

논문 2024-4-6 <http://dx.doi.org/10.29056/jsav.2024.12.06>

유사 오픈소스 라이선스와 라이선스 양립성 검사

조용준*†, 김현수*, 신동명*

Pseudo Open Source Licenses and License Compatibility Checking

YongJoon Joe*†, Hyun-Soo Kim*, Dong-Myung Shin*

요 약

오픈소스 소프트웨어(OSS)는 전통적인 라이선스를 통해 소스 코드 공개, 수정, 재배포에 관한 명확한 지침을 제시함으로써 소프트웨어 산업의 혁신을 주도해온 중요한 기반이다. 하지만 최근 클라우드 컴퓨팅과 마이크로서비스 아키텍처(MSA)의 급부상으로 OSS 개발 기반 사업에 큰 변화가 일어났다. 클라우드 서비스 제공업체(CSP)는 OSS를 자사의 관리형 서비스 일부로 포함하거나, 별도의 기여 없이 이를 호스팅하고 상업적으로 활용하게 되었고, 이는 기존의 OSS 개발 기업에게 지속 가능한 사업 구조와 공정한 보상, 생태계 기여 확보라는 난제를 제기한다.

이 문제의 해결책 중 하나로서, 클라우드 시대에 맞추어 재정립된 공유 소스 라이선스나 소스 사용 가능 라이선스 등 ‘유사 오픈소스 라이선스’ 모델이 등장한다. 이 라이선스는 전통적인 OSS 정의를 충족하지 못하나, 서비스 제공 방식, 비즈니스 모델 등의 제한을 가함으로써 CSP와 OSS 개발 기업 간의 사업적 균형을 회복하려는 방안이다. 하지만 이러한 새로운 라이선스의 출현은 단순히 소스 코드 수준에서의 라이선스 호환성 판단을 넘어, 배포 방식, 운영 환경, 서비스 간 연동 방식, 그리고 상업적 사용 방식에 대한 추가적인 정보 수집 및 분석을 필요로 만든다.

본 논문은 유사 오픈소스 라이선스가 소프트웨어 개발 생태계에 어떠한 영향을 미치며, 클라우드 기반 및 MSA 환경에서 발생하는 라이선스 양립성 문제를 판단하기 위한 정보 수집 및 검사 방안을 제안한다.

Abstract

Open Source Software (OSS) has been a crucial foundation driving innovation in the software industry by providing clear guidelines for source code disclosure, modification, and redistribution through traditional licenses. However, the recent rapid rise of cloud computing and Microservices Architecture (MSA) has brought significant changes to OSS-based businesses. Cloud Service Providers (CSPs) have begun incorporating OSS into their managed services or hosting and commercially utilizing it without separate contributions, presenting existing OSS development companies with challenges in maintaining sustainable business structures, securing fair compensation, and ensuring contributions to the ecosystem.

As one solution to these challenges, “Pseudo Open Source Software licenses” models, such as Shared Source Licenses and Source-Available Licenses, have emerged, redefined to suit the cloud era. These licenses do not fully comply with the traditional definition of OSS but aim to restore the business balance between CSPs and OSS development companies by imposing restrictions on service delivery methods and business models. However, the emergence of these new licenses necessitates the collection and analysis of additional information beyond mere source code-level license compatibility. This includes aspects such as distribution methods, operational environments, inter-service integration approaches, and commercial usage patterns.

This paper examines the impact of similar open source licenses on the software development ecosystem and proposes methods for information collection and examination to assess license compatibility issues arising in cloud-based and MSA environments.

한글키워드 : 오픈소스 소프트웨어 라이선스, 유사 오픈소스 라이선스, 클라우드 서비스 제공자, 마이크로 서비스 아키텍처, 라이선스 양립성

keywords : Open Source Software, Open Source-like License, Cloud Service Provider, Microservice Architecture, License Compatibility

* 엘에스웨어(주)

접수일자: 2024.11.25. 심사완료: 2024.12.15.

† 교신저자: 조용준(email: eugene@lsware.com)

게재확정: 2024.12.20.

1. 서론

오픈소스 소프트웨어(OSS)는 소프트웨어 산업에서 혁신을 촉진하는 중요한 역할을 하고 있다 [3]. OSS는 협업 개발과 개방된 접근성을 통해 발전해 왔으며, MIT, Apache, GPL과 같은 전통적인 OSS 라이선스는 소스 코드의 공유, 수정, 재배포에 대한 명확한 지침을 제공한다. 그러나 클라우드 컴퓨팅과 마이크로서비스 아키텍처(MSA)의 부상으로 인해 소프트웨어 개발 환경이 크게 변화하고 있다[1,2].

클라우드 시대에는 클라우드 서비스 제공업체(CSP)가 운영하는 중앙 집중식 플랫폼이 주도적인 역할을 하고 있다. 이들은 OSS를 자사의 독점 서비스 일부로 호스팅하고 배포한다. 이러한 변화는 OSS 개발자와 기업에게 지속 가능성과 공정한 보상 문제를 일으켰고, 공유 소스 라이선스와 소스 사용 가능 라이선스와 같은 새로운 라이선스 모델이 등장하여 라이선스 양립성을 판단하기 위해 배포, 사용, 비즈니스 관행에 대한 추가적인 정보가 필요하게 되었다.

본 논문에서는 새로운 라이선스 모델이 소프트웨어 개발에 미치는 영향과 이러한 라이선스가 적용된 소프트웨어를 활용시 라이선스 양립성 검사 방안을 제안한다.

2. 오픈소스 소프트웨어 라이선스의 변화

2.1 오픈소스 소프트웨어와 라이선싱

오픈소스 소프트웨어(OSS)는 협업 개발과 개방된 접근성을 통해 소프트웨어 산업의 혁신을 주도하고 있다. OSS는 소스 코드를 공개하여 개발자와 조직이 이를 기반으로 새로운 소프트웨어를 개발할 수 있게 한다[1].

2.1.1 OSS 라이선싱의 핵심 원칙

OSS 라이선싱은 투명성, 자유, 협업을 우선시하는 다음과 같은 원칙을 따른다.

1. 소스 코드 접근성: 소스 코드는 공개적으로 공유되어야 하며, 누구나 이를 연구, 수정, 개발할 수 있어야 한다.
2. 수정 및 공유의 자유: 사용자는 소프트웨어를 수정하고 원본 및 수정된 버전을 재배포할 권리가 있다.
3. 비차별성: 누구나 어떤 분야에서도 차별 없이 소프트웨어를 사용할 수 있어야 한다.
4. 저작권 및 공로 인정: 개발자와 기여자는 필수적인 저작권 표시 조항을 통해 기여를 인정받아야 한다.
5. 카피레프트 원칙: GPL과 같은 일부 라이선스는 파생 작업물(2차적 저작물)이 동일한 조건으로 배포하도록 요구한다.

2.1.2 OSS 개발 기업의 역할

OSS 개발 기업은 오픈소스 프로젝트를 유지하고 발전시키는 데 핵심적인 요소 중 하나이다. 이들은 소프트웨어를 무료로 제공하면서도 다음과 같은 방법을 통해 이익을 발생시키고 있다[4].

1. 고객 지원 서비스: OSS의 배포, 구성, 유지관리를 지원하는 유료 서비스 제공(레드햇 엔터프라이즈 리눅스(RHEL)는 지원 및 관리 서비스를 유료로 제공)
2. 기술 지침 및 컨설팅: OSS를 시스템에 통합하거나 비즈니스 요구 사항에 맞게 조정하는 기술 지원 제공. (Elastic은 Elasticsearch 통합에 필요한 컨설팅 서비스 제공)
3. 관리 서비스 및 호스팅: OSS의 호스팅 제공 (MongoDB Atlas는 클라우드에서 운용되는 MongoDB를 제공)
4. 인증 및 교육: OSS 사용자를 대상으로 인증, 교육 프로그램 및 워크숍 제공

2.2 클라우드와 마이크로서비스 아키텍처 (MSA)의 영향

클라우드 컴퓨팅과 마이크로서비스 아키텍처 (MSA)는 소프트웨어 개발 방식에 혁신적인 변화를 가져오고 있다. 이러한 기술은 애플리케이션의 배포, 확장, 관리를 더욱 유연하게 만들며, OSS의 채택과 통합을 촉진하고 있다.

2.2.1 클라우드 컴퓨팅의 부상

클라우드 컴퓨팅은 하드웨어 구입이 아닌 주문하는 방식으로 컴퓨팅 자원을 사용할 수 있는 방법을 제공하며, 소프트웨어 개발에 다음과 같은 변화를 가져오고 있다.

1. 접근성 확대: PostgreSQL, Redis와 같은 OSS 도구를 관리 서비스로 제공
2. 확장성 향상: 대규모 초기 투자 없이도 애플리케이션의 동적 확장 가능
3. 혁신 가속화: SW 자동화 및 배포 도구 제공

2.2.2 마이크로서비스 아키텍처(MSA)

MSA[8]는 대규모 애플리케이션을 독립적인 작은 서비스로 분리하여 설계와 유지 관리 효율화를 목적으로 하는 설계 방식이다. MSA의 주요 특징은 다음과 같다.

1. 개발 최소화: Docker, Kubernetes, gRPC와 같은 도구를 사용하여, 시스템 통합(SI) 대신 개별 SW를 연동하는 방식으로 시스템 개발
2. 개발 속도 향상: 독립된 팀이 병렬로 서비스를 개발할 수 있어 개발 속도 향상

2.3 클라우드 서비스와 MSA로 인한 OSS 개발 기업의 경제적 곤란

클라우드 서비스와 마이크로서비스 아키텍처의 등장으로 기존의 OSS 개발 기업이 제공하던

서비스가 대체되거나 수요가 감소하는 현상이 나타나고 있다.

클라우드 서비스 제공업체(CSP)는 자체 플랫폼에서 오픈소스 소프트웨어를 호스팅하고 관리형 서비스로 제공함으로써, 사용자들이 별도의 설치나 유지 보수 없이도 OSS를 활용할 수 있게 만들었다. 이는 고객 지원, 기술 지원, 관리 서비스 등을 통해 수익을 창출하던 OSS 개발 기업의 비즈니스 모델에 직접적인 영향을 미치고 있다 [4].

또한, MSA의 채택으로 기업들은 독립적인 마이크로서비스를 자체적으로 개발하고 통합하는 방향으로 전환하고 있어, 전문적인 OSS 지원이나 컨설팅에 대한 수요가 줄어들고 있다. 이러한 환경 변화로 인해 OSS 개발 기업의 기존 비즈니스 모델이 큰 위협을 받게 되면서, 지속 가능한 수익 구조를 마련하기 위한 새로운 전략이 요구되고 있다.

3. 유사 오픈소스 라이선스의 등장

3.1 정의와 주요 사례

개발 환경 변화로 인한 비즈니스 모델 붕괴에 대한 대응 방안으로서, OSS 개발 업체는 "유사 오픈소스 라이선스"로 분류되는 라이선스를 제시하고 있다. 이러한 라이선스는 OSS의 일부 원칙을 유지하면서도 현대적인 배포 및 수익화 문제를 해결하기 위해 추가적인 조항을 도입하고 있다.

3.1.1 정의

유사 오픈소스 라이선스는 "공유 소스 라이선스(Shared Source license)"[6] 또는 "소스 사용 가능 라이선스(Source-available license)"[5] 등으로 분류되며, 사용자에게 소스 코드에 대한 접근

근 권한을 부여하지만, 전통적인 오픈소스 정의에서 벗어난 추가적인 제한이나 조건을 부과한다. 주요 특징은 다음과 같다.

1. 소스 코드 접근성: 소스 코드는 접근 가능하지만, 특정 조건에 따라 재배포나 수정이 제한될 수 있다.
2. 사용 제한: 소프트웨어의 배포 방식, 특히 클라우드 환경이나 관리 서비스의 일부로서의 사용을 제한한다.
3. 비즈니스 지향 조항: 수익 공개, 사용 통계 제공, 수익 공유 의무 등의 비즈니스 관련 조건을 포함한다.
4. 비 오픈소스: OSS와 유사하지만, Open Source Initiative(OSI)의 오픈소스 정의(OSD)를 완전히 충족하지 않는다.

3.1.2 주요 유사 오픈소스 라이선스의 예시

① 서버 사이트 퍼블릭 라이선스(SSPL)[9]

도입한 곳: MongoDB, Inc.

주요 특징: GPL의 요구 사항을 서비스로서의 배포까지 확장한다.

관리 서비스로 소프트웨어를 제공하는 경우, 해당 서비스에 사용된 인프라와 관리 도구도 오픈소스화해야 한다.

동기: CSP가 MongoDB를 관리 서비스로 제공하면서 개발에 기여하지 않는 문제를 해결하기 위함이다.

② 엘라스틱 라이선스[7]

도입한 곳: Elastic NV(Elasticsearch, Kibana)

주요 특징: Elastic의 소프트웨어를 사용하여 경쟁 관리 서비스를 만드는 것을 제한.

개인 사용, 개발, 테스트는 무료이지만, 상업적 사용은 제한

동기: CSP(클라우드 서비스 제공자)가 Elastic의 제품을 직접 협력 없이 수익화하는 것을 방지

③ 컨플루언트 커뮤니티 라이선스[10]

도입한 곳: Confluent, Inc.(Apache Kafka 개선판)

주요 특징: 개발 및 프로덕션에서는 무료로 사용할 수 있지만, 소프트웨어를 서비스로 제공하는 것은 금지한다.

동기: Confluent의 비즈니스 모델을 보호하면서 커뮤니티의 기여 촉진

④ 언리얼 엔진 라이선스[11]

도입한 곳: Epic Games

주요 특징: 특정 수익 임계값을 초과하는 상업적 사용에 대해 수익 공유 요구

게임 개발자는 Epic Games와 수익의 일정 비율을 공유

동기: 소규모 개발자에게는 무료로 제공하면서, 엔진을 통한 수익에 대한 공정한 보상

⑤ 비즈니스 소스 라이선스(BSL)[12]

도입한 곳: MariaDB Corporation

주요 특징: 소스 코드는 일정 기간 동안 제한된 조건으로 제공되며, 이후 오픈소스 라이선스로 전환

동기: 단기적인 수익화와 장기적인 오픈소스 기여의 조화

3.2 새로운 라이선스 등장 배경

이러한 새로운 라이선스의 등장은 클라우드 컴퓨팅, MSA, 그리고 변화하는 비즈니스 환경에서 전통적인 OSS 라이선스가 직면한 한계를 극복하기 위한 것이다. 주요 동기는 다음과 같다.

클라우드 수익화 격차 해결: CSP가 OSS를 수익화하면서도 개발에 기여하지 않는 문제 해결 및 OSS 프로젝트에 대한 공정한 보상 보장.

OSS 기업의 비즈니스 모델 보호: CSP의 관행으로 인해 약화된 지원 서비스 및 컨설팅 등의 수익원 보호.

상호주의 촉진: 대규모 사용자로부터의 직접적인 기여를 장려하여 오픈소스 생태계 균형 지향

비즈니스 중심의 라이선싱: 수익 공유 및 사용 제한을 포함하여 비즈니스 모델에 맞는 라이선스 제공

4. 유사 오픈소스 라이선스가 소프트웨어 개발에 미치는 영향

4.1 라이선스 준수 요구 사항의 변화

유사 오픈소스 라이선스[13]는 클라우드 서비스 제공업체(CSP)와 클라우드 기반 서비스에 심각한 영향을 미친다.

1. 서비스 제공 제한: SSPL과 엘라스틱 라이선스는 CSP가 추가 의무 없이 관리 서비스를 제공하는 것을 제한한다.
2. 고객 행동 변화: 고객은 라이선스 준수의 불확실성으로 인해 원래의 OSS 기업과 직접 협력하는 것을 선호할 수 있다.
3. 수익 모델 재검토: CSP는 새로운 라이선스 조건에 따라 수익 모델과 서비스 제공 방식을 재검토해야 한다.

4.2 마이크로서비스 아키텍처(MSA)에 대한 영향

유사 오픈소스 라이선스는 MSA 환경에서의 개발과 배포에도 새로운 문제를 발생시킨다.

1. 종속성 관리 복잡성 증가: 다양한 라이선스 조건으로 인해 종속성 관리가 복잡해진다.
2. 서비스 간 상호 작용 검토: 라이선스 의무가 API 통신을 통해 확장될 수 있으므로, 서비스 간 상호 작용을 신중하게 평가해야 한다.
3. 배포 관행 조정: 제한적인 라이선스로 인해 개발자는 배포 방식을 재고해야 할 수 있다.
4. 아키텍처 설계 영향: 조직은 라이선스 준수를 위해 아키텍처를 설계할 때 새로운 고려 사항을 반영해야 한다.

5. 새로운 라이선스 양립성 판단 문제 해결을 위한 정보 수집 방법

5.1 코드와 바이너리를 넘은 라이선스 호환성 검사 확대

전통적인 라이선스 호환성 검사는 OSS 라이선스 준수를 보장하기 위해 주로 소스 코드와 바이너리를 분석하는 데 초점을 맞추고 있다. 그러나 새로 등장한 유사 오픈소스 라이선스는 코드 이상의 추가적인 요구 사항을 도입하고 있다. 여기에는 배포·운용 방식, 비즈니스 모델, 운영 관행 등이 포함되며, 특히 클라우드 서비스 제공업체(CSP) 환경의 구동 여부가 중요하다.

1. 배포·운용 방식: 소프트웨어의 클라우드 환경, 컨테이너, 마이크로서비스 등 배포 방식
2. 비즈니스 모델: 수익 공유, 사용 보고, 소프트웨어를 서비스로 제공하는 것에 대한 제한
3. 운영 관행: 서비스 제공, 네트워크 사용, 다른 시스템과의 통합과 관련된 의무를 준수하는지 확인해야 한다.
4. 인터넷 연결성: 사용자에게 대한 네트워크 배포 또는 연동 관계
5. 배포 방식: 컨테이너나 가상 이미지로 소프트웨어를 패키징하고 배포하는 방식에 관련된 조건

5.2 양립성 판단을 위한 정보 수집 방안

5.2.1 기존 도구의 한계

FOSSology[14], SPDX(Software Package Data Exchange)[15]와 같은 현재의 라이선스 준수 도구는 주로 소스 코드와 바이너리에서 라이선스를 식별하는 데 집중하고 있다. 이들은 배포 컨텍스트와 비즈니스 모델 제한과 같은 현대 라이선스의 미묘한 요구 사항을 평가하는 기능이 부족하다.

5.2.2 새로운 양립성 판단 방식에 필요한 정보 수집 방안

이러한 격차를 해결하기 위해, 본 논문에서는 위에서 언급한 것과 같은 정보를 바탕으로, 라이선스 양립성 여부를 판단하는 분석에 필요한 요구사항을 정리하였다. 이러한 기술적 요소를 확보함으로써 유사 오픈소스 라이선스를 포함하는 라이선스 양립성 판단을 자동화할 수 있다.

1. 배포·운용 방식 정보 수집: 온프레미스, 클라우드 기반, 하이브리드 등 검사 대상 서비스 및 소프트웨어의 배포 환경 정보를 수집·분석한다.
2. 비즈니스 모델 통합: 수익 모델, 서비스 제공 등 비즈니스 정보를 포함하여 상업적 사용 제한이 있는 라이선스가 필요로 하는 정보를 수집한다.
3. 소프트웨어 종속성 분석: 기존 라이선스 양립성 판단 서비스와 같이 소프트웨어 내의 모든 소프트웨어 구성 요소와 그 종속성을 매핑하여, 결합된 사용으로 인한 라이선스를 식별한다.
4. 서비스 간 통신: MSA 환경에서 서로 다른 서비스가 어떻게 상호 작용하는지 분석하여, API 호출이나 공유 리소스로 인한 결합 여부를 판단한다.

이와 같이, 기존의 소스코드 및 빌드 정보 분석에 더하여, 위와 같은 정보의 수집을 자동화해야만, 클라우드 기반 및 MSA 아키텍처를 채택한 서비스 및 소프트웨어의 라이선스 양립성 여부를 판단할 수 있다.

6. 결론

유사 오픈소스 라이선스의 등장은 소프트웨어 라이선싱의 진화에서 중요한 전환점을 나타낸다. 이러한 라이선스는 클라우드 컴퓨팅, MSA, 현대

비즈니스 모델로 인한 과제를 해결하기 위해 전통적인 오픈소스 소프트웨어(OSS)에서 벗어난 소프트웨어가 등장하게 되었다.

하지만, 이러한 변화는 준수 요구 사항, 클라우드 기반 서비스, MSA 아키텍처에 기반한 서비스를 개발하는 많은 소프트웨어·서비스 개발사에 심각한 영향을 미쳤고, 이에 대응하기 위한 양립성 판단 시스템은 아직 등장하지 못하였다.

본 논문에서는, 이러한 유사 오픈소스 라이선스를 포함하는 소프트웨어의 라이선스 양립성을 판단하기 위해서 어떠한 정보가 필요하며, 어떤 방식으로 수집되어야 할지를 제시하였다.

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2024년도 소프트웨어 저작권 연구개발 사업으로 수행되었음(과제명 : 클라우드 서비스 활용 구축 형태별 대규모 소프트웨어 라이선스 검증 기술개발, 과제번호 : RS-2023-00224818, 기여율 : 100%)

참고 문헌

- [1] Byungil Kim, "A Study on the Compliance with Open Source License and Liabilities in the Cloud Computing Environment", Journal of Korea Information Law, 22.1, 2018, https://www.kci.go.kr/kciportal/landing/article.kci?arti_id=ART002344885
- [2] Gan, Yu, et al. "An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems", Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 2019

- [3] European Commission, “Study about the impact of open source software and hardware on technological independence, competitiveness and innovation in the EU economy”, 2021.09.02., <https://digital-strategy.ec.europa.eu/en/library/study-about-impact-open-source-software-and-hardware-technological-independence-competitiveness-and>
- [4] Spijkerman, Zeena, and Slinger Jansen. “The open source software business model blueprint: A comparative analysis of 10 open source companies”, Software-intensive Business workshop, 128-143, 2018, <https://ceur-ws.org/Vol-2305/paper10.pdf>
- [5] OpenTap, “Introduction to Source-available Licensing”, OpenTap, 2024.03.21, <https://blog.opentap.io/introduction-to-source-available-Licensing>
- [6] Carver, Brian W. “Share and share alike: Understanding and enforcing open source and free software licenses”, Berkeley Technology Law Journal, Vol. 20 No. 1, 443-481, 2005, <http://www.jstor.org/stable/24117523>
- [7] elastic, “Elastic License”, Elasticsearch, <https://www.elastic.co/licensing/elastic-license>
- [8] Nadareishvili, Irakli, et al. “Microservice architecture: aligning principles, practices, and culture”, O’Reilly Media, Inc., 2016, 9781491956250
- [9] MongoDB, “Server Side Public License (SSPL)”, MongoDB, 2018.08.16, <https://www.mongodb.com/legal/licensing/server-side-public-license>
- [10] Confluent, “Confluent Community License Version 1.0”, Confluent, <https://www.confluent.io/confluent-community-license/>
- [11] Epic Games, “Unreal® Engine End User License Agreement”, Unreal Engine, <https://www.unrealengine.com/en-US/eula/>
- unreal
- [12] MariaDB, “Business Source License 1.1”, MariaDB, 2024, <https://mariadb.com/bsl11/>
- [13] Steven R. Argentieri, Michael Perlmutter “Moving Away From Open Source: Trends in Source-Available Licensing”, GOODWIN Law, 2024.09.25., <https://www.goodwinlaw.com/en/insights/publications/2024/09/insights-moving-away-from-open-source-trends-in-licensing>
- [14] Gobeille, Robert. “The fossology project”, Proceedings of the 2008 international working conference on Mining software repositories, 2008, 10.1145/1370750.1370763
- [15] Kapitsaki, Georgia M., Frederik Kramer, and Nikolaos D. Tselikas, “Automating the license compatibility process in open source software with SPDX”, Journal of systems and software, 131, 386-401, 2017, 10.1016/j.jss.2016.06.064

저 자 소 개



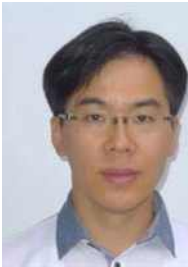
조용준(YongJoon Joe)

2011.03 : 큐슈대학교 전기정보공학과 학사
 2013.03 : 큐슈대학교 정보학부 석사
 2016.03 : 큐슈대학교 정보학부 박사 수료
 2013.04-2016.03 : 일본 학술진흥원 특별연구원
 2016.04- 현재 : 엘에스웨어(주) 소프트웨어연구소
 연구개발본부 기술이사
 <주관심분야> 오픈소스, 저작권, 병렬·분산
 컴퓨팅, 게임이론, 분산 제약 최적화 문제



김현수(Hyun-Soo Kim)

2019.02 : 단국대학교 소프트웨어학과 학사
2023.08 : 숭실대학교 AI·SW융합학과 석사
2019.01-현재 : 엘에스웨어(주) 소프트웨어연구소
연구개발본부 팀장
<주관심분야> 소프트웨어 공학, 딥러닝,
컴퓨터 비전, 분산신원증명, 빅데이터



신동명(Dong-Myung Shin)

2003.08 : 대전대학교 컴퓨터공학과 박사
2001-2006 : 한국정보보호진흥원(KISA)
응용기술팀 선임연구원
2006-2014 : 한국저작권위원회
저작권기술팀 팀장
2014-2016 : 한국스마트그리드사업단
보안인증팀 팀장
2016-현재 : 엘에스웨어(주) 소프트웨어연구소
연구개발본부 연구소장/상무이사
<주관심분야> 오픈소스 라이선스, 저작권 기술,
시스템/네트워크 보안, SW 취약점 분석·감정, 블
록체인 기술