

OpenChain-KWG

# Legal Subgroup 3차 회의

2024.2.14.(수)

ETRI  
박정숙

# Content

---

목차	주제	시간	발표자
Updates	관련 WG 활동 소개	15:00~15:10	
Introducing New Issues	SFC v. Vizio 소송 경과	15:10~15:30	이완근 변호사 (OSBC)
	중국의 오픈소스 2차적저작물 소송 판결 결과 소개	15:30~15:40	박원재(LGE)
	LF의 오픈소스 라이선스 컴플라이언스 리포트 소개	15:40~15:50	박정숙(ETRI)
Open Source Litigation Case Analysis	(소송사례) GPL-violations.org는 어떤 사항들을 문제삼았나?	15:50~16:20	김강보(안랩)
Open Source License Analysis	(주제발표) GPL의 발전: GPL-3.0, AGPL-3.0	16:20~16:50	엄송광(ETRI)
Miscellaneous		16:50~17:00	
Wrap Up			

# Updates: Current Status

---

- **AI Study Group**

- Discussing AI compliance matters
- <https://www.openchainproject.org/news/2023/12/20/ai-study-group-planning-call-2023-12-13>

- **OpenChain Legal WG**

- Discussing the Maturity Model from ORCRO
- <https://www.openchainproject.org/news/2024/01/29/openchain-legal-work-group-2024-01-17>

# Legal Workgroup

## • CMM(Capability Maturity Model)

- 지속적인 개선을 최적화할 목적으로 조직 내 비즈니스 기능의 역량, 적응성 및 탄력성 정도를 결정하기 위한 프레임워크
- OpenChain이 CMM 프레임워크 제공 가능
  - ✓ ISO 5230:2020은 조직 내에서 우수한 오픈소스 개발 기능이 보유하는 일련의 특성이 포함
  - ✓ 각 OpenChain 요구 사항은 비즈니스 기능에 대해 매핑될 수 있으며 각 비즈니스 기능에 성숙도가 할당될 수 있음
  - ✓ 계층 구조 제안
    - ❖ 최상위 요구사항은 모든 조직에 적용. 두 번째 수준 요구사항은 조직이 선택한 구현 방법에 맞게 조정
  - ✓ SW 개발(또는 오픈소스 SW 개발)뿐만 아니라 조직의 비즈니스 기능 전체에 걸쳐 기능 성숙도 모델링의 기존 모범 사례와 호환

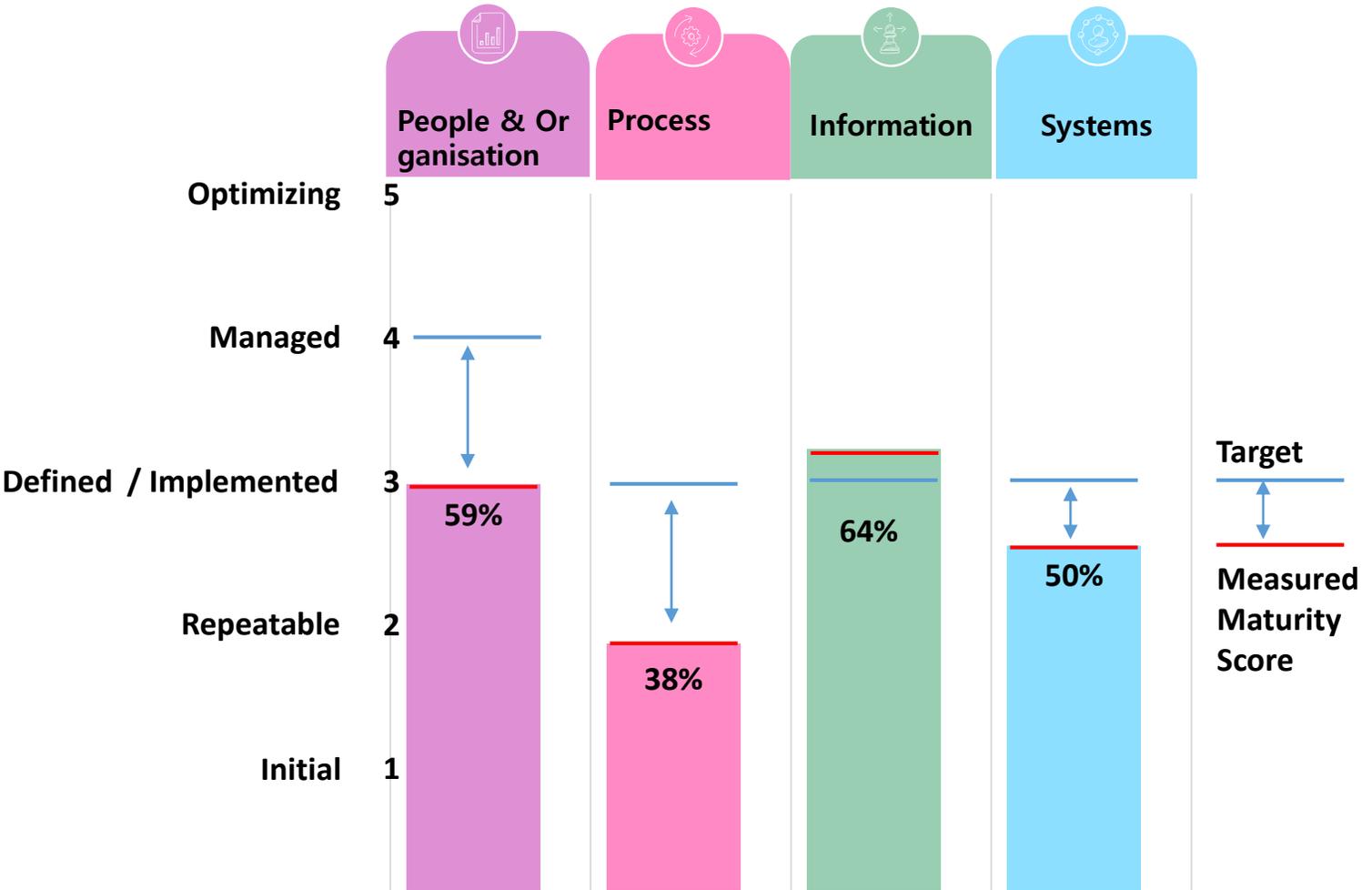
Level	Name of Level	Description
1	Initial	Minimal knowledge of open source compliance practice and procedure 오픈소스 규정 준수 관행 및 절차에 대한 최소한의 지식
2	Repeatable	Some steps towards compliance. Some systems in place, but application ad hoc 규정 준수를 위한 몇 가지 단계. 일부 시스템이 갖추어져 있지만 임시 응용이 적용됨
3	Defined/Implemented	Policies, practice and procedure in place, but not necessarily in operation 정책, 관행 및 절차가 확립되어 있지만 반드시 실행되는 것은 아님
4	Managed	Policies etc. in place, and in operation, and improved as considered necessary 정책 등이 마련되어 운영 중이며 필요하다고 판단되면 개선
5	Optimizing	Policies etc. in place, in operation, and actively managed using appropriate metrics and a process of continuous improvement. 적절한 지표와 지속적인 개선 프로세스를 사용하여 정책 등을 마련하고 운영하며 적극적으로 관리

# Cf) Capability Maturity Model

---

- 참고: <https://blog.naver.com/yjlee0420/140119932062>
- CMM이란?
  - 미국의 대표적인 SW 프로세스 모델
  - SW 개발 능력 측정 기준과 SW 프로세스 평가기준을 제공함으로써 정보 및 전산 조직의 성숙 수준을 평가할 수 있는 모델
- CMM 단계
  - 레벨 1(초기단계)
  - 레벨 2(반복가능한 단계)
  - 레벨 3(정의된 단계)
  - 레벨 4(관리된 단계)
  - 레벨 5(최적화 단계)

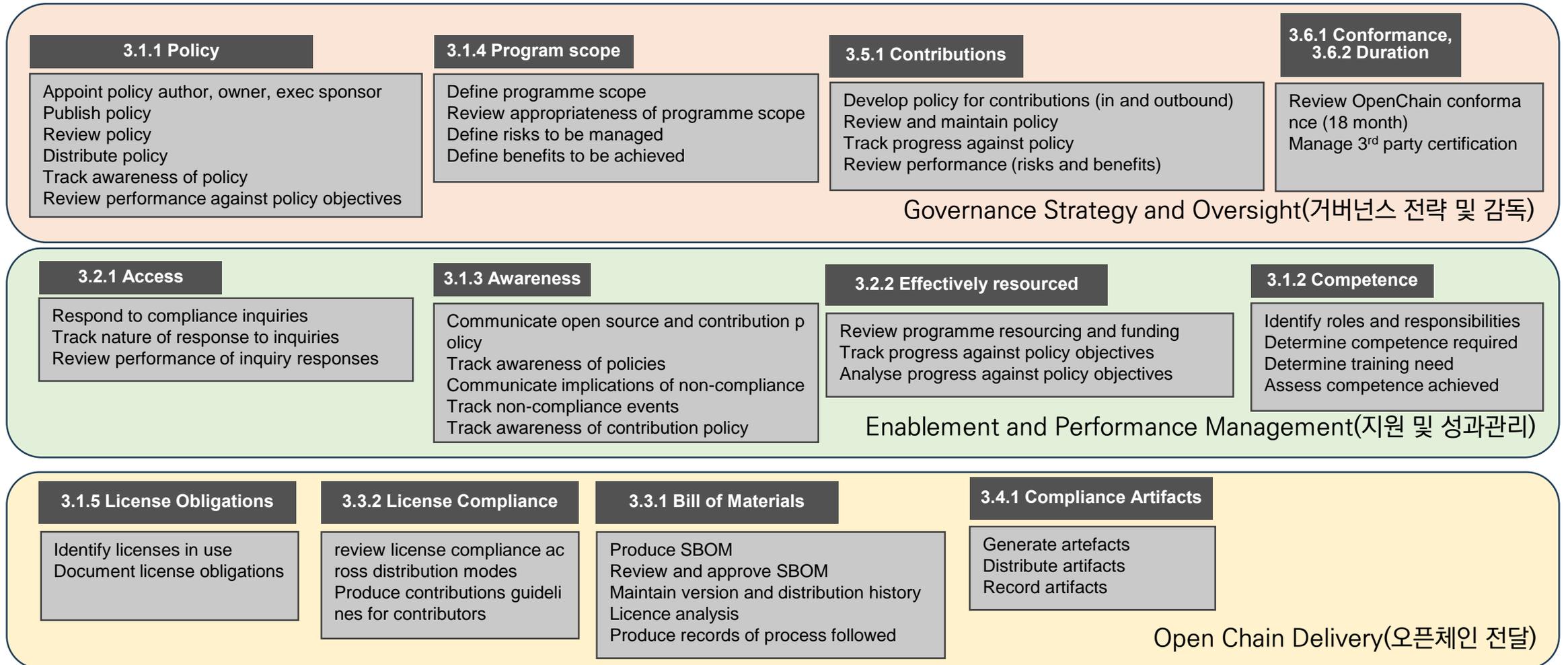
# Legal Workgroup



- 오픈소스 소프트웨어 개발을 관리하기 위해 개발된 조직의 역량은 ISO/IEC 5230:2020의 요구 사항에 따라 고려
- 4가지 유형의 기능으로 분류: 사람과 조직, 프로세스, 정보, 시스템.
- 성숙도: 5가지 완전성 수준에 따라 평가
- 목표 수준: 각 조직에 따라 다르며 비즈니스 위험에 대한 관점, 전달 우선순위에 따라 설정 가능
- 목표 수준과 측정된 수준 사이의 격차는 개선 기회를 정의하고 구현 계획으로 전환

# Legal Workgroup

## • ISO 요구사항 및 프로세스



# Legal Workgroup

## • 샘플 평가

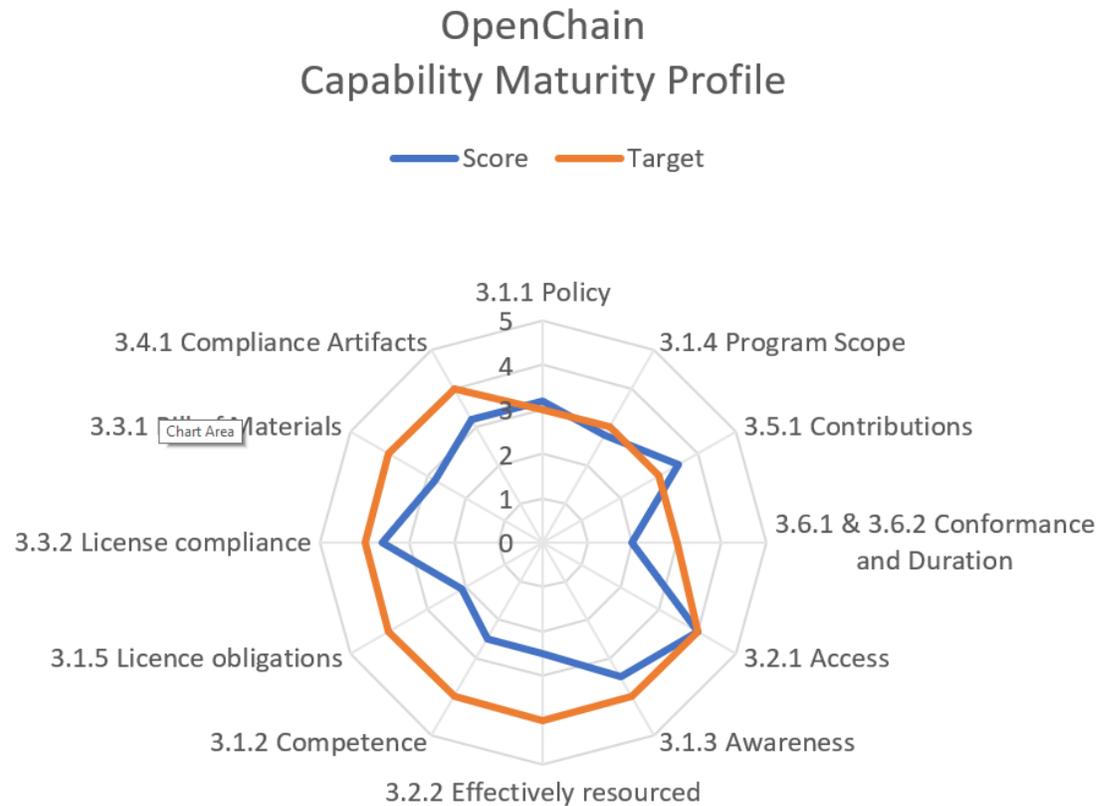
### 3.3.1 Bill of Materials

People and Organisation Capability		Processes Capability	
Attributes	P&O Maturity Questions	Process Attributes	Process Maturity Questions
Key role holders	Does a role exist for generating (or maintaining the system which generates) SBOMs?	Key processes	Does a process exist (automated or not) for generating SBOMs?
Development Team Leader		Produce SBOM	
Associated roles	Are role/responsibility holders suitably trained?	Review and approve SBOM	Does a process exist for reviewing and approving SBOMs?
DevOps Specialist	Do role/responsibility holders have the necessary competencies?	Maintain version and distribution history	As part of any process involving SBOMs, are suitable records kept?
		Produce records of process followed	

Information Capability		Systems Capability	
Information Attributes	Information maturity questions	Systems Attributes	Systems Maturity Questions
Key Information	Are component manifests made available for compliance purposes?	Key systems	Does the compliance toolchain have functionality for generating SBOMs?
Component manifests		Compliance toolchain*	
Correct SBOMs	Are SBOMs generated?	Emerging good practice Metadata repository (such as SW360)	Where issues are identified (e.g., a failing test) is it possible to remedy the issue in-situ?
SBOM records & metadata	Do SBOMs contain sufficient correct data for licence compliance?*		Is compliance metadata stored in a suitable system such as SW360?
	Are SBOMs generated in a way which facilitates other risk management or operational processes (e.g. security/vulnerability or export control)		
	Are standards (e.g. SPDX, CycloneDX) used to generate SBOMs		
	Are the standards used to generate SBOMs consistent across the organisation?		

# Legal Workgroup

## • Full Profile Assessment



격차와 우선순위에 대한 신속한 파악

4가지 기능 렌즈 각각으로 심층적인 분석 가능

오픈체인 프로그램 최적화 지원

# Introducing New Issues

---

- SFC v. Vizio 소송 경과
- 중국의 오픈소스 2차적저작물 소송 판결 결과 소개
- LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

# 1) SFC v. Vizio 소송 경과 및 쟁점 (OSBC 이완근)

---

- 발표자료1 참고

## 2) 중국의 오픈소스 2차적저작물 소송 판결 결과 소개(Case1)

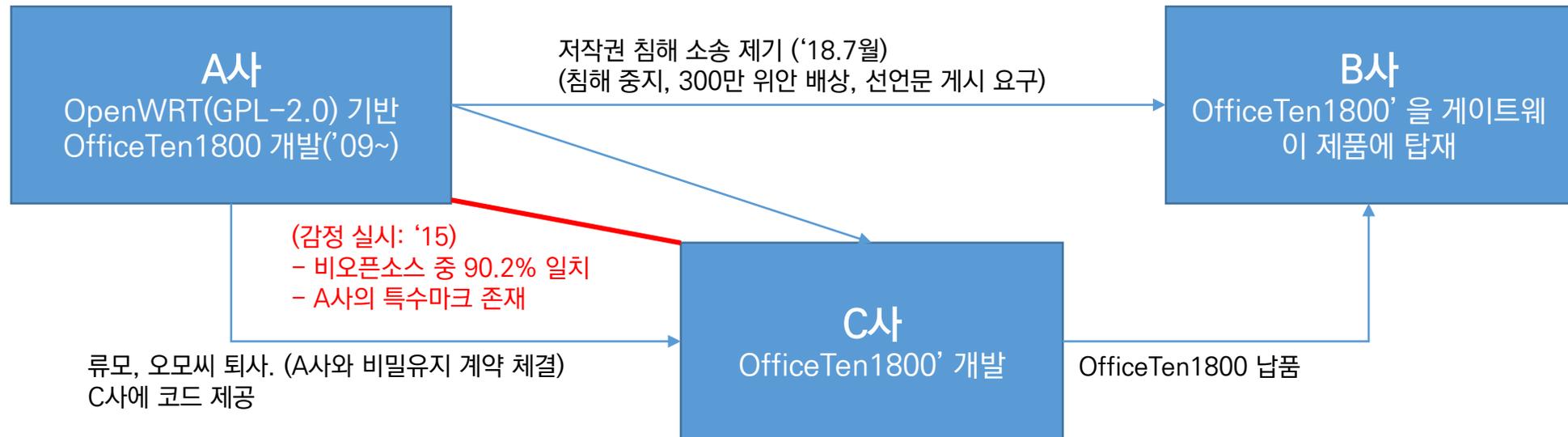
### • Case 1: GPL-2.0 저작권 위반

#### ➤쟁점

✓ B사 등은 침해 사실 부인: OfficeTen1800은 GPL-2.0을 기반으로 함

❖ A사가 코드를 공개하지 않는 것은 GPL-2.0을 위반한 것으로 저작권 없고 타인의 저작권 침해도 주장할 수 없다고 항변

#### ➤사실 관계도



## 2) 중국의 오픈소스 2차적저작물 소송 판결 결과 소개(Case1)

---

### ➤ 법원 판단

#### ✓ 1심(쑤저우 중급법원)

- ❖ 개발자가 오픈소스 제품을 수정하여 사용하거나 2차 개발을 한 경우에도 만약 개발자가 스스로의 노동력을 통해 독창적 작품을 생성했다면 개발자는 해당 작품에 대해 자신의 저작권을 가진다고 판단
- ❖ GPL-2.0 계약에 따르면 관련된 모든 SW가 공개되어야 한다고 볼 수는 없음
- ❖ B, C사의 침해 인정

#### ✓ 2심(최고인민법원)

- ❖ 항소 기각, 원심 판결 유지
- ❖ 사건 당사자가 OpenWRT SW의 관리자가 아니므로 A사의 침해 책임은 심리할 수 없다고 판단
- ❖ A사가 침해 행위에 대한 배상을 청구한 본 사건에서는 A사의 오픈소스 라이선스 위반 여부는 다루지 않음
- ❖ A사의 청구가 일부 받아들여졌으나, 향후 소송에서 A사가 계약 위반 및 권리 침해 책임을 면제받을 수 있음을 의미하는 것은 아니라고 판시

## 2) 중국의 오픈소스 2차적저작물 소송 판결 결과 소개(Case2)

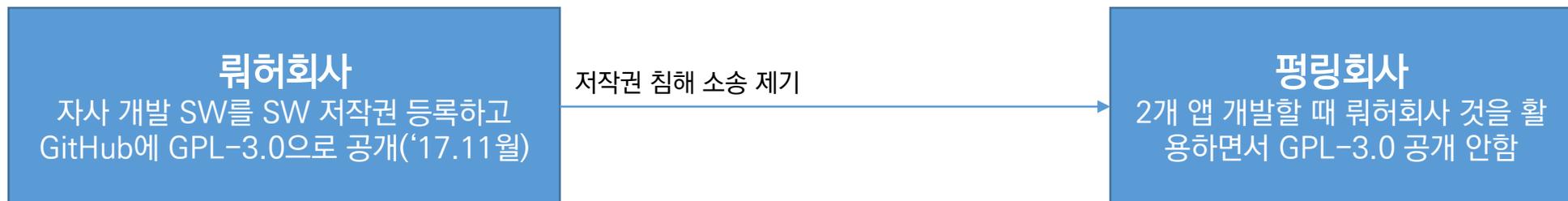
### • Case 2: GPL-3.0 저작권 위반

#### ➤쟁점

✓ 평링회사가 뤼허회사의 원고적격에 대해 이의 제기

❖ 저작권자가 뤼허회사 포함 34인이 존재하므로 뤼허회사가 유일한 저작권자가 아니며 소송 제기 시 나머지에 동의도 받지 않고, GPL-3.0 제10조에 소스코드 개발자는 소송을 제기할 수 없다고 명기되어 있으므로 뤼허회사는 원고적격에 해당하지 않는다고 주장

#### ➤사실 관계도



## 2) 중국의 오픈소스 2차적저작물 소송 판결 결과 소개(Case2)

### ➤ 법원 판단

#### ✓ 1심(광둥성 선전시 중급인민법원)

##### ❖ GPL-3.0 라이선스의 법적 효력

- GPL-3.0 라이선스의 내용을 효력 발생을 목표로 하는 계약의 특징을 지니므로 민사적 법률행위에 해당, 서면 형식으로 체결된 계약의 형태임
- 수권자와 사용자 간에 체결된 저작권 라이선스 계약으로 간주

##### ❖ 뤼허회사의 원고적격 여부

- 총 34인이 분쟁 프로그램의 개발에 참여했으나 뤼허회사가 창시자이자 관리자이므로 저작권자로 인정됨
- GPL-3.0 제10조는 라이선스를 위반한 사용자에게까지 그 저작권을 주장할 수 없다는 의미로 해석하기 힘들
- 뤼허회사가 소스코드의 제공자로서 원시적 권리를 가지고 있음
- 34인은 서로 아는 사이도 아니고 세계에 흩어져 있으므로 뤼허회사의 원고적격을 인정함

##### ❖ 평링회사의 저작권 침해 여부

- GPL-3.0 라이선스 규정에 따라 평링회사가 SW를 무상으로 공개할 의무가 있음에도 공개하지 않음
- 저작권 침해에 해당

##### ❖ 평링회사에게 서비스 중지, 100만 위안을 배상할 것을 판결

#### ✓ 항소심(최고인민법원)

##### ❖ 항소 기각, 원심 판결 유지

### 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

---

- **오픈소스 컴플라이언스 환경의 변화**

- 다양한 오픈소스 구성요소의 다양한 라이선스 요구사항을 탐색하고 라이선스를 준수하여 법적, 재정적 위험 완화 가능
- 회사 간 라이선스 및 계약 협상 대신 강력한 컴플라이언스 프로그램과 신중한 엔지니어링 관행 통해 컴플라이언스 위험 관리
- 정교한 컴플라이언스 전략과 도구가 필요

# 핵심적 컴플라이언스 해결 과제



<p>OSS 라이선스를 준수 위해서는 OSS를 제품이나 서비스에 통합할 때 저작권 고지를 준수하고 라이선스 의무를 이행해야 함</p>	<p>OSS 라이선스 준수를 보장하는 것은 다양한 라이선스 범위, 다양한 이용 약관 및 빠르게 변화하는 소프트웨어 특성으로 인해 복잡하고 까다로움</p>	<p>규정 준수 프로세스에는 제품이나 서비스에 통합된 모든 OSS를 식별하고 해당하는 모든 라이선스 의무를 이행하기 위한 계획을 고안하는 것이 포함</p>
<p>OSS 라이선스 준수를 효과적으로 관리하려면 포괄적인 기능을 갖춘 고급 소프트웨어 구성 분석(SCA) 도구가 필요</p>	<p>조직은 사용자에게 규정 준수 문제 또는 문의 사항을 해결하는 방법에 대한 가시성을 제공하여 개방성, 책임성 및 협업 문화를 촉진할 수 있음</p>	<p>규정 준수를 개발 프로세스에 통합함으로써 조직은 비준수 위험을 줄이는 동시에 건전한 내부 오픈 소스 거버넌스 문화를 촉진할 수 있음</p>
<p>조직은 적절한 도구를 활용하고 내부 지원을 받고 규정 준수 위험을 완화함으로써 OSS 라이선스 규정 준수를 효과적으로 대규모로 관리할 수 있음</p>	<p>SCA 도구는 정확하고 일관성이 있어야 하며, 복잡성을 처리하고, 모든 OSS를 식별하고, OSS 라이선스 환경에 대한 최신 정보를 유지해야 함</p>	<p>모든 OSS 및 라이선스 준수 관련 활동에 대한 투명하고 포괄적인 감사 추적을 유지해야 하기 때문에 감사 가능성은 조직에 중요한 과제</p>
<p>SCA 도구는 SW 개발 라이프사이클과 통합되고 오픈 소스 구성 요소 및 라이선스 요구 사항에 대한 코드를 자동으로 스캔할 수 있어야 함</p>	<p>AI 생성 코드는 조직이 개발자에게 제공되는 정책 옵션과 지침을 사용하여 초기에 해결할 수 있는 새로운 과제를 제시</p>	<p>OSS가 널리 보급됨에 따라 법적 및 평판 위험을 방지하기 위해 강력하고 자동화된 규정 준수 프로세스를 확립하는 것이 중요</p>

### 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

---

- 접근성(accessibility)

- 수많은 오픈소스 라이선스를 사용 중인 조직은 라이선스의 조건을 식별하고 준수하는 데 필요한 도구와 리소스 보유 필요
- 모든 개발자가 라이선스에 대해 동일한 전문지식을 가진 것은 아니므로 도구는 모든 개발자가 쉽게 접근가능해야 함
- 컴플라이언스 보장은 OSS를 받아들이고 오픈소스 프로젝트에 기여하는 개발자부터 시작됨
- 컴플라이언스 도구에는 사용자 친화적인 인터페이스, 명확한 문서, 사용자가 도구를 탐색하는 데 도움이 되는 액세스 가능한 교육 리소스가 있어야 함
- 개발자는 컴플라이언스 프로세스에 직접 참여함으로써 라이선스 요구사항을 이해하고 라이선스 조건을 준수할 수 있음
- 컴플라이언스 도구는 개발팀의 도구와 통합되어 SW 개발 프로세스에 쉽게 통합되어야 함
  - ✓ 통합은 개발 프로세스의 필수적인 부분으로 만들어 규정 준수 노력을 지원
  - ✓ 컴플라이언스를 개발 프로세스에 액세스하고 통합함으로써 조직은 미준수 위험을 줄이는 동시에 우수한 오픈소스 거버넌스 문화 촉진 가능

### 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

---

- 투명성(Transparency)

- 조직은 제품 및 서비스에 통합된 모든 OSS를 식별하고 해당하는 모든 오픈소스 라이선스를 준수하기 위한 메커니즘을 확립하는데 도움이 되는 안정적이고 투명한 컴플라이언스 도구에 액세스할 수 있어야 함
- 조직에는 개인이 잠재적인 규정 준수 문제나 발견된 문제를 보고할 수 있는 **문제 추적기**와 이러한 문제 및 문의를 처리하기 위한 **투명한 프로세스**가 필요
- 사용자와의 **정기적인 의사소통**은 사용자에게 우려사항 조사 진행 상황을 알리는데 중요
- **SBOM** 제공은 제품 또는 서비스 사용자에게 투명성을 제공하는 일반적인 방법
  - ✓ SBOM은 구성 요소 사용, 오픈소스 라이선스 및 사용 중인 오픈소스 구성 요소의 잠재적인 취약성 상태에 대한 가시성 제공
- 투명성은 오픈소스 라이선스 규정 준수에 대한 신뢰와 신뢰성을 구축하는 데 중요
- 조직은 사용자에게 컴플라이언스 **문제 또는 문의사항을 처리하는 방법에 대한 가시성**을 제공하고 책임감 있는 오픈소스 개발 관행에 대한 의지를 보여줌으로써 개방성, 책임감 및 협업 문화 촉진 가능

# 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

- 고급 기능 세트(advanced feature set)

- OSS가 더욱 보편화됨에 따라 법적 및 평판 위험을 방지하기 위해 강력한 컴플라이언스 프로세스를 마련하는 것이 필수
- 고급 기능 세트를 갖춘 규정 준수 도구는 복잡한 개발을 보완하고 포괄적인 규정 준수를 보장해야 함
  - ✓ 자동화된 라이선스 식별과 같은 고급 기능이 있어야 개발자가 오픈소스 구성 요소 및 관련 라이선스를 식별 가능
  - ✓ 사용자가 라이선스 조건 및 의무를 준수하도록 안내하는 라이선스 의무 관리 기능이 있어야 함
  - ✓ 조직의 내부 정책을 기반으로 잠재적인 규정 준수 위험을 식별하고 완화하도록 권장하는 위험 분석 기능도 필수
- 규정 준수 도구는 소스코드 저장소, 지속적인 통합 및 제공 시스템, 프로젝트 관리 도구 등 개발팀이 사용하는 다른 도구와의 통합 지원해야 함. 이러한 통합은 규정 준수 도구가 기존 개발 워크플로에 원활하게 연결되고 모든 개발 팀 구성원이 사용할 수 있도록 보장
- 도구는 조직의 제품과 서비스를 구축하고 실행하는 데 사용되는 모든 오픈소스 아티팩트를 검색하고 관리하는 것이 중요
- 현재의 많은 SCA 도구는 패키지 관리자가 관리하는 구성 요소만 살펴봄
  - ✓ 소스코드 파일, 패키지가 아닌 관리 바이너리 라이브러리, 소스코드 스니펫과 같은 중요한 아티팩트를 검사하지 않음
  - ✓ 이러한 도구 중 다수는 구성 요소 및 해당 전이적 종속성에 대한 최상위 라이선스만 살펴봄
- 완전한 오픈소스 규정 준수를 위해서는 패키지 내 검색 또는 타사 구성 요소에 대한 사전 검색된 SBOM 유사 오픈소스 공개 통과가 필요
- 도구는 컨테이너, 클라우드 네이티브 응용, 서버리스 아키텍처와 같은 다양한 SW 개발환경을 지원해야 함
- 예: 컨테이너
  - ✓ 컨테이너 이미지 라이선스 관리, 다양한 컨테이너 레지스트리 전반의 규정 준수 보장 등 새로운 오픈소스 규정 준수 문제도 발생
  - ✓ 고급 컨테이너 규정 준수 기능을 갖춘 규정 준수 도구를 사용하면 컨테이너 이미지가 오픈소스 라이선스 요구사항을 준수하고 규정 미준수 위험을 최소화할 수 있음

### 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

---

- 확장성(scalability)

- ▶ 조직의 OSS 코드의 양과 사용자 수가 빠르게 압도적으로 늘어날 수 있음
- ▶ 제품 및 서비스의 OSS와 광범위한 채택의 규모와 복잡성을 처리하기 위한 규정 준수 인프라와 지원도구를 갖추는 것이 필수적
- ▶ 규모에 맞게 규정 준수 프로세스를 관리하기 위한 전담 리소스 및 인력 확보와 같은 조직적 지원도 필요
- ▶ 규정 준수 전문가, 교육 프로그램 및 규정 준수가 SW 개발 프로세스의 필수 요소임을 보장하는 거버넌스 정책이 포함될 수 있음 → OSPO를 만드는 것이 일반화
- ▶ 대규모 오픈소스 라이선스 규정 준수를 보장하려면 확장 가능한 아키텍처, 최적화된 성능, 다른 도구와의 통합 지원을 갖춘 규정 준수 도구가 필요
- ▶ 전담 리소스 및 인력과 같은 조직적 지원도 필요: 외부 서비스 활용도 가능

### 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

---

- 속도(speed)

- 들어오는 모든 오픈소스 코드에 대한 규정 준수를 보장하는 속도는 조직이 내부 개발 속도와 제품 및 서비스에 통합된 OSS의 양을 따라잡아야 하기 때문에 중요한 과제임
- 모든 조직의 오픈소스 라이선스 규정 준수 인프라는 개발 프로세스 속도를 늦추지 않고 신속하게 결과를 제공해야 함
- 매우 빠른 속도로 코드를 스캔하고 오픈소스 구성 요소 및 라이선스 요구사항을 빠르고 정확하게 식별할 수 있는 도구를 사용하여 **자동화된 경량 규정 준수 정책 및 프로세스**를 구현함으로써 해결 가능
- 규정 준수 도구는 다른 도구에서 라이선스 정보를 쿼리할 수 있는 **API**와 개발자가 코드를 스캔하고 결과를 빠르게 받을 수 있는 **CLI**를 제공해야 함
- 보다 완전한 분석 및 규정 준수를 위해 **거래 속도와 관련된 모든 장단점과 위험을 인지하고 있어야 함**
- 속도의 또 다른 필수 측면은 규정 준수 프로세스를 완전히 **자동화**하는 능력
- 규정 준수 도구는 **SW 개발 수명 주기**와 **통합** 가능해야 하고 코드에서 오픈소스 코드 및 라이선스 요구사항을 **자동으로 스캔**할 수 있어야 함

# 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

- 정확성(accuracy)

- 규정 준수 도구는 정확한 구성 요소 식별, 라이선스 의무 감지, 위험 분석 제공이 필수
- 동일한 구성 요소에 여러 라이선스가 적용되거나 구성요소 간에 라이선스 종속성이 있는 경우도 처리 가능해야 함
- 대규모 또는 레거시 코드 기반을 처리할 때도 신뢰할 수 있는 결과를 제공해야 함
- 다양한 프로그래밍 언어와 프레임워크를 처리 가능하며 라이선스 요구사항의 변경 사항에 유연하게 대응해야 함
- 오픈소스 규정 준수는 정확한 오픈소스 라이선스 감지, 저작권 관리, 라이선스 의무사항에 크게 의존하므로 강력한 라이선스 감지 및 관리 기능과 데이터를 갖춘 도구를 선택하는 것이 중요함
- 도구가 코드에 있는 실제 라이선스 텍스트 대신 “최선의 추측”을 표시하는 것이 일반적이며 정확한 라이선스 공개를 생성하는 기능이 부족함
- 수정된 오픈소스 라이선스를 검색할 수 있는 기능을 갖는 것은 중요한 규정 준수 기능임
- 오픈소스 라이선스 준수 품질은 오픈소스 커뮤니티 자체에 따라 크게 다름. 규정 준수 도구는 검색된 라이선스와 오픈소스 프로젝트에 의해 공개되거나 저장소 관리자 메타데이터에 표시되는 최상위 라이선스 간의 불일치에 대해 경고할 수 있어야 함
- 규정 준수 도구는 오픈소스 라이선스 및 새로운 오픈소스 구성 요소의 변경 사항에 맞춰 정기적으로 업데이트되어야 함

### 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

---

- 감사가능성(auditability)

- 조직은 오픈소스 라이선스 준수와 관련된 모든 활동에 대한 명확한 **감사 추적**을 제공해야 하기 때문에 오픈소스 라이선스 준수에서는 감사 가능성이 중요
- 규정 준수 인프라는 감사 기능을 제공해야 함
- 조직은 변경 사항을 추적하고 누가 언제 변경했는지 식별 가능해야 함
- 도구는 표준 프로세스의 예외나 편차를 포함하여 **규정 준수 활동 내역**을 제공해야 함
  - ✓ 법률 준수, 위험 관리, 고객 및 오픈소스 커뮤니티와의 신뢰 구축에 필수
  - ✓ 도구는 감사용 보고서를 생성하여 오픈소스 라이선스 요구사항을 준수함을 입증할 수 있어야 함

# 3) LF의 오픈소스 라이선스 컴플라이언스 리포트 소개

- AI 생성 코드(AI-generated code)

- 학습 코드와 라이선스, AI 시스템이 출력의 전체 또는 일부를 재생성한 코드의 라이선스, 출력에 축적된 내용(그대로 복사)이 포함되는지, 오픈소스 코드 사본 및 해당 코드가 파생 저작물로 간주될 수 있는지의 여부와 관련
- 해결 과제
  - ✓ 저작권 문제
  - ✓ 라이선스 호환성 문제
  - ✓ 라이선스 준수 문제
  - ✓ SBOM 문제
- 정책 옵션
  - ✓ Conservative (보수적)
    - ❖ 개발자에게 코드 생성 시 AI 도구를 사용할지 말라고 조언
  - ✓ Selective (선택적)
    - ❖ 용도별: 컨텍스트를 기반으로 일부 사용을 조건부로 허용하지만 새로운 기능을 구현하기 위한 코드 생성같은 다른 사용 및 컨텍스트는 허용 안함
    - ❖ 도구별: 출력에 포함된 오픈소스 코드에 대한 출처를 제공하지 않는 AI 도구를 금지하는 동시에 규정 준수 및 라이선스 호환성 보장
  - ✓ Empower developers (개발자 역량 강화)
    - ❖ 도구 사용에 대한 권장사항과 지침 제공

# Open Source Litigation Case Analysis

---

- “GPL-violations.org는 어떤 사항들을 문제 삼았나?”(안랩 김강보님)
  - 발표자료2 참고

# Open Source License Analysis

---

- “GPL의 발전: GPL-3.0, AGPL-3.0” (ETRI 엄승광님)
  - 발표자료3 참고

**감사합니다**