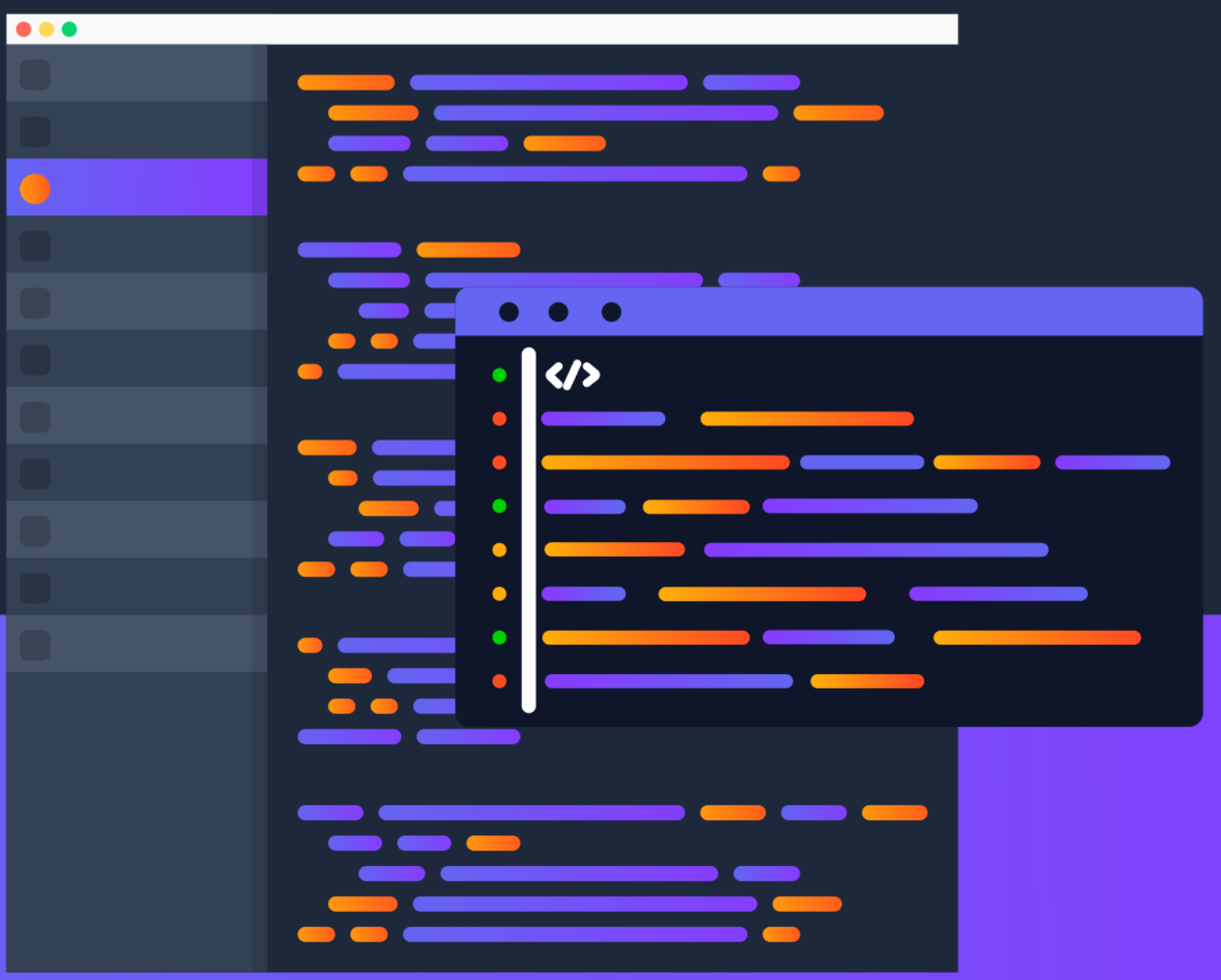
Whitepaper

Standardising Open Source Inventorying

The first **Open Source Inventory Engine** built for developers & modern DevSecOps teams.



SCANOSS

Content

Questions?

Please get in touch through www.scanoss.com

Background

Software Composition Analysis (SCA) tools are widely used to identify and manage open source components within codebases. Yet, despite their significant role in enabling OSS compliance, most SCA vendors have resisted the very principles they claim to support, openness and transparency. Instead, they offer closed, proprietary systems that create friction for developers and legal teams alike. These tools are difficult to integrate, incompatible with modern CI/CD workflows, and costly to maintain.

In an environment where software is built and deployed continuously, closed systems are no longer sustainable. What is needed is a shift: away from black-box auditing tools and towards developer-native, fully open infrastructure for continuous OSS inventorying.

Redefining Software Composition Analysis

Traditional SCA tools played a critical role in the early days of open source adoption, particularly in generating SBOMs for legal due diligence, end-of-cycle audits, and M&A processes. These tools helped establish baseline practices for open source compliance. However, their architecture was rooted in static development models.

Today, software is developed continuously and collaboratively. This shift exposed the limitations of legacy SCA tools. SCANOSS reimagines SCA from first principles. It introduces a platform that is fully open, standards-aligned, and purpose-built for continuous, developer-centric OSS compliance. Instead of relying on closed tools that hide how code is identified and matched, SCANOSS provides full transparency.

At the heart of SCANOSS is the principle of ”always-on” compliance. The platform allows teams to validate and inventory OSS components in real time, without needing to pause development for manual audits. It is designed to work naturally within modern DevSecOps pipelines, enabling proactive compliance rather than reactive correction.

Open-by-Design Platform

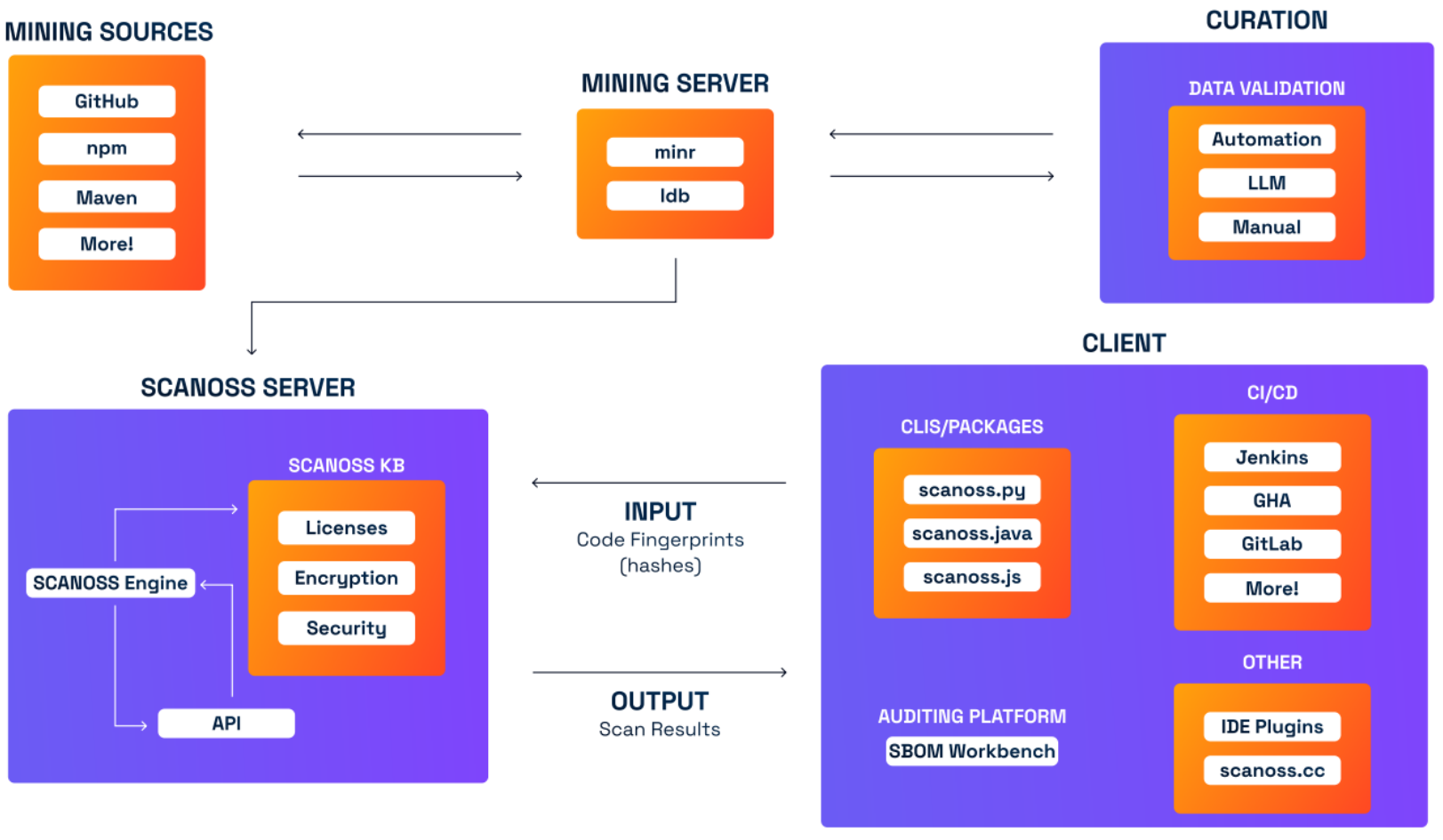
Every component of the SCANOSS platform is open source and modular, allowing organisations to deploy, customise, and extend their OSS governance strategy with confidence:

- Open Data Mining Tool ('minr'): Extracts, fingerprints and indexes OSS and proprietary source code.
- Open Database Engine ('ldb'): Purpose-built for high-throughput fingerprint matching at scale.
- Open Inventory Engine: The first fully open source inventory engine built for developers.
- Open RESTful API: OpenAPI-compliant interface for integration with CI/CD tools.
- Open Webhook: Enables automated scans on Git events like push or pull request.
- Open CLI Scanner ('scanner.py'): A Python-based command line tool for local and scripted validation.

Architecture

The backbone of SCANOSS is its OpenAPI-compliant RESTful API, which provides a rich set of endpoints for real-time source code inventorying, SBOM management, project provisioning, user management, and audit control. CI/CD systems and custom tools can interact with the inventory engine seamlessly through HTTPS, using simple JSON-based workflows.

The Inventory Engine operates at low latency by querying the underlying ldb database, which is optimised specifically for fingerprint lookups. SBOMs are generated using SPDX and CycloneDX standards (JSON and XML formats supported), ensuring compatibility with regulatory frameworks and third-party systems.



Transparent Fingerprinting with Winnowing

One of the major pain points in the current SCA landscape is the lack of standardisation in snippet detection. Most SCA vendors rely on proprietary fingerprinting techniques, which makes it nearly impossible to compare results across tools and raises concerns around transparency and vendor lock-in. In many cases, these tools require sensitive source code to be processed through closed binaries, creating avoidable security and compliance risks.

SCANOSS addresses this challenge by adopting the academically recognised Winnowing algorithm for snippet fingerprinting—a method long used in plagiarism detection and source similarity analysis. This approach is reproducible, well-understood, and available through multiple open source implementations. SCANOSS has adapted Winnowing for indexing and comparing massive volumes of source code while preserving transparency, auditability, and developer trust.

https://github.com/scanoss/wfp

The SCANOSS Database Engine: ldb

Traditional SQL and NoSQL databases struggle with the scale and speed needed for SCA tasks, where each file scan may involve thousands of quick, repetitive queries. Their complexity creates performance bottlenecks and unnecessary overhead.

To address this, SCANOSS developed `ldb`, a custom-built data engine optimised exclusively for OSS fingerprint matching. The engine uses a mapped linked-list structure with numeric keys, drastically reducing lookup times and keeping the memory footprint minimal. It is capable of distributing its data model across machines and has already indexed over 3 trillion fingerprints, enabling microsecond-level response times for even the largest codebases.

- Mapped linked-list architecture
- Over 3 trillion fingerprints indexed
- Microsecond-level response times
- Distributed and memory-efficient

https://github.com/scanoss/ldb

Inventory Engine and Continuous SBOM

The Inventory Engine compares source code (or Winnowing fingerprints) against the SCANOSS Knowledge Base. It produces JSON-formatted outputs that include metadata such as matched components, licences, snippet hashes, and URLs.

Optimised for low-latency processing, the engine can run locally or as part of a shared infrastructure. It is designed for real-time operation, enabling SBOMs to be generated incrementally and continuously throughout the development lifecycle.

https://github.com/scanoss/engine

Automating with the RESTful API and Webhooks

To support high-velocity environments, SCANOSS provides an open RESTful API and a pre-built Webhook server. These components enable automated inventory checks triggered by Git operations such as `push` or `pull` request.

When code is pushed to the repository, the SCANOSS Webhook retrieves modified files and optionally validates them against declared open source assets. The results are posted back to the repository as build status updates and comment badges. If undeclared OSS components or code snippets are found, the commit is flagged as failed. This tight integration turns OSS compliance into a lightweight and automated process, rather than a bottleneck at release time.

https://github.com/scanoss/API
https://github.com/scanoss/webhook

Developer Tooling: scanner.py CLI

For developers preferring on-demand validation or scripting workflows, SCANOSS offers `scanner.py`, a lightweight Python CLI tool. It recursively scans a directory, generates fingerprints, and returns detailed component metadata. Ideal for use in:

- Local development environments
- Git hooks and pre-merge checks
- Container build pipelines

https://github.com/scanoss/scanoss.py

Custom Knowledge Bases with minr

Beyond consuming the SCANOSS Knowledge Base, organisations can build their own using `minr`. This CLI tool allows users to download source code, extract metadata, and generate fingerprint indexes for custom OSS or proprietary components. These custom knowledge bases can be used to:

- Detect internal IP leakage
- Validate against curated lists of OSS
- Conduct comparative research on licensing risks

Minr operates in parallel across machines and instances and its data output is portable and scalable across devices.

https://github.com/scanoss/minr

Open, Transparent, Continuous OSS Governance

SCANOSS marks a fundamental shift in how OSS compliance is approached. By replacing opaque, proprietary systems with open, developer-centric tools, SCANOSS empowers organisations to manage software supply chain risks proactively, efficiently, and transparently.

Whether you're a DevOps engineer, a security lead, or a legal advisor, SCANOSS offers an adaptable toolkit that fits into your existing workflows. It is open, auditable, scalable and ready for the realities of modern software development.

https://github.com/scanoss

Get involved

The SCANOSS Platform is made entirely available as open source. The collaboration guidelines are available in the source code tree. Questions and suggestions are welcome at https://www.scanoss.com

Get in touch

SCANOSS offers commercial agreements with access to its complete Knowledge Base, additional features and Service Level Agreements.

Please contact info@scanoss.com for further information.

Published on 2025-06-25 by SCANOSS.com

SCANOSS