

# OLIVE CLI 소개

---

kakao

2022.05.25.(수)

오픈소스기술파트  
이기문 ricky.moon

---

# CONTENTS

- OLIVE Platform 소개
- OLIVE CLI 개요
- OLIVE CLI 명령어
- OLIVE CLI 적용(예정) 사례
- 로드맵
- Q & A



## OLIVE Platform 소개

## 쉽고, 빠르고, 정확한 오픈소스 검증 서비스



### 분석

#### 프로젝트 실시간 스캔

GitHub 연동과 파일 업로드의 2가지 방식으로 프로젝트를 추가 및 스캔할 수 있으며, CLI 도구도 지원하고 있습니다.



### 데이터 등록 및 공유

#### 오픈소스 데이터 등록 및 공유

오픈소스 데이터를 등록하고 DB에 추가해 추후의 자동 분석에 재활용할 수 있습니다.



### 검증

#### 빠르고 정확한 오픈소스 검증

신뢰할 수 있는 오픈소스 DB를 바탕으로 빠르고 정확하게 오픈소스를 검증할 수 있습니다.

2021년 6월 29일 [OLIVE Platform](#) 정식 오픈



체험하기

build.gradle, podfile과 같은 패키지 매니저 파일을 업로드해서 사용한 오픈소스 확인



## OLIVE CLI 개요

```

  OLIVE CLI

Usage:
olive-cli [-hV] [COMMAND]

Description:
OLIVE CLI

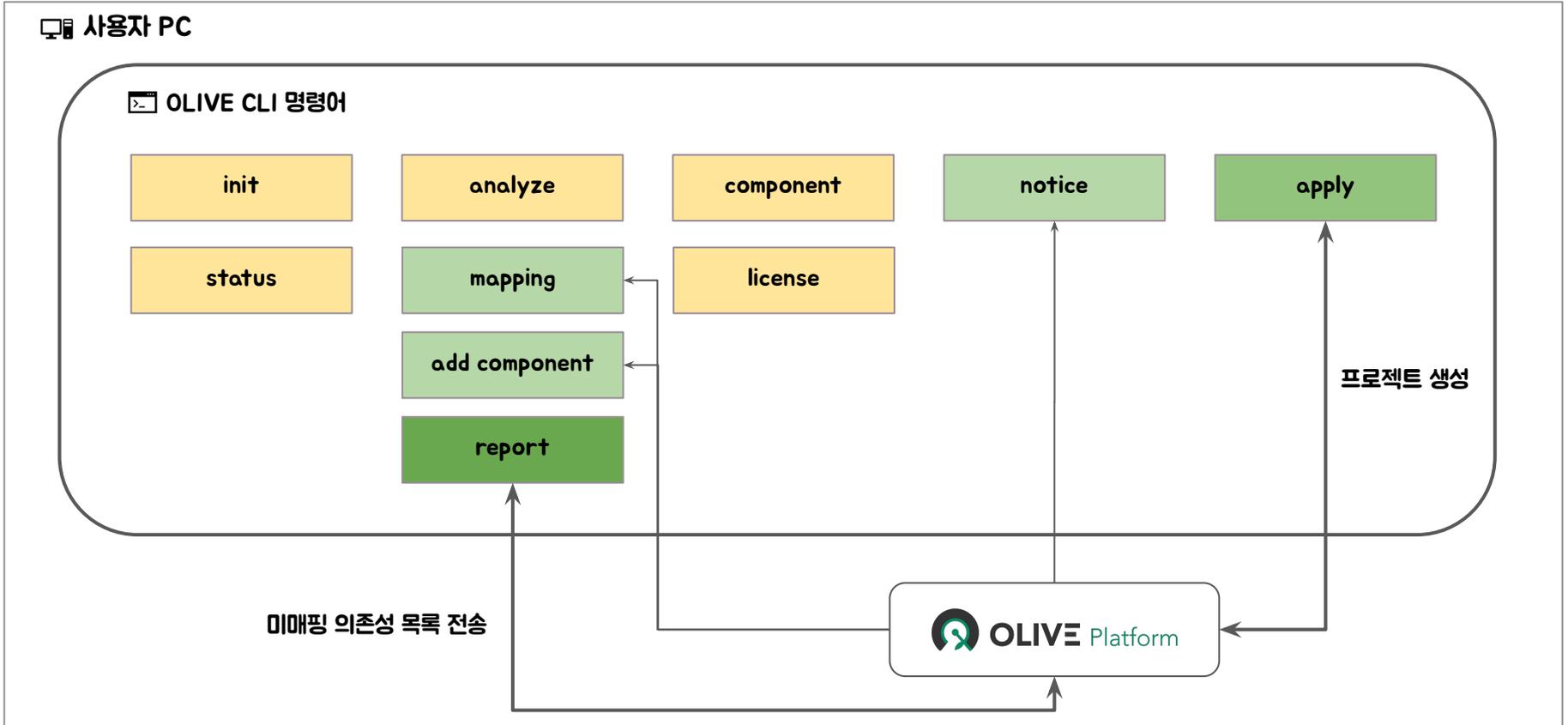
Options:
-h, --help      Show this help message and exit.
-V, --version   Print version information and exit.

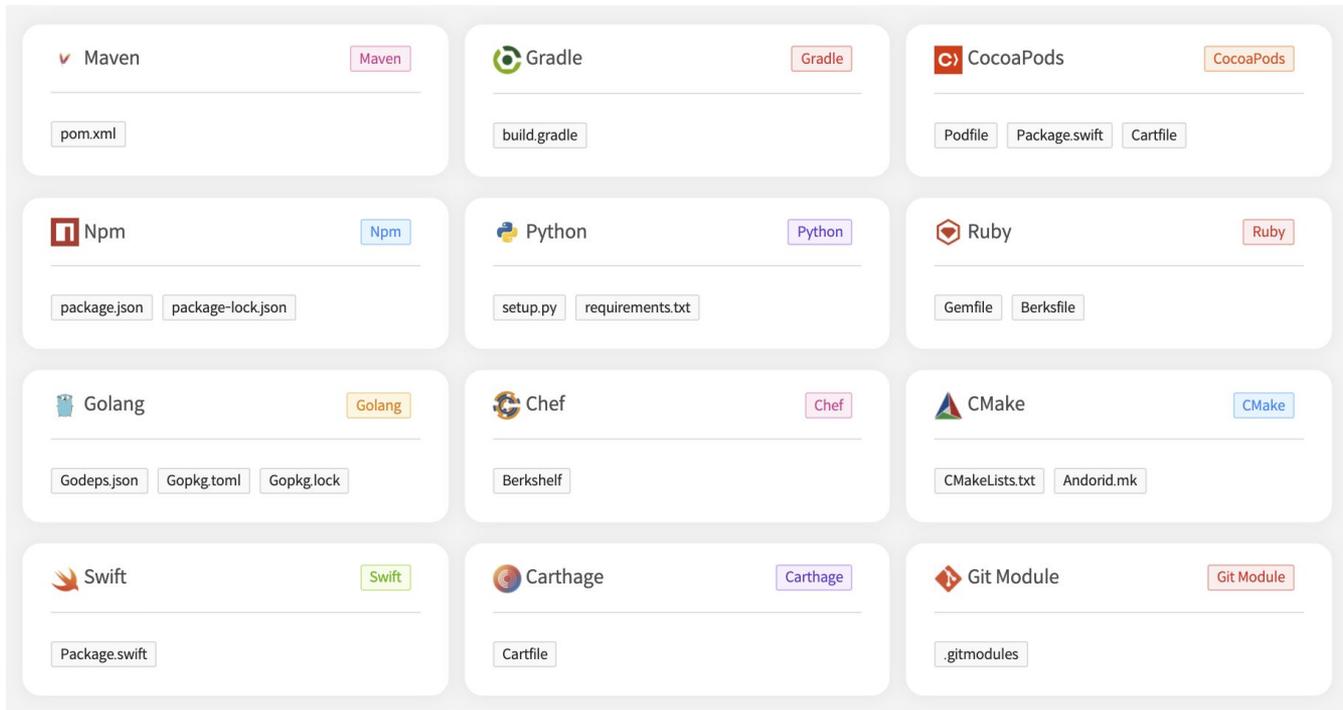
Commands:
init           Initialize olive-cli configuration styles
analyze       Analyzing the dependencies used in project
mapping       Mapping components based on analysis results
component     Shows the component list
license       Shows the license list
add           Add manual things in scan
report        Report the analysis results.
apply         Apply the results of analysis.
notice        Download the preview of opensource software notice.
status        Shows the current status
help          Displays help information about the specified command

```

사용자 PC 환경에서 OLIVE Platform 기능을 실행할 수 있도록 돕는 Command Line Interface

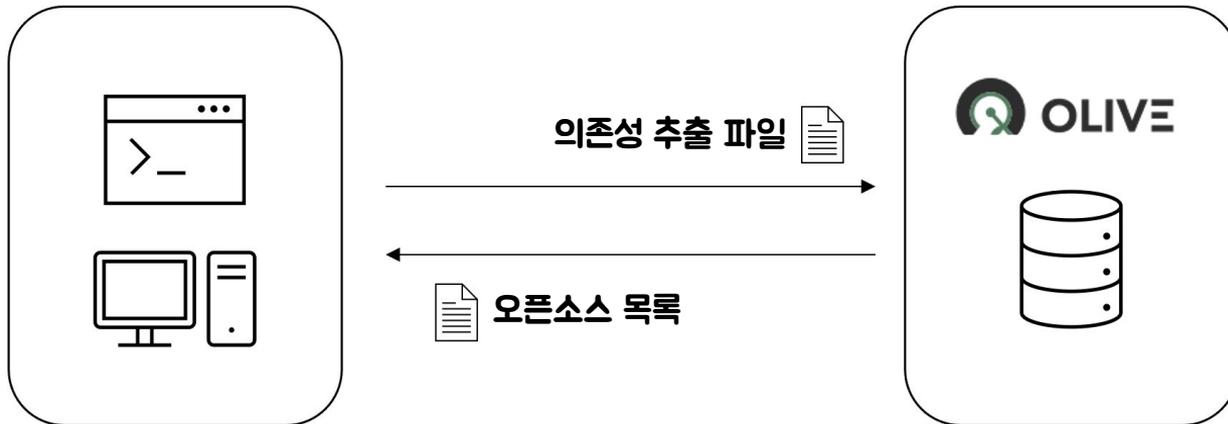
OLIVE CLI 명령어는 10개로 구성돼 있으며, 이 중 OLIVE API를 호출하는 명령어는 5개입니다.





OLIVE Platform에서 분석 지원하는 파일 목록을 동일하게 지원

OLIVE CLI는 보안상 웹 서비스 사용이 어렵거나, 소스코드 노출이 우려되는 경우 안심하고 사용할 수 있습니다.



- 서버에 소스코드가 전송되지 않음
- VCS(Version Control System)에 종속되지 않고 프로젝트를 사용자 PC 환경에서 분석
- 분석한 결과를 사용자 PC에서 별도 파일로 관리
- 명령어를 이용해 분석된 결과를 업로드하여 OLIVE Platform 프로젝트로 관리
- 특정 명령어(report 및 apply)를 실행하지 않으면 클라이언트 측의 어떤 데이터도 서버에 저장되지 않음



## OLIVE CLI 명령어

다음 명령어는 OLIVE Platform API 호출 없이 독립적으로 실행 가능합니다.

명령어	설명
init	OLIVE CLI 사용을 위한 초기 설정을 진행하며, config.yaml 설정파일이 생성됩니다.
analyze	config.yaml 설정파일 기반으로 의존성 분석을 시작합니다. 분석이 완료되면 dependency.json, dependency.csv 파일이 생성됩니다.
component	프로젝트가 사용한 컴포넌트를 조회합니다. 조회가 완료되면 component.json, component.csv 파일이 생성됩니다.
license	프로젝트가 사용한 컴포넌트의 라이선스를 조회합니다. 조회가 완료되면 license.json, license.csv 파일이 생성됩니다.
status	현재 프로젝트 상태를 조회할 수 있습니다.

다음 명령어는 OLIVE Platform API 호출이 필요합니다. 방화벽이 있다면 아웃바운드 설정을 추가해야 합니다.

명령어	설명	OLIVE API
mapping	analyze 명령어로 분석된 의존성을 컴포넌트에 매핑합니다. 매핑이 완료되면 mapping.json, mapping.csv, unmapping.csv 파일이 생성됩니다.	/api/v1/cli/mapping
add component	사용자가 수동으로 컴포넌트를 추가합니다.	/api/vi/cli/components
report	매핑이 되지 않은 의존성 목록을 관리자에게 전송합니다. 관리자가 해당 데이터를 리뷰하고 mapping 명령어를 다시 수행하여 컴포넌트에 매핑할 수 있습니다.	/api/vi/cli/report
apply	CLI 분석한 결과를 토대로 OLIVE Platform에 프로젝트를 생성합니다.	/api/vi/cli/apply
notice	프로젝트가 사용한 컴포넌트를 기반으로 고지문 미리보기를 생성합니다.	/api/vi/cli/notices

OLIVE CLI를 사용하기 위해 초기 설정을 하는 명령어

```
$ olive-cli init <title> --token=<token> [options] [--debug|-d] [--help|-h] [--force]
```

옵션	단축어	필수 여부	설명
-- token	-t	Y	OLIVE Platform 연동 시 필요한 API 토큰입니다.
-- source	-s	N	CLI 명령어 수행 대상 프로젝트 경로로 지정하지 않으면 현재 위치가 기준이 됩니다.
-- force	-f	N	기존 스캔을 모두 삭제하고 CLI를 재설정합니다.
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

init 명령어 실행 시 basePath(기본값: .olive) 하위에 config.yaml 파일이 생성됩니다.

### config.yaml 파일 예시

```

---
version: "v1"
projectTitle: "hello-world"
projectHash: "VDLLPU8XAP"
cliInfo:
  sourcePath: "/workspace/hello-world"
  basePath: "/workspace/hello-world/.olive"
  apiToken: "{token}"
  server: "https://olive-api.kakao.com"
  host: "https://olive.kakao.com"
scanInfo:
  revisionId: "1"
  hash: "1nZPdhxAJ"
  status: "MAPPED"
  resultPath: "/workspace/hello-world/.olive/1"
  executed: "2022-04-21T24:00:00.124245"
userConfig:
  gradleBuildVariant: ""
  excludeGradle: []
  isOpensource: "false"
  excludePaths:
    - "mvn"
# 설정파일 버전
# 프로젝트 타이틀
# OLIVE Platform에 생성된 프로젝트 해쉬값
# CLI 설정
# 명령어를 수행할 프로젝트 경로(기본값: init 명령어 수행 디렉토리)
# 결과가 저장되는 기본 경로(기본값: sourcePath 하위 '.olive')
# OLIVE Platform과 연동을 위한 API 토큰
# 호출 API 서버 주소
# OLIVE Platform 주소
# 스캔 수행 정보
# 스캔 revision ID
# 스캔 해쉬값
# 스캔 상태
# 명령어 결과 저장 경로
# 명령어 수행 시간
# 사용자 설정
# 안드로이드 빌드 시 필요한 Build Variant
# 스캔에서 제외할 Gradle
# 프로젝트를 오픈소스로 공개할 것인지 여부(기본값: false)
# 스캔 제외 경로

```

프로젝트를 스캔하여 사용한 의존성을 추출하며 다음의 2가지 파일로 저장됩니다.

- dependency.json: 분석된 모든 데이터의 모든 필드
- dependency.csv: 분석된 모든 데이터의 정제된 필드

```
$ olive-cli analyze [--config=<config_path>] [--debug|-d] [--help|-h]
```

옵션	단축어	필수 여부	설명
-- config	-c	N	설정파일의 경로(path) 지정
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

analyze 명령어를 실행하면 콘솔창에 프로젝트에서 사용한 의존성을 보여줍니다.

```
Dependencies are analyzed successfully. Detecting [2] dependency types.
Dependency report was saved as a json file at [path:/Users/ricky/Krew/OSA/CLI 테스트/sample-app-for-android-main/.olive/1/dependency.json].
[2] dependency types are detected:
  [3] dependencies of [BINARY]
  [9] dependencies of [GRADLE]
```

```
=====
Dependency Type: BINARY(3)
=====
```

ID	GroupID	ArtifactID	Version	Scope	Path
1		kakao-olive-camera	1.3.5		app/libs/kakao-olive-camera-1.3.5.jar
2		kakao-olive-side-menu	1.22.2		app/libs/kakao-olive-side-menu-1.22.2.jar
3		kakao-olive-super	1.5.6		app/libs/kakao-olive-super-1.5.6.jar

```
=====
Dependency Type: GRADLE(9)
=====
```

ID	GroupID	ArtifactID	Version	Scope	Path
1	androidx.appcompat	appcompat	1.2.0	implementation	app/build.gradle
2	androidx.constraintlayout	constraintlayout	2.0.4	implementation	app/build.gradle
3	androidx.test.espresso	espresso-core	3.3.0	androidTestImplementation	app/build.gradle
4	com.android.tools.build	gradle	3.5.2	classpath	build.gradle
5	junit	junit	4.12	testImplementation	app/build.gradle
6	androidx.test.ext	junit	1.1.0	androidTestImplementation	app/build.gradle
7	com.google.android.material	material	1.0.0	implementation	app/build.gradle
8	com.squareup.picasso	picasso	2.71828	implementation	app/build.gradle
9	org.whispersystems	signal-protocol-android	2.3.0	implementation	app/build.gradle

```
=====
```

추출한 의존성은 파일로 저장되어 관리가 용이합니다.

```

}, {
  "dependency" : {
    "revisionInfo" : {
      "revision" : "",
      "branch" : "",
      "empty" : true
    },
    "extraInfo" : {
      "repository" : "",
      "domain" : "",
      "empty" : true,
      "extension" : ""
    },
    "spec" : {
      "artifact" : "picasso",
      "scope" : "implementation",
      "license" : "",
      "group" : "com.squareup.picasso",
      "version" : "2.71828",
      "empty" : false
    },
    "empty" : false
  },
  "pos" : 27,
  "line" : "implementation 'com.squareup.picasso:picasso:2.71828'",
  "path" : "app/build.gradle",
  "message" : "",
  "state" : "SUCCESS",
  "attribute" : {
    "packageManagerType" : "MAVEN",
    "dependencyType" : "GRADLE",
    "analyzerType" : "PARSER"
  }
}, {

```

dependency.json

dependency					
dependencyType	groupId	artifactId	version	scope	path
BINARY		kakao-olive-super	1.5.6		app/libs/kakao-olive-super-1.5.6.jar
BINARY		kakao-olive-camera	1.3.5		app/libs/kakao-olive-camera-1.3.5.jar
BINARY		kakao-olive-side-menu	1.22.2		app/libs/kakao-olive-side-menu-1.22.2.jar
GRADLE	androidx.appcompat	appcompat	1.2.0	implementation	app/build.gradle
GRADLE	androidx.constraintlayout	constraintlayout	2.0.4	implementation	app/build.gradle
GRADLE	com.google.android.material	material	1.0.0	implementation	app/build.gradle
GRADLE	com.squareup.picasso	picasso	2.71828	implementation	app/build.gradle
GRADLE	junit	junit	4.12	testImplementation	app/build.gradle
GRADLE	androidx.test.ext	junit	1.1.0	androidTestImplementation	app/build.gradle
GRADLE	androidx.test.espresso	espresso-core	3.3.0	androidTestImplementation	app/build.gradle
GRADLE	org.whispersystems	signal-protocol-android	2.3.0	implementation	app/build.gradle
GRADLE	com.android.tools.build	gradle	3.5.2	classpath	build.gradle

dependency.csv

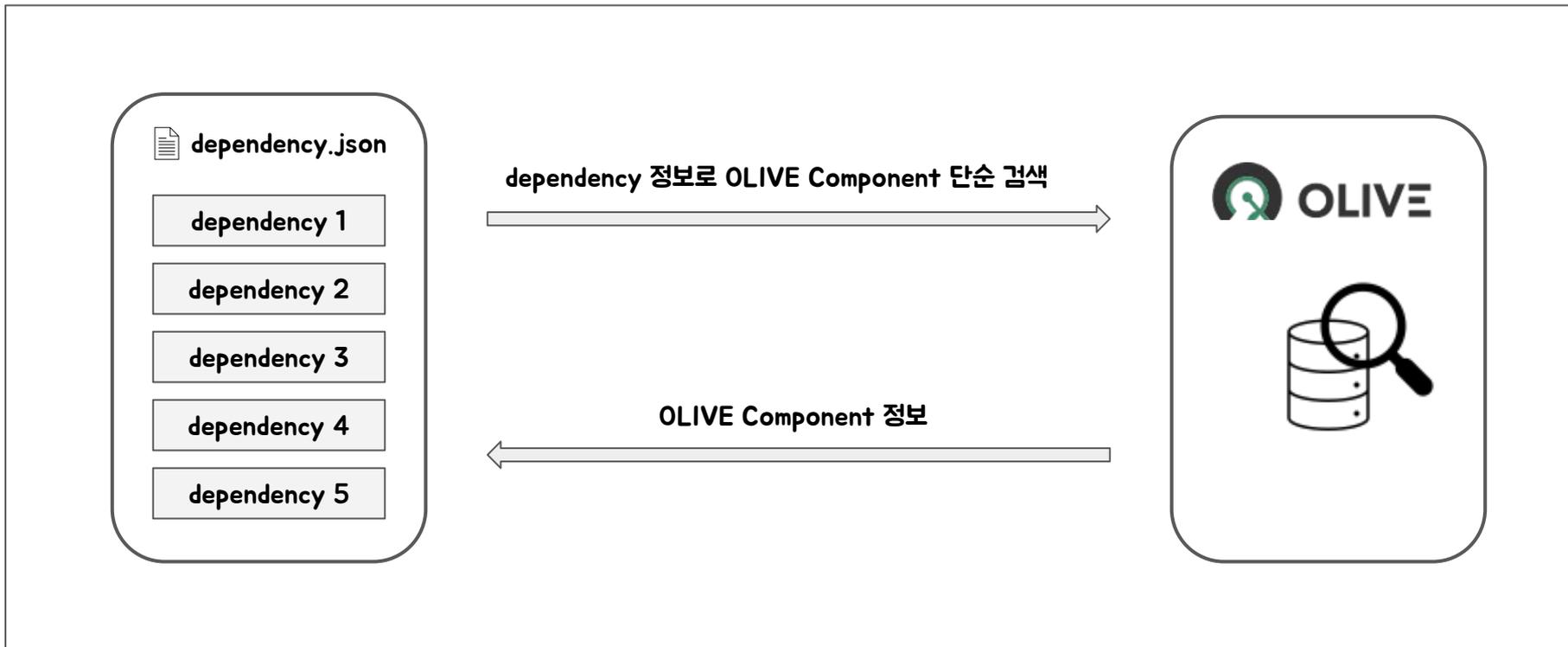
analyze 명령어를 통해 분석 및 추출한 의존성을 OLIVE 컴포넌트에 매핑하여, 다음의 3개 파일로 저장됩니다.

- mapping.json: 모든 데이터(매핑 및 매핑되지 않은 데이터 전체)의 모든 필드
- mapping.csv: 매핑된 모든 데이터의 정제된 필드
- unmapping.csv: 매핑된 모든 데이터의 정제된 필드

```
$ olive-cli mapping [--config=<config_path>] [--debug|-d] [--help|-h]
```

옵션	단축어	필수 여부	설명
-- config	-c	N	설정파일의 경로(path) 지정
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

OLIVE Platform API를 호출하지만, request로 전달된 어떠한 값도 저장하지 않습니다.



mapping 명령어를 실행하면 콘솔창에 프로젝트에서 사용한 컴포넌트를 보여줍니다.

- 줄임표로 생략된 부분은 json 파일과 csv 파일에서 확인 가능합니다.

```
Starting to map [3] dependencies of [BINARY] type...
[3] dependencies of [BINARY] are mapped successfully.
Starting to map [9] dependencies of [GRADLE] type...
[5] dependencies of [GRADLE] are mapped successfully.

=====
Mapping Components: [5]
=====
ID | Type | Name | Url | License | Copyright
---|-----|-----|-----|-----|-----
1 | GLOBAL | Android - platform - frameworks - support | https://android.googlesource.com/platform/frameworks/support | Apache License 2.0 | Copyright 2014 The Android Open Sourc...
2 | GLOBAL | Android Constraint Layout Library | https://developer.android.com/reference/android/support/constraint/packages | Apache License 2.0 | Copyright 2017 The Android Open Sourc...
3 | GLOBAL | Android Material Components | https://developer.android.com/reference/com/google/android/material/packages | Apache License 2.0 | Copyright 2018 The Android Open Sourc...
4 | GLOBAL | libsignal-protocol-java | https://github.com/signalapp/libsignal-protocol-java | GNU General Public License v3.0 or later | Copyright 2013-2019 Open Whisper Systems
5 | GLOBAL | Square Picasso | https://github.com/square/picasso | Apache License 2.0 | Copyright 2013 Square, Inc

=====
Unmapping Dependencies: [3]
=====
ID | Count | PackageManager | GroupID | ArtifactID | Version | Scope | Path | Line
---|-----|-----|-----|-----|-----|-----|-----|-----
1 | 1 | LIBRARY | | kakao-olive-camera | 1.3.5 | | ...akao-olive-camera-1.3.5.jar |
2 | 1 | LIBRARY | | kakao-olive-side-menu | 1.22.2 | | ...-olive-side-menu-1.22.2.jar |
3 | 1 | LIBRARY | | kakao-olive-super | 1.5.6 | | ...kakao-olive-super-1.5.6.jar |
=====
```

매핑된 컴포넌트와 미매핑된 의존성은 파일로 저장되어 관리가 용이합니다.

```

"component" : {
  "componentLicenses" : [ {
    "versionCondition" : "",
    "license" : {
      "fullName" : "Apache License 2.0",
      "identifier" : "Apache-2.0",
      "obligations" : [ {
        "description" : "Open-source licenses you use are subject to notices for any copies or works.",
        "guides" : [ ],
        "code" : "License Notice",
        "id" : 1,
        "name" : "License notices",
        "isIssued" : false
      } ],
      "id" : 64,
      "url" : "http://www.apache.org/licenses/LICENSE-2.0",
      "isReviewed" : true,
      "isFavorite" : true,
      "isIssued" : false,
      "isNotice" : true,
      "isSPDX" : true
    },
    "active" : true,
    "id" : 2017
  } ],
  "copyrights" : [ "Copyright 2018 The Android Open Source Project" ],
  "sourceUrl" : null,
  "componentType" : "GLOBAL",
  "gitHubInfo" : null,
  "representLicense" : {
    "fullName" : "Apache License 2.0",
    "identifier" : "Apache-2.0",
    "obligations" : [ {
      "description" : "Open-source licenses you use are subject to notices for any copies or works.",
      "guides" : [ ],
      "code" : "License Notice",
      "id" : 1,
      "name" : "License notices",
      "isIssued" : false
    } ],
    "id" : 64,
    "url" : "https://www.apache.org/licenses/LICENSE-2.0",
    "isReviewed" : true,
    "isFavorite" : true,
    "isIssued" : false,
    "isNotice" : true,
    "isSPDX" : true
  },
  "id" : 2015,
  "url" : "https://developer.android.com/reference/com/google/android/material/packages",
  "name" : "Android Material Components",
  "isReviewed" : true
},

```

mapping.json

mapping					
index	name	url	componentType	componentLicenses	copyright
1	Android Constraint Layout Library	https://developer.android.com/re	GLOBAL	Apache License 2.0	Copyright 2017 The Android Open Source Project
2	Android Material Components	https://developer.android.com/re	GLOBAL	Apache License 2.0	Copyright 2018 The Android Open Source Project
3	Square Picasso	https://github.com/square/picass	GLOBAL	Apache License 2.0	Copyright 2013 Square, Inc
4	Android - platform - frameworks -	https://android.googlesource.co	GLOBAL	Apache License 2.0	Copyright 2014 The Android Open Source Project
5	libsignal-protocol-java	https://github.com/signalapp/lib	GLOBAL	GNU General Public Licen	Copyright 2013-2019 Open Whisper Systems

mapping.csv

unmapping								
index	count	packageManager	groupId	artifactId	version	scope	path	line
1	1	LIBRARY		kakao-olive-side-menu	1.22.2		app/libs/kakao-olive-side-menu-1.22.2.jar	
2	1	LIBRARY		kakao-olive-camera	1.3.5		app/libs/kakao-olive-camera-1.3.5.jar	
3	1	LIBRARY		kakao-olive-super	1.5.6		app/libs/kakao-olive-super-1.5.6.jar	

unmapping.csv

사용자가 수동으로 컴포넌트를 추가합니다.

- basePath(기본값: .olive) 하위의 manual-component.yaml 파일에 컴포넌트 정보를 입력하여 추가합니다.
- OLIVE 컴포넌트를 검색하여 추가하거나 컴포넌트 정보를 생성해 새로운 컴포넌트를 추가할 수 있습니다.

```
$ olive-cli add component [--config=<config_path>] [--debug] [--help]
```

옵션	단축어	필수 여부	설명
-- config	-c	N	설정파일의 경로(path) 지정
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

## 컴포넌트 검색 후 추가

- OLIVE Platform의 컴포넌트 메뉴에서 추가하고자 하는 컴포넌트를 찾아 URL을 입력합니다.

```
manual-components:  
- path: "src/extra/custom.java"          # 컴포넌트를 사용한 위치  
  url: "https://github.com/kakao/custom-lib" # 추가하고자 하는 컴포넌트 URL
```

## 로컬 컴포넌트를 생성해 추가

- 컴포넌트 정보를 직접 입력합니다.
- 라이선스는 OLIVE Platform의 라이선스 메뉴에서 라이선스를 검색하여 해당 이름을 입력합니다.

```
manual-components:  
- path: "src/extra/custom.java"          # 컴포넌트를 사용한 위치  
  url: "https://github.com/kakao/custom-lib" # 추가하고자 하는 컴포넌트 URL  
create-data:  
  name: "custom-lib"                    # 컴포넌트 이름  
  license: "MIT"                        # 컴포넌트 라이선스  
  copyrights: ["kakao company"]         # 컴포넌트 Copyright 정보
```

mapping 명령어 수행 후 매핑되지 않은 데이터를 OLIVE Platform으로 리포팅합니다.

- 리포팅 된 데이터는 어떤 프로젝트에서 추출된 데이터인지 알 수 없습니다.
- 리포팅 된 데이터는 관리자가 리뷰하여 추후 mapping 명령어 수행 시 매핑 처리될 수 있습니다.

```
$ olive-cli report [--config=<config_path>] [--debug] [--help]
```

옵션	단축어	필수 여부	설명
-- config	-c	N	설정파일의 경로(path) 지정
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

## report 명령어 수행 시 OLIVE 서버에 저장되는 정보

- OLIVE CLI에서 mapping 수행 후 생성되는 매핑되지 않은 의존성 목록(unmapping.csv)
- mapping.json 결과 중 (mapping == false)인 데이터

```

}, {
  "name" : null,
  "componentType" : null,
  "dependencies" : [ ],
  "dependency" : {
    "path" : "parser/pom_xml/pom.xml",
    "fullName" : "com.github.scopt:scopt_${scala.binary.version}:3.7.0",
    "pos" : 0,
    "line" : "",
    "spec" : {
      생략...
    },
    "packageManagerType" : "MAVEN",
    "dependencyType" : "POM_XML"
  },
  "analysis" : {
    "attribute" : {
      생략...
    },
    "spec" : {
      생략...
    },
    "positions" : [ {
      생략...
    } ]
  },
  "url" : null,
  "licenses" : [ ],
  "copyrights" : [ ],
  "component" : null,
  "packageData" : null,
  "mapping" : false,
  "hasIssuedLicense" : false,
  "isExcludeScope" : false
}, {

```

mapping.json

unmapping								
index	count	packageManager	groupid	artifactId	version	scope	path	line
1	2	MAVEN		testLibrary	1.0.0	implementation	parser/gradlex/build2.gradle	implementation(name: 'testLibrary-1.0.0', ext: 'aar')
2	1	MAVEN	com.github.scopt	scopt_\${scala.binary.version}	3.7.0		parser/pom_xml/pom.xml	
3	1	COCOAPOD		ShowsKit			parser/pod/Podfile	pod 'ShowsKit'
4	1	COCOAPOD		Tamra			parser/pod/Podfile	pod 'Tamra', :path => './'
5	1	COCOAPOD		RealmSwift #			parser/pod/Podfile	pod 'RealmSwift' ## https://realm.io/docs/swift/latest/
6	1	COCOAPOD		SDKCommon	2.0.0-beta.4		parser/podspec/Sample.podspec	spec.dependency 'SDKCommon', '#[spec.version]'
7	1	COCOAPOD		SDKTest	2.0.0-beta.4		parser/podspec/Sample.podspec	spec.dependency 'SDKTest', '#[spec.version]'
8	1	COCOAPOD		KakaoOpenSDK/KakaoNavi		level2	parser/podspec/PodspecSampleInput.podspec	subLevel2.dependency '#[spec.name]/KakaoNavi'
9	1	COCOAPOD		KakaoOpenSDK/BB/CC		level3	parser/podspec/PodspecSampleInput.podspec	subLevel3.dependency '#[spec.name]/BB/CC'
10	1	CARTHAGE	jspahrsummers	libextobjc	0.4.1		parser/carthage/Carthfile	github 'jspahrsummers/libextobjc' == 0.4.1
11	2	CARTHAGE	jspahrsummers	xcconfigs			parser/carthage/Carthfile	github 'jspahrsummers/xcconfigs'
12	1	CARTHAGE	enterprise.local/ghc	git-error-translations			parser/carthage/Carthfile	github 'https://enterprise.local/ghc/desktop/git-error-translations'
13	1	CARTHAGE	enterprise.local/desktop	git-error-translations2			parser/carthage/Carthfile	git 'https://enterprise.local/desktop/git-error-translations2.git' "dev"
14	1	CARTHAGE	/directory/to	project			parser/carthage/Carthfile	git 'file://directory/to/project' "branch"
15	2	SWIFT_PM	tannernelson	vapor-stencil	0.0.0		parser/swiftpm/Package.swift	.package(uri: "https://github.com/tannernelson/vapor-stencil.git", r
16	1	SWIFT_PM	IBM-Swift	HeliumLogger	1.0.0		parser/swiftpm/Package.swift	.package(uri: "https://github.com/IBM-Swift/HeliumLogger", major
17	1	SWIFT_PM	IngmarStein	CommandLine	2.2.0-pre3..3.0.0		parser/swiftpm/Package.swift	.package(uri: "https://github.com/IngmarStein/CommandLine.git",

unmapping.csv

OLIVE Platform에 프로젝트를 생성하며, 웹에서 제공하는 다양한 기능을 사용할 수 있습니다.

- config.yaml, dependency.json, manual-component.json 파일로 프로젝트를 생성합니다.
  - 소스코드를 포함한 다른 정보는 전송되지 않습니다.

```
$ olive-cli apply [--config=<config_path>] [--debug|-d] [--help|-h]
```

옵션	단축어	필수 여부	설명
-- config	-c	N	설정파일의 경로(path) 지정
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

OLIVE Platform에 프로젝트를 생성하며, 웹에서 제공하는 다양한 기능을 사용할 수 있습니다.

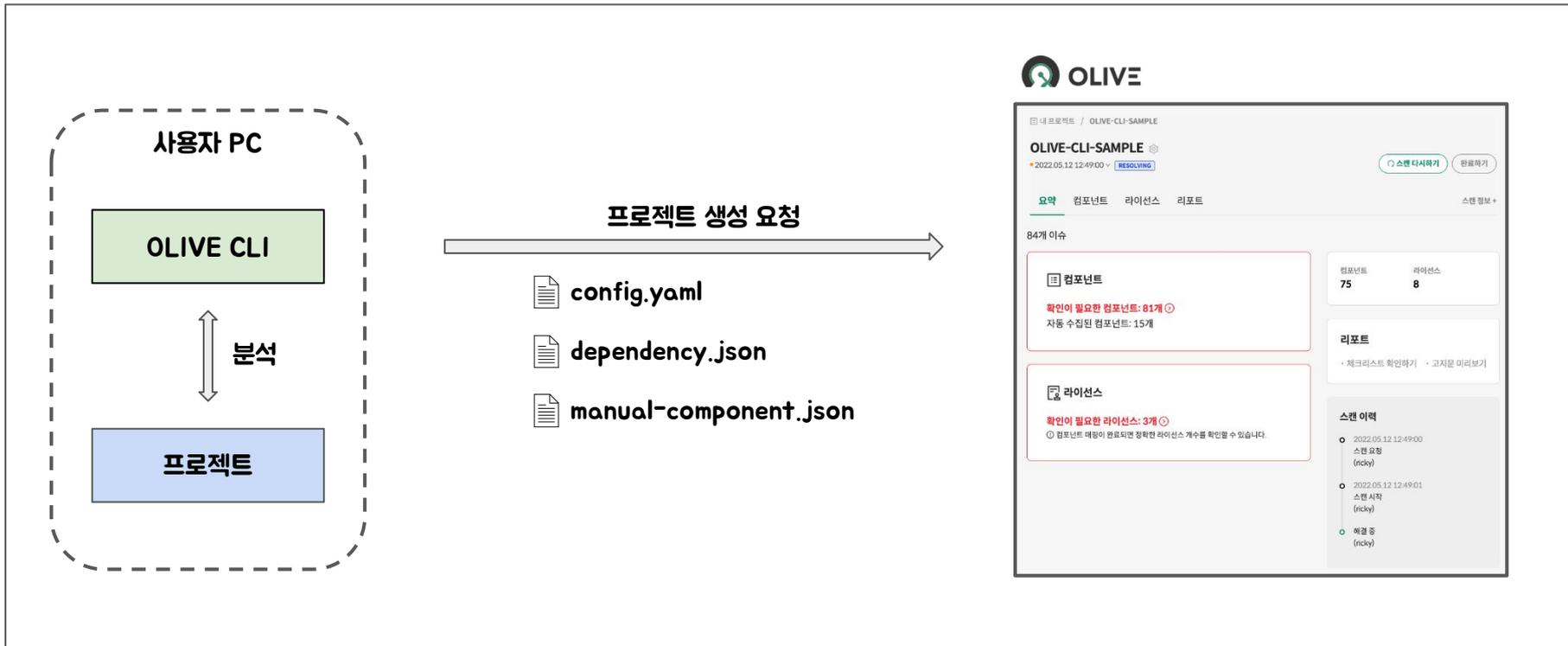
### apply 수행 시 저장되는 정보

- config.yaml (프로젝트명, 오픈소스 유무, excludePath, buildGradeVariant, excludeGradle)
- dependency.json (분석된 정보)
- manual-component.json (수동으로 추가한 컴포넌트 데이터)

```
$ olive-cli apply [--config=<config_path>] [--debug|-d] [--help|-h]
```

옵션	단축어	필수 여부	설명
-- config	-c	N	설정파일의 경로(path) 지정
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

소스코드 전송 없이 사용자 PC 환경에서 OLIVE CLI 분석 결과와 관련 파일을 전송하여 프로젝트를 생성합니다.



고지문 미리보기를 생성합니다.

```
$ olive-cli notice [--config=<config_path>] [--debug] [--help]
```

옵션	단축어	필수 여부	설명
-- config	-c	N	설정파일의 경로(path) 지정
-- debug	-d	N	디버그용 로그 출력
-- help	-h	N	도움말 출력

## 프로젝트에서 사용한 고지 의무 오픈소스 컴포넌트를 고지문으로 생성합니다.

This notice was created using Kakao's OLIVE Platform.

Use this notice for reference purposes only. Kakao does not guarantee the reliability or accuracy of the notice.

All responsibilities arising from the use of OLIVE Platform and the notice rest entirely with users, and Kakao assumes no responsibility for any user or third party.

### OSS Notice | hello-world

This application is Copyright © (owner name). All rights reserved.

The following sets forth attribution notices for third party software that may be contained in this application.

#### **lombok**

<https://github.com/rzwitserloot/lombok>

Copyright 2009-2015 The Project Lombok Authors.

▶ MIT License

#### **Spring Boot**

<https://github.com/spring-projects/spring-boot>

Copyright 2012-2020 Pivotal, Inc.

▶ Apache License 2.0

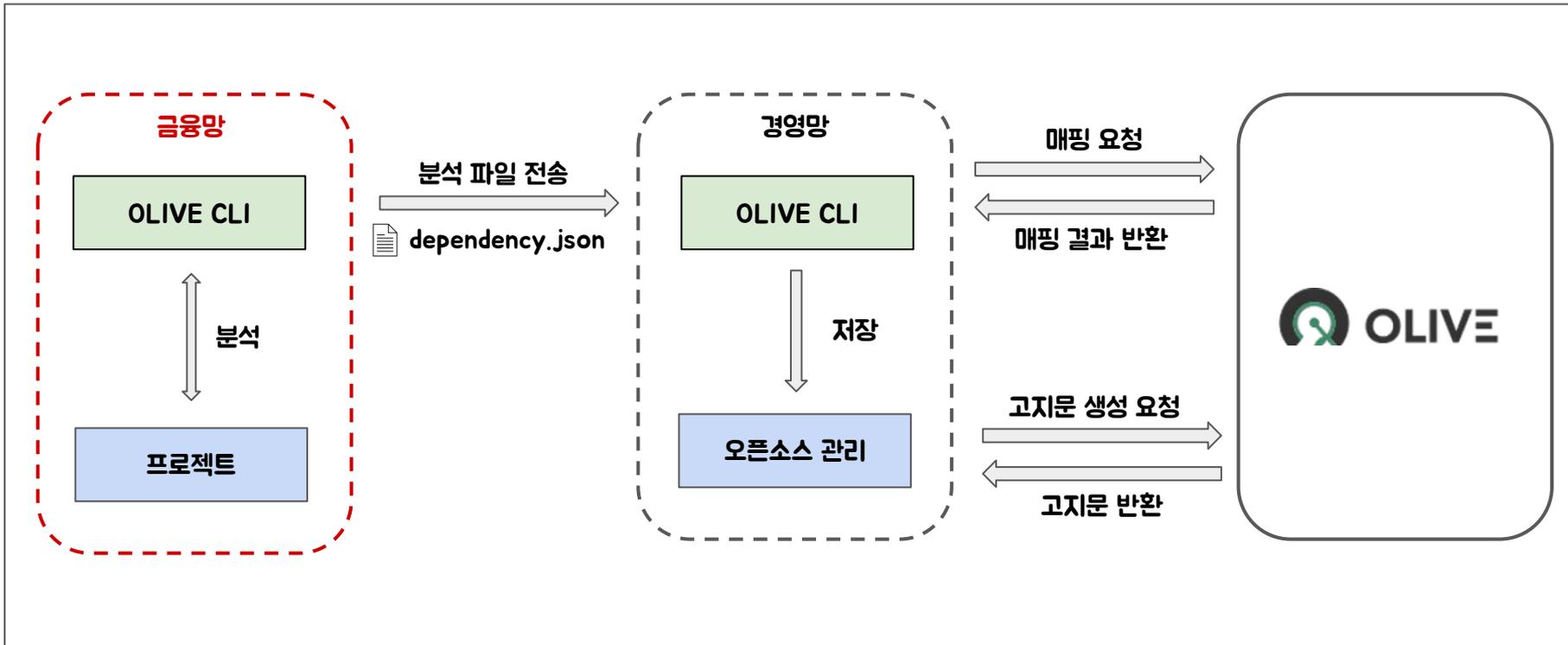
#### **Spring Cloud Azure Starter Storage Blob**

<https://microsoft.github.io/spring-cloud-azure>

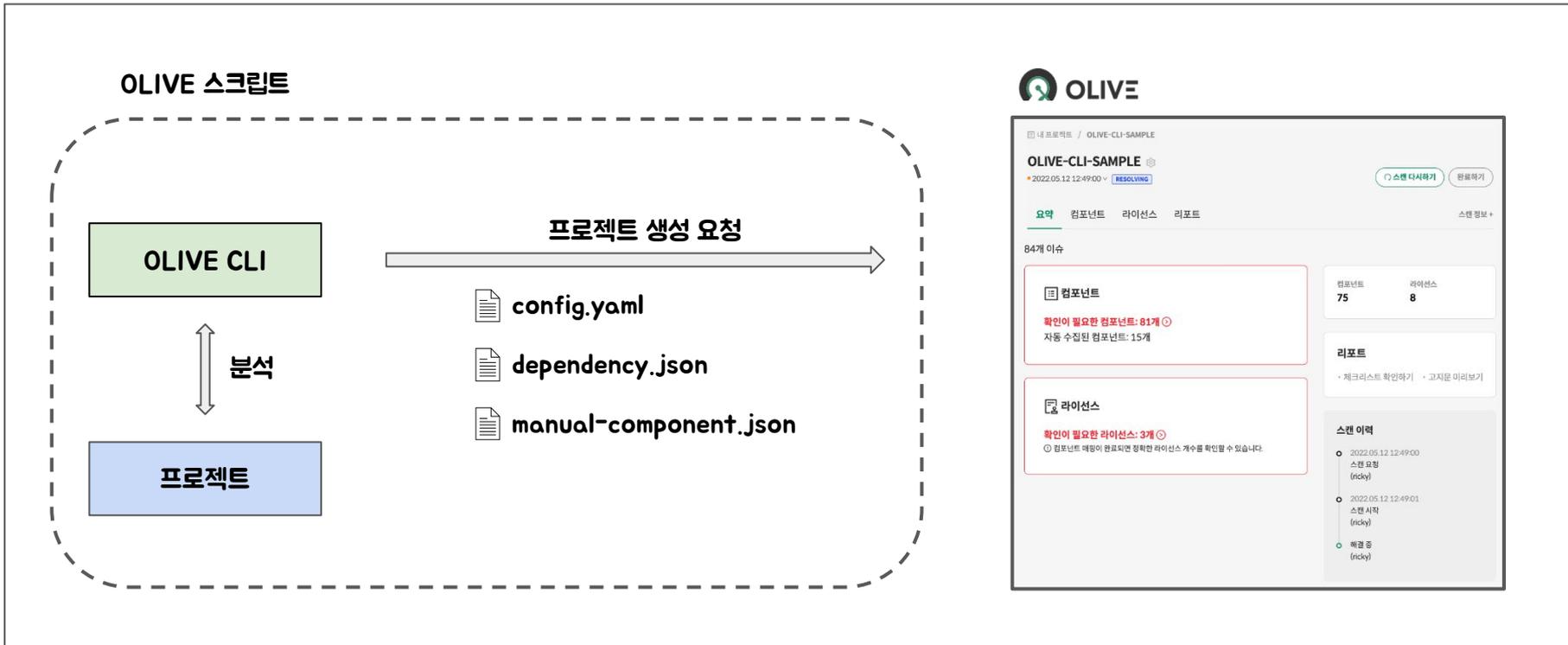
▶ MIT License

## OLIVE CLI 적용(예정) 사례

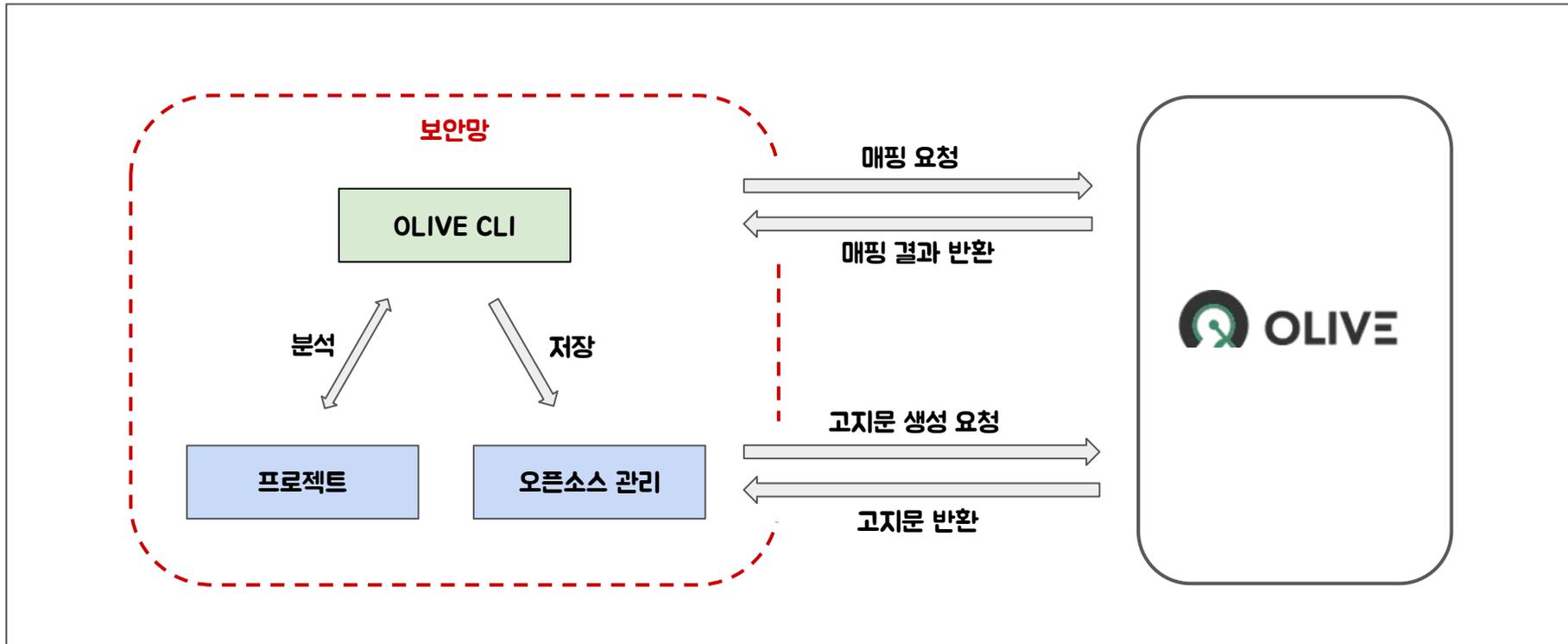
- 금융망과 경영망이 있는데 프로젝트는 외부 통신이 불가능한 금융망에 위치함
- 금융망에서 OLIVE CLI로 분석하여 분석 결과 파일만 경영망으로 전송해 오픈소스 검증



- 사용 중이거나 사용 예정인 오픈소스 프로젝트 분석을 스크립트로 자동화
- 오픈소스 분석 시스템을 DMZ망에 구축하여 OLIVE API 및 OLIVE Platform 웹 서비스 모두 사용할 수 있음



- 외부 통신이 불가능한 보안망에 오픈소스 검증 프로세스 구축
- OLIVE API 호출이 필요한 API는 보안성 검토 후 아웃바운드 설정을 통해 통신 가능하도록 설정



명령어 기능 추가 확대

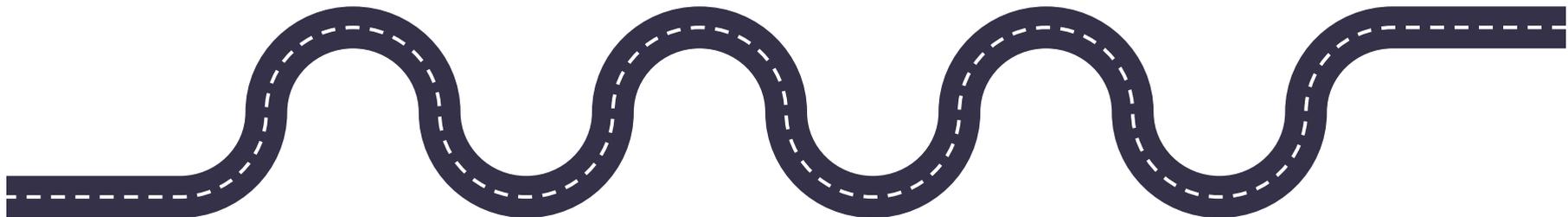
분석 정확도 고도화

CLI 오픈소스 공개



파서 분석 지원 파일 확대

빌드 분석 지원 확대



---

# Q & A

---

# Thank you

---

kakao