



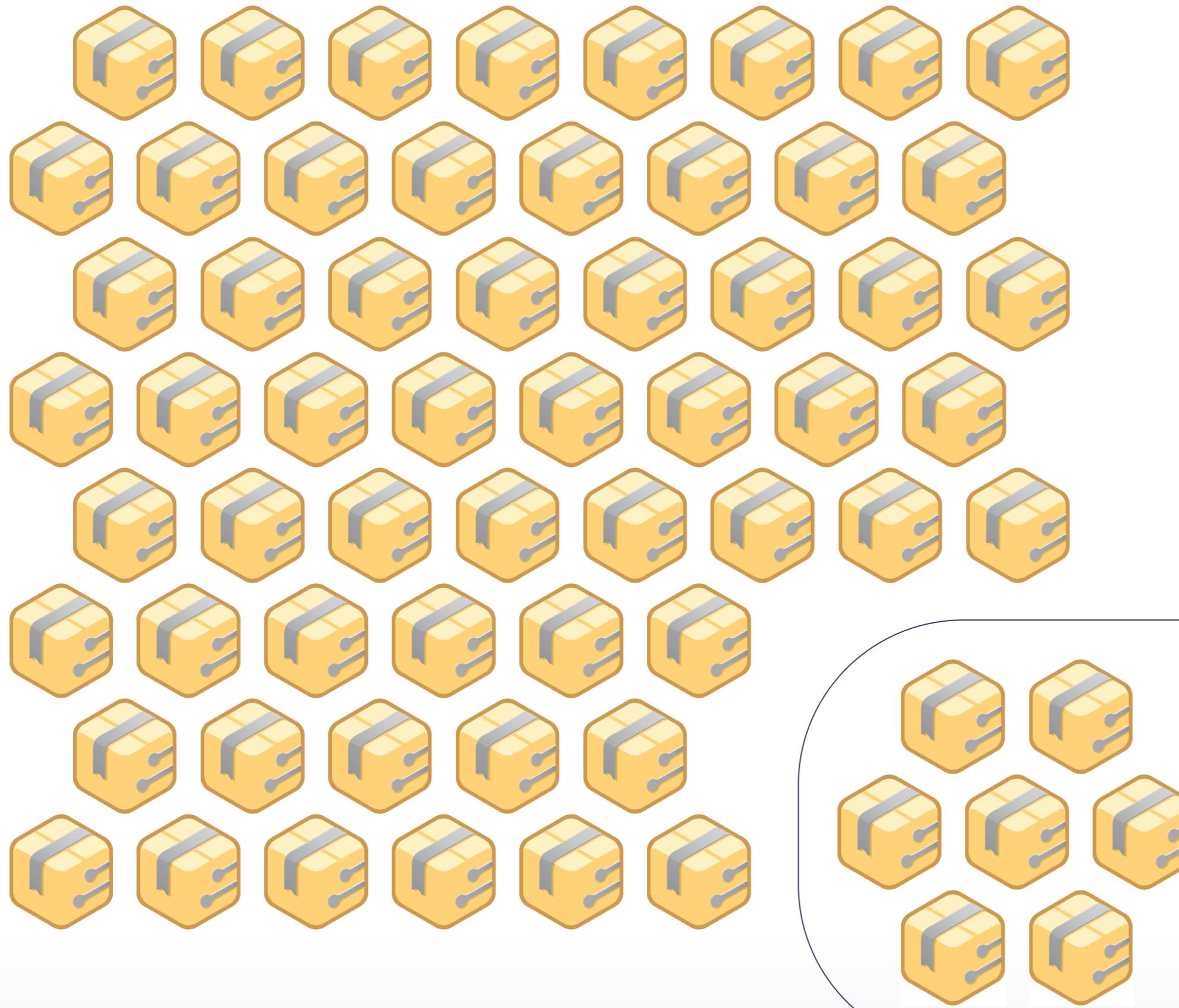
**FOSSID**

APRIL 2023

# **Types of SCA**

---

# Types of SCA Security VS License Compliance



## License Compliance Risks

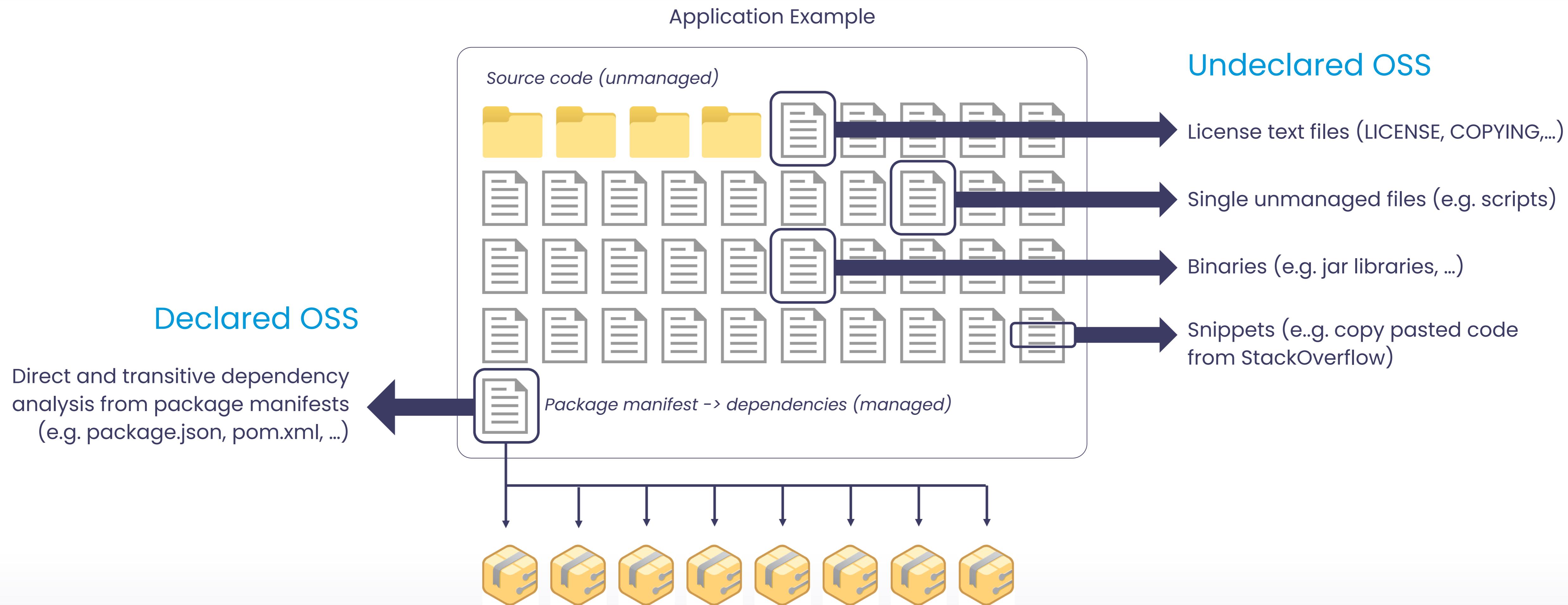
There are over 150M components of software in FossID's Knowledge Base today.

These software components are released under 2000+ different open source licenses.

## Security Risks

Only about 1M components have known vulnerabilities and exposures (CVEs)

# Types of SCA Declared VS Undeclared OSS



# Open Source is everywhere As is the need to properly manage it

**97%**

of codebases  
leverage Free Open  
Source Software

**81%**

of codebases have  
security vulnerability  
issues

**53%**

of codebases have  
license compliance  
issues

# FossID Key Strength – Detection Capabilities

With our tools, you should find more (compared to competition)

- Ability to **identify more** software
- Ability to identify in **higher level of detail**
- Ability to identify with **less false positives**

# PRODUCTS

---

# Workbench Platform

One interface to manage all 3<sup>rd</sup> party sw:

- Review scan results:
  - See OSS component, file & snippet matches
  - Review license and copyright information
- Generate reports
  - SBOM (SPDX),
  - Notice files, etc...
- Policy management
  - Implement and enforce company wide or project based OSS policies
  - Approve/reject OSS based on policies
  - Follow up through JIRA
- Catalog all 3<sup>rd</sup> party software used in your products
- Create customized user roles based on privileges

The screenshot displays the FOSSID Workbench interface. At the top, there's a navigation bar with links for 'FOSSID Workbench', 'Dashboard', 'Projects', 'Scans', 'Components', 'Licenses', 'Users', and 'System Utils'. To the right of the navigation are icons for settings, notifications, help, and user profile.

The main area has a heading 'Uncover license compliance' and a sub-instruction 'Scan your projects to get started'. It features three main sections: 'Scan a repository' (with bullet points: Find license declarations even inside source code, Collect all copyright statements, Detect snippets of open source), 'Add a project' (with bullet points: Group scans together, Licence governance at scale), and 'Command Line Interface' (containing the command 'fossid --host myserver --token mytoken fossid scan' and a link to 'Full documentation for Snyk Workbench CLI').

Below this is another section of the interface showing a search bar ('Search file, folder...') and a tree view of a project structure: 'project / folder / file.java'. To the right is a table of components with columns: NAME, VERSION, LICENSE, UPDATED, and STATUS. The table lists four components: Component 1 (version 1.2, GPL-2.0, updated 2022-05-05, pending), Component 2 (N/A, MPL, 2022-05-05, pending), Component 3 (3.4.5, Apache-2.0, 2022-05-05, pending), and Component 4 (2.0-beta, MIT, 2022-05-05, pending). Below the table are two code snippets: 'file.java' and 'other\_file.java', each showing a detailed copyright notice.

# Knowledge Base OSS Intelligence

## Comprehensive Open-Source Software (OSS) coverage

FossID products and services are powered by FossID's industry-leading OSS intelligence database.



### Best Coverage

Maintained by a [dedicated research team](#), it covers over 150M OSS components coming from 60+ public sources and user contribution sites such as GitHub and StackOverflow.



### Maximum Accuracy

FossID's scanning technology detects from whole OSS components and binaries to small [snippets of code](#) (as little as 6 lines of code).



### Highest Confidentiality

FossID never accesses or transfers your source code, scans cryptographic hashes instead. Alternatively, FossID allows for fully [on-prem air-gapped deployments](#).

# Knowledge Base Detection Capabilities



Quickly identify  
folders, libraries,  
archives or binaries

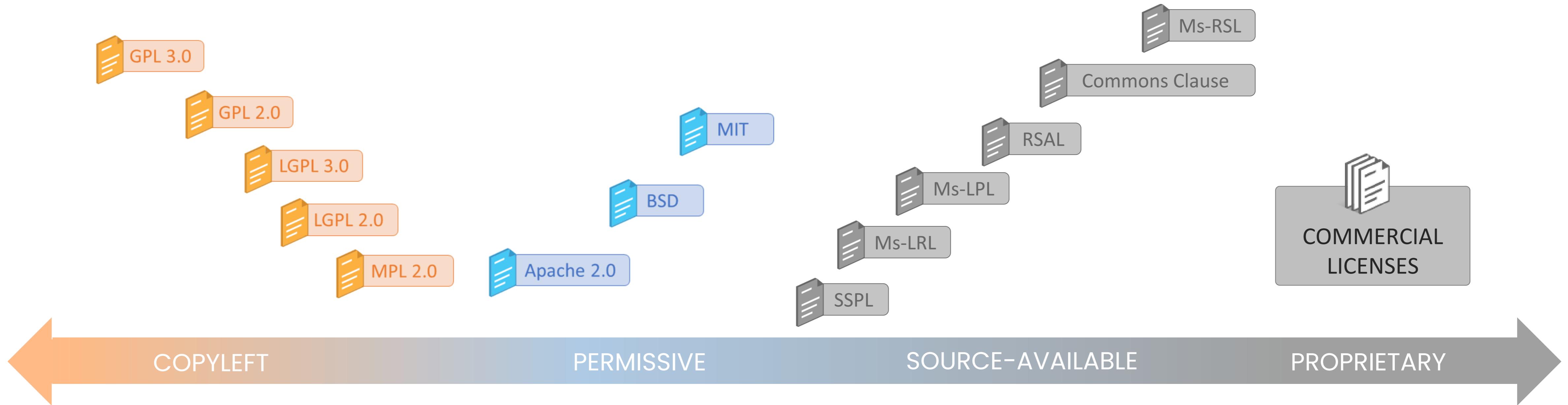


Detect full files in your  
code base even if they  
are modified



Identify smaller open  
source footprints like  
copy pasted code

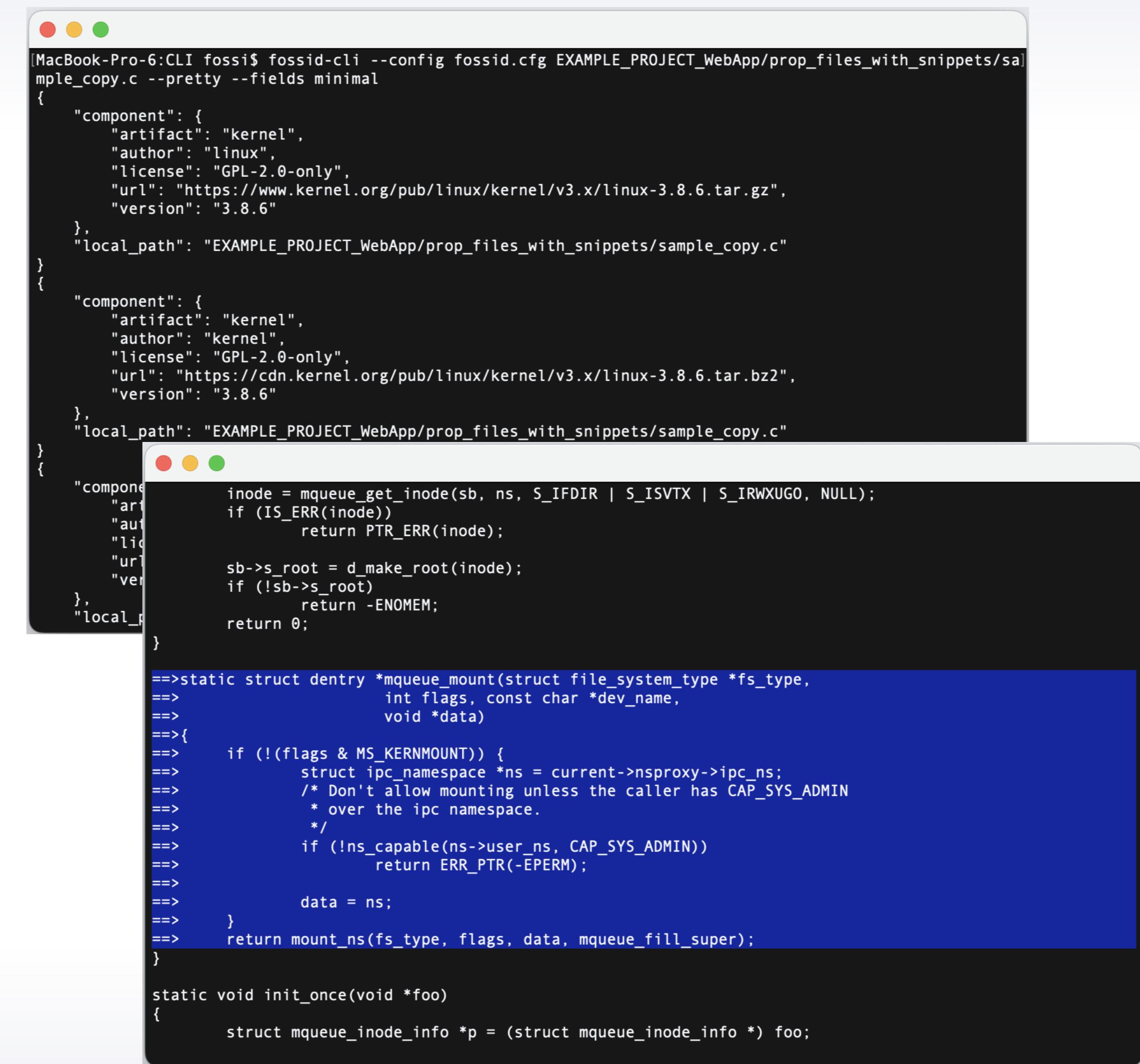
# Types of SCA Source-available licenses



# Automation Package

Integrate FossID Workbench in your SDLC:

- Command Line Interface
  - BareMetal scanning functionality
  - Linux, Windows and Mac OS
  - Can be used as stand alone  
(independently from Workbench)
- API
  - Access all Workbench functionality
  - Upload target code, trigger scans, generate reports, etc...



The image shows a Mac OS X terminal window with two visible panes. The top pane displays a JSON configuration file for the FossID CLI, specifying a component (kernel) with details like artifact, author, license, URL, and version. The bottom pane shows a portion of a kernel source code file, specifically the `mqueue.c` file, containing functions for mounting and initializing mqueue filesystems.

```
[MacBook-Pro-6:CLI fossi$ fossid-cli --config fossid.cfg EXAMPLE_PROJECT_WebApp/prop_files_with_snippets/sample_copy.c --pretty --fields minimal
{
  "component": {
    "artifact": "kernel",
    "author": "linux",
    "license": "GPL-2.0-only",
    "url": "https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.8.6.tar.gz",
    "version": "3.8.6"
  },
  "local_path": "EXAMPLE_PROJECT_WebApp/prop_files_with_snippets/sample_copy.c"
}

{
  "component": {
    "artifact": "kernel",
    "author": "kernel",
    "license": "GPL-2.0-only",
    "url": "https://cdn.kernel.org/pub/linux/kernel/v3.x/linux-3.8.6.tar.bz2",
    "version": "3.8.6"
  },
  "local_path": "EXAMPLE_PROJECT_WebApp/prop_files_with_snippets/sample_copy.c"
}

"component": {
  "artifact": "kernel",
  "author": "kernel",
  "license": "GPL-2.0-only",
  "url": "https://cdn.kernel.org/pub/linux/kernel/v3.x/linux-3.8.6.tar.bz2",
  "version": "3.8.6"
},
"local_path": "EXAMPLE_PROJECT_WebApp/prop_files_with_snippets/sample_copy.c"

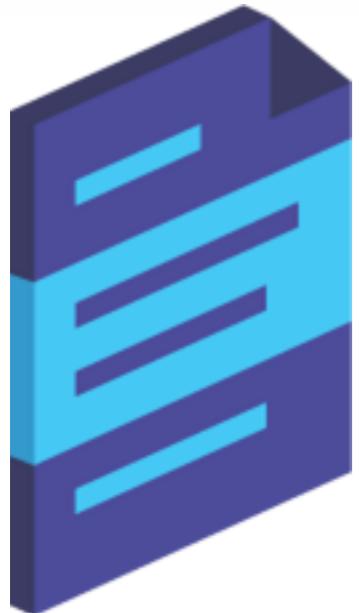
inode = mqueue_get_inode(sb, ns, S_IFDIR | S_ISVTX | S_IRWXUGO, NULL);
if (IS_ERR(inode))
    return PTR_ERR(inode);

sb->s_root = d_make_root(inode);
if (!sb->s_root)
    return -ENOMEM;
return 0;

==>static struct dentry *mqueue_mount(struct file_system_type *fs_type,
==>                                         int flags, const char *dev_name,
==>                                         void *data)
==>{
==>    if (!(flags & MS_KERNMOUNT)) {
==>        struct ipc_namespace *ns = current->nsproxy->ipc_ns;
==>        /* Don't allow mounting unless the caller has CAP_SYS_ADMIN
==>         * over the ipc namespace.
==>         */
==>        if (!ns_capable(ns->user_ns, CAP_SYS_ADMIN))
==>            return ERR_PTR(-EPERM);
==>
==>        data = ns;
==>    }
==>    return mount_ns(fs_type, flags, data, mqueue_fill_super);
}

static void init_once(void *foo)
{
    struct mqueue_inode_info *p = (struct mqueue_inode_info *) foo;
```

# VulnSnippet Finder Package



## Snippet detection for vulnerable OSS snippets

FossID's Knowledge Base snippet detection capabilities have been extended to include special detection of vulnerable OSS snippets.



### Extensive Coverage

Over 200k vulnerable snippets curated by experts.



### Risks in Undeclared OSS

Knowing the exact lines of code that introduce vulnerabilities is crucial when dealing with undeclared OSS.

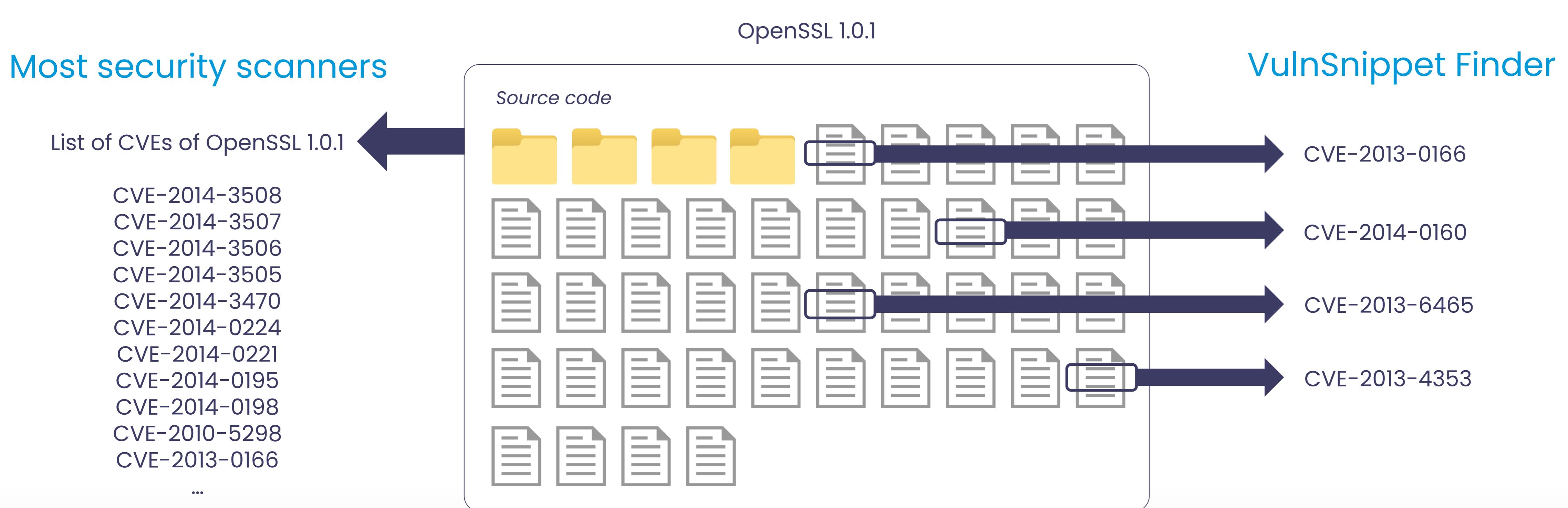


### Designed for automation

Only trigger build fails for vulnerabilities you know you have in your source code.

# VulnSnippet Finder Package

Most security scanners assume OSS vulnerabilities based on component/version  
while VulnSnippet Finder searches for the exact lines of code/snippets.



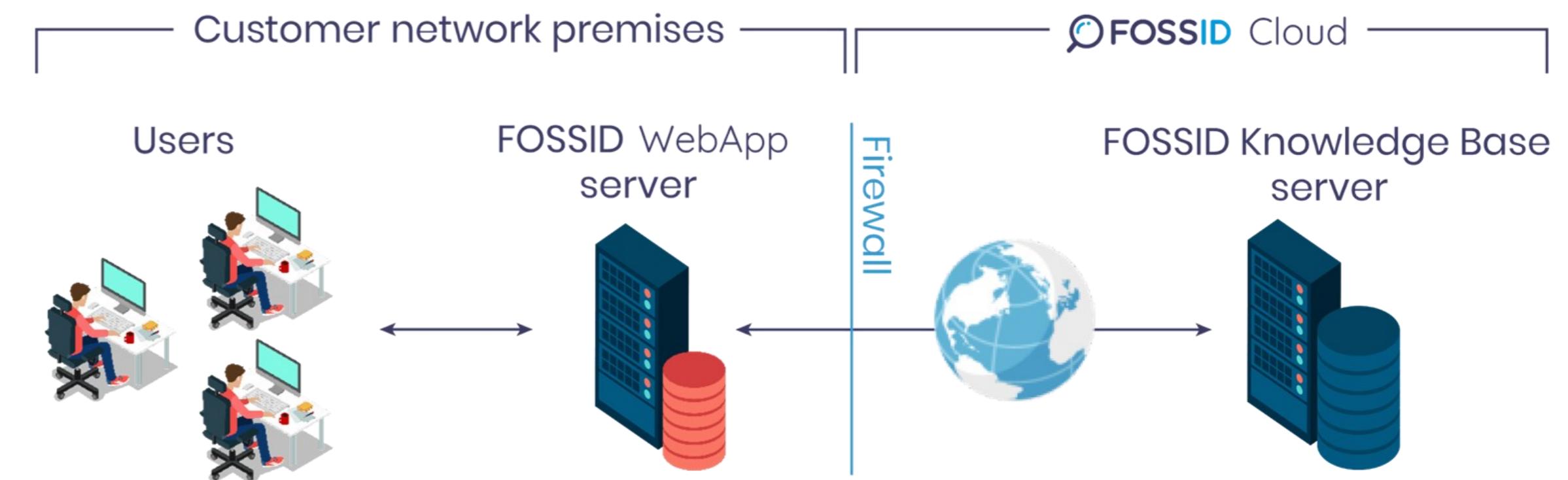
# DEPLOYMENT OPTIONS

---

# Deployment Options Hybrid & On-prem

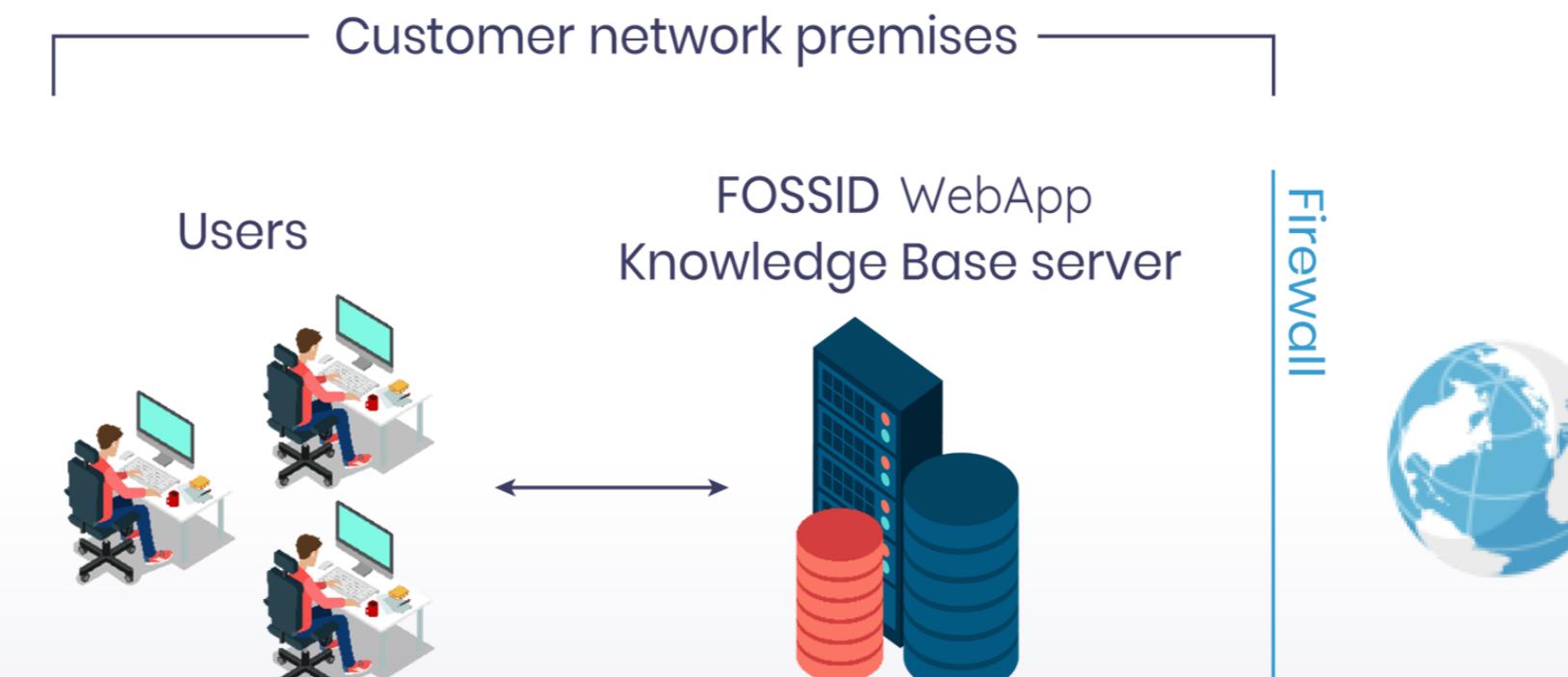
## Hybrid Deployment

- Knowledge Base in the cloud
- No source code is ever transmitted  
*Only digital signatures of source code are used to query the FOSSID Knowledge Base.*
- Continuous updates



## On-prem Deployment

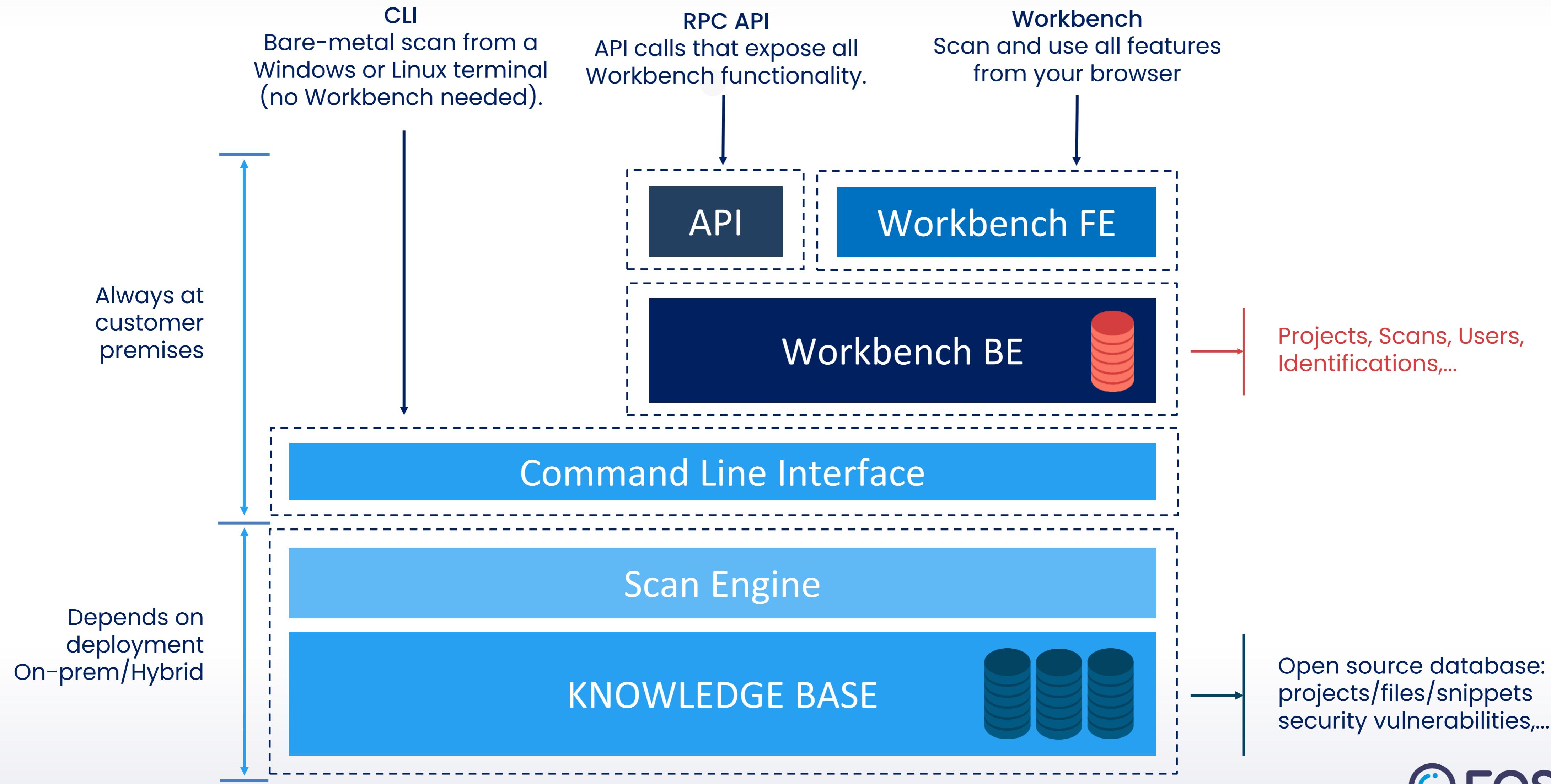
- Knowledge Base locally deployed
- No external network traffic involved  
*Performing scans does not involve any network traffic outside your network premises.*
- Monthly and weekly updates



# ARCHITECTURE

---

# FossID Technology Architectural View



# Thanks

---





# snyk

Develop fast.  
Stay secure.



+

+

+

+

+

+

+

+

# Developer Security Platform



Snyk  
Code

Snyk  
Open Source

Snyk  
Container

Snyk  
IaC

Snyk  
Cloud



## Developer Experience



Snyk  
Application intelligence



Snyk Learn  
Security Education



Snyk Intel  
Security intelligence

Empowerment

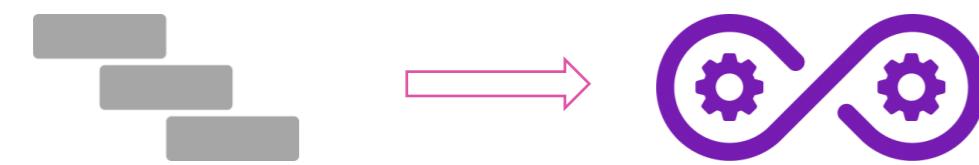
Extensibility

Governance

snyk

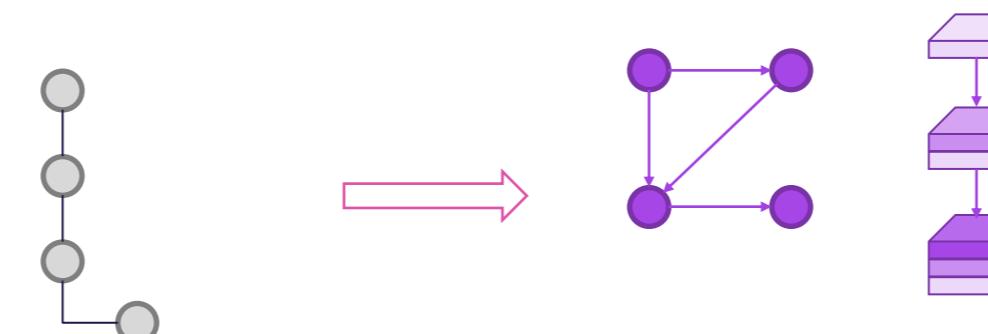
## Development Has Changed

### More development, faster



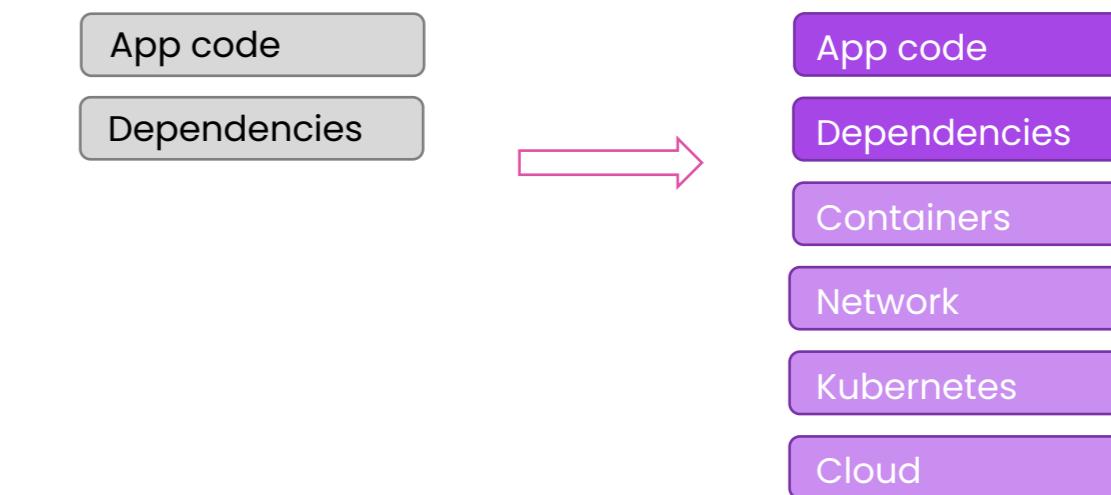
Need developers who understand and love to create secure applications

### Software supply chain more complex



Require guidance and visibility into the entire software supply chain

### The cloud is part of the code



Need to understand full scope of apps from code to cloud, and back to code

# A Modern Approach to Security is Required

## Traditional App Sec

Testing after development



Audit Based



Code and Infrastructure  
Secured Independently



## Dev-First

Continuous Testing



Fix Based

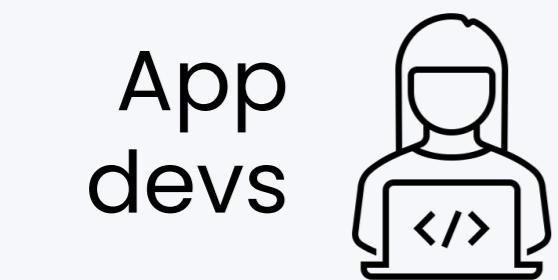


Holistic Cloud Native  
Application Context



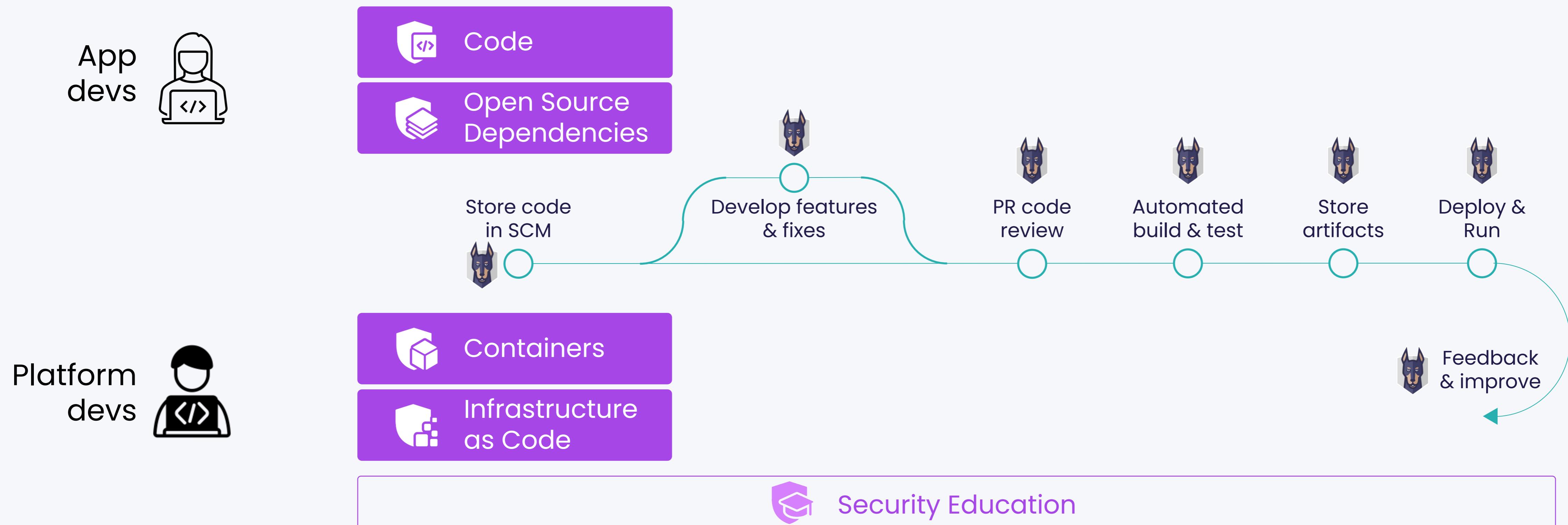
**Snyk empowers secure developers**

**All your  
developers**

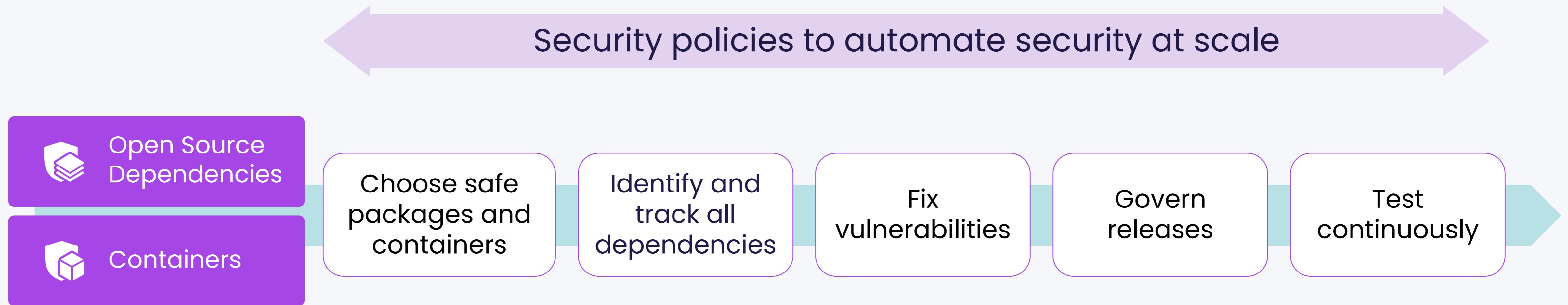


**Throughout  
their code**

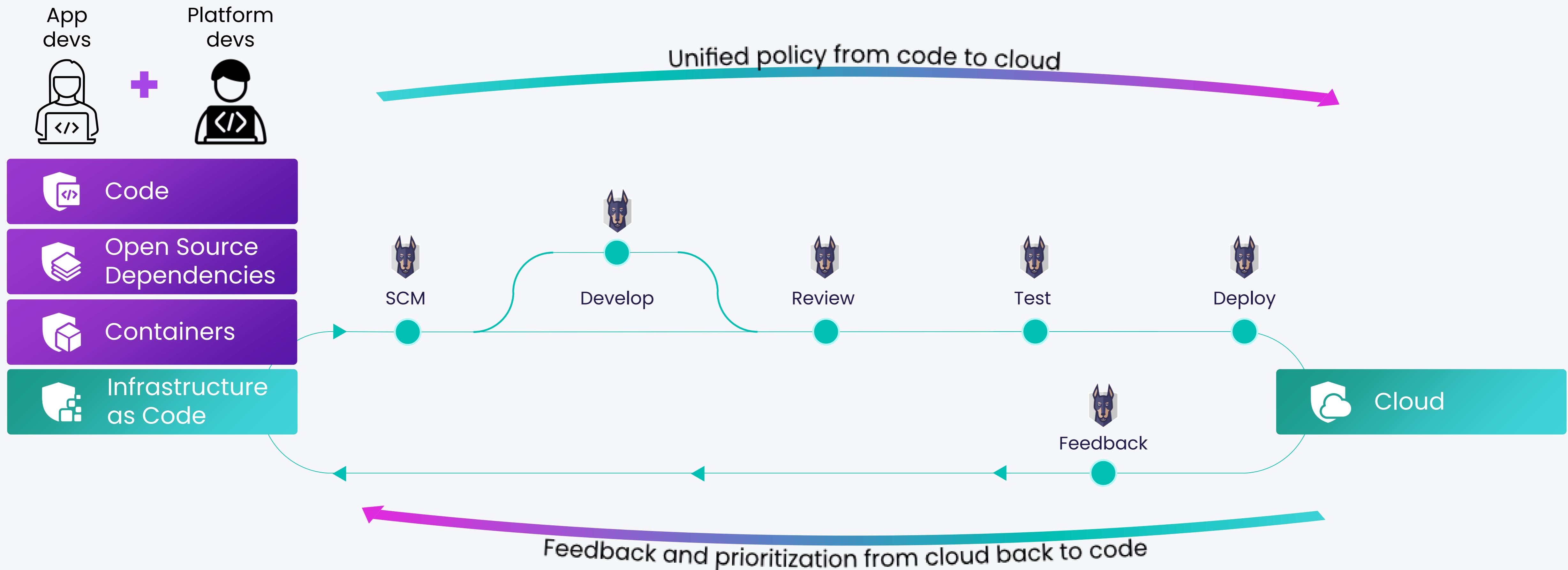
**In their natural  
workflow**



## Snyk Provides Visibility and Developer Guidance In Your Software Supply Chain

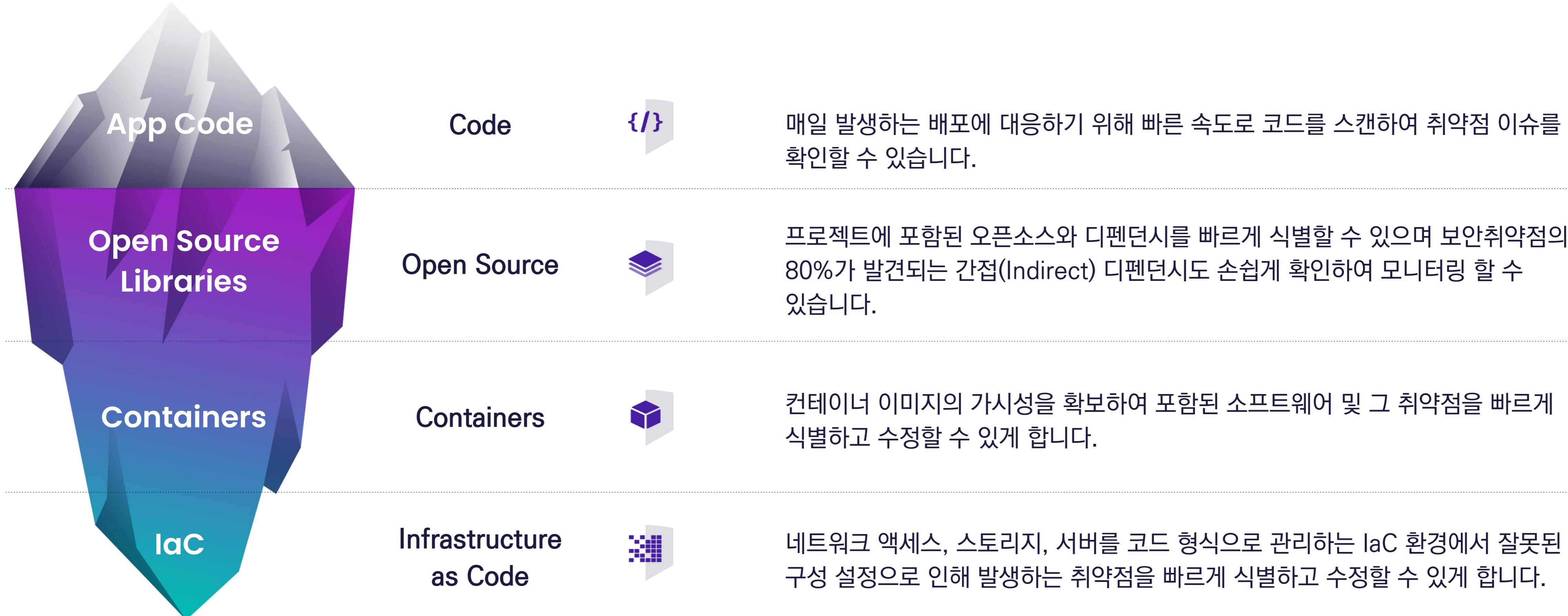


## Snyk empowers secure cloud developers



# Snyk 소개 | 개요

Snyk은 DevSecOps 환경에서 **개발 프로세스 지연 없이 지속적으로** 오픈소스의 보안취약점 및 라이선스 이슈를 식별, 모니터링 할 수 있도록 지원하는 **개발자 친화적인** 차세대 관리도구입니다.

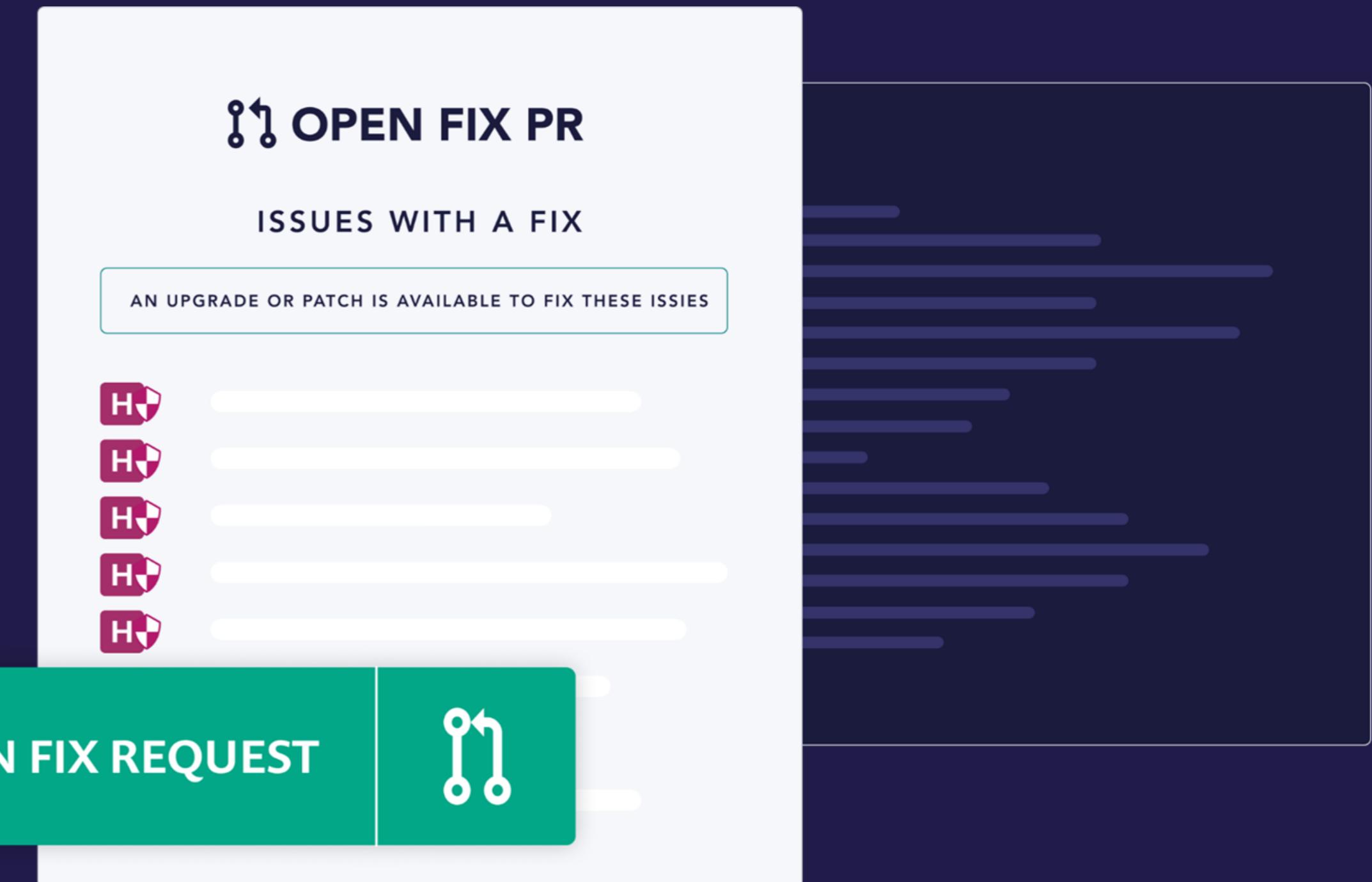


# Automated Remediation

## Solving the complex fix logic



**Single-click fix pull request**



## Be license compliant as early as coding

### Start early

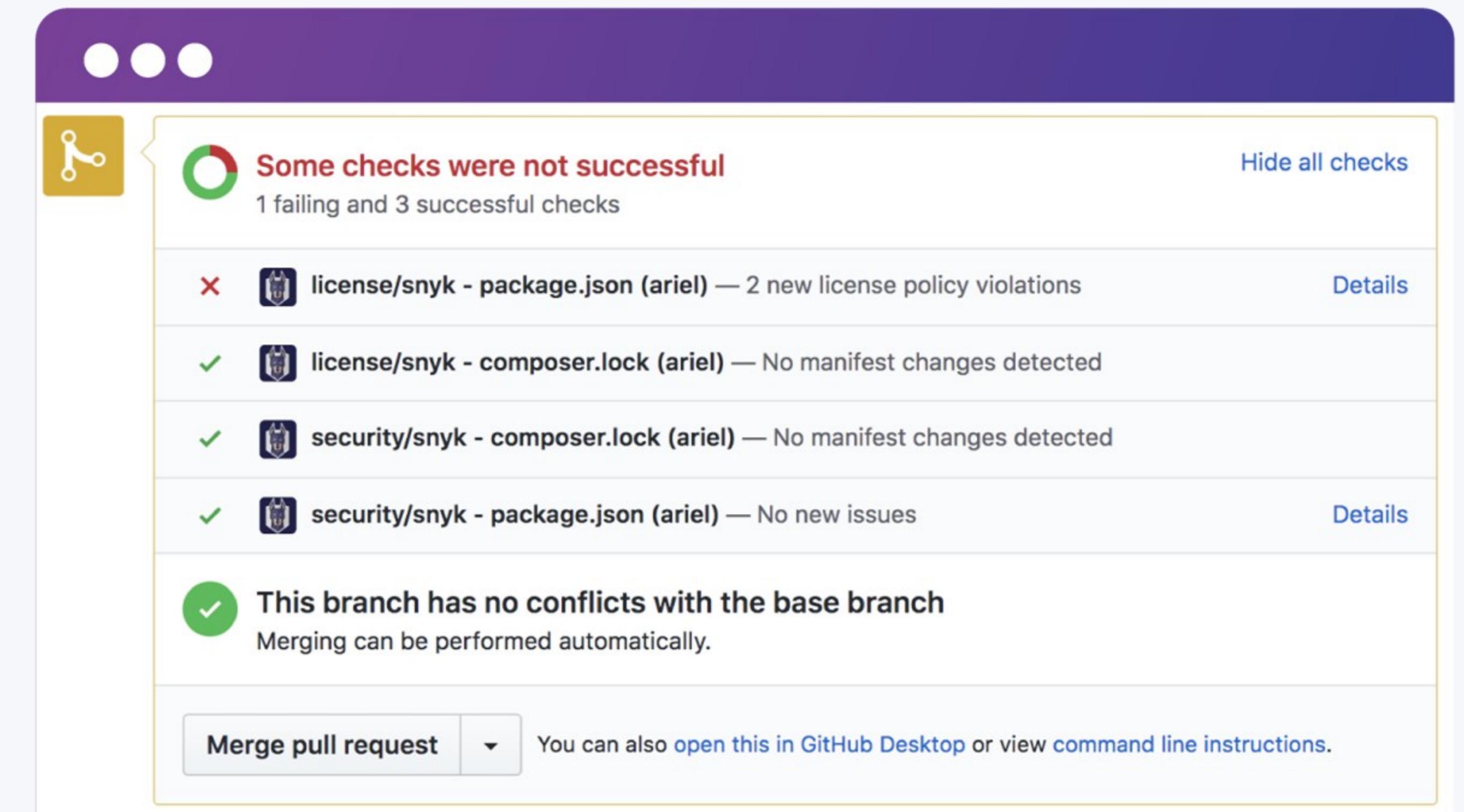
Verify compliance at every stage of development

### Scan

Get visibility to all the licenses that are being used.

### Comply

Define policies and take automatic actions to verify compliance.



**Copyright info**

**BOM report**

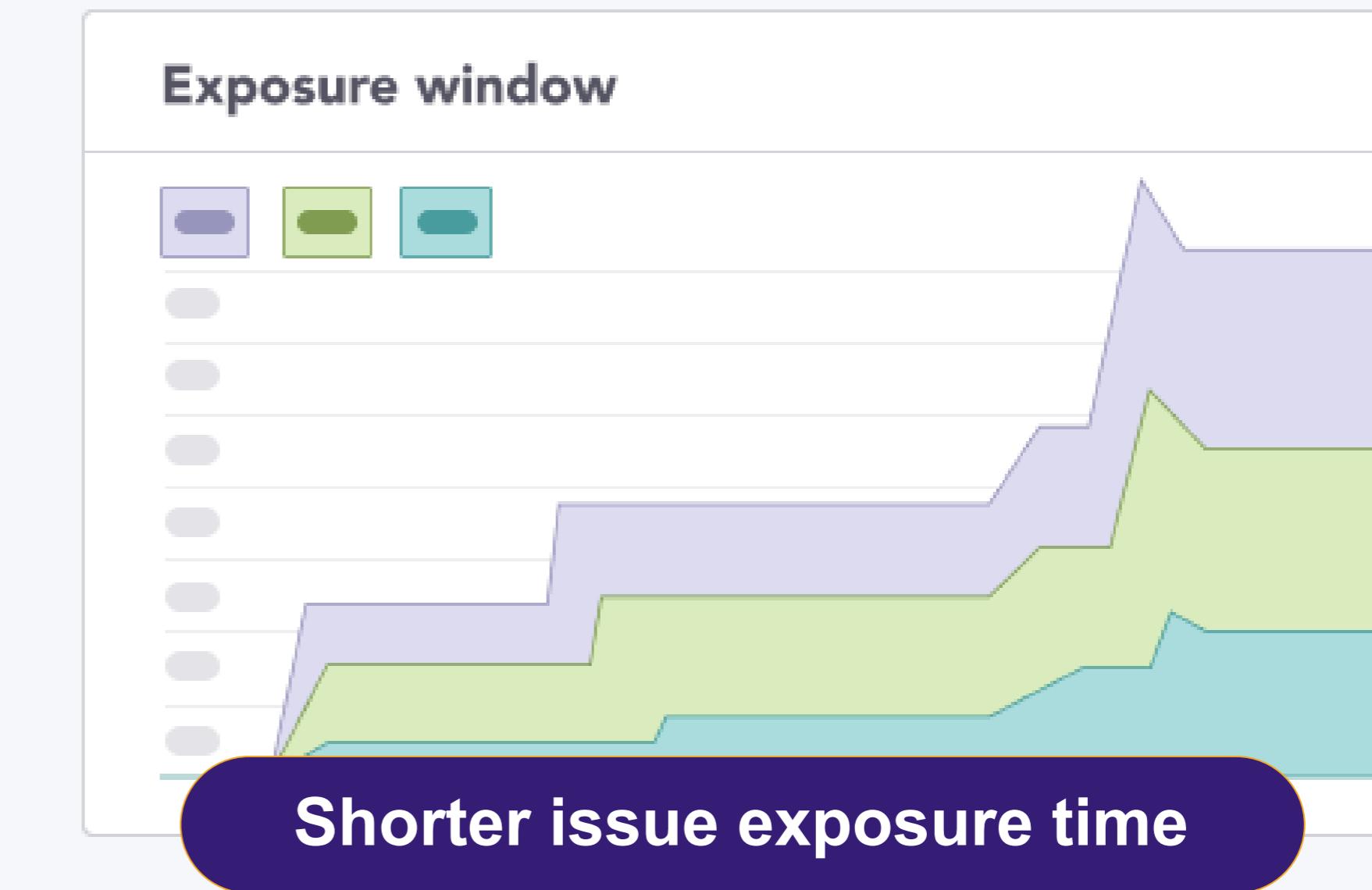
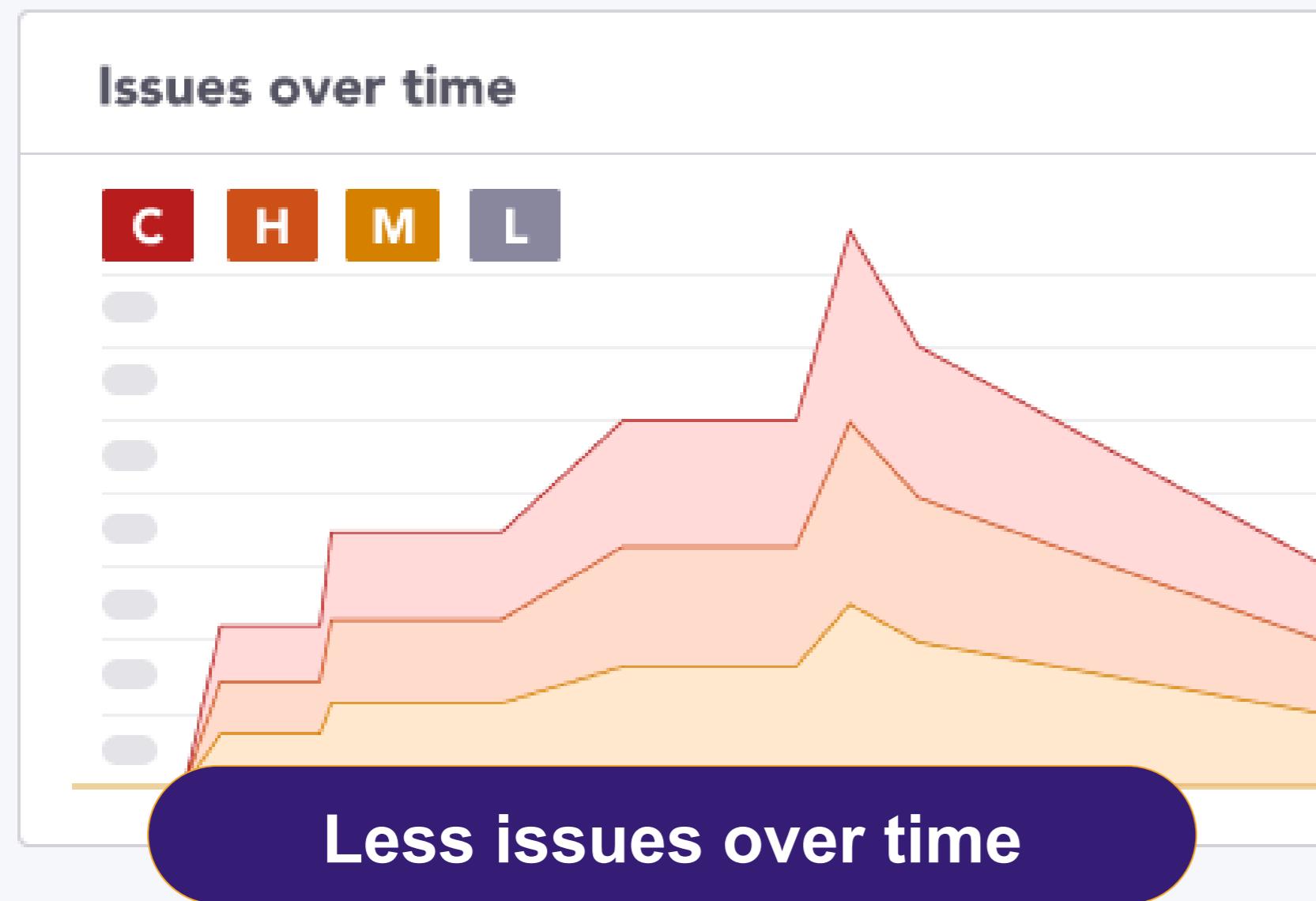
**Policies**

**Gating non-compliant packages**

**Legal team actionable instructions**

# Automated Remediation

## Fix MORE issues, QUICKER

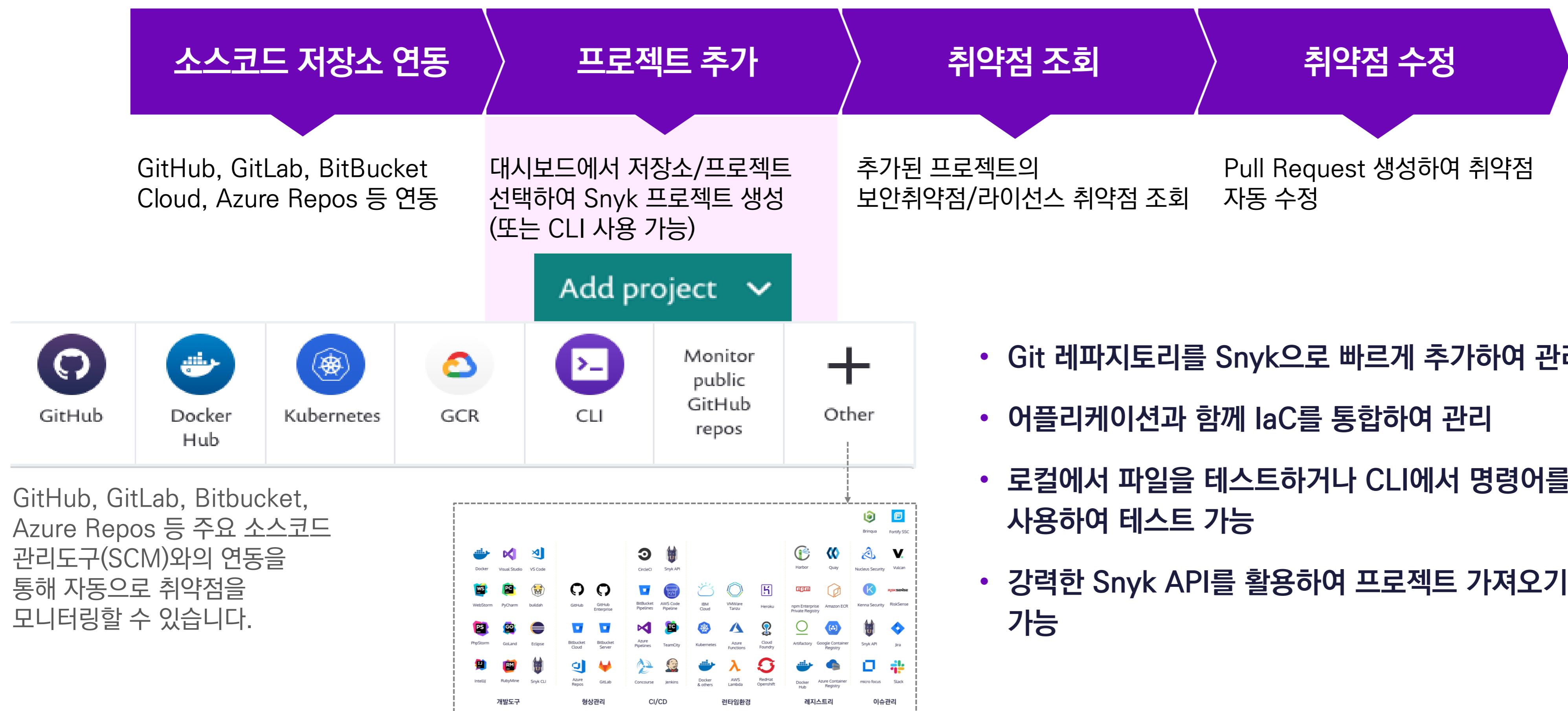


Accelerating MTTF (mean time to fix)

# Snyk 소개 | 사용 프로세스

Snyk은 저장소와 연동된 Snyk 웹UI를 통해 저장소/프로젝트를 관리하고 실시간으로 취약점을 조회하여 자동 수정하는 간편한 프로세스를 제공합니다.

## Snyk 사용 프로세스



- Git 레파지토리를 Snyk으로 빠르게 추가하여 관리
- 어플리케이션과 함께 IaC를 통합하여 관리
- 로컬에서 파일을 테스트하거나 CLI에서 명령어를 사용하여 테스트 가능
- 강력한 Snyk API를 활용하여 프로젝트 가져오기 자동화 가능

# Snyk 주요 기능 | 대시보드

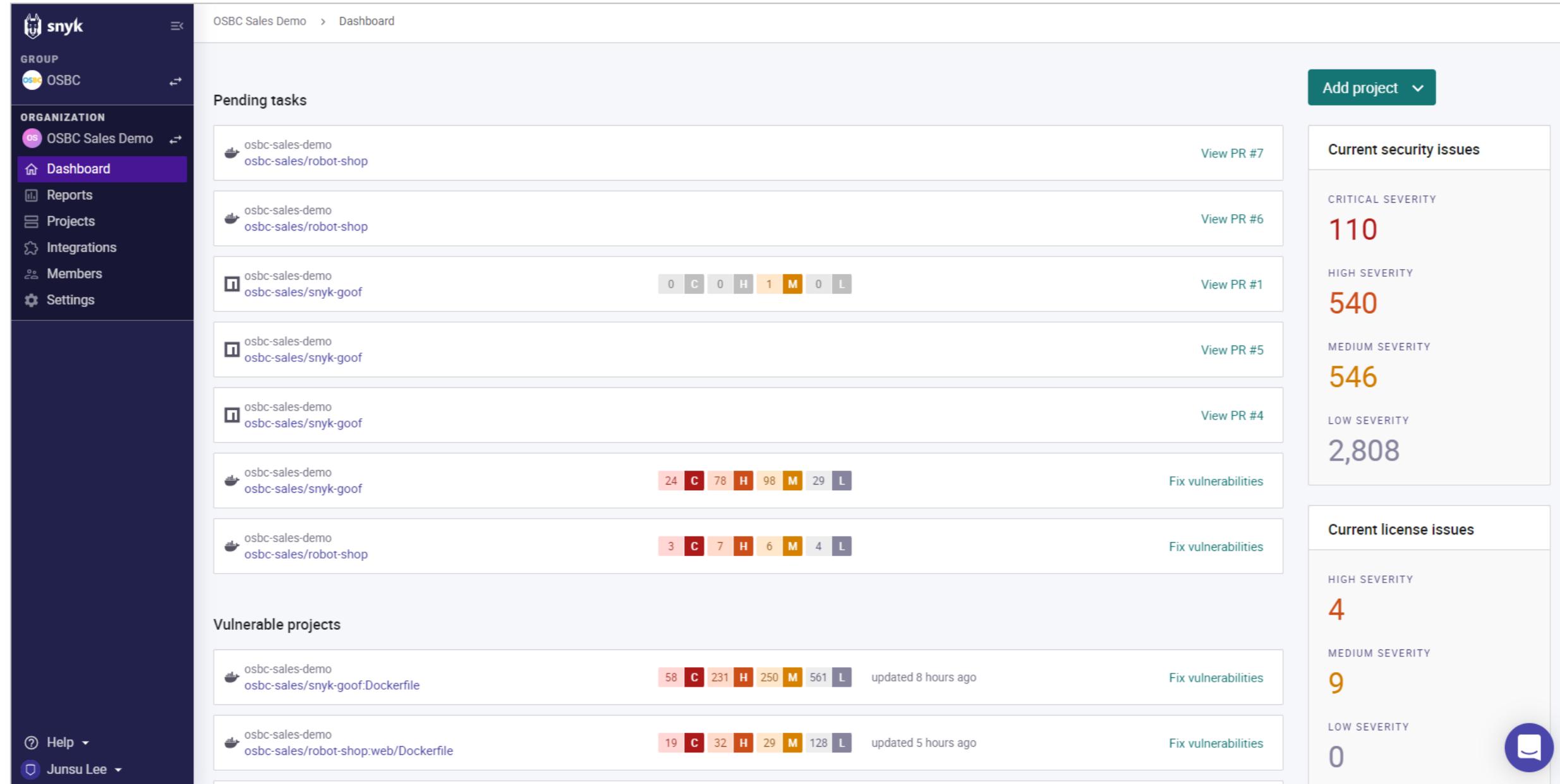
Snyk 대시보드에서는 사용자가 관리하고 있는 프로젝트 및 취약한 프로젝트의 현황을 직관적으로 표시해줍니다.

## 전반적인 보안취약점 상태

시간 경과에 따른 리스크,  
노출 현황, 보안을 위한 조치  
현황 모니터링

## 디펜던시 상태

전반적인 상태를 추적, 이슈  
컴포넌트를 식별하고 조치  
이행



## 프로젝트 관리

다수의 프로젝트에서 확인되는 보안취약점 및 라이선스 이슈를  
효율적으로 관리 가능

## 실행가능성

솔루션이 존재하는 이슈들을  
확인하고 자동으로 필터링

## 정책

자동화 되어있고 사용자 정의  
설정이 가능한 규칙을  
사용하여 보안 및 라이선스  
규정 준수 관리

# Snyk 주요 기능 | 프로젝트 현황 관리

Snyk의 프로젝트 현황 관리 화면에서 각 프로젝트별 Target 및 위험도 현황의 개요가 표시됩니다. 사용자는 각 프로젝트에 접근하여 이슈 확인, 수정, 세부 정보 확인 등의 세부 동작을 수행합니다.

The screenshot displays the Snyk Project Overview interface. On the left, a sidebar shows the organization structure under 'OSBC Sales Demo' with sections for Dashboard, Reports, Projects (which is selected), Integrations, Members, and Settings. Below the sidebar, four sections are listed: 'Snyk Code 분석 결과', 'Snyk Open Source 분석 결과', 'Snyk Container 분석 결과', and 'Snyk IaC 분석 결과'. The main content area shows a search bar at the top, followed by a summary message: 'The last import successfully processed 69 projects from 2 sources. View the last import log for more details.' Below this, two project entries are shown:

- osbc-sales/snyk-goof**: Contains 'package.json', 'Dockerfile', and 'Code analysis'. It has 61 vulnerabilities across severity levels C, H, M, and L. Last tested: a day ago.
- osbc-sales/robot-shop**: Contains 'cart/Dockerfile', 'catalogue/Dockerfile', 'dispatch/Dockerfile', 'fluentd/Kubernetes/fluentd.yaml', 'K8s/helm/templates/cart-deployment.yaml', and 'K8s/helm/templates/catalogue-deployment.yaml'. It has 46 vulnerabilities across severity levels C, H, M, and L. Last tested: a day ago.

A large purple rectangular callout highlights the second project entry, specifically the Dockerfile section, with the text: '제품 종류에 따라 프로젝트 아이콘, 분석 결과 출처 표기' (Indicates the source of analysis results based on product type).

제품 종류에 따라 프로젝트 아이콘, 분석 결과 출처 표기

# Snyk 주요 기능 | 프로젝트 세부

Snyk의 프로젝트 현황 관리 화면에서 각 프로젝트별 Target 및 위험도 현황의 개요가 표시됩니다. 사용자는 각 프로젝트에 접근하여 이슈 확인, 수정, 세부 정보 확인 등의 세부 동작을 수행합니다.

The screenshot shows the Snyk interface. At the top, there's a purple header bar with the Snyk logo, user profile 'jslee', and navigation links for Dashboard, Reports, Projects, Integrations, and Members. On the far right are settings, notifications, and a user icon.

**프로젝트 개요**

This section displays details for the project 'snyk-goof' (master branch). It includes the package.json file, creation date (Mon 10th Jan 2022), and a button to 'Retest now'. Below this are sections for 'IMPORTED BY' (Junsu Lee), 'PROJECT OWNER' (Add a project owner), 'ENVIRONMENT' (Add a value), and 'BUSINESS CRITICALITY' (Add a value).

**이슈 목록**

This section shows a list of issues. The 'Issues' tab is selected, showing 91 issues. Other tabs include 'Fixes' and 'Dependencies' (700). A search bar and a 'Fix these vulnerabilities' button are at the top of the list.

**오픈소스 디펜던시 목록**

This section lists vulnerabilities. The first item is 'npmconf - Uninitialized Memory Exposure' (Score: 756). It provides details like severity (High), priority score (42), and fix information (Introduced through npmconf@0.0.24, Fixed in npmconf@2.1.3). Buttons for 'Ignore' and 'Fix this vulnerability' are available.

# Snyk 주요 기능 | 프로젝트 세부 – Dependencies 결과화면

Dependencies 결과 화면에서 오픈소스 사용 내역(컴포넌트명, 버전 및 라이선스) 정보를 확인할 수 있습니다. 포스아이디 대비 최신버전에 대한 정보와 출시일 정보를 추가로 확인 가능합니다.

The screenshot shows the Snyk web interface for a project named 'snyk-goof'. The top navigation bar includes links for Dashboard, Reports, Projects, Integrations, Members, and account settings. The main content area displays project metadata like creation date, snapshot status, and various configuration sections. Below this is a summary bar with 'Issues' (93), 'Fixes', and 'Dependencies' (700). The 'Dependencies' tab is active, showing a table of dependencies with columns for name, latest version, last published date, security score (ISSUES), license, copyright, and paths. A search bar is also present above the table. The left sidebar lists 'Open Source Components' and 'Version' information. The right sidebar provides links for 'Issues', 'Licenses', 'Copyrights', and 'Paths'.

DEPENDENCY	LATEST	LAST PUBLISHED	ISSUES	LICENSES	COPYRIGHTS	PATHS
abbrev@1.0.7	1.1.1	Sep 28, 2017	0 C 0 H 0 M 0 L	ISC license	© Isaac Z. Schlueter...	2
abbrev@1.1.1	1.1.1	Sep 28, 2017	0 C 0 H 0 M 0 L	ISC license	© Isaac Z. Schlueter...	2
accepts@1.1.4	1.3.7	Apr 30, 2019	0 C 0 H 0 M 0 L	MIT license	© 2014 Jonathan Ong...	1
accepts@1.2.13	1.3.7	Apr 30, 2019	0 C 0 H 0 M 0 L	MIT license	© 2014 Jonathan Ong...	1
acorn@5.7.1	8.7.0	18 days ago	0 C 1 H 0 M 0 L	MIT license	© 2012-2018 by vari...	1
adm-zip@0.4.11	0.5.9	3 months ago	0 C 1 H 0 M 0 L	MIT license	© 1999 Masanao Izum...	1

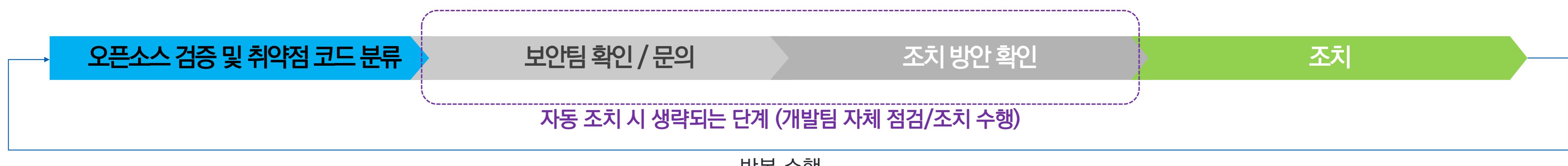
# Snyk 주요 기능 | 프로젝트 세부 – Pull Request 생성을 통한 자동 조치

Snyk Open Source의 Auto Fix 기능을 이용해 취약한 버전의 오픈소스를 클릭 한 번에 자동 패치하여 관리할 수 있습니다.

이슈 목록에서 Fix 버튼 클릭

Fix PR(Pull Request) 생성

Approve & Merge하여 반영



# Snyk IDE 플러그인 | Snyk for Visual Studio Code 예시

The screenshot shows the Snyk Code Vulnerability extension integrated into the Visual Studio Code interface. On the left, the 'OPEN SOURCE SECURITY' section displays 86 vulnerabilities found in package-lock.json, with a focus on adm-zip@0.4.7 - Arbitrary File Write via Archiv... and handlebars@4.0.14 - Prototype Pollution. The 'CODE SECURITY' section shows 21 vulnerabilities in index.js, with a detailed view of an 'Unsanitized input from the HTTP request body' issue at line 161. This issue is categorized as High severity and involves child\_process.exec, leading to a Command Injection vulnerability. The code snippet highlights the problematic line: `exec('identify ' + url, function (err, stdout, stderr) {`. The right panel provides context on the vulnerability, mentioning it was fixed by 54 projects and includes example fixes from chaitin/passionfruit. It also offers options to ignore the suggestion on the line or in the entire file.

SNYK

JS index.js 9+ ×

routes > JS index.js > `create` > `create`

135    var time = t.slice(reminder + remindToken.length);  
136    time = time.replace(/\n\$/,'');  
137  
138    var period = hms(time);  
139  
140    console.log('period: ' + period);  
141  
142    // remove it  
143    t = t.slice(0, reminder);  
144    if (typeof period != 'undefined') {  
145      t += '[' + ms(period) + ']';  
146    }  
147  
148 }  
149    return t;  
150 }

exports.create = function (req, res, next) {  
 // console.log('req.body: ' + JSON.stringify(req.body));  
 var item = req.body.content;  
 var imgRegex = /!\\[alt text\\]\\((http.\*\\s.\*\\)/;  
 if (typeof (item) == 'string' && item.match(imgRegex)) {  
 var url = item.match(imgRegex)[1];  
 console.log('found img: ' + url);  
 }  
 exec('identify ' + url, function (err, stdout, stderr) {  
 console.log(err);  
 if (err !== null) {  
 console.log('Error (' + err + '):' + stderr);  
 }  
 });  
 else {  
 item = parse(item);  
 }  
 new Todo({  
 content: item,  
 updated\_at: Date.now(),  
 }).save(function (err, todo, count) {  
 if (err) return next(err);  
 /\*  
 \*  
 \* res.setHeader('Data', todo.content.toString('base64'));  
 \* res.redirect('/');  
 \*/  
 res.setHeader('Location', '/');  
 res.status(302).send(todo.content.toString('base64'));  
 // res.redirect('/#' + todo.content.toString('base64'));  
 });  
};

OPEN SOURCE SECURITY

Snyk found 86 vulnerabilities

Analysis took 6.19s, finished at 03:28 PM, 10/12/21

package-lock.json goof - 86 vulnerabilities

C adm-zip@0.4.7 - Arbitrary File Write via Archiv...

C handlebars@4.0.14 - Prototype Pollution

C kerberos@0.0.24 - DLL Injection

C lodash@4.17.4 - Prototype Pollution

H adm-zip@0.4.7 - Directory Traversal

H ajv@6.10.2 - Prototype Pollution

H ansi-regex@3.0.0 - Regular Expression Denial o...

H bl@2.2.0 - Remote Memory Exposure

CODE SECURITY

Snyk found 21 vulnerabilities

Analysis took 2.96s, finished at 03:28 PM, 10/12/21

index.js routes - 11 vulnerabilities

H Unsanitized input from the HTTP request body ...

H Unsanitized input from the HTTP request body ...

H Unsanitized input from the HTTP request body ...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

M This endpoint handler performs a file system o...

CODE QUALITY

Snyk found 4 issues

Analysis took 2.96s, finished at 03:28 PM, 10/12/21

index.js routes - 3 issues

utils.js goof - 1 issue

HELP & FEEDBACK

Help Snyk to make a better extension

Send us feedback or report a bug

Top 3 FAQ

1. How to get the most out of Snyk's extension?

2. How to ignore files and directories?

Add default .dcignore file to your workspace

Add a custom .dcignore file to your workspace

Snyk Code Vulnerability ×

High

Unsanitized input from the HTTP request body [:155] flows [:155, :155, :157, :157, :158, :158, :158, :158, :159, :161, :161] into child\_process.exec [:161], where it is used to build a shell command. This may result in a Command Injection vulnerability.

This vulnerability happens on line 161

More info

Security Maintenance Synclet Spawn Command

This vulnerability was fixed by 54 projects. Here are 3 example fixes.

chaitin/passionfruit Example 1/3

```
})  
.post('/spawn', async ctx => {  
  let pid = await state.device.spawn([ctx.request.body.bundle])  
  let { device, bundle } = ctx.request.body  
  let dev = await FridaUtil.getDevice(ctx.params.device)  
  let pid = await dev.spawn([ctx.request.body.bundle])  
  // todo: attach  
  ctx.body = { status: 'ok' }
```

Support multiple synclets for a provider.  
Handle errors in spawn  
Refactoring log command.

Do you want to hide this suggestion from the results?

Ignore on line 161 Ignore in this file

# Snyk 부가 지원 기능 | Snyk Vulnerability DB

Snyk Vulnerability DB는 가장 포괄적이고, 정확하며 시기적절한 정보를 제공하는 오픈소스 취약점 정보 DB입니다.

The screenshot shows a detailed view of a vulnerability entry. At the top, there's a navigation bar with 'snyk Vulnerability DB', 'Developer Tools', and 'About Snyk'. Below it, a breadcrumb trail shows 'Snyk Vulnerability Database > Maven > org.springframework:spring-beans'. A search bar is at the top right. The main content area has a title 'Remote Code Execution' and a sub-section 'Affecting org.springframework:spring-beans package, versions [ ,5.2.20) [5.3.0, 5.3.18]'. It features a large red circular badge with a '9.8 CRITICAL' rating. To the right of the badge, there's a 'Snyk CVSS' section with five metrics: Exploit Maturity (Mature), Attack Complexity (LOW), Confidentiality (HIGH), Integrity (HIGH), and Availability (HIGH). Below this is a 'See more' button. Further down, there's a 'NVD' section with a '9.8 CRITICAL' badge. A 'Test your applications' button is present. On the left, there's an 'Overview' section with a brief description of the package and its vulnerabilities, followed by a 'Note' section with several bullet points about exploitability and remediation. At the bottom, there's a 'PoC' section with a terminal window showing a command-line exploit script.

## ✓ 다양한 패키지매니저 기반 정보 제공

npm, Maven, Nuget, pip, cocoapods 등 패키지매니저와 Linux 배포판별 오픈소스 및 비정형 C/C++ 기반 오픈소스까지 폭넓게 지원

## ✓ 자체 전문가들이 최초 발견한 취약점 보유

Snyk 자체 전문가들이 최초로 발견한 취약점 정보를 다량 보유하여 NVD 정보 대비 더 빠르게 정보 제공

## ✓ 조치 정보 제공

취약점 패치 정보 뿐만 아니라 Workaround 및 PoC 등 포괄적인 정보 제공

# Snyk 부가 지원 기능 | Snyk Advisor

Snyk Advisor를 통해 오픈소스 패키지의 최신 취약점 정보를 제공합니다.

The screenshot shows the Snyk Advisor interface. At the top, there's a search bar with 'Search packages' and a dropdown for 'npm'. Below the search bar, a purple banner says 'Find the best package for your next project.' and 'Search and compare over 1 million open source packages.' The main content area displays a search result for 'nodemon v2.0.7'. It includes a 'Package Health Score' of 90 / 100, a popularity chart showing weekly downloads (3,810,952), and a maintenance chart showing commit frequency and open issues. Below this, there are four sections: 'Popularity' (with a download icon), 'Maintenance' (with a code icon), 'Security' (with a lock icon), and 'Community' (with a people icon). Each section has a brief description and a link to more details. At the bottom, it says 'Popular packages for:' followed by the npm logo and a dropdown menu.

## ✓ 100만개 이상의 오픈소스 패키지 정보

Javascript, Python, Go 기반의 오픈소스 패키기 정보 보유

## ✓ 패키지 평가 등 통계 정보 제공

평판(인기도), 유지보수 현황, 보안취약점, 커뮤니티 활성화 정도 등 다양한 지표를 기준으로 한 평가 점수 등 다양한 통계 정보 제공

## ✓ 패키지별 세부 정보 제공

코드 샘플, 참고 가능한 유사 패키지, FAQ 등

# Snyk 부가 지원 기능 | Snyk Learn

Snyk Learn을 통해 개발자를 위한 보안 조치 교육 프로그램을 제공합니다.

The screenshot shows the Snyk Learn interface for a SQL injection course. At the top, there are 'Log in' and 'Sign up' buttons. Below that, a breadcrumb navigation shows 'Snyk Learn > JavaScript SQL injection'. The main title is 'SQL injection'. Underneath, it says 'SQL injection: the basics' and 'TL;DR'. On the left, there is a code editor window showing Java code for deserialization:

```
1 package class;
2 
3 public class EvilUser implements Serializable {
4     private String command;
5 
6     public EvilUser(String command) {
7         this.command = command;
8     }
9 
10    private void readObject(ObjectInputStream in) throws IOException {
11        in.defaultReadObject();
12        in.readInt();
13        in.readObject();
14        in.readInt();
15    }
16 }
```

Below the code editor, a note states: "It implements the `Serializable` Interface, which tells Java that this class is OK for serialization and deserialization." There are 'Previous' and 'Next' navigation buttons at the bottom. To the right of the code editor, there is a sidebar titled 'JS - SQL injection' with a checklist. Below the sidebar, there are icons for various programming languages: Python, Java, C, C++, C#, PHP, JavaScript, and Docker. A large blue arrow points from the sidebar towards the language icons.

## ✓ 보안 전문가가 제작한 무료 교육

개발자를 위해 업계 전문가가 제작한 전문 교육으로  
직관적인 온라인 교육 제공

## ✓ 필요한 관심 주제를 바로 학습 가능

즉석 학습 가능한 컨텐츠로 취약점 타입별로 바로  
검색하여 학습할 수 있는 환경 제공

## ✓ 자가 테스트 가능한 환경 제공

자체 코드에서 발견된 문제를 기반으로  
보안취약점을 이해하고, 수정하여 이슈 방지 가능

# Snyk 특장점 | 다양한 도구와 연동체계



Docker Visual Studio VS Code



WebStorm PyCharm buildah



PhpStorm GoLand Eclipse



IntelliJ RubyMine Snyk CLI



Github GitHub Enterprise



Bitbucket Cloud Bitbucket Server



Azure Repos GitLab



CircleCI Snyk API



BitBucket Pipelines AWS Code Pipeline



IBM Cloud VMWare Tanzu Heroku



Kubernetes Azure Functions Cloud Foundry



Concourse Jenkins



Docker & others AWS Lambda RedHat OpenShift



Brinqua Fortify SSC



Nucleus Security Vulcan



Kenna Security RiskSense



Artifactory Google Container Registry



Snyk API Jira



Docker Hub Azure Container Registry

개발도구

형상관리

CI/CD

런타임환경

레지스트리

이슈관리

# Snyk 특장점 | Snyk Vulnerability DB

Snyk Intel Vulnerability DB는 CVE 취약점 대비 441% 더 많은 취약점 정보를 포함하고 있고 NVD 대비 최대 46일 더 빠르게 취약점을 탐지하였습니다. NVD에 기록된 JavaScript 관련 취약점의 92%는 Snyk이 먼저 탐지한 취약점들입니다.



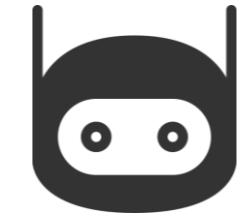
## 다양한 취약점DB로부터 풍부한 데이터 확보

CVE, NVD와 같은 리소스를 통해 수집되는 데이터는 분석, 테스트 및 정제되어 데이터베이스에 포함됩니다.



## 새로운 취약점에 대한 독점 리서치

Snyk 보안팀은 제로데이 취약점과 같은 주요 컴포넌트의 심각한 취약점을 발견하기 위해 노력하고 있습니다.



## 위협 인텔리전스 시스템

보안게시판, Jira 게시판, Github 커밋 등을 통해 미보고된 취약점을 자동으로 식별합니다.



## 커뮤니티와의 관계

커뮤니티와 협업하고 신규 취약점에 대한 현상금 제도를 운영하여 공개커뮤니티로부터 수백개의 취약점을 수집하고 있습니다.



## 학계와의 협업

버클리, 버지니아공대, 워터루 등 학계의 박사급 연구진들과 파트너관계를 맺고 도구, 방법론 및 데이터를 교환하고 있고 결과는 독점적으로 공개됩니다.

# Snyk 특장점 | 주요 특장점

1

## 개발자 우선 접근방식, DevSecOps 실현

개발자가 편리하게 사용하도록 설계되어 마찰이 없고 직관적입니다. 개발자가 현재 사용하는 도구에서 직접 문제를 쉽게 찾을 수 있을 뿐만 아니라 빠르게 수정할 수 있도록 지원합니다.

2

## 업계 최고의 광범위한 보안 인텔리전스 데이터베이스

Snyk의 데이터베이스는 기존 상용 데이터베이스보다 441% 더 많은 취약점 정보를 포함하고 기존 상용 취약점 대비 평균 46일 더 빠르게 취약점을 식별하였습니다.

3

## 자동 Fix를 통한 관리 리소스 최소화

소스코드 저장소와 연동된 UI를 통해 Pull Request를 생성하고 자동으로 Fix하거나 CLI에서는 명령어 하나로 자동으로 취약점 없는 버전으로 Fix 할 수 있는 기능을 제공합니다.

# Thank you!



snyk