Shining Light on the Open Source Supply Chain: The Risk in Community Health

Georg Link, PhD

SCaLE 22x, March 8, 2025, Pasadena, CA, USA

Abstract

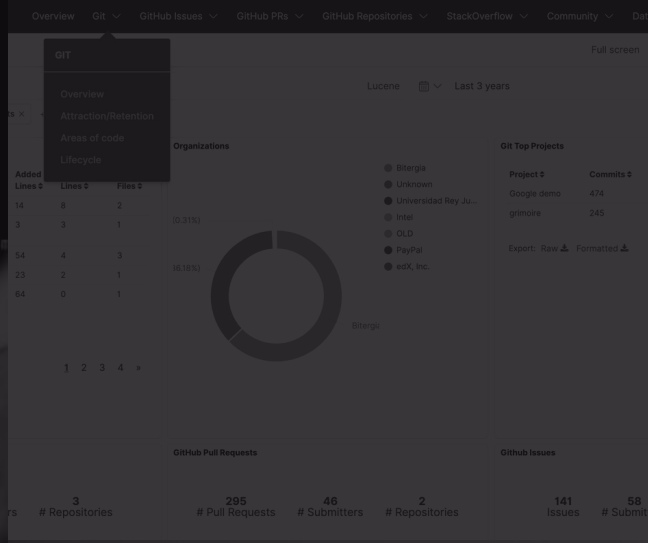
This talk introduces the open source tool GrimoireLab that can shine lights onto those dark corners of your open source supply chain. We will also show how GrimoireLab was used in a novel Risk Assessment Model for the Maturity and Sustainability of open source dependencies, designed to address this critical challenge.
By using the GrimoireLab tool, combining concepts from the CHAOSS project and cloud-native deployment maturity models, our approach goes beyond traditional Software Bill of Materials (SBOM) analysis to evaluate the ongoing maintenance activity and community health of OSS projects. This enables organizations to:
- Assess the long-term viability of their open source dependencies.
- Make informed decisions about library selection and integration.
- Proactively mitigate risks associated with unhealthy or unsustainable communities.

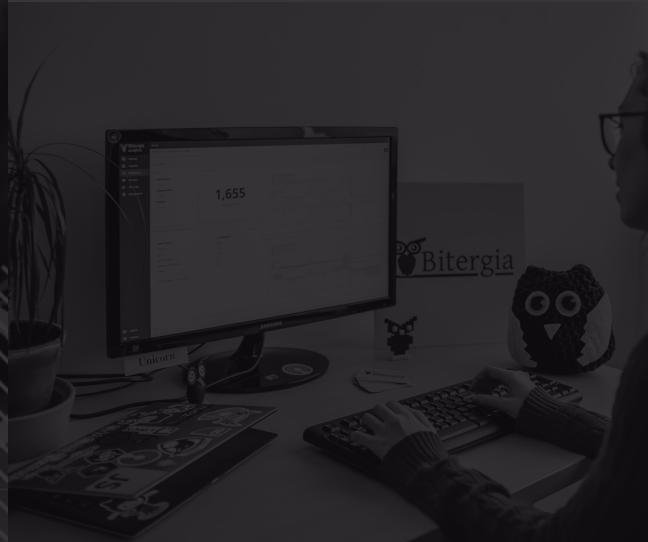
This talk will delve into the model's design and implementation with GrimoireLab, using Kubernetes as a case study. By adopting this approach, organizations can build a more resilient and sustainable software foundation, ensuring the long-term health of their open source supply chain.
Join us in prioritizing the health of open-source communities! Discover how supporting these vital ecosystems can enhance your development processes and safeguard your supply chain.

This enables organizations to:

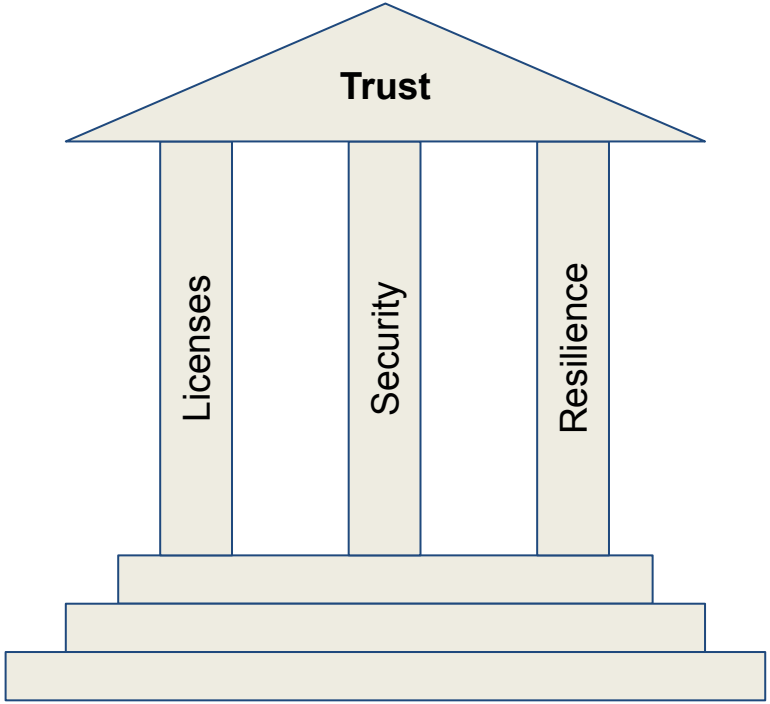
- Assess the long-term viability of their open source dependencies.
- Make informed decisions about library selection and integration.
- Proactively mitigate risks associated with unhealthy or unsustainable communities.

One Big Idea


Overview
Git
GitHub Issues
GitHub PRs
GitHub Repositories
StackOverflow
Community
GIT
Overview
Attraction/Retention
Areas of code
Lifecycle
Full screen
Lucene
Last 3 years
Organizations
Bitergia
Unknown
Universidad Rey Ju...
Intel
OLD
PayPal
eqX, Inc.
Git Top Projects
Project
Commits
Google demo
474
grimoire
245
Export: Raw Formatted
GitHub Pull Requests
Github Issues
3
Repositories
295
Pull Requests
46
Submitters
2
Repositories
141
Issues


1,655
Bitergia

Three Pillars OSS Strategy to build Trust



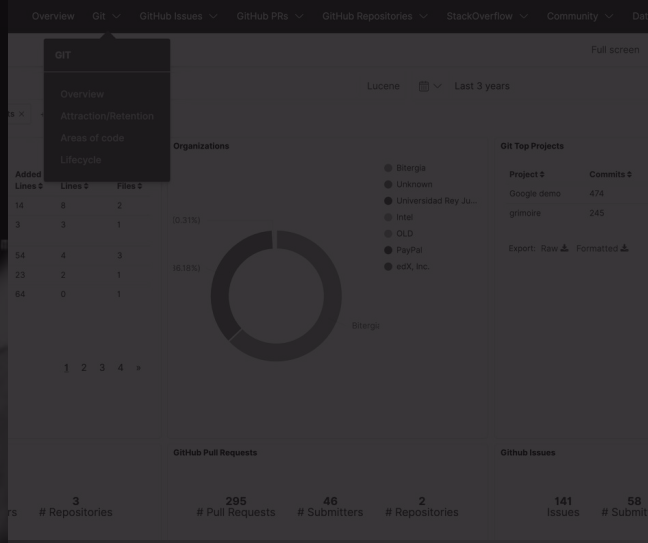
Community Activity Indicates Resilience

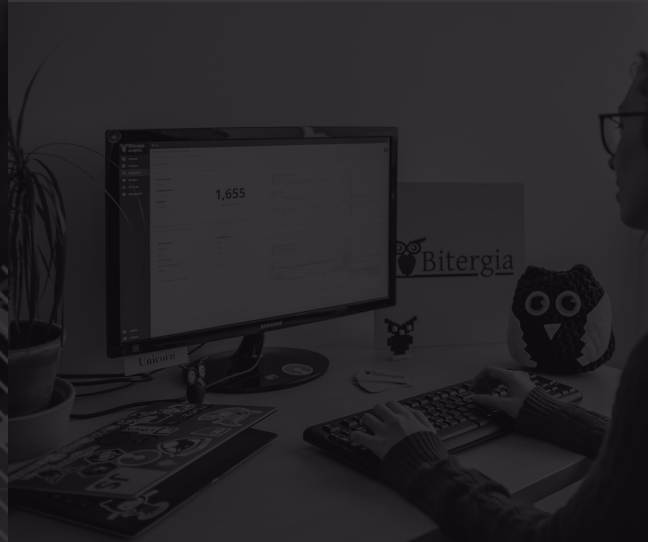
Thrive ?

Abandon?

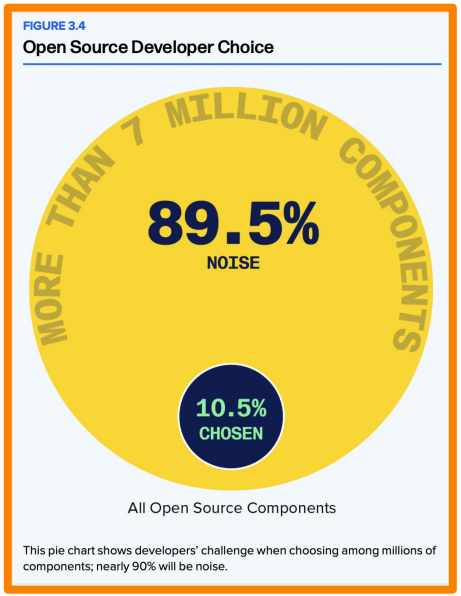
The community activity today is a leading indicator for the software project's future.

Why we care


Overview
Git
GitHub Issues
GitHub PRs
GitHub Repositories
StackOverflow
Community
GIT
Overview
Attraction/Retention
Areas of code
Lifecycle
Full screen
Lucene
Last 3 years
Organizations
Bitergia
Unknown
Universidad Rey Ju...
Intel
OLD
PayPal
eqX, Inc.
Git Top Projects
Project
Commits
Google demo
474
grimoire
245
Export: Raw Formatted
GitHub Pull Requests
Github Issues
3
Repositories
295
Pull Requests
46
Submitters
2
Repositories
141
Issues


1,655
Bitergia

Dilemma: Choice and Maintenance



https://www.sonatype.com/state-of-the-software-supply-chain

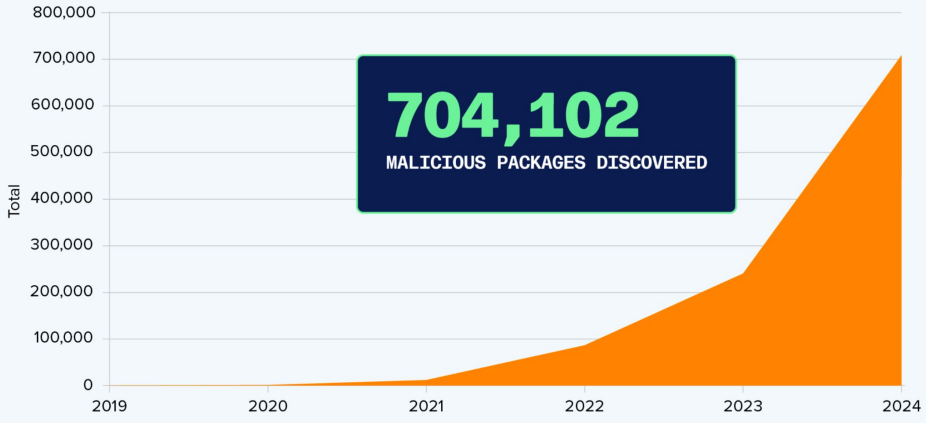
Sonatype 9th State of Software Supply Chain report:

"Consider this: last year, we revealed that a staggering **85% of projects in Maven Central** — the largest public repository for Java open source components — **are inactive.** In other words, developers are faced with a perplexing array of choices, with only a fraction of them leading to active, well-maintained projects."

Software Supply Chain Attacks are on the Rise

FIGURE 1.1

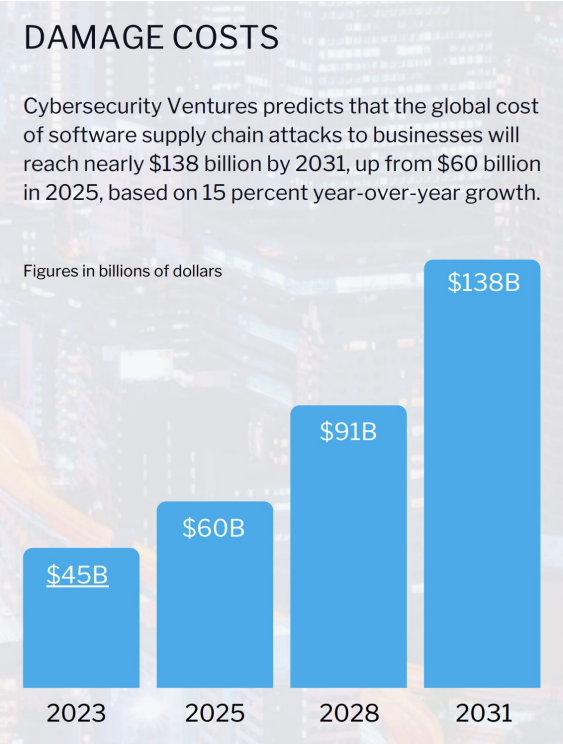
Next Generation Software Supply Chain Attacks (2019-2024)



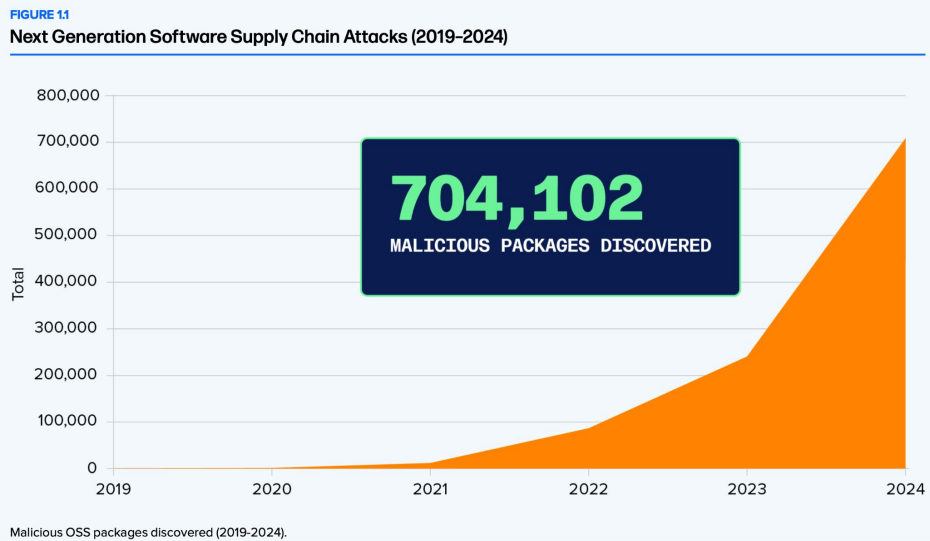
Malicious OSS packages discovered (2019-2024).

https://www.sonatype.com/state-of-the-software-supply-chain

Software Supply Chain Attacks are on the Rise

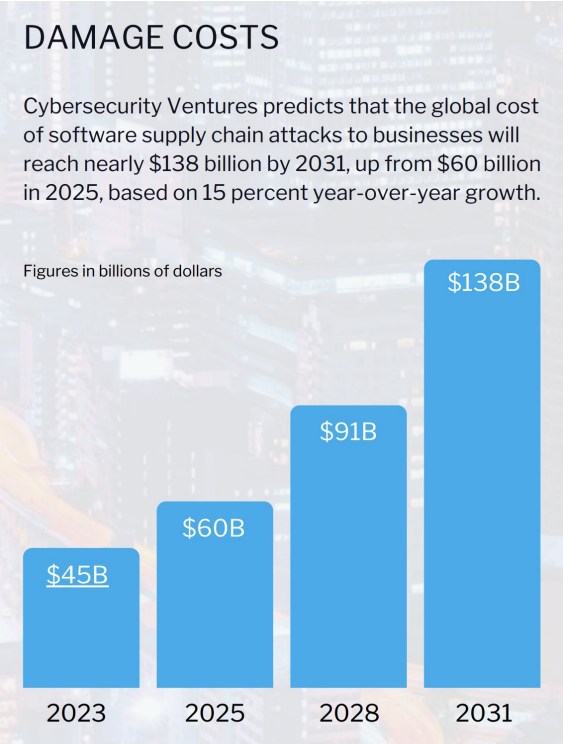


https://go.snyk.io/2023-supply-chain-attacks-report.html



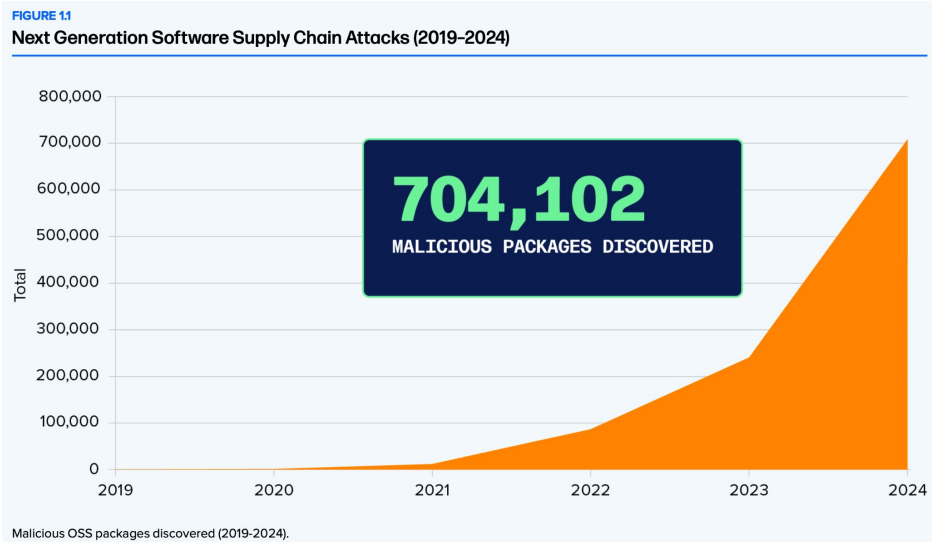
https://www.sonatype.com/state-of-the-software-supply-chain

Software Supply Chain Attacks are on the Rise



https://go.snyk.io/2023-supply-chain-attacks-report.html

Gartner predicts that by 2025, **45 percent** of organizations worldwide will have experienced attacks on their software supply chains, a three-fold increase from 2021.
(Gartner, Mar 7, 2022)



https://www.sonatype.com/state-of-the-software-supply-chain

Research Found the SolarWinds Cyber Attack Cost Affected Companies in Key Sectors 11% of Total Annual Revenue on Average

Results indicate cyber-related information sharing is increasing, signaling a positive response to national-and industry-level calls to action

By Business Wire

Jun 28, 2021

Meet Georg Link

Open Source Strategist

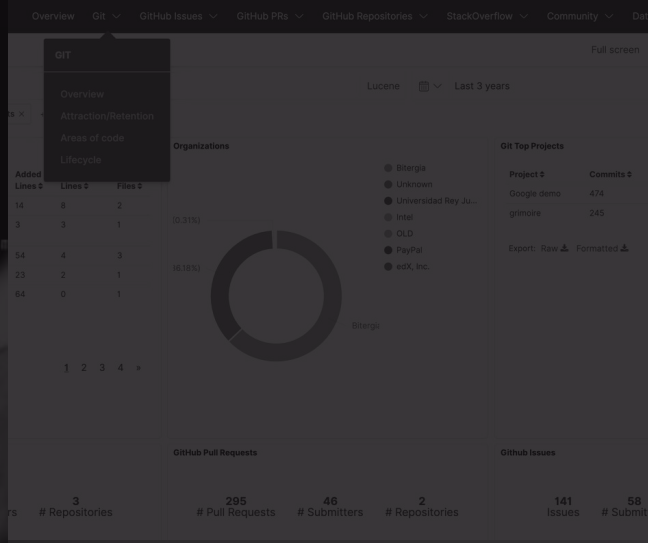
- Business focus
- 20+ years in open source
- Co-Founder of CHAOSS
- Community Builder

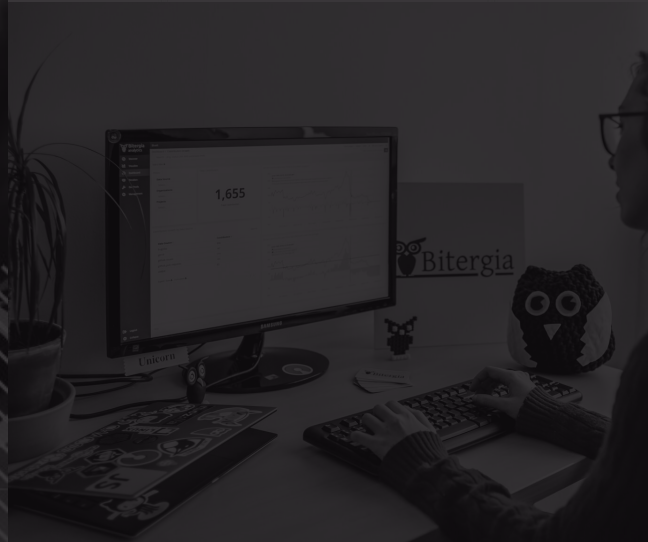
“My mission is to improve the health and sustainability of open source.”

Agenda

- SBOMs
- Trust: Risk Assessment Model
- Example of Kubernets' Go Dependencies
- GrimoireLab: The Open Source Tool

SBOMs


Overview
Git
GitHub Issues
GitHub PRs
GitHub Repositories
StackOverflow
Community
GIT
Overview
Attraction/Retention
Areas of code
Lifecycle
Full screen
Lucene
Last 3 years
Organizations
Bitergia
Unknown
Universidad Rey Ju...
Intel
OLD
PayPal
eqX, Inc.
Git Top Projects
Project
Commits
Google demo
474
grimoire
245
Export: Raw Formatted
GitHub Pull Requests
Github Issues
3
Repositories
295
Pull Requests
46
Submitters
2
Repositories
141
Issues

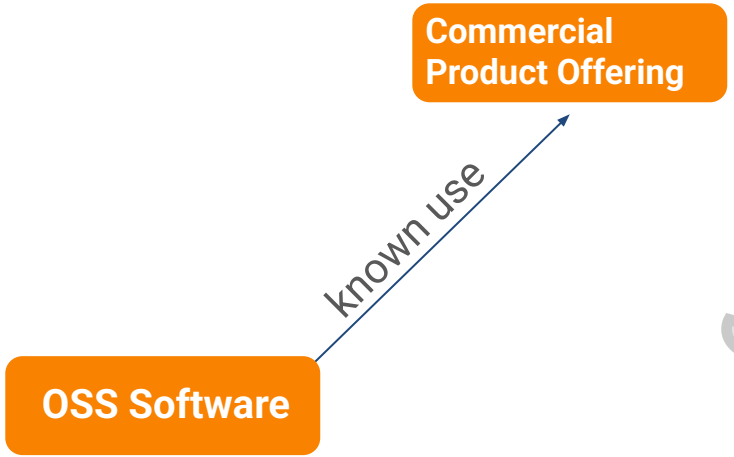

1,655
Bitergia

Software ages like Milk, not Wine

Trust: Expiration Label and Source Information

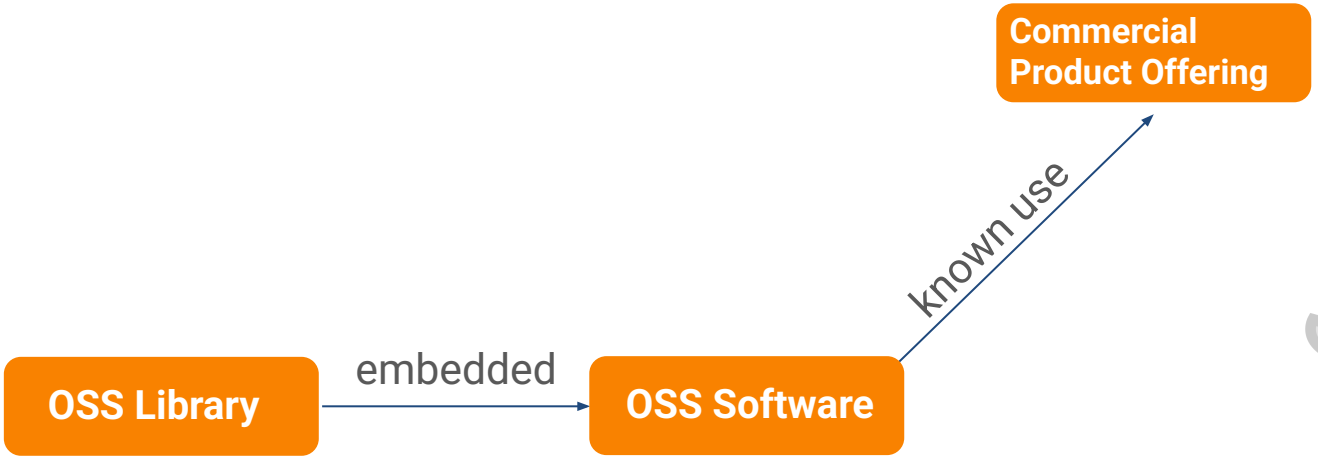
Context: Unmanaged OSS Use → Unknown Risk

- Developers use open source software
- 70% - 95% of software includes open source



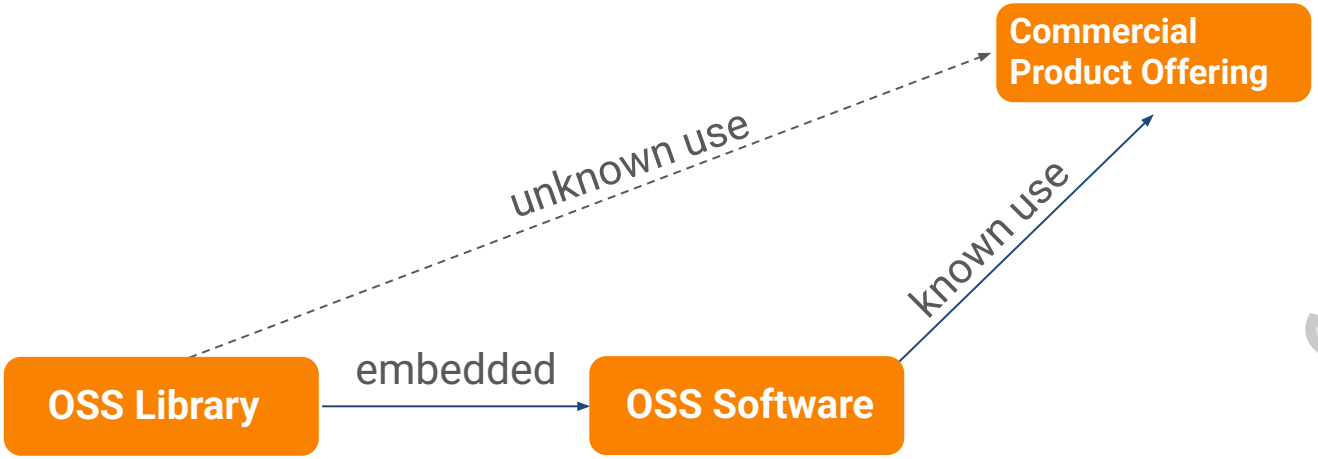
Context: Unmanaged OSS Use → Unknown Risk

- Developers use open source software
- 70% - 95% of software includes open source
- Unmanaged OSS use → unknown dependencies



Context: Unmanaged OSS Use → Unknown Risk

- Developers use open source software
- 70% - 95% of software includes open source
- Unmanaged OSS use → unknown dependencies → **unknown risk**



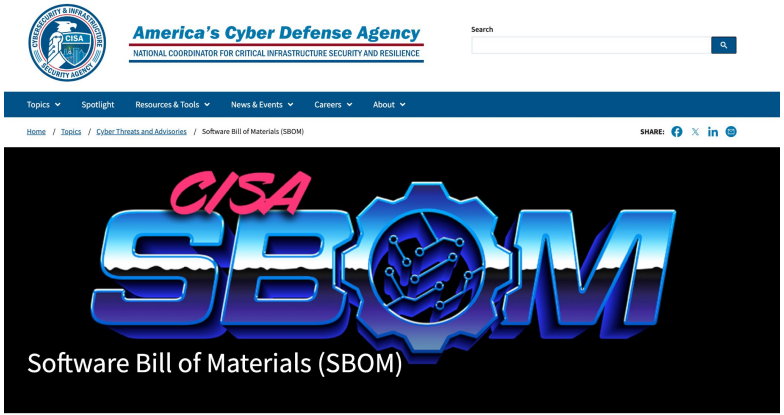
Articulated: Software Bill of Material (SBOM)

An SBOM is a nested inventory, a list of ingredients that make up software components.

Regulatory Pressure towards SBOMs since:

Executive Order 14028 of May 12, 2021 (Improving the Nation's Cybersecurity)

https://bidenwhitehouse.archives.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/



https://www.cisa.gov/sbom

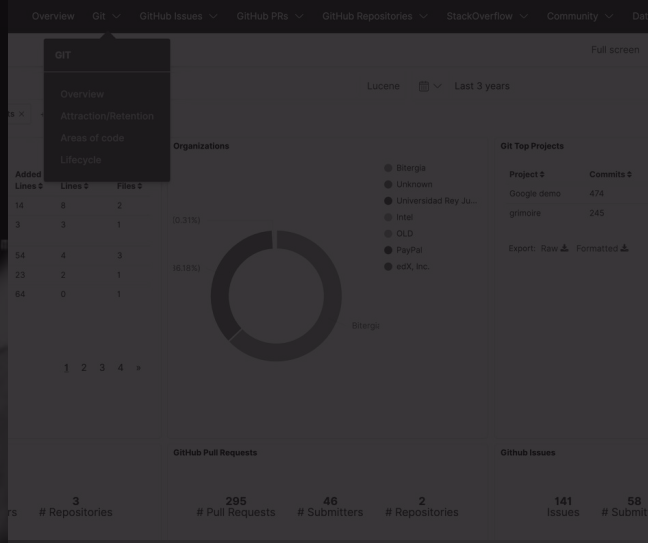
Latest driver: Cyber Resilience Act (CRA)

(34) When integrating components sourced from third parties in products with digital elements during the design and development phase, manufacturers should, in order to ensure that the products are designed, developed and produced in accordance with the

When integrating components sourced from third parties … manufacturers should, … exercise due diligence with regard to those components, including free and open-source software components …

the market and for the support period, apply to products with digital elements in their entirety, including to all integrated components. Where, in the exercise of due diligence, the manufacturer of the product with digital elements identifies a vulnerability in a component, including in a free and open-source component, it should inform the person or entity manufacturing or maintaining the component, address and remediate the vulnerability, and, where applicable, provide the person or entity with the applied security fix.

Trust:
Risk Assessment Model

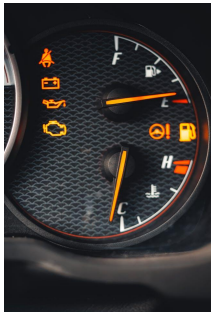


Imagine a Car

State Today - relying on instruments:

- No Gas
- Flat Tires
- Warning Symbols
- Error Codes

→ You know how to fix today's situation



Future Support - leveraging origin information:

- Availability of Replacement Parts
- Skilled Workers to Repair
- Network of Repair shops
- Life Expectancy of Car

→ Unsupported Oldtimer vs. Supported Modern Car



Imagine a Car - a metaphor for software

State Today - relying on instruments:

- Flat Tires
- No
- Wa
- Err

→ You k uation

Licenses

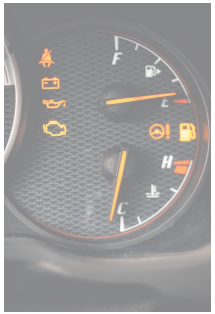
Security

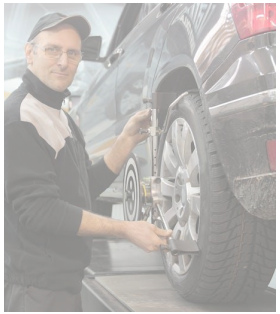
Future Support - leveraging origin information:

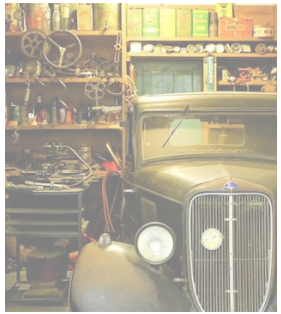
- Availability of Replacement Parts
- Skill
- Netv
- Life

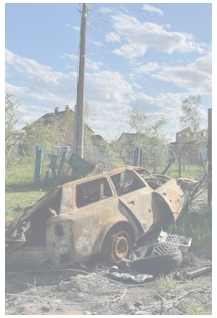
→ Unsupp ted Modern Car

Resilience









Analysis: Trust in OSS Libraries to Manage Risk

Licenses

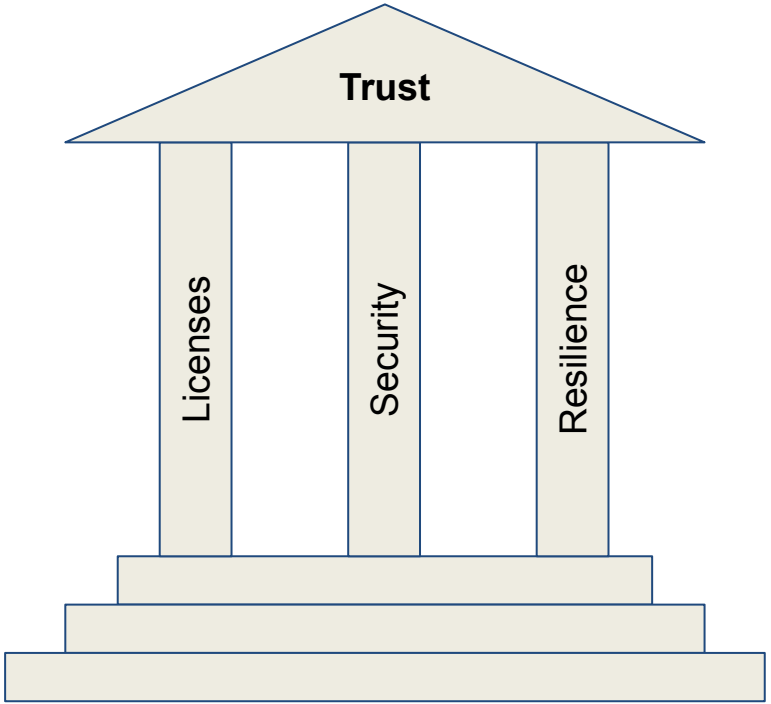
- License scanners

Security

- Software Composition Analysis (SCA)
- Vulnerability Databases

Resilience

- Community Health Metrics (CHAOSS)
  - Maintainability
  - Sustainability
  - Project Health



Thesis: Tracking OSS Resilience is Proactive Risk Management

Resilience

The capacity of an OSS project to recover quickly from difficulties and continue releasing quality software.

Risk Management

Evaluation of risks and procedures to avoid or minimize their impact.

Community Activity Indicates Resilience

Thrive ?

Abandon?

The community activity today is a leading indicator for the software project's future.

Indicators for Risk: “Under-maintained Projects”

“Community Smells” include 7 metrics:

Community cannot handle demand
- Backlog Management Index
- Review Efficiency Index

Community does not address work quickly
- Median Lead Time for Issues
- Median Lead Time for Pull Requests

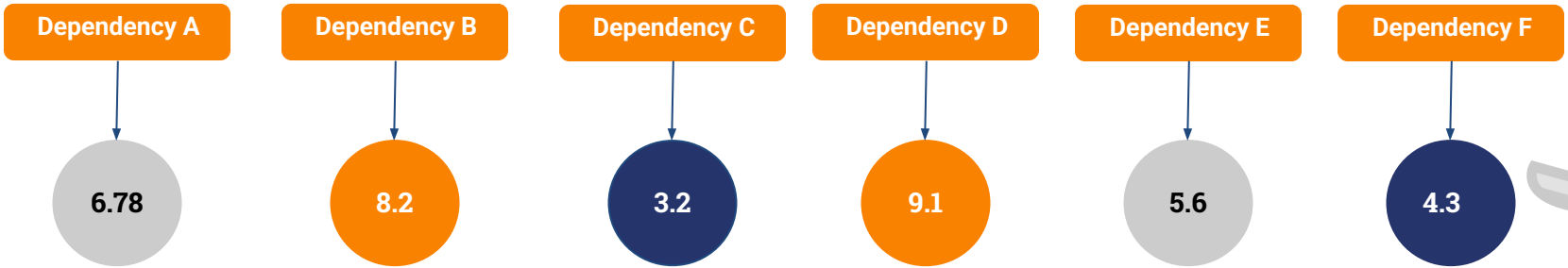
Community lacks sufficient talent
- Retention Rate
- Growth of Active Contributors
- Contributor Absence Factor (Bus or Pony Factor)

A single Risk Score per OSS library

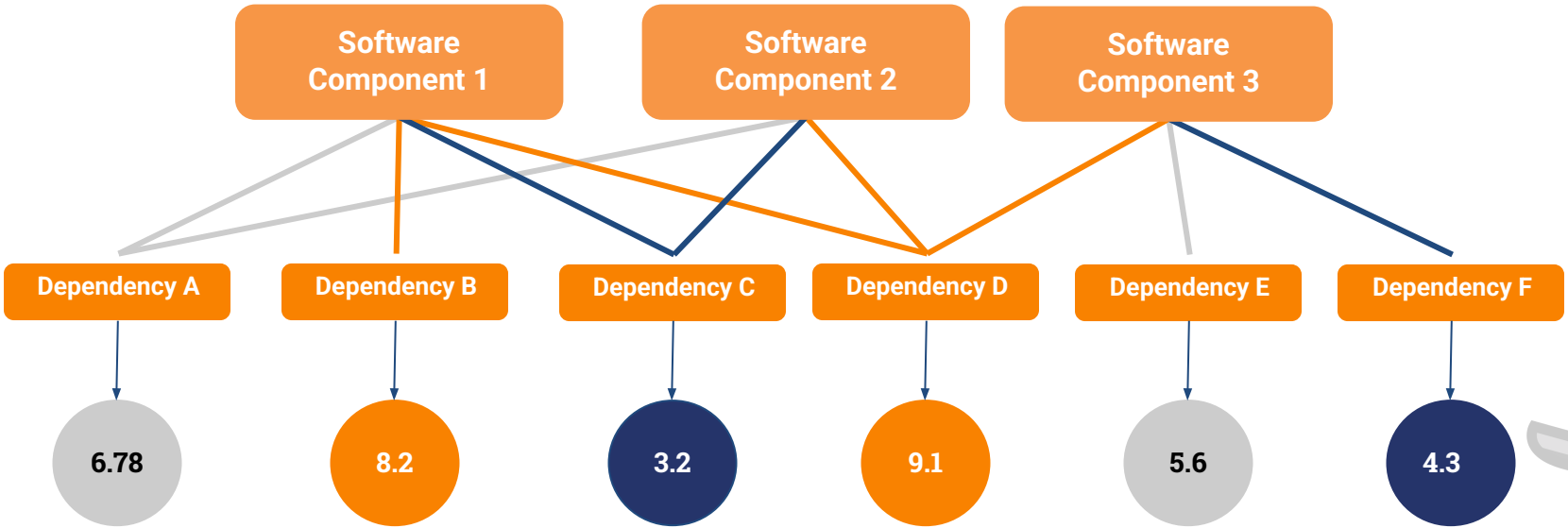
Normalized the 7 metrics

Combined into one score for each dependency

Benchmarked against datasets of similar OSS libraries



Risk Model - Aggregate By Component




Example of Kubernetes'
Go Dependencies

project: Kubernetes - Golang Deps × | Add filter

Risk Model Overview Dashboard

Check the Risk Model Help Dashboard for more information about this analysis.

For more details, pin a filter by origin and visit the Risk Model Dashboard for Individual Projects.

Filters

Team

Select...

Project Category

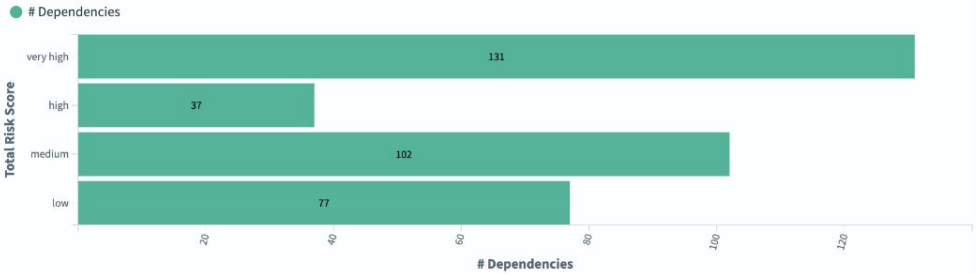
Kubernetes - Golang Deps ×

Overview

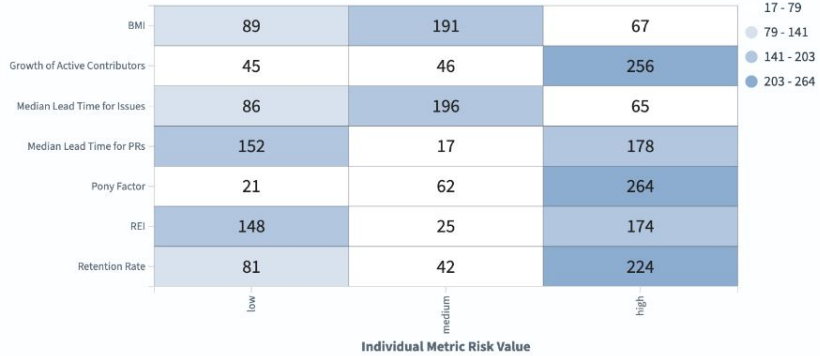
347
Dependencies analyzed

Bitergia
Team

Dependencies by Risk value



Risk Value per Metric, by number of Dependencies



Overall results by dependency repository

Filter...

Repository	Category	Risk Value	Risk Score (over 10)	# "Low risk" metrics	# "Medium risk" metrics	# "High risk" metrics	Last analyzed on
https://github.com/json-iterator/go	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/spf13/afero	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/Azure/go-ansiterm	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/ThalesIgnite/crypto11	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/coreos/go-semver	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/curioswitch/go-reassign	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/davecgh/go-spew	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35

Export: Raw Formatted

1 2 3 4 5 ... 33

Risk Model Overview Dashboard

Check the Risk Model Help Dashboard for more information about this analysis.

For more details, pin a filter by origin and visit the Bi

Filters

Team

Select...

Project Category

Kubernetes - Golang Deps ×

Bitergia
Team

Dependencies by Risk value

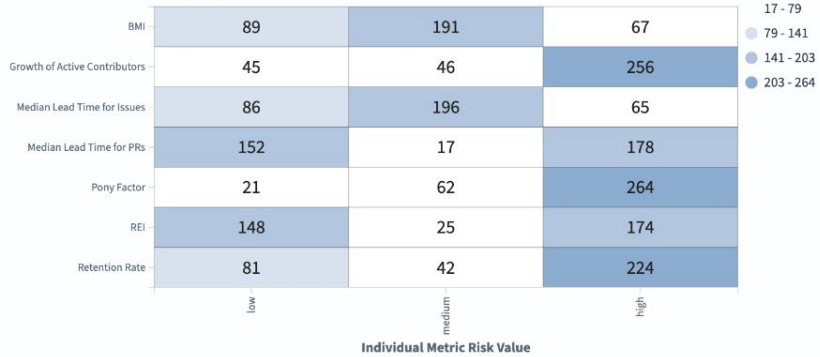
Dependencies

very high 131

Project Category

Kubernetes - Golang Deps ×

Risk Value per Metric, by number of Dependencies



Filter...

Repository	Category	Risk Value	Risk Score (over 10)	# "Low risk" metrics	# "Medium risk" metrics	# "High risk" metrics	Last analyzed on
https://github.com/json-iterator/go	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/spf13/afero	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/Azure/go-ansiterm	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/ThalesIgnite/crypto11	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/coreos/go-semver	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/curioswitch/go-reassign	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/davecgh/go-spew	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35

Export: Raw Formatted

1 2 3 4 5 ... 34

Bitergia Analytics

Dashboards | Demo - Risk Model Overview Dashboard

Full screen | Share | Clone | Edit | g

project: Kubernetes - Golang Deps × | Add filter

Risk Model Overview Dashboard

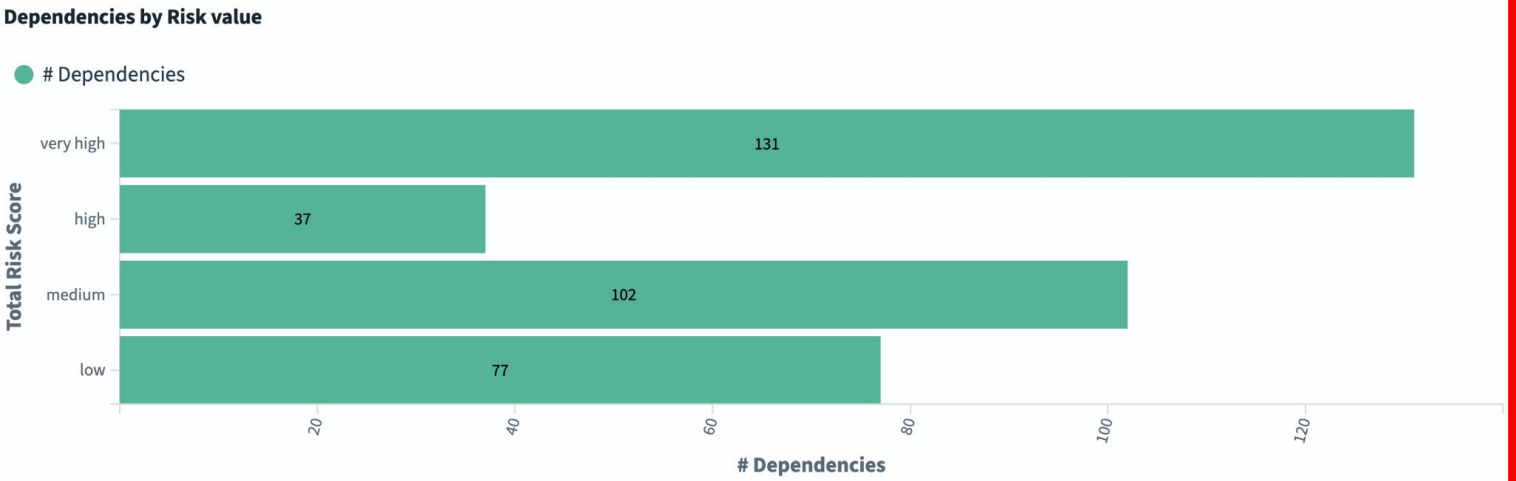
Check the Risk Model Help Dashboard for more information about this analysis.

For more details, pin a filter by origin and visit the Risk Model Dashboard for Individual Projects.

Filters

Overview

Dependencies by Risk value



	low	medium	high
REI	148	25	174
Retention Rate	81	42	224

Individual Metric Risk Value

# "Medium risk" metrics	# "High risk" metrics	Last analyzed on
0	7	Jun 27, 2024 @ 12:35
0	7	Jun 27, 2024 @ 12:35
1	6	Jun 27, 2024 @ 12:35
1	6	Jun 27, 2024 @ 12:35
1	6	Jun 27, 2024 @ 12:35
	6	Jun 27, 2024 @ 12:35
1	6	Jun 27, 2024 @ 12:35

https://github.com/davecgh/go-spew | Kubernetes - Golang Deps | very high | 9.29 | 0

Export: Raw | Formatted

1 2 3 4 5 ... 35

Bitergia Analytics

Dashboards

project: Kubernetes - Gola

Risk Model Overvi

Check the Risk Model He

For more deta s, pin a f

Filters

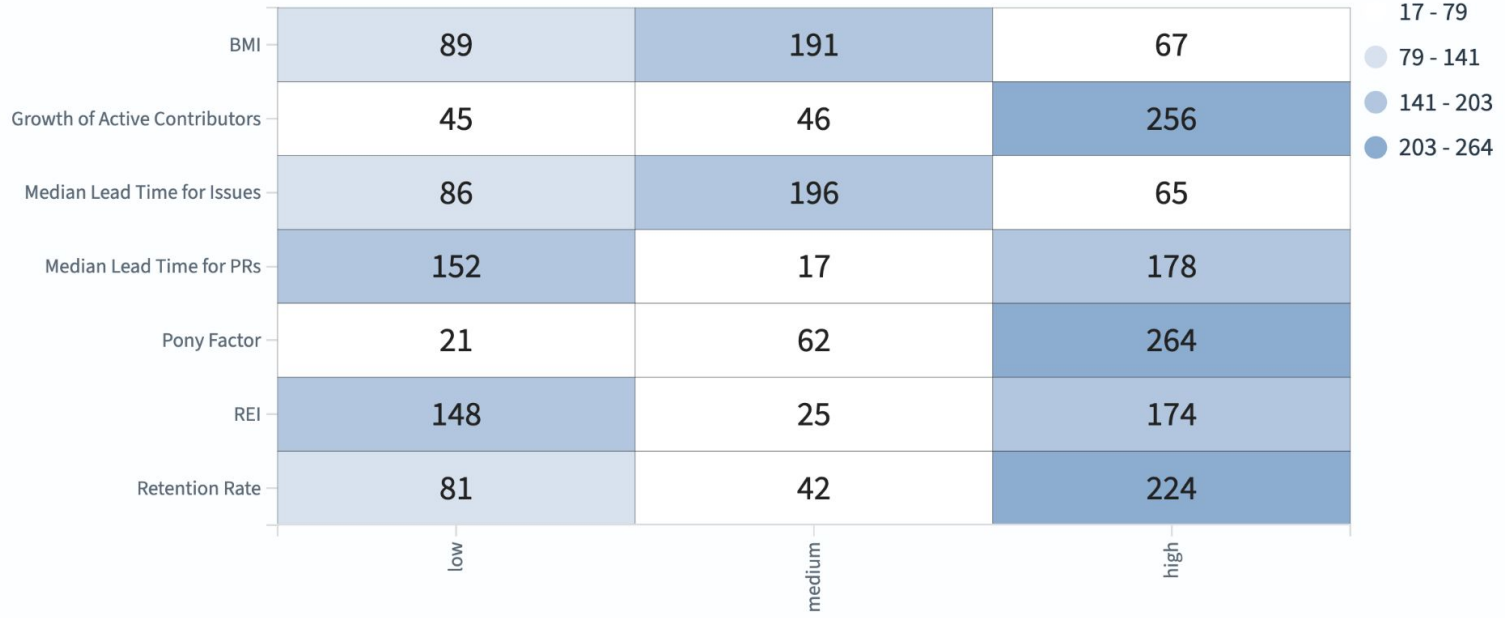
Team

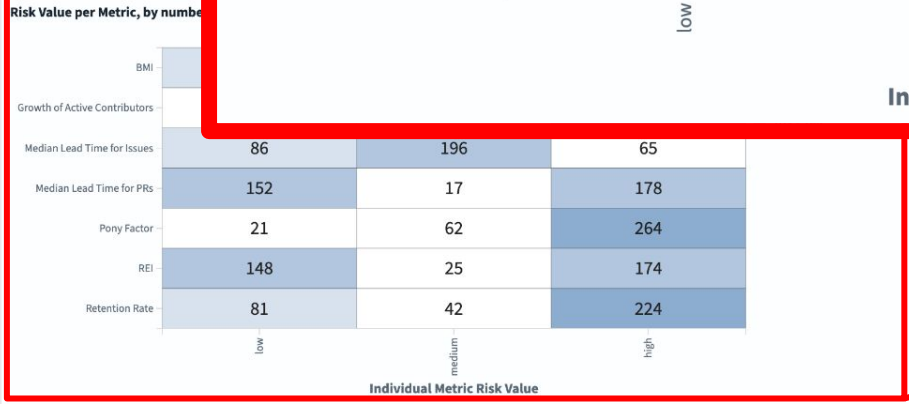
Select.

Project Category

Ku ernetes - Golang Deps ×

Risk Value per Metric, by number of Dependencies





	Deps						12:35
https://github.com/coreos/go-semver	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/curioswitch/go-reassign	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/davecgh/go-spew	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35

Export: Raw Formatted

1 2 3 4 5 ... 36

Risk Model Overview Dashboard

Check the Risk Model Help Dashboard for more information about this analysis.

For more details, pin a filter by origin and visit the Risk Model Dashboard for Individual Projects.

Filters

Team

Select...

Project Category

Kubernetes - Golang Deps ×

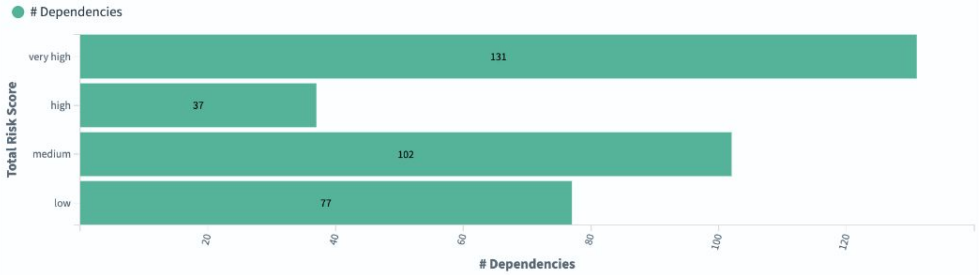
Overview

347

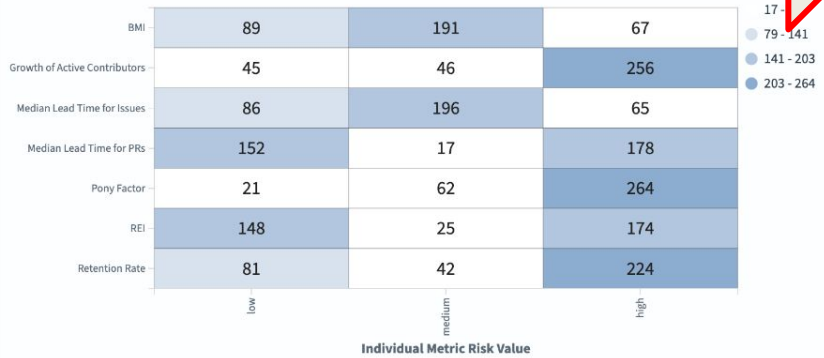
Dependencies analyze

Drill Down

Dependencies by Risk value



Risk Value per Metric, by number of Dependencies



Overall results by dependency repository

Filter...

Repository	Category	Risk Value	Risk Score (over 10)	# "Low risk" metrics	# "Medium risk" metrics	# "High risk" metrics	Last analyzed on
https://github.com/json-iterator/go	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/spf13/afero	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/Azure/go-ansiterm	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/ThalesIgnite/crypto11	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/coreos/go-semver	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/curioswitch/go-reassign	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/davecgh/go-spew	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35

Export: Raw Formatted

1 2 3 4 5 ... 37

Search DQL Last 5 years rounded to the year Show dates Refresh

project: Kubernetes - Golang Deps × origin: https://github.com/stretchr/testify × Add filter

Risk Model Dashboard for Individual Projects

Make sure to filter by a given repository before using this dashboard.

Check the Risk Model Help Dashboard for more information about this analysis.

Go back to the Risk Model Overview Dashboard.

Total Risk Score

github.com/stretchr/testify	medium	4.29	https://github.com/stretchr/t
Package Name	Total Risk	Total Risk Score (over 10)	Package Repository

Select an dependency first to check its risk

Team: Select...

Dependency Pkg Name: Select...

Dependency Repository: Select...

Project Category: Kubernetes - Golang Deps ×

Risk Model: Detailed view - dependency selector table

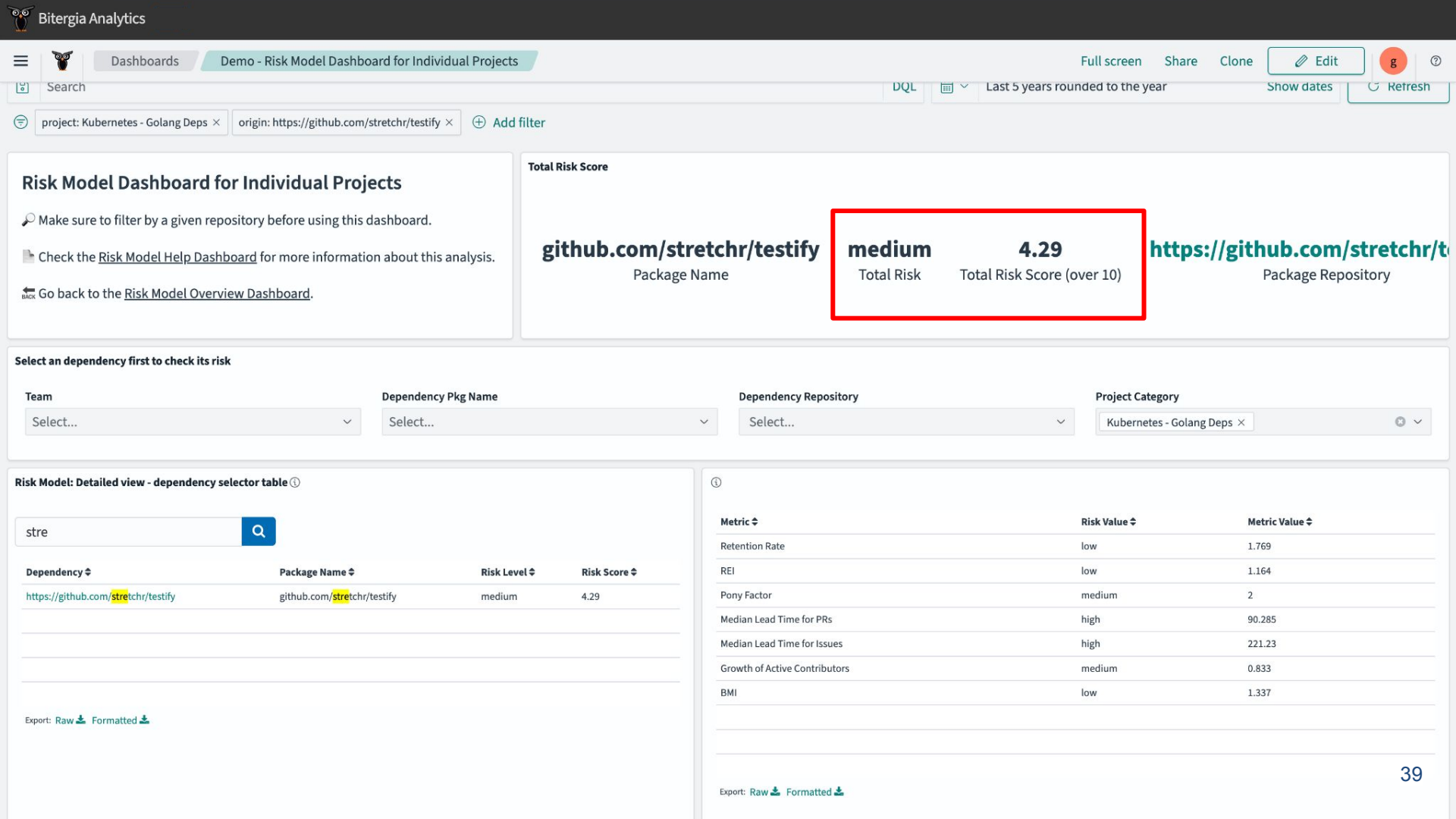
stre

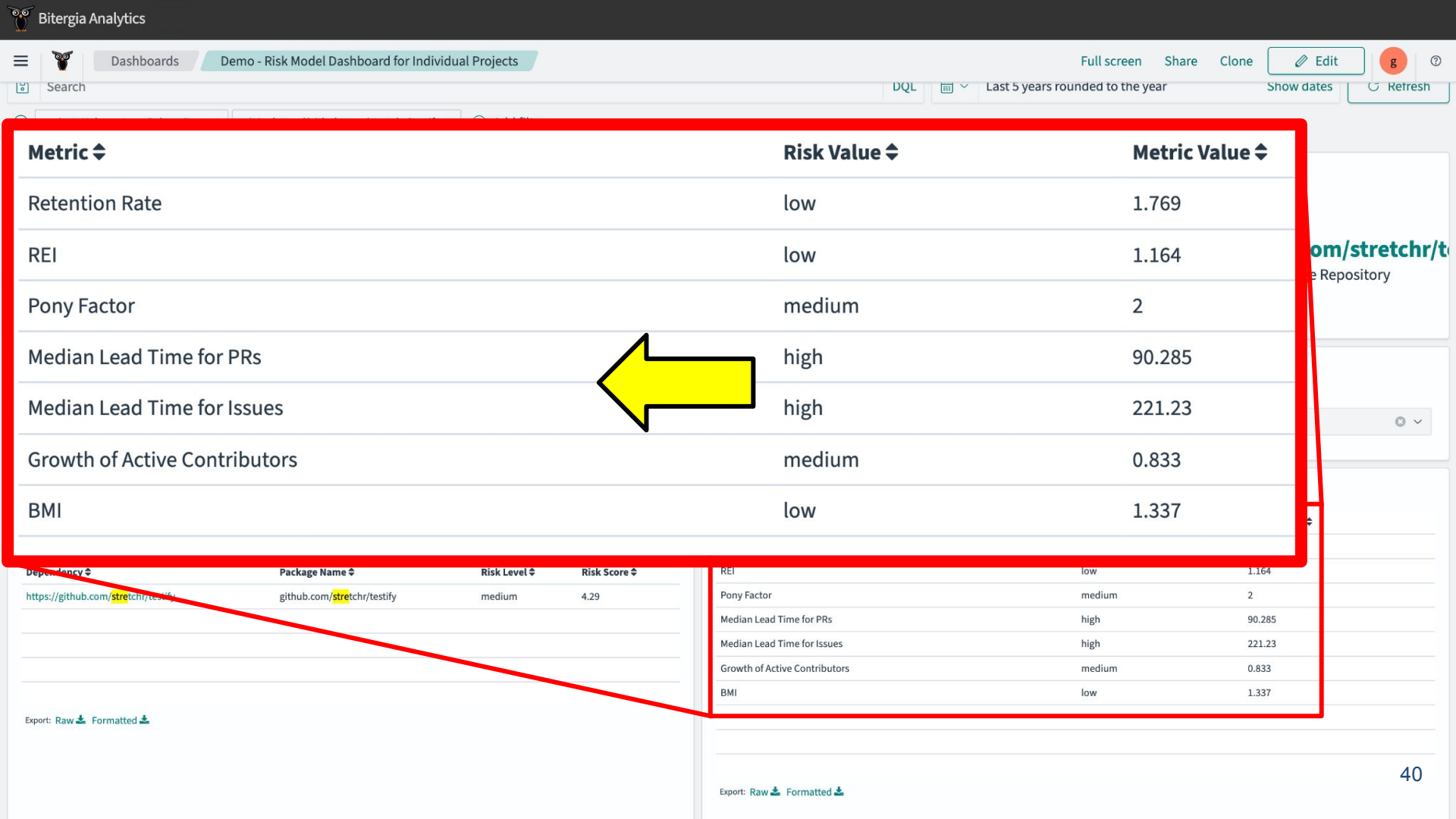
Dependency	Package Name	Risk Level	Risk Score
https://github.com/stretchr/testify	github.com/stretchr/testify	medium	4.29

Export: Raw Formatted

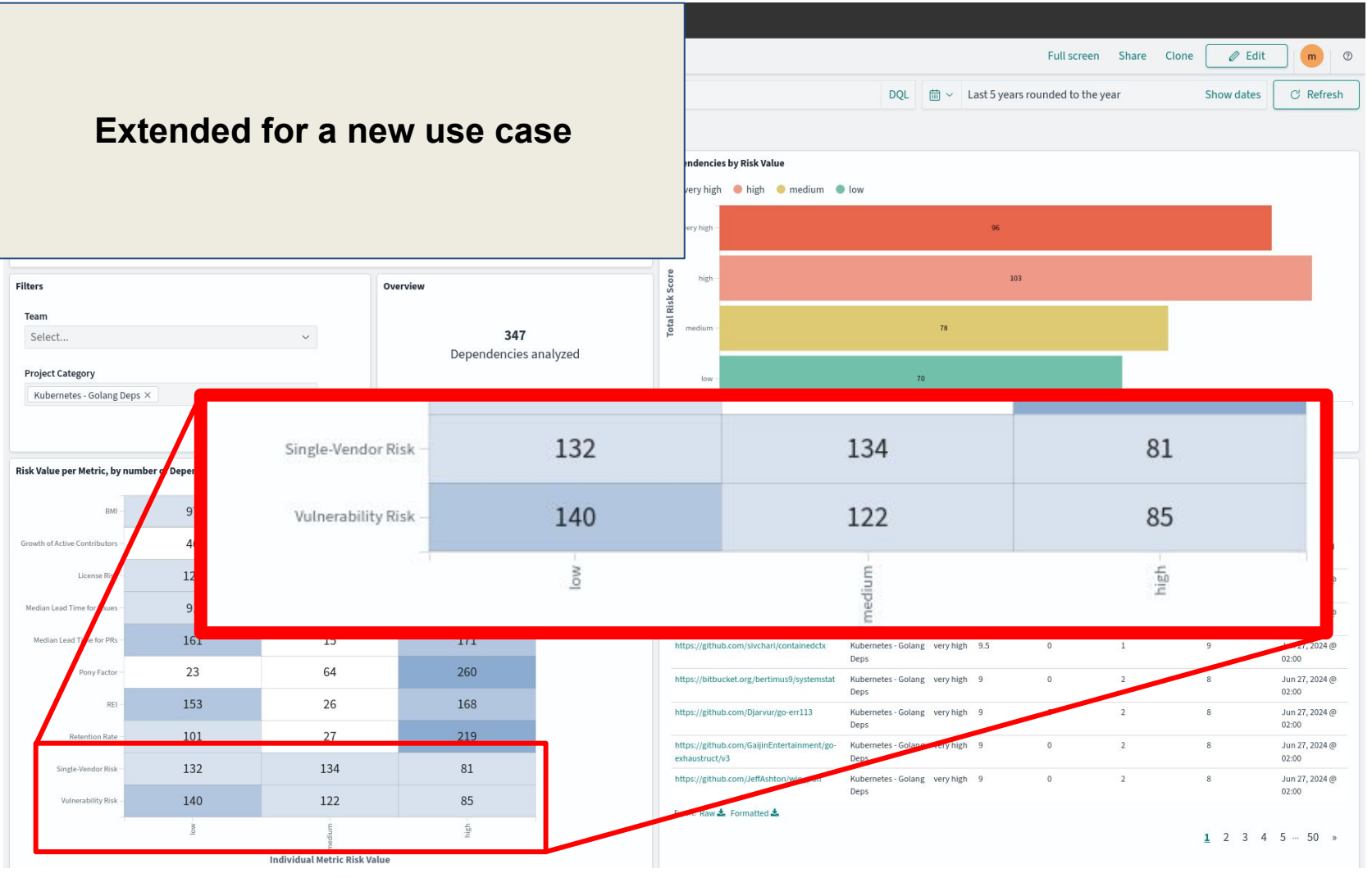
Metric	Risk Value	Metric Value
Retention Rate	low	1.769
REI	low	1.164
Pony Factor	medium	2
Median Lead Time for PRs	high	90.285
Median Lead Time for Issues	high	221.23
Growth of Active Contributors	medium	0.833
BMI	low	1.337

Export: Raw Formatted

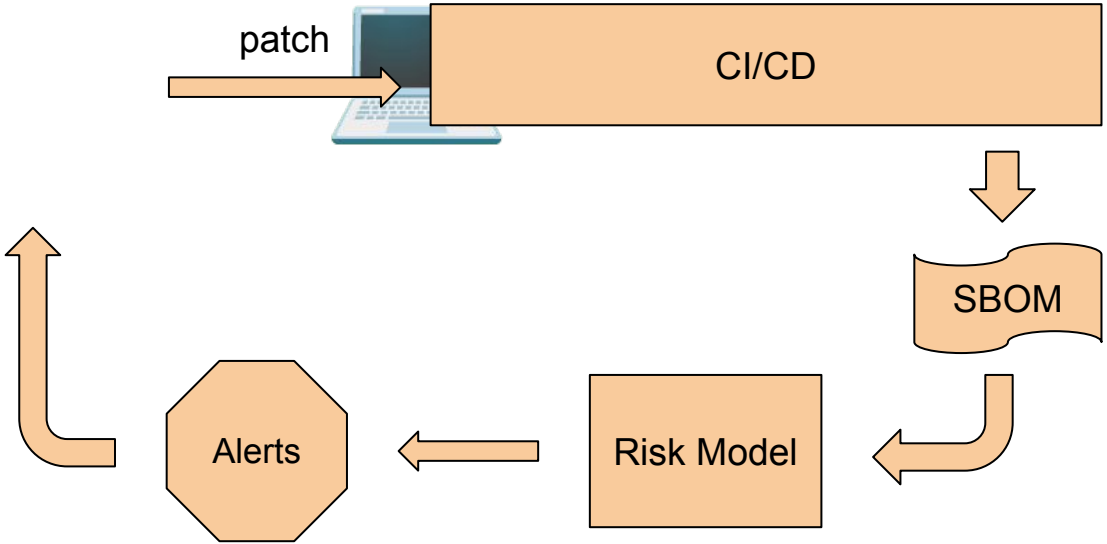

Bitergia Analytics
Dashboards
Demo - Risk Model Dashboard for Individual Projects
Full screen
Share
Clone
Edit
Search
DQL
Last 5 years rounded to the year
Show dates
Refresh
project: Kubernetes - Golang Deps
origin: https://github.com/stretchr/testify
Add filter
Risk Model Dashboard for Individual Projects
Make sure to filter by a given repository before using this dashboard.
Check the Risk Model Help Dashboard for more information about this analysis.
Go back to the Risk Model Overview Dashboard.
Total Risk Score
github.com/stretchr/testify
Package Name
medium
Total Risk
4.29
Total Risk Score (over 10)
https://github.com/stretchr/t
Package Repository
Select an dependency first to check its risk
Team
Select...
Dependency Pkg Name
Select...
Dependency Repository
Select...
Project Category
Kubernetes - Golang Deps
Risk Model: Detailed view - dependency selector table
stre
Dependency
Package Name
Risk Level
Risk Score
https://github.com/stretchr/testify
github.com/stretchr/testify
medium
4.29
Export: Raw Formatted
Metric
Risk Value
Metric Value
Retention Rate
low
1.769
REI
low
1.164
Pony Factor
medium
2
Median Lead Time for PRs
high
90.285
Median Lead Time for Issues
high
221.23
Growth of Active Contributors
medium
0.833
BMI
low
1.337
Export: Raw Formatted


Bitergia Analytics
Dashboards
Demo - Risk Model Dashboard for Individual Projects
Full screen
Share
Clone
Edit
Search
DQL
Last 5 years rounded to the year
Show dates
Refresh
Metric
Risk Value
Metric Value
Retention Rate
low
1.769
REI
low
1.164
Pony Factor
medium
2
Median Lead Time for PRs
high
90.285
Median Lead Time for Issues
high
221.23
Growth of Active Contributors
medium
0.833
BMI
low
1.337
Package Name
Risk Level
Risk Score
github.com/stretchr/testify
medium
4.29
Export: Raw Formatted
40

Extended for a new use case



Next iteration: CI/CD Integration




GrimoireLab:
The Open Source Tool

Story of GrimoireLab

- 2004 LibreSoft @ University Rey Juan Carlos in Spain
- 2012 Bitergia offers commercial services with Metrics Grimoire
- 2016 GrimoireLab starts, using ElasticSearch for Dashboarding
- 2017 Founding of CHAOSS
- 2024 version 1.0 released

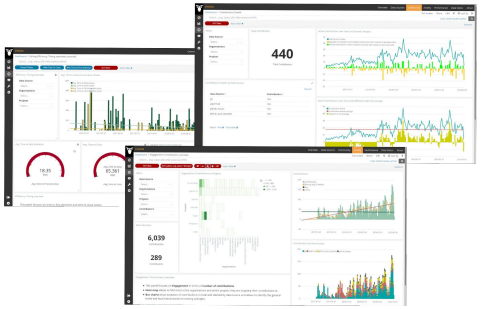
Example: Mozilla Foundation

“[...] holistic view of our contributor ecosystem’s network structure, health and impact [...]”

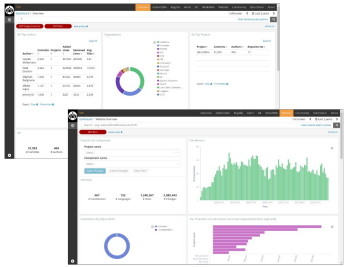
“[...] we’re able to visually describe these distinct contributor communities as well as how they are interconnected [...]”

https://report.mozilla.community/

Platforms built with GrimoireLab

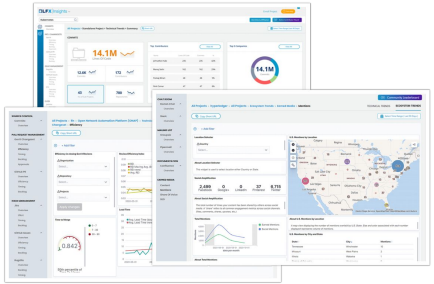


Bitergia Analytics

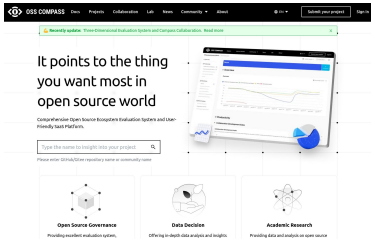


The Document Foundation

Mystic



Linux Foundation Insights



OSS Compass

Collecting data from OSS communities

Data Collection

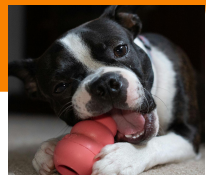
Digital footprints from data source

(biased towards activities that are logged)

Enrichment

Translate data into information

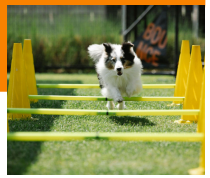
(connect and unify for consistency)

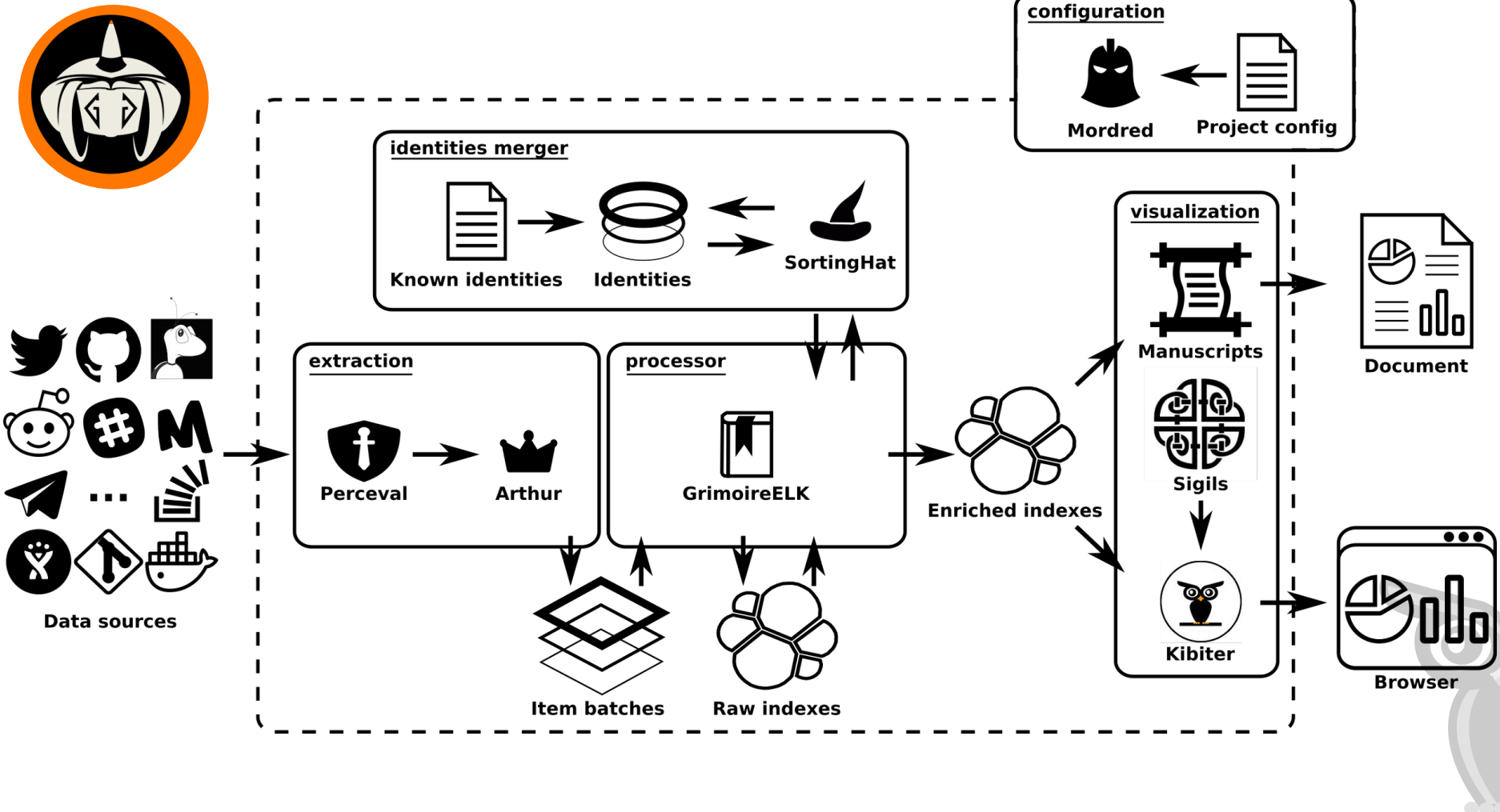


Visualization and Reporting

Gain insights and decide actions

(tell stories and convince)




configuration
Mordred
Project config
identities merger
Known identities
Identities
SortingHat
visualization
Manuscripts
Document
Sigils
extraction
Perceval
Arthur
processor
GrimoireELK
Enriched indexes
Kibiter
Browser
Data sources
Item batches
Raw indexes

SortingHat to disambiguate contributors

08/2008 - 07/2011 Bankhaus C. L. Seeliger
10/2011 - 05/2015 TU Braunschweig
08/2015 - 05/2019 University of Nebraska at Omaha
05/2019 - now Bitergia

git

Georg J.P. Link <linkgeorg@gmail.com>
Georg Link <linkgeorg@gmail.com>
Link, Georg <glink@unomaha.edu>
Georg Link <georglink@bitergia.com>

GeorgLink

linkgeorg@gmail.com
glink@unomaha.edu
georglink@bitergia.com

PHABRICATOR

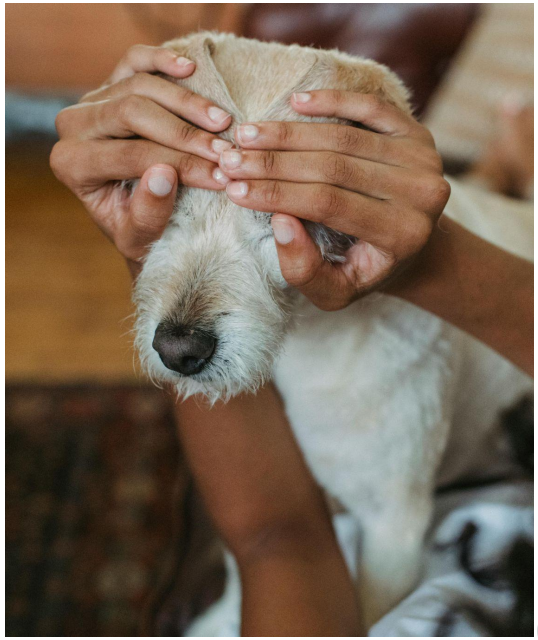
georglink@bitergia.com

slack

GeorgLink

Excursion: Minding data privacy

- GDPR is gold standard
- Opt-in vs. Opt-out
- Enriching data from data sources
- Offering a “remove my data” feature



GrimoireLab 2.0 roadmap

- **Maintenance effort:** ⬇️
  Graphical user interface and an API for configuring data collection
- **Scalability and performance:** ⬆️
  Currently, 3,500 high-active repositories require three days of data analysis before the data is ready for the user
- **Integration with other tools:** 🔀
  Support more tools for visualizing and analyzing the data

How to Get Started?

Open Source: GrimoireLab tutorial

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

Commercial Support: Bitergia Risk Radar

- https://bitergia.com/risk-radar/

Shining Light on the Open Source Supply Chain: The Risk in Community Health

Thank you and please reach out!

SCaLE 22x
March 8, 2025
Pasadena, CA, USA

Georg Link, PhD
georglink@bitergia.com