

논문 2025-4-3 <http://dx.doi.org/10.29056/jsav.2025.12.03>

스크립트 기반 게임 방송 제어 라이선스 구현 평가

조용준*† , 홍두표* , 장성일* , 신동명*

Evaluation of a Script-Based License for Game Streaming Control

YongJoon Joe*† , Du-Pyo Hong* , Sung-Il Jang* , Dong-Myung Shin*

요 약

게임 방송은 게임 산업의 핵심 마케팅 채널로 성장했으나, 스토리 중심 게임에서는 라이브/VOD를 통해 엔딩·반전 등 핵심 서사가 노출되어 구매 동기와 저작권자의 경제적 이익을 침해할 수 있다. 현재 퍼블리셔의 스트리밍 가이드라인은 자연어로 제공되어 자동 준수 판단과 사후 재현이 어렵고, 해석의 모호성으로 적용이 비일관적이라는 문제가 있다. 본 논문은 게임 상태와 방송 메타데이터를 입력으로 하는 라이선스 스크립트 기반 실시간 제어 모델을 정의하고, OBS 플러그인과 게임 제어 모듈로 구성된 프로토타입을 통해 가이드라인 조건을 스크립트로 매핑하여 집행하는 구조를 제안한다. 또한 실제 가이드라인에서 추출한 조건을 분류체계로 정리하고, stateless 평가 모델로 표현·집행 가능한 범위와 한계를 정량적으로 제시하며, 실행 로그가 라이선스 위반 판정 및 감정을 위한 증거 구조로 활용될 수 있음을 논의한다.

Abstract

Game streaming has grown into a core marketing channel for the game industry. However, for story-driven games, live streams and VODs can reveal key narrative elements - such as endings and plot twists - thereby undermining purchase incentives and harming the economic interests of copyright holders. Current publisher streaming guidelines are typically written in natural language, which makes automated compliance assessment and post-hoc reproducibility difficult, and their interpretive ambiguity leads to inconsistent enforcement.

This paper defines a license-script-based real-time control model that takes game state and stream metadata as inputs. Using a prototype consisting of an OBS plug-in and a game control module, it proposes an architecture that maps guideline conditions into executable scripts and enforces them in real time. The paper also organizes conditions extracted from real-world guidelines into a classification scheme, quantitatively presents the scope and limits of what can be expressed and enforced via a stateless evaluation model, and discusses how execution logs can serve as evidentiary artifacts for determining license violations and supporting forensic evaluation.

한글키워드 : 게임 방송, 저작권 이용허락, 자동 집행, 실시간 준수, 전자 계약서

keywords : Game streaming, Copyright license, Auto execution, Real-time compliance, Electronic contract

* 엘에스웨어(주)

† 교신저자: 조용준(email: eugene@lsware.com)

접수일자: 2025.11.26. 심사완료: 2025.12.09.

게재확정: 2025.12.20.

1. 서론

1.1 연구 배경

게임 방송은 지난 10여 년간 단순한 팬 커뮤니티

케이션 수단을 넘어, 게임 산업의 핵심 마케팅 채널이자 2차적 저작물 유통 플랫폼으로 성장하였다[1]. YouTube, Twitch, 아프리카TV 등 주요 스트리밍 플랫폼에서 게임 콘텐츠는 전체 시청 시간의 상당 부분을 차지하며, 퍼블리셔 입장에서도 신작 홍보와 커뮤니티 형성에 있어 방송인의 역할이 점차 중요해지고 있다.

그러나 스토리 중심의 싱글플레이 게임에서는 이러한 방송 활동이 양날의 검으로 작용한다. 라이브 스트리밍이나 VOD를 통해 엔딩, 주요 반전, 히든 루트 등 핵심 서사가 공개될 경우, 잠재 구매자의 게임 경험 및 구매 의욕이 훼손되고, 결과적으로 저작권자의 경제적 이익이 침해될 수 있다. 특히 어드벤처 게임, 비주얼노벨, 미스터리 장르에서는 스토리 자체가 콘텐츠의 핵심 가치를 구성하기 때문에[2][3], 이러한 우려가 더욱 두드러진다.

이에 따라 게임 퍼블리셔들은 스트리밍 가이드라인을 제정하여 방송 가능 구간, 방식, 수익화 조건 등을 규정하고 있다[4][5][7]. 일본의 인디·중소 퍼블리셔(Entergram, MAGES, Nihon Falcom 등)는 캡처 기반 컷오프나 체험판 범위 제한 정책을 운용하고, 글로벌 퍼블리셔(Atlus, Square Enix, Nintendo 등)는 발매 후 일정 기간 동안 특정 엔딩이나 추가 콘텐츠의 방송을 금지하는 엠바고 정책을 적용한다. 한국에서도 Buried Stars나 Cheritz의 Mystic Messenger와 같이 라이브/VOD 차별적 허용, 유료 DLC 스토리 보호 등 세분화된 가이드라인이 등장하고 있다.

1.2 문제 정의

이러한 스트리밍 가이드라인은 대부분 자연어 형태로 제공되기 때문에, 다음과 같은 문제점이 존재한다.

첫째, 자동화의 어려움이다. 게임 방송이 가이

드라인을 준수하고 있는지 여부를 시스템이 자동으로 판단하기 어렵다. 현재로서는 저작권자가 수동으로 모니터링하거나 플랫폼의 신고 시스템에 의존해야 하는 실정이다.

둘째, 판단 상황의 재현 불가능성이다. 방송 시점에 실제로 어떤 조건이 유효했는지, 방송인이 어떤 정보를 바탕으로 판단했는지를 사후에 재현하거나 검증하기 어렵다. 이는 분쟁 발생 시 책임 소재를 명확히 하는 데 장애가 된다[6].

셋째, 조건 적용의 비일관성이다. 자연어 가이드라인의 해석에는 모호성이 수반되므로, 동일한 정책이 방송인마다 다르게 적용될 수 있다. 저작권자가 복잡한 조건을 제시할 경우 의도치 않은 판단 오류가 발생하거나, 이를 피하기 위해 지나치게 단순화된 타협안이 제시되기도 한다.

1.3 관련 연구

기존의 스포일러 및 저작권 보호 접근법은 크게 플랫폼 수준의 사후 대응과 콘솔·서비스 수준의 캡처 제어로 나뉜다. Twitch는 DMCA 절차에 따라 권리자의 신고가 접수되면 콘텐츠를 삭제하지만, 라이브 스트리밍 중 스포일러를 실시간으로 필터링하는 기능은 제공하지 않는다[23]. YouTube의 Content ID 역시 사후 처리된 영상을 대상으로 하며, 라이브 필터링보다는 업로드된 콘텐츠에 초점을 맞춘다. 콘솔 수준에서는 Sony의 Spoiler Block 특허가 개발자 메타데이터와 플레이어 진행도를 활용하여 아직 도달하지 않은 스토리 요소를 숨기는 방식을 제안하였으나, 이는 라이브 스트리밍이나 저작권 집행이 아닌 플레이어 경험 향상에 초점을 맞춘다. PlayStation의 캡처 차단 기능은 플래그가 지정된 씬에서 녹화를 일시 중지하는 방식으로 민감한 콘텐츠의 블랙아웃을 구현하지만, 퍼블리셔의 세부 정책과 연동된 실시간 제어와는 거리가 있다.

한편, AI 기반 접근법은 시청자 댓글이나 텍스트 콘텐츠를 분석하여 스포일러를 필터링하는 방식을 연구하고 있다[24]. 그러나 이러한 접근법은 스포일러 콘텐츠가 이미 배포된 후 시청자 측에서 차단하는 구조이므로, 라이브 스트리밍 환경에서 저작권자와 방송인을 동시에 보호하는 데에는 한계가 있다. 콘솔 캡처 플래그나 시청자 측 필터링 모두 게임 엔진과 라이브 스트리밍 인코더, 그리고 퍼블리셔의 정책을 실시간으로 연동하지 못한다는 공통적인 한계를 지닌다. 이러한 격차를 해소하기 위해, 게임 상태와 방송 메타데이터를 기반으로 퍼블리셔 정책을 실행 가능한 규칙으로 변환하여 실시간으로 집행하는 라이선스 스크립트 기반 프로토콜이 제안되었다.

1.4 선행 연구

선행 연구[22]에서는 이러한 문제를 해결하기 위해 게임 소프트웨어와 방송 소프트웨어 간의 실시간 통신 프로토콜과, 동적으로 해석 가능한 전자계약인 라이선스 스크립트(license script)를 이용한 실시간 스포일러 노출 제어 시스템을 제안하였다. 저작권자가 지정한 조건에 따라 화면과 음성을 자동으로 차단·제어함으로써, 스포일러 등 노출을 원하지 않는 콘텐츠를 보호하는 것을 목표로 하는 이 시스템은 게임 상태 제어 모듈(GCM: Game Control Module), 방송 상태·출력 제어 플러그인(BCP: Broadcasting Control Plugin), 라이선스 스크립트를 관리하는 PDS(Personal Data Store)로 구성된다.

그러나 선행 연구는 프로토콜 설계와 개념적 구조 제안 및 실증에 초점을 맞추었기에, 현존하는 다양한 스트리밍 가이드라인을 어디까지 표현하고 집행할 수 있는지에 대한 포괄적인 검증까지는 수행하지 못하였다.

1.5 연구 목적, 범위와 가정

이러한 배경에서 본 논문은 선행 연구의 라이선스 스크립트 기반 실시간 제어 프로토콜 상에서, 현존하는 가이드라인의 라이선스 조건을 어느 범위까지 표현·집행할 수 있는지, 그리고 한계가 어디서 나타나는지를 분석하는 것을 목적으로 한다.

또한 본 논문은 현재 구현된 게임 상태와 방송 메타데이터만을 사용하는 stateless 평가 모델을 기본으로 하고, 이 모델로 구현 가능한 라이선스 조건의 범위를 먼저 평가한다. "한 명의 방송인에게는 여러 엔딩 중 하나만 방송 허용"과 같이 방송 세션 간 누적 이력이 필요한 조건에 대해서는, 게임 코드에 침습적인 수정을 최소화하는 stateful 확장 모델의 필요성을 제안한다.

1.6 연구 기여

본 논문은 게임 상태와 방송 상태 메타데이터를 입력으로 하는 라이선스 스크립트 기반 제어 시스템의 표현 가능성을 평가하고, 라이선스 조건 유형에 따른 현재 구현의 구조적 한계를 분석한다. 대다수의 현실적 라이선스 조건이 stateless 모델로 표현 가능함을 확인하고, stateful 확장이 필요한 조건과 시스템 범위 밖의 조건을 명확히 구분하고, 라이선스 스크립트와 실행 로그가 라이선스 위반 판정 및 라이선스 스크립트의 안전성 감정에서 어떤 역할을 할 수 있는지 논의하고, 스크립트 버전 관리, 방송인 식별자, 메타데이터 표준화, 테스트 및 감정 도구의 필요성 등 향후 과제를 제시한다.

2. 게임 방송 가이드라인 분석 및 라이선스 조건 분류체계

2.1 게임 방송과 라이선스 스크립트 기반 제어

게임 방송은 저작권법상 복제, 공중송신, 2차

적 저작물 작성 등 다양한 권리와 연관된다 [1][4]. 스토리 중심 게임에서 엔딩, 최종장, 핵심 반전의 공개는 잠재 구매자의 게임 경험과 저작권자의 경제적 이익에 직접적인 영향을 미치기 때문에 [12], 퍼블리셔는 게임 방송 가이드라인을 통해 방송 가능 범위, 방식, 수익화 조건 등을 규정하고 있다 [5][15][20].

선행 연구에서는 이러한 가이드라인을 자동 집행하기 위해, 게임 소프트웨어와 방송 소프트웨어 간의 실시간 통신 프로토콜과 라이선스 스크립트 기반 제어 시스템을 제안하였다. 본 논문은 이 선행 연구를 전제로, 실제 가이드라인의 라이선스 조건을 어디까지 표현·집행할 수 있는지를 분석한다.

2.2 라이선스 조건 분류체계

다양한 퍼블리셔의 게임 방송 가이드라인 [5][11][13][14][15][17][19][20]으로부터, 라이선스 조건은 다음 여섯 가지 관점으로 분류했다.

스토리 진행 기반

- 챕터/에피소드 컷오프: 특정 챕터까지만 허용, 이후 금지
- 인게임 날짜 기반: 스토리 내 시간을 기준으로 컷오프 설정
- 루트/엔딩 기반: 진 엔딩·히든 루트 금지, 특정 엔딩만 허용
- 부분 허용: 체험판 범위, 초반 N분, 전체의 1/3~1/2만 허용

이 유형은 게임 내부 스토리 플래그에만 의존하므로, 라이선스 스크립트로 가장 직관적으로 표현 가능하다.

시간 기반(현실 세계)

- 발매 전 전면 금지
- 발매 후 N일간 엔딩·후반부 금지, 이후 해제
- 명시적 해제 일자 설정

이러한 제약은 퍼블리셔에 의해 빈번하게 변

경될 수 있기 때문에, PDS에서 주기적으로 갱신 받는 방식으로 구현한다.

유료 스토리(DLC·애프터 스토리)

유료 DLC, 애프터 스토리 등 유료로만 접근 가능한 스토리의 방송을 금지하는 정책이다. 이때의 제어는 영상만/음성만 제어하는 방식 등이 가능하다.

라이브 vs VOD/자동 아카이브

- 라이브만 허용, VOD·녹화 금지: 후반부 라이브는 허용하되 다시보기 제공 금지
- 방송 플러그인이 자동 아카이브, 녹화 등의 메타데이터를 제공

제업로드·제3자 업로드는 시스템이 관찰할 수 없는 영역이기에, 본 시스템에서 제어하지 않는다. 수익화·행위자·플랫폼

- 수익화: 플랫폼 파트너 프로그램 내 수익화만 허용, 특정 구간 수익화 금지, 전면 비수익화 요구
- 행위자: 개인 팬 방송만 허용, 법인·사업 목적은 별도 계약 요구 [19][20]
- 플랫폼: 특정 플랫폼(음악 라이선스 연계)으로 제한

위의 정보는 계약 체결 단계에서 결정되어 있기 때문에, 스크립트의 입력으로 사용할 수 있다. 형식·변형·에티켓(정책 레이어)

- 무편집·무코멘터리 영상 차제 [17][18]
- 컷신·BGM만 모은 "스토리 덤프" 영상 금지 [15]
- 스포일러 경고 표시 요구 [15]

이러한 조건은 게임 상태·방송 메타데이터만으로 판단하기 어렵고, 실시간 제어 시스템의 구현 범위를 넘어선다. 본 논문에서는 분류체계에 포함하되, 자동 집행 대상에서는 제외한다.

2.3 분류체계 요약

이 분류체계는 이후 장에서 라이선스 스크립트

설계와 표현력 평가의 입력 스펙으로 사용된다.

3. 라이선스 스크립트 기반 실시간 제어 모델

본 장에서는 라이선스 조건 분류체계를 표현하고 집행하기 위한 라이선스 스크립트의 데이터 모델, DSL 구조, 실행 모델을 정의한다.

표 1 기존 정책의 조항 분류 체계
Table 1. Classification system of existing policy provisions

관점	주요 조건 유형	자동 제어 가능 여부
스토리 진행	챕터 컷오프, 엔딩/루트 금지, 부분 허용	O
시간	발매 전/후 제한, 엠바고	O
유료 스토리	DLC·애프터 스토리 번호	O
라이브/VOD	라이브만 허용, 아카이브 금지	O
수익화·행위자·플랫폼	파트너 수익화, 개인/법인, 플랫폼 제한	O
형식·변형·에티켓	편집 요구, 스포일러 경고	X

3.1 시스템 개요

본 논문이 전제로 하는 시스템은 선행 연구에서 제안된 게임-방송 소프트웨어 간 실시간 통신 프로토콜과 라이선스 스크립트를 기반으로 한다. 구성 요소는 다음 세 가지로 요약된다.

게임 제어 모듈(GCM): 게임 클라이언트 내부에 포함되며, 스토리 진행도(챕터·루트·엔딩 ID,

인게임 날짜), 세그먼트 ID(본편/체험판/DLC 구분), 콘솔·엔진이 제공하는 캡처/방송 허용 플래그 등을 수집하여 방송 제어에 필요한 게임 상태를 생성한다.

방송 제어 플러그인 (BCP): OBS 등 방송 소프트웨어에 탑재되는 플러그인으로, 플랫폼 종류 (YouTube, Twitch 등), 스트림 키를 통해 확인한 방송인 ID, 플랫폼 수준의 수익화 상태, 자동 아카이브/로컬 녹화 설정 등의 메타데이터를 관리한다. 또한 라이선스 스크립트의 평가 결과에 따라 영상·음성을 블랙아웃하거나, 자동 아카이브·수익화 설정을 변경하는 제어를 수행한다.

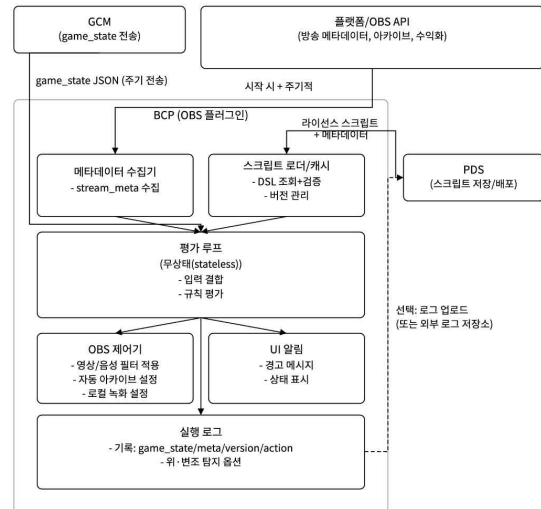


그림 1 실시간 제어 모델 전체 구성도
Fig. 1. Overall configuration of Real-time control

PDS: 스트리머, 게임사, 계약 플랫폼 사이에서 체결된 라이선스 계약의 표현인 라이선스 스크립트를 보관·배포하며, 스크립트의 버전·유효기간·행위자 속성을 관리한다. 계약 플랫폼에서 결정된 개인/법인 여부, 계약 유형 등은 PDS를 통해 BCP·GCM에 전달된다.

전체 프로토콜은 (1) Discovery, (2) Mutual Check, (3) Active Control 단계로 동작한다. 본

장에서는 이 중 Active Control 단계에서 사용되는 데이터 모델과 스크립트 실행 모델을 정의한다.

3.2 입력 데이터 모델

라이선스 스크립트는 매 평가 시점마다 "현재 프레임의 게임 상태"와 "방송 메타데이터"를 입력으로 받는 함수로 정의한다.

3.2.1 게임 상태(game_state)

게임 상태는 다음과 같은 필드를 포함한다.

- game_id: 게임 식별자(버전·플랫폼 구분 포함)
- story_segment_id: 현재 재생 중인 스토리 세그먼트 ID (챕터/에피소드, 특정 컷신, 퍼즐/증거 화면 등으로 세분화 가능)
- route_id / ending_id: 루트/엔딩 식별자(정규 엔딩, 진 엔딩, 히든 엔딩 등)
- progress_ratio: 전체 스토리 대비 진행 비율 (예: 0.0~1.0), 체험판/초반 N분/1/3~1/2 허용 규칙 대응
- in_game_date: 인게임 날짜/시간(날짜 기반 컷오프 표현용)
- is_dlc_content: 현재 세그먼트가 유료 DLC/애프터 스토리인지 여부
- is_trial: 현재 세그먼트가 체험판 범위인지 여부
- elapsed_minutes: 게임 시작 후 경과 시간 (분), 초반 N분 허용 규칙 대응
- capture_allowed: 콘솔·엔진 차원에서의 캡처 허용/금지 플래그

이 필드들은 퍼블리셔·개발사가 게임 빌드 단계에서 태깅하는 것을 전제로 한다. 특히 유료 DLC·애프터 스토리, 엔딩·최종장, 주요 반전 구간 등은 별도의 story_segment_id 또는 플래그로 명시되어야 한다.

3.2.2 방송 메타데이터(stream_meta)

방송 메타데이터는 BCP와 PDS가 제공하는 정보로 구성된다.

- platform: 방송 플랫폼 종류(예: YOUTUBE, TWITCH, NICONICO 등)
- streamer_id: 스트림 키로부터 매핑된 방송인 ID
- actor_type: 계약 플랫폼에서 판정한 행위자 유형(INDIVIDUAL / ENTITY)
- contract_type: 계약 유형(예: FAN_NONCOMMERCIAL, PARTNER_MONETIZED 등)
- monetized: 현재 스트림이 플랫폼 파트너 프로그램을 통한 수익화를 사용 중인지 여부
- auto_archive_enabled: 플랫폼에서 자동 다시보기/VOD 저장이 활성화되어 있는지 여부
- local_recording_enabled: 로컬 녹화 기능 사용 여부
- current_time: 실세계 시간(UTC 기준), 발매 후 경과일/엠바고 날짜 계산용

여기서 개인/법인 구분, 팬/사업 목적 구분은 방송 소프트웨어가 아니라 별도의 계약 플랫폼에서 처리된 결과를 PDS가 속성으로 제공하는 것으로 전제한다. 따라서 라이선스 스크립트는 actor_type, contract_type을 신뢰 가능한 계약 정보로 취급하며, BCP는 이를 수정하지 않는다.

지역 제한(시청자 위치 기반)이나 재업로드 여부는 현재 기술·정책 환경에서 플러그인 수준에서 관측하기 어렵기 때문에 데이터 모델에서 제외한다.

3.2.3 라이선스 메타데이터(license_meta)

라이선스 스크립트 자체에 부착되는 메타데이터는 다음과 같다.

- license_id: 스크립트(계약)의 식별자 및 버전
- game_id: 적용 대상 게임

- valid_from, expire_at: 계약 효력 시작/종료 시각
- publisher_id: 스크립트를 발행한 퍼블리셔·권리자 ID
- jurisdiction: 계약 관할(국가/지역) - 법적 해석을 위한 참고 정보
- log_policy: 로그 보관·감사 정책(보존 기간, 익명화 수준 등)

이 메타데이터는 라이선스 위반 판정 시 "어떤 스크립트가 언제까지 유효했는지"를 재구성할 수 있게 하는 핵심 증거 자료가 된다.

3.3 Stateless 평가 모델

본 논문에서 구현하는 기본 실행 모델은 stateless 평가 모델이다. 즉, 라이선스 스크립트는 매 평가 시점 t마다

$(game_state(t), stream_meta(t)) \rightarrow action(t)$

을 계산하는 순수 함수로 정의되며, 과거 시점의 입력이나 출력에 직접 접근하지 않는다.

3.3.1 Stateless 모델의 설계 의도

단순성·예측 가능성: 각 프레임 또는 이벤트에 대해 동일한 입력이 주어지면 항상 동일한 출력이 나온다.

성능·지연 최소화: 추가 상태 관리가 없으므로 평가 비용이 작고, 실시간 방송 지연에 미치는 영향이 적다.

재현 용이성: 나중에 라이선스 위반 판정 시, 해당 시점의 game_state, stream_meta, 스크립트 버전만 있으면 시스템이 어떤 판단을 했어야 하는지 재계산할 수 있다.

이 모델 하에서는 2장에서 정의한 대부분의 라이선스 조건(챗터 컷오프, 체험판·부분 허용, 발매 후 N일 제한, 유료 DLC 금지, 라이브/VOD 구분, 수익화·플랫폼 조건 등)이 현재 입력만으로 판정 가능하다.

3.3.2 Stateful 확장을 위한 최소 인터페이스
다만 "여러 엔딩 중 방송인별로 하나만 공개 허용"과 같이 방송 세션 간 누적 이력에 의존하는 라이선스 조건은 stateless 모델만으로는 표현할 수 없다.

본 논문에서는 구현 범위를 stateless 모델로 한정하되, 향후 확장을 위해 개념적으로 다음과 같은 최소 상태 인터페이스를 상정한다.

- get_state(key): PDS 또는 방송 소프트웨어가 관리하는 외부 상태 조회

- set_state(key, value): 외부 상태 갱신

예를 들어 "방송인별로 이미 공개한 엔딩 목록"을 state["streamer_id:endings_shown"]에 저장하고, 새로운 엔딩을 방송하려 할 때 해당 목록을 참조하여 허용 여부를 판단하는 구조를 설계할 수 있다. 이때 상태 저장 위치는 게임이 아니라 BCP 또는 PDS에 두어, 기존 게임 코드를 방송 전용으로 침습적으로 변경하지 않는 것을 원칙으로 한다. 구체적인 stateful 모델과 구현 가능성 평가는 6장에서 별도로 논의한다.

3.4 스크립트 버전·시점 관리

게임 방송에 대한 라이선스 조건은 시간에 따라 내용이 변경될 수 있으며, 이는 저작권·마케팅 전략(출시 직후 강한 제한, 일정 기간 후 완화 등)에 따라 자연스러운 현상이다.

- PDS는 각 게임·계약 조합에 대해 단일한 최신 스크립트 버전을 제공하며, valid_from, expire_at 필드를 통해 효력 기간을 명시한다.

- 방송 시작 시, BCP는 PDS로부터 현재 시점에 유효한 스크립트를 취득하고, 방송 도중에도 주기적으로(또는 expire_at 이전에) 갱신 여부를 확인한다.

- 스크립트가 갱신되면, 새로운 스크립트와 평가 시각을 함께 로그에 기록하여, 나중에 "어떤 규칙이 언제부터 적용되었는지"를 추

적할 수 있게 한다.

이 구조는 향후 분쟁 시 "방송 당시 적용되던 라이선스 조건"을 재구성하는 데 필요한 최소 정보를 제공한다.

4. 구현

프로토타입은 Windows 10/11 환경에서 OBS 30.2.3에 C++ 기반 플러그인(BCP) 형태로 구현하였다. GCM→BCP, BCP→PDS 통신은 HTTP 기반으로 JSON 메시지를 송수신한다. 본 프로토타입은 표현력/집행 가능성 검증을 목적으로 하며, 전송 구간 암호화는 적용하지 않았다.

4.1 구현 목표 및 설계 원칙

프로토타입 구현에서 설정한 주요 설계 원칙은 다음과 같다.

비침습적 통합: 기존 게임 코드에 최소한의 모듈만 삽입하고, 방송 시스템(OBS 등)에는 플러그인 형태로 추가하는 방식으로 설계한다. 기존 게임 실행 흐름과 방송 환경에 침습적인 변경을 가하지 않는 것을 목표로 한다.

Stateless 평가 우선: 3장에서 정의한 stateless 모델까지만 실제 구현 대상으로 포함하고, stateful 확장은 개념 수준에 머문다.

4.2 게임 제어 모듈(GCM) 구현

GCM은 게임과 독립적으로 동작하는 경량 모듈로, 게임 엔진이 제공하는 상태 조회 함수를 호출하여 현재 게임 상태를 취득한다. 게임 엔진에는 현재 상태(chapter_id, route_id, progress_ratio, is_dlc_content 등)를 반환하는 조회 함수만 추가하면 되며, GCM이 주기적으로 (예: 1초 단위) 해당 함수를 호출하여 상태를 수집한다. 수집된 정보는 JSON 형식으로 직렬화되

어 통해 방송 플러그인(BCP)으로 전송된다.

4.3 방송 제어 플러그인(BCP) 구현

BCP는 OBS와 같은 방송 소프트웨어에 탑재되는 플러그인으로, 라이선스 스크립트를 로딩·평가하고 해당 결과를 실제 방송 출력에 반영한다. BCP는 방송 시작 시 platform, streamer_id, monetized, auto_archive_enabled, local_recording_enabled 등의 방송 메타데이터를 수집하고, PDS로부터 해당 게임·방송인 조합에 적용되는 스크립트를 취득하여 캐싱한다.

BCP의 핵심은 평가 루프로, GCM으로부터 수신한 game_state와 방송 메타데이터를 결합하여 라이선스 스크립트를 stateless 평가하고, 산출된 액션에 따라 영상/음성 필터 적용, 자동 아카이브/녹화 설정 변경, 경고 메시지 표시 등의 제어를 수행한다.

4.4 PDS 및 계약 플랫폼 연동

PDS는 라이선스 스크립트와 계약 정보를 관리하는 중앙 저장소 역할을 한다.

4.4.1 스크립트 저장 구조

각 스크립트는 다음과 같은 메타데이터와 함께 저장된다.

- license_id, version
- game_id, publisher_id
- applicable_platforms (예: YOUTUBE, TWITCH)
- valid_from, expire_at (발매 전 금지, 발매 후 일정 기간 엔딩 금지 등 구현에 사용)
- actor_type_filter, contract_type_filter (개인/법인, 비상업/파트너 수익화 구분을 위한 필터 조건)

실제 스크립트 본문은 별도 필드에 저장되며, 파서가 해석 가능한 DSL 형식을 따른다.

4.4.2 계약 플랫폼과의 연계

계약 플랫폼은 방송인과 퍼블리셔 사이의 계약 체결·변경을 담당하고, 그 결과를 PDS에 반영한다.

- 방송인이 특정 게임에 대한 이용 허락을 신청
- 퍼블리셔가 승인
- 계약 플랫폼이 streamer_id, actor_type, contract_type_filter, valid_until 등을 PDS에 등록
- 이후 BCP가 PDS에 질의할 때 이 정보를 포함한 스크립트가 반환

이 구조를 통해, "개인 팬 방송은 허용, 법인 계정은 별도 계약 필요", "파트너 수익화 허용, 기타 영리 이용 금지"와 같은 정책을 스크립트에 자연스럽게 반영할 수 있다.

4.5 라이선스 조건 사례별 스크립트 매핑

2장에서 정리한 가이드라인 사례를 라이선스 스크립트 DSL로 매핑한 예시는 다음과 같다.

체험판/초반부만 허용: "체험판 또는 초반 약 20분까지만 허용" 규칙은 다음과 같이 표현할 수 있다.

```
if game_state.is_trial == false
  and game_state.elapsed_minutes > 20
then BLOCK_VIDEO + BLOCK_AUDIO
```

라이브/VOD 차별적 허용: 후반 구간에서 라이브는 허용하되 VOD·녹화는 금지하는 정책은 다음과 같다.

```
if game_state.story_segment_id in LATE_GAME_SEGMENTS then ALLOW + DISABLE_AUTO_ARC HIVE + DISABLE_LOCAL_RECORDING
```

유료 DLC 보호: "유료 DLC·애프터 엔딩은 공개 금지" 규칙은 DLC 플래그만으로 판정 가능하다.

```
if game_state.is_dlc_content == true then BLOCK_VIDEO + BLOCK_AUDIO
```

발매 후 기간 제한: "진 엔딩은 발매 후 30일 간 금지" 규칙은 시간 비교로 구현된다.

```
if game_state.ending_id == "TRUE"
  and now < RELEASE_DATE + 30d
then BLOCK_VIDEO + BLOCK_AUDIO
```

이러한 규칙은 모두 현재 상태만으로 판정 가능한 stateless 모델로 구현된다. 반면 "방송인별로 하나의 엔딩만 공개 허용"과 같이 방송 이력에 의존하는 규칙은 stateful 확장이 필요하며, 본 프로토타입에는 포함하지 않는다.

5. 평가

5.1 평가 목표 및 방법

본 평가의 목표는 (1) 2장에서 정의한 라이선스 조건 분류체계가 stateless 라이선스 스크립트로 어느 정도까지 표현·집행 가능한지 확인하는 표현력 평가와, (2) 퍼블리셔·개발자·방송인 입장에서 부담을 검토하는 운영 평가이다. 다양한 퍼블리셔 가이드라인에서 대표 정책을 선정하여 라이선스 스크립트 DSL로 기술하고, GCM·BCP·PDS가 연결된 프로토타입 상에서 동작 여부를 확인하였다.

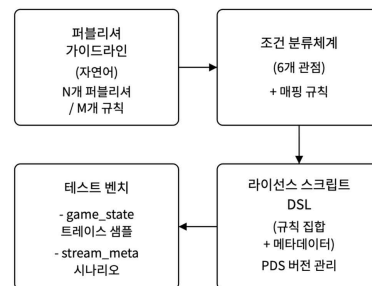


그림 2 평가 절차

Fig 2. Evaluation Procedure

5.2 표현력 평가

스토리 진행 및 부분 허용 라이선스 조건은 게임이 chapter_id, scenario_id, in_game_date, is_trial, progress_ratio 등을 제공하는 전제 하에 stateless 스크립트로 구현 가능함을 확인하였다. 예를 들어 "챕터 4 이후 금지"는 when game.chapter_id <= 4 then allow else block, "전체 50% 이하만 허용"은 when game.progress_ratio <= 0.5 then allow else block으로 표현된다.

시간 기반 라이선스 조건(발매 전 금지, 발매 후 N일 제한)은 스크립트 메타데이터에 release_date, embargo_until을 상수로 포함하고 stream.current_time과 비교하는 방식으로, 라이브/VOD 구분 라이선스 조건은 auto_archive_enabled, local_recording_enabled 플래그를 참조하는 방식으로 각각 stateless 표현이 가능하였다.

유료 스토리(DLC) 라이선스 조건은 content_paywall 플래그로, 수익화·행위자·플랫폼 라이선스 조건은 monetized, actor_type, platform 등 방송 메타데이터만으로 판정 가능하여 stateless 모델에 자연스럽게 포함된다.

멀티 엔딩·루트 라이선스 조건은 두 부류로 나뉜다. "진 엔딩 금지", "특정 루트 금지" 등 단순 금지/허용은 ending_id, route_id만으로 stateless 하게 구현 가능하다. 반면, "한 방송인이 여러 엔딩을 모두 공개하지 못하도록 제한"과 같은 정책은 방송인별 과거 이력에 의존하므로 stateful 확장이 필요하며, 이는 6장에서 별도로 분석한다.

5.3 운영 및 사용성 평가

5.3.1 퍼블리셔·개발자 관점

퍼블리셔·게임 개발자는 두 가지 주요 작업을 수행해야 한다.

게임 내 상태 태깅: 챕터/에피소드, 시나리오 세그먼트, DLC 구간, 엔딩·루트, 스포일러 민감도 등을 데이터 테이블 또는 시나리오 스크립트에 태깅하는 작업이 필요하다. 이미 내부적으로 비슷한 구조를 갖춘 스토리 게임일수록 추가 비용은 낮았으며, 기존에 "녹화 금지 구간" 등을 구현한 타이틀에서는 재활용 가능한 정보가 많았다.

라이선스 스크립트 작성·관리: 2장에서 분석한 자연어 가이드라인을 DSL 규칙으로 옮기는 작업

표 2. 조항별 평가 결과
Table 2. Evaluation about each clauses

관점(2장)	대표 정책(예)	필수 입력 필드	제어성공여부
스토리 진행	초반 20분만 허용	is_trial, elapsed_minutes	0
시간	발매 후 30일 엔딩 금지	ending_id, current_time(UTC), release_date	0
유료 스토리	DLC 금지	is_dlc_content	0
라이브/VOD	라이브 허용+VOD 금지	story_segment_id, auto_archive_enabled, local_recording_enabled	0
수익화/행위자/플랫폼	파트너만 수익화 허용	monetized, actor_type, contract_type, platform	0

은 초기에는 일부 학습 비용이 필요하지만, 패턴이 반복되기 때문에 게임사 내부 템플릿을 만들면 재사용성이 높았다. 정책 변경(예: 발매 후 N일 경과 후 제한 완화, DLC 출시 등)은 PDS에서 스크립트 버전을 갱신하는 방식으로 반영할 수 있어, 게임 클라이언트를 패치하지 않고도 세부 라이선스 조건을 수정할 수 있었다.

전반적으로, 스토리 데이터가 구조화되어 있는 게임을 기준으로 할 때, 초기 스크립트 설계 비용은 있지만, 이후 유지·운영 비용은 비교적 안정적이라는 점을 확인하였다.

5.3.2 방송인 관점

방송인(스트리머)는 기존 방송 흐름에 다음과 같은 요소가 추가된다:

- 방송 시작 시, 플러그인이 PDS와 통신하여 라이선스 스크립트를 자동으로 로딩
- 라이선스 조건 구간 진입 시, 화면·음성이 자동으로 전환되거나, 자동 아카이브/녹화 설정이 허용 범위 밖이면 경고 메시지가 표시됨

실험 단계에서 관찰한 바, 시스템상의 프로세스는 추가되지만, 방송인의 활동, 즉 일반적인 게임 플레이·방송 과정에는 큰 변화를 주지 않으면서, 라이선스 조건 구간에서만 제한이 자동으로 적용되도록 동작하는 것이 가능하였다.

다만, 자동 아카이브를 허용한 상태로 후반 스포일러 구간에 진입했을 때 방송이 차단되거나 경고가 반복되는 상황은 방송인 입장에서 "예상치 못한 방해"로 느껴질 수 있다.

6. 라이선스 조건 유형별 표현 가능성과 Stateless 한계 분석

6.1 기술적 제약에 따른 조건의 표현 영역

Stateless 표현 가능: 단일 시점 t 의 $game_state(t)$ 와 $stream_meta(t)$ 만으로 허용/차단 여부를 결정할 수 있는 경우이다. 2장에서 정리한 대부분의 라이선스 조건이 여기에 속한다. 스토리 진행 기반(챕터 컷오프, 인게임 날짜, 체험판/초반부 허용), 시간 기반(발매 전 금지, 발매 후 N일 제한), 유료 스토리(DLC 금지), 라이브/VOD 구분, 수익화·행위자·플랫폼 조건, 엔딩/루트 단순 금지 등은 모두 현재 값만으로 판정 가능하다.

상태 저장 필요: 방송인이 과거에 무엇을 방송했는지 등 세션 간 누적 이력이 필요한 경우이다. "여러 엔딩 중 한 방송인에게 하나만 허용"과 같은 정책은 ($game_id$, $streamer_id$)를 키로 하여 이미 공개한 $ending_id$ 를 참조해야 한다.

시스템 범위 밖: 형식·편집 여부, 제3자 재업로드, 시청자 지역 등 GCM·BCP·PDS 구조로는 관측할 수 없는 정보에 의존하는 조건이다. 무편집·무코멘터리 금지, 컷신/음악만 모은 영상 금지 등은 영상·음성 내용 분석을 요구하며, 정책·약관·사후 저작권 집행에 의존해야 한다.

6.2 상태 기반 확장의 설계 방향

방송인별 상태 저장 위치로는 게임 내부 저장과 방송 시스템(BCP/PDS) 저장이 있다. 전자는 여러 방송인 식별이 어렵고 침습적이므로, 본 논문에서는 게임을 방송 전용으로 수정하지 않는다는 원칙에 따라 BCP/PDS에 ($streamer_id$, $game_id$, $ending_id$) 조합을 저장하는 방식을 권장한다.

7. 저작권 관점 분석

2장에서 살펴본 국내외 게임 방송 가이드라인은 모두 저작권자·퍼블리셔가 자신의 저작물(게

임·스토리·음악)의 공개 범위와 방식을 통제하기 위한 규범이라는 점에서, 법적으로는 이용허락(라이선스) 조건에 해당한다[1][4][7]. 3-6장에서 제안한 시스템은 이러한 조건을 기계가 실행 가능한 라이선스 스크립트로 형식화하고, 실시간으로 집행하며, 그 과정을 로그로 남긴다. 이는 라이선스 위반 여부를 판정하기 위한 증거 구조를 제공한다는 점에서 의미가 있다.

7.1 라이선스 스크립트의 법적 위치: 실행 가능 라이선스

2장에서 살펴본 게임 방송 가이드라인은 법적으로 이용허락(라이선스) 조건에 해당한다[1][4][7]. 본 논문의 라이선스 스크립트는 이러한 자연어 규정을 게임 상태(game_state)와 방송 메타데이터(stream_meta)에 대한 조건식과 제어 명령으로 변환하여, 자동화 할 수 있게 한 것이다.

라이선스 스크립트는 기술적으로는 게임 - 방송 소프트웨어 - PDS 사이에서 공유되는 정형 규칙 집합이며, 법적·계약적으로는 자연어 계약의 "기술적 부속 문서" 또는 실행 가능 라이선스 조항에 해당한다. 라이선스 위반 판정에서 중요한 포인트는 "어떤 시점에 어떤 조건이 유효했는가", "스크립트가 자연어 가이드라인의 의도 범위 내에서 작성되었는가", "방송인이 해당 조건에 동의한 상태에서 방송했는가"이다.

7.2 저작권 보호 관점에서의 기능

라이선스 스크립트의 제어 내용은 저작권법상의 공중송신/전송권(라이브 스트리밍), 복제권·배포권(VOD, 아카이브), 2차적 저작물 작성권(코멘터리·편집 영상)과 밀접하게 대응한다[4][7][17][20].

스토리 중심 게임에서 엔딩·최종장, 히든 엔딩·루트, 유료 DLC·에프터 스토리는 콘텐츠의

핵심 가치를 구성한다. 라이선스 스크립트는 이러한 규칙을 "게임 상태와 방송 메타데이터"에 대한 명시적 조건으로 표현하고 실시간 제어함으로써, "마케팅·커뮤니티를 위한 허용 범위"와 "구매자 전용 영역"을 시스템 차원에서 구분한다.

7.3 라이선스 위반 판정을 위한 증거 구조

라이선스 스크립트 기반 시스템이 라이선스 위반 판정에 활용되려면, 라이선스 스크립트 메타데이터와 방송 세션 로그가 체계적으로 남아야 한다. 이러한 정보가 있다면 특정 시점의 상황을 재구성하여 "시스템이 어떤 결정을 했는지, 그것이 스크립트·계약과 일치했는지"를 기술적으로 검증할 수 있다[6].

한편, 로그는 방송인의 활동 이력을 담고 있으므로, 위변조 방지(서명·해시·타임스탬프), 보존 기간·접근 권한 설정 등의 고려가 필요하다.

8. 결론 및 향후 연구

본 논문은 라이선스 스크립트 기반 제어 시스템이 실제 퍼블리셔의 게임 방송 가이드라인을 어느 정도까지 구현할 수 있는지를 검증하였다. 다양한 퍼블리셔의 가이드라인을 분석하여 라이선스 조건 분류체계를 도출하고, 게임 상태와 방송 메타데이터를 입력으로 하는 stateless 평가 모델을 설계·구현하였다. 평가 결과, 스토리 진행 기반 컷오프, 발매 전후 기간 제한, 유료 DLC 보호, 라이브/VOD 구분, 수익화·행위자·플랫폼 조건 등 대다수의 현실적 라이선스 조건이 stateless 모델로 표현 가능함을 확인하였다.

반면, "방송인별로 하나의 엔딩만 허용"과 같이 세션 간 이력에 의존하는 조건은 stateful 확장이 필요하며, 형식·편집 요구나 재업로드 금지 등은 시스템 관측 범위 밖임을 규명하였다. 본

연구는 stateless 평가 모델에 한정되어 있고, 자연어 가이드라인과 스크립트 사이의 의미 차이 분석, 대규모 서비스 환경에서의 성능 검증 등은 향후 과제로 남는다.

향후 연구는 stateful 확장 모델의 설계·검증, 메타데이터·식별자 표준화, 자연어 - 스크립트 간 정합성 확보, 다수 권리자 스크립트 합성, 테스트·검증 도구 개발, 법제도와 연계 등으로 확장될 수 있다.

본 논문은 라이선스 스크립트와 실시간 제어 프로토콜을 통해, 스토리 중심 게임의 게임 방송 가이드라인을 기계가 해석하고 집행 가능한 구조로 구체화하였다. 라이선스 스크립트 기반 시스템이 실시간 스포일러 제어, 저작권·판매 보호, 라이선스 위반 판정 지원이라는 세 관점에서 실질적인 도입 가능성을 가진다는 것을 시사하며, 게임 방송 환경에서 저작권 보호와 창작·커뮤니티 활동 사이의 균형을 보다 정교하게 설계할 수 있는 기반을 제공한다.

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2024년도 신기술 융합 저작권 기술개발 사업으로 수행되었음(과제명 : Web3.0 탈중앙화 환경에서 창작자간의 저작권 이용허락 거래 자동화 기술 개발, 과제번호 : RS-2024-00441360, 기여율 : 100%)

참 고 문 헌

[1] Game Rating and Administration Committee; Korea Association of Game Industry; Korea Copyright Protection Agency; Korea Copyright Commission; Korea Creative Content Agency, "Guide for Responding to Illegal Games and

Game Copyright Infringement", Sep. 2022, Korea Creative Content Agency, <https://www.kcopa.or.kr/download.do?uuiid=88ffd019-7496-449c-8fd7-2efcbf5c01fa.pdf>

[2] S. Baik, "A Storytelling Methodology for Game Content Planning: Focusing on a Cultural-Semiotic Methodology", *Revue d'Etudes Francaises*, 59, pp.267-284, 2007

[3] C.-I. Park, "A Study on Interactive Storytelling Using Animations and Computer Games", *Journal of Korea Design Forum*, 17, pp.189-200, 2007

[4] Toki Kawase, "YouTubers Need to Know: When 'Let's Play' Videos Become Illegal Under Japanese Copyright Law", <https://monolith.law/en/general-corporate/youtuber-game-law>

[5] ATLUS, "Persona 5 Royal Video & Livestreaming Guidelines", <https://atlus.com/persona-5-royal-video-livestreaming-guidelines/>, Jun. 2023.

[6] R. Stanton, "Japanese court throws the book at YouTuber who uploaded copyrighted game clips and spoilers: Two years in the slammer and ¥1m fine", *PC Gamer*, Sep, 8, 2023. <https://www.pcgamer.com/japanese-court-throws-the-book-at-youtuber-who-uploaded-copyrighted-game-clips-and-spoilers-two-years-in-the-slammer-and-yen1m-fine/>

[7] H. Shin, "Regulation of Game Let's Play Videos Under Japanese Copyright Law", Jun. 2023, Korea Copyright Protection Agency, <https://kcopa.or.kr/download.do?uuiid=698e00d0-b278-442c-b502-6267d753d3a3.pdf>

[8] ConcernedApe, "Stardew monetization policy for Twitch Streamers, YouTubers, etc.", *Stardew Valley Official*. <https://www.stardewvalley.net/stardew-monetization-policy-for-twitch-streamers-youtubers-etc/>

[9] T. Cavanagh, "Video Policy", *Distractionware*. <https://distractionware.com/blog/video-policy/>

- [10] Unknown Worlds, “Video Policy”, Unknown Worlds Entertainment. <https://unknownworlds.com/video-policy>
- [11] sen (Mad Father), “Let’s Play Guidelines/ 實況ガイドライン”, Steam Community, 2016. <https://steamcommunity.com/app/483980/discussions/0/350542683210415300/>
- [12] C. D’Anastasio, “YouTubers Clash With Visual Novel Developers Over Spoiler Videos”, Kotaku, Nov. 2016. <https://kotaku.com/youtubers-clash-with-visual-novel-developers-over-spoil-1788145960>
- [13] Sekai Project, “Streaming Policy”, CLANNAD General Discussions, Steam Community. <https://steamcommunity.com/app/324160/discussions/0/485622866452750480/>
- [14] Playism, “Maid Cafe on Electric Street is Out Now!”, SteamDB Patch Notes. <https://steamdb.info/patchnotes/16129702/>
- [15] Playism, “Streaming Guidelines for PLAYISM Titles”, Playism News, Jun. 2021. https://playism.com/en/news/2021/0601/streamer_guidelines/
- [16] r/visualnovels, “PSA: Official Localization Company Policies related to Visual Novels”, Reddit, 2020. https://www.reddit.com/r/visualnovels/comments/gkdzi3/psa_official_localization_company_policies/
- [17] Dramatic Create, “Streaming Guideline”, Dramatic Create Official. <https://dramaticcreate.com/1/en/guideline.html>
- [18] Jellyfish Parade, “Video Policy”, itch.io. <https://jellyfishparade.itch.io/>
- [19] CCMC Corporation, “Game Streaming Guidelines (holo Indie)”, CCMC Corporation. <https://ccmc-corp.com/en/terms/>
- [20] Nintendo, “Nintendo Game Content Guidelines for Online Video & Image Sharing Platforms”, Nintendo. https://www.nintendo.co.jp/networkservice_guideline/en/index.html
- [21] Square Enix, “FINAL FANTASY VII REBIRTH Guidelines for User Generated Content”, Square Enix. <https://sqex.to/vii-rebirth-ugc>
- [22] Y. Joe, S. Jang, D. Shin, “Real-time Game - Broadcast Guidance Protocol for Spoiler Prevention in Live Game Streaming”, Journal of Web Engineering, 24(8), 2025, DOI: 10.13052/jwe1540-9589.2484
- [23] Twitch, “How To File a DMCA Notification or Counter-notification”, Twitch Legal. <https://www.twitch.tv/p/en/legal/dmca-guidelines/>
- [24] S. Jeon, S. Kim, J. Yu, “SpoilerNet: A Deep Learning Approach for Spoiler Detection in User Reviews”, Proceedings of the 13th International Conference on Web and Social Media (ICWSM), pp.272-280, 2019.

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



홍두표(Du-Pyo Hong)

2024.2 숭실대학교 컴퓨터학과 석사
2024.1-현재 : 엘에스웨어(주) 주임연구원
<주관심분야> 클라우드, 빅데이터, 블록체인



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



장성일(Sung-Il Jang)

2019.8 숭실대학교 컴퓨터학과 석사
2021.8 숭실대학교 소프트웨어학과 박사수료
2021.9-현재 : 엘에스웨어(주) 수석연구원
<주관심분야> 시스템 프로그래밍, 분산 컴
퓨팅, 블록체인