공인전자문서중계자 간 중계전송 표준연계 기술규격

v1.0

2026년 6월

KISA 한국인터넷진흥원
KOREA INTERNET & SECURITY AGENCY

목 차

<그림 차례>

1. 개 요

1.1. 목적

"공인전자문서중계자 간 중계전송 표준연계 기술규격"(이하 '본 규격')은 공인전자주소 등록자(이하 '이용자')들이 서로 다른 공인전자문서중계자(이하 '중계자') 서비스를 이용하는 경우에도 전자문서를 송·수신함에 있어 제약이 없도록 중계시스템들 간에 상호 전자문서를 중계하려는 목적으로 작성되었다.

본 규격은 전자문서의 송신자와 수신자가 서로 다른 중계자에 등록되어 있는 경우에도 발송요청을 받은 송신자의 중계시스템이 수신자의 중계시스템으로 전자문서를 중계하고, 중계시스템들 간 상호 수신 및 열람에 대한 처리 상태를 공유하도록 지원한다.

이를 통해 이용자가 공인전자문서중계자의 서비스를 선택함에 있어서 선택권을 넓히고 궁극적으로 전자문서의 유통을 활성화하는 것을 목적으로 한다.

1.2. 적용대상 및 범위

본 규격은 중계자들을 주 대상으로 하며, 중계자들을 관리·감독하는 전담기관도 그 대상으로 한다.

본 규격은 다음과 같은 기능 범위를 대상으로 한다.

- 중계자가 전담기관이 운용하는 시스템에 중계자 정보를 등록하는 기능
- 송신과 수신을 선택한 중계자가 서로 다를 경우, 이용자와는 관계없이 중계자들 간에 전자문서를 연계하도록 하는 기능
- 송신과 수신을 선택한 중계자가 서로 다를 경우, 이용자가 전자문서를 열람할 수 있도록 하는 기능

2. 용어 정리

본 기술규격에서 사용하는 용어의 정의는 다음과 같다.

1. “연계”란 2개 이상의 프로그램이나 모듈을 상호 간 작용하도록 하는 것을 말한다.

2. “전자문서 유통허브시스템(이하 ‘유통허브시스템’)”이란 「전자문서 및 전자거래 기본법」(이하 ‘전자문서법’) 제22조에서 규정하고 있는 전담기관이 공인전자문서중계자 서비스 인프라를 위해 구축 및 운영하고 있는 시스템을 말한다.

3. “전담기관”이란 전자문서법 제22조에서 규정하고 있는 기관으로 공인전자문서중계자 제도를 관리하는 기관을 말한다.

4. “발송기관”이란 고지나 안내 등을 위해 전자고지서 등 전자문서를 발송하는 기관을 말한다.

5. “중계자 목록(Whitelist)”이란 전자문서법에 따라 인증받은 중계자들을 인식할 수 있는 정보들의 집합을 말한다.

6. “수신 동의”란 이용자가 특정 발송기관이 발송하는 전자문서를 이용약관 동의 등을 거쳐 수신받겠다는 전자적 의사표시를 말한다.

7. “송신중계자”란 전자문서의 송신자와 수신자가 각기 다른 중계자에 등록되어 있는 경우, 송신자가 등록되어 전자문서를 송신하는 역할을 하는 중계자를 말한다.

8. “수신중계자”란 전자문서의 송신자와 수신자가 각기 다른 중계자에 등록되어 있는 경우, 수신자가 등록되어 전자문서를 수신하는 역할을 하는 중계자를 말한다.

9. “인증토큰”이란 사용자가 자신의 신분을 증명한 뒤 서비스 제공자로부터 발급받은 토큰으로서 허가된 사용자임을 증명하기 위해 사용된다.

10. “열람인증토큰”이란 URL방식 전자문서 열람일 경우에 수신자가 수신중계자로부터 발급받는 인증토큰을 말하며 발송기관에 저장된 전자문서의 열람에 사용된다.

11. “중계자 정보”란 중계자 명칭, 중계자 인증 일시, 연락처 정보, 서비스 정보 등 이용자가 중계자를 파악할 수 있도록 하는 정보를 말한다.

12. “클라이언트 아이디”란 유통허브시스템에서 중계자에게 발급한 임의의 식별 값으로서 중계자

플랫폼 시스템의 인증을 위해 사용된다.

13. “클라이언트 비밀번호”란 유통허브시스템에서 클라이언트 아이디 발급 시 함께 발급하는 값으로서 중계자 플랫폼 인증을 위한 역할을 담당하는 임의의 문자열이다.

14. “유통증명서”란 공인전자주소를 통하여 전자문서가 송신 또는 수신되거나 열람된 경우 전자문서법 제18조의5 제1항 각 호의 사항이 포함된 정보를 담은 증명서를 말한다.

3. 중계자 간 연계 방안

3.1. 개요

송·수신자가 서로 다른 중계자에 가입한 경우, 송·수신자 사이에 전자문서를 유통하기 위해 중계자 간 연계를 위한 기본정보를 공유하는 사전준비 절차와 중계자 간 전자문서를 유통하는 절차가 필요하다.

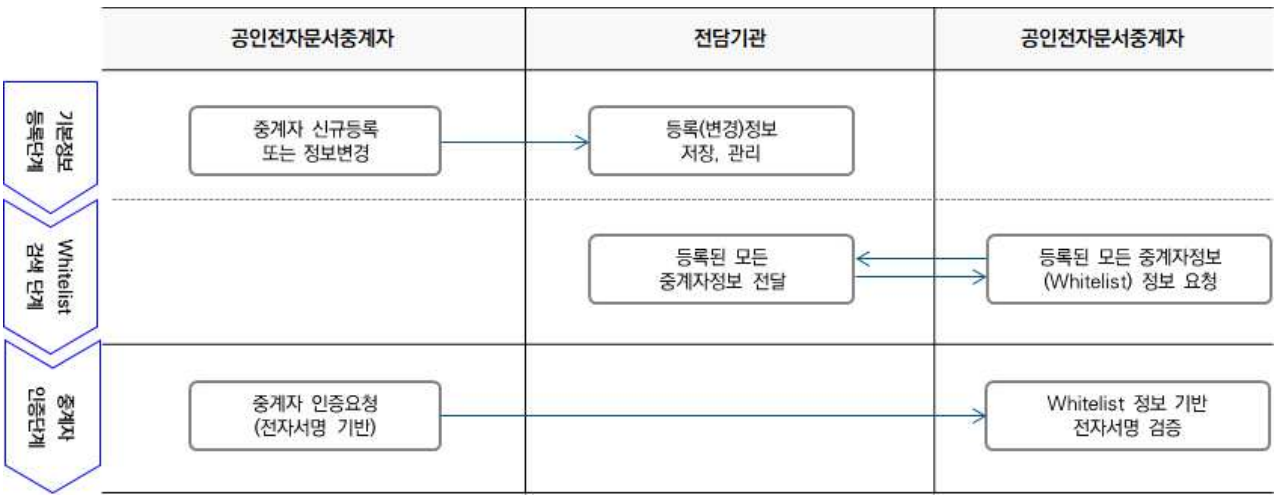
먼저 서로 다른 중계자 간 전자문서를 유통하기 위해서는 상대방 중계자가 현재 중계자로 인증받은 유효한 사업자인지에 대한 확인이 필요하며, 유효한 사업자인 경우 중계자 간 연계를 위한 기본정보(네트워크 주소, 공개키 등)를 서로 공유하여야 한다.

이를 위해 중계자는 전담기관에서 관리하는 유통허브시스템을 통해 주기적으로 공인전자문서 중계자에 대한 기본정보를 획득하는 "1) 중계자 정보 등록(변경) 및 획득 업무"를 진행하여야 한다.

1) 중계자 정보 등록(변경) 및 획득 업무

- 중계자가 전담기관에게 요청하는 시점에 공인전자문서중계자로서 인증이 유효한 중계자 목록(Whitelist)을 요청하고 이를 획득하는 업무임

- 이 업무는 중계자가 신규 인증되거나 기존 중계자의 정보가 변경(인증상태의 변경 여부, 연계를 위한 네트워크 주소, 중계자의 전자서명 공개키 등)된 경우 각 중계자는 신규 또는 변경된 정보를 전담기관의 유통허브시스템에 등록하는 단계, 중계자가 유통허브시스템에 등록된 중계자의 목록(Whitelist)을 검색 요청하는 단계, 상호 연계대상인 중계자를 인증하기 위한 중계자 인증 단계로 구성됨



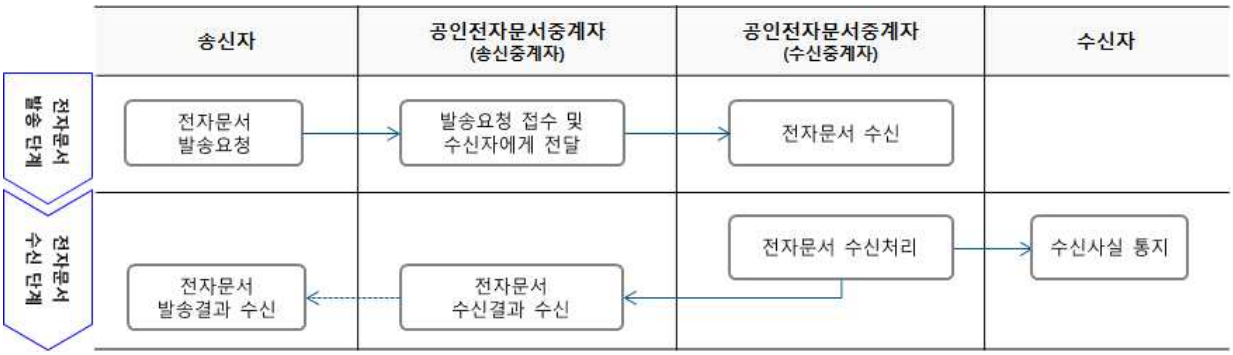
[그림 1] 중계자 정보 등록(변경) 또는 획득 절차

※ 중계자 간 상호 연계를 위해서는 반드시 연계 전에 위 업무를 통해 획득한 인증된 중계자 목록(Whitelist)를 기반으로 중계자 인증단계를 거쳐야 한다.

이러한 준비 과정을 통해 사전 준비가 완료되면 이제 중계자 간 전자문서 중계전송을 위한 전자문서 송·수신 단계가 진행된다. 송신자가 송신중계자를 통해 전자문서를 발송하고 수신자가 수신중계자를 통해 전자문서를 수신하는 이 과정이 "2) 전자문서 발송 및 수신 업무"이다.

2) 전자문서 발송 및 수신 업무

- 송신자가 송신요청을 한 중계자와 수신자가 수신하고자 하는 중계자가 서로 다른 중계자인 경우에 전자문서를 송·수신하는 업무
- 이 업무는 송신자가 계약(또는 '가입')한 송신중계자에게 전자문서 발송을 요청하면 송신중계자는 전자문서 수신 대상자가 수신하고자 하는 수신중계자에게 전자문서를 발송하는 단계와 수신중계자가 전달받은 전자문서를 수신자에게 정상 전달 후 그 결과(수신확인)를 송신중계자에게 전달하는 단계로 구성됨



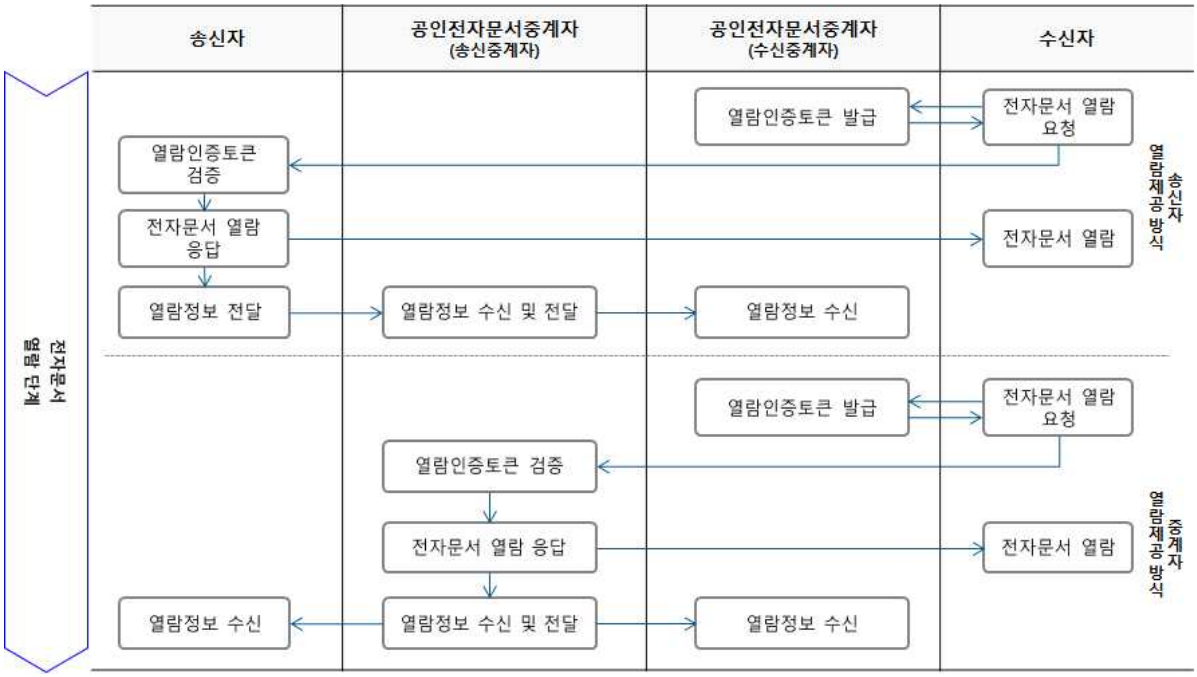
[그림 2] 전자문서 발송 및 수신 절차

수신자가 전자문서를 수신한 이후에는 전자문서를 열람하는 단계와 그 결과(열람확인)를 송신 중계자에게 전달하는 "3) 전자문서 열람 업무"가 진행된다.

3) 전자문서 열람 업무

- 수신자가 수신한 전자문서를 열람하는 업무
- 열람유형은 발송기관에서 전자문서 원문에 대한 URL과 무결성 정보만을 발송하는 'Ⓐ 송신자 열람제공 방식'과 직접 전자문서 원문을 첨부해서 전달하는 'Ⓑ 중계자 열람제공 방식'의 2개 유형으로 구분

※ 첨부방식의 열람의 경우 전자문서에 대한 열람서비스를 제공하는 것은 송신중계자가 담당하는 것을 기본으로 함



[그림 3] 전자문서 열람 절차

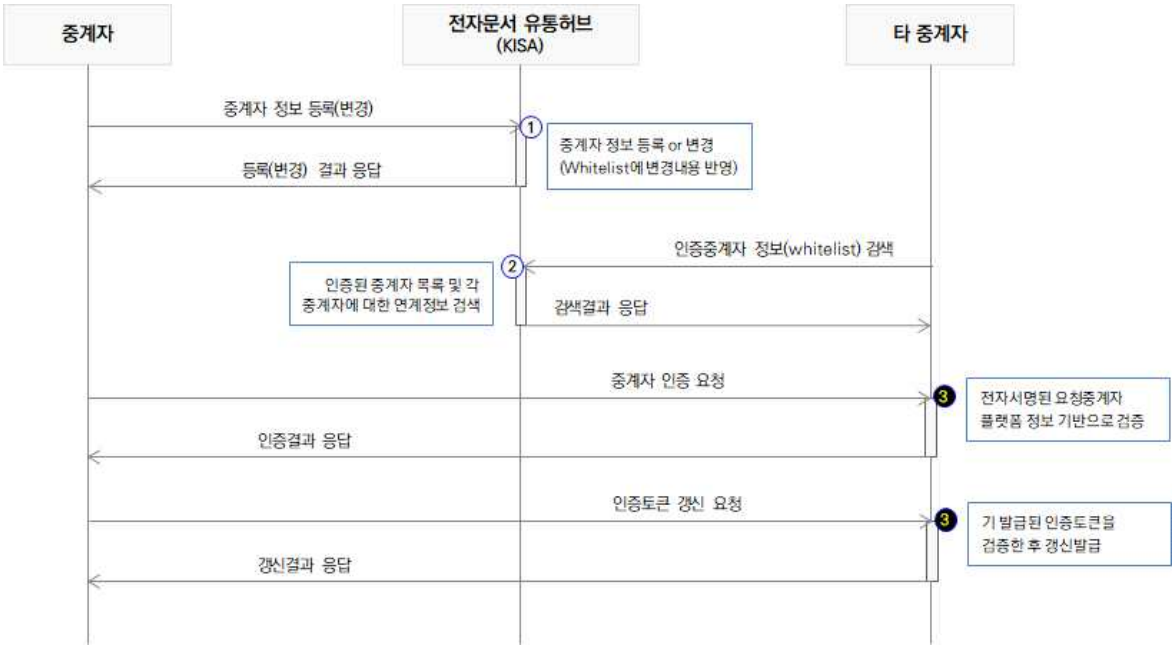
3.2. 업무별 상세 연계 절차

송·수신 중계자 간 전자문서 유통의 각 업무('중계자 정보 등록(변경) 및 획득 업무', '전자문서 발송 및 수신 업무', '전자문서 열람 업무')별 상세 절차는 아래와 같다.

각 업무를 수행하기 위한 처리 절차는 각 업무의 처리 흐름도에 표기된 번호 순으로 진행된다. 각 절차 중 송·수신 중계자가 동일하지 않아 표준화된 인터페이스가 필수인 절차는 "❶, ❷ ..." 와 같이 표기하였고 표준화된 인터페이스가 아닌 절차는 "①, ② ..."로 구분하여 표기하였으며, 본 규약에서는 표준화가 필수인 인터페이스에 대해서만 정의한다.

3.2.1. 중계자 정보 등록(변경) 및 획득 업무

□ 처리절차에 따른 흐름도



[그림 4] 중계자 정보 등록 및 중계자 간 인증 절차

□ 흐름에 따른 처리 상세내역

1) 중계전송에 참여하는 모든 (송·수신)중계자는 전담기관으로부터 클라이언트 아이디와 클라이언트 비밀번호를 부여받아 유통허브시스템에 접근할 수 있는 권한을 부여받은 후에, 반드시 자신의 플랫폼에 대한 기본정보(네트워크 주소, 공개키 등)를 유통허브시스템에 등록('❶ 중계자정보 등록(변경)')한다.

2) 또한 중계자는 중계서비스 운영 중 기본정보(네트워크 주소, 공개키 등)가 변경된 경우에도 변경된 정보를 전담기관의 유통허브시스템에 등록('❶ 중계자정보 등록(변경)')하여야 한다.

※ 전담기관에서는 중계자 정보의 등록 및 변경은 온/오프라인으로 모두 제공하며, 오프라인으로 변경정보를 제공할 때는 반드시 네트워크 주소나 공개키 등의 변경 전에 변경예정 일시와 함께 전담기관에 해당 내용을 보고하여야 함

3) 중계자 시스템 간 연계를 위하여 모든 중계자는 주기적으로 전담기관의 유통허브시스템에 등록된 타 중계자의 기본정보(네트워크 주소, 공개키 등)를 검색('❷ 중계자목록(whitelist) 검색')한 후, 획득한 타 중계자 정보(네트워크 주소, 공개키 등)를 관리하여야 한다.

4) 중계자가 상대방 중계자에 연계하기 위해서는 반드시 중계자 인증요청을 한 후 인증토큰을 발급('❸ 중계자 인증토큰 발급 및 갱신')받아야 한다. 이를 위한 세부 절차는 다음과 같다.
4-1) 연계를 요청하는 중계자는 자신의 플랫폼 정보를 개인키로 전자서명한 후 상대방 중계자

에게 전달하여야 한다.

4-2) 연계요청을 수신한 상대방 중계자는 관리하고 있던 whitelist 정보에서 요청 중계자의 공개키를 획득하고 이를 기반으로 요청 중계자가 유효한지를 검증하여야 한다.

※ *중계자 검증 과정에서 연계요청을 수신한 중계자가 요청 중계자에 대한 검증이 실패하는 경우에는 유통허브시스템에 접속하여 요청 중계자에 대한 최신 정보를 다시 검색한 후, 이를 기반으로 재검증을 수행(중계자가 유통허브로부터 주기적으로 whitelist를 획득함에 따라 타 중계자에 대한 최신정보가 반영되지 못한 경우를 고려한 절차로 이를 구현할 것을 권고함)*

4-3) 연계요청을 수신한 중계자는 검증결과가 유효하면 인증토큰을 발급한 후, 이를 요청 중계자에게 응답메시지로 전달한다.

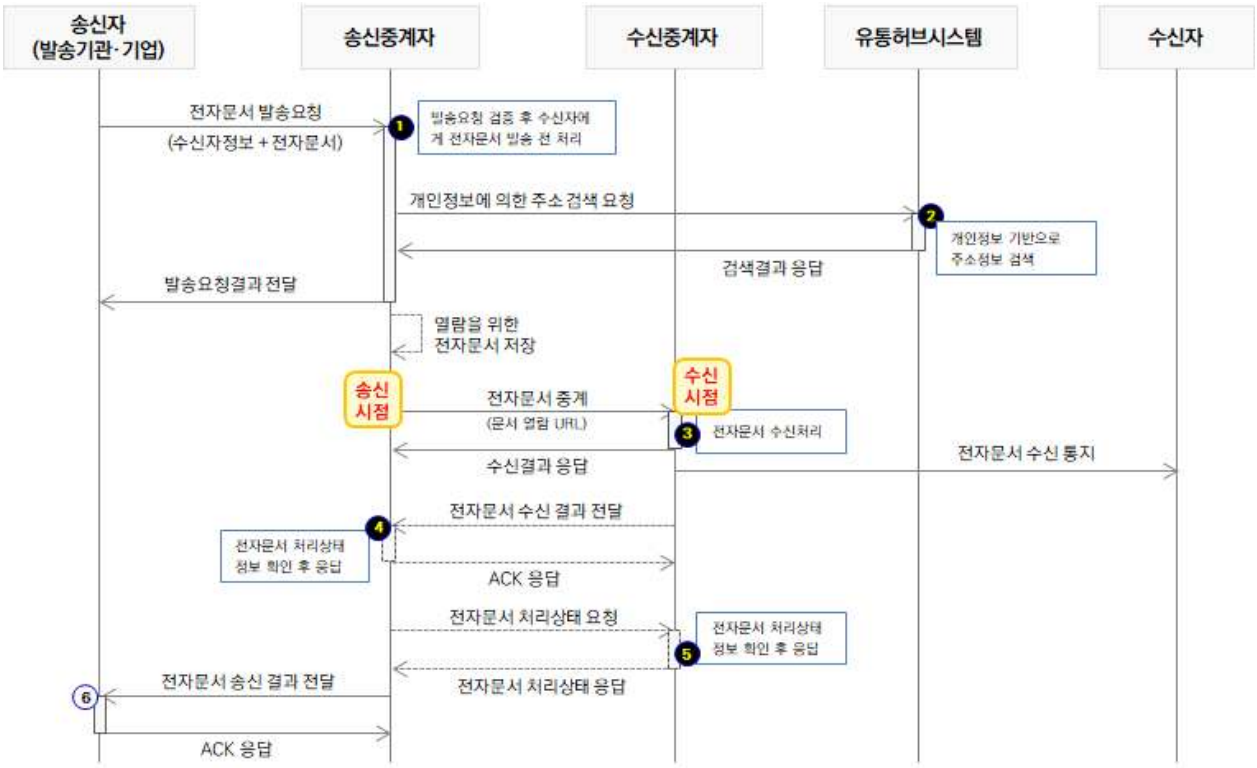
5) 중계자가 상대방 중계자와 계속 연계하고자 하는 경우, 상대방 중계자로부터 발급받은 인증토큰이 유효한 상태에서 인증토큰의 유효기간을 연장하기 위해 인증토큰 갱신을 요청('❸ 중계자 인증토큰 발급 및 갱신')할 수 있다.

□ 처리 절차별 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
❶ 중계자정보 등록(변경)	▪ 중계자는 유통허브시스템에 자신의 플랫폼 기본정보(네트워크 주소, 공개키)를 등록하고, 이후 변경 등 발생 시 유통허브시스템 내 정보를 갱신	공인전자문서 중계자	유통허브 시스템	"전자문서유통허브_표준연계 기술규격_v1.2" 참조
❷ 중계자목록 검색(Whitelist)	▪ 중계자는 타 중계자와의 시스템 연계를 위해 전담기관의 유통허브시스템에 접근하여 타 중계자의 목록 및 공개키 정보를 획득	공인전자문서 중계자	유통허브 시스템	"전자문서유통허브_표준연계 기술규격_v1.2" 참조
❸ 중계자 인증토큰 발급 및 갱신	▪ 연계 요청 중계자가 상대방 중계자와의 연계를 위해 반드시 사전에 수행해야 하는 API ▪ 요청 중계자는 자신의 플랫폼 정보를 개인키로 전자서명한 후 연계 대상 중계자에게 전송하여 인증토큰 발급을 요청하거나, 기 발급받은 인증토큰의 유효기간을 연장하기 위해 인증토큰의 갱신을 요청	연계요청 중계자	연계대상 중계자	발급받은 인증토큰은 중계자 간 API 호출 시, 메시지 헤더에 인자로 추가하여 전달

3.2.2. 전자문서 발송 업무

□ 처리 절차에 따른 흐름도



[그림 5] 전자문서 발송 처리 흐름도

□ 흐름에 따른 상세처리 내역

1) 송신자는 전자문서 수신자의 수신 플랫폼(플랫폼 아이디)과 식별자(공인전자주소) 정보를 포함하여 송신중계자에게 전자문서 발송을 요청('❶ 전자문서 발송 요청')한다.

※ *송신자가 송신중계자에게 전자문서 발송요청 시, 전자문서 직접 전달 또는 전자문서 열람 URL과 무결성 정보(ex: 전자문서 해시) 전달 방식이 모두 가능하며, 이 전달 방식에 따라 전자문서 열람단계에서 열람절차가 달라짐*

2) 송신자가 송신중계자에게 수신자의 수신플랫폼과 공인전자주소를 직접 전달한 경우가 아니면, 송신중계자는 유통허브시스템에 접속하여 수신자 정보를 기반으로 수신플랫폼과 공인전자주소를 검색 요청('❷ 공인전자주소 검색')한다.

3) 송신자가 전자문서를 직접 첨부하여 발송요청한 경우 송신중계자는 전자문서의 열람을 위해 전자문서를 자체적으로 저장한 후, 전자문서 열람URL을 생성한다.

※ *송신중계자가 직접 수신중계자에게 전자문서를 첨부하여 전달하는 경우에는 송신자, 송신중계자, 수신중계자 간 개인정보 위수탁 협약 등의 추가 검토가 반드시 필요함*

4) 수신자의 수신 플랫폼이 송신중계자의 플랫폼과 동일한 경우에는 송신중계자 내부 절차에 따라 수신 및 열람을 처리하고, 송신중계자가 아닌 타 중계 플랫폼인 경우에는 수신중계자에게 전자문서를 중계('❸ 전자문서 중계')한다.

5) 전자문서를 수신한 수신중계자는 처리 효율성 및 부하를 고려하여 "5-1) 비동기식 처리방안" 또는 "5-2) 동기식 처리방안" 절차에 따라 선택적으로 처리한다.

5-1) 비동기식 처리방안: 수신중계자는 대량으로 전송되는 메시지 처리를 위해 요청메시지에 대한 기본적 검증(*) 후, 수신접수 처리를 완료하고, 송신중계자는 전송한 전자문서에 대해 처리상태를 "송신 완료"로 관리함.

5-2) 동기식 처리방안: 수신중계자는 수신한 요청메시지에 대한 기본적인 검증(*)과 요청 메시지 내 전자문서 전송정보가 유효(**)한지 검증한 후, 유효한 전자문서는 수신처리를, 유효하지 않은 전자문서에 대해서는 수신 실패처리를 하고 결과를 응답메시지로 전달함. 이때 송신중계자는 수신한 결과 값에 따라 "수신 완료" 또는 "송신 실패"로 관리함

** 송신 메시지가 API 규격에 적합한지, 인증토큰은 유효한지 검증함*

*** 전자문서의 전송정보: 전자문서번호, 수신자 주소, URL 또는 첨부파일의 적합성 등*

※ *송신중계자는 전자문서 중계를 완료한 후, 처리상태가 "송신 완료" 또는 "수신 완료" 상태인 경우, 수신중계자에게 전송한 시각을 송신시각으로 설정하여 유통허브시스템에 "전자문서 유통정보 등록" API를 이용하여 유통정보를 보고할 수 있음(선택)*

※ *수신중계자는 송신중계자로부터 받은 전자문서를 정상적으로 수신자의 수신함에 전달하여 "수신 완료"상태가 되면, 전자문서를 송신중계자로부터 받은 시각을 수신시각, 송신중계자가 중계요청 메시지에 보낸 송신시각을 송신시각으로 설정하여 유통허브시스템에 "전자문서 유통정보 등록" API를 이용하여 유통정보를 보고하여야 함(필수)*

※ *유통허브시스템은 수신중계자가 보낸 정보를 기준으로 송신시각과 수신시각을 산정하는 것을 기본으로 함. 다만 수신중계자 시스템의 문제로 인해 유통정보가 유통허브시스템에 보고되지 않은 경우에도 송신중계자가 송신사실을 보호받기 위해 송신정보를 유통허브시스템에 보고할 수 있도록 함*

6) 수신중계자는 수신자에게 전자문서 수신사실을 통지한다.

7) 수신중계자가 전자문서를 수신한 후에 송신중계자에게 수신결과에 대한 응답을 정상적으로 전달하지 못하였거나 "5-1) 비동기식 처리방안"으로 접수만 한 경우, 수신중계자는 수신처리가 완료된 후 송신중계자에게 접수문서의 수신 결과에 대한 정보를 전달('❹ 전자문서 수신결과 전달)하여야 한다.

8) 만약 송신중계자가 수신중계자로부터 중계한 전자문서의 수신결과 정보를 일정 시간 내에 받지 못한 경우 발송한 전자문서에 대한 처리 상태를 수신중계자에게 요청('❺ 전자문서 처리상태 요청)하여 발송된 전자문서의 상태정보를 파악할 수 있다.

9) 송신중계자는 전자문서 송신결과를 송신자에게 전달('⑥ 전자문서 송신결과 전달')한다.

※ *이는 선택적 절차로 송신자와 송신중계자 간 협의에 의해 결정한다.*

□ 오류 발생에 따른 처리 방안

송신중계자와 수신중계자간 전자문서를 중계하는 과정에서 시스템이나 네트워크 상 다양한 문제에 의해 오류가 발생될 수 있습니다.

각 단계별 발생 가능한 오류유형과 그에 따른 대처방안은 다음과 같습니다.

[그림 6] 전자문서 발송 단계별 오류발생 유형

① 송신중계자가 전자문서 발송메시지에 대해서 HTTP 오류 메시지를 받은 경우

○ 오류 발생 상황

- 요청 메시지가 HTTP 문법에 맞지 않거나 클라이언트 인증 실패와 같은 HTTP 헤더정보 오류 또는 서버 내부의 오류발생 등의 문제로 수신중계자가 송신중계자에게 HTTP 오류코드(HTTP 4xx, 5xx)를 응답한 경우로 기본적인 전송이 실패하였음을 의미함

○ 오류 발생에 따른 대응방안

수신중계자 오류발생 유형	수신중계자 처리 내용	송신중계자 대응 방안
▪ HTTP서버가 수신한 요청 메시지가 HTTP 프로토콜에 적합하지 않은 경우 ▪ HTTP 헤더의 인증토큰이 유효하지 않거나 접근권한이 없는 경우 ▪ 요청 리소스가 서버에 없는 경우	▪ HTTP프로토콜에서 정의한 오류코드에 따라 HTTP 4xx 오류 응답	▪ 클라이언트의 오류로 요청메시지 수정(시스템 보완) 후 재처리하여야 함 ▪ “송신 중” 상태의 전자문서를 “송신실패”로 처리
▪ 서버 과부하 또는 내부 문제로 요청에 대한 정상적 처리가 불가한 경우	▪ HTTP프로토콜에서 정의한 오류코드에 따라 HTTP 5xx 오류 응답	▪ 수신중계자 서버 오류이므로 재전송 대상임 ▪ “송신 중” 상태의 전자문서를 “송신실패”로 인지하고 “재전송 대상”으로 분류하여 동일한 요청 메시지로 재전송 처리

② 송신중계자가 전자문서 발송 후 응답메시지를 받지 못한 경우

○ 오류 발생 상황

- 송신중계자가 네트워크나 수신중계자 시스템 오류로 인해 '전자문서 전송 API' 호출에 대한 응답메시지를 받지 못했거나, 송신중계자 자체 시스템 오류로 인해 응답메시지 수신에 실패한 경우로 전자문서 전송에 대한 성공, 실패 여부를 알지 못하는 것을 의미함

○ 오류 발생에 따른 대응방안

수신중계자 오류 발생 유형	수신중계자 처리 내용	송신중계자 대응 방안
▪ 수신중계자가 요청메시지를 정상적으로 처리한 후, 응답메시지 전송 중 오류로 응답이 전송되지 않은 경우 ▪ 수신중계자가 요청메시지를 받지 못하였거나, 처리 과정에서의 오류, 응답메시지 전송과정에서의 오류로 인해 응답이 전송되지 않은 경우 ▪ 송신중계자가 응답메시지 수신처리 과정에서 오류가 발생하여 응답메시지 수신에 실패한 경우	-	전송에 대한 응답을 받지 못한 전자문서에 대해 수신중계자에게 수신 처리상태정보를 요청한 후 전달받은 응답 상태에 따라 아래와 같이 전자문서 처리상태 반영 ▪ "수신" 응답: 수신이 완료된 상태로 "수신"상태 및 수신일시 업데이트 ▪ "열람" 응답: 수신 후 열람까지 완료된 상태로 "열람"상태 및 수신일시, 열람일시를 모두 반영 ▪ "수신실패" 응답: 수신중계자가 전자문서 수신처리 과정에서 전자문서 전송정보가 유효하지 않아서 수신에 실패한 문서로서 송신중계자는 해당 전자문서를 "송신실패"로 인지함 ▪ "미수신" 응답: 수신중계자가 전자문서 전송요청 메시지를 수신하지 못한 경우로 송신중계자는 해당 전자문서를 "송신실패"로 인지하고 재전송 대상으로 분류하여 재전송 처리

※ *수신중계자가 전송요청에 대해 비동기식 처리를 함으로써 송신중계자가 "송신완료"로 기록한 전자문서에 대해서는 수신중계자의 처리시간에 따라 송신중계자가 수신상태를 요청하는 시점에 "미수신" 상태로 응답을 받을 수 있음. 이때 자동 재전송을 하게 되면 수신중계자가 전자문서를 중복 수신하는 상황이 빈번하게 발생될 수 있으므로 관리자 확인 후 처리 필요*

③ 송신중계자가 '처리 오류'라는 응답메시지를 받은 경우

- 오류 발생상황에 대한 설명: 송신중계자가 '전자문서 전송 API' 호출에 대해 비즈니스 응답이 포함된 응답메시지를 받았으나, 응답메시지에 처리결과가 '실패'로 기록된 경우로서, 송신중계자가 전자문서 전송에 실패하였음을 의미함

※ *HTTP 응답코드는 '200'으로 전달함으로써 HTTP 통신은 성공으로 처리*

○ 오류 유형별 대처 방안

수신중계자 오류 발생 유형		수신중계자 처리 내용	송신중계자 대응 방안
요청자가 보낸 요청메시지(JSON)가 규격에 맞지 않은 경우		▪ 응답메시지(JSON)내 처리결과를 실패로 기록하여 전송	▪ 해당 전자문서(edocNum 기준) 상태: “송신 실패”로 처리 ▪ 시스템 오류 보완 후 재전송 처리 대상이 됨
수신자 주소가 수신중계자의 회원이 아닌 경우		▪ 응답메시지(JSON)내 처리결과를 실패로 기록하여 전송	
전자문서 중복수신 오류	동일한 edocNum으로 기 수신된 동일한 전자문서가 있는 경우 (동일한 전자문서란 제목, 송수신자 공인전자주소, 전자문서 열람 URL 및 첨부파일 해쉬값이 모두 동일한 경우임)	▪ 기존 정상 수신한 전자문서의 상태값은 변경하지 않고 전자문서 중복수신 오류로 전송 ▪ 기존 전자문서 상태정보(수신, 열람 등에 대한)를 응답 메시지에 같이 전송	▪ 응답메시지로 수신 받은 해당 전자문서(edocNum 기준) 상태정보를 내부 시스템에 반영 ▪ 송신자에게도 상태정보를 전달함
	동일한 edocNum으로 기 수신된 다른 전자문서가 있는 경우 (다른 전자문서란 제목, 송수신자 공인전자주소, 전자문서 열람 URL 및 첨부파일 해쉬값 중 하나라도 다른 경우임)	기존 정상 수신한 전자문서의 상태값은 변경하지 않고 edocNum 중복 오류로 전송	▪ 해당 전자문서(edocNum 기준) 상태: “송신 실패”로 처리 ▪ 송신 중계자가 기 발송한 전자문서 중 동일한 edocNum이 있거나 타 중계자와 edocNum이 중복된다는 의미로 시스템의 중대 결함(오류 보완 후 처리하여야 함)

□ 처리 절차별 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
❶ 전자문서 발송요청	▪ 송신자의 발송요청시스템에서 전자문서를 수신자에게 발송해 줄 것을 중계자에게 요청하는 인터페이스	송신자	송신 중계자	“공인전자문서중계자와_송신시스템_간_표준연계_기술규격” 참조
❷ 공인전자주소 검색	▪ 유통허브시스템이 수신자의 (개인)정보를 기반으로 수신 플랫폼정보 및 공인전자주소를 검색하여 제공하는 인터페이스	송신 중계자	유통허브 시스템	“전자문서유통허브_표준연계 기술규격_v1.2” 참조
❸ 전자문서 중계	▪ 송신 중계자가 타 중계 플랫폼을 통해 수신동의를 한 수신자에게 전자문서를 발송하는 API	송신 중계자	수신 중계자	
❹ 전자문서 수신결과 전달	▪ 송신중계자가 전달한 전자문서를 수신중계자가 접수처리만 하여 응답한 후, 비동기식으로 수신처리를 한 후에 전자문서의 수신결과를 송신중계자에게 전달하는 API	수신 중계자	송신 중계자	
❺ 전자문서 처리상태 조회	▪ 송신중계자가 발송한 전자문서의 상태정보를 응답받지 못한 경우, 전자문서의 현재 처리 상태를 수신중계자에게 확인하는 API	송신 중계자	수신 중계자	

3.2.3. 전자문서 열람단계

3.2.3.1 송신자 열람제공 방식

□ 처리 절차에 따른 흐름도

[그림 7] 전자문서 열람 처리 흐름도(송신자 열람제공 방식)

□ 처리 흐름에 따른 상세

1) 수신자는 수신중계자의 전자문서함에 수신된 전자문서에 대한 열람을 요청한다.('① 열람인증토큰 요청')

※ 수신자가 송신자(발송기관)에게 원문열람 요청 시 수신자가 정당한 이용자임을 증명할 수 있도록 수신중계자는 수신자에게 열람인증토큰을 발급하여야 하며, 송신자가 검증 가능하도록 표준화된 구조("A - 부록. 열람인증토큰" 참조)로 발급하여야 한다.

2) 열람인증토큰을 발급받은 수신자는 해당 토큰을 이용하여 송신자가 제공하는 전자문서 원문에 대한 열람 URL을 호출('② 전자문서 열람요청')하여야 한다.

3) 수신자로부터 열람요청을 받은 송신자는 요청과 함께 전달받은 열람인증토큰의 유효성 검증을 송신중계자에게 요청('❸ 열람인증토큰 검증 API')하여야 한다.

4) 송신중계자는 열람인증토큰을 검증하기 위해 먼저 송신중계자에서 자체 관리하고 있는

whitelist 정보를 기반으로 수신중계자(열람인증토큰 발급자)의 전자서명을 검증하고 인증토큰의 유효기간 등을 추가 검증하여 해당 인증토큰 유효성을 확인하여야 한다.

6) 송신중계자가 whitelist에 있는 수신중계자(열람인증토큰 발급자)의 공개키로 토큰 내 전자서명값 검증에 실패하면 수신중계자에게 열람인증토큰 발급에 사용한 유효한 공개키를 요청하여 획득('❹ 공개키정보 요청')한 후, 이를 기반으로 열람인증토큰의 유효성을 검증하고 검증결과를 송신자에게 전달한다.

※ 이 절차는 열람인증토큰의 전자서명에 사용된 수신중계자의 인증서가 변경되었으나 수신중계자가 전담기관의 유통허브시스템에 이를 통지하고 갱신하지 않은 상황에 대처하기 위함

7) 송신자는 송신중계자로부터 받은 검증결과(성공 또는 실패)에 따라 열람을 요청한 수신자에게 전자문서 원문을 제공하거나 열람불가 메시지를 전달하여야 한다.

8) 송신자(전자문서 열람서비스 제공자)는 전자문서열람에 성공적으로 응답하면 열람결과와 열람시각을 송신중계자에게 전달('❺ 전자문서 열람결과 전달')하여야 한다.

9) 송신중계자는 송신자로부터 열람결과와 열람시각을 받으면 수신중계자에게 이를 전달('❻ 전자문서 열람결과 전달')하여야 한다.

※ *송신중계자는 송신자로부터 전자문서 열람시각 정보를 받으면 이를 열람시각으로 설정하여 유통허브시스템에 "전자문서 유통정보 등록" API를 이용하여 유통정보를 보고하여야 하며, 열람이 2번 이상 발생하여도 열람시각은 최초 열람이 성공한 시각으로 설정하여야 함(필수)*

□ 처리 절차별 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
❷ 전자문서 열람요청	▪ 수신자가 수신중계자로부터 본인확인 후 발급받은 열람인증토큰을 기반으로 송신자가 전달한 전자문서 열람 URL을 호출	수신자	송신자	
❸ 열람인증토큰 유효성 검증	▪ 송신자가 수신자로부터 받은 열람인증토큰의 유효성을 검증하기 위해 송신중계자에게 검증요청하는 인터페이스	송신자	송신 중계자	“공인전자문서중계자와_송신시스템_간_표준연계_기술규격” 참조
❹ 공캐키 정보 요청	▪ 송신중계자가 whitelist에 있는 열람인증토큰 발급자(수신중계자)의 공개키로 토큰 내 전자서명 값 검증에 실패하면 수신중계자에게 직접 공개키를 요청(획득)하는 인터페이스	송신 중계자	수신 중계자	
❺ 전자문서 열람결과 전달	▪ 송신자가 송신중계자에게 전자문서 열람시각정보를 포함한 열람결과를 전달하는 인터페이스	송신자	송신 중계자	“공인전자문서중계자와_송신시스템_간_표준연계_기술규격” 참조
❻ 전자문서 열람결과 전달	▪ 송신중계자가 수신중계자에게 전자문서 열람시각정보를 포함한 열람결과를 전달하는 인터페이스	송신 중계자	수신 중계자	

3.2.3.1 중계자 열람제공 방식

□ 처리 절차에 따른 흐름도

[그림 8] 전자문서 열람 처리 흐름도(중계자 열람제공 방식)

□ 처리 흐름에 따른 상세

1) 수신자는 수신중계자의 전자문서함에 수신된 전자문서에 대한 열람을 요청한다.('① 열람인증토큰 요청')

※ 수신자가 송신자(발송기관)에게 원문열람 요청 시 수신자가 정당한 이용자임을 증명할 수 있도록 수신중계자는 수신자에게 열람인증토큰을 발급하여야 하며, 송신자가 검증 가능하도록 표준화된 구조("A - 부록. 열람인증토큰" 참조)로 발급하여야 한다.

2) 열람인증토큰을 발급받은 수신자는 해당 토큰을 이용하여 송신중계자가 제공하는 전자문서 원문에 대한 열람 URL을 호출('❷ 전자문서 열람요청')하여야 한다.

3) 송신중계자는 수신자로부터 받은 열람인증토큰을 검증하기 위해 먼저 송신중계자에서 자체 관리하고 있는 whitelist 정보를 기반으로 수신중계자(열람인증토큰 발급자)의 전자서명을 검증하고 인증토큰의 유효기간, 문서 hash 값의 동일성 등을 추가 검증하여 해당 인증토큰의 유효성을 확인하여야 한다.

4) 송신중계자가 whitelist에 있는 수신중계자(열람인증토큰 발급자)의 공개키로 토큰 내 전자서명값 검증에 실패하면 수신중계자에게 열람인증토큰 발급에 사용한 유효한 공개키를 요청하여 획득('❸ 공개키정보 요청')한 후, 이를 기반으로 열람인증토큰의 유효성을 검증하고 검증결과를 송신자에게 전달한다.

※ 이 절차는 열람인증토큰의 전자서명에 사용된 수신중계자의 인증서가 변경되었으나 수신중계자가 전담기관의 유통허브시스템에 이를 통지하고 갱신하지 않은 상황에 대처하기 위함

5) 송신중계자는 열람인증토큰의 검증결과(성공 또는 실패)에 따라 열람을 요청한 수신자에게 전자문서 원문을 제공하거나 열람불가 메시지를 전달하여야 한다.

6) 송신중계자는 수신자의 전자문서열람 요청에 성공적으로 응답하고 나면 열람결과와 열람시각을 수신중계자에게 전달('❹ 전자문서 열람결과 전달')하여야 한다.

7) 송신중계자가 송신자에게 열람결과 및 열람시각정보를 전달('⑤ 전자문서 열람정보 전달')하는 방식에 있어서는 송신중계자와 송신자 간의 협의에 따라 이루어진다.

□ 처리 절차별 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
❷ 전자문서 열람요청	▪ 수신자가 수신중계자로부터 본인확인 후 발급받은 열람인증토큰을 기반으로 송신자가 전달한 전자문서 열람 URL을 호출	수신자	송신자	
❸ 공캐키 정보 요청	▪ 송신중계자가 whitelist에 있는 열람인증토큰 발급자(수신중계자)의 공개키로 토큰 내 전자서명 값 검증에 실패하면 수신중계자에게 직접 공개키를 요청(획득)하는 인터페이스	송신 중계자	수신 중계자	
❹ 전자문서 열람결과 전달	▪ 송신중계자가 수신중계자에게 전자문서 열람시각 정보를 포함한 열람결과를 전달하는 인터페이스	송신 중계자	수신 중계자	“공인전자문서중계자와_송신시스템_간_표준연계_기술규격” 참조

4. 연계 인터페이스(API)

4.1. API 개요

중계자 간 연계방안의 각 단계에서 도출한 표준 연계 인터페이스(API)는 다음과 같다.

단계 구분	API		요청자	서비스 제공자 (응답자)	HTTP Method	요청 URI
	번호	API 이름				
(1) 중계자 정보획득 단계	❸	중계자 인증 토큰 발급	중계자	중계자	POST	https://[중계자서비스도메인]/auth/platform
(2) 전자문서 발송단계	❸	전자문서 중계	송신 중계자	수신중계자	POST	https://[중계자서비스도메인]/api/sendedoc
	❹	전자문서 수신결과 전달	수신 중계자	송신중계자	POST	https://[중계자서비스도메인]/api/notifyedocStatus
	❺	전자문서 처리상태 조회	송신 중계자	수신중계자	POST	https://[중계자서비스도메인]/api/edocStatus
(3) 전자문서 열람단계	❷	전자문서 열람요청	수신자	송신자 or 송신중계자	GET	https://[전자문서 열람 URL]
	❹	공개키 정보 요청	송신 중계자	수신중계자	GET	https://[중계자서비스도메인]/api/pubKey
	❻	전자문서 열람결과 전달	송신 중계자	수신중계자	POST	https://[중계자서비스도메인]/api/notifyreadDate

4.2. API 공통사항

중계자 간 연계를 위해 필요한 모든 API는 다음 기준에 맞춰 정의한다.

- HTTPS 기반의 REST 아키텍처를 기반으로 함
- 중계자 간 네트워크 보안은 TLS 1.2 이상으로 함
- API 요청자는 서비스 제공자로부터 “1. 사용자 인증토큰 발급”요청을 통해 access token을 획득한 후 access token을 기반으로 API를 호출함

4.2.1. HTTP 헤더

본 규격의 모든 메시지는 요청 및 응답에 따라 아래 명시된 헤더 값을 HTTP 헤더에 필수로 기술하여야 한다. 다만 Access Token을 발급받기 위한 “(1) 중계자정보 획득단계”의 “❸ 중계자 인증” API의 경우에는 헤더의 값 중 “Authorization” 값은 기술하지 않는다.

본 규격에서 각 중계자 간 연계를 위해 제공하도록 정의된 API는 “(1) 중계자정보 획득단계”의 “❸ 중계자 인증” API를 통해 인증토큰(Access Token)을 발급받고 이를 헤더의 “Authorization” 값에 전달하여 사용한다.

다만 이 중 “(3) 전자문서 열람단계”의 “❷ 전자문서 열람요청”에서는 수신자가 수신중계자로부터 발급받은 열람인증토큰을 헤더의 “Authorization” 값에 넣어 전달한다.

□ 요청메시지 HTTP 헤더

- Authorization: API를 제공하는 시스템(중계자 또는 유통허브)으로부터 인증절차를 거쳐 발급받은 인증토큰(Access Token)을 전달
- platform-id: 요청 중계자시스템의 인증번호를 전달
- req-UUID: 요청메시지의 고유식별값(UUID)을 전달
- req-date: 요청메시지 발송시각을 전달
  ex) 2020-10-01 22:34:30

□ 응답메시지 HTTP 헤더

- platform-id: API 호출에 응답하는 시스템이 중계자인 경우는 중계자시스템의 인증번호를, 유통허브시스템인 경우에는 “hub”를 전달
- res-UUID: 응답메시지의 고유식별값을 전달
- res-date: 응답메시지의 발송시각을 전달
  ex) 2020-10-01 22:34:30

□ HTTP 헤더 예시

```
POST /api/eaddr HTTP/1.1

Accept: application/json
Connection: keep-alive
Content-Length: 83
Content-Type: application/json
Host: relaySvc.aaa.com
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ2cdC5jb.....
platform-id: aaa-01-aaaapp
req-UUID: f73746a3-b65a-433f-a62b-8e5ee86aadf2
req-date: 2020-10-01 22:34:30.123
```

4.2.2. 요청 및 응답 메시지

□ 메시지 구조

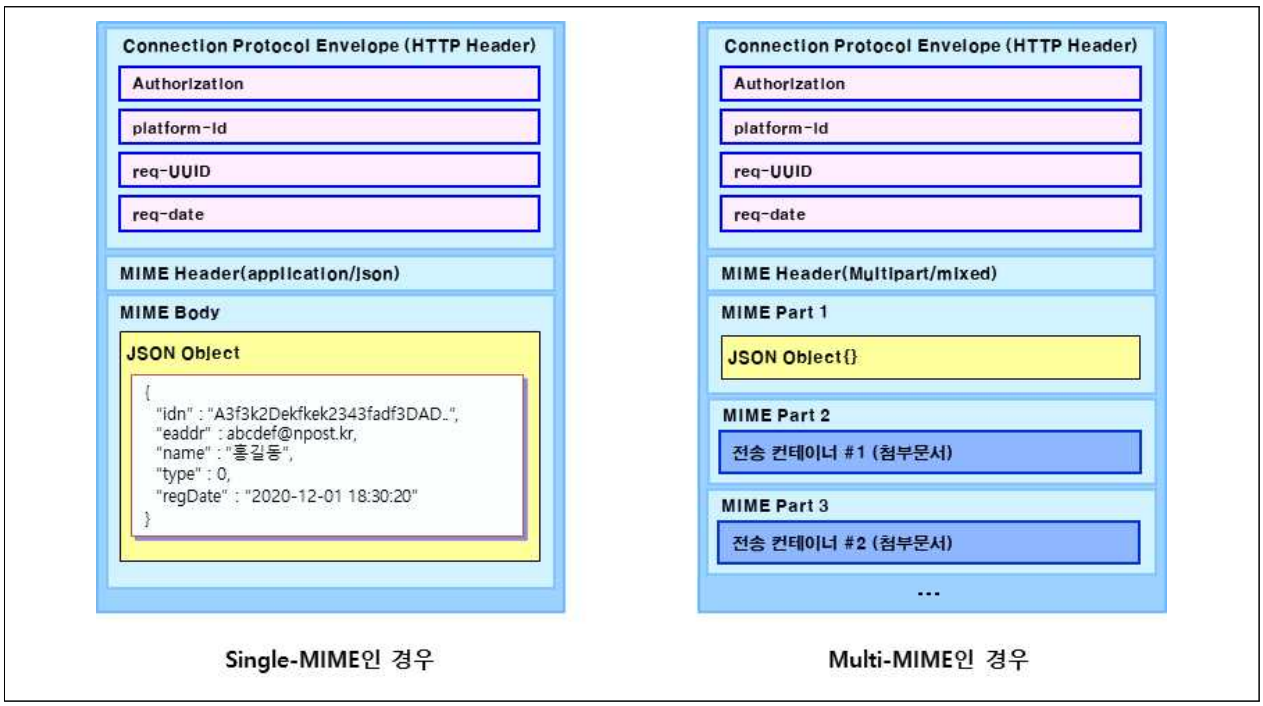
요청이나 응답메시지가 별도의 첨부문서 없이 업무메시지만으로 구성된 경우에는 메시지 헤더 정보의 content-type이 “application/json(또는 text/plain)”인 Single-MIME 구조를, 업무메시지 외에 첨부문서가 있는 경우에는 “multipart/mixed”인 Multi-MIME 구조로 전달한다.

메시지가 Multi-MIME 구조인 경우, 해당 메시지는 첫 번째 MIME에 업무메시지가, 두 번째 MIME부터는 첨부문서가 순차적으로 들어가는 구조로 구성된다.

각 API의 요청 및 응답메시지 구조는 API의 상세설명에서 정의한다.

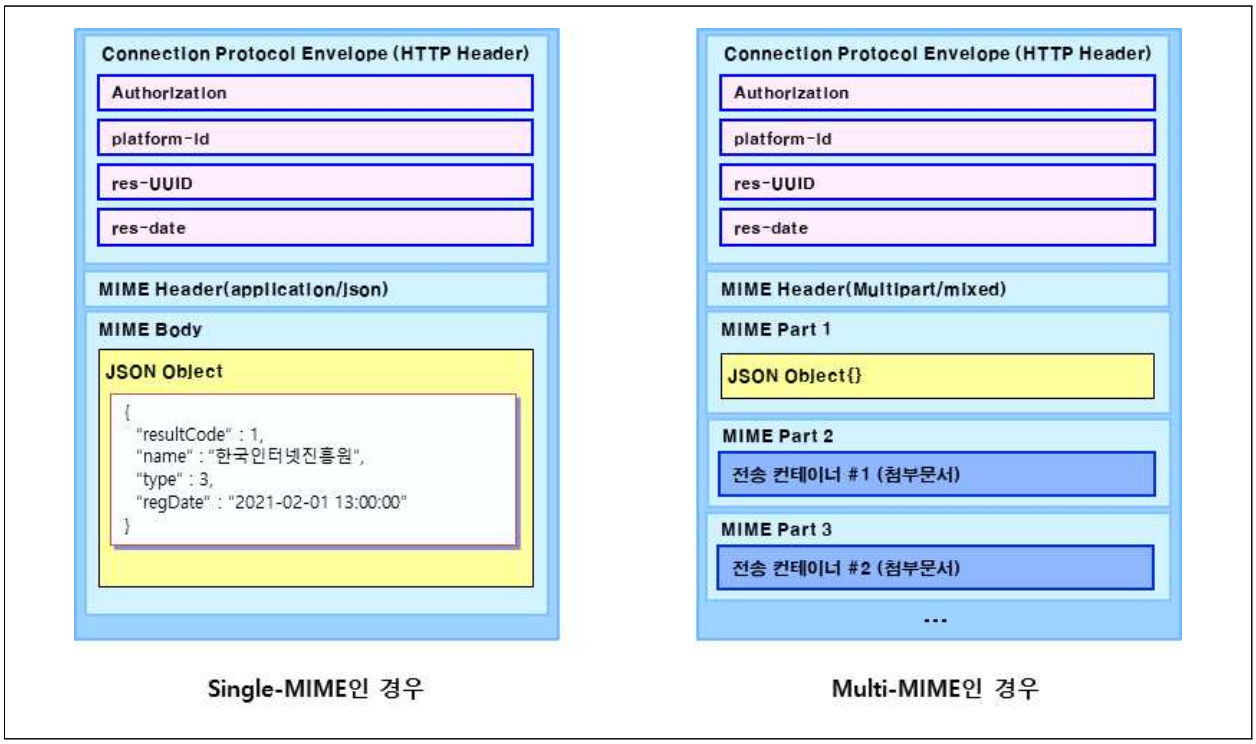
1) 요청메시지 요건

- HTTP 메소드가 GET인 API의 요청메시지는 일반적인 URI 호출방식을 사용하며 요청 파라미터에 대한 인코딩(UTF-8)을 필수로 수행해야 한다.
- HTTP 메소드가 POST, PUT, PATCH, DELETE 인 API의 요청메시지는 항상 업무메시지에 해당하는 JSON Object를 같이 전달하며, 요청메시지에 첨부문서가 있는 경우에는 Multi-MIME 구조로 구성하여 첨부문서도 같이 전달한다.
- 요청메시지 구성방안



2) 응답메시지 요건

- 본 규격에서 HTTP 상태코드 200은 요청메시지 전달의 성공만을 의미하며 요청 업무에 대한 처리결과가 성공하였음을 의미하지는 않는다. HTTP 통신상 오류 발생 시 이에 대해서는 오류원인에 따라 Bad Request(400), Unauthorized(401), Internal Server Error(500) 등의 HTTP 상태코드로 오류를 응답한다.
- 응답메시지의 업무메시지 영역에 있는 resultCode(성공:1, 실패:0)항목을 사용하여 요청에 대한 업무처리 결과의 성공 및 실패 여부를 전달한다.
- 응답메시지 구성방안



□ MIME 헤더 내 Content-Disposition

업무메시지와 첨부문서를 명확하게 식별하기 위해서 본 규격에서는 Content-Disposition의 name 필드를 활용하며, MIME으로 첨부해서 보낼 수 있는 content는 “업무메시지”, “알림메시지”, “첨부파일”이 있다. 이는 응답메시지도 동일하다.

- 업무메시지 : 각 요청메시지의 본문(JSON Object로 표현)에 해당하며 “msg“로 기술
- 알림메시지 : 전자문서 발송 시 선택적으로 삽입가능한 text/html, text/plain 타입의 content로서 “notice“로 기술함(전자문서 수신 알림메시지)
- 첨부파일 : 전자문서, 유통증명서 등 본 규약에서 주고받는 모든 파일은 ”file“로 기술

```
#업무메시지
Content-Disposition: attachment; name="msg"
Content-Type: application/json;

#알림메시지
Content-Disposition: attachment; name="notice"
Content-Type: text/html; charset="UTF-8"

# 첨부문서
Content-Disposition: attachment; name="file" filename="지방세납부고지서.jpg"
Content-Type: application/octet-stream
```

4.2.3. 데이터 표기방안

□ 시각정보 표기방안

본 규격에서 사용하는 표준시는 KST(대한민국 표준시)로 UTC(협정 세계시)에서 9시간을 더한 시간을 표준으로 사용하며, 입력받을 시간은 24시간을 기준으로 입력받는다.
(예시. 한국시각 기준 20년 12월 1일 오후 9시 35분 43초일 경우 → 2020-12-01 21:35:43)

□ 데이터 타입별 표기방안

타입	정의	표기(예시)	해석
int	정수형 데이터	3	최대 3자리의 정수(실제 범위는 각 API별 설명참고)
String	문자열 데이터	10, 88	10 또는 88길이의 고정길이 문자열
		3~64	최소길이 3, 최대길이 64인 변동길이 문자열
Array	배열형 데이터	1~500	1~500개 원소를 가지는 배열

본 규격에서 데이터타입에 따른 표기 및 해석은 다음과 같다.

* 문자열 길이는 byte 수가 아닌 글자 수를 의미한다(ex. "가나다"=3, "abc"=3, "123"=3)

4.3. 연계 API 상세설명

4.3.1. 중계자 인증토큰 발급

본 API는 중계자가 타 중계자가 제공하는 서비스 인터페이스를 연계하기 전에 타 중계자에게 인

증을 요청하기 위해 사용하는 인터페이스로 서비스를 제공하는 중계자에 의해 제공되는 API이다.

중계자는 타 중계자가 제공하는 다른 모든 API를 이용하고자 할 때, 접근을 허가받기 위해 HTTP 헤더의 "Authorization"항목에 본 API를 통해 발급받은 인증토큰(Access Token)을 넣은 요청 메시지를 전달한다.

1) API 기존정보

요청 URI	메소드	응답형식
https://[중계자서비스도메인]/auth/platform	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수 여부	설명
grantType	String	18	Y	토큰요청 유형 "client_credentials" 고정
clientId	String	16~32	Y	클라이언트 ID
clientAssertionType	String	54	Y	JWT 기반 인증방식을 나타내는 식별자 "urn:ietf:params:oauth:client-assertion-type:jwt-bearer" 고정
clientAssertion	String	32~ 2048	Y	요청자의 개인키로 서명한 JWT 인증 토큰

※ clientAssertion의 구조는 "부록 A. 중계자 인증 관련 토큰" 부분 참조

□ 요청메시지 예시

```
{
  "grantType": "client_credentials"
  "clientId": "client123",
  "clientAssertionType"="urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
  "clientSecret": "eyJhbGciOiJFUzI1NiIs..."
}
```

□ 응답메시지 구조

파라미터	타입	길이	필수 여부	설명
resultCode	int	1	Y	처리결과 (1:성공, 0:실패)

성공	accessToken	String	32~2048	Y	액세스 토큰
	tokenType	String	6	Y	토큰유형 “Bearer” 고정
	expiresIn	int	6	Y	토큰 유효시간(초)
	scope	String	-	N	허가된 권한 범위
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

※ “액세스 토큰”이란 중계자가 인증을 요청하는 요청자에게 사용자 인증 후 발급하는 인증토큰을 말한다.

※ accessToken 구조는 “부록 A. 중계자 인증 관련 토큰” 부분 참조

□ 응답메시지 예시

```
{
  “resultCode“ : 1,
  “accessToken“ : “eyJhbGciOiJIUzI1NiIs...“,
  “tokenType“: “Bearer“,
  “expiresIn“: 3600
}
```

4.3.2. 전자문서 중계

본 API는 송신중계자가 수신중계자에게 전자문서를 송신하기 위해 사용하는 인터페이스로 수신중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/sendedoc	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수 여부	설명
edocList	array	1..N	Y	전송하고자 하는 전자문서정보 목록
edocNum	String	33	Y	전자문서번호
subject	String	100	Y	전자문서제목
edocCode	String	10	N	전자문서유형 정보(발송 전자문서 구분을 위해 송신자가 관리하는 전자문서코드 정보)
sendEaddr	String	3~64	Y	송신자의 공인전자주소
sendSubEaddr	String	3~64	N	송신자 공인전자주소의 하위 계정(*)
recvEaddr	String	3~64	Y	수신자의 공인전자주소
recvSubEaddr	String	3~64	N	수신자 공인전자주소의 하위 계정(*)
sendDate	String	19	Y	송신일시
transferType	String	4	Y	전자문서 전송방식(url(**) 또는 file)
url	String	2048	N	열람 url
contentsList	array	1..N	N	전송대상 알림메시지 및 첨부파일의 해시정보
contentId	String	100	N	전송 전자문서 메시지 패키지 내에서 콘텐트(알림메시지 또는 첨부파일)를 포함한 MIME의 content-id 값 - 전송메시지 내 multi MIME 중 해당하는 콘텐트가 첨부된 MIME의 content-id 값을 기술 - 파일이 직접 첨부되지 않은 URL방식의 경우 이 항목을 기술하지 않으나, 이 경우에도 알림메시지를 첨부한 경우에는 이 항목에 안내문이 들어간 MIME의 content-id를 기술
contentHash	String	44, 66	Y	contentId에 해당하는 안내문 및 전자문서의 해쉬값 contentId가 없는 경우에는 URL로 전달하는 전자문서의 해쉬값을 기술
authExpTime(***)	String	19	N	문서열람 인증 만료일시
readExpTime(****)	String	19	N	문서열람 만료일시

* 공인전자주소의 하위 계정: 기업・기관의 경우에는 주소 하위에 복수의 계정을 보유할 수 있음

** url: transferType이 url일 경우 필수항목임

*** authtExpTime: 발송기관이 설정한 값으로 열람인증이 가능한 일자로 이 시각까지 한번이라도 열람이 되지 않은 경우에는 문서 열람이 불가함

**** readExpTime: 발송기관이 설정한 값으로 이 시각 이후로는 문서의 재열람이 불가함

□ 요청메시지 예시

```
{ "edocList" : [
  {"edocNum" : "20220107_KISA000001_0001234567890",
   "subject" : "보험계약서",
   “edocCode”: “a00001”,
   "sendEaddr" : "aaa_123",
   "recvEaddr" : "abc_12345",
   "sendDate" : "2021-01-07 09:00:00",
   "transferType" : "url",
   "url" : "https://npost.kr/msg/2d6e1a624aafcb60d95728f...",
   "contentsList" : [
      { "contentId" : "notice_001",
        "contentHash" : "074f2c0ad51e2d6e1a624aafcb60d95728fd02854022d0dc8d90334564303b5" },
      { "contentHash" : "0dsr3c0ad5sd3d6e1a624aafcb60d95728fd02854022d0dc8d903ew16435tg5" }
    ] } ,
  {"edocNum" : "20220107_KISA000001_0002345678901",
   "subject" : "보험계약서",
   “edocCode”: “a00001”,
   "sendEaddr" : "aaa_123",
   "recvEaddr" : "def_67890",
   "sendDate" : "2021-01-07 09:00:00",
   "transferType" : "url",
   "url" : "https://npost.kr/msg/2d6e1a624aafcb60d95728f...",
   "contentsList" : [
      { "contentId" : "notice_002",
        "contentHash" : "074f2c0ad51e2d6e1a624aafcb60d95728fd02854022d0dc8d90334564303b5" },
      { "contentHash" : "0dsr3c0ad5sd3d6e1a624aafcb60d95728fd02854022d0dc8d903ew16435tg5" }
    ] } ,
  {"edocNum" : "20220107_KISA000001_0003456789012",
   "subject" : "보험계약서",
   “edocCode”: “a00001”,
   "sendEaddr" : "aaa_123",
   "recvEaddr" : "ghi_12345",
   "sendDate" : "2021-01-07 09:00:00",
   "transferType" : "url",
   "url" : "https://npost.kr/msg/2d6e1a624aafcb60d95728f...",
   "contentsList" : [
      { "contentId" : "notice_003",
        "contentHash" : "074f2c0ad51e2d6e1a624aafcb60d95728fd02854022d0dc8d90334564303b5" },
      { "contentHash" : "0dsr3c0ad5sd3d6e1a624aafcb60d95728fd02854022d0dc8d903ew16435tg5" }
    ] }
] }
```

- 송신자가 전송하는 전자문서가 모두 transferType이 “url”이고 전자문서안내문을 보내지 않는 경우에 요청메시지는 single MIME 구조를 가지며, contentsList내에 contentId 항목이 존재하지 않음
- 송신자가 전송하는 전자문서가 transferType이 “url”이고 전자문서안내문을 보내는 경우에 요청메시지는 multi MIME 구조며, 첨부파일이 들어가는 “MIME Part 2”부터 전자문서 안내문(content-disposition의 name항목이 “notice”임)을 포함하여 전송함. 이때 contentsList내에 contentId 항목에는 전자문서 안내문을 포함한 MIME의 content-id 값을 기술
- 송신자가 전송하는 전자문서가 transferType이 “file”인 경우에는 전자문서안내문의 전송여부에 관계없이 요청메시지는 multi MIME 구조며, 첨부파일이 들어가는 “MIME Part 2”부터 전자문서안내문 또는 첨부파일(content-disposition의 name항목이 “notice”나 “file”임)을 포함하여 전송함. 이때 contentsList내에 contentId 항목에는 전자문서안내문 또는 첨부파일을 포함한 MIME의 content-id 값을 기술

□ 응답메시지 구조

파라미터				타입	길이	필수 여부	설명
resultCode				int	1	Y	처리결과 (1:성공, 0:실패)
접수완료또는수신완료	reqAccTime			String	19	Y	요청메시지 접수시간
	resultList			array	1..N	N	전자문서별 수신 처리결과 목록 (resultCode가 1인 경우)
		edocNum		String	33	Y	수신한 전자문서번호
		recvResultStatus		int	1	Y	전자문서번호별 처리결과 [정상처리 상태] 2 : 송신완료(접수완료). 3 : 수신완료 [실패/예외 상태] 20 : 송신실패(접수실패), 30 : 수신실패
		성공	recvDate	String	19	N	전자문서 수신일시(수신자에게 수신문서 전달 성공 시, i.e. recvResultStatus 값이 3인 경우)
		실패	recvErrCode	String	10	Y	전자문서번호별 전자문서 수신오류 시 오류코드 (recvResultStatus 값이 20, 30인 경우)
			recvErrMsg	String	256	N	전자문서번호별 전자문서 수신오류 시 오류메시지
실패	errCode			String	10	Y	요청메시지 처리 과정의 오류 시 오류코드 (resultCode가 0인 경우)
	errMsg			String	256	N	요청메시지 처리 과정의 오류 시 오류메시지

□ 응답메시지 예시


```
{ "resultCode" : 1,
  "resultList" : [
  {"edocNum" : "20220107_KISA000001_0001234567890",
   "recvResultStatus" : 3,
   "recvDate" : "2021-01-07 09:01:00" },
  {"edocNum" : "20220107_KISA000001_0002345678901",
   "recvResultStatus" : 3,
   "recvDate" : "2021-01-07 09:01:02" },
  {"edocNum" : "20220107_KISA000001_0003456789012",
   "recvResultStatus" : 20,
   "recvErrCode" : "ERR-02-204",
   "recvErrMsg" : "수신자 공인전자주소 없음"}
] }
```


4.3.3. 전자문서 수신결과 전달

본 API는 전자문서 수신중계자가 송신중계자로부터 전자문서를 접수받은 후, 비동기식으로 수신처리를 완료한 다음 송신중계자에게 수신처리한 결과정보를 전달하는 인터페이스이다. 이때 수신처리라 함은 수신중계자가 송신중계자로부터 전달받은 전자문서에 대해 수신자 정보를 찾아서 수신자의 문서함에 전자문서를 전달하고, 수신자에게 전자문서가 접수되었음을 통지하는 과정을 말한다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/notifyedocStatus	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터		타입	길이	필수 여부	설명
resultList		array	1~N	Y	전자문서번호별 처리상태 목록
	edocNum	String	33	Y	전자문서번호
	status	int	1	Y	[정상 처리 상태] 3 : 수신완료 [실패/예외 상태] 30 : 수신실패
	recvDate	String	19	N	수신일시(처리상태가 3인 경우 필수)
	edocErrCode	String	10	N	전자문서번호별 전자문서 수신오류 시 오류코드(status 값이 30 인 경우)
	edocErrMsg	String	256	N	edocErrCode에 대한 오류 메시지

□ 요청메시지 예시

```
{ "resultCode" : 1,
  "resultList" : [
  {"edocNum" : "20220107_KISA000001_0001234567890",
   "status" : 3,
   "recvDate" : "2021-01-07 09:01:00" },
  {"edocNum" : "20220107_KISA000001_0002345678901",
   "status" : 30,
   "edocErrCode" : "ERR-02-504",
   "edocErrMsg" : "수신자 공인전자주소 오류"} ]
}
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

□ 응답메시지 예시

```
{
 “resultCode” : 1
}
```

4.3.4. 전자문서 처리상태 조회

본 API는 전자문서 송신중계자가 발송한 전자문서의 수신 또는 열람상태에 대한 결과값을 받지 못했을 때 수신중계자에게 이를 확인하기 위해 처리상태를 요청하는 인터페이스로 수신중계자에 의해 제공된다.

※ 송신중계자와 수신중계자는 다음과 같은 기준으로 전자문서 처리상태 코드 값을 공유한다.

상태명	코드값	설명
송신완료 (접수완료)	2	송신중계자가 수신중계자에게 전자문서 송신은 완료하였으나 아직 수신중계자로부터 수신자에게 전자문서를 정상적으로 전달하였음을 확인받지 못한 상태로서, 송신중계자가 비동기로 수신중계자에게 전자문서를 전송하고, 수신중계자로부터 수신접수까지만 완료된 상태임
수신완료	3	수신중계자가 수신한 전자문서를 정상적으로 수신자에게 전달하였으나 수신자가 아직 열람하지는 않은 상태임
열람완료	4	수신자가 수신한 전자문서를 정상적으로 열람한 상태임
송신실패 (미접수)	20	수신자가 공인전자주소에 가입하지 않았거나, 수신중계자 시스템의 오류, 네트워크상의 오류 등의 이유로 송신중계자가 전자문서 송신에 실패한 상태임 (수신중계자가 송신중계자에게 전자문서 처리상태 조회에 대한 응답을 보낼 때는 해당 전자문서 자체를 수신하지 못한 미수신 상태를 지칭함)
수신실패	30	수신중계자가 수신한 전자문서를 수신자에게 전달하지 못한 상태로서, 수신자의 미가입, 탈퇴 등의 이유로 수신자의 공인전자주소가 없는 상태임
열람실패	40	수신자가 수신한 전자문서를 열람하는 과정에서 열람기한 만료, 전자문서 열람시스템의 오류 등의 이유로 전자문서 열람에 실패한 상태임(단, 최초 열람이 성공하여 열람상태가 ‘4’가 된 상태에서는 이후 열람이 실패하더라도 상태가 ‘40’로 변경되지는 않음)

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/edocStatus	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수 여부	설명
edocNums	array	1..N	Y	처리 상태를 알고 싶은 전자문서번호 목록

□ 요청메시지 예시

```
{
 “edocNums” : [ “20220107_KISA000001_0001234567890”, ″20220107_KISA000001_0002345678901″,
″20220107_KISA000001_0003456789012″ ]
}
```

□ 응답메시지 구조

파라미터			타입	길이	필수 여부	설명
resultCode			int	1	Y	처리결과 (1:성공, 0:실패)
성공	resultList		array	1~N	Y	전자문서번호별 처리상태 목록
		edocNum	String	33	Y	전자문서번호
		status	int	1	Y	전자문서 처리상태 [정상 처리 상태] 2 : 송신완료(접수완료). 3 : 수신완료, 4 : 열람성공 [실패/예외 상태] 20 : 송신실패(접수실패), 30 : 수신실패, 40 : 열람실패)
		recvDate	String	19	N	수신일시(처리상태가 3, 4, 40인 경우 필수)
		readDate	String	19	N	열람일시(처리상태가 4인 경우 필수이며 최초 열람일시)
		edocErrCode	String	10	N	전자문서번호별 전자문서 수신오류 시 오류코드(status 값이 20, 30, 40인 경우)
		edocErrMsg	String	256	N	전자문서번호별 전자문서 수신오류 시 오류메시지
실패	errCode		String	10	Y	요청메시지 처리 과정의 오류 시 오류코드
	errMsg		String	256	N	요청메시지 처리 과정의 오류 시 오류메시지

□ 응답메시지 예시

```
{ "resultCode" : 1,
  "resultList" : [
  {"edocNum" : "20220107_KISA000001_0001234567890",
   "status" : 3,
   "recvDate" : "2021-01-07 09:01:00" },
  {"edocNum" : "20220107_KISA000001_0002345678901",
   "status" : 4,
   "recvDate" : "2021-01-07 09:01:02",
   "readDate" : "2021-01-07 09:10:30" },
  {"edocNum" : "20220107_KISA000001_0003456789012",
   "status" : 30 }
] }
```


4.3.5. 전자문서 열람요청

수신자가 전자문서 열람을 요청하면 수신중계자는 열람을 위한 본인인증을 수행한 후 수신자에게 열람인증토큰을 발급해준다.

수신자는 이 열람인증토큰을 HTTP헤더의 "Authorization" 항목에 넣어 송신자가 전달한 전자문서 열람 URL을 호출한다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[전자문서 열람 URL]	GET	HTTP response

2) 메시지 구조

□ 요청메시지 구조

- 비즈니스 요청정보 없음

□ 요청메시지 예시

```
https://[전자문서 열람 URL]
```

□ 응답메시지 구조

- HTTP response 구조

□ 응답메시지 예시

```
HTTP/1.1 200 OK
Server: apache
Content-Type: text/html
Connection: keep-alive
Content-Length: 250

<!-- 열람하고자 하는 전자문서 본문 ---->
```

4.3.6. 공개키 정보 요청

발송자로부터 열람인증토큰에 대한 검증을 요청받은 송신중계자는 열람인증토큰을 발급한 수신중계자의 공개키(열람인증토큰에 있는 발급자의 플랫폼 ID 정보를 기반으로 whitelist에서 공개키를 검색)로 열람인증토큰을 검증하여야 한다.

이때 열람인증토큰 검증에서 오류가 발생되면 송신중계자는 수신중계자가 제공하는 본 API를 이용하여 수신중계자에게 열람인증토큰 발급에 사용한 인증서의 공개키 정보를 요청하여 이를 기반으로 열람인증토큰을 재검증한다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/pubKey	GET	JSON

2) 메시지 구조

□ 요청메시지 구조

- 비즈니스 요청정보 없음

□ 요청메시지 예시

```
https://[중계자서비스 도메인]/api/pubKey
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
성공	pubKey	String	-	Y	열람인증토큰 발급 시 사용한 공개키를 base64로 인코딩한 값
	expTime	String	19	Y	공개키의 유효기간 만료일시
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

□ 응답메시지 예시

```
{
 "resultCode" : 1,
 "pubKey" : "MIIF6jCCBNKgAwIBAgIDSUkdMA0GCSqGSIb3DQEBCwUAME8xCzAJBgNVBAY..",
 "expTime": "2022-08-15 23:59:59"
}
```

4.3.7. 전자문서 열람정보 전달

본 API는 전자문서 수신중계자가 수신자의 전자문서 열람사실을 전자문서 송신중계자에게 전달하기 위해 사용하는 인터페이스로 송신중계자에 의해 제공된다.

열람일시는 기전달된 수신일시보다 미래시점이면서 현재시각보다 과거시점이어야 한다. 송신중계자는 시각정보가 역전되었는지를 검증해야 한다. 송신중계자는 전자문서의 열람일시가 한번 등록되면 수신중계자에 의해 열람정보가 재수신되어도 기존 정보가 수정되지 않도록 처리해야 한다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/notifyreadDate	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수 여부	설명
edocNum	String	33	Y	전자문서번호
readDate	String	19	Y	열람일시

□ 요청메시지 예시

```
{
 “edocNum” : “20220107_KISA000001_0001234567890”,
 “readDate” : “2022-01-07 13:10:00”
}
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

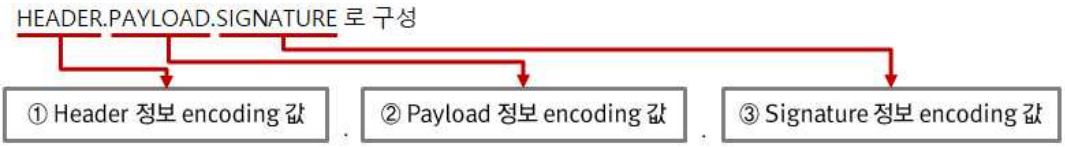
□ 응답메시지 예시

```
{
 “resultCode” : 1
}
```

A. 부록 -중계자 인증 관련 토큰

1) 중계자 인증 관련 토큰 구조의 개요

- 중계자 연계 전 인증을 위한 “중계자 인증토큰 발급” API에서 사용하는 토큰은 2가지로 “clientAssertion”, “accessToken”이 있다.
- 토큰은 JWT(JSON Web Token) 구조체로 기본 구조는 다음과 같다.
- 기본구조



2) 중계자 인증 관련 토큰의 항목별 구성내역

토큰의 구성항목 중 Header와 signature의 값은 2개 토큰이 모두 동일한 구조를 가지고 있으며, payload의 경우만 다른 구성을 가지고 있다.

각 항목을 구성하는 정보구조는 다음과 같다.

□ Header 정보 encoding 값

- JWT를 검증하는데 필요한 정보로 typ(토큰 타입), alg(서명알고리즘), kid(검증에 사용할 공개키 식별값)로 구성됨
- Header 정보를 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용

```
{
“alg” : “ES256”
“typ” : “JWT”,
“kid” : “05c43a1f”
}
```

→ header

```
HEADER = base64UrlEncode(header) =
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjA1YzQzYTFmIn0
```

□ Payload 정보 encoding 값

- Palyload는 토큰에 전달하고자 하는 값을 포함하는 파트임

- “clientAssertion”의 payload 정보 구성
  - iss(토큰발급자인 중계자 플랫폼-ID), iat(토큰발급시각), exp(토큰만료시각), domain(중계자 서비스 도메인)으로 구성됨
- “accessToken”의 payload 정보 구성
  - iss(토큰발급자인 중계자 플랫폼-ID), sub(토큰수신자인 중계자 플랫폼-ID), iat(토큰발급시각), exp(토큰만료시각)로 구성됨
- 각 토큰은 aud, scope 및 중계자시스템 업무 수행에 필요한 사용자 정의 클레임(Custom Claim)등을 추가할 수 있다.
- 각 토큰에 따라 구성된 Payload 정보를 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용
- “clientAssertion”의 payload 예시

```
{
“iss” : “platform-01-app”,
“iat” : 1768010222,
“exp” : 1768010225,
“domain” : “http://api.relayAgency_A.co.kr”
}
```

payload

```
PAYLOAD = base64UrlEncode(payload) =
eyJpc3MiOiJwbGF0Zm9ybS0wMS1hcHAiLCJpYXQiOjE3NjgwMTAyMjIsImV4cCI6MTc2ODAxMDIyNSwiZG9tYWluIjoiaHR0cDovL2FwaS5yZWxheUFnZW5jeV9BLmNvLmtyIn0
```

- “accessToken”의 payload 예시

```
{
“iss” : “torpedo-01-bizframe”,
“sub”: “platform-01-app”,
“iat” : “1768010225“,
“exp”: “1768013825”
}
```

payload

```
PAYLOAD = base64UrlEncode(payload) =
eyJpc3MiOiJ0b3JwZWRvLTAxLWJpemZyYW1lIiwiYXVkIjoicGxhdGZvcm0tMDEtYXBwIiwiaWF0IjoxNzY4MDEwMjI1LCJleHAiOjE3NjgwMTM4MjV9
```

□ Signature 정보 encoding 값

- 인코딩된 헤더와 페이로드를 점(.)으로 연결한 후 헤더에 정의된 알고리즘과 개인키를 이용하여 서명한 값으로 구성됨
- 서명한 값을 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용

```
SIGNATURE = ECDSASHA256(HEADER + "." + PAYLOAD, privateKey) =
ZdrQfu9cWkkhZzG4g-vAuooqDF6cFsuXgREJbRnX8jP_Df4_jUTnO0Fa6P3azT1ZTTBaf1KrpK1BuHXWzWf5eQ
```

□ 중계자 인증관련 토큰 예시

- Header, Payload, Signature 값을 "."으로 연결한 값

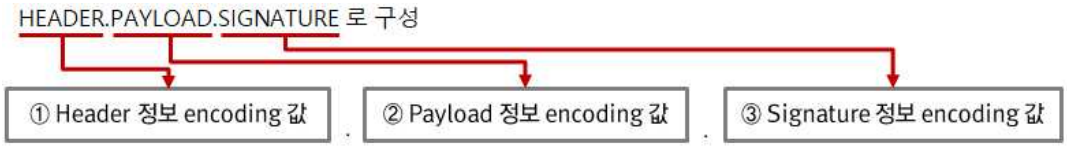
```
중계자 인증토큰 clientAssertion(JWT) =
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjA1YzQzYTFmIn0.eyJpc3MiOiJwbGF0Zm9ybS0wMS1hcHAiLCJpYXQiOjE3NjgwMTAyMjIsImV4cCI6MTc2ODAxMDIyNSwiZG9tYWluIjoiaHR0cDovL2FwaS5yZWxheUFnZW5jeV9BLmNvLmtyIn0.ZdrQfu9cWkkhZzG4g-vAuooqDF6cFsuXgREJbRnX8jP_Df4_jUTnO0Fa6P3azT1ZTTBaf1KrpK1BuHXWzWf5eQ

중계자 인증토큰 accessToken (JWT) =
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjE0MDIzYjJmIn0.eyJpc3MiOiJ0b3JwZWRvLTAxLWJpemZyYW1lIiwiYXVkIjoicGxhdGZvcm0tMDEtYXBwIiwiaWF0IjoxNzY4MDEwMjI1LCJleHAiOjE3NjgwMTM4MjV9.MMgBGElhmc3yC7W4GmXshxaE_Zlp4mtW_W-eB3zmEAwnxaWdtZRzfd-GGxD_OG4cUsP8vpgVPQnoFNK_obaoVg
```

B. 부록 -열람인증토큰

1) 열람인증토큰 구조 개요

- 열람인증토큰은 JWT(JSON Web Token) 구조체
- 기본구조



2) 열람인증토큰 항목별 구성내역

□ Header 정보 encoding 값

- JWT를 검증하는데 필요한 정보로 typ(토큰 타입), alg(서명알고리즘), kid(검증에 사용할 공개키 식별값)로 구성됨
- Header 정보를 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용

```
{
“alg” : “ES256”
“typ” : “JWT”,
“kid” : “05c43a1f”
}
```

header

HEADER = base64UrlEncode(header) =
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjA1YzQzYTFmIn0

□ Payload 정보 encoding 값

- 토큰에 전달하고자 하는 값을 포함하는 파트로 iss(토큰발급자로서 중계자의 플랫폼-ID), sub(토큰수신자로서 수신자 공인전자주소), iat(토큰발급시각), exp(토큰만료시각), edocNum(전자문서번호), contentHash(전자문서해시값)로 구성됨
- 중계자시스템 업무 수행에 필요한 사용자 정의 클레임(Custom Claim)등 추가 가능
- Payload 정보를 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용

```
{
“iss” : “platform-01-app”,
“sub”: “user1245a“,                                    payload
“iat” : 1768010222,
“exp”: 1768010225,
“edocNum”: “20210701_KISA000001_0001234567890”,
“contentHash”: “44cb730c420480a0477b505ae68af508fb90f96cf0ec54c6ad16949dd427f13a”
}

PAYLOAD = base64UrlEncode(payload) =
eyJpc3MiOiJwbGF0Zm9ybS0wMS1hcHAiLCJhdWQiOiJ1c2VyMTI0NWEiLCJpYXQiOjE3Njg
wMTAyMjIsImV4cCI6MTc2ODAxMDIyNSwiZWRvY051bSI6IjIwMjEwNzAxX0tJU0EwMDAw
MDFfMDAwMTIzNDU2Nzg5MCIsImNvbnRlbnRIYXNoIjoiNDRjYjczMGM0MjA0ODBhMDQ3
N2I1MDVhZTY4YWY1MDhmYjkwZjk2Y2YwZWM1NGM2YWQxNjk0OWRkNDI3ZjEzYSJ9
```

□ Signature 정보 encoding 값

- 인코딩된 헤더와 페이로드를 점(.)으로 연결한 후 헤더에 정의된 알고리즘과 개인키를 이용하여 서명한 값으로 구성됨
- 서명한 값을 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용

```
SIGNATURE = ECDSASHA256(HEADER + ″.″ + PAYLOAD, privateKey) =
MEUCIQCkUsmkMVgIpWKOoW7Upl0ePnLxNvH4YQ5NZQITby4L7wIgc9TkpNI1_98Q571H
wS9kEtMvtBfaTpUnK80fpoIU0Wg
```

□ 열람인증토큰 예시

- Header, Payload, Signature 값을 “.”으로 연결한 값

```
열람인증토큰(JWT) =
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjA1YzQzYTFmIn0.eyJpc3MiOiJwbGF0Z
m9ybS0wMS1hcHAiLCJhdWQiOiJ1c2VyMTI0NWEiLCJpYXQiOjE3NjgwMTAyMjIsImV4cCI6
MTc2ODAxMDIyNSwiZWRvY051bSI6IjIwMjEwNzAxX0tJU0EwMDAwMDFfMDAwMTIzNDU
2Nzg5MCIsImNvbnRlbnRIYXNoIjoiNDRjYjczMGM0MjA0ODBhMDQ3N2I1MDVhZTY4YWY1
MDhmYjkwZjk2Y2YwZWM1NGM2YWQxNjk0OWRkNDI3ZjEzYSJ9.MEUCIQCkUsmkMVgIp
WKOoW7Upl0ePnLxNvH4YQ5NZQITby4L7wIgc9TkpNI1_98Q571HwS9kEtMvtBfaTpUnK80f
poIU0Wg
```

C. 부록 -오류코드 정의

번호	코드 값	코드 값 정의	설명	비고
1	ERR-02-001	요청메시지 구조오류	요청메시지가 기술규격에 정의된 구조에 맞지 않은 경우 (문법적 오류)	API 공통오류
2	ERR-02-002	인증토큰 오류	요청메시지의 HTTP헤더에 있는 인증토큰이 유효하지 않은 경우	
3	ERR-02-003	메시지 헤더정보 오류	메시지 헤더정보의 값이 적절하지 않은 경우에 대한 오류(platform-id 값 오류, req-UUID 중복오류, req-date 값 오류 등)	
4	ERR-02-004	메시지 패키지 오류	메시지 패키지가 적절하지 않아서 발생된 오류 각 API의 요청 및 응답메시지에 따라 MIME 패키지(single-MIME 또는 multi-MIME)가 적절하게 구성되지 못한 경우	
5	ERR-02-005	응답시스템 내부 오류	응답시스템 내부의 오류에 의해 처리되지 못한 경우	
6	ERR-02-101	인증요청토큰 정보의 오류	인증토큰 발급을 위해 전달한 요청토큰(clientAssertion) 내의 Header, payload에 기술된 정보의 오류	“4.3.1. 중계자 인증토큰 발급” API 오류
7	ERR-02-102	인증요청토큰 서명값 오류	인증토큰 발급을 위해 전달한 요청토큰(clientAssertion) 내의 서명값(Signature)의 오류	
8	ERR-02-201	전자문서 중복오류	동일한 전자문서번호로 기 수신된 동일한 전자문서가 있음(제목, 송수신자 공인전자주소, 전자문서 URL 및 첨부파일 해쉬값이 모두 동일한 경우)	“4.3.2. 전자문서 중계” API 응답 시, “recvErrCode” 항목 또는 “4.3.3. 전자문서 수신결과 전달” API 응답 시, “edocErrCode” 항목에 기술되는 오류 “4.3.4. 전자문서 처리상태 조회” API 응답 시, “edocErrCode” 항목에 기술되는 오류
9	ERR-02-202	전자문서번호 중복 오류	기 수신된 전자문서번호와 동일한 전자문서번호이나 서로 다른 전자문서인 경우(제목, 송수신자 공인전자주소, 전자문서 열람 URL 및 첨부파일 해쉬값 중 하나라도 다른 경우)	
10	ERR-02-203	송신자 공인전자주소 오류	송신자 공인전자주소가 기재되어 있지 않은 경우	
11	ERR-02-204	수신자 공인전자주소 오류	수신중계자에게 수신자의 공인전자주소(하위계정이 있는 경우 이를 포함)가 존재하지 않거나 현재 사용 중이지 않는 경우	
12	ERR-02-205	전자문서 URL 오류	전송방식이 URL 방식의 전자문서에서 URL값이 유효하지 않은 경우	
13	ERR-02-206	콘텐트 정보의 오류	multi-MIME패키지 내에 contentId에 해당하는 콘텐트가 없거나 해시값이 일치하지 않는 경우	
14	ERR-02-207	전송방식의 오류	전송방식이 URL방식의 전자문서에서 contentid에 해당하는 콘텐트가 알림메시지가 아닌 첨부파일인 경우	

번호	코드 값	코드 값 정의	설명	비고
15	ERR-02-208	만료일시 표기 오류	“문서열람 인증 만료일시(authExpTime)” 또는 “문서열람 만료일시(readExpTime)”의 일시가 일시 데이터 표기 방법에 맞지 않는 경우	
16	ERR-02-301	전자문서 열람 실패	열람인증토큰 검증에 실패, 열람기일 만료, 존재하지 않는(유효하지 않은) URL 등 다양한 오류로 인해 열람 실패한 경우	“4.3.4. 전자문서 처리상태 조회” API 응답 시 , “edocErrCode” 항목에 기술되는 오류
17	ERR-02-401	미발송 전자문서	송신중계자가 열람정보에 해당하는 전자문서를 발송한 이력이 없음	“4.3.7. 전자문서 열람정보 전달” API 오류
18	ERR-02-402	열람일시 표기 오류	“열람일시(readDate)”의 일시가 일시 데이터 표기 방법에 맞지 않는 경우	
19	ERR-02-999	기타 오류	정의된 오류 이외의 오류	

규 격 연 혁

버전	제・개정일	제・개정내역
v1.0	2026년 6월 11일	▪ 제정