Kakao 오픈소스 검증 시스템
OLIVE 소개

KAKAO
황민호, 김영환

오픈소스 관리 어떻게 하고 있나요?

Kakao 사용중인 오픈소스 : 약 2,800개 (2019년8월기준)

kakao

1 Open Source Management Program 시작

Kakao OSA(Open Source Advocate) 조직 신설

EXCEL 수동검증

EXCEL 수동검증

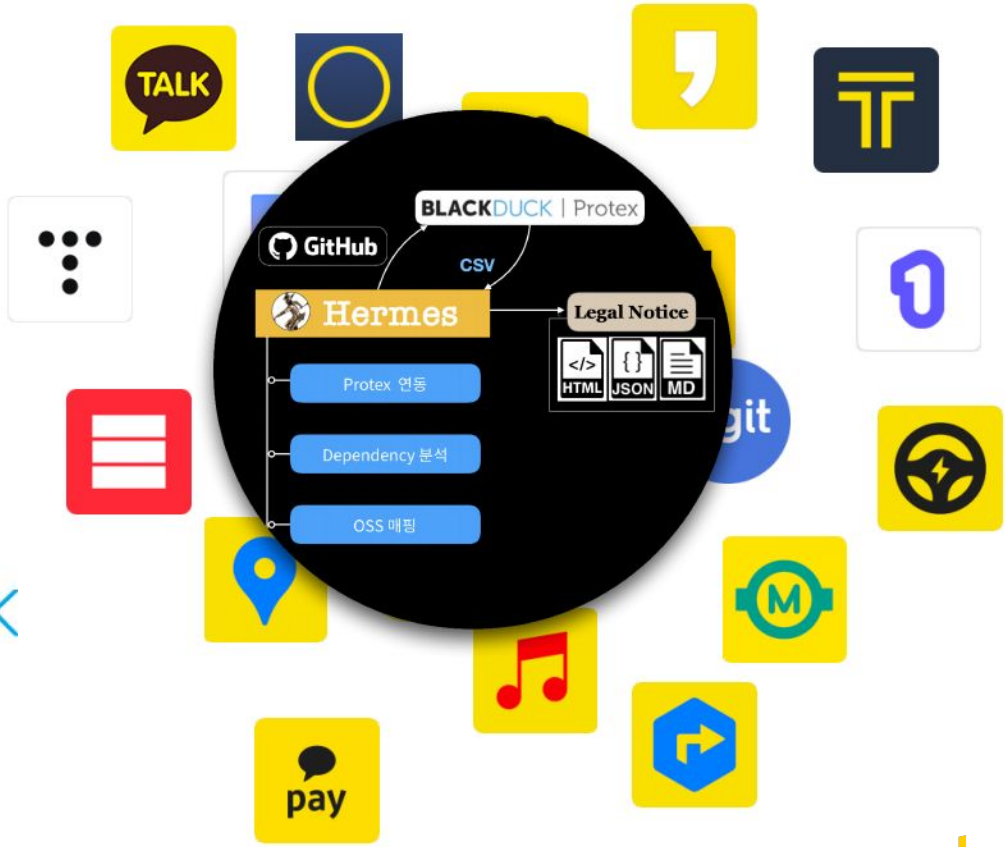
kakao

1 Open Source Management Program 시작

Hermes + Protex 오픈소스 검증

HERMES

BLACKDUCK



kakao

1 Open Source Management Program 시작

Hermes + Protex 오픈소스 검증

Analysis

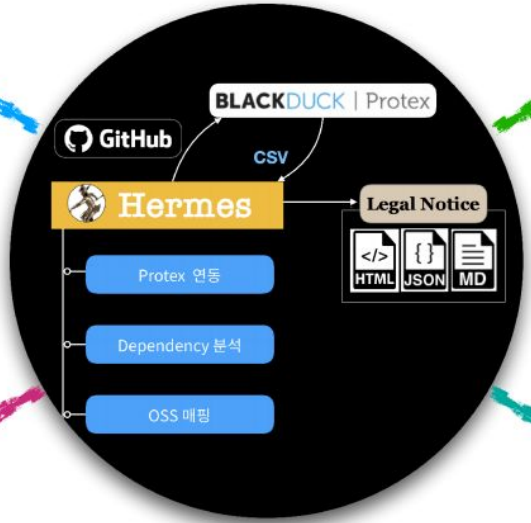
Dependency 설정 파일 분석

Protext 연동을 통한 소스코드 분석

Mapping

분석된 Dependency 와 오픈소스 매핑

이전 검증결과를 기반으로 한 자동 매핑



License

컴포넌트 별 라이선스 정보 식별

라이선스 의무사항 및 충돌 등 이슈 확인

Database

Open Source 정보 수집

SPDX 기반으로 License 데이터 구축

Project 검증내역 관리

kakao

1 Open Source Management Program 시작

개발 완료 → 오픈소스 검증 요청 → 검증 진행 → 리뷰 → 고지문 발급 → 배포

서비스 개발자 → 서비스 담당자 → OSA → 서비스 담당자 → OSA → 서비스 개발자

서비스 개발자

이 오픈소스 써도 되나요?

내일 배포인데요..

코드분석에 시간이 너무 오래 걸리는데..?

OSA에서 검증이 끝나야만 결과를 알 수 있나요?

헉헉.. 검증이 밀리고 있어.. 다음검증은 멀까?

수정사항이 생겼는데 검증을 다시 해야하나요?

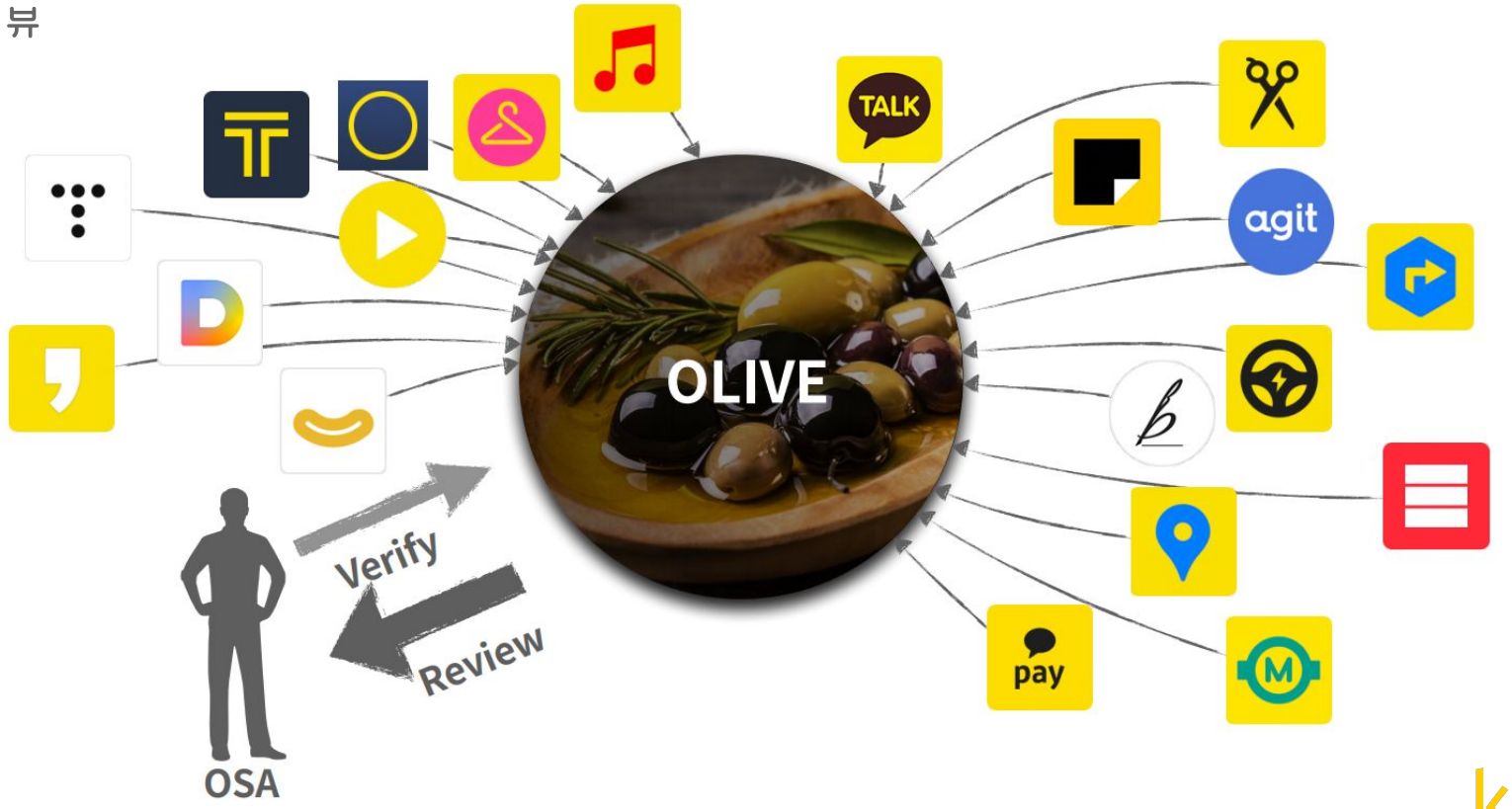
kakao

2 오픈소스 검증시스템 OLIVE

OLIVE 서비스 목표

2 오픈소스 검증시스템 OLIVE

OSA 리뷰



2 오픈소스 검증시스템 OLIVE

유서서비스 사례 조사

CodeEye (국내)

OLIS 에서 제공하는 오픈소스SW 라이선스 검사 서비스
웹 버젼과 Client 버전 두가지 형태로 검사 방식 제공
OLIS 회원가입 후 검사 계정 신청을 요청하고 관리자 승인 후 사용

Fossa

오픈소스 라이선스 Compliance 및 Security 서비스 제공
의존성 보고서 및 라이선스 고지문 생성
JS Foundation 오픈소스 라이선스 인증 제공 업체로 선정 (2018.04)

ScanCode

소스코드 및 바이너리 파일에서 라이선스, 저작권 및 종속성 등을 식별
Eclipse 재단, FSF, ClearlyDefined, Fabric8 등 프로젝트에서 사용
엔터프라이즈 솔루션으로 DejaCode 도 서비스 중

DependencyTrack

지능형 소프트웨어 공급망 구성 요소 분석 플랫폼
SBom(Software Bill-of-Materials) 기능 활용
API 기반 설계로 CI/CD 환경에 적합

WhiteSource

OLIS 에서 제공하는 오픈소스SW 라이선스 검사 서비스
웹 버젼과 Client 버전 두가지 형태로 검사 방식 제공
OLIS 회원가입 후 검사 계정 신청을 요청하고 관리자 승인 후 사용

BlackDuck (Synopsys)

코드 스캔방식 기반의 오픈소스 관리 SW
Snippet분석, meta-data 분석 등 Protex 기존 기능을 확장하여 제공
2,000,000 이상 프로젝트 데이터 베이스를 보유

VersionEye

보안취약점, 라이선스 위반 사항 및 버전 종속성 체크
일부 유료사용자가 있었으나 유지비용 문제 등으로 현재는 서비스 종료
Docker 이미지 및 소스코드등은 모두 오픈 되어 있음

Fossology

오픈소스 라이선스 준수 SW 시스템 및 툴킷
코드상의 라이선스 및 저작권을 스캔
정규표현식 기반의 스캔(Nomos)와 Text유사도 매칭기반의 스캔(Monk) 제공

kakao

2 오픈소스 검증시스템 OLIVE

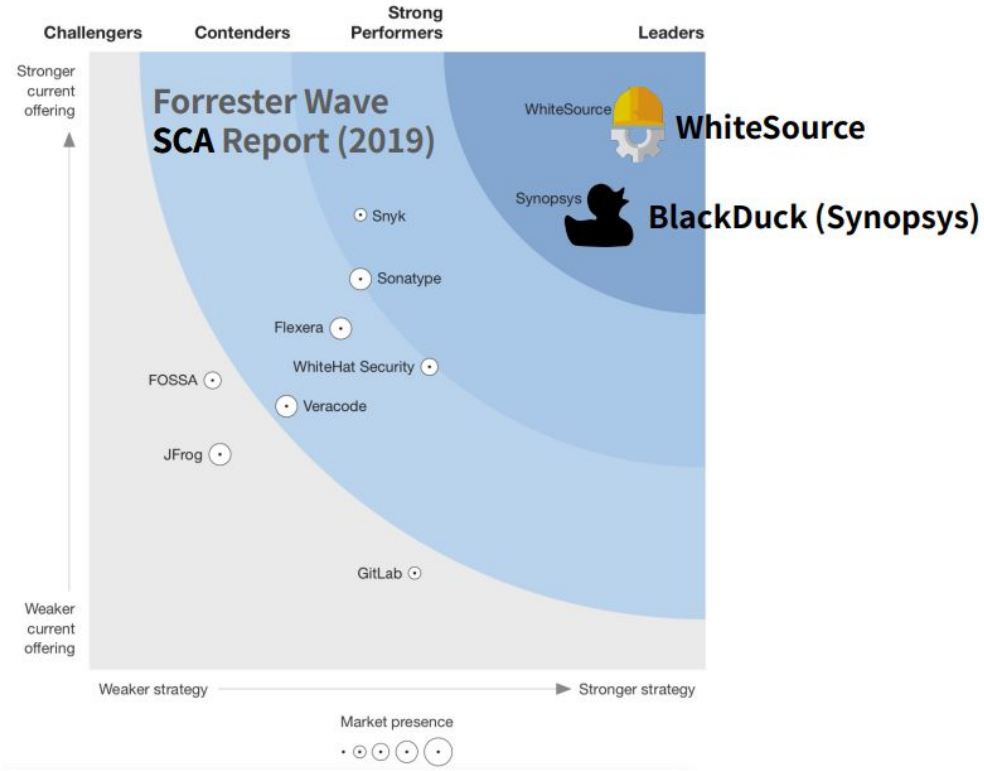
SCA (Software Composition Analysis)

사용자에게 공개 소스 인벤토리에 대한 가시성을 제공하는 일련의 도구에 대한 비교적 새로운 업계 용어

1 세대 : 오픈 소스 코드 스캔

2 세대 : 지속적인 오픈 소스 구성 요소 관리

3 세대 : 효과적인 사용 현황 분석



kakao

2 오픈소스 검증시스템 OLIVE Open Source License Identify Verify & Extract

GitHub

SourceCode

OLIVE

Notice

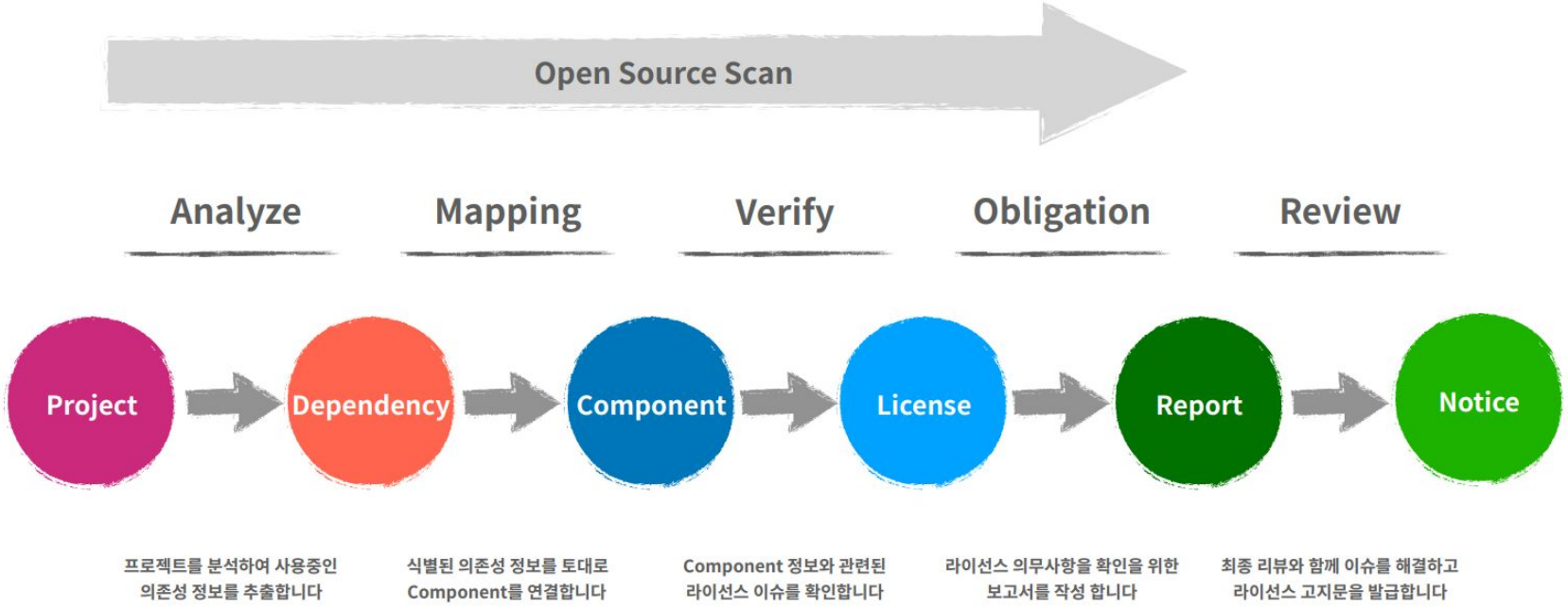
프로젝트를 정확히 분석

사용된 OSS Component를 식별

리뷰를 거친 후 고지문 발급

kakao

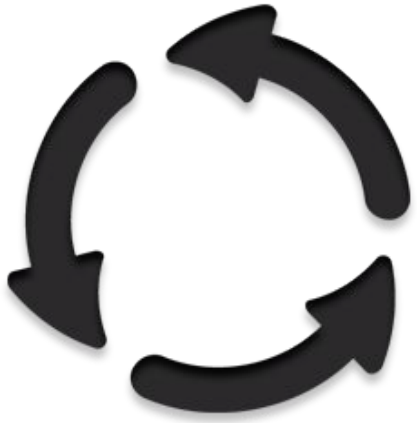
2 오픈소스 검증시스템 OLIVE



kakao

2 오픈소스 검증시스템 OLIVE

검증 주기



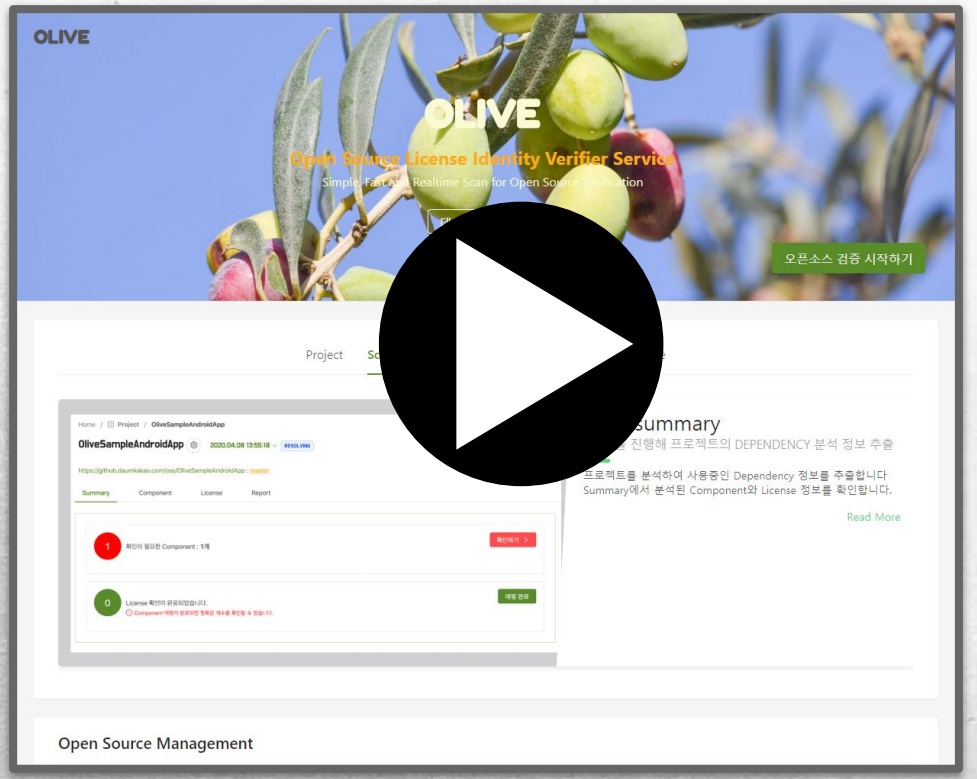
- 개발자가 검증 양식을 작성하여 Scan 요청시 (Agit)
- 관리대상 프로젝트 정기적으로 스캔 (4개월 / 6개월 / 12개월)
- 고지문 발급 이후 컴포넌트가 추가되는 경우 (Hotfix)
- 3rd Party 및 연결된 프로젝트의 Scan이 필요한 경우
- 오픈소스 프로젝트로 전환되는 경우
- 개발자가 오픈소스 확인을 필요로 하는 경우

2 오픈소스 검증시스템 OLIVE

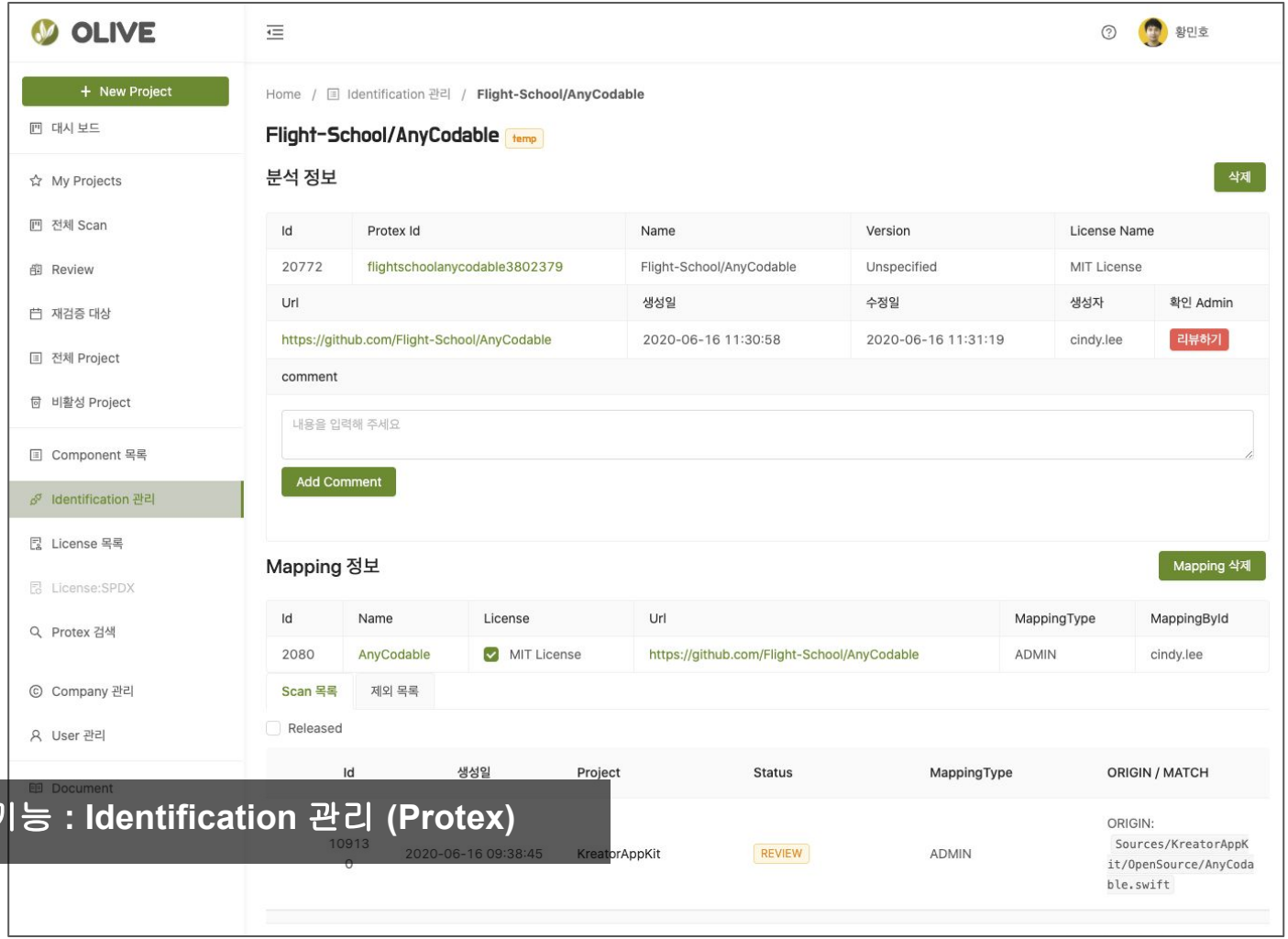
데모 시연

https://t1.daumcdn.net/osa/olive-docs/olive-demo.mp4

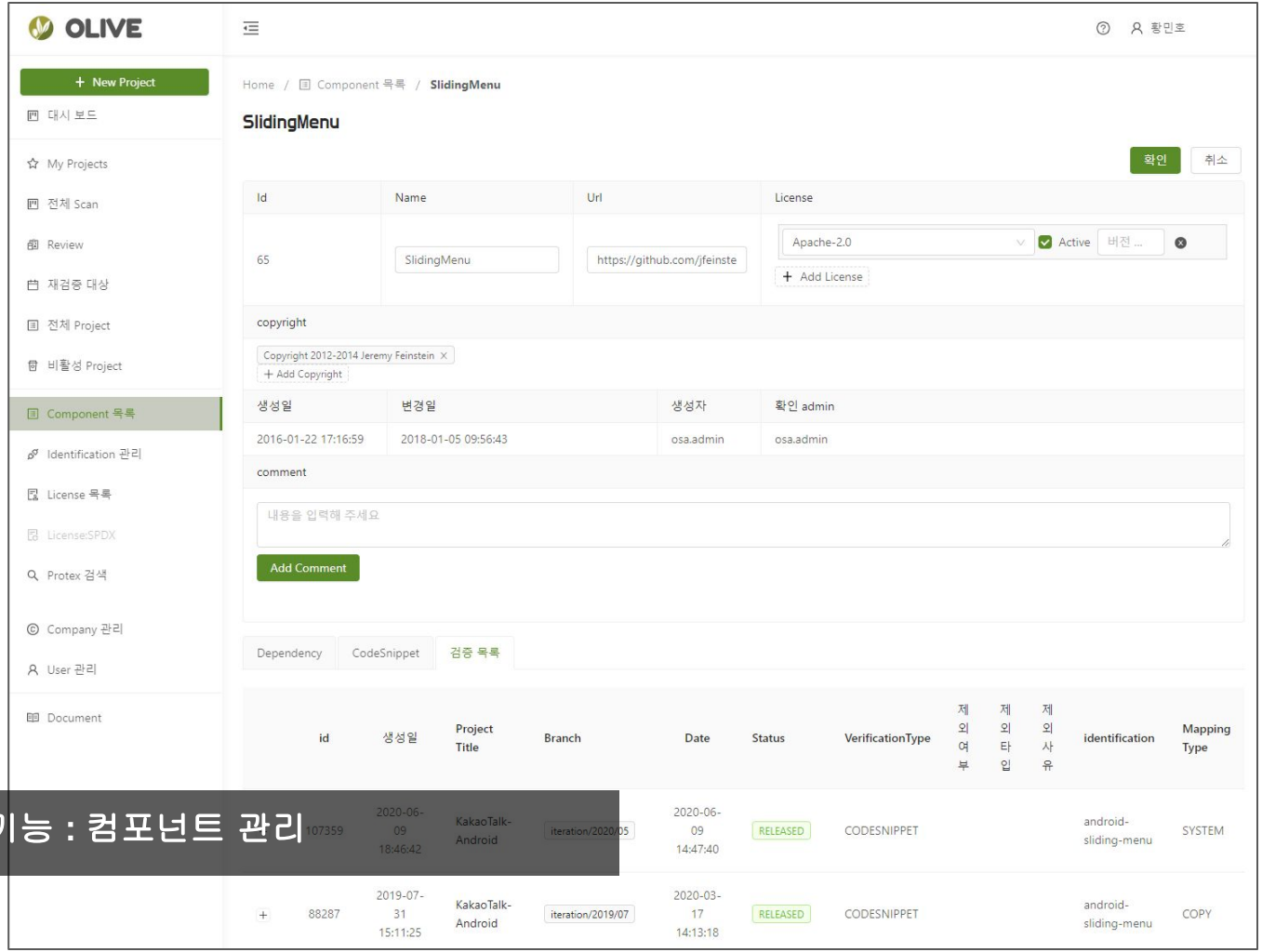
- 안내페이지
- 계정관리/로그인
- 내 프로젝트 목록
- 프로젝트 추가 (멀티 Repository 지원)
- Github 연동 후 Meta정보 조회
- 스캔
- 스캔 요약정보 조회
- 고지문 미리보기
- 컴포넌트 식별
- 컴포넌트 검색
- (임시) 컴포넌트 생성 : 이름, URL, 라이선스, Copyright
- 컴포넌트 맵핑
- 라이선스 조회
- 라이선스 이슈 처리
- 고지문 발급대상 요청 및 승인
- 프로젝트 리뷰 요청 및 리뷰
- 리뷰 결과 공유
- 담당자 확인
- 고지문 발급



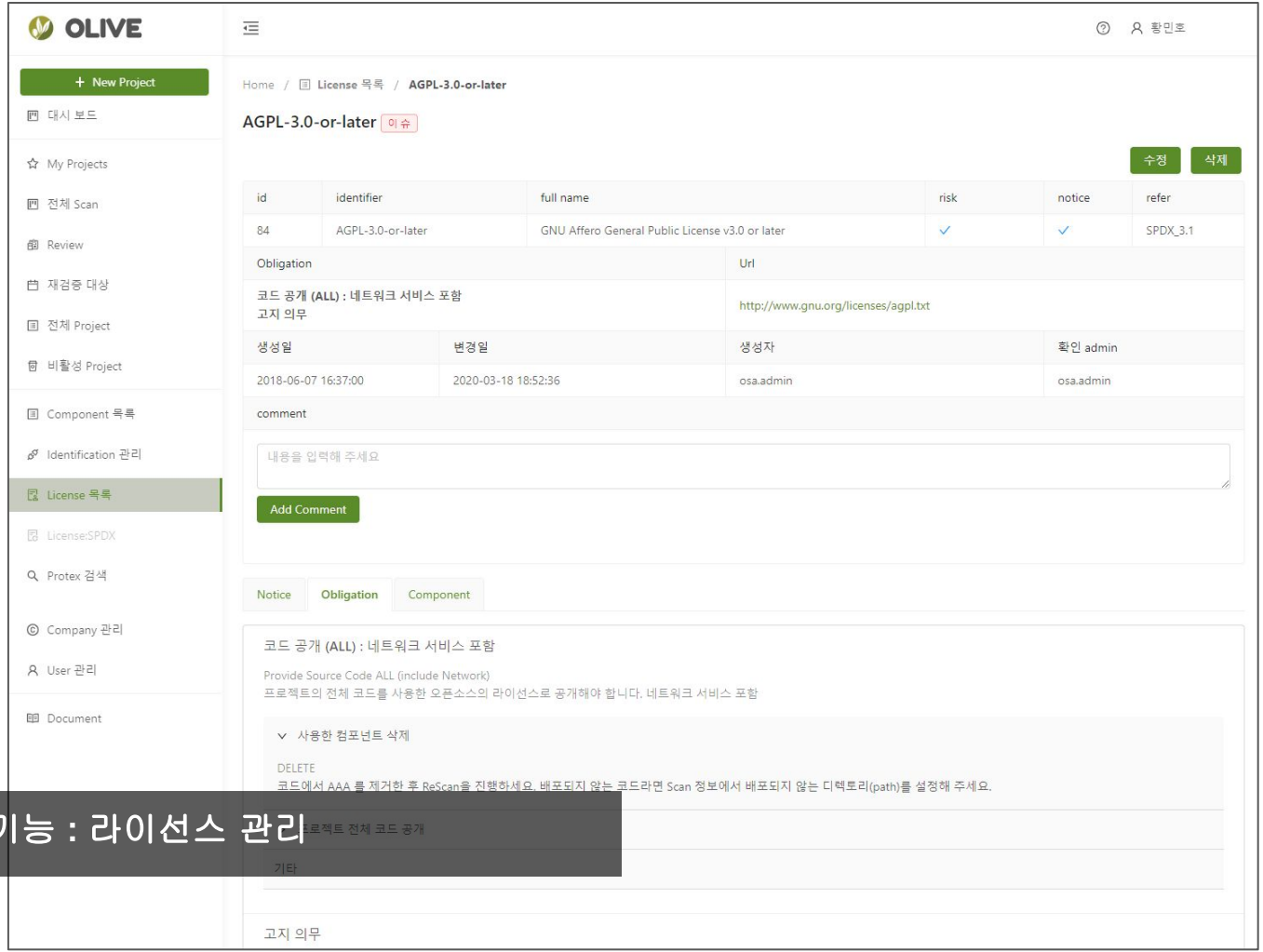
kakao



관리자 기능 : Identification 관리 (Protex)

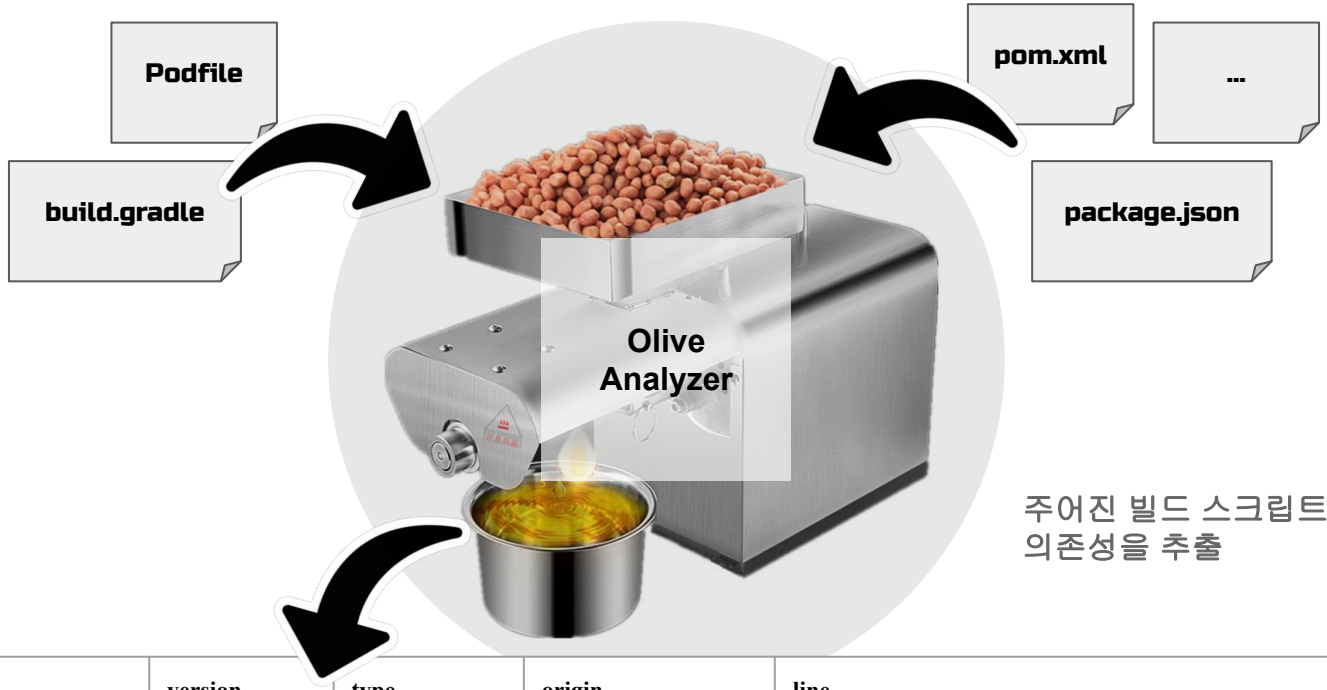


관리자 기능 : 컴포넌트 관리



관리자 기능 : 라이선스 관리

3 오픈소스 검증시스템 OLIVE 분석 모듈 : Olive Analyzer (Olive Oil)



주어진 빌드 스크립트 파일을 분석해서
의존성을 추출

artifact	version	type	origin	line
KakaoOpenSDK		COCOAPOD	Podfile	podfile ‘KakaoOpenSDK’
dagger	1.2.3	MAVEN	build.gradle	implementation “com.google.dagger:dagger:1.2.3”

kakao

3 오픈소스 검증시스템 OLIVE 분석 모듈 : 지원 Spec

파서/빌드 방식을 통해서 다양한 빌드 스크립트에 대한 분석 지원

파서(Parser) 방식

Maven	build.gradle, pom.xml
COCOAPODS	Podfile,Package.swift, Cartfile
node.js	package.json, package-lock.json
python	setup.py, requirements.txt
Ruby	Gemfile, Berksfile
Golang	Godeps.json, Gopkg.toml, Gopkg.lock
Native	CMakeList.txt, Android.mk
Etc	사용자 정의, .gitmodules 등

빌드(Build) 방식

gradle	build.gradle

kakao

3 오픈소스 검증시스템 OLIVE 분석 모듈 : 분석 예 - iOS

Podfile

```
target `MyApp` do
    use_frameworks!
    pod `Alamofire`, `~> 3.0`
end
```

target ‘MyApp’ do
⇒ Scope 정보 추출 ‘MyApp’

pod ‘Alamorefile’, ‘~> 3.0’
⇒ artifact(Alamorefile) 및 버전(~> 3.0) 정보 추출

kakao

3 오픈소스 검증시스템 OLIVE 분석 모듈 : 분석 예 - Java, Android

build.gradle

```
dependencies {
    implementation project("mylibrary")
    implementation fileTree(dir: `libs`, include: [`*.jar`])
    implementation `com.example.android:app-magic:12.3`
}
```

의존성 추출 시작 지점 확인

mylibrary 프로젝트 분석

libs 파일 하위의 jar 파일 분석

group(com.example.android), artifact(app-magic) 및 버전 (12.3) 정보 추출

참고: local library 및 binaries의 경우 분석 제외
(파일 검색으로 처리)

kakao

3 오픈소스 검증시스템 OLIVE 분석 모듈 : 빌드 방식의 필요성

Parser 방식의 한계

```
dependencies {

        implementation d.aaa
        debugImplementation d.bbb
}
```

Parsing ERROR

이런 경우에도 **Analyzer**는 분석 결과를 만들어 줘야 한다.

```
dependencies {

        implementation d.aaa
        implementation d.bbb
}
```

Gradle Build

```
implementation - Implementation only dependencies for
source set 'main'. (n)
+--- org.developer:android:1.7.26
+--- org.apache:executer:2.11.0
```

kakao

4 오픈소스 검증시스템 OLIVE 로드맵

오픈소스 프로젝트로 공개

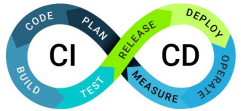
누구나 쉽게 사용할 수 있는 분석 라이브러리 제공

다양한 파서/빌드 시스템 추가 지원

gradle이외 python, swift 등 추가 빌드 방식 지원

분석 정확도 향상

Groovy, Kotlin 등 추가 언어 분석 하는 못하는 경우 개선



CI/CD 연동

Github 연동을 통해 PR 반영 시점에 자동으로 검증을 수행하고 결과를 공유

kakao

4 오픈소스 검증시스템 OLIVE 로드맵

kakao