

DIRECT P2P

P2P 통신 영상전송 / 편집 서비스

3조 E동I들

팀원: 김민수, 이수진, 최하경

1. 프로젝트 개요

1.1 서비스 목적

본 프로젝트는 UDP 홀펀칭(UDP Hole Punching) 기술을 활용하여 NAT 환경에서도 두 클라이언트가 중앙 서버를 경유하지 않고 직접 P2P 연결을 수립하고, 콘텐츠 데이터(주요 고용량 고화질 영상)를 고속으로 송수신 및 즉석 편집할 수 있는 전송 서비스를 구축하는 것을 목적으로 합니다.

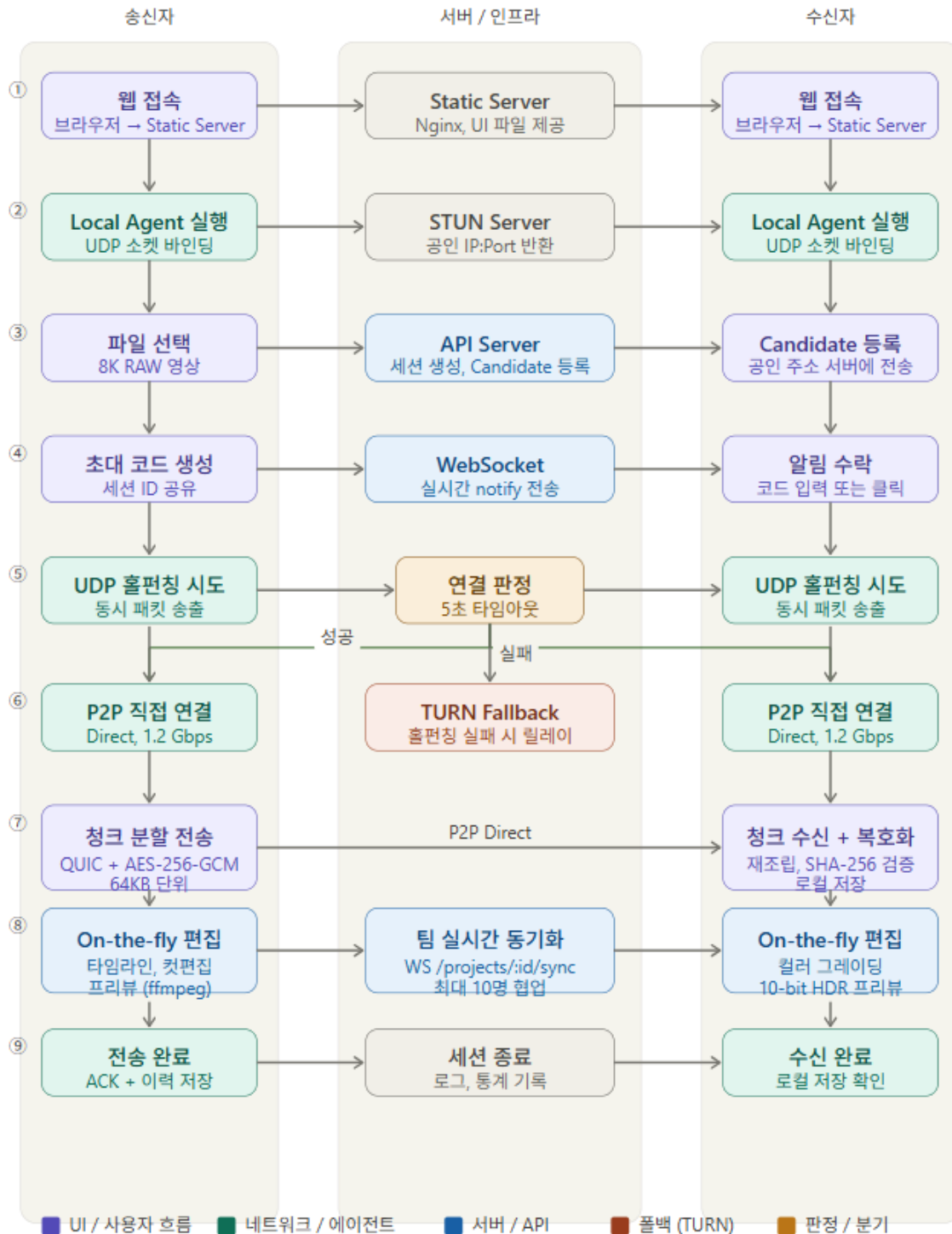
주요 기술 스택

- Backend: Go (매니징 서버), Go (클라이언트 통신 모듈)
- Protocol: UDP, TURN, STUN

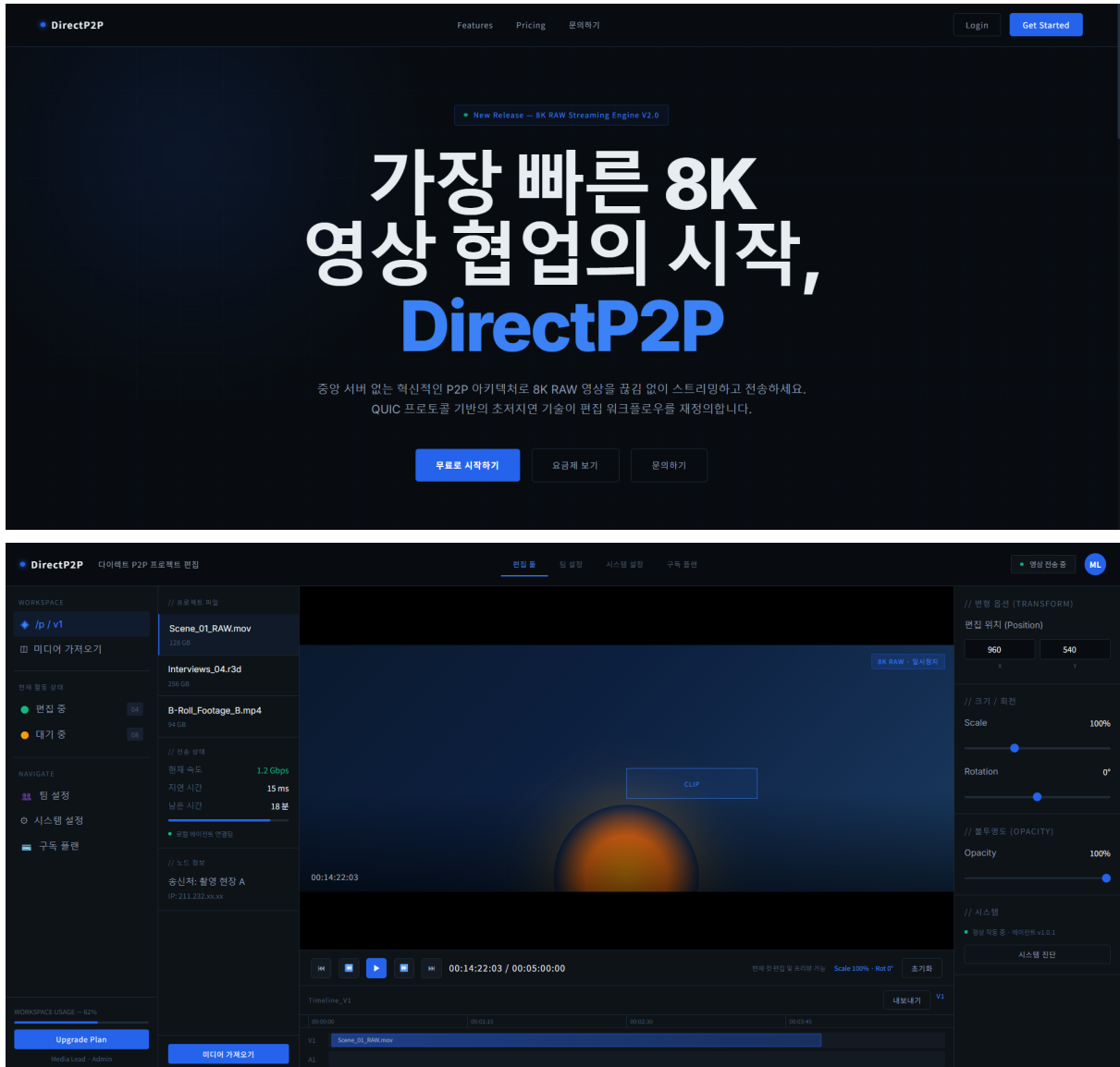
1.2 시스템 구성 요소

컴포넌트	역할 요약	주요 기술
Static Server	웹 프론트엔드 정적 파일 및 에이전트 설치파일 제공	Nginx / CDN
API Server	사용자 관리, 세션 관리, 피어 주소 교환(Signaling)	REST API / WebSocket
STUN Server	클라이언트 홀펀칭을 위해 외부 주소를 제공	Google Public STUN
TURN Server	홀펀칭 실패 시 데이터 릴레이 폴백 제공	TURN Protocol (RFC 5766)
Local Agent	UDP 소켓 관리, 홀펀칭 수행, 콘텐츠 송수신	Go
Web Frontend	연결 상태 및 전송 진행 현황 시각화 UI	React / HTML5

2. 사용자 시나리오



3. UI(예상도)



4. 사용할 기술스택

4.1 사용할 언어

- Backend: go
- Frontend: React
- Database: sqlite

4.2 사용 오픈소스

- Gin
- go-sqlite3
- Google Public STUN 서버

- 별도의 서버 구축 없이 내 공인 IP가 뭔지 테스트해보고 싶을 때 바로 쓸 수 있는 무료 API
 - 구글의 API이기 때문에 전 세계 어느 곳에서든 응답속도가 좋다.
- libp2p (<https://github.com/libp2p/go-libp2p>)
 - 출판징 로직과 피어 간 상태 관리를 도와주는 라이브러리
- coturn (<https://github.com/coturn/coturn>)
 - TURN server 사용을 위한 오픈소스
- Nginx
 - 정적 콘텐츠 서빙
- ffmpeg
 - 동영상 파일 트랜스코딩
- jenkins
 - 자동화 빌드 & 배포

5. 주요 기능 및 요구사항

- 주소 식별 및 공유
 - 공인 IP/Port 추출: 클라이언트는 외부 STUN 서버에 요청을 보내 자신의 NAT 외부 주소를 파악한다.
 - 피어 등록 (Registration): Go로 작성된 중앙 서버에 자신의 ID와 파악된 공인 주소를 등록한다.
 - 상대방 조회 (Discovery): 연결하고자 하는 대상의 ID를 검색하여 상대방의 공인 주소 정보를 받아온다.
- 출판징 로직
 - 동시 패킷 송출 (Simultaneous Punching): Go언어로 작성된 클라이언트 모듈이 양방향에서 서로의 공인 주소로 UDP 패킷을 전송한다.
 - 상태 유지 (Keep-alive): NAT 매핑 테이블이 만료되어 구멍이 닫히지 않도록 주기적으로 더미(Dummy) 패킷을 송신한다.
 - 연결 재시도 (Retry): 패킷 손실로 인해 초기 출판징 실패 시, 지수 백오프(Exponential Backoff) 알고리즘을 적용해 재시도한다.
- QUIC 기반 고속 데이터 전송
 - QUIC 스트림 생성: UDP 출판징이 성공하여 세션이 확립되면, 그 위에 QUIC 프로토콜을 얹어 멀티플렉싱 스트림을 생성한다.
 - 혼잡 제어 및 손실 복구: UDP의 단점인 데이터 유실 문제를 QUIC의 자체적인 에러 정정(FEC) 및 재전송 메커니즘으로 해결하여 신뢰성을 확보한다.
 - 연결 마이그레이션(Connection Migration): 클라이언트의 네트워크 환경(Wi-Fi ↔ LTE)이 변해 IP가 바뀌더라도, QUIC의 Connection ID를 이용해 끊김 없는 P2P 통신을 유지한다.
 - 보안 강화 (TLS 1.3): 모든 P2P 전송 데이터를 QUIC 내장 암호화 표준인 TLS 1.3으로 보호하여 스니핑을 방지한다.
- On-the-fly 영상편집
 - 영상의 경우, 전송 중간에 미리 확인하고 필요하다면 간단한 컷편집을 할 수 있다.

6. 기능명세서

시스템 구성 요소

컴포넌트	역할 요약	주요 기술
Static Server	웹 프론트엔드 정적 파일 및 에이전트 설치파일 제공	Nginx / CDN
API Server	사용자 관리, 세션 관리, 피어 주소 교환(Signaling)	REST API / WebSocket
TURN Server	홀펀칭 실패 시 데이터 릴레이 폴백 제공	TURN Protocol (RFC 5766)
Local Agent	UDP 소켓 관리, 홀펀칭 수행, 콘텐츠 송수신, 영상 스트리밍 생성	Go
Web Frontend	연결 상태 및 전송 진행 현황 시각화 UI	React / HTML5

6.1 Static Server

6.1.1 기능 개요

Static Server는 사용자가 웹 브라우저를 통해 P2P 콘텐츠 네트워크 서비스에 접근할 수 있도록 웹 프론트엔드 정적 파일(HTML, CSS, JS)을 제공하며, 로컬 에이전트 설치 파일의 다운로드 엔드포인트도 함께 제공합니다.

6.1.2 상세 기능 명세

정적 파일 서빙

기능 ID	기능명	설명	우선순위
SS-001	웹 앱 서빙	React/Vue/HTML 기반 SPA 번들 파일 제공 (index.html, JS, CSS, 이미지)	필수
SS-002	HTTPS 지원	TLS/SSL 인증서 적용, HTTP → HTTPS 자동 리다이렉트	필수
SS-003	캐시 제어	정적 자원에 Cache-Control 헤더 설정 (JS/CSS: max-age=31536000, HTML: no-cache)	권장
SS-004	GZIP/Brotli 압축	전송 용량 최소화를 위한 응답 압축 처리	권장
SS-005	SPA 라우팅 처리	모든 경로 요청을 index.html로 fallback (클라이언트 사이드 라우팅 지원)	필수

에이전트 설치 파일 제공

기능 ID	기능명	설명	우선순위
SS-010	플랫폼별 에이전트 배포	Windows(.exe), macOS(.dmg/.pkg), Linux(.deb/.rpm) 설치 파일 다운로드 URL 제공	필수
SS-011	버전 메타데이터 제공	최신 에이전트 버전 정보 JSON API 엔드포인트 (/agent/latest.json)	권장
SS-012	파일 무결성 검증	SHA-256 체크섬 파일(.sha256) 함께 제공하여 다운로드 무결성 보장	권장
SS-013	다운로드 통계	에이전트 다운로드 횟수 및 플랫폼별 통계 수집 (익명)	선택

6.1.3 API 엔드포인트

Method	경로	설명
GET	/	웹 프론트엔드 SPA 제공
GET	/agent/download/:platform	플랫폼별 에이전트 설치 파일 다운로드 (windows/mac/linux)
GET	/agent/latest.json	최신 에이전트 버전 메타데이터 반환

6.2 API Server

6.2.1 기능 개요

API Server는 사용자 인증/관리, 세션 수립, 피어 간 주소(Candidate) 교환(Signaling), 그리고 연결 상태 모니터링 등 P2P 연결 성립에 필요한 모든 신호 교환 기능을 담당합니다. 웹 프론트엔드 및 로컬 에이전트와 REST API 및 WebSocket을 통해 통신합니다.

6.2.2 사용자 관리

인증 기능

기능 ID	기능명	설명	우선순위
API-001	회원가입	이메일/비밀번호 기반 계정 생성. 이메일 인증 발송	필수
API-002	로그인	이메일/비밀번호 검증 후 랜덤 Token 발급	필수
API-003	로그아웃	Token 무효화 처리	필수
API-004	비밀번호 변경	현재 비밀번호 확인 후 새 비밀번호로 변경	권장
API-005	회원 탈퇴	계정 및 관련 데이터 삭제 처리	권장

사용자 정보 관리

기능 ID	기능명	설명	우선순위
API-006	프로필 조회	사용자 이름, 이메일, 가입일, 에이전트 연결 상태 조회	필수
API-007	프로필 수정	닉네임, 프로필 이미지 등 수정	권장
API-008	에이전트 등록	로컬 에이전트가 최초 실행 시 사용자 계정에 디바이스 등록	필수
API-009	에이전트 목록 조회	사용자 계정에 연결된 에이전트 디바이스 목록 반환	필수

6.2.3 피어 주소 교환 (Signaling)

세션 관리

기능 ID	기능명	설명	우선순위
API-010	전송 세션 생성	송신자가 세션 생성 요청 → 세션 ID 및 초대 코드 발급	필수
API-011	세션 참여	수신자가 세션 ID/초대 코드로 세션에 참여	필수
API-012	세션 상태 조회	세션의 현재 상태 조회 (대기/연결중/연결완료/종료)	필수
API-013	세션 종료	교환 완료 또는 오류 시 세션 정리 및 상태 업데이트	필수
API-014	세션 이력 조회	사용자의 과거 전송 세션 목록 및 결과 조회	권장

Candidate 주소 교환

기능 ID	기능명	설명	우선순위
API-020	Candidate 등록	에이전트가 STUN으로 획득한 공인 IP:Port를 서버에 등록	필수
API-021	Candidate 조회	상대방 에이전트의 Candidate 주소 목록 조회	필수
API-022	WebSocket 실시간 알림	Candidate 등록/갱신 이벤트를 상대방에게 실시간 Push	필수
API-023	TURN 자격증명 발급	출판칭 실패 대비 TURN 서버 접속용 임시 자격증명 발급	필수

6.2.4 API 엔드포인트 목록

Method	경로	인증	설명
POST	/auth/register	없음	회원가입
DELETE	/auth/me	Token	회원탈퇴
POST	/auth/login	없음	로그인 (Token 발급)
POST	/auth/logout	Token	로그아웃
GET	/users/me	Token	내 프로필 조회
GET	/users/me/agents	Token	등록된 에이전트 목록

POST	/sessions	Token	전송 세션 생성
POST	/sessions/:id/join	Token	세션 참여
GET	/sessions/:id	Token	세션 상태 조회
DELETE	/sessions/:id	Token	세션 종료
POST	/sessions/:id/candidates	Token	Candidate 주소 등록
GET	/sessions/:id/candidates	Token	상대방 Candidate 조회
POST	/sessions/:id/turn-credentials	Token	TURN 자격증명 발급
WS	/ws	Token	WebSocket 실시간 이벤트 채널

6.2.5 WebSocket 이벤트

이벤트명	방향	설명
session:peer_joined	서버 → 클라이언트	상대방이 세션에 참여함
session:candidate_added	서버 → 클라이언트	상대방의 새 Candidate 주소가 등록됨
session:status_changed	서버 → 클라이언트	세션 상태 변경 알림
session:peer_disconnected	서버 → 클라이언트	상대방 연결 끊김
agent:ping	클라이언트 → 서버	에이전트 생존 확인 Heartbeat

6.3 TURN Server

6.3.1 기능 개요

TURN(Traversal Using Relays around NAT) 서버는 UDP 홀펀칭이 실패하는 엄격한 NAT 환경(Symmetric NAT 등)에서 두 피어 간의 데이터 릴레이를 담당합니다. ICE-like Fallback 메커니즘을 통해 P2P 직접 연결이 불가능한 경우에도 서비스 연결 보장을 제공합니다.

6.3.2 상세 기능 명세

기능 ID	기능명	설명	우선순위
TURN-001	릴레이 할당 (Allocate)	피어 요청 시 TURN 서버 측 릴레이 포트(Relayed Transport Address) 할당	필수
TURN-002	권한 부여 (CreatePermission)	특정 피어 IP에 대한 데이터 송수신 권한 등록	필수
TURN-003	채널 바인딩 (ChannelBind)	오버헤드 절감을 위한 채널 번호 기반 빠른 릴레이 경로 수립	권장
TURN-004	데이터 릴레이 (Send/Data)	한쪽 피어에서 수신한 데이터를 상대 피어에게 중계 전달	필수
TURN-005	자격증명 검증	API Server가 발급한 임시 자격증명(HMAC-SHA1 기반) 유효성 검증	필수
TURN-006	할당 갱신 (Refresh)	활성 릴레이 할당의 만료 시간 갱신 처리	필수
TURN-007	UDP/TCP 이중 지원	UDP 기반 릴레이 우선, TCP 폴백 지원	권장
TURN-008	대역폭 제한	피어당 릴레이 대역폭 상한 설정으로 서버 과부하 방지	권장
TURN-009	연결 통계 수집	릴레이 세션 수, 전송 바이트, 오류율 등 지표 수집	선택

6.3.3 자격증명 발급 방식

TURN 임시 자격증명 생성 방식 (Time-Limited Credentials)

- username = 만료 Unix Timestamp + ':' + 사용자 ID (예) '1700000000:user123'
- password = HMAC-SHA256(secret_key, username) → Base64 인코딩
- 유효 시간 = API Server 발급 시점으로부터 최대 1시간
- API Server가 발급 → 에이전트가 TURN 인증에 사용 → TURN Server가 검증

6.3.4 Fallback 동작 조건

조건	동작
UDP 홀펀칭 성공	P2P 직접 연결 유지, TURN 미사용
홀펀칭 실패 (Timeout 5초)	TURN 릴레이 자동 전환 (Fallback)
TURN 연결 중 P2P 연결 성공	P2P로 자동 업그레이드 (ICE-like)
TURN 서버 장애	에러 알림 및 재시도 최대 3회

6.4 Local Agent (클라이언트)

6.4.1 기능 개요

Local Agent는 사용자의 디바이스에 설치되는 경량 데몬 프로세스입니다. **UDP** 소켓을 관리하고, **STUN** 서버를 통해 공인 주소를 파악하며, **UDP** 흘편칭으로 상대방과 직접 **P2P** 연결을 수립합니다. 연결 수립 후 콘텐츠 데이터를 청크 단위로 분할하여 고속 전송합니다. 웹 프론트엔드와 로컬 **HTTPS API**로 통신합니다.

6.4.2 설치 및 초기화

기능 ID	기능명	설명	우선순위
AGT-001	플랫폼별 설치	Windows/macOS/Linux 네이티브 설치 패키지 지원	필수
AGT-002	시스템 시작 시 자동 실행	OS 부팅 시 에이전트 자동 시작 (Windows: 서비스, macOS/Linux: Daemon)	권장
AGT-003	로컬 HTTP API 서버	웹 프론트엔드와 통신하기 위한 로컬 포트(예: 17432) HTTP 서버 구동	필수
AGT-004	계정 연동	API Server에 로그인 후 발급받은 토큰으로 디바이스 등록	필수
AGT-005	자동 업데이트 확인	Static Server에서 최신 에이전트 버전 확인 및 업데이트 안내	권장

6.4.3 UDP 소켓 관리 및 STUN

기능 ID	기능명	설명	우선순위
AGT-010	UDP 소켓 바인딩	로컬 UDP 포트 바인딩 및 소켓 풀 관리	필수
AGT-011	STUN 요청	STUN 서버에 Binding Request 전송 → 공인 IP:Port(Mapped Address) 획득	필수
AGT-012	Candidate 수집	로컬 IP, STUN 획득 공인 IP 등 복수의 Candidate 주소 수집	필수
AGT-013	Candidate 등록	수집한 Candidate를 API Server에 등록	필수
AGT-014	NAT 유형 감지	Full Cone / Restricted / Port Restricted / Symmetric NAT 유형 판별	권장
AGT-015	Keepalive 유지	NAT 매핑 유지를 위한 주기적 UDP Keepalive 패킷 전송 (기본: 30초)	필수

6.4.4 UDP 흘편칭 및 연결 수립

기능 ID	기능명	설명	우선순위
AGT-020	상대방 Candidate 수신	API Server/WebSocket을 통해 상대방 Candidate 목록 수신	필수
AGT-021	동시 흘편칭 시도	수신한 모든 Candidate 주소로 동시에 UDP 패킷 전송하여 NAT 구멍 뚫기	필수

AGT-022	연결 성공 판정	양방향 UDP 패킷 수신 확인 시 P2P 연결 수립 완료 판정	필수
AGT-023	TURN Fallback 전환	출편칭 5초 타임아웃 시 TURN 릴레이 연결로 자동 전환	필수
AGT-024	연결 품질 측정	RTT(왕복 지연) 및 패킷 손실을 측정 및 리포트	권장
AGT-025	P2P 재연결	연결 끊김 감지 시 자동 재연결 시도 (최대 3회)	권장

6.4.5 콘텐츠 송수신

데이터 전송 (송신측)

기능 ID	기능명	설명	우선순위
AGT-030	파일/데이터 선택	로컬 파일 또는 스트림 데이터 전송 대상 지정	필수
AGT-031	청크 분할 전송	데이터를 고정 크기 청크로 분할 (기본: 64KB)	필수
AGT-032	청크 시퀀스 관리	각 청크에 시퀀스 번호 부여, 순서 보장 및 재전송 처리	필수
AGT-033	전송 속도 제어	네트워크 상태에 따른 동적 송신 속도 조절 (혼잡 제어)	권장
AGT-034	전송 메타데이터 전송	파일명, 크기, MIME 타입, 체크섬(SHA-256) 선행 전송	필수
AGT-035	종단 간 암호화	전송 데이터 AES-256-GCM 암호화 처리 (중간자 공격 방어)	필수

데이터 수신 (수신측)

기능 ID	기능명	설명	우선순위
AGT-040	청크 수신 및 버퍼링	수신 청크를 메모리 버퍼에 저장, 순서 재조립	필수
AGT-041	누락 청크 재요청	누락된 청크 시퀀스 번호를 송신측에 재전송 요청 (NACK)	필수
AGT-042	데이터 복호화	수신 데이터 AES-256-GCM 복호화	필수
AGT-043	무결성 검증	전체 수신 데이터 SHA-256 체크섬 검증	필수
AGT-044	로컬 저장	수신 완료 데이터를 지정 경로에 저장	필수
AGT-045	수신 확인 응답	ACK 패킷 전송으로 송신측에 수신 완료 통보	필수

6.4.6 로컬 HTTP API

Method	경로	설명
GET	/status	에이전트 상태 및 연결 정보 반환
GET	/session	현재 전송 세션 정보 조회
POST	/session/start	전송 세션 시작 요청
POST	/session/join	세션 참여 요청

GET	/transfer/progress	현재 전송 진행률 및 속도 조회
POST	/transfer/cancel	전송 취소
GET	/peers	연결된 피어 목록 및 연결 상태
GET	/logs	에이전트 로그 조회
POST	/peers/:id	데이터 전송

6.5 Web Frontend

6.5.1 기능 개요

Web Frontend는 사용자가 브라우저를 통해 **P2P** 콘텐츠 네트워크 서비스를 이용할 수 있는 그래픽 인터페이스입니다. 전송 세션 관리, 연결 수립 과정 모니터링, 실시간 전송 진행 현황 표시를 핵심 기능으로 합니다. 로컬 에이전트와 **Local HTTPS API**로, **API Server**와는 **REST/WebSocket**으로 통신합니다.

6.5.2 화면 구성

화면명	경로	설명
홈/대시보드	/	서비스 소개, 에이전트 설치 안내, 빠른 시작
로그인	/login	이메일/비밀번호 로그인
회원가입	/register	계정 생성
에이전트 설치 가이드	/install	플랫폼별 에이전트 설치 안내 및 다운로드
전송 시작	/send	파일 선택 및 전송 세션 생성
전송 수신	/receive	세션 코드 입력 및 수신 대기
연결 모니터	/monitor	P2P 연결 과정 및 전송 현황 실시간 표시
전송 이력	/history	과거 전송 세션 목록 및 상세
설정	/settings	사용자 정보, 에이전트 설정 관리

6.5.3 상세 기능 명세

에이전트 연결 상태 표시

기능 ID	기능명	설명	우선순위
WEB-001	에이전트 감지	로컬 HTTP API(localhost:17432) 폴링으로 에이전트 실행 여부 감지	필수
WEB-003	설치 안내 팝업	에이전트 미설치/미실행 시 설치 가이드 팝업 표시	필수

전송 세션 관리

기능 ID	기능명	설명	우선순위
-------	-----	----	------

WEB-010	파일 선택	드래그앤드롭 또는 파일 탐색기로 전송 대상 파일 선택	필수
WEB-011	세션 코드 생성/공유	생성된 세션 초대 코드를 클립보드 복사 또는 QR코드로 공유	필수
WEB-012	세션 코드 입력	수신측에서 코드 입력 후 세션 참여	필수
WEB-013	상대방 확인	세션 참여 후 상대방 정보(닉네임) 표시 및 전송 시작 확인	필수

연결 과정 실시간 시각화 (핵심 기능)

기능 ID	기능명	설명	우선순위
WEB-020	연결 단계 표시	STUN 요청 → Candidate 교환 → 출판칭 → 연결완료/TURN 폴백 단계를 스텝 인디케이터로 표시	필수
WEB-021	STUN 결과 표시	획득한 공인 IP:Port 및 NAT 유형 표시	권장
WEB-023	출판칭 시도 로그	시도 중인 Candidate 주소 및 성공/실패 결과 로그 스트림 표시	권장
WEB-024	연결 경로 표시	P2P 직접 연결 또는 TURN 릴레이 경유 여부, RTT 표시	필수
WEB-025	연결 실패 안내	출판칭 실패 시 TURN 폴백 전환 알림 및 이유 설명	필수

전송 진행 현황 표시

기능 ID	기능명	설명	우선순위
WEB-030	전송 진행률 바	전체 데이터 대비 전송 완료 비율 Progress Bar로 표시	필수
WEB-031	전송 속도 표시	현재 전송 속도(MB/s) 및 평균 속도 실시간 표시	필수
WEB-032	남은 시간 예측	현재 속도 기반 예상 완료 시간(ETA) 표시	권장
WEB-033	전송량 표시	전송 완료 용량 / 전체 용량 (예: 45 MB / 100 MB) 표시	필수
WEB-034	완료 알림	전송 완료 시 성공 토스트 알림 및 결과 요약 표시	필수
WEB-035	전송 취소	진행 중 전송 취소 버튼 및 확인 다이얼로그	필수
WEB-036	오류 처리 표시	전송 중 오류 발생 시 오류 코드 및 재시도 안내	필수

전송 이력

기능 ID	기능명	설명	우선순위
WEB-040	이력 목록 조회	날짜, 파일명, 상대방, 전송 결과, 크기 목록 표시	권장
WEB-041	이력 상세 조회	선택한 전송의 연결 유형, 소요 시간, 평균 속도 등 상세 표시	선택

6.6 전체 연결 흐름 (End-to-End Flow)

단계	액터	동작 설명
1	송신자 Web	웹 페이지 접속
2	송신자 (Web + Agent)	웹 프론트엔드 → 로컬 에이전트에 세션 시작 요청
3	송신자 Agent	STUN 서버에 UDP 전송 → 공인 IP:Port 획득
4	API Server	송신자 Candidate 등록 완료
5	수신자 Web	웹 페이지 접속
6	수신자 (Web + Agent)	웹 프론트엔드 → 로컬 에이전트에 세션 시작 요청
7	수신자 Agent	STUN 서버에 UDP 전송 → 공인 IP:Port 획득
8	API Server	수신자 Candidate 등록 완료
9	송신자 Web	수신자 선택 → 수신자에게 websocket을 통해 notify 전송 그리고 초대 코드 생성
10	수신자 Web	websocket을 통해 온 파일 수신 알림 notify 수락 또는 초대 코드 입력
11	API Server	notify 수락 완료 확인 → 세션 종료 → 자원 정리
12	양측 Web	상대방 Candidate 수신 → Agent에게 전달
13	양측 Agent	상대방 Candidate 수신 → 동시에 UDP 출판칭 시도
14-A	양측 Agent (성공)	양방향 UDP 패킷 확인 → P2P 연결 수립
14-B	양측 Agent (실패)	TURN 자격증명 요청 → TURN 릴레이 연결 수립
15	Web Frontend	연결 수립 완료 표시 (P2P 직접 / TURN 릴레이 경로 표시)
16	송신자 Agent	청크 분할 → QUIC TLS 암호화 적용 → 순차 전송
17	수신자 Agent	청크 수신 → 재조립 → 복호화 → 무결성 검증 → 저장
18	Web Frontend	전송 진행률 / 속도 / ETA 실시간 표시
19	양측 Agent	전송 완료 확인 → 추후 빠른 전송 시작을 위해 연결 유지

7. 파일 구조

direct-p2p/	# 프로젝트 루트
.gitmodules	# 서브모듈 설정 (예: coturn 등 외부 의존성)
LICENSE	# 전체 프로젝트 라이선스
README.md	# 프로젝트 개요 및 사용법
cmd	# 실행 진입점 (각 컴포넌트의 main)
api-server	# API 서버 실행 엔트리
main.go	# API 서버 시작 (라우터, DB, WS 초기화)
client-agent	# 로컬 에이전트 실행 엔트리
main.go	# 에이전트 데몬 시작 (UDP, STUN, 로컬 API)
turn-server	# TURN 서버 실행 관련
run.sh / run.ps1	# coturn 실행 스크립트 또는 wrapper
API_Server	# 중앙 시그널링 / 관리 서버
LICENSE	# API 서버 설명 및 실행 방법
README.md	
handler	# HTTP / WebSocket 핸들러
auth.go	# 로그인/회원가입 처리
session.go	# 세션 생성/조회 API
ws.go	# WebSocket 이벤트 처리
service	# 비즈니스 로직 계층
auth_service.go	# 인증 로직
session_service.go	# 세션/피어 매칭 로직
repository	# DB 접근 계층
user_repo.go	# 사용자 DB 처리
session_repo.go	# 세션 데이터 저장/조회
middleware	# 공통 미들웨어
auth_middleware.go	# JWT 인증 처리
cors.go	# CORS 설정
Client_Agent	# 사용자 로컬에서 실행되는 에이전트
LICENSE	# 에이전트 동작 및 설치 설명
README.md	
controller	# 로컬 HTTP API (프론트와 통신)
api.go	# /status, /session 등 엔드포인트
puncher	# 네트워크 핵심 로직
stun.go	# STUN 요청 (공인 IP 확인)
hole_punch.go	# UDP 홀펀칭 구현
crypto	# 암호화 처리
aes_gcm.go	# AES-256-GCM 암호화
worker	# 백그라운드 작업
keepalive.go	# NAT 유지용 패킷 전송
monitor.go	# 연결 상태 모니터링
Coturn	# TURN 서버 (fallback relay)
LICENSE	# coturn 사용 및 설정 설명
README.md	
config	# TURN 서버 설정 파일
turnserver.conf	

Frontend	# 웹 프론트엔드 (React SPA)
LICENSE	
README.md	# 프론트 실행 방법
package.json	# npm 의존성 및 스크립트
public	# 정적 파일 (index.html 등)
src	
api	# API 호출 로직
api.js	# API Server 호출
agent.js	# Local Agent 호출
components	# 재사용 UI 컴포넌트
ProgressBar.jsx	# 전송 진행률 표시
StatusIndicator.jsx	# 연결 상태 표시
pages	# 주요 페이지
Dashboard.jsx	# 메인 화면
Monitor.jsx	# 연결 상태 시각화
Settings.jsx	# 사용자 설정
Nginx	# 정적 파일 서버 및 리버스 프록시
LICENSE	
README.md	# nginx 설정 설명
conf	
nginx.conf	# SPA 서빙 + API 프록시 설정
configs	# 공통 설정 파일
config.yaml	# 서비스 설정 (포트, DB 등)
.env	# 환경 변수 (비밀값 포함 가능)
deployments	# 배포 관련
docker	# Dockerfile / docker-compose
scripts	# 배포 및 설치 스크립트
install.sh	
install.ps1	
api	# API 명세
openapi.yaml	# Swagger / OpenAPI 스펙

8. CI/CD 설계

8.1 DirectP2P 메인 리포지토리

여러 하위 프로젝트를 `git submodule` 형태로 참조

각 자식 리포지토리에서 `submodule_updated` 이벤트를 수신하면:

- `master` 브랜치에서 `submodule`을 최신 커밋으로 업데이트
- 정적 분석 및 검증 수행
 - `govulncheck`
 - `go vet`
 - i. 이러한 분석 결과를 github issue tracker에 issue로 등록
- Webhook을 통해 Jenkins 호출
 - 전체 프로젝트 빌드
 - 배포 수행

8.2 API_Server

GitHub Actions 기반 CI 수행

- push 이벤트 발생 시:
 - 부모(메인) 리포지토리에 `submodule_updated` 이벤트 전송
 - 자동화 테스트 수행

8.3 Client_Agent

GitHub Actions 기반 CI 수행

- push 이벤트 발생 시:
 - 부모(메인) 리포지토리에 `submodule_updated` 이벤트 전송
 - 자동화 테스트 수행

8.4 Frontend

GitHub Actions 기반 CI 수행

push 이벤트 발생 시:

- 부모 리포지토리에 `submodule_updated` 이벤트 전송
- 정적 리소스 빌드 수행
 - HTML / CSS / JS 번들링

8.5 Nginx

Frontend 빌드 산출물(html, css, js)을 대상으로:

- Jenkins를 통해 자동 배포 수행
- Nginx에서 정적 파일 서빙

8.6 전체 흐름 요약

자식 리포지토리(API, Agent, Frontend)에 변경 발생 (push)

각 리포지토리가 부모 리포지토리에 `submodule_updated` 이벤트 전송
메인 리포지토리:

- submodule 최신화
- 정적 분석 수행
- Jenkins Webhook 호출

Jenkins:

- 전체 빌드
- 서비스 및 정적 파일 배포

9. SBOM

9.1 기본 식별 정보

- 소프트웨어명 : DIRECT P2P
- 포맷: CycloneDX v1.5 (JSON/XML)
- 작성자/조직 정보 : S-개발자 3조 E돌 I들 (김민수, 배민기, 이수진, 최하경)
- 작성 도구: Manual Compile & syft v1.0.0
- 문서 생성일: 2026년 3월 26일

9.2 대상 소프트웨어 정보

- 버전: v1.0.0
- 공급자: 3조 E돌I들
- 설명: UDP 유폴칭 기반 NAT 통과 및 QUIC 고속 데이터 전송 서비스
- 패키지 타입: Application (Client-Server Architecture)

9.3 구성 요소 목록

이름	버전	패키지 URL	타입	비고
Go Runtime	v1.22.0	pkg:golang/org.golang@1.22.0	Runtime	백엔드 및 에이전트 구동
Gin Framework	v1.9.1	pkg:golang/github.com/gin-gonic/gin	Library	API 서버 프레임워크
libp2p	v0.32.0	pkg:golang/github.com/libp2p/go-libp2p@v0.32.0	Library	출편칭 및 상태 저장 관리
coturn	v4.6.2	pkg:generic/coturn@4.6.2	Application	오픈소스 TURN 서버
React	v18.2.0	pkg:npm/react@18.2.0	Framework	웹 프론트엔드
SQLite	v3.45.1	pkg:generic/sqlite@3.45.1	Data Store	로컬/서버 데이터베이스
Nginx	v1.25.3	pkg:generic/nginx@1.25.3	server	스택 콘텐츠 서버
OpenSSL(TLS)	3.0.13	pkg:generic/openssl@3.0.13	crypto	데이터 암호화 라이브러리
ffmpeg	v6.1.1	pkg:generic/ffmpeg@6.1.1	video transcoding	영상 조작 라이브러리
jenkins	v2.452.1	pkg:generic/jenkins@2.452.1	Application	CI/CD 파이프라인 자동화 및 빌드/배포 오케스트레이션

9.4 의존성 관계

```

DirectP2P v1.0.0
├── Static Server (Nginx)
│   └── Nginx 1.25.3
├── Web Frontend (JavaScript / React)
│   └── React 18.2.0
├── API Server (Go)
│   ├── Go Runtime 1.22.0
│   ├── Gin v1.9.1
│   ├── SQLite 3.45.1
│   └── libp2p v0.32.0
├── Local Agent (Go)
│   ├── Go Runtime 1.22.0
│   ├── libp2p v0.32.0
│   ├── ffmpeg
│   └── OpenSSL 3.0.13
└── TURN Server (coturn)
    └── coturn 4.6.2

```

9.5 라이선스 정보

컴포넌트	라이선스
Gin	MIT
libp2p	MIT
Go-sqlite3	MIT
Nginx	BSD 2 clause
OpenSSL	Apache 2.0
coturn	BSD 3 clause
ffmpeg	LGPL

| 전체 프로젝트 라이선스: MIT

9.6 취약점 정보

- 검증된 라이브러리 사용으로 현재 식별된 취약점은 없으나, 보안 무결성을 유지하기 위해 분기별 의존성 점검을 실시할 예정입니다. 오픈소스 커뮤니티의 보안 패치 노트를 지속적으로 검토하여, 보안 취약점이 보고되는 즉시 상위 버전으로의 마이그레이션 또는 긴급 업데이트를 수행하는 선제적 대응 프로세스를 운영하겠습니다

9.7 보안 및 무결성 정보

- 피어 인증 및 보안 통신: **libp2p**의 보안 전송 계층을 활용하여 피어 간 고유 ID(PeerID) 기반의 인증을 수행하며, **TLS 1.3** 프로토콜을 통해 핸드셰이크 단계부터 모든 통신을 암호화함.
- 전송 암호화: QUIC 프로토콜 자체의 암호화 기능과 더불어, **libp2p** 수준에서 지원하는 **AES-256-GCM** 또는 **ChaCha20-Poly1305** 기반의 스트림 암호화 적용.
- 데이터 무결성: **libp2p**의 멀티스트림(multistream) 환경에서 각 메시지 프레임에 대한 **SHA-256** 체크섬 검증을 수행하여 네트워크 전송 중 데이터 변조를 원천 차단.
- 리플레이 공격 방지: QUIC의 연결 식별자(Connection ID) 및 **libp2p**의 시퀀스 번호 관리를 통해 패킷 재전송 공격(Replay Attack)을 방지함.

9.8 빌드 정보

- **Go**: 1.22.0
- 빌드 방식
 - API Server & Agent → go build
- 빌드 산출물
 - agent.exe / agent.bin
 - server binary
 - frontend SPA

9.9 공급망 정보

- 외부 서비스
 - Google STUN Server
- 배포 구조
 - Nginx → Static 파일 제공
 - API Server → signaling
 - TURN → fallback

9.10 외부 참조

1. 표준 프로토콜 명세 (Standards)

- **RFC 5389**: Session Traversal Utilities for NAT (STUN)
- **RFC 5766**: Traversal Using Relays around NAT (TURN)
- **RFC 9000**: QUIC: A UDP-Based Multiplexed and Secure Transport
- **RFC 5245**: Interactive Connectivity Establishment (ICE)
 - 출판청 절차의 표준입니다.

2. 핵심 라이브러리 및 오픈소스 (Implementations)

- **Gin Framework**: <https://github.com/gin-gonic/gin> (API 서버 프레임워크)
- **libp2p (Go)**: <https://github.com/libp2p/go-libp2p> (P2P 네트워크 스택)
- **quic-go**: <https://github.com/quic-go/quic-go> (libp2p 내부 QUIC 전송 계층)
- **go-sqlite3** : <https://github.com/mattn/go-sqlite3> (임베디드 데이터베이스)
- **coturn**: <https://github.com/coturn/coturn> (TURN 서버 구현체)
- **Nginx** : <https://nginx.org/>
 - **ffmpeg**: <https://www.ffmpeg.org>
 - **jenkins**: <https://www.jenkins.io>

3. 프로젝트 자원 (Project Assets)

- **GitHub Repository**: <https://github.com/E2IDLE/DirectP2P>

9.11 배포 정보

- 본 프로젝트는 GitHub Actions 기반의 CI/CD(지속적 통합 및 배포) 파이프라인을 통해 소프트웨어의 공급망 보안과 배포 무결성을 유지합니다.

1. 빌드 및 아티팩트 생성 (CI: Continuous Integration)

- **컴파일 및 빌드**: GitHub Actions 워크플로우를 통해 Go 1.22 환경에서 각 OS별(Windows, macOS, Linux) 바이너리를 교차 컴파일(Cross-compile)함.
- **프론트엔드 빌드**: Node.js 환경에서 React 소스를 최적화(Minify/Bundle)하여 정적 에셋 생성.
- **무결성 확보**: 빌드 직후 모든 실행 파일에 대해 SHA-256 체크섬(Checksum)을 생성하여 파일 변조 여부를 추적함

2. 자동 배포 프로세스 (CD: Continuous Deployment)

- 중앙 서버 배포:
 - **main** 브랜치 푸시 시 **Docker** 이미지를 빌드하여 컨테이너 레지스트리에 저장.
 - 중앙 관리 서버(**API Server**)에 컨테이너 기반으로 자동 갱신 배포.
- 에이전트 및 웹 서빙:
 - 빌드된 에이전트 설치 파일과 프론트엔드 정적 파일을 **Nginx**가 구동 중인 **Static Server**로 **SCP/SFTP**를 통해 자동 전송.
 - 최신 버전 정보가 담긴 **latest.json** 메타데이터를 갱신하여 에이전트의 자동 업데이트 감지 기능을 지원.

3. 패키지 관리 및 유통

- 백엔드/에이전트: **Go Modules (go.mod)**를 통해 외부 의존성 버전을 고정하여 빌드 재현성(**Reproducibility**) 확보.
- 프론트엔드: **NPM** 패키지 관리자를 통해 라이브러리 의존성 및 보안 취약점(**npm audit**) 관리.
- 배포 위치:
 - 웹 UI: **https://~**

4. 설치 및 권한 관리

- 시스템 등록:
 - **Windows**: 설치 프로그램을 통해 시스템 서비스로 등록.
 - **Linux/macOS**: **systemd** 또는 **launchctl** 스크립트를 배포 패키지에 포함하여 백그라운드 데몬화.
- 보안: 모든 배포 엔드포인트는 **HTTPS(TLS)**를 강제하여 전송 중인 설치 파일의 가로채기 방지함.

9.12 저작권 및 법적 정보

- Copyright © 2026 E돌이들
- License: MIT

10. 비기능 요구사항

10.1 성능

항목	목표 수치	비고
출편칭 성공 시간	5초 이내	Full Cone / Restricted NAT 환경 기준
TURN 폴백 전환 시간	출편칭 실패 후 2초 이내	-
P2P 전송 속도	이론 회선 속도의 80% 이상	LAN 환경 기준
청크 재전송 지연	100ms 이내	NACK 감지 후 재전송 기준
Web API 응답 시간	95% 요청 200ms 이내	-

10.2 보안

항목	요구사항
전송 암호화	종단 간 AES-256-GCM 암호화 (P2P 및 TURN 릴레이 모두 적용)
인증	Token
TURN 자격증명	HMAC-SHA256 기반 시간 제한 임시 자격증명 (유효 1시간)
API 보안	HTTPS 필수, CORS 정책 적용, Rate Limiting
에이전트 통신	로컬 HTTP API는 localhost 바인딩만 허용

10.3 호환성

항목	요구사항
에이전트 OS	Windows 10+, macOS 11+, Ubuntu 20.04+ / Debian 10+
웹 브라우저	Chrome 90+, Firefox 88+, Edge 90+, Safari 14+
NAT 지원	Full Cone, Restricted Cone, Port Restricted Cone NAT (출편칭), Symmetric NAT (TURN 폴백)
프로토콜	STUN (RFC 5389), TURN (RFC 5766), UDP/TCP

11. 오류 코드 정의

오류 코드	영역	설명	처리 방안
E1001	STUN	STUN 서버 응답 없음	재시도 최대 3회 후 오류 보고
E1002	STUN	공인 주소 획득 실패	네트워크 설정 확인 안내
E2001	홀펀칭	홀펀칭 타임아웃 (5초)	TURN 폴백 자동 전환
E2002	홀펀칭	모든 Candidate 연결 실패	TURN 폴백 자동 전환
E3001	TURN	TURN 인증 실패	API Server에서 자격증명 재발급
E3002	TURN	TURN 서버 연결 불가	서버 상태 확인 및 사용자 오류 안내
E4001	전송	체크 전송 타임아웃	재전송 요청(NACK)
E4002	전송	무결성 검증 실패	전체 재전송 요청
E4003	전송	디스크 용량 부족	수신측 저장 공간 확보 안내
E5001	API	인증 토큰 만료	토큰 갱신 후 재요청
E5002	API	세션 없음	세션 재생성 안내
E6001	에이전트	에이전트 미실행	에이전트 설치/실행 안내 팝업