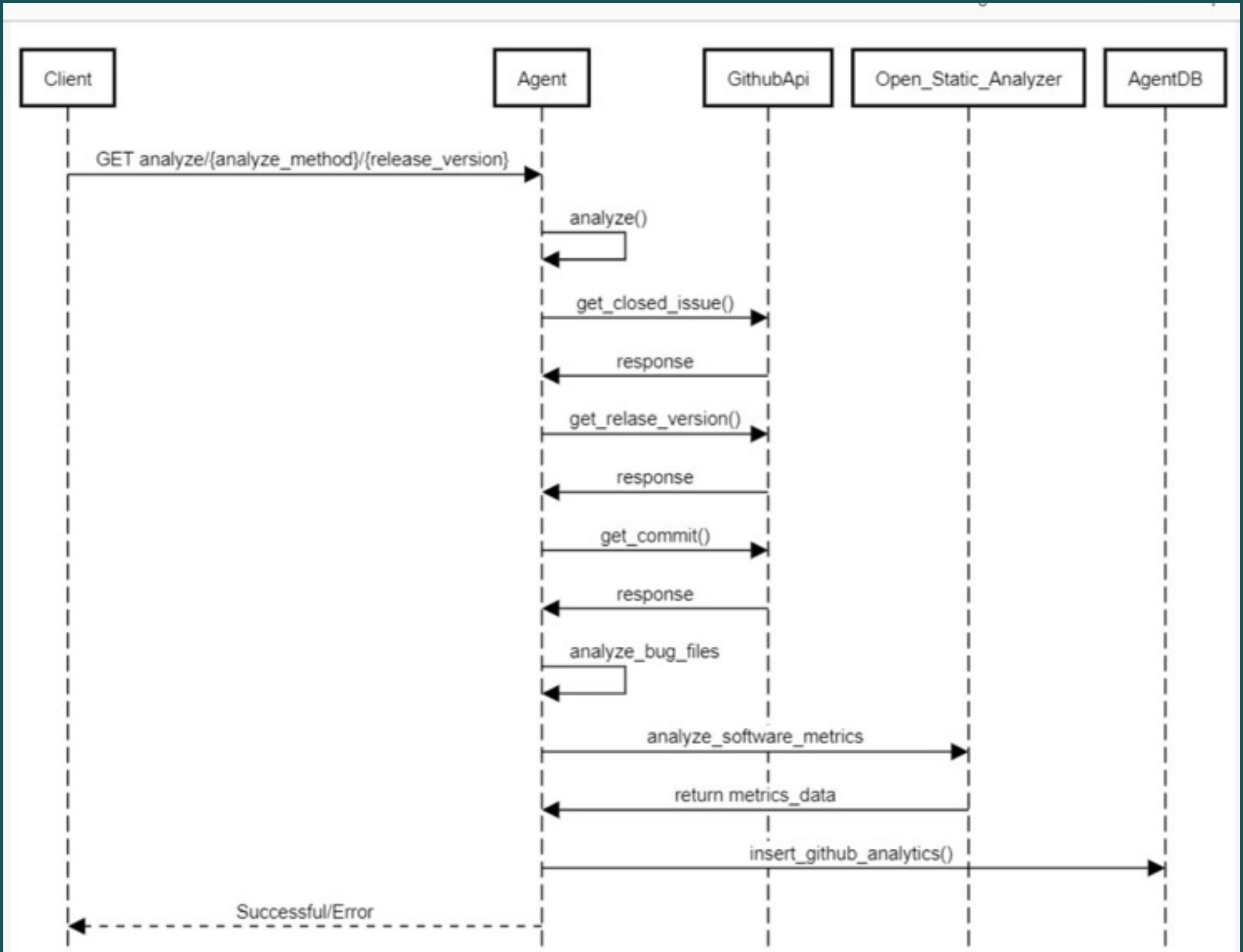


The FSCQQA agent

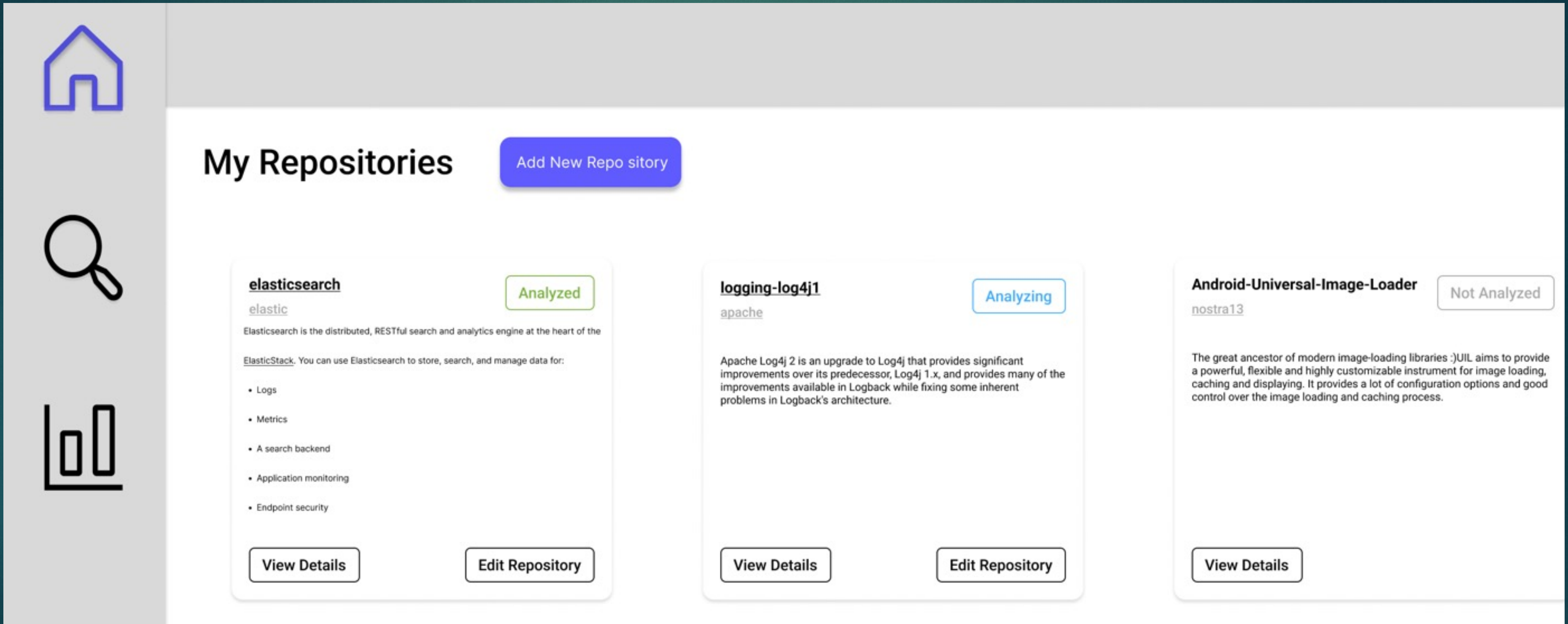
- ✓ generates local defect database for each site from a GitHub repository by extracting commit/issue histories and analyzing them.
- ✓ collects software metrics, such as lines of code and cyclomatic complexity, for each commit/issue.
- ✓ extracts source code related metrics for a specific version using tools, such as OpenStaticAnalyzer
- ✓ manages the local database for defects and metrics.

To mitigate security concerns related to the FSCQQA agent, its source code should be open.

The FSCQQA agent



The FSCQQA User Interface



My Repositories

elasticsearch

elastic

Elasticsearch is the distributed, RESTful search and analytics engine at the heart of the ElasticStack. You can use Elasticsearch to store, search, and manage data for:

- Logs
- Metrics
- A search backend
- Application monitoring
- Endpoint security

logging-log4j1

apache

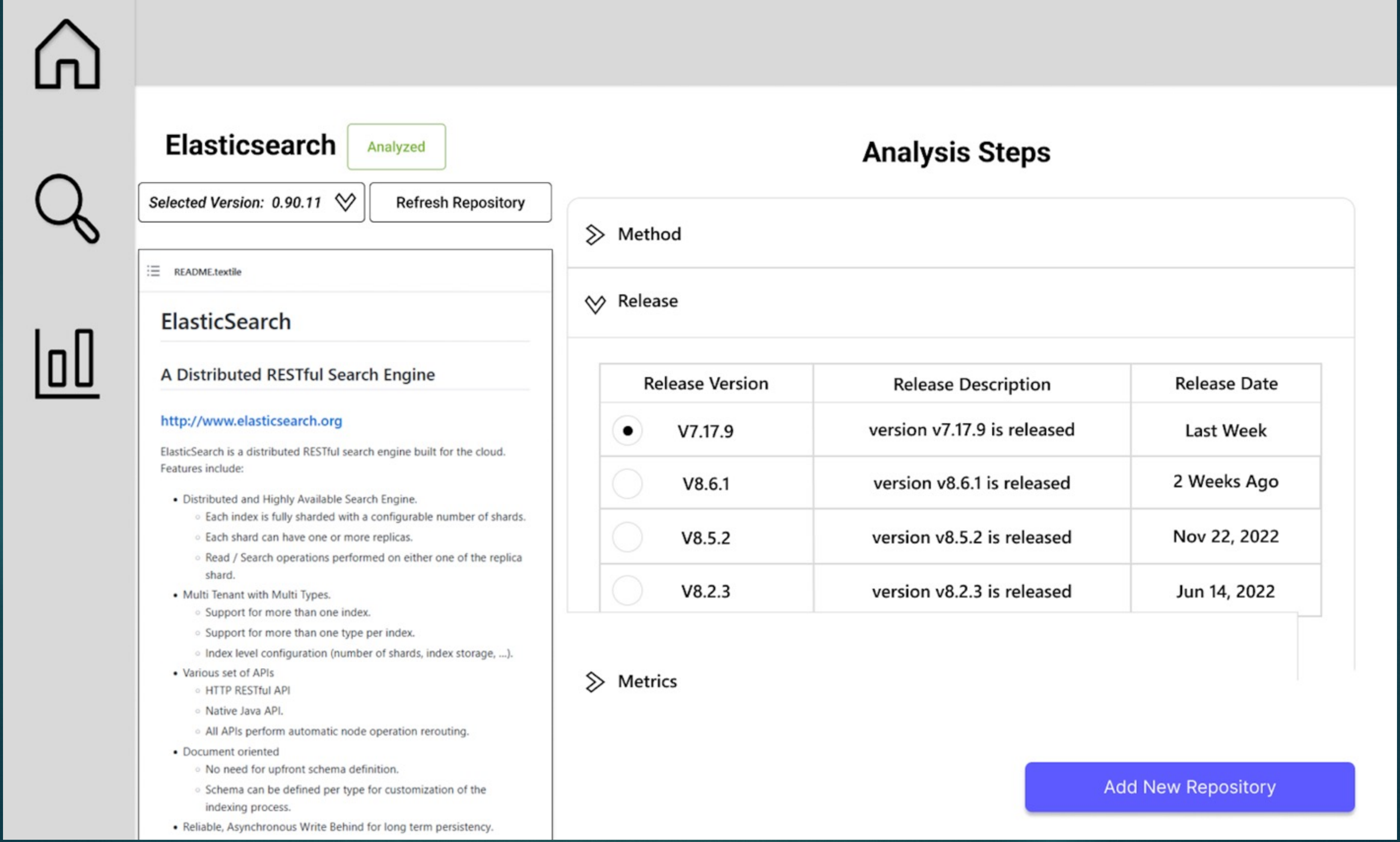
Apache Log4j 2 is an upgrade to Log4j that provides significant improvements over its predecessor, Log4j 1.x, and provides many of the improvements available in Logback while fixing some inherent problems in Logback's architecture.

Android-Universal-Image-Loader

nostra13

The great ancestor of modern image-loading libraries :)UIL aims to provide a powerful, flexible and highly customizable instrument for image loading, caching and displaying. It provides a lot of configuration options and good control over the image loading and caching process.

The FSCQQA User Interface



ElasticSearch

A Distributed RESTful Search Engine

http://www.elasticsearch.org

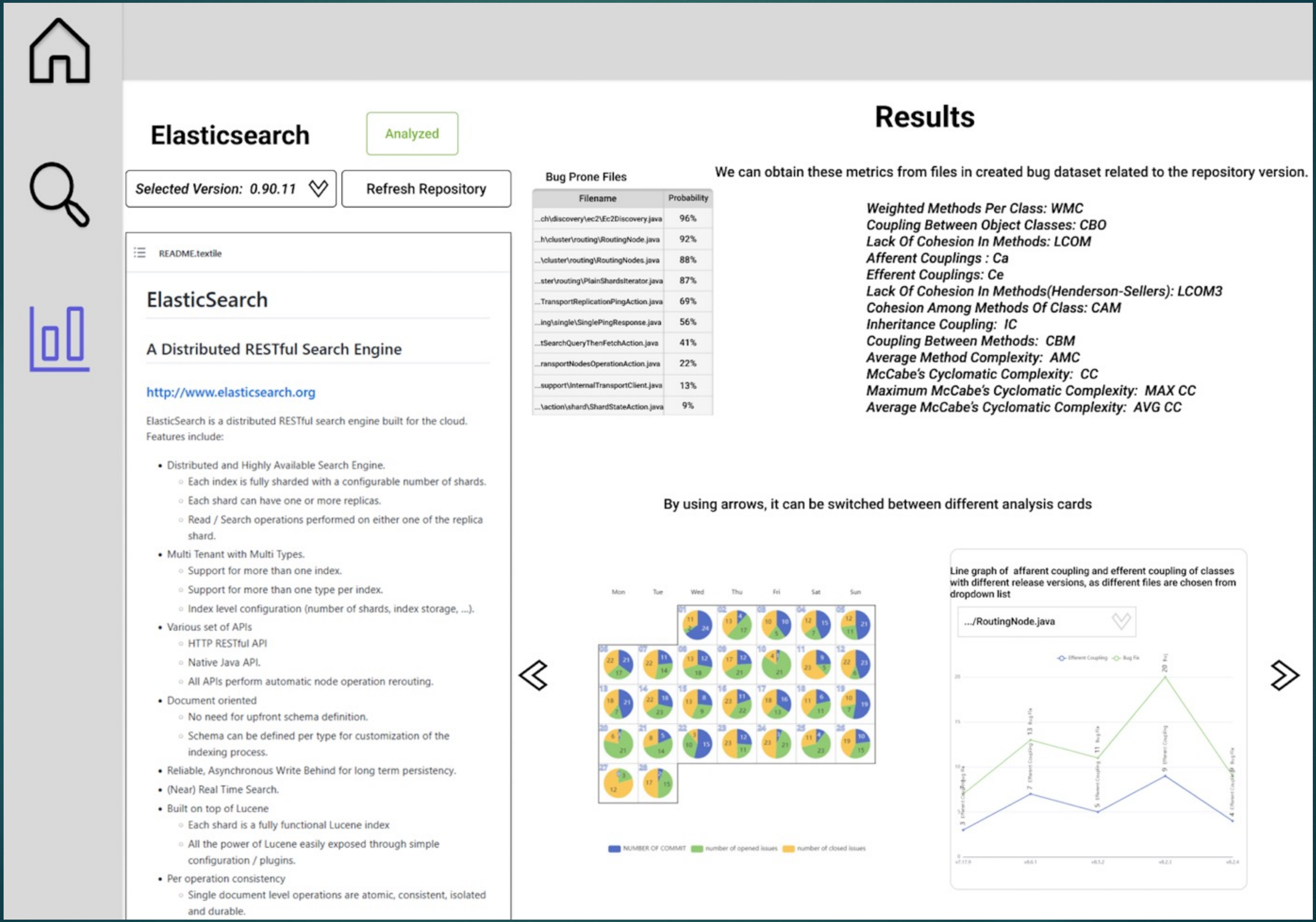
ElasticSearch is a distributed RESTful search engine built for the cloud. Features include:

- Distributed and Highly Available Search Engine.
  - Each index is fully sharded with a configurable number of shards.
  - Each shard can have one or more replicas.
  - Read / Search operations performed on either one of the replica shard.
- Multi Tenant with Multi Types.
  - Support for more than one index.
  - Support for more than one type per index.
  - Index level configuration (number of shards, index storage, ...).
- Various set of APIs
  - HTTP RESTful API
  - Native Java API.
  - All APIs perform automatic node operation rerouting.
- Document oriented
  - No need for upfront schema definition.
  - Schema can be defined per type for customization of the indexing process.
- Reliable, Asynchronous Write Behind for long term persistency.

Analysis Steps

	Release Version	Release Description	Release Date
●	V7.17.9	version v7.17.9 is released	Last Week
○	V8.6.1	version v8.6.1 is released	2 Weeks Ago
○	V8.5.2	version v8.5.2 is released	Nov 22, 2022
○	V8.2.3	version v8.2.3 is released	Jun 14, 2022

The FSCQQA User Interface



Elasticsearch

Selected Version: 0.90.11

ElasticSearch

A Distributed RESTful Search Engine

http://www.elasticsearch.org

ElasticSearch is a distributed RESTful search engine built for the cloud. Features include:

- Distributed and Highly Available Search Engine.
  - Each index is fully sharded with a configurable number of shards.
  - Each shard can have one or more replicas.
  - Read / Search operations performed on either one of the replica shard.
- Multi Tenant with Multi Types.
  - Support for more than one index.
  - Support for more than one type per index.
  - Index level configuration (number of shards, index storage, ...).
- Various set of APIs
  - HTTP RESTful API
  - Native Java API.
  - All APIs perform automatic node operation rerouting.
- Document oriented
  - No need for upfront schema definition.
  - Schema can be defined per type for customization of the indexing process.
- Reliable, Asynchronous Write Behind for long term persistency.
- (Near) Real Time Search.
- Built on top of Lucene
  - Each shard is a fully functional Lucene index
  - All the power of Lucene easily exposed through simple configuration / plugins.
- Per operation consistency
  - Single document level operations are atomic, consistent, isolated and durable.

Filename	Probability
...ch\discovery\ec2\Ec2Discovery.java	96%
...h\cluster\routing\RoutingNode.java	92%
...\cluster\routing\RoutingNodes.java	88%
...ster\routing\PlainShardsIterator.java	87%
...TransportReplicationPingAction.java	69%
...ing\single\SinglePingResponse.java	56%
...tSearchQueryThenFetchAction.java	41%
...ransportNodesOperationAction.java	22%
...support\InternalTransportClient.java	13%
...\action\shard\ShardStateAction.java	9%

We can obtain these metrics from files in created bug dataset related to the repository version.

Weighted Methods Per Class: WMC
Coupling Between Object Classes: CBO
Lack Of Cohesion In Methods: LCOM
Afferent Couplings : Ca
Efferent Couplings: Ce
Lack Of Cohesion In Methods(Henderson-Sellers): LCOM3
Cohesion Among Methods Of Class: CAM
Inheritance Coupling: IC
Coupling Between Methods: CBM
Average Method Complexity: AMC
McCabe's Cyclomatic Complexity: CC
Maximum McCabe's Cyclomatic Complexity: MAX CC
Average McCabe's Cyclomatic Complexity: AVG CC

By using arrows, it can be switched between different analysis cards

Line graph of affarent coupling and efferent coupling of classes with different release versions, as different files are chosen from dropdown list

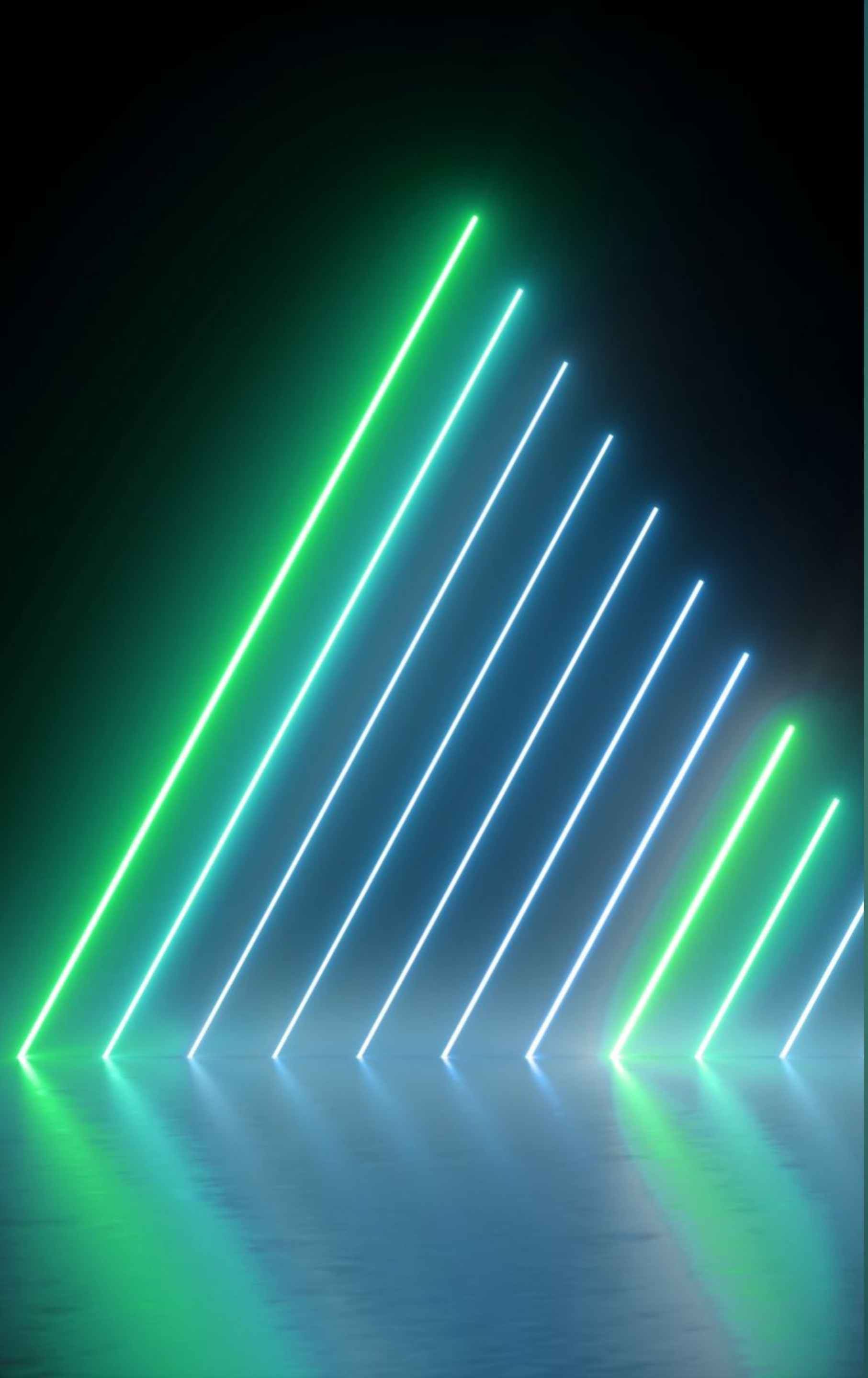
Conclusion

With the proposed FSCQQA platform, sites are not required to disclose their codes with any other site while aiming for high source code quality and low defect ratio.

At each site, local defect datasets will be generated and analyzed.

The analysis results as defect metrics and the source code metrics obtained from the static analysis will be shared within the federated network and can be queried.

Trend analysis can be conducted at the central site and shared with consortium sites.



Future Work

- Machine Learning model validation and performance improvement.
- Check project status from mobile application.
- New Git hosting service
- Extension for IDEs

THANK YOU…