Open-Source Security: Challenges, Solutions and Opportunities

Liu Yang

Professor at Nanyang Technological University,
Co-Founder of Scantist

Open-Source Usage

SOFTWARE DEVELOPMENT TODAY USES A LOT OF OPEN-SOURCE

60 – 90%
APPLICATION CODE IS OPEN SOURCE

100K+
OPEN-SOURCE RELEASES PER DAY

300M
OPEN-SOURCE LIBRARIES BY 2026

1 : 8
ONE OPEN-SOURCE COMPONENT CALLS 8 OTHERS ON AVERAGE

Open-Source Security Risks

OPEN-SOURCE RISKS NEED TO BE MANAGED

200+

Daily published vulnerabilities in OSS

1 : 4

Data Breaches caused by OSS Vulnerabilities

76%

Portion of Asia Pacific organizations that have no process of managing OSS

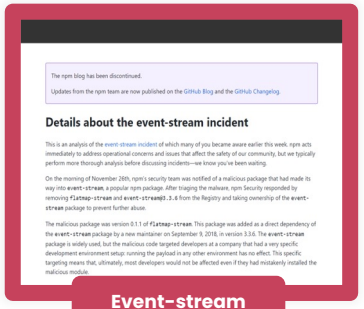
Publicity Accessible

Easier to be utilized by malicious agencies

500+

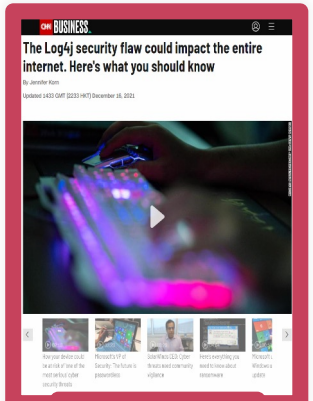
Licenses of OSS

THE CONSEQUENCES

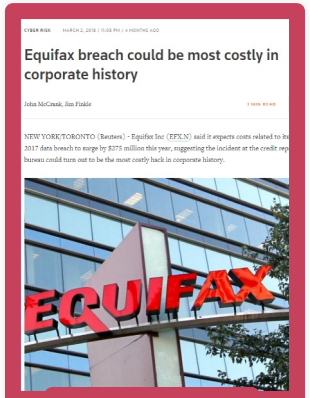


Event-stream

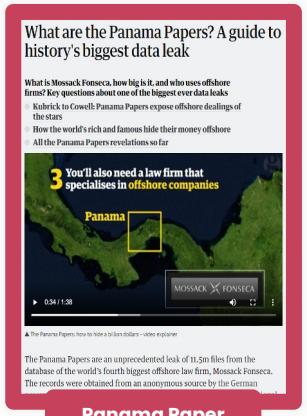
OpenSSL



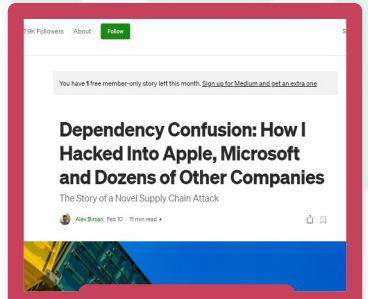
Log4Shell



Equifax



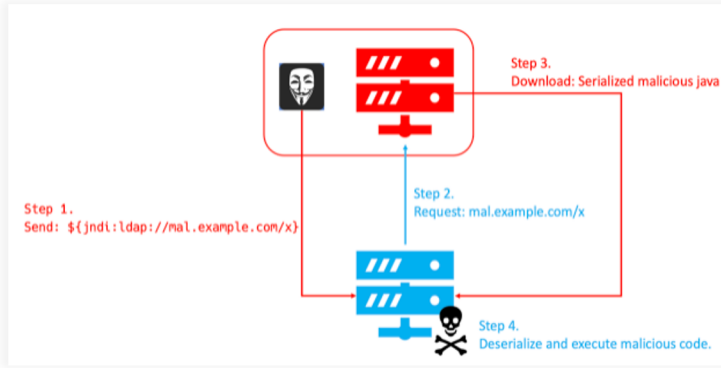
Panama Paper



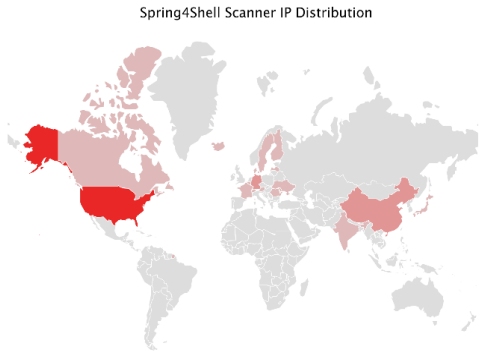
Dep Confusion

40,000+ KNOWN OPEN-SOURCE VULNERABILITIES TO DATE

OSS vulnerabilities emerge in big waves

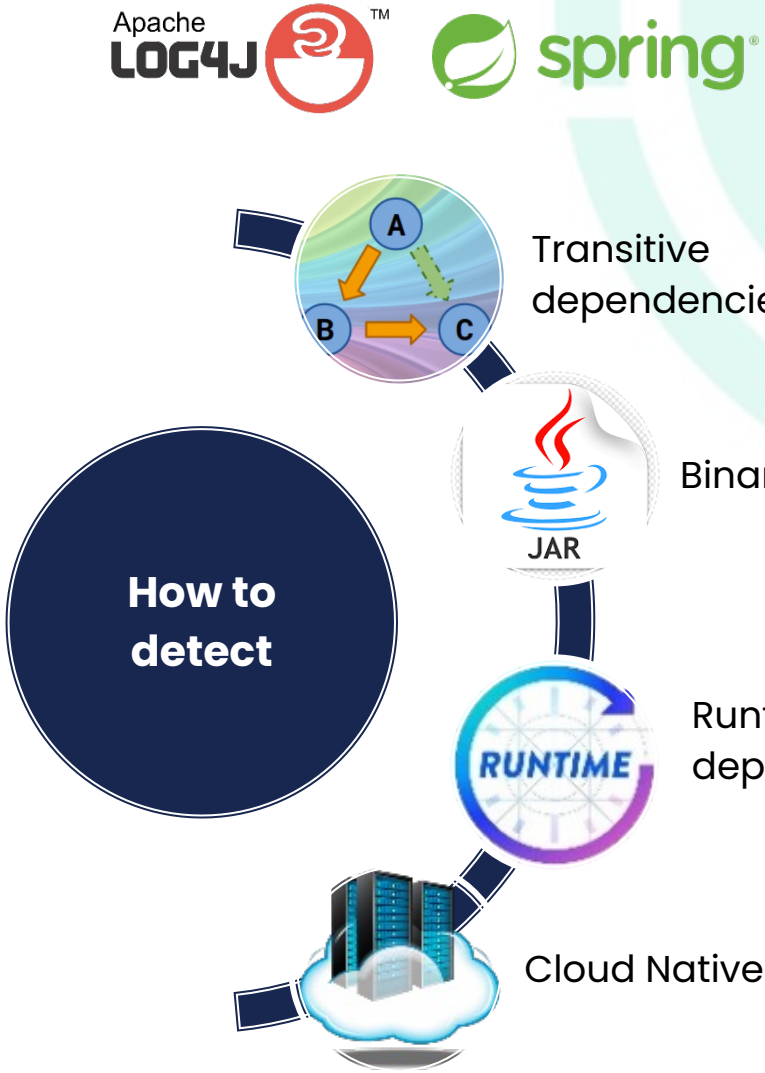


Example of attack based on log4shell vulnerability (CVE-2021-44228)

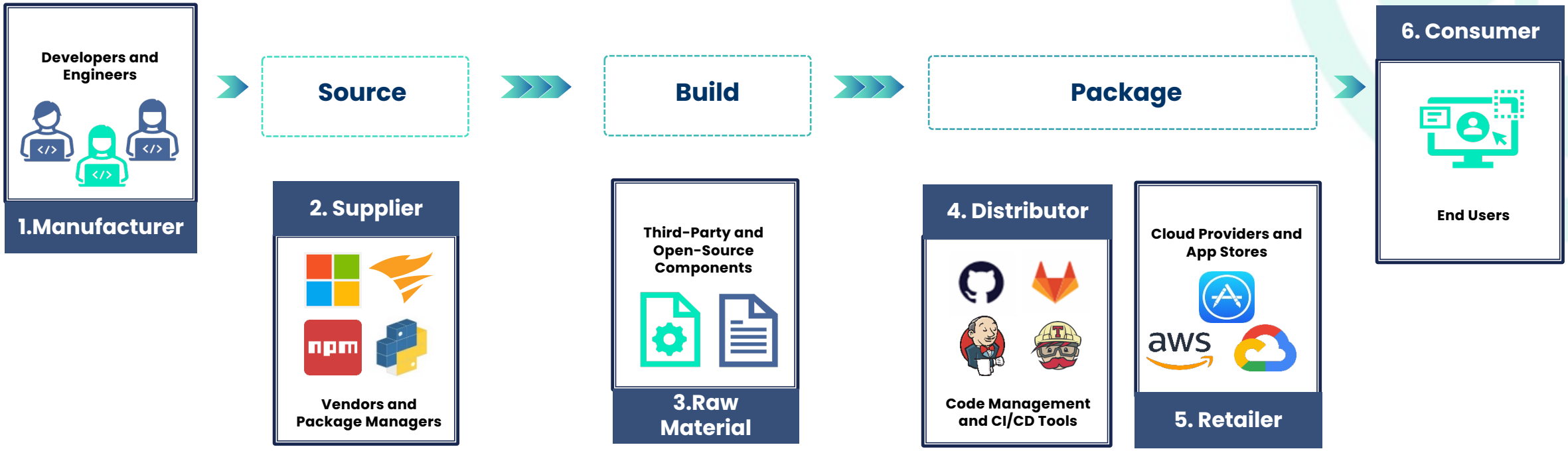


Part of IP location distribution of scanning and utilizing Spring4Shell vulnerability

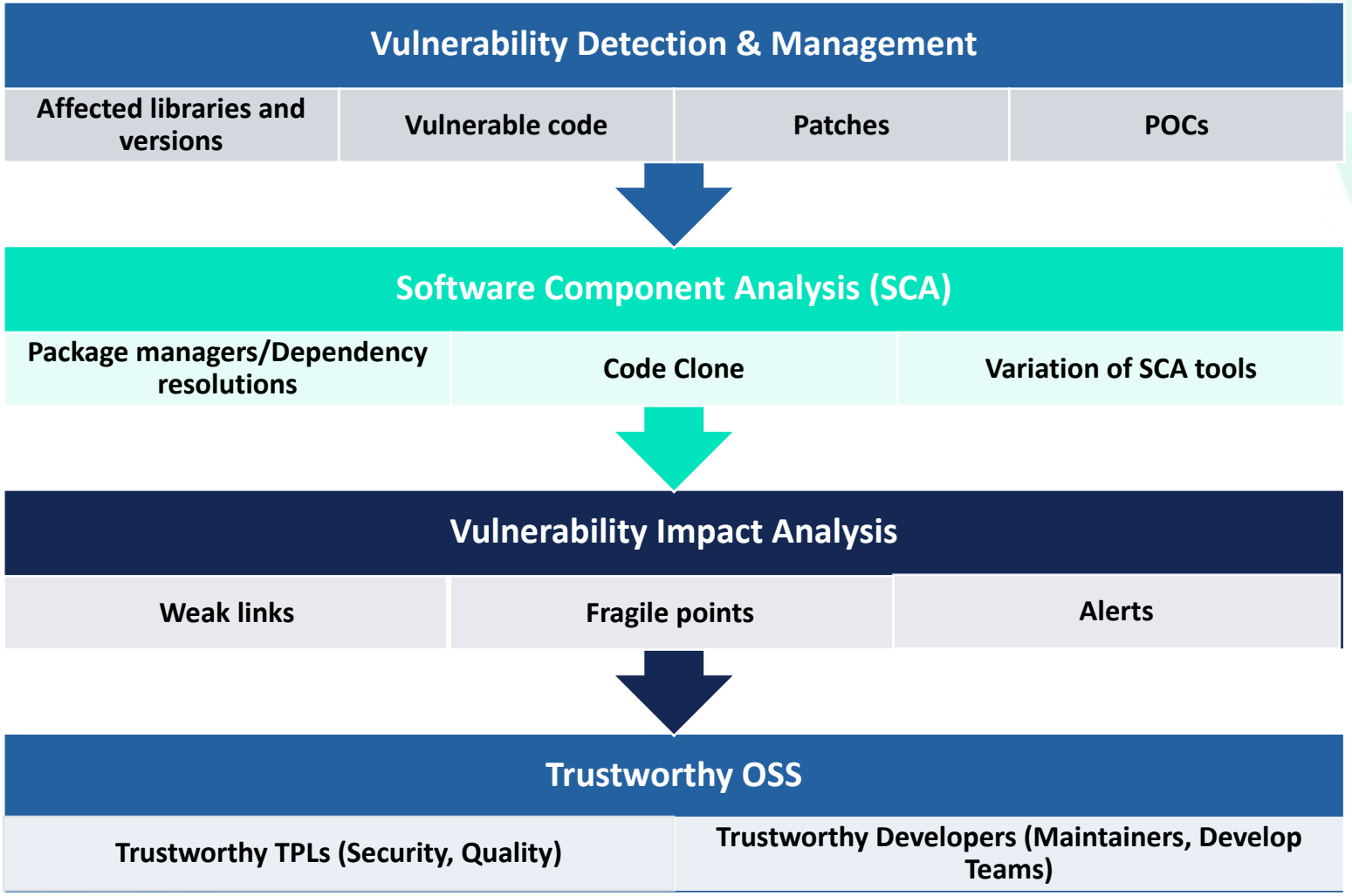
Hackers utilize vulnerabilities, e.g., Log4Shell, Spring4Shell to deploy malicious software



The Modern-day Software Supply Chain



Challenges in OSS Security



OSS Related Research (Outline)

Vulnerability Detection & Management

- Vulnerable Signature Matching for Partial Code
  - **MVP for Source (Usenix 20)**
- Vulnerability Patch Finder
  - **Tracer (TSE21)**

Software Component Analysis (SCA)

- **ATVHunter for Java Binary and APK (ICSE 21, TSE 21, TSE 22)**
  - **An Empirical Assessment of Security Risks of Global Android Banking Apps (ICSE 20)**
- **Modx for C/C++ Binary (ICSE 22)**
- **SCA Comparison (ICSE23 under review)**
- **LiDetector: License Incompatibility Detection for Open Source Software (TOSEM 22)**

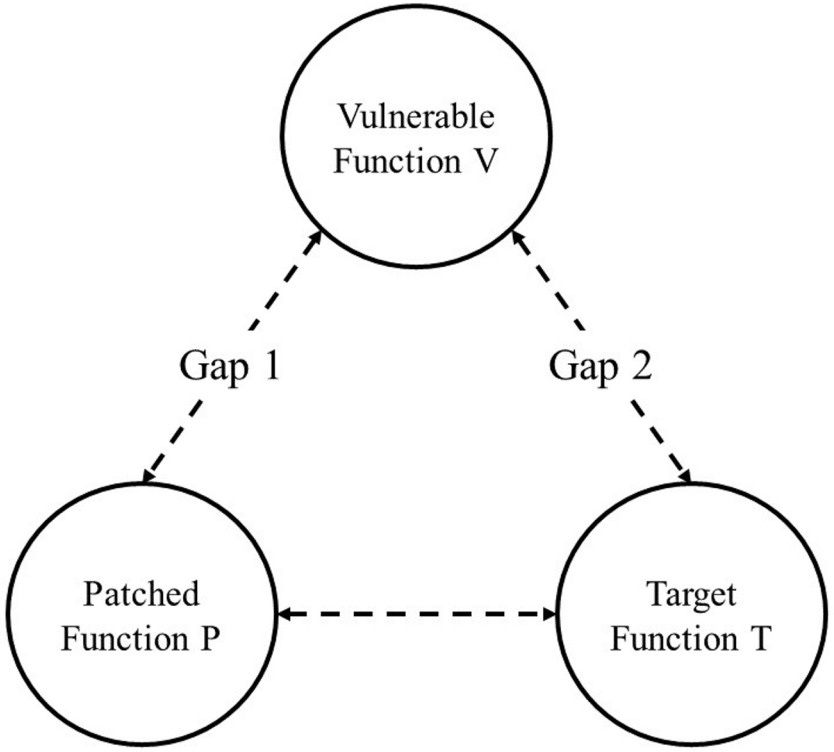
Trustworthy OSS

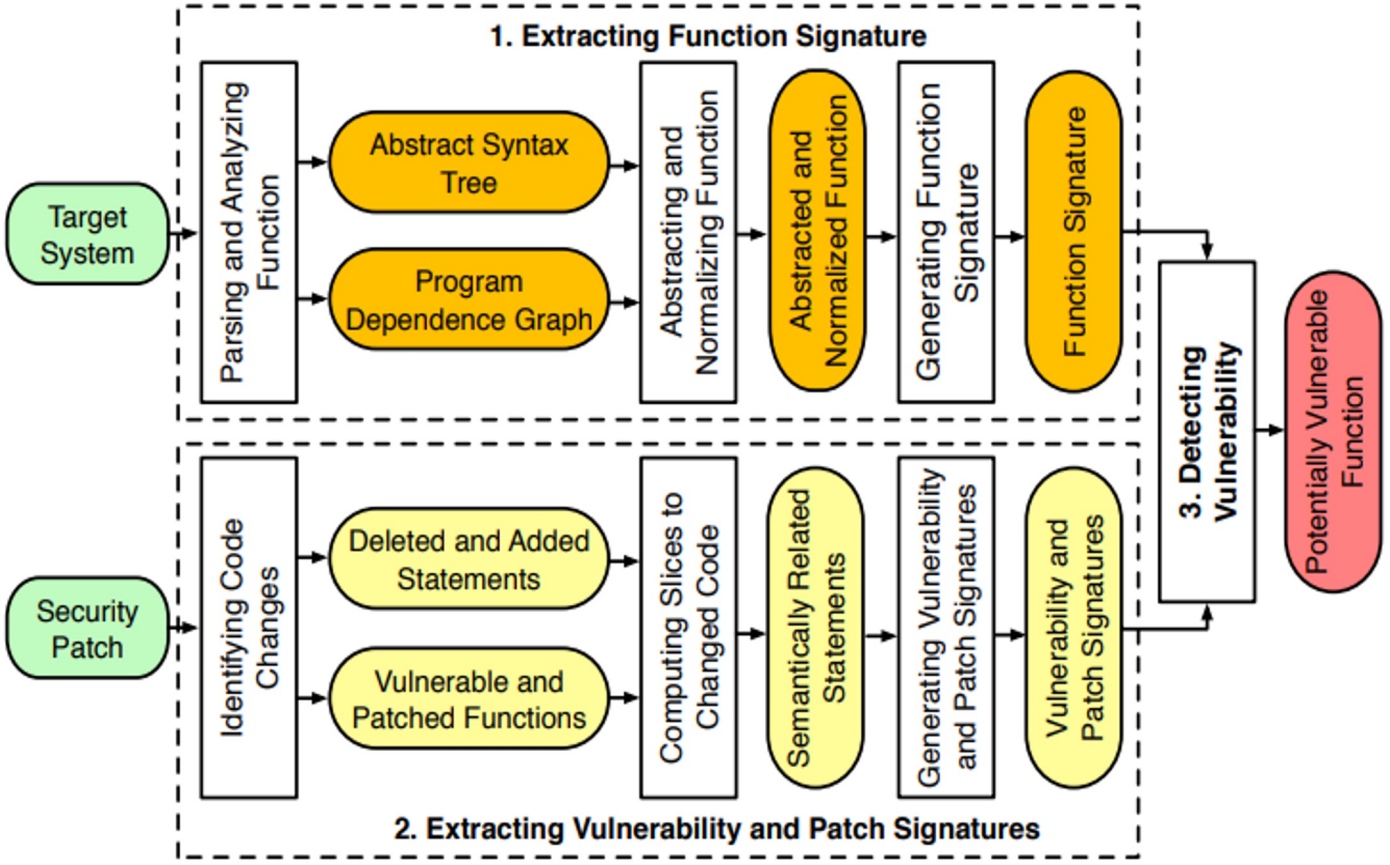
- **Maven Critical Libraries (ICSE23 under review)**
- **Semantic Version Checker (ASE 22)**

Vulnerability Impact Analysis

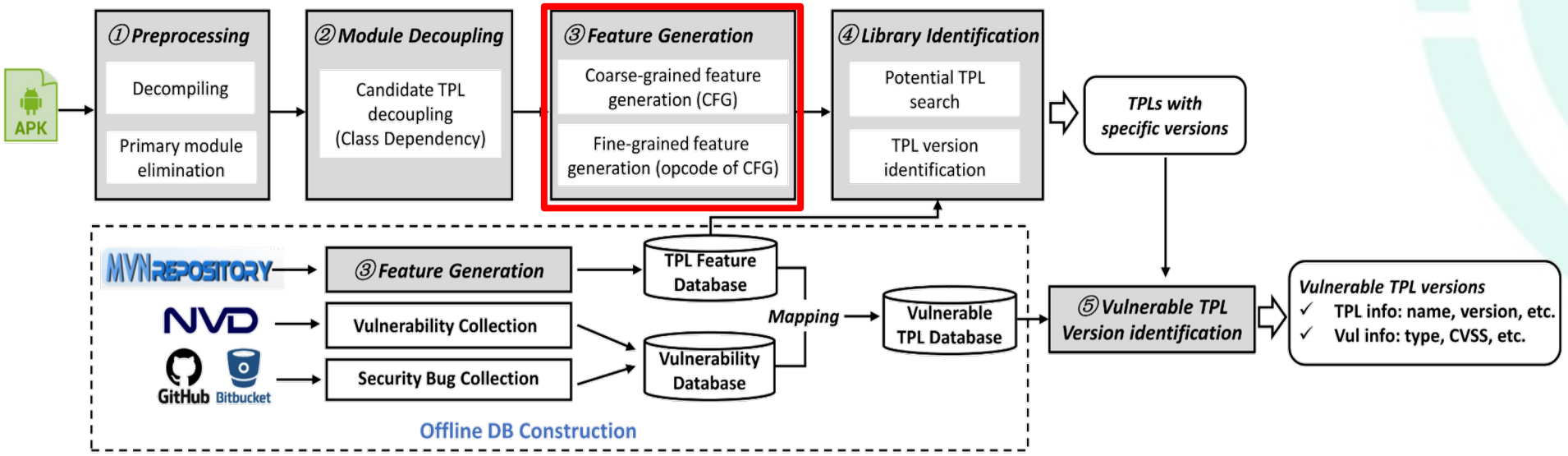
- **Demystifying the Vulnerability Propagation and Its Evolution via Dependency Trees in the NPM Ecosystem (ICSE 22)**
- **Reachability analaysis of JavaScript and its impact in the NPM Ecosystem (TOSEM 22)**
- **C/C++ Dependency Analysis (ASE 22)**

MVP: Semantics Based Vulnerability Detection





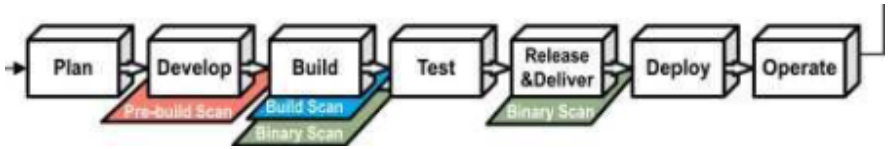
ATVHunter: Reliable Version Detection of Third-Party Libraries for Vulnerability Identification



- Various import mode
- Different version similarity
- Code obfuscation

- Two-phase detection
  - Coarse-grained feature (CFG) stores semantic information
  - Fine-grained feature which can effectively distinguish tiny differences among different versions
- Fuzzy Hashing

Maven SCA and Scan Accuracy



DevSecOps with Scan Mode

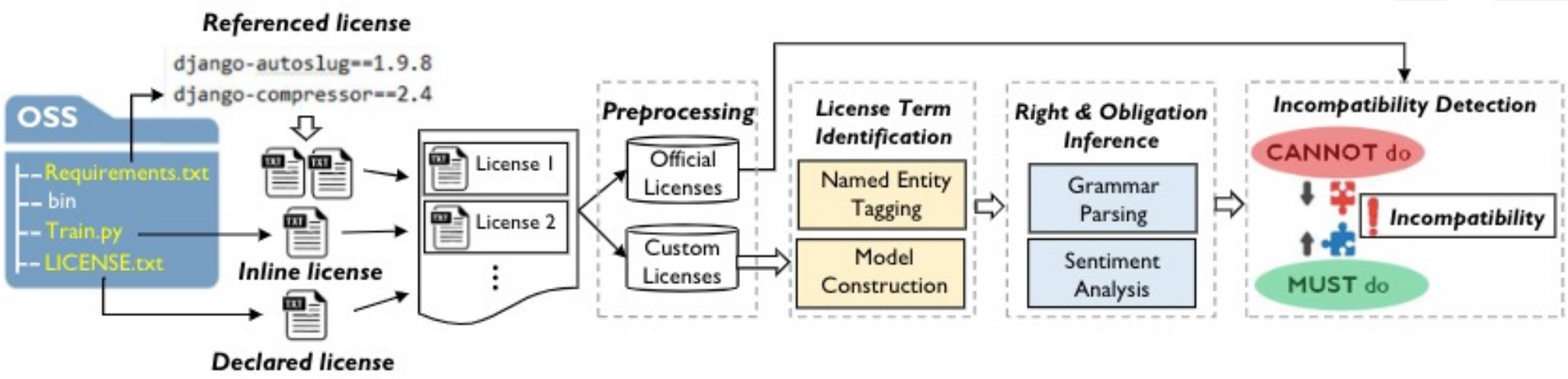
	Build Scan				Pre-build based Scan	
elements	OWASP	Steady	Scantist	OSSIndex	Scantist	Dependabot
dependency-Management	✓	✓	✓	✓	✓	X
exclusion	✓	✓	✓	✓	✓	X
parent	✓	✓	✓	✓	✓	X
aggregation	✓	✓	✓	✓	✓	X
profiles	✓	✓	✓	✓	✓	✓
optional	✓	✓	✓	✓	✓	✓
version-range	✓	✓	✓	✓	✓	✓
variable as version	✓	✓	✓	✓	✓	X

Maven Dependency Managing Features

- **SCA tools should consider three aspects, namely the scan algorithms, modes, and scopes.**
- **The experimental results show that there are large gaps between the detection results of existing tools and the ground truth provided according to SCA Scope Guidelines.**

Lida, esecfse2022scastudy, Software Composition Analysis for Vulnerability Detection: An Empirical Study on Java Projects

LiDetector: License Incompatibility Detection



- Dataset: 1,846 projects
- Lidetector identified 1,346 projects with license incompatibility issues, most of which are caused by Project licenses vs. component license.

Vulnerability Propagation and Its Evolution via Dependency Trees in the NPM Ecosystem

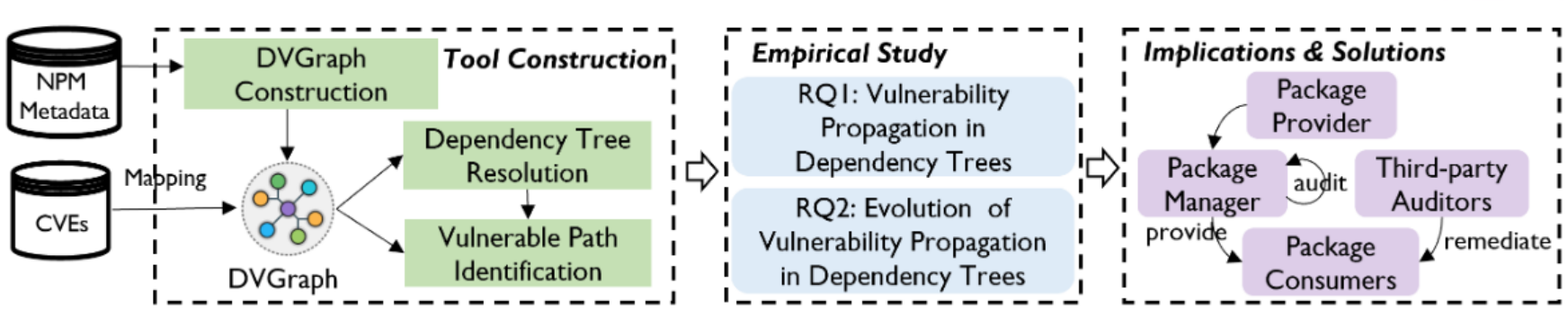
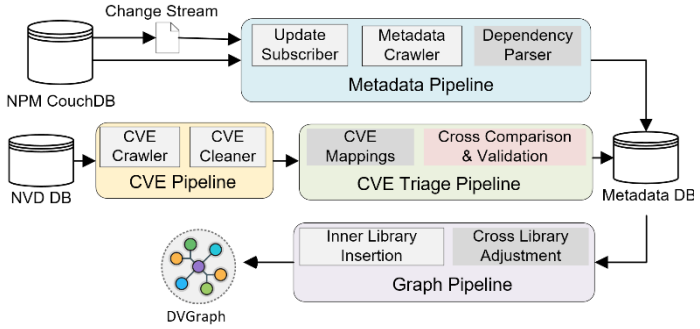
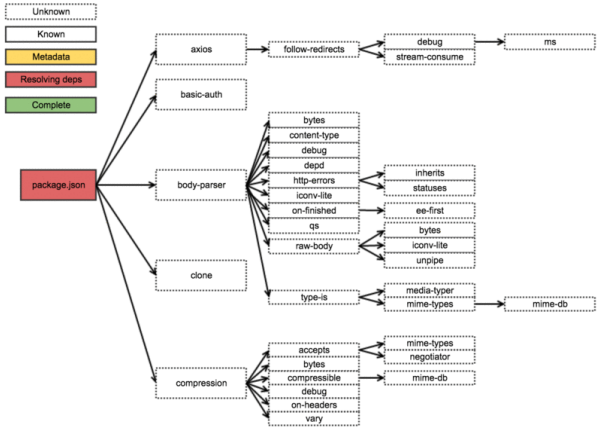
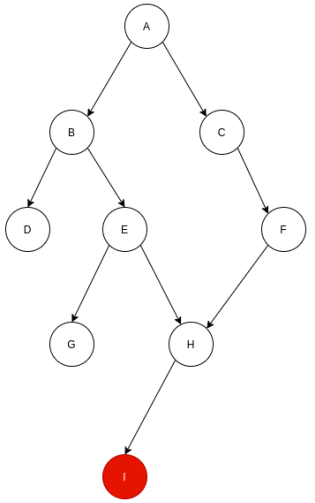


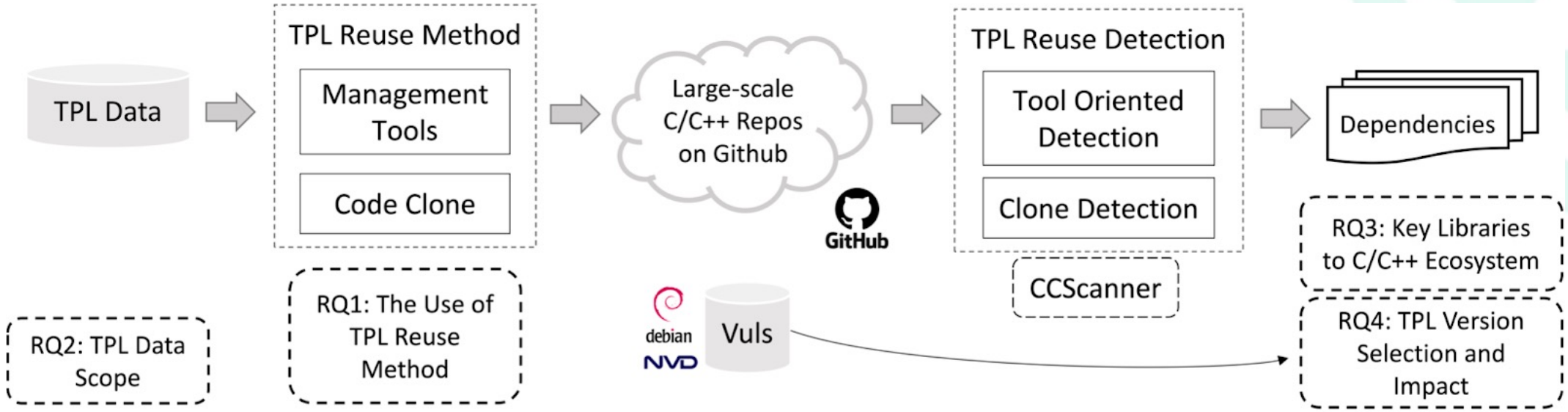
Figure 1: Overview of our work



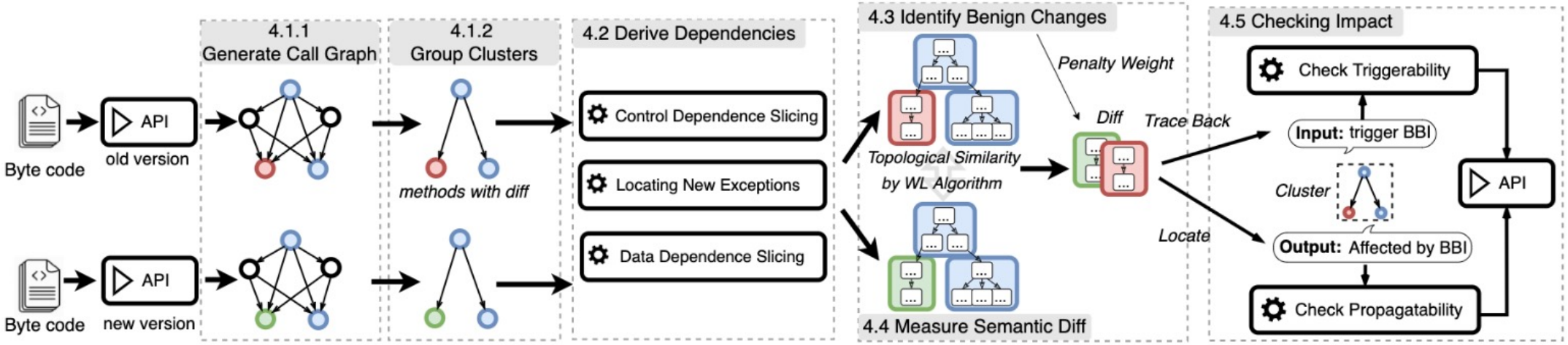




Towards Understanding Third-party Library Dependency in C/C++ Ecosystem



Has My Release Disobeyed Semantic Versioning? Static Detection Based on Semantic Differencing



Opportunity and Challenge co-exist

- Vulnerability Detection & Management
  - **Affected libraries and versions**
  - **Vulnerable code**
  - **Patches**
  - **POCs**
- Software Component Analysis (SCA)
  - **Package managers/Dependency resolutions**
  - **Code Clone**
  - **Variation of SCA tools**
- Vulnerability Impact Analysis
  - **Weak links**
  - **Fragile points**
  - **Alerts**
- Trustworthy OSS
  - **Trustworthy TPLs (Security, Quality)**
  - **Trustworthy Developers (Maintainers, Develop Teams)**

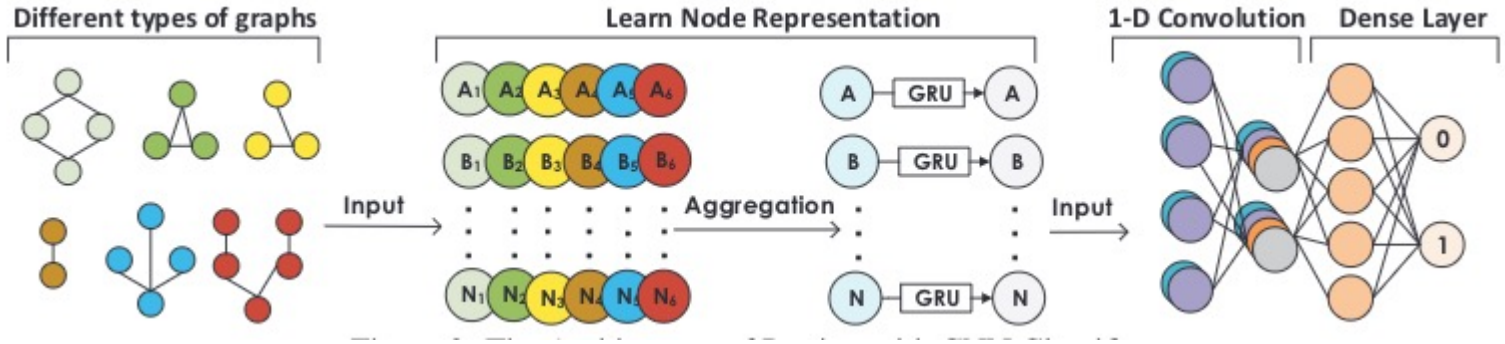
Vulnerability Analysis	• AI-Based Vulnerability Data Collection • Vulnerability Detection
SCA 2.0	• Software Compositional Analysis for Full SDLC • SAST Integration
Ecosystem Analysis	• Supply Chain Security • Vulnerability Impact Analysis • Ecosystem-wide Pre-warning
OSS Risk Analysis	• License Compliance / Conflict Detection • Development Risk Analysis • Malicious Maintainer Identification
OSS Governance	• Software Digitization Platform • OSS Health Profile– Osspert • OSS-Ops

Automated Vulnerabilities Identification via Deep Learning

1. Code Representation of func in joint graph structure

2. Learning Representation through graph neural networks

3. Graph-level classification to detect vulnerable functions



THE CHALLENGES OF OSS MANAGEMENT

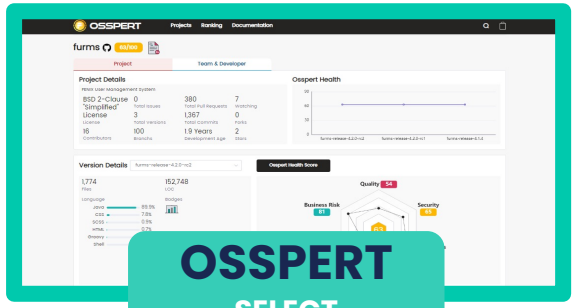
What are you using?
97 % of all applications use open source, with 267 open-source components on average

Is it vulnerable?
261 CVEs were disclosed in the last 7 days. These affect open source components that have been downloaded 63 million times.

Are there other risks?
Licensing risks, business continuity, maintainability, code quality and more

THE GAP BETWEEN DEVELOPMENT AND SECURITY IS GROWING

THE SOLUTION



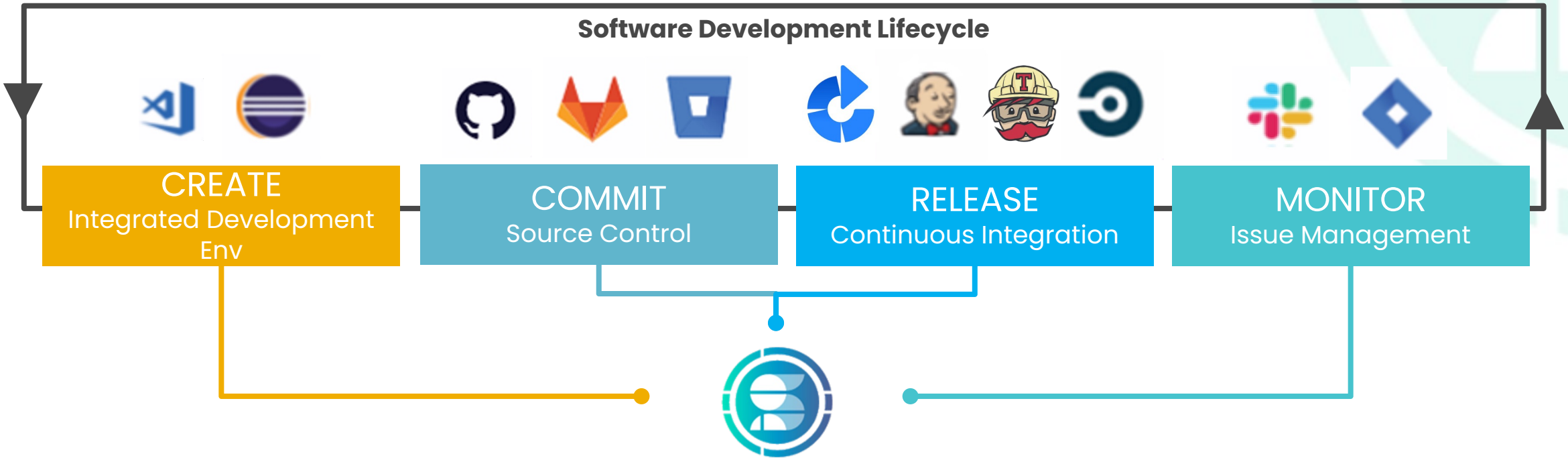
Holistic health assessment of open-source

Managed services to secure and deliver open-source

Detection and remediation of open-source vulnerabilities

END-TO-END OPEN-SOURCE RISK MANAGEMENT

SCA2.0： Software Compositional and Risk Analysis towards the full SDLC



From **source-code to binaries**, from **development to production** – Scantist solutions are built to ensure continuous security

OSS Health Profile—OSSPERT

Maintaining Open Source Software Health——Osspert.com

Comprehensive &Dynamic Analysis for open source projects

- Quality
- Security
- Activeness
- OSS
- Maintainability
- Business Risk
- Develop risk
- Popularity
- Usability
- Team Health

THANK YOU

yangliu@ntu.edu.sg

Try out at
scantist.io