

명세서

발명의 명칭: 전자문서 유통 시스템 및 전자문서 유통 방법

기술분야

- [1] 본 발명은 기업/기관 등과 함께 일반 개인, 소기업들도 신뢰성을 확보받을 수 있는 전자문서 유통 체계를 구축하는 것이 가능한 전자문서 유통 시스템 및 전자문서 유통 방법에 관한 것이다.

배경기술

- [2] 일반적으로 전자문서 유통은 기업/기관들이 개별적인 고유 규약을 기반으로 특정 산업군 또는 커뮤니티 내에서만 한정적으로 이루어져 왔다.
- [3] 또한, 일반 개인들 간, 개인과 기업/기관들 간에는 신뢰적 전자유통의 개념이 없이 이메일을 보조 수단으로 활용하거나, 개인, 개인사업자, 소기업들이 대기업 사이트에 접속하는 방법을 통해서만 온라인 소통이 가능한 단점이 있어왔다.
- [4] 따라서, 일정 규모의 유통 시스템을 보유할 수 있는 기업뿐만 아니라, 일반 개인, 개인사업자, 소기업들도 유통에 대한 신뢰성을 보장받을 수 있는 전자문서 유통 기반 인프라의 구축이 기대되고 있다.

발명의 상세한 설명

기술적 과제

- [5] 본 발명은 상기와 같은 종래의 문제점을 해결하기 위한 것으로서, 본 발명의 목적은 일정 규모의 전자문서 유통시스템을 보유할 수 있는 기업/기관 등과 함께 일반 개인, 소기업들도 신뢰성을 확보할 수 있는 전자문서 유통 시스템 및 전자문서 유통 방법을 제공하는 것이다.

과제 해결 수단

- [6] 상기와 같은 목적을 가지는 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템은, 전자문서를 유통하는 시스템에 있어서, 전자주소를 기반으로 메시지를 송수신하고 메시지 송수신에 대한 유통증명서를 발급 및 관리하는 유통 메시징서버를 통해 전자문서를 유통하는 송수신 개체; 상기 송수신 개체의 전자주소를 등록/관리하며, 상기 송수신 개체 간의 전자문서 유통 경로를 설정하고, 상기 송수신 개체에게 전자문서의 표준서식을 제공하며, 송수신 개체 간의 전자문서 유통 과정에서 오류가 발생하였을 시에 메시지 전송을 대행하고 유통증명서를 발급하는 유통 허브; 및 유통증명서를 전달받아 보관하며, 신뢰할 수 있는 제3자 보관기관; 을 포함한다.
- [7] 상기와 같은 목적을 가지는 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템은, 송수신 개체와 유통허브를 포함하는 전자문서 유통 시스템에서 전자문서를 유통하는 방법에 있어서, (a) 송신 개체는 수신 개체의 주소 정보에 대응되는 물리적 주소 정보를 유통허브를 통해 획득한 후에, 전자문서를 첨부한 메시지를 상기 물리적 주소로 전송하는 단계; (b) 메시지를 수신한 수신 개체는

수신 메시지 및 송신 개체에 대한 적합성 검증 결과에 따라 수신증명서 또는 오류증명서를 발급하여 송신 개체에게 전달하는 단계; 및 (c) 수신 개체에게 메시지를 전송하였으나 실패한 송신 개체는 유통허브에 메시지 전송 대행을 의뢰하며, 메시지 전송 대행 의뢰를 받은 유통허브는 송신증명서를 발급하여 송신 개체에게 전달하고 수신 개체에게 메시지를 전송한 후에 상기 (b) 단계를 수행하는 단계; 를 포함한다.

- [8] 상기와 같은 구성 및 방법을 가지는 본 발명은 기업/기관 등과 함께 개인, 소기업들도 신뢰성을 확보받을 수 있는 전자문서 유통 체계를 구축하는 것이 가능한 효과가 있다.

도면의 간단한 설명

- [9] 도 1은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템의 구성 예를 도시한 도면.
- [10] 도 2는 도 1의 유통 메시징서버에 대하여 설명하기 위한 도면.
- [11] 도 3은 도 1의 유통 클라이언트에 대하여 설명하기 위한 도면.
- [12] 도 4는 도 1의 주소 디렉터리 서버에 대하여 설명하기 위한 도면.
- [13] 도 5는 도 1의 전자문서 서식등록기에 대하여 설명하기 위한 도면.
- [14] 도 6은 도 1의 유통 중계서버에 대하여 설명하기 위한 도면.
- [15] 도 7은 도 1의 외부 연계 게이트웨이에 대하여 설명하기 위한 도면.
- [16] 도 8은 도 1의 전자문서 유통 시스템 내에서 공인전자주소가 가지는 효력범위를 설명하기 위한 도면.
- [17] 도 9는 본 발명의 바람직한 실시예에서 공인전자주소의 신청/발급 및 운영 체계를 설명하기 위한 도면.
- [18] 도 10 내지 도 13은 본 발명의 바람직한 실시예에서 전자문서를 유통할 시에 메시지 보안에 대하여 설명하기 위한 도면.
- [19] 도 14 내지 도 17은 본 발명의 바람직한 실시예에서 메시지 송수신 프로세서에 대하여 설명하기 위한 도면.
- [20] 도 18은 본 발명의 바람직한 실시예에서 물리적 주소 획득 프로세스를 설명하기 위한 도면.
- [21] 도 19 내지 도 21은 본 발명의 바람직한 실시예에서 유통 중계 지원 프로세스를 설명하기 위한 도면.
- [22] 도 22 내지 도 24는 본 발명의 바람직한 실시예에서 공인전자주소 등록 등 관리 프로세스를 설명하기 위한 도면.
- [23] 도 25와 도 26은 본 발명의 바람직한 실시예에서 전자문서 서식 적용 프로세스를 설명하기 위한 도면.
- [24] 도 27 내지 도 29는 본 발명의 바람직한 실시예에서 스램 메시지 처리 프로세스를 설명하기 위한 도면.
- [25] 도 30과 도 31은 본 발명의 바람직한 실시예에서 전자문서 열람서비스

개념도를 도시한 도면.

- [26] 도 32는 본 발명의 바람직한 실시예에서 외부 연계 게이트웨이 서버 개념도를 도시한 도면.
- [27] 도 33은 본 발명의 바람직한 실시예에서 외부 시스템과 연계하여 전자문서가 유통되는 절차를 설명하기 위한 도면.
- [28] 도 34 내지 도 38은 본 발명의 바람직한 실시예에 따른 전자문서 유통체계 하에서 전자문서 유통을 위해 각 구성요소들 간에 서로 연계되어 동작하기 위한 통신 프로토콜의 구성을 설명하기 위한 도면.
- [29] 도 39 내지 도 43은 본 발명의 바람직한 실시예에 따른 전자문서 유통체계 하에서 오류 처리 방안을 설명하기 위한 도면.
- [30] 도 44 내지 도 51은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템에서 유통 메시징서버와 주소 디렉터리 서버 간의 연계 인터페이스를 설명하기 위한 도면.
- [31] 도 52 내지 도 63은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템에서 유통 메시징서버 상호 간의 연계 인터페이스를 설명하기 위한 도면.
- [32] 도 64 내지 도 78은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템에서 유통 클라이언트와 유통 메시징서버 간의 연계 인터페이스를 설명하기 위한 도면.
- [33] 도 79 내지 도 71은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템에서 유통 메시징서버와 유통 중계서버 간의 연계 인터페이스를 설명하기 위한 도면.
- [34] 도 82는 본 발명의 다른 실시예에 있어서 유통 메시징서버 시스템이 인증을 받아 공인전자주소로서 등록을 받기 위한 절차를 도시한 도면.
- [35] 도 83은 본 발명의 다른 실시예에 있어서 송수신 개체가 수신자로서의 역할을 할 시에 스팸메시지를 전자문서 유통허브에 신고하는 절차를 도시한 도면.
- [36] 도 84는 본 발명의 다른 실시예에 있어서 송수신 개체가 수신자로서의 역할을 할 시에 송신 상대방에 대하여 수신거부를 할 것인지 여부를 실시간으로 확인하는 경우의 절차를 도시한 도면.
- [37] 도 85는 본 발명의 다른 실시예에 있어서 송수신 개체가 수신자로서의 역할을 할 시에 송신 상대방에 대하여 수신거부를 할 것인지 여부를 주기적으로 확인하는 경우의 절차를 도시한 도면.
- [38] 도 86 내지 도 100은 본 발명의 다른 실시예에 따른 유통 메시징서버 시스템에 대하여 설명하기 위한 도면.
- [39] 도 101 내지 도 105는 본 발명의 다른 실시예에 따른 전자문서 유통 시스템 및 방법에 적용되는 유통 프로토콜에 대하여 설명하기 위한 도면.
- [40] 도 106과 도 107은 본 발명의 다른 실시예에 따른 전자문서 서식 등록기에 대하여 설명하기 위한 도면.
- [41] 도 108은 본 발명의 다른 실시예에 따른 전자문서 유통 시스템 및 방법에

적용되는 전자문서 패키징에 대하여 설명하기 위한 도면.

- [42] 도 109 내지 도 114는 본 발명의 다른 실시예에 따른 유통 클라이언트 어플리케이션에 대하여 설명하기 위한 도면.
- [43] 도 115는 본 발명의 또다른 실시예에 있어서 주소 디렉터리 서버가 전자주소를 발급하는 체계를 도시한 도면.
- [44] 도 116은 본 발명의 또다른 실시예에 있어서 주소 디렉터리 서버가 전자주소를 발급하는 프로세스를 도시한 도면.
- [45] 도 117과 도 118은 본 발명의 또다른 실시예에 있어서 주소 디렉터리 서버를 통해 주소 정보를 검색하는 프로세스의 예를 도시한 도면.
- [46] 도 119(a)는 표 37의 우편 유통체계를 도시한 도면이고, 도 119(b)는 표 37의 e-메일 유통체계를 도시한 도면이며, 도 119(c)는 표 37의 전자문서 교환(EDI) 유통체계를 도시한 도면이고, 도 119(d)는 표 37의 업무관련 시스템 유통체계를 도시한 도면.
- [47] 도 120(a)는 표 38의 자동 처리 방식을 도시한 도면이고, 도 120(b)는 표 38의 반자동 처리 방식을 도시한 도면.
- [48] 도 121(a)는 표 39의 웹 방식을 도시한 도면이고, 도 121(b)는 표 39의 어플리케이션 방식을 도시한 도면.

발명의 실시를 위한 형태

- [49] 이하, 첨부한 도면 및 표를 참조하여 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템 및 전자문서 유통 방법에 대하여 설명하면 다음과 같다.
- [50] 도 1에는 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템의 구성 예를 도시하였다.
- [51] 도 1을 참조하면 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템은, 전자주소를 기반으로 메시지를 송수신하고 메시지 송수신에 대한 유통증명서를 발급 및 관리하는 유통 메시징서버를 통해 전자문서를 유통하는 송수신 개체(101); 상기 송수신 개체(101)의 전자주소를 등록/관리하며, 상기 송수신 개체(101) 간의 전자문서 유통 경로를 설정하고, 상기 송수신 개체(101)에게 전자문서의 표준서식을 제공하며, 송수신 개체(101) 간의 전자문서 유통 과정에서 오류가 발생하였을 시에 메시지 전송을 대행하고 유통증명서를 발급하는 전자문서 유통허브(102); 및 유통증명서를 전달받아 보관하며, 신뢰할 수 있는 제3자 보관기관(공전소, 공인전자문서 보관소; 103); 을 포함하여 구성된다.
- [52] 상기 송수신 개체(101)의 유통 메시징서버는, 송수신한 메시지는 사용자별로 상태 정보를 포함하여 메시지함에 보관하고, 메시지 송수신 이력을 편집 및 삭제가 불가능한 매체에 소정 기간 동안 보관하며, 메시지 송수신에 대한 유통증명서를 발급하여 상기 제3자 보관기관에 보관을 의뢰하고, 상기 전자문서 유통허브(102)의 주소 디렉터리 서버와의 연계를 통해 상기 송수신

개체(101)에게 전자주소 등록 및 검색, 수정, 삭제를 포함하는 기능을 사용할 수 있도록 하며, 소정 기간 이상 보관된 메시지들을 외부 저장장치에 이관하여 보관한다.

[53] 상기 전자주소는, 상기 송수신 개체(101)가 상기 전자문서 유통허브(102)의 주소 디렉터리 서버를 통해 발급받은 이용자 식별기호; 상기 송수신 개체(101)가 필요한 경우에 자체적으로 부여하는 고유한 값이며, 해당 송수신 개체(101) 내에서 고유한 값인 추가 식별기호; 상기 이용자 식별기호와 추가 식별기호 사이에 위치하는 구분 기호; 를 포함한다.

[54] 상기 전자문서 유통허브(102)는 전자문서 서식 등록기를 구비하며, 상기 전자문서 서식 등록기는 전자문서 표준서식의 등록, 삭제 및 정보 수정을 포함하는 관리를 수행하며, 전자문서 표준서식을 문맥(context)에 따라 추가로 분류하고, 전자문서 표준서식이 사용될 수 있는 문맥(context)에 대한 등록, 수정을 포함하는 관리를 수행한다.

[55] 상기 전자문서 유통허브(102)는 송수신 개체(101) 간의 전자문서 유통 과정에서 오류가 발생하였을 시에 메시지 전송을 대행하고 유통증명서를 발급하는 유통 중계서버를 구비하며, 상기 유통 중계서버는 송수신 개체(101)로부터 메시지 전송을 의뢰받으면 메시지 전송을 대행한 후에 메시지 전송을 의뢰한 송수신 개체(101)에게 송신증명서를 발급하며, 의뢰받은 메시지 전송을 실패하였을 시에는 메시지 전송을 의뢰한 송수신 개체(101)에게 오류 메시지를 전송한다.

[56] 상기 전자문서 유통허브(102)는 외부 시스템과의 연계를 위한 외부 연계 게이트웨이 서버를 구비하며, 상기 외부 연계 게이트웨이 서버는 전자주소를 기반으로 메시지를 송수신하는 유통 메시징서버를 구비하며, 연계된 외부 시스템과 전자문서 유통 시스템 간의 송수신 전자주소의 검증/변환 기능과, 연계된 외부 시스템과 전자문서 유통 시스템 간의 메시지 검증/변환 기능, 연계된 외부 시스템과 전자문서 유통 시스템 간의 전자문서에 적용된 보안의 검증/변환 기능, 연계된 외부 시스템과 전자문서 유통 시스템 간의 전자문서의 적합성을 검증하고 상호간 변환하는 기능을 제공한다.

[57]

[58] 상술한 바와 같은 구성을 가지는 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템 및 이를 이용한 전자문서 유통 방법에 대하여 도 1 내지 도 118을 참조하여 상세히 설명하면 다음과 같다. 이하에서 본 발명에 대하여 상세히 설명함에 있어서 도 1의 도면 부호는 생략하도록 한다.

[59]

[60] [전자문서 유통 시스템의 구조]

[61] 전자문서 유통의 신뢰성 및 안전성을 보장하기 위해 본 발명에 따른 전자문서 유통 시스템이 유통체계가 갖추어야 하는 요건은 다음 ① 내지 ⑦과 같다.

[62] ① 유통체계에 참여하는 송수신 개체 및 송수신자는 사전에 정의된 방식으로

전자문서를 송수신하여야 하며, 관리기관 또는 전자문서 중계자의 서비스 레벨 협약(SLA)에 동의하여야 한다.

- [63] ② 유통체계 내 참여하는 송수신 개체 및 공인 송수신자는 신원확인이 가능하여야 하며, 전자문서 유통 행위는 그 사실에 대한 부인방지가 가능하여야 한다.
- [64] ③ 유통체계에서 송수신 개체 및 공인 송수신자를 식별하기 위한 정보인 공인전자주소는 법인 또는 개인 단위로 부여가 되고 등록기관에 의해 등록 및 관리되어야 한다.
- [65] ④ 유통체계 내 전자문서 유통 시, 유통증명서는 반드시 생성되어야 하며, 유통 행위에 참여한 송수신 개체 및 제3자 보관기관에 전송 및 보관되어야 한다.
- [66] ⑤ 모든 전자문서 유통행위는 P2P(Peer to Peer) 통신을 기본으로 하나, 다양한 환경적 요인으로 인해 통신에 실패한 경우 이를 지원하기 위한 체계가 있어야 한다.
- [67] ⑥ 유통체계 내에서 당사자 간의 전자문서 교환뿐만 아니라, 전자문서 열람서비스 등 다양한 부가서비스도 지원될 수 있어야 한다.
- [68] ⑦ 유통체계 이외의 외부 시스템과의 연계를 지원하는 게이트웨이가 제공되어야 한다.
- [69] 본 발명에 따른 전자문서 유통 시스템은 공인전자주소를 기반으로 하며, 유통 메시징서버를 소유하고 있는 송수신 개체들 간에 전자문서를 주고 받는 P2P 통신을 기본으로 한다.
- [70] 이와 같은 본 발명에 따른 전자문서 유통 시스템의 구조는 도 1과 같고, 시스템 내의 구성 요소에 대한 설명은 아래 표 1 및 표 2와 같으며, 주요 프로세스는 아래 표 3과 같다.
- [71] 표 1

번호	개체명	설명
1	송수신 개체	유통 메시징서버 등 전자문서 유통에 필요한 시스템 등을 구비하여 유통체계에서 사전에 약속된 방식으로 전자문서 유통에 참여하는 기업 또는 기관, 전자문서 중계자를 포함한 일반적인 개념
2	전자문서 중계자	유통 메시징서버 등 전자문서 유통에 필요한 시스템을 구비하지 못한 송수신자에게 전자문서 유통서비스를 제공하기 위해 인증을 받은 제3자 기관
3	전자문서 유통허브	송수신 개체들 간의 전자문서 유통을 지원해주는 시스템들을 통칭하는 것으로 주소관리, 경로설정, 오류 및 보안처리, 외부와의 연계 등의 작업을 수행
4	송수신자	전자문서를 유통하는 기본 단위로서 실제 전자문서를 송신수신하는 최종 사용자. 공인 송수신자를 포함한 일반적인 개념
5	공인 송수신자	전자문서 중계자에 가입하여, 공인전자주소를 발급받아 전자문서 유통 서비스를 이용하는 송수신자
6	제3자 보관기관	지식경제부 장관의 지정을 받아 타인을 위하여 전자문서를 보관 또는 증명하거나 그 밖에 전자문서와 관련된 업무를 수행하는 법인

[72] 표 2

번호	개체명	설명
1	유통 메시징서버	송수신자를 대행하여 사전에 약속된 방식으로 전자문서를 유통하는 메시징 시스템으로 송수신 개체 또는 전자문서 중계자에 설치되는 시스템
2	유통 클라이언트	송수신자가 전자문서를 유통하기 위해 사용하는 어플리케이션을 통칭하는 말로서, 메시지의 편집과 유통 메시징서버를 통하여 메시지의 송수신 등의 기능을 제공하는 어플리케이션(예: 아웃룩, 웹메일 클라이언트 등)
3	주소 디렉터리서버	공인전자주소 기반 전자문서 유통체계에 참여하는 송수신 개체 및 송수신자의 공인전자주소를 등록,관리하며 송수신을 위해 필요한 주소정보를 제공해주는 시스템
4	전자문서 서식등록기	주문서, 송장, 세금계산서 등 구조화될 수 있는 전자문서의 표준서식을 송수신 개체가 공개적으로 이용할 수 있도록 등록,관리,제공해주는 시스템
5	유통 중계서버	유통체계에서 송수신 개체 간 전자문서 유통과정에서 오류가 발생하였을 때, 송신 개체를 대신하여 메시지 전송을 대행하는 시스템
6	외부 연계 게이트웨이 서버	유통체계가 외부 시스템 등과 연계하기 위한 신뢰성있는 통로 역할을 하는 시스템
7	NTP 서버	Network Time Protocol 서버로 유통체계 내 시간을 요청하는 시스템들에게 시간을 보내주는 역할을 하는 서버
8	등록대행 시스템	등록대행기관이 공인전자주소의 신청 접수 및 등록 등의 업무를 처리하기 위한 시스템

[73] 표 3

번호	프로세스명	설명
1	메시지 송수신	송수신 개체 간에 (전자문서를 포함한) 메시지를 주고받는 행위로, 송수신 개체 내에 있는 유통 메시징서버를 통하여 메시지를 송신 및 수신하고, 유통증적 정보를 담은 유통증명서를 주고 받는 행위
2	물리적 주소 획득	전자문서를 전송하기에 앞서, 수신자의 공인전자주소에 해당하는 물리적 주소를 알기 위해 전자문서 유통허브 내에 있는 주소 디렉터리 서버에 질의하고 물리적 주소를 수신하는 행위
3	유통 중계 지원	네트워크 및 수신자 시스템 오류 등 다양한 오류에 의해 송수신개체 간 전자문서 유통이 원활하지 않을 때, 유통 중계서버가 전자문서 전송을 대행해주는 프로세스
4	유통증명서 보관 등	송신 개체가 수신한 유통증명서를 사전에 협약된 제3자 보관기관에 보관하는 행위(필요에 따라 메시지도 보관)
5	공인전자주소 등록 등 관리	송수신 개체 또는 공인 송수신자의 공인전자주소 등록, 변경 등을 하기 위한 프로세스
6	전자문서 서식 적용	유통 클라이언트에서 전자문서 서식등록기에 등록되어 있는 전자문서를 활용하는 프로세스
7	스팸 메시지 처리	특정 송수신자가 스팸 발송 등 유통체계 내에서 부적절한 행위를 하였을 때 이를 신고 및 처리하는 프로세스
8	오류처리	유통체계 내에 전자문서 유통 등 다양한 실패사태가 발생하였을 때, 이에 대한 원인 분석 및 이에 대해 대응,보완하는 행위

[74]

[75] [전자문서 유통 시스템의 구성 요소]

[76] ① 유통 메시징서버

[77] 유통 메시징서버는 송수신 개체 내에 있는 메시징 시스템으로 전자문서 유통의 핵심적인 역할을 담당한다. 유통 메시징서버는 물리적으로 하나의 전자주소(IP Address)를 가지나, 하위의 사용자를 위해 복수의 사용자 계정을 발급하고 관리할 수 있어야 한다. 각 사용자 계정을 관리하기 위해 유통 메시징서버는 사용자 계정별 메시지함을 관리하여야 하며, 유통 메시징서버는 사용자들의 계정과 메시지함을 안전하고 신뢰성 있게 관리하여야 할 책임이 있다.

[78] 유통 메시징서버의 기능 개념도는 도 2와 같으며, 기능에 대한 설명은 아래 표 4와 같다.

[79] 표 4

번호	기능명	설명
1	메시지 송신,수신	- 유통 프로토콜에 따라 메시지를 송신하고 수신
2	사용자별 메시지함 관리	- 송수신한 메시지는 사용자 계정 또는 내부 구분자에 따라 메시지함에 보관- 메시지함에 보관된 송신문서는 다음 6단계의 상태정보 관리1) 송신 중: 문서 전송 후, 수신자로부터 아무런 응답을 받지 못한 상태 2) 송신 완료: 수신자의 유통 메시징서버로부터 '수신증명서'를 받은 상태3) 송신 위탁: 송신 실패된 이후, 유통 중계서버에게 송신을 위탁한 상태 4) 송신 실패: 수신자의 유통 메시징서버 내부에서 오류가 발생하여 SOAP Fault 메시지를 리턴하거나 송수신하는 과정에서 네트워크 오류가 발생한 경우5) 열람 실패: 수신자가 메시지를 열람하는 과정에서 오류가 발생한 경우 6) 열람 완료: 수신자의 유통 메시징서버로부터 '열람증명서'를 받은 상태 - 메시지함에 보관된 수신문서는 다음 4단계의 상태정보 관리1) 검증 오류: 수신한 메시지에 대한 기본구조 검증에서 오류 발생 2) 수신 확인 전: 당해 문서 수신자가 수신문서 목록을 읽기 전3) 수신 확인: 당해 문서 수신자가 수신문서 목록을 읽음4) 열람 확인: 당해 문서 수신자가 수신문서에 대한 상세 내용을 읽음 - 수신자에 의해 삭제 요청이 오면 해당 수신문서를 물리적으로 삭제 처리해야 함- 메시지함에서 송신메시지, 수신증명서, 열람증명서 등은 서로 연관될 수 있도록 연관정보를 가져야 함
3	송수신 이력관리	- 유통 메시징서버는 송수신 이력을 편집 및 삭제가 불가능한 매체에 일정기간 보관할 수 있어야 함- 보관하여야 하는 송수신 이력정보 1) 송신이력: 메시지id, 연관메시지id, 송신자, 수신자, 송신시간, 송신문서에 대한 해쉬값2) 수신이력: 송신자, 수신자(사용자 계정 포함), 수신시간, 수신문서에 대한 해쉬값
4	메시지 보완	- 유통 프로토콜에서 제시하는 메시지 보완 기능(전자서명, 서명 검증, 암호/복호화 등)을 수행하여야 함
5	메시지 패키징 및 검증 처리	- 송신하는 문서는 전송 전에 유통 프로토콜에 정의된 메시지 구조로 패키징되어야 함- 수신한 문서는 수신 후에

		유통 프로토콜에 정의된 메시지 구조에 의해 검증,파싱되고 필요한 정보가 추출되어야 함
6	유통증명서 발급 및 관리	- 유통 메시징서버는 문서 송수신 사실에 대한 내용을 증명할 수 있도록 유통증명서를 발급하고 이를 관리해야 함 - 발급된 유통증명서를 수신하자마자 제3자 보관기관에 보관, 의뢰- 유통증명서 이력정보 : 유통증명서id, 발급시각, 관련 메시지id, 유통증명서 원본(선택적), 제3자 보관기관에 보관 후 수신한 제3자 보관기관의 등록증명서 등 - '전자문서 유통증명서' 기술규격 참조
7	주소 디렉터리 서버 연계	- 주소 디렉터리 서버가 제공하는 주소정보 등록 및 검색 프로세스에 따라 주소정보를 관리할 수 있어야 함- 주소 디렉터리 서버가 제공하는 서비스를 호출할 수 있는 클라이언트 기능이 있어야 함- 주소 디렉터리 서버가 제공하는 주소정보 등록, 검색, 수정, 삭제 기능을 원격에서 호출하는 서비스 클라이언트 기능이 제공되어야 함
8	제3자 보관기관 연계	- 유통 메시징서버는 유통증명서의 보관요청을 위해 제3자 보관기관 외부에 설치되어 있는 유통 메시징서버에 보관요청 메시지를 전송- 제3자 보관기관의 유통 메시징서버는 제3자 보관기관에 유통증명서 보관을 위한 보관요청 클라이언트를 호출- 제3자 보관기관의 유통 메시징서버가 직접 유통증명서를 생성한 경우에는 생성시점에 보관요청 클라이언트를 호출 - 보관요청 클라이언트는 제3자 보관기관의 연계인터페이스 규격에 따라 제3자 보관기관에 보관을 요청함
9	내부 시스템 연계 인터페이스	- 유통 메시징서버가 유통 클라이언트가 아니라 기업 내부 시스템일 경우, 기업 내부 시스템과 직접 연계될 수 있는 기능들을 제공하여야 함
10	유통 클라이언트 관리	- 유통 클라이언트와 관련된 사용자 계정관리, 사용자 인증, 환경정보 등을 관리
11	부가 기능 관리	- 메시지 유통과 관련된 이력 및 통계정보 등 - 시스템 관리 : 시스템 모니터링 등 - 환경정보 관리 : 유통 메시징서버 전체에 대한 환경정보 관리와 사용자별로 환경설정 기능 등을 제공하여야 함 - 문서양식(Form) 관리

12	메시지 보존 관리	- 1년 이상 보관된 메시지들을 자동으로 보존하기 위해 외부 저장장치에 이관하는 시스템 기능
----	--------------	--

- [80] 위의 표 4에 개시된 기능 이외에, 유통 메시징서버의 신뢰성을 제고하기 위해서는 서버 시스템 관리에 있어서 다음 ① 내지 ④와 같은 원칙을 준수하여야 한다.
- [81] ① 시스템 관리자는 개인의 메시지함을 보거나 임의로 삭제할 수 없어야 함
- [82] ② 서버 시스템 관리와 관련된 이력정보는 임의로 삭제할 수 없어야 하며, 불변성 매체 등에 10년 이상 보관하여야 함
- [83] ③ 시스템 관리자는 인증된 유통 메시징서버 솔루션의 기본 기능 등을 임의로 변경할 수 없음
- [84] ④ 시스템 관리와 관련된 업무 지침을 작성하여 그에 따라 관리가 이루어져야 함
- [85] ② 유통 클라이언트
- [86] 유통 클라이언트는 유통체계 내에 참여하는 송수신자들을 위해 메시지 송신 및 수신, 수신된 전자문서 열람 및 관리 등의 UI(User Interface)를 제공하는 어플리케이션이다. 유통 클라이언트는 독자적으로 문서를 송수신할 수 없으며, 반드시 유통 메시징서버와 연계되어야 한다.
- [87] 유통 클라이언트는 유통 메시징서버를 통하여 메시지 전송을 요청하거나, 유통 메시징서버를 통해 수신된 메시지를 조회한다. 유통 메시징서버는 사용자 계정별로 메시지함을 관리하게 되며, 유통 클라이언트는 수신문서 중 사용자 계정정보 확인을 통해 본인 계정에 보관되어 있는 메시지에만 접근이 가능하여야 한다.
- [88] 유통 클라이언트는 송수신 개체의 요구에 의해 C/S 형태의 어플리케이션 또는 웹 형태의 화면으로 구현될 수 있다.
- [89] 유통 클라이언트의 기능 개념도는 도 3과 같으며, 기능에 대한 설명은 아래 표 5와 같다.
- [90] 표 5

번호	기능명	설명
1	사용자 인증	-유통 클라이언트는 유통 메시징서버로부터 사용자 계정을 확인한 후 로그인 세션 정보를 받아야 함. 유통 클라이언트가 사용자 인증을 받기 위한 방법으로는 1)인증서를 기반으로 한 사용자 인증 또는 2)ID/PW를 기반으로 한 사용자 인증 등이 있음. ID/PW 기반으로 운용시, PW에 대한 보안 정책을 강제할 수 있어야 함. 1주 단위의 PW변경, 어려운 문자/숫자 조합, IP주소 변경 금지 등
2	메시지 생성	-유통 클라이언트는 신규 메시지를 작성할 수 있는 사용자 인터페이스를 제공하여야 함. 메시지를 전송하기 위해 필요한 기본정보들 중 환경정보에 의해 기 설정된 값이 아닌 항목은 입력할 수 있도록 제공. 메시지 본문은 필수 항목은 아니며, 선택적으로 본문을 추가,작성할수 있음
3	메시지 목록 조회 및 상세내용 열람	-유통 클라이언트는 사용자 계정에 해당하는 각 메시지의 목록을 조회하는 기능을 제공하여야 함. 첨부문서를 포함하여, 메시지의 상세정보를 열람할 수 있는 기능을 제공하여야 함
4	메시지 폴더 관리	-유통 클라이언트는 유통 메시징서버의 메시지함을 기반으로 송신과 수신 메시지를 유통 메시징서버가 제공하는 상태정보에 따라 사용자에게 각 메시지의 상태를 알려주어야 함. 임시보관함, 지운 메시지함을 제공하거나 사용자가 직접 메시지 폴더를 정의하고 관리할 수 있도록 하는 기능을 제공하는 것은 선택사항
5	기본정보 및 환경정보 관리	-유통 클라이언트는 메시지 송수신과 관련해서 필요한 환경정보를 관리하는 기능을 제공하여야 함. 유통 클라이언트는 유통 메시징서버에서 관리하고 있는 환경정보와 동기화되어야 함. 아폴리, 유통 메시징서버의 개별 환경정보를 설정하고 관리하는 기능 제공. 유통 클라이언트의 시스템 환경에 대한 부가정보 관리하는 어플리케이션의 개발범위에 따라 정의하여 제공
6	문서 작성	-등록되어 있는 전자문서 서식을 불러와서 전자문서를 작성하는 기능
7	메시지 송신 요청 & 수신	-유통 클라이언트는 사용자 계정 정보를 바탕으로 유통 메시징서버와의 연계 인터페이스를 통해 메시지

	메시지 Get	송신기능과 수신메시지 가져오기 기능을 수행하여야 함
8	메시지 보안 처리	- 전자서명 또는 암호화/복호화 등 메시지에 대한 보안 처리를 할 수 있어야 함
9	문서 서식 등록 및 검색	- 전자문서 서식등록기 또는 외부에 위치해 있는 전자문서 서식을 유통 클라이언트로 등록할 수 있는 기능 - 유통 클라이언트 내에 있는 전자문서 서식을 검색할 수 있는 기능
10	주소록 관리	- 자주 사용하는 공인전자주소 등을 관리하는 기능- 수신한 메시지를 통해 자동으로 공인전자주소를 등록,관리하는 기능을 선택적으로 구현
11	기업내 시스템과 연계	- 메시지 내의 전자문서를 기업 내의 그룹웨어, 업무관련 시스템에 등록시키는 등의 연계 기능
12	메시지 보존 관리	- 정책적으로 설정해 놓은 보관연한이 지난 메시지의 보존을 위해, 외부 저장장치 등에 이관하는 기능- 이 경우, 메시지와 관련된 문맥을 파악할 수 있도록 유통증명서, 로그정보 등 관련 정보까지 포괄적으로 보존 처리되어야 함
13	스팸 신고	- 스팸 등 부적절한 목적으로 수신된 메시지를 신고하는 기능
14	문서 열람 지원	- 선택적 기능으로, 송신 개체의 시스템 또는 제3자 보관기관에 전자문서를 보관하고, 이를 열람할 수 있는 권한만 전송하는 기능- 수신자는 열람 권한을 가지고 전자문서를 다운로드 받지 못하고, 오직 열람만 가능

[91] **③ 주소 디렉터리 서버**

[92] 주소 디렉터리 서버는 송·수신 개체와 공인 송·수신자에 대한 주소 정보를 관리하고, 물리적 주소를 제공하기 위한 시스템이다.

[93] 주소 디렉터리 서버는 다음 ① 및 ②와 같은 기본 기능을 제공한다.

[94] ① 수신 개체의 유통 메시징서버의 물리적 주소(IP 주소) 관리 및 제공

[95] ② 공인전자주소의 정보를 등록, 수정 등 관리 기능

[96] 아울러, 주소 디렉터리 서버는 화이트리스트 정보를 관리하는 기능을 기본적으로 가지며, 사용자로부터 스팸메시지에 대한 신고를 접수하고 이를 기준으로 블랙리스트 정보를 관리한다.

[97] 주소 디렉터리 서버는 웹 포털화면을 통해 공인전자주소와 관련된 정보를 송수신 개체 또는 공인 송수신자에게 제공하며 연계 인터페이스를 통해 유통

메시징서버 및 관련 어플리케이션들이 주소 디렉터리 서버에서 제공하는 서비스를 이용할 수 있다.

[98] 주소 디렉터리 서버의 기능 개념도는 도 4와 같으며, 기능에 대한 설명은 아래 표 6과 같다.

[99] 표 6

번 호	기능명	설명
1	공인전자주소 관리	- 송수신 개체 및 공인 송수신자의 공인전자주소의 신규 등록 및 변경 등 관리 - 공인전자주소를 소유한 송수신 개체의 정보 및 이력정보 열람 등
2	스팸신고 관리	- 유통 클라이언트로부터 수신한 스팸 등에 대한 신고 접수 및 결과 통지 기능 등
3	화이트/블랙 리스트 관리	- 공인전자주소 목록인 화이트리스트 생성 및 관리/보관-화이트리스트에 대한 검색 요청 접수 및 처리 기능-스팸 등 부적절한 목적으로 사용한 공인전자주소에 대한 블랙리스트 생성 및 수정 등 관리-블랙리스트에 대한 검색 기능
4	주소정보 검색 및 물질적 주소 통호	- 유통 메시징서버로부터 요청받은 공인전자주소의 물리적 주소 요청 접수 및 이를 리턴해주는 기능- 이와 관련된 이력정보의 검색 기능
5	웹 포털 관리	- 주소 관리와 관련된 사용자 인터페이스 제공- 주소 관리와 관련된 시스템 관리자 인터페이스 제공- 주소 디렉터리 서버의 시스템 환경정보 관리- 주소 관련 각종 통계정보 관리

④ 전자문서 서식등록기

[100]

[101] 전자문서 서식등록기는 송수신 개체 간에 사전에 약속된 표준전자문서를 이용하여 자동으로 처리될 수 있거나, e-Form 형태의 전자문서 등을 활용할 수 있도록 전자문서 유통허브에서 제공하는 시스템이다.

[102] 전자문서 서식 등록기는 전자문서 서식을 관리하는 서버 엔진과 유통 클라이언트가 이를 검색 및 다운로드할 수 있도록 기능을 제공하는 인터페이스와 웹 포털 인터페이스 등을 제공한다.

[103] 전자문서 서식등록기의 기능 개념도는 도 5와 같으며, 기능에 대한 설명은 아래 표 7과 같다.

[104] 표 7

번호	기능명	설명
1	전자문서 서식 관리	- 전자문서 서식의 등록, 삭제, 정보 수정 등 관리 - 전자문서 등록/삭제 등과 관련된 내역 통지
2	전자문서 검색 및 수신 관리	- 전자문서 서식의 검색 기능 제공 - 검색된 전자문서 서식의 다운로드 등 수신
3	전자문서 서식 연계	- 유통 클라이언트와 직접 연계된 상태에서 전자문서를 검색, 다운로드하는 기능 등을 제공
4	문맥(컨텍스트) 관리	- 해당 전자문서 서식명이 똑같다고 하더라도 국가나 산업 등의 문맥(컨텍스트)에 따라 다른 서식이 사용될 수 있기 때문에 전자문서 서식을 문맥에 따라 추가 분류 - 국가, 지역, 산업, 기업, 프로세스 등 전자문서 서식이 사용될 수 있는 문맥(컨텍스트)에 대한 등록, 수정 등 관리
5	전자문서 서식 심사&평가 기능	- 사용자가 전자문서 서식을 등록하기 위해 제출하고 대기 - 평가자에 의해 평가된 이후, 정식 등록되거나 반환하는 프로세스*전자문서 서식, 제출 방법 등에 대해서는 추가 공지
6	웹 포털 관리	- 전자문서 관리와 관련된 사용자 인터페이스 제공 - 전자문서 관리와 관련된 시스템 관리자 인터페이스 제공 - 전자문서 서식등록기 서버의 시스템 환경정보 관리 - 전자문서 서식등록기의 각종 통계정보 관리

[105] ⑤ 유통 중계서버

[106] 유통 중계서버는 전자문서 유통허브 내에 있는 시스템으로 유통체제에서 송수신 개체 간 전자문서 유통과정에서 오류가 발생하였을 때, 송신 개체를 대신하여 메시지 전송을 대행하는 시스템이다.

[107] 유통 중계서버는 내부적으로 유통 메시징서버의 기능을 내장하고 있으므로, 이를 통해 기본적인 전자문서 송수신 서비스를 제공할 뿐 아니라, 유통 중계서버만이 가지는 중계 의뢰접수 및 최종 실패시 오류 메시지 전송 등의 서비스를 연계 인터페이스를 통해 유통 메시징서버에 제공한다.

[108] 유통 중계서버의 기능 개념도는 도 6과 같으며, 기능에 대한 설명은 아래 표 8과 같다.

[109] 표 8

번호	기능명	설명
1	메시지 Routing 정보처리	송수신 개체에 있는 유통 메시징서버에 대한 경로 설정 기능
2	전송오류시 재처리 작업	메시지를 수신 개체에 전송 시 오류 발생시 재 전송 등의 처리를 하는 기능
3	송신증명서 발급	메시지 전송을 의뢰한 송신 개체에게 송신증명서를 발급하는 기능
4	전송실패 시 오류메시지 전송	의뢰받은 메시지 전송이 실패 시 송신개체에게 오류 메시지를 전송하는 기능
5	전자문서 중계 의뢰	유통 메시징서버와 연계된 상태에서 메시지 중계를 의뢰받는 기능
6	이력/통계 정보 관리	메시지 유통중계와 관련된 이력이나 통계 정보를 보관 및 처리하는 기능
7	중계현황 모니터링	송수신 개체에 대한 유통 중계 현황 제공 및 시스템 관리자에 의한 중계 현황 모니터링 기능

[110] **㉔ 외부 연계 게이트웨이 서버**

[111] 외부 연계 게이트웨이 서버는 기존에 운용되고 있거나 유통체계 내로 포함되기
힘든 외부 시스템과 유통체계가 연계하기 위한 신뢰성 있는 통로 역할을 하는
시스템이다.

[112] 공공부문의 경우, 행정정보 공동이용센터 또는 전자문서 유통지원센터 등을
통하여 민원서류 등을 전자적으로 유통하고 있는 데, 이러한 시스템들과
연계하기 위한 채널로써 외부 연계 게이트웨이 서버가 활용되어 질 수 있다.
공공부문뿐만 아니라 기타 외부 시스템들과도 연계하기 위한 채널 역할을
수행할 수 있다.

[113] 외부 연계 게이트웨이 서버의 기능 개념도는 도 7과 같으며, 기능에 대한
설명은 아래 표 9와 같다.

[114] 표 9

번호	기능명	설명
1	유통 메시지정서버 모듈	유통 메시지정 서버에 있는 기능 중 일부 사용
2	검증/변환 모듈	연계되는 외부 시스템 별로 대응되는 검증/변환하는 모듈
3	전자주소 검증/변환	유통체계와 외부 연계 시스템 간 송수신 주소의 검증 및 변환 기능
4	메시지 패키지 검증/변환	유통체계와 외부 연계 시스템 간 메시지 패키지의 검증 및 변환 기능
5	전자문서 보안 검증/변환	유통체계와 외부 연계 시스템 간 전자문서에 적용된 보안의 검증 및 변환 기능
6	문서적합성 검증/변환	유통체계와 외부 연계 시스템 간 전자문서의 적합성을 검증하고 상호 간 변환하는 기능
7	시스템 관리	외부 연계 게이트웨이 서버의 시스템 관리
8	통계정보 조회	외부 연계 게이트웨이 이용과 관련한 통계정보 조회 기능

[115] ② 공인전자주소

[116] 유통체계에 참여하는 송수신 개체와 공인 송수신자는 고유의 공인전자주소를 발급받아야 한다.

[117] 공인전자주소는 sharp#, 숫자[0-9], 하이픈[-], 마침표[.]의 조합으로 구성한다.

[118] 공인전자주소의 구성 체계는 아래 표 10과 같다.

[119] 표 10

구분	공인전자주소		내용
	구분 자	이용자 식별기호	
개인용도	#	000-0000-0000	숫자 및 하이픈으로 구성된 13자리 기호로 조합된 신청자의 임의부여 숫자
기타용도		000-00-00000	숫자 및 하이픈으로 구성된 12자리 기호로 조합된 사업자등록번호 또는 고유등록번호

[120] 공인전자주소 구성 체계와 관련된 원칙은 다음 ① 내지 ③과 같다.

[121] ① "#"의 앞부분은 이용자의 편의를 위하여 문자 [A-Z][a-z], 한글[완성된 한글 글자 2,350자], 숫자[0-9], 하이픈[-], 마침표[.]의 조합으로 구성된 이용자 추가

식별기호를 선택적으로 사용할 수 있다. 이 경우에 이용자 추가식별 기호는 전자문서 유통 메시징서버에서 자체 관리한다.

[122] ② 공인전자주소의 이용자 추가식별기호는 하이픈 또는 마침표로 시작하거나 끝나지 않아야 하며, 길이는 30바이트 이하로 한다.

[123] ③ 공인전자주소의 이용자 추가식별기호로는 사회적 관습, 미풍양속을 해치는 문자 및 숫자의 조합, 그 밖에 관리기관의 장이 정하는 제한 기호는 사용할 수 없다.

[124] 공인전자주소와 실 물리적 주소(유통 메시징서버의 실 물리적 주소)인 IP Address와는 연관관계에 대해서는 주소 디렉터리 서버를 통해서만 관리가 된다. 공인전자주소와 유통 메시징서버의 실 물리적 주소는 1:1 또는 N:1의 관계를 갖으며, 이에 따라 하나의 공인전자주소가 여러 개의 물리적 주소를 갖는 경우는 존재하지 않는다.

[125] 공인전자주소에 대한 정보(문서)의 법적 수신책임은 "#" 뒤에 존재하는 기업/기관/개인이 가져야 하며, 이용자 추가식별기호에 의한 배부는 기업/기관/개인이 편의를 위해 구분한 것이므로 송수신 개체는 이용자 추가식별기호에 의한 배포에 대해서 자체적으로 책임을 져야 한다. 이 경우 송수신 개체는 이용자 추가식별기호에 해당하는 사용자 인증에 대한 정책 및 관리 요령을 준비하여야 하며, 요령에 따라 철저히 관리하여야 한다. 아울러, 송수신 개체는 유통체계에 참여하기에 앞서 관리기관과 서비스 레벨협약(SLA)을 체결하면서 내부 구분자를 포함한 공인전자주소 내용을 포함하여 체결하여야 한다.

[126] 유통체계 내에서 공인전자주소가 가지는 효력범위는 도 8과 같이 표현될 수 있다.

[127] 이용자 추가식별기호는 송수신 개체 내에서 고유한 값이어야 한다. 이용자 추가식별기호에 대한 부여방식은 개별 송수신 개체가 책임을 지는 것을 기본으로 하며, 이용자 추가식별기호에 의한 전자문서의 배부 역시 송수신 개체가 책임져야 하며, 문제 발생시 송수신 개체가 해결하여야 한다. 이러한 이용자 추가식별기호는 유통체계의 효력범위에 포함되지 않으나, 관리기관과의 서비스 레벨 협약(SLA) 등에 의해 규정받을 수 있다.

[128] 이용자 추가식별기호는 기업의 업무 편의를 위해 전자문서를 분배하기 위한 용도로 사용되며, 주소 디렉터리 서버에 등록하지 않고 기업 내부의 정보로서만 사용한다.

[129] 상술한 바와 같은 공인전자주소의 다른 예로서 다음과 같은 구조가 가능하다.

[130] 공인전자주소 = ID + 구분기호 + 등록자

[131] 여기서, 상기 "ID"는 영문(또는 한글, 숫자), 하이픈[-] 및 마침표[.] 등이 조합되어 구성되고, 상기 "구분 기호"는 #이며, 상기 "등록자"는 영문(또는 한글, 숫자)와 마침표[.]가 조합되어 구성된다.

[132] 이러한 공인전자주소의 일 예로서 "swconvergence#mke.go.kr" 가 있으며, 이와

같은 예를 구성함에 있어서 "swconvergence"는 정부기관의 부서명, "mke"는 정부기관, "go"는 특성, "kr"는 국가를 나타내도록 하였다.

- [133] 공인전자주소의 신청/발급과 운영 체계는 도 9와 같으며, 이와 관련된 구성 요소에 대한 설명은 아래 표 11과 같다.

[134] 표 11

구성 요소	역할
관리기관(공인 전자주소 관리총괄)	- 관리기관은 공인전자주소의 최상위 관리주체로서 협의의 공인전자주소 정보를 관리- 송수신개체 및 공인 송수신자에 대한 고유 공인전자주소 ID를 발급 및 변경 관리
등록대행기관	- 관리기관의 위임을 받아 공인전자주소 신청 및 심사를 하는 기관
송수신 개체	- 협의의 공인전자주소(등록주소)의 가장 기본이 되는 단위- 기업/기관의 경우, 하나의 공인전자주소 하위에 복수 사용자를 위한 사용자 계정 또는 이용자 추가식별기호를 발급, 관리 및 유일성 보장
(공인 또는 내부)송수신자	- 전자문서유통에 참여하는 실 사용자- 공인 송수신자는 전자문서 중계자를 통해 개설한 사용자 계정을 가진 공인전자주소를 가지며, 법적 책임을 가진 송수신 단위임- 내부 송수신자는 송수신에 대한 법적 책임을 가진 송수신 개체 내에서 이용자 추가식별기호를 통해 관리되는 송수신 단위로서, 법적 책임을 지지 않으며, 전자주소 디렉터리 서버에 등록되지 않음

[135] ⑤ 전자문서 정보패키지

- [136] 전자문서 정보패키지는 메시지 내부에 포함되어 있는 전자문서에 대한 메타데이터를 명시함으로써 그룹웨어 등과 같은 기업 내부 시스템에서 해당 전자문서를 자동으로 등록하거나 처리하는 것을 지원하기 위해 필요하다.

- [137] 전자문서 정보 패키지는 메시지 유통에 반드시 필수적인 요소는 아니기 때문에 업무상 필요한 곳에서만 선택적으로 사용될 수 있다. 단, 사용 시에는 다음 ①② 및 ③과 같은 요건을 준수하여야 한다.

- [138] ① 전자문서 정보패키지는 메시지에 포함되는 전자문서와 별개의 파일 형태로 전송되어야 한다.

- [139] ② 전자문서 정보패키지는 XML 파일형태로 제공되어야 한다.

- [140] 전자문서 정보패키지의 메타데이터는 아래 표 12와 같다.

- [141] 표 12

번호	메타 정보	필수/선택	설 명
1	전자문서 명	필수	메시지 내에 포함되어 있는 전자문서에 대한 파일명
2	전자문서 식별자	필수	전자문서를 식별할 수 있는 고유 식별자
3	전자문서 유형 구분	필수	해당 전자문서가 전자문서 서식등록기에 등록되어 있는 전자문서인지, 기업/기관 내에서 사용되는 전자문서인지, 자체 전자문서 인지를 구분하는 식별자 0: 전자문서 서식 등록기에 있는 전자문서 1: 기업/기관 내에서 공통으로 사용되는 전자문서 2: 자체적으로 사용되는 전자문서
4	전자문서 유형 식별자	선택	전자문서 유형 구분이 0 또는 1인 경우에, 유형을 식별할 수 있는 식별자
5	전자문서 서명 값	선택	전자문서에 대한 전자서명 값
6	기타 정보	선택	전자문서와 관련된 기타 정보

[142]

[143] [메시지 보안]

[144] 유통체계에서 가장 중요한 부분 중의 하나가 전송 메시지에 대한 보안이다. 유통되는 메시지에 대한 보안으로는, ① 송수신 사실에 대한 부인방지, ② 전송 메시지에 대한 무결성 보장, ③ 전송 상대방에 대한 인증, 및 ④ 전송 메시지에 대한 기밀성 보장이 요구되는데, 이 가운데 ①, ②, ③은 전송 메시지에 대한 전송자의 전자서명으로 지원될 수 있으며, ④는 전송 메시지에 대한 암호화를 통해 제공되어야 한다.

[145] 유통체계의 가장 기본이 되는 유통 메시징서버 간 전자문서 유통에 있어서 적용되는 보안은 도 10과 같이 메시지 전자서명과 암호화를 지원하고 있다. 각 구간에는 전송보안을 위해 네트워크 암호화가 적용되어야 한다.

[146] 콘텐츠에 대한 전자서명은 어플리케이션 별 고유 영역으로 본 발명을 설명함에 있어서는 언급하지 않는다. 그리고, 본 발명에서는 수신자 공개키로 암호화를 하는 것이 기본이나 수신자의 인증서가 없는 경우 또는 수신자가 내부 송수신자인 경우에는 수신개체 암호화만 선택 가능하다. 또한, 메시지 전송과정에서 메시지에 대한 인증정보를 포함해서 유통 메시징서버에 전송하며, 이때 인증정보는 유통 메시징서버가 클라이언트를 인증하기 위한 용도로 주로 사용된다. 또한, 유통 메시징서버가 유통 클라이언트를 인증하기

위해 인증서 기반의 전자서명 외에 ID/PW기반, SSO(Single Sign On)에 의한 토큰정보 기반 등 다양한 인증방식을 채택할 수 있다.

[147] 이하, 전자서명 방법에 대하여 상세히 설명하면 다음과 같다.

[148] 유통 메시징서버는 다른 시스템(타 유통 메시징서버, 주소 디렉터리 서버, 유통 중계서버)과의 연계 시 반드시 자신의 공인인증서를 기반으로 전자서명이 되어야 한다. 유통체계 내 구성요소 간 연계를 위한 모든 유통 메시지는 기본적으로 전자서명이 되어야 하나, 유통 클라이언트와 유통 메시징서버 간 전자서명은 선택사항으로 인증서 기반의 사용자 인증방식인 경우에만 전자서명을 적용한다. 단, 이 경우 유통 메시징서버는 유통 클라이언트와의 유통 메시지에 대한 사용자 인증, 무결성, 송수신 사실 부인방지에 대한 모든 책임을 져야 한다.

[149] 이하, 암호화 방안에 대하여 상세히 설명하면 다음과 같다.

[150] 유통체계에서 첨부되는 문서는 보안을 위해 전송자가 암호화를 선택한 후, 전송이 가능하다. 이 부분은 문서에 대한 기밀성을 위한 부분으로 네트워크 암호화와는 구별되며, 네트워크 암호화가 적용된 경우에도 유통문서를 추가적으로 암호화를 할 수 있다.

[151] 암호화되는 구간은 도 10과 같이 ①송신자의 유통 클라이언트에서부터 수신자의 유통 메시징서버 또는 ②송신자의 유통 클라이언트에서 수신자의 유통 클라이언트까지가 될 수 있다. 수신자가 공인 송·수신자이고 공인인증서를 주소 디렉터리서버에 같이 등록해 놓은 경우에는 "② 송신자에서 수신자까지 구간"에서 암호화를 수행하고, 그렇지 않은 경우에는 "① 송신자에서 수신개체까지 구간"에서 암호화가 수행된다.

[152] 첨부되는 문서를 암호화할 때 송신자는 "① 송신자에서 수신개체까지 구간"에서 암호화가 유지되는 경우에는 송신자의 유통 메시징서버, 수신자의 유통 메시징서버, 수신자의 유통 클라이언트 등 3단계에서 모두 복호화가 가능하도록 암호화를 하여야 한다. "② 송신자에서 수신개체까지 구간"에서 암호화가 유지되어야 하는 경우에는 송신자는 송신자의 유통 메시징서버와 수신자의 유통 메시징서버가 복호화가 가능하도록 암호화를 하여야 한다.

[153] 송신개체와 수신개체의 유통 메시징서버는 첨부된 전자문서가 암호화된 경우에는 암호화된 상태로 이력관리를 위해 보관하다가, 송수신자가 복호화된 문서를 기준으로 유통증명서를 검증하고자 할 때 복호화를 해서 검증할 수 있어야 한다. 이를 위해 유통 메시징서버는 폐기된 인증서의 개인키 및 개인키 접근을 위한 비밀번호를 계속 관리하여야 한다.

[154] 이하, 암호화 방안에 있어서 암호화 개요를 설명하면 다음과 같다.

[155] 유통체계에서 유통되는 메시지가 송신자에 의해 기밀성을 보장해야 한다고 판단된 경우에는 반드시 다음과 같은 암호화 과정을 준수해야 한다.

[156] 암호문은 국내외에서 각종 표준으로 이용되는 IETF RFC 3852 "CMS(Cryptographic Message Syntax)"에서 제시하는 ContentInfo 구조체로 표현된

Enveloped-Data Content Type을 사용한다.

[157] *RFC 3852 CMS

[158] 1) IETF는 TCP/IP와 같은 인터넷 운영 프로토콜의 표준을 정의하는 주체이다. IETF는 IAB(Internet Architecture Board, 인터넷의 기술적 진화에 대한 Internet Society의 감독기구)의 감독을 받으며, IETF 구성원들은 Internet Society의 개인 또는 조직의 구성원들로부터 선발된다. IETF에서 제작된 표준들은 RFC의 형태로 나타내어지며 국내외 많은 PKI 기반 솔루션(각종 인증시스템, 타임스탬프, 제3자 보관기관 규격 등)은 이러한 RFC 표준 문서를 기반으로 만들어진다.

[159] 2) CMS(Cryptographic Message Syntax)는 최초 RSA 사에 작성한 "PKCS#7 v1.5"를 근간으로 만들어졌으며, 이를 IETF에서 규격화한 RFC 표준으로 작성한 것이 RFC2630이다. 최초 PKCS#7에는 key transfer(암호화에 이용된 대칭키를 RSA를 이용하여 상대방에게 전달) 방식만이 있었으나 RFC2630의 CMS에서는 key agreement(DH 알고리즘을 이용하여 키를 전달하는 방식) 등이 추가되었다.

[160] 3) 후에 알고리즘 부분을 별도 분리 및 다양한 키 관리 기법을 적용한 RFC3369가 2002년도에 제정되었으며, RFC3369의 내용 중 문제가 되는 부분이 많이 보고되었고 이를 최종 수정한 버전이 본 발명에서 적용한 RFC 3852이다.

[161] 추가 적용 표준으로, 암호문 생성 시 Content Encryption(실제 전송되는 전자세금계산서 패키지)에서 사용되는 알고리즘 및 알고리즘에 해당하는 파라미터 등은 IETF RFC 3370 "Cryptographic Message Syntax(CMS) Algorithm" 및 IETF RFC 4010 "Use of the SEED Encryption Algorithm in Cryptographic Message Syntax(CMS)"을 따른다.

[162] 이하, 암호화 방안에 있어서 암호화 대상 데이터에 대하여 설명하면 다음과 같다.

[163] 도 11을 참조하면, 전달되는 메시지의 암호화 대상은 다음 ① 및 ②와 같다.

[164] ① 메시지의 두 번째 MIME에 들어가는 유통정보

[165] ② 메시지본문 내용 중 본문의 실 내용이 들어가는 <Text> 영역과 첨부문서

[166] 메시지 본문에 있는 Text 및 첨부문서는 각각 독립적으로 암호화되어 해당 위치에 수록된다.

[167] 첫 번째 암호화 대상은 수신자에게 전달하고자 하는 본문 내용으로서 XML 본문 중 <Text>항목 내의 값을 대상으로 한다.

[168] 다음은 대상 데이터의 구성방법이다. 데이터는 ASN.1 Basic Encoding Rules(BER)을 따르며 Distinguished Encoding Rules(DER)를 준수하도록 한다.

[169] 표 13

MainText ::= OCTET STRING

[170] 위 표 13의 MainText는 텍스트 형태의 본문내용이다.

[171] 두 번째 이후의 암호화 대상 데이터는 세 번째 MIME부터 첨부되는

첨부문서로서 각 첨부문서 단위로 독립적으로 암호화된 후 해당 MIME에 들어가게 된다.

- [172] 다음은 대상 데이터의 구성방법으로, 데이터 구성은 첫 번째 암호화 방식과 동일하게 ASN.1 Basic Encoding Rules(BER)을 따르며 Distinguished Encoding Rules(DER)를 준수하여야 한다.

- [173] 표 14

AttachedDoc ::= URL

- [174] 위 표 14의 AttachedDoc는 binary 형태의 첨부되는 문서이다.
 [175] 이하, 암호화 방안에 있어서 암호화 처리 절차 및 구조를 설명하면 다음과 같다.
 [176] 아래 표 15는 RFC 3852의 EnvelopedData의 구성이다.
 [177] 표 15

EnvelopedData ::= SEQUENCE {version CMSversion,originatorInfo [0] IMPLICIT
 OriginatorInfo OPTIONAL,recipientInfos RecipientInfos,encryptedContentInfo
 EncryptedContentInfo,unprotectedAttrs [1] IMPLICIT UnprotectedAttributes
 OPTIONAL }

- [178] 표 15에서, version은 RFC 3852의 syntax version number 구성을 따른다.
 [179] iginatorInfo는 key management 알고리즘을 이용하지 않고 CRL을 전송할 필요가 없으므로 사용하지 않는다. (RFC 3852에 정의되어 있음)
 [180] 현재 이용할 수 있는 암호용 인증서의 알고리즘이 RSA이므로 RecipientInfos의 KeyTransRecipientInfo를 통해 수신자가 복호화할 수 있는 키(content-encryption key)를 전달한다.
 [181] encryptedContentInfo에는 내부에 정의된 Algorithm Identifier의 알고리즘을 기반으로 암호화한 MainText 또는 AttachedDoc를 넣어준다.
 [182] unprotectedAttrs는 이 버전에서는 별도로 이용하지 않으므로 송신자측에서 관리의 목적으로 값을 넣어줄 수 있으나 수신자측에서는 이를 풀어보거나 값을 이용할 필요는 없다.
 [183] 다음 ① 및 ②는 RFC 3852의 EnvelopedData의 생성에 있어 주요 부분에 대한 설명이다.
 [184] ① EncryptedContentInfo의 생성
 [185] 표 16

EncryptedContentInfo ::= SEQUENCE {contentType
 ContentType,contentEncryptionAlgorithm
 ContentEncryptionAlgorithmIdentifier,encryptedContent [0] IMPLICIT
 EncryptedContent OPTIONAL }

- [186] - ContentType은 id-data(암호화된 데이터가 어떤 정보인지를 알려주는

구분자-OID- 정보임)를 넣는다.

- [187] - contentEncryptionAlgorithm은 실제 암호화에 이용된 대칭키 알고리즘 정보를 넣는다.
- [188] - encryptedContent의 입력은 위 정의된 TaxInvoicePackage의 DER 인코딩된 값을 contentEncryptionAlgorithm에 정의된 알고리즘 방식으로 암호화한 OCTET STRING(Binary 값)이다.
- [189] ② RecepientInfo의 생성
- [190] 표 17

RecipientInfos ::= SET SIZE (1..MAX) OF RecipientInfo
 RecipientInfo ::= CHOICE
 { ktri KeyTransRecipientInfo, kari [1] KeyAgreeRecipientInfo, kekri [2]
 KEKRecipientInfo, pwri [3] PasswordRecipientInfo, ori [4] OtherRecipientInfo }

- [191] - 복호화를 할 대상이 송신 개체, 수신 개체는 필수이며, 수신자는 공인인증서가 있는 경우에만 가능하므로 실제 RecipientInfos는 하위에 RecipientInfo를 최소 2개에서 최대 3개까지 갖게 된다.
- [192] - 암호용 인증서가 RSA를 이용하므로 ktri(상대방 RSA 공개키 등을 이용하여 데이터를 암호화한 대칭키를 보내는 방식)만을 이용하여 구성하도록 한다.
- [193] 이하, 암호화 방안에 있어서 메시지에 대한 OID 정의를 설명한다.
- [194] 메시지 구성을 위한 Object Identifier는 다음 ① 및 ②와 같다.
- [195] ① EnvelopedData Type : RFC3852 CMS에서 실제 데이터를 전달하는 포맷은 ContentInfo라는 구조체이며 내부에 있는 데이터가 어떤 데이터인지를 확인할 수 있도록 ContentType에 넣어주는 정보이다.
- [196] 표 18

- id-envelopedData- OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
 rsadsi(113549) pkcs(1) pkcs7(7) 3 }

- [197] ② EncryptedContentInfo의 ContentType : 암호화된 데이터를 넣는 구조체인 EncryptedContentInfo 구조체에서 내부에 있는 데이터가 어떤 데이터인지를 확인할 수 있도록 ContentType에 넣어주는 정보이다.
- [198] 표 19

- id-data- OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549)
 pkcs(1) pkcs7(7) 1 }

- [199] 이하, 암호화 방안에 있어서 암호화 알고리즘을 설명한다.
- [200] 암호화에서 이용되는 알고리즘은 크게 두 가지로 구분된다.
- [201] ① 대상 데이터를 직접 암호화하는데 이용하는 대칭키 알고리즘
- [202] ② 대칭키를 수신자만 복호화할 수 있게 전달하는 공개키 알고리즘
- [203] 공개키 알고리즘은 이용되는 인증서가 GPKI 또는 NPKI 체계의 암호용

인증서이므로 RSA 기반 알고리즘을 이용하게 되며, 대칭키 알고리즘에 대해서는 반드시 아래에 속한 대칭키 암호알고리즘 세 가지(SEED, ARIA, 3DES) 중 하나를 택해서 사용해야 한다.

[204] 송신자측은 대칭키 암호알고리즘 세 가지 중 한 가지만 지원해도 관계없으나, 수신자측은 세 가지 알고리즘에 대해 모두 지원이 가능하여야 한다.

[205] ① 비대칭키 알고리즘(RSA): 랜덤하게 생성되어 데이터를 암호화한 대칭키 정보를 상대방에게 안전하게 암호화하여 전달하는 데 이용되며, 예제는 아래 표 20과 같다.

[206] 표 20

RSA Encryption- rsaEncryption- OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 1 }

[207] ② 대칭키 알고리즘(SEED, ARIA, 3DES): 랜덤하게 생성되어 실제 전달 데이터를 암호화하는데 이용되며, 예제는 아래 표 21과 같다.

[208] 표 21

Triple-DES CBC - des-ede3-cbc- OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) encryptionAlgorithm(3) 7 }- Algorithm의 파라미터는 반드시 존재해야하며 parameters는 반드시 CBCParameter를 가지고 있어야 한다.또는 SEED CBC- id-seedCBC - OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kisa(200004) algorithm(1) seedCBC(4) - Algorithm의 파라미터는 반드시 존재해야하며 parameters는 반드시 SeedCBCParameter를 가지고 있어야 한다.또는 ARIA CBC- id-aria128-cbc OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) gcm(100001) gpki-alg(1) aria128-cbc(20) } - id-aria192-cbc OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) gcm(100001) gpki-alg(1) aria192-cbc(21) } - id-aria256-cbc OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) gcm(100001) gpki-alg(1) aria256-cbc(22) } Algorithm의 파라미터는 반드시 존재하여야 하며 parameters는 반드시 ARIACBCParameter를 가지고 있어야 한다. ARIACBCParameter ::= ARIAIV -- Initialization Vector ARIAIV ::= OCTET STRING(SIZE(16))

[209] 이하, 암호와 방안에 있어서 암호화용 인증서의 획득과 검증에 대하여 설명한다.

[210] 암호문 생성을 위해서는 실제 데이터를 복호화하려는 수신자측의 인증서를 획득하여야만 가능하다. 인증서의 획득을 위해 송신자는 주소 디렉터리 서버에 연계하여 수신자(또는 수신개체)에 대한 인증서를 획득하여야 한다. 이때 획득한 인증서가 수신개체의 인증서인지, 공인 수신자의 인증서인지에 따라 기밀성이 유지되는 구간이 달라지게 된다.

[211] 송신자는 전송메시지에 대해 암호화를 선택한 경우, 획득한 수신자(또는

수신개체)인증서를 바탕으로 암호화를 수행한 후, 수신자에게 메시지를 전송한다. 메시지 전송과정에서 오류가 발생하여 유통 중계허브에 메시지 전송을 의뢰하는 경우에도 암호화된 내용은 변경 없이 그대로 전달한다.

- [212] 이하, 암호화 방안에 있어서 EnvelopedData의 구성에 대하여 설명한다.
- [213] 도 12에는 실제 수신자에게 전달되는 암호화된 본문을 나타내고 있으며, 실제 값들의 연계 관계에 대해 보다 정확하게 파악할 수 있을 것이다.
- [214] - ContentInfo : RFC 3852에 표현된 것으로 RFC 3852의 구성 데이터인 SignedData, EnvelopedData, EncryptedData 등을 넣어주는 일종의 컨테이너이다. 구조체의 contentType은 content가 어떤 정보인지를 가리킨다. 본 지침에서는 id-envelopedData 라는 구분자(Object Identifier)를 넣어주어야 한다.
- [215] - EnvelopedData : 암호화 정보를 전달하기 위한 구조체(앞 절의 설명 참조)
- [216] - EncryptedContentInfo : 암호화된 정보를 보관하는 구조체이다. 구조체의 contentType은 encryptedContent가 어떤 정보인지를 가리킨다. 본 지침에서는 id-data 라는 구분자(Object Identifier)를 넣어주어야 한다.
contentEncryptionAlgorithm은 SEED, ARIA, 3DES 중의 하나에 대한 구분자(OID)를 넣어주어야 하며 encryptedContent에 랜덤하게 생성된 비밀키를 이용하여 해당 알고리즘으로 암호화한 데이터를 OCTET STRING(바이너리 데이터)으로 넣어준다.
- [217] - RecipientInfo : 어떤 방법을 이용하여 수신자가 복호화할 지를 선택하는 구조체이다. 본 지침에서는 KeyTransRecipientInfo를 이용해야 한다.
- [218] - KeyTransRecipientInfo : 수신자가 복호화할 수 있도록 위에서 서술한 encryptedContent를 암호화한 랜덤한 비밀키를 수신자의 공개키를 이용하여 암호화하여 전달하는 데 이용하는 구조체이다. 암호화한 비밀키는 encryptedKey에 넣어 주며, 누구의 공개키를 이용하였는지에 대한 정보인 RecipientIdentifier 및 비밀키를 암호화하는데 이용한 알고리즘 정보인 KeyEncryptionAlgorithmIdentifier 등을 포함한다.
- [219] 이하, 복호화 방안에 대하여 설명한다.
- [220] 암호화된 전자문서를 수신한 수신자는 도 12를 통해 설명하는 절차를 따라 암호화된 본문과 첨부문서를 복호화한다. 수신자는 암호화된 본문과 첨부문서를 각각 복호화함으로써, 평문형태의 본문과 첨부문서를 얻을 수 있다.
- [221] - 1단계: 암호문에서 EnvelopedData 구조를 추출하고 이를 읽어 들인다.
- [222] - 2단계: EnvelopedData 구조체에서 복호용 정보 구조체(RecipientInfo)를 추출한뒤 추출한 복호용 정보 구조체로부터 암호화된 대칭키 정보(KeyTransRecipientInfo)를 얻는다.
- [223] - 3단계: 수신자는 암호화된 대칭키 정보로부터 추출한 encryptedKey를 수신자의 개인키(인증서의 공개키와 맵핑되는)를 통해 복호화함으로써 본문 및 첨부문서를 암호화 할 때 사용한 대칭키를 얻는다.
- [224] - 4단계: 1단계에서 추출한 EnvelopedData 구조체에서 본문 또는 첨부문서의

암호화된 구조체를 얻는다.

[225] - 5단계: 3단계를 통해 획득한 대칭키를 이용하여 4단계에서 추출한 암호화된 본문 또는 첨부문서를 복호화 한다.

[226] - 6단계: 최종적으로 복호화된 본문과 첨부문서를 획득한다.

[227]

[228] [네트워크 보안]

[229] 송신자와 수신자간 유통되는 메시지의 기밀성을 위해 전자문서 유통의 전 구간(메시징송신자의 유통 클라이언트와 유통 메시징서버 간, 송신자와 수신자의 유통 메시징서버 간, 수신자의 유통 메시징서버와 유통 클라이언트 간)에서 네트워크 보안을 위해 SSL(Secure Sockets Layer)을 적용한다.

[230]

[231] [메시지 송수신 프로세스]

[232] 유통체계 내 관계자들 간, 시스템들 간에는 다양한 업무 프로세스가 존재한다. 유통체계 내 가장 기본적인 메시지 송수신 프로세스가 있으며, 이를 지원하기 위한 여러 프로세스가 존재한다.

[233] 메시지 송수신 프로세스는 송신자와 수신자 간에 메시지를 직접 주고 받는 형태로 우편이나 e-메일과 같이 상대방에게 문서가 교환되는 프로세스이다.

[234] 송신자와 수신자간 메시지를 송수신하기 위해, ①수신자에 대한 물리적 주소 및 보안정보 획득, ②메시지 전송 및 전송확인, ③업무수신자의 수신확인, ④유통증명서 발급 및 보관의 4단계 프로세스로 이루어진다. 이때, 유통증명서에 대한 프로세스는 증빙을 받고자 하는 내용에 따라 상세 프로세스의 흐름이 달라질 수 있는데, 이에 대한 기본적인 흐름을 도 14에서 제시하고 있으며, 상세한 설명은 아래 표 22와 같다.

[235] 표 22

번호	프로세스 명	설 명
1	수신자에 대한 물리적 주소 및 보안정보 획득	◆ 송신 개체는 상대방에 대한 주소정보를 바탕으로 실제 메시지가 전달되어야 할 물리적 주소정보 및 보안정보(송신메시지에 대한 수신암호를 필요로 하는 경우)를 주소 디렉터리 서버에 요청함으로써 이를 획득◆ 이 과정에서 주소 디렉터리 서버는 요청한 수신 개체의 공인전자주소가 블랙리스트 또는 화이트리스트에 있는지를 확인(블랙리스트에 있을 경우, 메시지 전송 오류를 송신 개체에 통보)◆ 송신 개체의 유통 메시징서버가 주소 디렉터리 서버에 수신자에 대한 물리적 주소 및 보안정보를 요청한 후, 이를 수신
2	메시지 송수신 및 전송 확인	◆ 송신 개체는 메시지를 기술 규격에 맞게 패키징 한 후, 유통 메시징서버의 공인인증서를 기반으로 전자서명을 수행◆ 유통 메시징서버는 앞서 획득한 물리적 주소에 패키징하고 전자서명된 메시지를 전송함◆ 메시지를 수신한 수신 개체의 유통 메시징서버는 메시지의 기본 패키징 구조, 전자서명에 대한 유효성, 송신자에 대한 적합성을 검증한 후, 수신확인을 위한 수신증명서 또는 오류메시지를 생성함◆ 수신 개체의 유통 메시징서버는 생성한 수신증명서를 송신 개체에게 전송함◆ 송신 개체의 유통 메시징서버는 수신증명서를 수신하여, 1)수신증명서의 적합성을 검증, 2)검증된 정보를 수신증명서에 첨부, 3)수신증명서를 자체 보관 및 제3자 보관기관에 보관 요청 ◆ 전송과 전송확인 과정은 동기식 메시지 처리로 이루어짐
3	업무수신자의 수신확인	◆ 송신자가 메시지 전송시점에 수신 개체의 열람증명서를 요청한 경우, 수신 개체는 메시지 열람 시점에 송신 개체에게 열람확인을 증빙할 수 있는 열람증명서를 생성하여 이를 전송◆ 수신 개체가 송신 개체에게 열람증명서를 전송하면, 이를 수신한 송신 개체는 1)열람증명서의 적합성을 검증하고, 2)검증된 정보를 열람증명서에 첨부, 3)열람증명서를 자체 보관 및 제3자 보관기관에 보관 요청, 4)수신 개체에게 수신확인 메시지를 동기식으로 전송
4	유통증명서	◆ 각 단계별로 유통에 대한 증빙을 받고자 하는 경우 송신

	발급 및 보관	개체는 각 단계에 따라 수신, 열람, 송신에 대한 증명서를 수신 개체 또는 유통 중계서버로부터 발급받아 이를 제3자 보관기관에 보관함으로써 유통에 대한 법적증빙의 근거를 확보
--	---------	---

- [236] 메시지 송수신 프로세스는 전송 프로세스와 수신 프로세스로 구분할 수 있으며, 전송 프로세스는 유통 클라이언트와 유통 메시징서버 간의 연계 방식에 의해 동기식 전송과 비동기식 전송으로 세분화된다.
- [237] 이하, 도 15 및 표 23을 참조하여 동기식 메시지 송수신프로세스에 대하여 설명하면 다음과 같다.
- [238] 동기식 메시지 송수신 프로세스는 송신 개체의 유통 클라이언트가 유통 메시징서버에 전송을 요청하면 이를 수신 개체의 유통 메시징서버에 실시간 전송을 하고 이에 대한 응답을 송신자의 유통 메시징서버를 통해 동기식으로 돌려받는 프로세스이다. 동기식 프로세스는 전송에 대한 결과를 즉각적으로 유통 클라이언트가 확인할 수 있으므로 업무 프로세스에 대한 정의가 단순해질 수 있다.
- [239] 이러한 동기식 메시지 송수신 프로세스에 대한 상세한 설명은 아래 표 23과 같다.
- [240] 표 23

번호	프로세스명	설명
1	수신자에 대한 물리적 주소 및 보안정보 획득	◆ 상술한 메시지 송수신 프로세스와 동일하며, 세부 프로세스는 후술하는 물리적 주소 획득 프로세스 참조
2	메시지 전송 및 전송 확인	◆ 송신 개체의 유통 클라이언트는 문서를 작성하여 첨부한 후 전송대상 및 방식(압축화 여부, 열람증명서 수신 여부 등)을 설정하여 송신 개체의 유통 메시징서버에 전송 요청을 하고 이에 대한 응답을 받기 위해 통신을 끊지 않고 기다림◆ 송신 개체의 유통 메시징서버는 이를 받아서 바로 수신 개체의 유통메시징서버에 전송함◆ 송신 개체의 유통 메시징서버는 수신 개체로부터 전송에 대한 응답메시지를 수신함(이에 대한 절차는 상술한 메시지 송수신 프로세스의 메시지 전송 및 전송확인 프로세스를 참조)◆ 송신 개체의 유통 메시징서버는 수신한 응답메시지를 유통 클라이언트와 연결된 통신구간을 통해 동기식으로 돌려줌
3	업무수신자의 수신확인	◆ 상술한 메시지 송수신 프로세스와 동일
4	유통증명서 발급 및 보관	◆ 상술한 메시지 송수신 프로세스와 동일

[241] 이하, 도 16 및 표 24를 참조하여 비동기식 메시지 송수신프로세스에 대하여 설명하면 다음과 같다.

[242] 비동기식 전송 프로세스는 송신 개체의 유통 클라이언트가 유통 메시징서버에 전송을 요청하면 유통 메시징서버에 전송요청에 유효성 여부만을 검증한 후 요청에 대한 수신확인 메시지를 유통 클라이언트에 돌려준다. 이를 수신 개체의 유통 메시징서버에 실시간 전송을 하고 이에 대한 응답을 송신 개체의 유통 메시징서버를 통해 동기식으로 돌려받는 프로세스이다.

[243] 비동기식 프로세스는 전송하고자 하는 메시지의 용량이 크거나 하나의 메시지에 대해 복수의 수신자를 지정한 경우처럼 메시지 전송에 대한 시간이 많이 걸려서 클라이언트가 응답을 기다리기 힘든 상황에서 사용한다.

[244] 이러한 동기식 메시지 송수신 프로세스에 대한 상세한 설명은 아래 표 24와 같다.

[245] 표 24

번호	프로세스명	설명
1	수신자에 대한 물리적 주소 및 보안정보 획득	◆ 상술한 메시지 송수신 프로세스와 동일하며, 세부 프로세스는 후술하는 물리적 주소 획득 프로세스 참조
2	메시지 전송 요청	◆ 송신 개체의 유통 클라이언트는 문서를 작성하여 첨부한 후 전송대상 및 방식(암호화 여부, 열람증명서 수신 여부 등)을 설정하여 송신자의 유통 메시징서버에 전송 요청을 함 ◆ 송신 개체의 유통 메시징서버는 이를 받아서 전송요청에 대한 유효성을 검증한 후, 전송요청에 대한 수신확인을 동기식으로 유통 클라이언트에 돌려주고 통신을 종료함
3	수신자에게 메시지 전송	◆ 송신 개체의 유통 메시징서버는 유통 클라이언트로부터 받은 전송요청 메시지를 찾아, 수신자에게 전송함 ◆ 송신 개체의 유통 메시징서버는 수신자로부터 동기식으로 전송에 대한 응답메시지(수신증명서 또는 오류 메시지)를 받은 후, 이를 최초에 전송요청했던 유통 클라이언트의 수신함에 넣어 둠
4	전송결과 수신	◆ 유통 클라이언트는 유통 메시징서버에 접속하여 자신에게 수신된 메시지를 가져오기 위해 수신 메시지 Get 요청을 함 ◆ 수신 개체로부터 전송된 응답메시지가 있으면 유통 클라이언트는 이를 응답메시지로 수신함
5	업무수신자의 수신확인	◆ 상술한 메시지 송수신 프로세스와 동일
6	유통증명서 발급 및 보관	◆ 상술한 메시지 송수신 프로세스와 동일

[246] 이하, 메시지 수신 프로세스에 대하여 상세히 설명하면 다음과 같다.

[247] 문서 수신자가 유통 클라이언트를 통해 메시지를 수신하는 프로세스는 도 17과 같으며, 프로세스에 대한 설명은 아래 표 25와 같다. 수신자의 유통 메시징서버가 송신자로부터 메시지를 수신하면 이에 대한 응답으로 수신증명서를 전송하고 최종 수신자의 메시지함에 문서를 넣어 둔다.

[248] 유통 클라이언트는 수시로 유통 메시징서버에 수신된 메시지에 대한 목록을 요청하고, 새로 수신된 메시지가 있으면 수신메시지 목록을 응답 메시지로 받게

되며, 이 중 상세정보를 요청하는 메시지를 통해 수신문서를 Get한다.

[249] 표 25

번호	프로세스명	설명
1	메시지 수신	◆ 수신 개체는 메시지를 수신하면, 수신한 메시지에 대한 수신 응답메시지를 송신 개체에게 돌려주고, 수신 메시지를 해당 사용자의 사서함에 보관
2	수신메시지 목록 Get	◆ 수신 개체의 유통 클라이언트는 연계된 유통 메시징서버 시스템에 인증을 거친 후 수신문서를 요청◆ 수신 개체의 유통 메시징서버는 수신문서를 요청한 사용자의 사서함에 보관된 수신 문서목록을 동기식 응답으로 유통 클라이언트에 전달
3	수신문서 Get	◆ 수신자가 수신메시지의 목록에서 메시지에 대한 상세정보 보기를 요청하면, 유통 클라이언트는 유통 메시징서버 시스템에 해당 메시지의 첨부문서를 포함한 상세정보 전달을 요청◆ 수신 개체의 유통 메시징서버는 사서함에 보관된 수신 문서의 상세정보 및 첨부문서 원본을 유통 클라이언트에 동기식 응답으로 전달
4	열람증명서 전송(선택적 프로세스)	◆ 최초 송신자가 수신 담당자의 열람확인을 요청한 경우, 수신자의 유통 메시징서버 시스템은 사용자가 수신문서에 대한 상세정보요청을 한 시점에 해당 메시지의 송신자에게 열람증명서를 포함한 메시지를 전송◆ 단 첨부문서에 암호화가 된 경우에는 상세정보를 가져간 시점이 아니라, 사용자가 첨부문서를 복호화한 후 열람상태를 유통메시징서버에 전달한 시점에 열람증명서를 생성하여 전송자의 유통 메시징서버로 전달하여야 함. 이때 복호화과정에서 오류가 발생하면 오류에 대한 상태정보를 전달하고 수신 유통 메시징서버는 열람증명서 대신에 복호화 오류메시지를 전송자에게 전달함◆ 수신 개체의 유통 메시징서버는 송신 개체의 유통 메시징서버로부터 전송한 열람증명서 메시지(또는 오류메시지)에 대한 수신 응답메시지를 수신

[250] [물리적 주소 획득 프로세스]

[251] 유통체계에 참여하는 송신 개체는 상대방에게 메시지를 전송하기 전에 공인전자주소 정보를 바탕으로 물리적인 실 주소 정보를 반드시 알아야 하며, 부가적으로 첨부하는 문서를 암호화하기 위해서는 주소 디렉터리 서버에 있는

수신자의 공개키 정보를 획득하여야 한다.

[252] 공인전자주소의 물리적 주소를 획득하는 절차는 필수 단계로서 아래 1) 내지 4)가 있다.

[253] 1) 송신 개체는 수신 개체의 주소정보를 기준으로 수신 개체에 대한 물리적 주소정보 및 보안정보 획득을 위해 주소 디렉터리 서버에 질의

[254] 2) 주소 디렉터리 서버는 송신 개체의 질의를 수신/검증한 이후, 이를 처리

[255] 3) 송신 개체는 수신받은 물리적 주소를 기준으로 경로설정을 하여 수신 개체에 메시지 전송

[256] 4) 수신 개체의 유통 메시지정서버는 이를 받아서 사용자 계정 또는 내부 구분자에 따라 메시지를 내부적으로 분배

[257] 또한, 유통체제에서 공인전자주소의 물리적 주소를 획득하는 방식은 2가지로 구분할 수 있으며, 아래 ① 및 ②와 같다.

[258] ① 유통 클라이언트가 수신자의 주소정보를 입력하는 시점에 주소 디렉터리 서버에 검색 요청을 하여 물리적 주소와 수신자 공개키를 가져오는 방식:

1)공인전자주소의 유효성을 사전에 체크하기 위함, 2)유통 클라이언트와 유통 메시지정서버(송신 개체) 간 메시지 암호화가 필요할 때

[259] ② 유통 클라이언트가 유통 메시지정서버에게 메시지 송신을 요청한 이후, 유통 메시지정서버가 주소 디렉터리 서버로부터 물리적 주소를 가져오는 방식

공인전자주소의 물리적 주소 및 보안정보 획득의 프로세스는 도 19와 같다.

[260]

[261]

[유통 중계 지원 프로세스]

[262]

[263]

유통체제는 송신개체와 수신개체 간 직접 통신(P2P)을 기본으로 한다. 그러나 부가적으로 네트워크, 수신 개체의 유통 메시지정서버의 오류 등에 의해 메시지 유통이 장애가 발생하였을 경우, 사용자의 편의 및 유통의 원활한 지원을 위해 유통을 중계/대행하는 중계 프로세스를 제공한다.

[264]

송신 개체가 메시지를 수신 개체에게 전송하는 과정에서 오류가 발생하여 전송이 실패하는 경우에 전자문서 유통허브 내 유통 중계서버는 송신 개체를 대행해서 메시지를 위탁/전송함으로써 송신 개체의 전송 사실을 입증하고 송신 개체의 시스템적인 부담을 덜어주는 역할을 수행한다.

[265]

도 23은 이에 대한 기본적인 흐름을 제시하고 있으며, 도 24는 유통 중계서버가 메시지를 중계하는 프로세스를 보여주고 있다.

[266]

아래 표 26은 유통 중계 프로세스를 단계적으로 설명하고 있다.

[267]

표 26

번호	프로세스명	설명
1	메시지 송신 대행 요청	◆ 송신개체는 메시지 패키징 및 보안처리 후 수신개체에게 메시지 전송 ◆ 전송과정에서 오류가 발생하여 전송이 최종적으로 실패되었을 때 송신개체는 유통 중계서버에 메시지를 대행해서 전송해 줄 것을 의뢰 ◆ 전송의뢰를 접수한 유통 중계서버는 송신개체에게 송신증명서를 동기식으로 발급, 전송
2	메시지 위탁 중계	◆ 유통 중계서버는 중계의뢰를 받은 메시지를 전송. 전송 실패 시 일정 간격으로 재 시도를 함(재시도 횟수 및 간격은 유통 중계서버의 정책에 따라 결정) ◆ 유통 중계서버가 전송에 최종적으로 실패할 경우, 메시지 중계를 의뢰한 송신개체에 전송실패 메시지를 전달
3	수신 및 열람 증명서 발급	◆ 수신개체가 메시지를 정상적으로 수신한 이후, 수신증명서를 유통 중계서버에 전송 ◆ 수신자가 전자문서를 열람한 이후, 수신개체는 열람증명서를 유통 중계서버를 경유하지 않고 송신개체에 직접 전송

[268]

[269] [유통증명서 등 보관 프로세스]

[270] 유통체계 내에서 이루어지는 모든 유통행위와 관련하여 그 결과로써, 유통증명서는 반드시 생성되어 제3자 보관기관에 보관되어야 한다. 이는 유통증적을 담은 유통증명서를 법적으로 인정받은 제3자 보관기관에 보관함으로써, 유통사실에 대한 법적 추정력을 확보받을 수 있기 때문이다.

[271] 유통증명서를 보관하는 프로세스는 전자문서 유통과는 별개의 프로세스로서, 유통행위 사실을 증명하기 위한 지원 프로세스이다. 이를 위해서, 모든 유통 메시징서버는 사전에 유통증명서를 수신, 보관할 수 있는 기능을 갖춘 특정 제3자 보관기관에 가입되어 있어야 한다.

[272] 아울러, 송신 개체가 전자문서 내용증명을 원할 경우, 유통증명서 이외에 메시지 전체를 제3자 보관기관에 보관할 수도 있다.

[273] 제3자 보관기관이 유통증명서를 수신하여 보관하기 위해서는 다음 ① 및 ②와 같은 2개의 부가적인 시스템을 구비하여야 한다.

[274] ① 제3자 보관기관 사업자의 유통 메시징서버 : 유통체계 내의 유통 메시징서버와 유통증명서를 송수신 하기 위해 필요한 시스템

[275] ② 제3자 보관기관 연계 클라이언트 모듈 : 제3자 보관기관 사업자의 유통 메시징서버를 통해 수신한 유통증명서를 제3자 보관기관에 보관하기 위해 제3자 보관기관 연계인터페이스와 통신하기 위한 모듈

[276] 단, 제3자 보관기관 사업자가 전자문서 중계자를 겸할 경우, 유통 메시징서버는 추가적으로 필요하지 않을 수 있다.

[277] 도 28은 송수신 개체와 제3자 보관기관 사업자 간 유통증명서를 보관하는 프로세스를 보여주고 있으며, 유통증명서 보관 프로세스를 단계적으로 살펴보면 아래 표 27과 같다.

[278] 표 27

번호	프로세스명
1	송신자의 유통 메시징서버는 수신자의 유통 메시징서버로부터 유통증명서 수신◆수신 유통 메시징서버로부터 수신증명서를 받거나 중계허브로부터 송신증명서를 받는 경우는 응답메시지로 유통증명서를 수신◆수신 유통 메시징서버로부터 열람증명서를 받거나 중계허브를 통해 수신증명서를 받는 경우는 요청메시지로 유통증명서를 수신
2	-수신받은 유통증명서를 검증하여 유효성이 확인되면, 유통증명서와 증명서 검증정보를 사전에 설정된 제3자 보관기관의 유통 메시징서버로 전송 - 유통증명서가 유효하지 않으면, 수신자의 유통 메시징서버에게 유통증명서 검증오류 메시지를 통지하여 재전송 요청 ◆요청메시지로 유통증명서를 받은 경우는 이에 대한 응답메시지로 검증오류 메시지를 전송(동기식)◆응답메시지로 유통증명서를 받은 경우는 새로운 요청메시지로 검증오류 메시지를 전송(비동기식)◆유통증명서 검증오류 메시지를 받은 경우에는 새로운 요청 메시지로 유통증명서를 재전송하거나 증명발급 실패 메시지를 전송(비동기식)◆유효한 유통증명서를 받지 못한 경우에는 전자문서 전송에 실패한 것으로 보고 전자문서를 다시 전송하여야 함
3	제3자 보관기관의 유통 메시징서버는 제3자 보관기관 보관요청 메시지를 수신하면, 유통증명서 및 검증정보 보관을 위한 보관요청 Client를 호출(제3자 보관기관이 전자문서 중계자를 겸하고 있을 경우, 유통 메시징서버가 제3자 보관기관 보관요청 Client를 직접 호출함(로컬 보관요청))
4	유통증명서 보관요청 Client는 제3자 보관기관의 연계인터페이스 규격에 따라 유통증명서와 검증정보를 제3자 보관기관에 보관 요청

[279]

[280] [공인전자주소 등록 등 관리 프로세스]

[281] 송수신 개체들이 유통체계에 참여하기 위해서는 공인전자주소를 신청하여 등록받아야 하며, 등록대행기관 및 관리기관은 공인전자주소와 관련된 정보를 등록 등 관리하여야 한다. 공인전자주소정보 관리 프로세스에는 공인전자주소와 관련된 등록, 변경, 삭제 등에 대한 관리 프로세스와

블랙리스트/화이트리스트 관리 프로세스가 있다.

- [282] 관리기관은 공인전자주소의 효율적인 관리를 위해 공인전자주소 등록대행기관을 두어 공인전자주소를 관리한다.
- [283] 등록대행기관은 다음 ① 내지 ④와 같은 업무를 수행한다.
- [284] ① 공인전자주소 신청자의 신원확인 등 심사 업무
- [285] ② 공인전자주소 등록자의 등록정보 변경 업무
- [286] ③ 공인전자주소 등록 말소 등 업무 지원
- [287] ④ 기타 공인전자주소 관리와 관련된 업무
- [288] 관리기관은 등록대행기관으로 제3자 보관기관 및 전자문서 중계자 중에서 요건을 만족하는 자를 선정할 수 있다.
- [289] 이하, 공인전자주소 등록 프로세스에 대하여 설명하면 다음과 같다.
- [290] 유통체계에 참여하고자 하는 기업/기관/개인 등은 공인전자주소를 신청해야 하며, 등록대행기관은 신청된 정보를 심사, 처리하여 결과를 통보하여야 한다. 이와 관련된 프로세스는 도 22와 같다.
- [291] 이하, 공인전자주소 등록정보의 변경 프로세스에 대하여 설명하면 다음과 같다.
- [292] 기 등록된 공인전자주소와 관련된 정보는 여러 가지 사정으로 변경할 수 있으나, 다만 공인전자주소와 소유자 정보는 동일성을 유지하여야 하기 때문에 변경할 수 없다.
- [293] 공인전자주소와 관련된 정보 변경에 대한 권한은 등록대행기관에 위임하여 처리하도록 한다. 단, 정보 변경에 대한 이력정보를 관리기관과 등록대행기관 간 서비스 레벨협약(SLA)에 따라 보관하여야 한다.
- [294] 이와 관련된 프로세스는 도 23과 같으며, 도 23을 참조하면, 공인전자주소의 변경은 본인만이 가능하다. 개인의 경우, 공인전자주소, 등록번호, 이름은 변경불가하기 때문에 공인전자주소를 탈퇴하고 신규로 생성하여야 한다. 그리고, 기업의 경우, 공인전자주소는 변경 불가하며, 등록번호(사업자 등록번호) 및 상호명은 변경시, 반드시 해당 정보로 변경되어 받은 신규 공인인증서와 같이 변경하여야 한다.
- [295] 도 24에는 등록대행기관 변경 프로세스를 나타내고 있으며, 이와 같은 도 24를 참조하면, 등록대행기관을 변경하고자 하는 경우에는 1)기존 등록대행기관으로부터 탈퇴, 2)신규 등록대행기관을 통한 신규 등록과정을 거쳐야 한다. 이 경우, 주소 디렉터리 서버에 공인전자주소 일시 유보를 요청할 수 있어야 한다. 주소 디렉터리 서버는 모든 공인전자주소 탈퇴 시, 공인전자주소 일시 유보를 선택할 수 있도록 함으로써 등록대행기관 변경 시 일정기간 동안 공인전자주소를 유지할 수 있도록 한다.
- [296] 이하, 공인전자주소 갱신 및 사용정지, 삭제 프로세스에 대하여 설명하면 다음과 같다.
- [297] 기 등록된 공인전자주소는 설정한 사용연한에 맞춰 갱신하여야 한다.

공인전자주소 등록 이후, 서비스 정책에 기반하여 설정해 놓은 사용연한이 지나기 전에 소유자가 갱신하여야 한다. 만약 소유자가 갱신하지 않으면, 공인전자주소에 대한 소유권을 잃게 되고 공인전자주소는 자동 말소된다.

[298] 아울러, 공인전자주소의 만료기간이 되지 않더라도 신청자가 사용정지나 말소를 원할 경우, 이에 대한 기능을 제공해주어야 한다.

[299]

[300] [전자문서 서식 적용 프로세스]

[301] 이 프로세스는 메시지 유통 이후 단계의 활용도를 높이기 위한 프로세스이다. 메시지 내에 포함되어 있는 전자문서를 기업 내부 시스템에서 자동 또는 반자동화된 방식으로 처리하기 위함이다. 유통 메시징서버는 당사자간 메시지 송수신 기능을 전담하며, 유통 클라이언트는 송수신하기 위한 메시지를 송수신자가 이용하기 쉽도록 인터페이스를 제공해준다. 그 이후 단계는 메시지 내에 있는 전자문서를 활용하는 단계이다. 전자문서 서식등록기나 전자문서 서식은 전자문서 활용단계를 보다 효율적으로 운용하기 위한 방법 등을 제공한다.

[302] 유통체계를 따라 유통되는 문서의 양식에는 제한이 없다. 이미지, 오피스 문서, 동영상 등이 가능하나, 사용자의 편의성을 높이기 위해 유통체계에서는 서식형태의 문서 작성 기능을 부가적으로 제시한다.

[303] 이와 같은 부가기능은 전자문서교환(EDI) 기능을 도입한 것으로, 송수신 개체 간에 약속된 전자문서 포맷을 기반으로 문서 데이터를 송수신하여 수신 개체 내부시스템에서 수신한 전자문서를 자동으로 변환처리할 수 있도록 하였다.

[304] 전자문서의 서식은 다음 ① 및 ②와 같은 2가지 방식으로 활용 가능하다.

[305] ① 전자문서 유통허브에 있는 전자문서 서식등록기를 활용하는 방식으로 전자세금계산서와 같이 표준화된 전자문서 서식을 주로 사용

[306] ② 송수신 개체 간 합의된 전자문서 서식을 공유하여 활용하는 방식으로 특정 기업에 특화된 비표준화된 전자문서 서식을 주로 사용

[307] 이하, 전자문서 서식등록기 활용에 대하여 상세히 설명하면 다음과 같다.

[308] 기업, 기관이나 개인사용자는 전자문서 서식등록기에 등록된 전자문서 서식을 검색한 후, 유통 클라이언트에 등록해서 사용할 수 있다. 전자문서 서식등록기를 활용하는 방식은 다음 ① 및 ②와 같은 2가지 방식이 있다.

[309] ① 유통 클라이언트에서 직접 전자문서 서식등록기에 있는 전자문서를 검색하여 가져오는 방식

[310] ② 전자문서 서식등록기 사이트에서 전자문서를 검색하여 로컬 PC에 다운로드 후, 유통 클라이언트에 등록하여 사용하는 방식

[311] 전자문서 서식등록기는 표준화된 전자문서 서식을 등록/활용하기 때문에, 관리기관이 체계적으로 운영/관리하여야 하여 다음 ① 내지 ③과 같은 기준을 가질 수 있다.

[312] ① 사용자는 전자문서 서식등록기 사이트를 통해서 신청하여야 한다. 신청은

사이트에서 제공하는 포맷과 절차에 따라야 한다.

[313] ② 등록/신청한 전자문서는 관리기관의 심사를 거쳐 등록되어야 한다.

[314] ③ 각각의 전자문서는 체계적인 컨텍스트(문맥) 기반에서 분류되어야 한다.

[315] 표 28

구분	설 명	비 고
지 역	해당 전자문서가 국제적으로 유통될 수 있는지, 특정 지역에서 유통될 수 있는지, 특정 국가에서만 유통될 수 있는 지 구분	
구 문	해당 전자문서에 적용된 구문이 EDIFACT 기반인지, XML 기반인지, 사실 포맷인지 구분	
산 업	해당 전자문서가 특정 산업에만 적용될 때 사용. 예를 들어 구매주문서가 무역부문에서 사용되는 지, 아니면 제조 또는 유통, 물류부문에서 사용되는 지 구분	
제 품	해당 전자문서가 특정 회사 제품 포맷을 사용할 경우. PDF 포맷인지, 특정 회사 e-Form 포맷인지 등 구분	
기 업	해당 전자문서가 특정 기업에만 사용할 수 있는 경우 구분	
기 타	위의 구분이외에 다른 컨텍스트(문맥)로 구분할 수 있을 경우	

[316] 도 25에는 전자문서 서식등록기를 활용하는 프로세스에 대한 기본적인 흐름을 제시하고 있으며, 구체적으로는 도 25a에는 전자문서 서식등록기와 유통 클라이언트가 직접 연계되는 경우를 나타내고 있고, 도 25b에는 전자문서 서식등록기 사이트를 이용하는 경우를 나타내고 있다.

[317] 아래 표 29에는 유통 클라이언트와 직접 연계되어 서식을 활용하는 프로세스에 대하여 나타내었다.

[318] 표 29

번호	구분	설명
1	전자문서 서식 검색	유통 클라이언트에서 바로 전자문서 서식등록기에 있는 전자문서 서식을 검색
2	서식 가져오기	서식을 찾았을 경우, 이를 유통 클라이언트로 가져옴
3	서식 등록	유통 클라이언트로 가져온 서식을 유통 클라이언트에 등록
4	전자문서 서식 불러오기/작성	유통 클라이언트에 등록되어 있는 전자문서 서식을 불러와서 전자문서를 작성하고 이를 메시지에 첨부

[319] 아래 표 30은 전자문서 서식등록기 사이트를 활용하는 프로세스에 대하여 나타내었다.

[320] 표 30

번호	구분	설명
1	전자문서 서식 검색	전자문서 서식등록기 사이트에 접속하여 필요로 하는 전자문서 서식을 검색
2	서식을 로컬PC에 저장	검색된 서식을 다운로드하여 로컬PC에 저장
3	서식 등록	로컬 PC에 저장된 서식을 유통 클라이언트에 등록
4	전자문서 서식 불러오기/작성	유통 클라이언트에 등록되어 있는 전자문서 서식을 불러와서 전자문서를 작성하고 이를 메시지에 첨부

[321] 이하, 합의된 전자문서 서식 활용에 대하여 설명하면 다음과 같다.

[322] 특정 기업에 한정된 서식을 기업이 운영하는 사이트 등에서 서식을 배포하여 특정 기업과 관련된 당사자들과 거래를 하기 위한 목적으로 사용하기 위함이다.

[323] 도 26은 합의된 전자문서 서식을 이용하는 절차를 나타내고 있으며, 도 26과 관련된 프로세스에 대한 설명은 아래 표 31과 같다.

[324] 표 31

번호	구분	설명
1	전자문서 서식 배포	기업 A는 자신의 시스템에서 처리할 수 있는 전자문서 서식을 자신이 운영하는 웹사이트에 배포하여 거래하는 기업에서 이를 활용할 수 있도록 지원
2	서식을 로컬PC에 저장	기업 B2는 웹사이트에서 서식을 다운로드하여 로컬PC에 저장
3	서식 등록	로컬 PC에 저장된 서식을 기업B의 유통 클라이언트에 등록
4	전자문서 서식 불러오기/작성	유통 클라이언트에 등록되어 있는 전자문서 서식을 불러와서 전자문서를 작성하고 이를 메시지에 첨부
5	메시지 전송	유통 메시징서버를 통하여 메시지를 전송
6	첨부된 전자문서의 자동/반 자동 처리	수신받은 메시지에 첨부되어 있는 전자문서를 변환 등의 처리를 통하여 기업A의 내부 시스템에 자동 또는 반자동으로 등록

[325]

[326] [스팸 메시지 처리 프로세스]

[327] 유통체계에서 송신자는 인증된 유통 메시징서버를 통해 전송을 하고, 수신자 역시 이를 통해 수신하므로 스팸이 발송되었을 때 송신자에게 책임을 물을 수 있는 기반구조를 가지고 있다.

[328] 그러나, 특정 송신자가 전자문서 중계자에 사용자 계정을 개설하고 이를 이용해서 상업적인 목적을 위한 메시지 등을 전송하는 경우가 있을 수 있다. 현재의 인증방식이 시스템의 기술적 내용에 대한 인증만을 대상으로 하고 있어서 스팸 메시지 등을 초기에 원천적으로 봉쇄하기가 쉽지 않은 상황이다.

[329] 이러한 스팸 메시지 등의 유통을 차단하기 위해서, 유통체계에서는 인증목록 관리 기반의 화이트리스트, 스팸이나 악의적인 의도를 가진 블랙리스트를 관리할 수 있는 체계를 제공하여 유통체계의 안전성과 신뢰성을 제고하도록 한다.

[330] 스팸메시지 신고 및 송신 상대방에 대한 확인을 위한 기능은 필수기능으로 모든 유통 메시징서버는 이 기능을 반드시 구축하여야 한다.

[331] 이하, 스팸 메시지 신고방안에 대하여 설명하면 다음과 같다.

[332] 스팸 메시지 신고절차를 단계적으로 살펴보면 아래 표 32와 같다.

[333] 표 32

번호	프로세스명
1	수신자가 메시지를 수신한 시점에서 스팸메시지라고 판단되면, 수신자는 유통 메시징 서버를 통해 주소 디렉터리 서버에 해당 메시지를 수신 메시지로 신고함
2	유통 메시징서버로부터 스팸메시지 신고를 접수한 주소 디렉터리 서버는 접수되었음에 대한 확인 메시지를 되돌려 줌
3	주소 디렉터리 서버를 관리하는 주체인 관리기관은 해당 메시지를 분석하고, 송신자에 대한 조사를 통해 송신자의 공인전자주소에 대해 블랙리스트 추가 여부를 심사하고 판단함
4	최종적으로 블랙리스트 대상자로 확정이 되면, 주소 디렉터리 서버는 해당 공인전자주소를 블랙리스트에 추가한 후, 송신자에게 블랙리스트 추가에 대한 내용을 통지함
5	주소 디렉터리 서버는 스팸메시지 요청에 대한 처리 결과를 스팸신고자(수신자)에게 전달함

[334] 수신자는 수신한 메시지가 스팸메시지라고 판단이 되면 도 27과 같은 프로세스에 의해 스팸메시지를 주소디렉터리 서버에 신고한다.

[335] 화이트리스트와 블랙리스트의 역할은 다음과 같다.

[336] ① 화이트리스트: 송신 유통 메시징서버가 인증을 받고 정식으로 등록된 메시징서버에 대한 정보만 기록됨

[337] ② 블랙리스트: 전송자의 주소가 스팸발송자로 등록된 경우 블랙리스트로 등록됨

[338] ※ 동일한 유통 메시징서버를 통해 블랙리스트에 등록되는 스팸 주소가 중복 발생되는 경우, 관리기관은 유통 메시징서버에 대한 인증취소 여부를 판단한 후, 인증을 취소하고 화이트리스트에서 삭제할 수 있음.

[339] 이하, 스팸메시지 수신 시 처리 방법에 대하여 설명하면 다음과 같다.

[340] 수신 개체는 메시지 수신 시, 송신 상대방이 신뢰할만한 정당한 사용자인지 여부를 확인하기 위해 주소디렉터리 서버의 화이트리스트와 블랙리스트를 확인한 후, 수신거부를 할지 여부를 결정한다.

[341] 송신자에 대한 확인은 수신시점에 ①실시간 확인을 하거나, 수신자의 유통 메시징서버 시스템에 Cache형태로 관리하고 있는 목록을 통해 확인하는 ②주기적 확인 방법이 있다.

[342] ①실시간 확인 프로세스: 수신 개체는 메시지를 수신하는 시점에 주소 디렉터리 서버에 송신 개체의 주소가 화이트리스트, 블랙리스트에 등록되었는지 여부를 판단한 후 메시지에 대한 수신거부 여부를 결정한다.

[343] 표 33

번호	프로세스명
1	수신자의 유통 메시징서버는 메시지를 수신하면 주소 디렉터리 서버에 정당한 사용자인지 여부를 확인하기 위해 확인요청 메시지를 전달
2	주소 디렉터리 서버는 요청 받은 사용자의 주소정보가 화이트리스트에 포함되어 있는지 여부를 확인
3	해당 주소가 화이트리스트에 없으면 주소 디렉터리 서버는 바로 확인 요청자에게 등록되지 않은 사용자임을 결과메시지로 돌려주고, 화이트리스트에 있으면 다시 해당주소가 블랙리스트에 등록된 주소인지 여부를 확인
4	주소 디렉터리 서버는 확인 요청자에게 블랙리스트 등록여부에 대한 결과 메시지를 리턴
5	수신자는 주소 디렉터리 서버로부터 송신자가 정당한 사용자가 아님(화이트리스트에 없거나 블랙리스트에 등록된 경우)으로 결과메시지를 받은 경우 수신메시지를 자체적으로 스팸메시지로 처리한 후, 주소디렉터리 서버로부터 받은 처리결과 메시지와 스팸메시지 수신이력을 기록하고 보관
6	스팸메시지에 대한 처리 이력은 반드시 한달 이상 보관함으로써, 해당 송신자에 대한 수신거부의 정당성을 확인

[344] 도 28은 스팸메시지를 실시간으로 확인하는 프로세스를 보여준다.

[345] ②주기적 확인 프로세스: 수신자는 주소 디렉터리 서버로부터 화이트리스트와
블랙리스트를 주기적으로 받아서 자체 관리하고 이를 기반으로 송신자의
주소가 화이트리스트, 블랙리스트에 등록되었는지 여부를 판단한 후 메시지에
대한 수신거부 여부를 결정

[346] 표 34

번호	프로세스명
1	수신 개체의 유통 메시징서버는 미리 주소 디렉터리 서버로부터 최신의 화이트리스트와 블랙리스트를 요청한 후 자체적으로 관리. 이때, 리스트의 변동사항 발생 시 자동 통지 요청여부를 같이 전달함(변동사항 발생 자동통지를 요청한 경우에도 주소 디렉터리 서버에 최신 리스트를 가져오기 위한 요청은 주기적으로 함으로써 리스트 정보가 최대한 1일 이상 차이 나지 않도록 함)
2	주소 디렉터리 서버는 화이트리스트 및 블랙리스트에 변동사항이 발생하면 변동 통지요청을 한 사용자에게 변동내역을 broadcasting함
3	리스트에 대한 변동사항을 받은 유통 메시징서버는 자체 관리하는 리스트의 정보를 수정함으로써 동기화를 시켜줌
4	수신자는 메시지를 수신하면 주소 디렉터리 서버에 정당한 사용자인지 여부를 확인하기 위해 자체 관리하는 리스트를 확인
5	수신자는 자체 관리하는 리스트 점검결과 송신자가 정당한 사용자가 아님(화이트리스트에 없거나 블랙리스트에 등록된 경우)으로 판단한 경우 수신메시지를 자체적으로 스팸메시지로 처리한 후, 스팸메시지 수신이력을 기록하고 보관함
6	스팸메시지에 대한 처리 이력은 반드시 한달 이상 보관함으로써, 해당 송신자에 대한 수신거부의 정당성을 확인할 수 있도록 함

[347] 도 29는 유통 메시징서버와 주소 디렉터리 서버 간 스팸메시지 관리를 위해 주기적으로 확인하는 프로세스를 보여준다.

[348]

[349] [오류 처리 프로세스]

[350] 유통체제에서 오류 발생 유형은 아래 ①과 ②를 포함하는 2가지로 구분된다.

[351] ① 동기식 응답에 대한 오류 발생 : 동기식 응답에 대한 오류의 경우는 요청메시지에 대한 처리결과를 받을 때까지 요청자는 대기하고 있는 상태이므로 오류를 즉각 인지 가능

[352] ② 비동기식 응답에 대한 오류 발생 : 요청자는 요청 내용만 전달한 후 그에 대한 처리결과를 차후에 받게 되므로 추가적인 오류처리가 이루어져야 함

[353] 이하, 동기식 오류처리 방안에 대하여 설명하면 다음과 같다.

[354] 동기식 전송에 대한 모든 오류는 전송자가 즉각 확인 가능하므로 재전송하는 것을 기본으로 한다. 재전송 방식에 대해서는 유통체제에 참여하는 기업이나 기관의 시스템 정책에 따라 결정되나, 동일한 메시지 전송에 대해서는 동일한 MessageId 값을 설정하여 다시 보내는 것을 기본으로 한다.

[355] 동기식 오류처리와 관련한 유형은 다음 ① 내지 ④와 같다.

- [356] ① 요청메시지 송신실패 : 송신자가 메시지를 전송하는 과정에서 전송오류가 발생하여 수신자에게 요청메시지가 전달되지 못하는 경우로서, 송신자는 전송시도에 대한 timeout 또는 네트워크 오류 메시지 등을 통해 송신 실패를 인지하게 된다.
- [357] ② 응답메시지 수신실패 : 송신자가 메시지를 정상적으로 전송하였으나 수신자로부터 응답메시지를 받는 과정에서 오류가 발생한 경우이다.
"①요청메시지 송신실패"와 구분이 되지 않으므로 오류에 대해서 동일한 방식으로 처리하나, 수신자는 요청메시지를 정상적으로 받았기 때문에 처리 방식이 다르다.
- [358] ③ 오류메시지 수신 : 송신자가 메시지를 정상적으로 전송하였으나 수신자가 메시지를 처리하는 과정에서 오류가 발생한 경우이다. 이 경우 송신자의 처리방식은 오류메시지의 유형에 따라 다르다.
- [359] ④ 3단계 동기식 오류 : 유통 클라이언트가 유통 메시징서버를 통해 타 유통 메시징서버, 주소 디렉터리 서버, 유통 중계허브와 연계되는 3개의 개체 간 메시지 유통은 연계 유형 중 최종적 결과를 즉각적으로 확인하기 위해 동기식으로 연계하는 방안을 지원한다. 이 과정에서 유통 메시징서버와 수신자 간 연계단계에서 오류가 발생하면 유통 메시징서버는 즉각적으로 오류를 발생시킨 후, 이를 유통 메시징서버에 응답메시지로 전달하여야 한다.
- [360] 이하, 비동기식 오류처리 방안에 대하여 설명하면 다음과 같다.
- [361] 유통 클라이언트가 유통 메시징서버를 통해 타 유통 메시징서버, 주소 디렉터리 서버, 유통 중계서버와 연계되는 3개체 간 메시지 유통은 유통 클라이언트가 최종 수신자의 상황에 독립적으로 운영할 수 있도록 비동기식 방식의 연계를 지원하기도 한다.
- [362] 비동기식 전송에 대한 최종 오류는 동기식 전송과 달리, 전송자가 즉각 확인할 수가 없으므로 유통 메시징서버가 최종적으로 오류를 확인한 시점에 유통클라이언트를 위한 오류 메시지를 발생시키고, 이를 유통 클라이언트가 수신할 수 있도록 하여야 한다.
- [363]
- [364] [전자문서 열람서비스]
- [365] 전자문서 열람서비스는 송신자와 수신자 간에 전자문서를 직접 주고 받는 교환의 형태가 아닌, 송신자의 시스템 또는 제3자 보관기관에 보관되어 있는 전자문서를 열람할 수 있도록 제공하는 서비스이다.
- [366] 전자문서 열람서비스는 기존 유통체계를 그대로 이용한다. 단, 송신자는 수신자에게 전자문서를 담은 메시지를 전송하는 것이 아니라, 송신자 시스템 또는 제3자 보관기관에 보관되어 있는 전자문서를 열람할 수 있는 열람권한을 담은 메시지를 전송하는 것이 특징이다.
- [367] 이와 관련된 절차는 다음과 같다.
- [368] ①수신자의 공개키 획득

- [369] ②전자문서의 보관
- [370] ③열람권한 및 DRM 등 보안이 적용된 증서 생성
- [371] ④열람권한 등 증서의 전송 및 전송확인
- [372] ⑤수신자에 의한 전자문서 열람
- [373] ⑥유통(열람)증명서 또는 열람증적 발급 및 보관
- [374] 전자문서 열람서비스는 송신자가 자체 시스템을 사용하는 방법과 제3자를 이용하는 방법으로 구분할 수 있다.
- [375] 도 30은 송신자가 자체 시스템을 활용하여 전자문서 열람서비스를 이용하는 흐름을 제시하고 있으며, 도 30에서 명시하고 있는 절차에 대하여 설명하면 아래 표 35와 같다.
- [376] 표 35

번호	프로세스명	설명
1	수신 개체 인증서의 공개키 획득	전자문서 열람 권한을 생성할 때 필요한 수신 개체 인증서의 공개키를 주소 디렉터리 서버로부터 획득
2	전자문서 보관 및 열람권한 생성	전자문서를 송신 개체의 자체 시스템에 보관
3	열람권한 및 DRM 등 보안 적용한 증서 생성	보관된 전자문서를 열람할 수 있는 권한 및 DRM 등 보안을 적용한 열람권한 증서를 생성
3	열람권한 전송	열람권한을 포함한 증명서를 수신 개체에게 전송
4	전자문서 열람	수신 개체의 수신자는 열람권한을 가지고 송신 개체의 전자문서 열람시스템에 접속하여 전자문서를 열람
5	유통(열람)증명서 보관	수신 개체의 수신자가 전자문서를 열람하였을 경우, 송신 개체는 유통(열람)증명서를 생성하고 이를 제3자 보관기관에 보관

- [377]
- [378] 위와 같은 서비스를 제공하기 위해서 송신자는 유통체계 이외에 전자문서 열람서비스를 제공할 수 있는 시스템을 구비하여야 한다.
- [379] 도 31에는 송신 개체가 제3자를 활용하여 전자문서 열람 서비스를 이용하는 흐름을 나타내었으며, 도 31에서 제시하고 있는 절차에 대하여 설명하면 아래 표 36과 같다.
- [380] 표 36

번호	프로세스명	설명
1	수신 개체 인증서의 공개키 획득	전자문서 열람 권한을 생성할 때 필요한 수신 개체 인증서의 공개키를 주소 디렉터리 서버로부터 획득
2	전자문서 보관 및 열람서비스 의뢰	송신 개체는 전자문서를 제3자 보관기관에 보관, 의뢰하고, 이에 대한 열람서비스를 의뢰(수신 개체의 공인전자주소 기재)
3	열람권한 및 DRM 등 보안이 적용된 증서 생성	제3자 보관기관 보관 의뢰받은 전자문서를 열람할 수 있는 권한증서와 DRM 등 보안이 적용된 증서를 생성
4	열람권한 증서 전송	제3자 보관기관 열람권한을 포함한 증서를 수신 개체에게 전송
5	전자문서 열람	수신 개체의 수신자는 열람권한을 가지고 제3자 보관기관에 접속하여 전자문서를 열람
6	유통(열람)증명서 보관	제3자 보관기관 수신 개체의 수신자가 전자문서를 열람하였을 경우, 유통(열람)증명서를 생성하고 이를 보관
7	유통(열람)증명서 전송	아울러, 제3자 보관기관 열람서비스를 의뢰한 송신 개체에게 유통(열람)증명서를 전송

[381]

[382] [기업 내 시스템과의 연계 방법]

[383] 기업/기관은 내부적으로 다양한 전자문서를 생산하고 보관하기도 하고, 외부
기업/기관 등과 다양한 방식으로 전자문서를 주고받는다.[384] 우편 등과 같은 오프라인으로 교환하기도 하고, 이메일이나 업무관련
시스템으로 교환하기도 한다. 이들 각각의 유통 체계는 기업/기관 내부로 연계될
때 아래 표 37 및 도 119와 같은 방식으로 연계된다.

[385] 표 37

구분	설명
우편유통체계	- 거래 당사자는 공문 등 서류를 우편으로 발송 - 기업/기관 담당자가 공문 등 서류가 담긴 우편 수신- 우편을 뜯어 공문 등 서류를 수령, 등록하고 내부 결재- 결재후, 문서 수신함에 서류 형태로 보관 -(장점) 1)오래전부터 사용되었기 때문에 익숙한 방식, 2)IT시스템 비용 등이 들지 않음 -(단점) 1)유통에 시간이 많이 걸림, 2)서비스 수수료 및 서류 보관 등에 비용 발생, 3)검색 등에 시간이 많이 걸림
e-메일 유통체계	- 거래 당사자는 공문 등 서류를 e-메일로 발송 - 기업/기관 담당자가 이메일 프로그램으로 공문 등 서류가 담긴 e-메일 수신 -(내부 시스템과 연동된 경우) 수신된 메시지 내에 있는 전자문서를 내부 시스템으로 반자동 형태로 등록, 처리-(내부 시스템과 연동되지 않은 경우) 수신된 메시지 내에 있는 전자문서를 담당자 로컬PC에 저장. 내부 시스템에 접속하여 저장된 전자문서를 등록,처리 -(장점) 1)사용하기 편함, 2)업무의 보조수단으로 전 세계적으로 널리 활용, 3)비용이 거의 들지 않음 -(단점) 1)담당자들 간에 교환되기 때문에 담당자들의 추가적인 업무처리가 필요하고, 담당자들의 실수나 변동 등에 의해 업무 장애가 발생할 수 있음 2)e-메일 프로토콜의 불안정성으로 인해 수신 오류 등이 발생하였을 경우, 법적인 분쟁이 발생할 수 있음 3)유통되는 전자문서를 기업/기관이 총괄적으로 관리하지 못하고, 담당자들에 의해 지역적으로 처리, 관리될 수밖에 없음 4)업무의 단순한 보조수단으로 1회성 사용목적, 보관/삭제에 대한 개념 취약 5)개인에 의한 e-메일 관리가 보편화되었으며, 기업 e-메일에 대한 사용과 관리 원칙 및 통제 부족 6)대부분 워드문서 등 비구조화된 문서에 대한 유통으로 자동화된 처리 불가(반자동 내지 수동 처리)
전자문서 교환(EDI)유통체계	- 거래 당사자는 전자문서를 EDI를 통해 발송 - 수신자는 EDI 어플리케이션을 통하여 전자문서 수신 - 수신된 전자문서는 자동 변환, 처리되어 내부 시스템에 등록 - 기업/기관 담당자는 내부 시스템에 등록된 전자문서를 업무에 따라 확인, 처리. -(장점) 1)사람의 간섭 없이 전자문서를 자동으로 변환,처리함으로써 업무 효율성 높음, 2)정확한 전송이 가능하며, 전송 오류에 대한 분쟁시 중계자 책임 -(단점) 1)초기 투자비용이 많이 소요되거나 아웃소싱할 경우 서비스 수수료 발생, 2)사전에 거래당사자와 업무, 문서 등에 대해 합의가 필요, 3)구조화된(표준화된) 문서만 유통 가능. 비구조화된 문서 유통 불가

업무관련시스템유동체계	- 거래 당사자는 웹 브라우저를 통해 수신자가 운영하는 웹사이트에 로그인 등 이용자 인증을 통하여 접속 - 웹사이트에서 제공하는 절차와 방법으로 문서 작성 및 파일 첨부 처리 - 기업/기관 담당자는 업무관련 시스템에 접속하여 수신받은 전자문서 확인. 업무관련 시스템 자체가 내부 시스템이기 때문에 특별한 처리는 없음
	- (장점) 1)수신자는 자동으로 전자문서 등을 수신하기 때문에 수신자의 업무 효율성 높음, 2)송신자는 추가적으로 프로그램 설치 불필요 및 유통 비용이 발생하지 않음 - (단점) 1)수신자 위주의 유통체계로 송신자는 일일이 수신자가 운영하는 시스템에 접속해야 하기 때문에 송신자의 불편 가중, 2)전자문서가 수신자 시스템에만 남기 때문에 향후 분쟁 발생시 송신자에게 불리

[386] 공인전자주소 기반 전자문서 유통체계는 다음과 같은 방식으로 기업/기관 내부와 연계될 수 있다.

[387] 이하, 내부 시스템과 연동하는 방식에 대하여 설명하면 다음과 같다.

[388] 내부 시스템과 연동하는 방식은 주로 기업이나 기관에서 유통 메시징서버를 설치할 때 사용하는 형태로, 유통 클라이언트를 내부 시스템에 시스템 통합(SI) 형태로 개발하는 방식이며, 상세한 설명 및 그림은 아래 표 38 및 도 120과 같다.

[389] 표 38

구분	설명
자동처리방식	- 기업 내부 시스템에서 이용하는 이용자 인증체계 공유 - 전자문서는 표준 전자문서 서식 또는 합의된 전자문서 서식 사용 - 특정 공인전자주소로 온 메시지의 경우, 자동 처리될 수 있도록 유통 문서 처리시스템 필요(메시지 내 전자문서의 파싱, 적합성 검증 등) - 공인전자주소에 따라 해당 내부 시스템으로 전송,등록
반자동처리방식	- 기업 내부 시스템에서 이용하는 이용자 인증체계 공유 - 유통 클라이언트에는 전자계약시스템 등 내부 시스템과 연동체계가 구축되어 있어야 함 - 기업/기관 담당자는 유통 클라이언트 내용에 따라 내부 시스템에 반자동 형태로 등록

[390] 이하, 내부 시스템과 연동하지 않는 방식에 대하여 설명하면 다음과 같다.

[391] 내부 시스템과 연동하지 않는 방식은 주로 전자문서 증계자에게 사용자 계정을 발급받아 이용하는 공인 송수신자에 적합한 형태로써, 전자문서 증계자가

제공하는 웹 형태의 유통 클라이언트를 이용하거나 전자문서 중계자가 배포하는 유통 클라이언트 어플리케이션을 로컬 PC에 설치하여 이용하는 방식이며, 상세한 설명 및 그림은 아래 표 39 및 도 121과 같다.

[392] 표 39

구분	설명
웹방식	- 전자문서 중계자가 운영하는 웹사이트에 회원 가입하여 사용자 계정 확보 - 웹사이트를 기반으로 유통 메시징서버 접속. 수신받은 메시지 내 전자문서를 로컬PC에 저장 - 기업 내부 시스템이 있는 소기업의 경우, 로컬PC에 저장되어 있는 전자문서를 내부 시스템에 등록, 보관
어플리케이션방식	- 전자문서 중계자는 유통 클라이언트 설치 파일 배포 - 송수신자는 유통 클라이언트 어플리케이션을 로컬PC에 설치 - 유통 클라이언트 어플리케이션으로 유통 메시징서버 접속. 수신받은 메시지 내 전자문서를 로컬PC에 저장 - 기업 내부 시스템이 있는 소기업의 경우, 로컬PC에 저장되어 있는 전자문서를 내부 시스템에 등록, 보관

[393] *웹방식은 웹방식으로 사용자가 접속, 처리하는 방식으로, 개별 사용자가 로컬PC에 프로그램을 설치할 필요 없음.

[394] *어플리케이션방식은 개별 사용자가 설치 프로그램을 다운받아, 로컬PC에 프로그램을 설치한 다음, 유통 메시징서버에 접속하여 사용.

[395]

[396] [외부 시스템과의 연계 방안]

[397] 유통체계는 자체 시스템과 인터넷을 기반으로 유통 네트워크를 가진다. 전자문서 유통은 유통체계 내에서만 이루어지기 때문에 제약이 있을 수 있다. 유통체계는 직접 연결되지 않은 시스템과의 전자문서 유통을 위해 외부 연계 게이트웨이 서버를 두고 있다.

[398] 외부 연계 게이트웨이 서버의 기본 개념도는 도 32와 같다.

[399] 외부 연계 게이트웨이 서버는 중간 경유지 역할을 수행한다. 한쪽은 유통체계를 위한 유통 메시징서버 등을 가지고 있으며, 다른 한쪽은 외부 시스템과의 연계를 위한 각각의 어댑터들을 가지고 있다.

[400] 이와 같이, 외부 시스템과 연계하기 위해서는 업무적인 요소와 기술적인 요소를 고려하여야 한다.

[401] 업무적인 요소는 연계와 관련한 업무절차, 방법 등에 대한 당사자 간 합의로, 관리기관과 외부 연계 시스템 관리기관은 상호간에 서비스 레벨 협약(SLA)을

맺어야 한다.

- [402] 기술적인 요소는 연계와 관련해 필요한 이용자 인증, 메시징, 메시지 포맷 등과 관련한 기술 요소를 말한다.
- [403] 외부 시스템과 연계를 하기 위한 기술적인 원칙을 정리하면 다음 ① 내지 ⑥과 같다.
- [404] ①(주소) 송신자는 중간경유지 주소와 최종 목적지 주소를 기재하여야 한다.
- [405] ②(이용자 인증) 송신자는 외부 시스템이 송신자를 인증할 수 있도록 이용자 정보를 제공하여야 한다. 사전에 외부 시스템에 가입되어 있을 경우, 외부 시스템의 인증 식별자를 사용할 수 있다.
- [406] ③(메시지 분해 & 조립) 수신한 메시지를 분해해서 외부 시스템에 맞는 메시지로 조립하여야 한다.
- [407] ④(메시지 보안 등) 메시지에 적용된 암호화, DRM 등에 대한 보안 등을 검증/변환하여야 한다.
- [408] ⑤(메타데이터 정보) 메시지 내에 포함되어 메시지 및 전자문서 관련 정보들을 검증/변환하여야 한다.
- [409] ⑥(외부 연계 시스템에 대한 식별) 유통체계와 연계되는 외부 시스템은 추가나 변경될 수 있다. 외부 시스템에 대한 정보는 주소 디렉터리 서버에서 관리하며, 유통 메시징서버에서는 필요한 경우에 주소 디렉터리 서버에 질의하여 처리하여야 한다.
- [410] 외부 시스템과 연계하여 전자문서가 유통되는 절차는 도 33과 같다.
- [411]
- [412] 상술한 바와 같은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템 및 이를 이용한 전자문서 유통 방법에 있어서, 전자문서 유통이 이루어지기 위해서는 주소 디렉터리서버, 유통 메시징서버, 유통 클라이언트, 유통 중계서버와 같은 구성 요소들이 필요하며, 이들 구성요소들은 전자문서유통의 전체적인 흐름 속에서 서로 연계되어야만 한다. 이처럼 각 구성요소들 간에 서로 연계되어 동작하기 위해서는 연계를 위한 통신 프로토콜, 메시지 교환방식, 연계 메시지 구조 등이 정의되어야 한다.
- [413] 이하에서는 각 구성요소들 간의 연계를 위해 먼저 공통적인 기반 통신 프로토콜 및 메시지 교환방식을 정의하고 각 연계유형별 메시지 구조를 표준으로 정의하여 제시하며, 이로써 본 발명은 서로 다른 환경 및 개발방법에 의해 구축된 구성요소들 간의 연계가 원활하게 이루어지고 상호운용이 가능하도록 할 수 있다.
- [414]
- [415] [연계를 위한 기반 통신 프로토콜]
- [416] 공인전자주소 기반 전자문서 유통체계 하에서 각 구성요소들 간의 정보 및 전자문서를 유통하기 위해 필요한 전자문서 유통 연계 인터페이스에 있어서, 유통 연계 메시지는 "ebXML Message Services v2.0 규격"(이하 ebMS)을 기반으로

하며, 이를 계층적으로 확장하여 보다 일반화된 형태로 정의된다. ebXML 기반 구조는 SOAP, SOAP with Attachment, Security 그리고 Reliability 등 독립적이지만 서로 밀접한 관계가 있는 요소들로 구성되어 있다. "연계를 위한 기반 통신 프로토콜"(이하 기반 통신 프로토콜)은 이러한 기본 요소들을 기반으로 유통체제에서 필요한 요소들을 정의하고 이를 유기적으로 재조합한 형태로 구성된다.

- [417] 기반 통신 프로토콜은 전송하고자 하는 메시지를 구성하는 패키징, 메시지 봉투 구성, 메시지 보안 그리고 최종적으로 메시지를 전송하고 수신하는 메시지 송수신으로 구성된다.
- [418] 이하, 기반 프로토콜 구성에 있어서 메시지 패키징에 대하여 설명하면 다음과 같다.
- [419] 유통 연계 메시지의 전체 메시지 구조는 ebMS v2.0 규격을 준용한다. 본 기반 통신 프로토콜에서 정의하는 메시지는 두 개의 논리적인 MIME Part를 가진다.
- [420] 첫 번째 MIME Part는 헤더 컨테이너라고 불리는데, SOAP 메시지를 포함하고 있으며, SOAP 메시지는 다시 Header와 Body로 구성된다.
- [421] 두 번째 MIME Part부터는 페이로드 컨테이너라고 불리는데, 어플리케이션 레벨의 메시지 및 첨부 문서를 포함한다.
- [422] 첫 번째 MIME Part에 대한 상세 설명은 아래에서 하도록 한다. 이 영역에는 메시지 유통을 위한 공통적인 정보(메시지의 라우팅 관련 정보, SOAP 메시지 교환패턴, 전자서명, 오류 정보 및 두 번째 첨부되는 데이터에 대한 위치 정보 등)가 기술된다.
- [423] 두 번째 MIME Part는 각 연계 인터페이스 별 요청 및 응답메시지를 첨부하게 되며, 이 연계 인터페이스의 유형에 따라 세 번째 이후 MIME Part의 존재여부가 결정된다. 유통체제를 이용하여 전자문서나 증명서를 전달할 때 세 번째 MIME Part에 포함된다.
- [424] 유통 연계 메시지의 기본적인 구조는 도 34와 같으며, 도 34에서 ① "SOAP-ENV: Header"는 유통 프로토콜 규격에 따라 구성되며 MessageHeader 및 Signature 정보로 구성되고, ② "SOAP-ENV: BODY"는 유통 프로토콜 규격에서 정의한 Manifest 요소 정보 및 사용자 로그인 정보가 들어가며, ③ "전송 컨테이너 #1(페이로드 컨테이너 #1)"은 요청메시지 및 응답메시지를 포함하는 부분이며, 연계 인터페이스의 유형 및 요청, 응답, 오류 메시지 여부에 따라 비즈니스 문서의 상세 내용이 정의되고, ④ "전송 컨테이너 #2(페이로드 컨테이너 #2)"는 연계인터페이스의 유형에 따라 첨부되어야 하는 문서가 페이로드 컨테이너 #2부터 순차적으로 들어간다.
- [425] 이러한 유통 연계 메시지는 Simple Object Access Protocol(SOAP) 1.1 및 SOAP Messages with Attachment 와 같은 표준 규격을 준수해야 한다.
- [426] 도 34에서, 유통 연계 메시지의 모든 MIME Header 요소들은 SOAP Messages with Attachments 규격을 준수해야 한다. 추가로, 메시지 내의 Content-Type MIME

Header는 반드시 SOAP 메시지 문서를 포함하는 MIME Body 부분의 MIME 미디어 유형과 동일한 type 속성을 가지고 있어야 한다. SOAP 규격에 따른 SOAP 메시지의 MIME 유형은 "text/xml" 값을 가져야 한다고 되어 있다. 루트 부분은 [RFC2045]에 준하는 구조를 가지는 Content-ID MIME Header를 포함할 것과 Multipart/Related 미디어 유형에 대한 필수적인 파라미터에 추가하여, start 파라미터([RFC2387]에서는 선택사항)가 항상 존재해야 한다. multipart/related 메시지 패키지의 MIME Header 예제는 아래 표 40과 같다.

[427] 표 40

```
Content-Type: multipart/related; type="text/xml";
boundary="boundaryValue";start=messagepackage-123@example.com--boundaryValue
Content-ID: <messagepackage-123@example.com>
```

[428] 도 34에서, 첫번째 MIME Part 헤더 컨테이너는 반드시 SOAP 메시지를 포함하고 있어야 한다. 헤더 컨테이너의 MIME Content-Type header는 SOAP 규격에 따라 "text/xml" 값을 가져야 한다. Content-Type 헤더는 "charset" 속성을 포함해야 하며, 예제는 아래 표 41과 같다. 그리고, MIME charset 속성은 SOAP 메시지를 생성하는데 사용되는 문자 군을 식별하기 위해 사용된다. MIME charset 속성 값과 헤더 컨테이너에 위치하는 SOAP 메시지의 인코딩 선언부는 반드시 일치해야 하며 그 값은 UTF-8이어야 한다. 헤더 컨테이너의 예제는 아래 표 42와 같다.

[429] 표 41

```
Content-Type: text/xml; charset="UTF-8"
```

[430] 표 42

```
Content-ID: <messagepackage-123@example.com> --| HeaderContent-Type: text/xml;
charset="UTF-8" . ||<SOAP:Envelope --|SOAP Message | xmlns:SOAP="
http://schemas.xmlsoap.org/soap/envelope/"> || <SOAP:Header> || ... ||
</SOAP:Header> || <SOAP:Body> || ... || </SOAP:Body> || </SOAP:Envelope> --| |
```

[431] 도 34에서, 페이로드 컨테이너의 개수는 연계 인터페이스 종류에 따라 달라질 수 있다. 각 페이로드 컨테이너는 ebMS 규격에 따라 SOAP Body 내의 Manifest 요소에 의해 참조가 되어야 한다. 예제는 아래 표 43과 같다.

[432] 표 43

```
Content-ID: <domainname.example.com> -----| ebXML MIME |Content-Type:
application/xml -----| | Payload<Invoice> -----| |
Container<Invoicedata> | Payload | ... || </Invoicedata> | </Invoice> -----| |
```

[433] 도 34에서, 본 발명에서 필수 요소로 지정한 MIME Header 외에 구현 편의상

MIME Header들을 추가하는 것도 가능하다. 이때, 추가되는 MIME Header는 반드시 [RFC2045]에 명시된 항목이어야 한다. 그러나, 추가적인 MIME Header에 대해서는 이를 추가하지 않은 측에서 이를 인지하고 해석할 필요는 없다.

[434] 이하, 기반 프로토콜 구성에 있어서 메시지 봉투 구성에 대하여 설명하면 다음과 같다.

[435] SOAP 규격에 준하여 모든 확장 요소 내용은 유효한 네임스페이스로 한정되어야 한다. 본 발명에서 정의된 모든 ebXML SOAP 확장 요소 내용은 ebXML SOAP Envelope 확장 네임스페이스로 한정되어야 한다. 네임스페이스 선언부들은 SOAP Envelope, Header 또는 Body 요소들에 포함되어 있거나, 각 SOAP 확장 요소들에 직접 포함되어 있을 수 있다.

[436] SOAP Envelope는 SOAP 메시지의 Root 항목으로 SOAP 메시지 내의 각종 Namespace들은 선언한다. 선언해야 할 Namespace는 다음과 같다.

[437] 표 44

항목	Namespace URL
SOAP	http://schemas.xmlsoap.org/soap/envelope/
Digital Signature	http://www.w3.org/2000/09/xmldsig#
Xlink	http://www.w3.org/1999/xlink
Xsi	http://www.w3.org/2001/XMLSchema-instance

[438] 메시지 봉투 스키마 구조는 도 35와 같으며, 메시지 봉투 예제는 아래 표 45와 같다.

[439] 표 45

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
<SOAP:Headerxmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema
/msg-header-2_0.xsd"xsi:schemaLocation="http://www.oasis-open.org/committees/eb
xml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:MessageHeader ...> ... </eb:MessageHeader> </SOAP:Header>
<SOAP:Bodyxmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/
msg-header-2_0.xsd"xsi:schemaLocation="http://www.oasis-open.org/committees/eb
xml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:Manifest eb:version="2.0"> ... </eb:Manifest>
</SOAP:Body></SOAP:Envelope>
```

- [440] SOAP Header 요소는 SOAP Envelop 요소의 첫번째 자식 요소로서 다음 ① 내지 ④와 같은 확장 요소를 포함한다.
- [441] ① MessageHeader: 메시지의 라우팅 정보(To/From, 등)와 메시지에 관한 다른 문맥정보를 포함한 필수 요소
- [442] ② SyncReply: 메시지를 송수신하는 방식이 동기식임을 나타내는 요소
- [443] ③ Signature: SOAP 메시지 및 첨부 문서에 대한 전자서명 값을 나타내는 요소
- [444] ④ ErrorList: 메시지 구문 검증, 메시지 전자서명 검증 등 메시지를 처리하는 과정에서 오류가 발생하여 오류 메시지를 반환할 때 해당 오류 내역을 담는 요소
- [445] SOAP Body 요소는 SOAP Envelope 요소의 두 번째 자식 요소로서 Manifest와 같은 확장 요소를 포함한다. Manifest는 페이로드 컨테이너 또는 웹과 같이 다른 장소에 위치한 데이터를 가리키는 요소이다.
- [446] Manifest 요소는 페이로드 컨테이너를 참조하기 위해 반드시 존재해야 한다. Manifest 요소는 한 개 이상의 Reference 요소들로 구성된 복합 요소이다. 각 Reference 요소는 페이로드 컨테이너에 담긴 페이로드 문서(들)의 일부로 포함되거나 URL로 접근 가능한 원거리의 자원인 메시지에 관련된 데이터를 식별한다. SOAP Body에는 페이로드 데이터를 실지 않도록 권고하고 있다. Manifest의 목적은 다음 ① 및 ②과 같다.
- [447] ① ebXML 메시지와 관련된 특정한 페이로드를 쉽게 직접적으로 접근할 수 있게 함
- [448] ② 파싱 작업 없이도 어플리케이션이 페이로드를 처리할 수 있는지 판단할 수 있도록 함
- [449] Manifest 요소는 다음 ① 내지 ③과 같은 속성과 요소들로 구성되어 있다.
- [450] ① 한 개의 id 속성
- [451] ② 한 개의 version 속성
- [452] ③ 한 개 이상의 Reference 요소
- [453] Reference 요소는 다음 ① 내지 ②와 같은 하위 요소들로 구성된 복합 요소이다.
- [454] ① 0개 이상의 Schema 요소: 부모 Reference 요소에서 식별된 인스턴스 문서를 정의하는 스키마(들)에 대한 정보
- [455] ② 0개 이상의 Description 요소: 부모 참조 요소에 의해 Reference된 페이로드 객체에 대한 설명
- [456] Reference 요소는 그 자체가 [XLINK]의 단순 링크이다. XLINK는 현재 W3C의 후보 권고안(CR)이다. 여기에서 XLINK가 사용된 것은 연관관계의 설명을 명확하게 하기 위한 용어로 제공되어 있음을 알린다. XLINK 프로세서 또는 엔진의 사용이 필수적이지는 않지만 구현 요구 사항에 따라 유용할 수 있다. Reference 요소는 위에서 제공된 요소의 내용과 더불어 다음 ① 내지 ⑤와 같은 속성 내용을 포함하고 있다.
- [457] ① id: Reference 요소에 대한 XML ID
- [458] ② xlink-type: 이 속성은 XLINK 단순 링크로 요소를 정의. 이는 "simple"이라는

고정된 값을 가짐

- [459] ③ xlink:href: 이 필수 속성은 참조된 페이로드 객체의 URI 값. 이는 [XLINK] 명세의 단순 링크에 준하는 것이어야 함
- [460] ④ xlink:role: 이 속성은 페이로드 객체나 그의 목적을 설명하는 자원을 식별. 존재한다면 [XLINK] 명세에 준하는 유효한 URI 값을 가져야 함
- [461] ⑤ 이 외에 다른 유효 네임스페이스인 속성들이 존재할 수 있음. 수신 MSH는 위에서 정의한 것 외에 외부의 네임스페이스 속성들은 무시할 수 있음
- [462] Schema 요소는 참조하는 항목이 그 것을 기술하는 스키마(들)를 가지고 있다면 (예, XML Schema, DTD, 또는 Database Schema), 그 Schema 요소는 Reference 요소의 자식 요소로 존재해야 한다. 이는 스키마와 버전을 식별하는 방법으로 사용되며, 부모 Reference 요소에 의해 식별되는 페이로드 객체를 정의한다. Schema 요소는 다음 ① 및 ②와 같은 속성을 가지고 있다.
- [463] ① location: 스키마의 필수 URI
- [464] ② version: 스키마의 버전 식별자
- [465] xlink:href 속성이 content id(URI scheme "cid")인 URI를 포함하고 있으면 그 content-id를 가지는 MIME은 메시지의 페이로드 컨테이너에 표현되어 있어야 한다. 그렇지 않으면 errorCode를 MimeProblem으로, severity를 Error로 하는 오류를 발신당사자에게 전달해야 한다. xml:href 속성이 content id(URI scheme "cid")인 URI를 포함하고 있지 않으면 URI는 해석되지 못하고, 구현에 따라 오류를 전달해야 할지 말아야 할지 결정해야 한다. 오류가 전달되어야 한다고 결정되면 errorCode를 MimeProblem으로, severity를 Error로 하는 오류를 발신당사자에게 전달해야 한다.
- [466] 아래 표 46은 전형적인 한 개의 페이로드 MIME Body 부분을 가지는 메시지의 Manifest를 나타낸다.
- [467] 표 46

```
<eb:Manifest eb:id="Manifest" eb:version="2.0"> <eb:Reference eb:id="pay01"
xlink:href="cid:payload-1" xlink:role="http://regrep.org/gci/purchaseOrder">
<eb:Schema eb:location="http://regrep.org/gci/purchaseOrder/po.xsd"
eb:version="2.0"/> <eb:Description xml:lang="en-US">Purchase Order for 100,000
widgets</eb:Description> </eb:Reference></eb:Manifest>
```

- [468] 이하, 기반 프로토콜 구성에 있어서 메시지의 세부 구성 요소에 대하여 설명하면 다음과 같다.
- [469] MessageHeader 요소는 모든 ebXML 메시지에 표현되어야 하는 필수 요소로서 반드시 SOAP Header 요소의 자식 요소로 표현되어야 한다. MessageHeader 요소는 다음과 같은 하위 요소들로 구성된 복합 요소이다.
- [470] MessageHeader의 Element 구조는 아래 표 47과 같다.
- [471] 표 47

항목 명	설명			반복 회수	유형	길이
From	■ 메시지 송신 송수신 개체 정보			1..1		
	PartyId	type	■ 'ecf_cd'로 고정			
		■ 송신자를 식별하는 코드■ 유통 메시징 서버의 경우 인증번호 ■ 유통 클라이언트의 경우 자체 관리 번호 설정■ 주소 디렉터리서버, 유통 중계서버 경우는 개체 코드값 설정		1..1	S	13
	Role	■ 송신자 역할■ 'sender'로 고정		1..1	S	최대256
To	■ 메시지 수신 송수신 개체 정보			1..1		
	PartyId	type	■ 'ecf_cd'로 고정	1..1	S	13
		■ 수신자를 식별하는 코드■ 유통 메시징 서버의 경우 인증번호 ■ 유통 클라이언트의 경우 자체 관리 번호 설정■ 주소 디렉터리 서버, 유통 중계서버 경우는 개체 코드값 설정		1..1	S	13
	Role	■ 수신자 역할■ 'receiver'로 고정		1..1	S	최대256
CPAId	■ 거래협업정의서 아이디■ 연계인터페이스 유형에 따라서 코드값 설정			1..1	S	최대256
Conversati onId	■ 송수신 트랜잭션 구분자			1..1	S	최대256
Service	■ 거래협업정의서에 정의된 업무 서비스 구분자			1..1	S	최대 256
Action	■ Service 내의 특정 업무 프로세스 구분자■ Service 내에서 유일한 값			1..1	S	최대 256

MessageData	■ 메시지를 식별하기 위한 데이터	1..1		
	MessageId ■ 하나의 메시지가 가지는 유일한 식별자	1..1	S	최대 256
	Timestamp ■ 메시지 생성 시간 ■ UTC 형식 ■ ex> 2008-07-31T06:29:39.724Z	1..1	S	24
	RefToMessageId ■ 응답메시지에만 존재 ■ 요청메시지의 MessageId	0..1	S	최대 256
Other	■ 인터페이스 종류에 따른 확장 요소 ■ 요소명은 해당 인터페이스에 따라 다른 명칭을 가짐 ■ 세부 상세 내용은 해당 연계 인터페이스(5장, 6장, 7장, 8장) 참조	0..1		

[472] MessageHeader의 스키마 구조는 도 36과 같으며, MessageHeader 항목 코드표는 아래 표 48과 같고, 업무별 Service/Action 항목은 아래 표 49와 같다.

[473] 표 48

식별코드항목	코드값	코드값 정의
PartyId	ads	주소디렉터리 서버 개체 코드
	ech	유통 중계서버 개체 코드
CPAId	urn:ads-and-ecm-cpa	유통 메시징서버와 주소디렉터리 서버간 연계인터페이스 사용시
	urn:ecm-and-ecm-cpa	유통 메시징서버 상호간 연계인터페이스 사용시
	urn:ecm-and-ecm-cpa	유통 클라이언트와 유통 메시징서버 간 연계인터페이스 사용시
	urn:ech-and-ecm-cpa	유통 메시징서버와 유통 중계서버간 연계인터페이스 사용시

[474] 표 49

항목	Service	Action	정의
유통 메시징서버와 주소디렉터리 서버간 연계인터페이스 사용시	urn:ads-service	request	요청
		response	응답
유통 메시징서버 상호간 연계인터페이스 사용시	urn:ecm-service	request	요청
		response	응답
유통 클라이언트와 유통 메시징서버 간 연계인터페이스 사용시	urn:ecc-service	request	요청
		response	응답
유통 메시징서버와 유통 중계서버 간 연계인터페이스 사용시	urn:ech-service	request	요청
		response	응답

[475] MessageHeader 예시는 아래 표 50과 같다.

[476] 표 50

```
<eb:MessageHeader SOAP:mustUnderstand="1" eb:id="MessageHeader"
eb:version="2.0"> <eb:From> <eb:PartyId eb:type="ecf_cd">1234567</eb:PartyId>
<eb:Role>sender</eb:Role> </eb:From> <eb:To> <eb:PartyId
eb:type="ecf_cd">4567899</eb:PartyId> <eb:Role>receiver</eb:Role> </eb:To>
<eb:CPAId>urn:ecm-and-ecm-cpa</eb:CPAId><eb:ConversationId>urn:ecm-and-ec
m-cpa:0210050643</eb:ConversationId> <eb:Service>urn:ecm-service</eb:Service>
<eb:Action>request</eb:Action>
<eb:MessageData><eb:MessageId>20110210-170644Z-00057@127.0.0.18d1f96bf-9
cd6-4049-9fdb-a6c0ed9af467</eb:MessageId>
<eb:Timestamp>2011-02-10T08:06:44.810Z</eb:Timestamp> </eb:MessageData>
<eb:DuplicateElimination></eb:DuplicateElimination></eb:MessageHeader>
```

[477] SyncReply가 존재한다면 동기식 송신임을 의미하며 다음 ① 내지 ④의 속성을 가진다.

[478] ① id 속성

[479] ② version 속성

[480] ③ SOAP actor 속성(반드시 "http://schemas.xmlsoap.org/soap/actor/next" 값을 가져야 함)

[481] ④ SOAP mustUnderstand 속성

[482] SyncReply 요소의 예제는 아래 표 51과 같다.

[483] 표 51

```
<eb:SyncReply eb:id="3833kkj9" eb:version="2.0" SOAP:mustUnderstand="1"
SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
```

- [484] 유통 연계 메시지는 송수신 과정에서 발생할 수 있는 다양한 위험 요소에 대응하기 위하여 반드시 전자적으로 서명되어야 한다. 따라서 Signature 요소가 SOAP Header의 자식요소로 반드시 존재해야 한다.
- [485] 유통 연계 메시지에서 전자서명의 대상이 되는 부분은 SOAP 메시지 전체와 페이로드 컨테이너에 포함된 메시지 및 첨부문서들이다. 각 서명대상 정보는 Digest되어 전자서명 정보 내에 각각 포함된다.
- [486] [XMLDSIG] 규격에 따라 전자서명을 수행하는 과정은 다음 ① 내지 ④와 같다.
- [487] ① SOAP Envelope에 SignatureMethod, CanonicalizationMethod, Reference 요소를 가진 SignedInfo 요소와 필수 페이로드 객체를 [XMLDSIG]에 규정된 대로 생성
- [488] - SignedInfo 하위의 첫 번째 Reference 항목은 SOAP 메시지 전체를 대상으로 하므로 URI 값에 ""를 기술한다.
- [489] - 두 번째 Reference 항목부터는 페이로드 컨테이너의 개수만큼 반복하여 기술하도록 하며, 이때 URI 값은 첨부문서의 MIME Header에 정의된 content ID 값을 기술한다. (Digest 대상은 Mime Header를 제외한 Content 부분임)
- [490] ② 정규화 한 뒤 [XMLDSIG]에 지정된 대로 SignedInfo에 지정된 알고리즘을 기준으로 SignedInfo의 SignatureValue를 산출
- [491] ③ [XMLDSIG]에 지정된 대로 SignedInfo, KeyInfo, SignatureValue 요소를 포함하는 Signature 요소를 생성
- [492] ④ SOAP Header의 Signature 요소를 SOAP Header 요소에 포함
- [493] 전자서명 시 사용되는 알고리즘 정보는 다음과 같다. 알고리즘은 W3C "XML-Signature Syntax and Processing" (RFC3275)의 알고리즘 부분(6.0 Algorithms)을 기본적으로 따른다. 또한 국내 고유 알고리즘을 지원하기 위해 TTAS.IF-RFC3075 "확장성 생성 언어 전자서명 구문과 처리 (XML-Signature Syntax and Processing)" (한국정보통신기술협회, 2004년)에서 정의된 알고리즘을 이용한다.
- [494] 다음은 유통 프로토콜에서 이용하는 알고리즘 목록이다. 메시지 송수신 시 전자서명의 생성 및 검증 과정에서의 모호성을 최소화 하기 위해서 다음 ① 내지 ⑤ 목록 이외의 알고리즘은 사용을 제한한다.

[495] ① 전자서명 Namespace

[496] 표 52

```
<... xmlns:ds="http://www.w3.org/2000/09/xmldsig#" ... >
```

[497] ② 해쉬 (Digest)

[498] ;데이터를 축약하는 데 사용되는 알고리즘은 공인인증체계의 관련 규정을 준수하도록 한다.

[499] 표 53

```
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
또는 <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256"/>
```

[500] ③ 전자서명(Signature)

[501] ;메시지 전자서명 시 사용되는 알고리즘은 공인인증체계의 관련 규정을 준수하도록 한다.

[502] 표 54

```
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>또는<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
```

[503] ④ 정규화(Canonicalization)

[504] ;논리적으로 동일한 문서에 대해 물리적으로 여러 가지 표현이 가능 방식한 XML의 특성으로 인해 같은 문서에 대해 전자서명 값이 다르게 나올 수 있다. 이러한 현상을 방지하기 위해 반드시 정규화 과정을 거쳐야 한다. 정규화는 주석 없는 정규 XML(Canonical XML, omits comments)을 사용하도록 한다.

[505] 표 55

```
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```

[506] ⑤ 변환(Transform)

[507] ;전체 XML 데이터 중에서 실제 서명 대상이 되는 데이터를 가공하고 선택하는 과정을 거치는 알고리즘으로 다양한 변환 알고리즘이 존재하나 그 중에서 3가지만을 이용하도록 한다. 첫 번째는 전자서명이 서명 대상 내에 포함되는 형식을 따르므로 Enveloped Signature 변환이고, 두 번째는 상기 설명한 정규화(Canonicalization) 그리고 세 번째는 서명 대상 정보를 선택하는 XPath 필터링(XPath Filtering)이다.

[508] 표 56

```
<Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>및<ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>및<ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116"><ds:XPath>not(ancestor-or-self::node()[@SOAP:actor=&quot;urn:oasis:names:tc:ebxml-msg:actor:nextMSH&quot;]| ancestor-or-self::node()[@SOAP:actor=&quot;http://schemas.xmlsoap.org/soap/actor/next&quot;])</ds:XPath></ds:Transform>
```

[509] 전자서명 구문의 구조는 도 37과 같으며, 전자서명된 메시지 예제는 아래 표 57과 같다.

[510] 표 57

```

<?xml version="1.0" encoding="utf-8"?><SOAP:Envelope
xmlns:xlink="http://www.w3.org/1999/xlink"xmlns:SOAP="http://schemas.xmlsoap.
org/soap/envelope/"xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/sch
ema/msg-header-2_0.xsd"xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
"xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/http://www.oasis-o
pen.org/committees/ebxml-msg/schema/envelope.xsdhttp://www.oasis-open.org/com
mittees/ebxml-msg/schema/msg-header-2_0.xsdhttp://www.oasis-open.org/committee
s/ebxml-msg/schema/msg-header-2_0.xsd"> <SOAP:Header> <eb:MessageHeader
eb:id="..." eb:version="2.0" SOAP:mustUnderstand="1"> ... </eb:MessageHeader>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#"> <SignedInfo>
<CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/> <Reference
URI=""> <Transforms> <Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
<XPath> not(ancestor-or-self::node())[ @SOAP:actor=
&quot;urn:oasis:names:tc:ebxml-msg:actor:nextMSH&quot;] |
ancestor-or-self::node()[ @SOAP:actor=
"http://schemas.xmlsoap.org/soap/actor/next"] ) </XPath> </Transform> <Transform
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
</Transforms> <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>...</DigestValue> </Reference> <Reference
URI="cid://blahblahblah/"> <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>...</DigestValue> </Reference> </SignedInfo>
<SignatureValue>...</SignatureValue> <KeyInfo>...</KeyInfo> </Signature>
</SOAP:Header> <SOAP:Body> <eb:Manifest eb:id="Mani01" eb:version="2.0">
<eb:Reference xlink:href="cid://blahblahblah/"
xlink:role="http://ebxml.org/gci/invoice"> <eb:Schema eb:version="2.0"
eb:location="http://ebxml.org/gci/busdocs/invoice.dtd"/> </eb:Reference>
</eb:Manifest> </SOAP:Body></SOAP:Envelope>

```

- [511] ErrorList 요소는 메시지 구문 검증, 메시지 전자서명 검증 등 통신 프로토콜 처리 과정에서 오류가 발생하는 경우에 Header 하위에 생성하여 송신자에게 동기식으로 반환해야 한다. ErrorList 요소가 생성되는 경우에는 반드시

MessageHeader 요소 내에 RefToMessageId가 존재해야 하며 RefToMessageId는 오류가 발생한 메시지의 MessageId를 지칭해야 한다.

[512] ErrorList 요소는 다음 ① 내지 ⑤와 같은 속성들을 가진다.

[513] ① id 속성

[514] ② SOAP mustUnderstand 속성

[515] ③ version 속성

[516] ④ highestSeverity 속성

[517] ⑤ 한 개 이상의 Error 요소

[518] 보고될 오류가 없으면 ErrorList 요소는 존재해서는 안 된다. ErrorList의 구조는 도 38과 같다.

[519] highestSeverity 속성은 모든 Error 요소들의 가장 심각한 상태를 표시한다. 특히, 어떤 Error 요소가 severity를 Error 라고 설정되어 있으면, highestSeverity는 Error로 설정되어야 하며, 그렇지 않은 경우에는 highestSeverity를 Warning으로 설정해야 한다.

[520] Error 요소는 다음 ① 내지 ⑥과 같은 속성들을 가진다.

[521] ① id 속성

[522] ; id 속성은 문서 내에서 ErrorList 요소를 유일하게 식별하는 역할을 한다.

[523] ② codeContext 속성

[524] ; codeContext 속성은 errorCodes의 네임스페이스 또는 스키마를 나타낸다. 이는 반드시 URI이어야 한다. 이 속성의 기본값은 urn:oasis:names:tc:ebxml-msg:service:errors이다. 이 속성에 기본 값이 없으면, 그 명세의 구현은 errorCodes를 사용한다는 것을 가리킨다.

[525] ③ errorCode 속성

[526] ; 필수 errorCode 속성은 오류를 가지고 있는 메시지의 오류가 지닌 본질을 지시한다.

[527] ④ severity 속성

[528] ; 필수 속성인 severity 속성은 오류의 심각성을 나타냅니다. 유효한 값들은 Warning 및 Error와 같다. Warning는 오류가 존재하지만 대화 중의 다른 메시지들은 정상적으로 생성되고 있다는 것을 가리키며, Error는 복구 불가능한 오류가 메시지에 존재하고 대화 중 더 이상 다른 메시지들은 생성되지 않는다는 것을 가리킨다.

[529] ⑤ location 속성

[530] ; location 속성은 오류가 존재하는 메시지 부분을 가리킨다. 만약에 오류가 ebXML 요소 내에 존재하고 요소가 "well-formed"라면 location 속성의 내용은 [Xpointer]이어야 한다.

[531] ⑥ Description 속성

[532] ; Description 요소의 내용은 xml:lang 속성에서 정의된 언어로 오류의 서술적인 설명을 제공한다. 보통, 이것은 XML 파서나 메시지를 검증하는 소프트웨어가

생성한 메시지가 된다. 이 의미는 이 내용은 Error 요소를 생성했던 소프트웨어의 판매자나 개발자에 의해 정의된다는 것을 의미한다.

[533] ErrorList의 예제는 아래 표 58과 같다.

[534] 표 58

```
<eb:ErrorList eb:id="3490sdo", eb:highestSeverity="error" eb:version="2.0"
SOAP:mustUnderstand="1"> <eb:Error eb:errorCode="SecurityFailure"
eb:severity="Error" eb:location="URI_of_ds:Signature"> <eb:Description
xml:lang="en-US">Validation of signature failed<eb:Description> </eb:Error>
<eb:Error ...> ... </eb:Error></eb:ErrorList>
```

[535] 유통 프로토콜을 기반으로 메시지를 송수신하는 과정에서 오류가 발생하면, 오류를 인지한 송수신 개체는 상대방에게 오류 내용을 보고해야 한다. 보고해야 할 오류는 메시지 구조 오류, 메시징 오류 및 보안 오류와 같다.

[536] 본 발명에서 정의하는 유통 프로토콜보다 하위 레이어에 속하는 HTTP 및 Socket과 같은 데이터 통신 프로토콜과 관련된 오류들은 데이터 통신 프로토콜에서 지원하는 표준 메커니즘에 의해 발견되고 보고되어야 하며, 본 발명에서 정의하는 오류 보고 메커니즘을 사용하지 않는다.

[537] 오류코드는 오류 대상 및 유형별로 구분되며 자세한 내용은 아래 표 59와 같다.

[538] 표 59

오류 코드	내용	상세 설명
ValueNotRecognized	요소 내용이나 속성 값이 인식되지 않는다.	비록 문서가 well formed이고 유효하지만 요소/속성의 값이 인식할 수 없는 값이고 그러므로 ebXML 메시지 서비스에 의해 사용되어질 수 없는 값을 포함한다.
NotSupported	요소나 속성이 지원되지 않는다.	비록 문서가 well formed이고 유효하고 요소나 속성이 이 명세의 규칙이나 제약을 따르고 있지만 메시지를 처리할 수 있는 ebXML 메시지 서비스에 의해 지원되지 않는다.
Inconsistent	요소 내용이나 속성 값이 또 다른 요소나 속성에 불일치한다.	비록 문서가 well formed이고 유효하고 이 명세의 규칙과 제약을 따르지만 요소와 속성의 내용이 다른 요소나 그들의 속성에 일치하지 않는다.
OtherXml	요소 내용이나 속성 값 속에서의 또 다른 오류	비록 문서가 well formed이고 유효하지만 그 요소 내용이나 속성 값이 이 명세 내의 규칙과 제약에 따르지 않고 다른 오류 코드에 속하지 않는다. Error 요소의 내용은 문제의 본질을 가리키는데 사용되어야 한다.
DeliveryFailure	메시지 전송 실패	수신된 메시지가 대개 혹은 확실히 다음 목적지에 보내지지 못했다. 만약에 severity가 Warning으로 설정되어 있으면 메시지가 배달될 가능성은 적다.
TimeToLiveExpired	메시지가 존재할 수 있는 시간이 초과됨	메시지가 수신되었으나 MessageHeader 요소의 TimeToLive 요소가 제약한 시간을 초과한 시각에 수신되었다.
SecurityFailure	메시지의 보안 검사에 실패	메시지를 보낸 당사자의 서명의 검증 또는 권한 또는 실명 검사에 실패했다.
Unknown	알 수 없는 오류	어떤 오류 종류에도 속하지 않는 오류가 발생함을 뜻한다. Error 요소의 내용이 문제의 본질을 가리키는데 사용되어야 한다.

[539] 이하, HTTP 바인딩 방안에 있어서 HTTP를 통한 메시지 전송 방안에 대하여 설명하면 다음과 같다.

[540] HTTP 바인딩 예제는 아래 표 60과 같다.

[541] 표 60

```

POST /servlet/ebXMLhandler HTTP/1.1Host: www.example2.comSOAPAction:
"ebXML"Content-type: multipart/related; boundary="Boundary";
type="text/xml";start="<ebxhmheader111@example.com>"--BoundaryContent-ID:
<ebxhmheader111@example.com>Content-Type: text/xml<?xml version="1.0"
encoding="UTF-8"?><SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"xmlns:eb="http://www.oa
sis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<SOAP:Header> <eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
<eb:From> <eb:PartyId eb:type="ecf_cd">123456789</eb:PartyId>
<eb:Role>sender</eb:Role> </eb:From> <eb:To> <eb:PartyId
eb:type="ecf_cd">912345678</eb:PartyId> <eb:Role>receiver</eb:Role> </eb:To>
<eb:CPAId>urn:ecm-and-ecm-cpa</eb:CPAId>
<eb:ConversationId>20001209-133003-28572</eb:ConversationId>
<eb:Service>>urn:ecm-service</eb:Service> <eb:Action>request</eb:Action>
<eb:MessageData>
<eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
<eb:Timestamp>2001-02-15T11:12:12.724Z</eb:Timestamp> </eb:MessageData>
</eb:MessageHeader> </SOAP:Header> <SOAP:Body> <eb:Manifest
eb:version="2.0"> <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
xlink:role="XLinkRole" xlink:type="simple"> <eb:Description
xml:lang="en-US">Purchase Order 1</eb:Description> </eb:Reference>
</eb:Manifest> </SOAP:Body></SOAP:Envelope>--BoundaryContent-ID:
<ebxmlpayload111@example.com>Content-Type: text/xml<?xml version="1.0"
encoding="UTF-8"?><purchase_order> <po_number>1</po_number>
<part_number>123</part_number> <price
currency="USD">500.00</price></purchase_order>--Boundary--

```

[542] 이하, HTTP 바인딩 방안에 있어서 **HTTP Response 코드**에 대하여 설명하면 다음과 같다.

[543] 본 발명에서 HTTP 레벨의 응답 코드를 반환하기 위해서 [RFC2616]에 정의된 HTTP 응답 코드를 이용해야 한다. 주요 응답 코드는 아래 표 61과 같다.

[544] 표 61

상태 코드	연관 메시지	의미
200	OK	요청이 성공적으로 처리됨
400	Bad Request	요청에 문법적으로 잘못된 부분이 있음
401	Unauthorized	클라이언트가 올바른 허가를 받지 않고 허가가 필요한 페이지에 접근하려 함
404	Not Found	이 주소에서는 어떤 내용도 발견할 수 없음
500	Internal Server Error	서버 내부의 오류로 인해 요청을 정상적으로 처리할 수 없음
503	Service Unavailable	처리할 수 있는 한계를 벗어나 과도하게 요청이 들어와서 서버가 현재 해당 요청을 처리할 수 없음

- [545] 이하, HTTP 바인딩 방안에 있어서 HTTP 전송 보안 방안에 대하여 설명하면 다음과 같다.
- [546] 유통체계 내의 모든 유통 메시징서버와 유통메시지 서버 간 전송이나 유통 메시징 서버와 유통 클라이언트 간 전송은 네트워크 전송 보안을 위해서 반드시 SSL(Secure Socket Layer) V3.0을 이용한 HTTP/S(Secure Hypertext Transfer Protocol)를 사용하여 처리하여야 한다.
- [547] 본 발명에 따른 유통체계에서 오류 발생 유형은 크게 동기식 응답에 대한 오류 발생의 경우와 비동기식 응답에 대한 오류 발생 경우로 구분된다.
- [548] 동기식 응답에 대한 오류의 경우는 요청메시지에 대한 처리결과를 받을 때까지 처음 요청자는 대기하고 있는 상태이므로 오류를 즉각 인지할 수 있으나, 비동기식 응답에 대한 오류는 요청자는 요청 내용만 전달한 후 그에 대한 처리결과를 차후에 받게 되므로 추가적인 오류처리가 이루어져야 한다.
- [549] 이하, 오류 처리 방안에 있어서 동기식 오류처리 방안에 대하여 설명하면 다음과 같다.
- [550] 유통 메시징서버와 타 유통 메시징서버, 주소 디렉터리 서버, 유통 클라이언트, 유통 중계서버 간의 두 개체 간 모든 메시지 유통은 동기식 방식의 유통이다. 이외에도, 유통 클라이언트가 유통 메시징서버를 통해 타 유통 메시징서버, 주소 디렉터리 서버, 유통 중계서버와 연계되는 3 개체 간 메시지 유통은 연계 유형에 따라 동기식 또는 비동기식 방식으로 연계된다.
- [551] 동기식 전송에 대한 모든 오류는 전송자가 즉각 확인 가능하므로 재전송하는 것을 기본으로 한다. 재전송 방식에 대해서는 유통체계에 참여하는 기업이나 기관의 시스템의 정책에 따라 결정되나, 동일한 메시지 전송에 대해서는 동일한 MessageId 값을 설정하여 다시 보내는 것을 기본으로 한다.
- [552] 동일한 MessageId 값으로 메시지를 보냄으로써 메시지 전송과정의 오류가

전송시점의 오류가 아니라 수신자가 수신 성공 후 동기식으로 응답메시지를 전송하는 과정에서의 오류에 발행한 경우에 대해서도 중복메시지 수신을 감지함으로써 동일한 요청을 중복해서 처리하는 것을 방지한다.

- [553] 동기식 오류의 송신자와 수신자는 각각 연계유형에 따라 유통 메시징서버, 주소 디렉터리 서버, 유통 클라이언트, 유통 중계서버가 될 수 있다.
- [554] ① 요청메시지 송신실패: 송신자가 메시지를 전송하는 과정에서 전송오류가 발생하여 수신자에게 요청메시지가 전달되지 못하는 경우로서, 송신자는 전송시도에 대한 timeout 또는 네트워크 오류 메시지 등을 통해 송신 실패를 인지하게 된다. 도 39에는 요청메시지 송신 실패 시의 프로세스를 도시하였으며, 처리 절차는 아래 1) 내지 3)과 같다.
- [555] 1) 메시지 송신자가 전송하는 과정에서 전송오류가 발생하는 경우로서 많은 경우가 네트워크 오류에 의해 발생됨
- [556] 2) 송신자는 HTTP 오류와 같은 오류 메시지를 받게 되면 동일한 메시지를 다시 재전송 요청하여야 함
- [557] 3) 송신자는 수신자에게 수신확인 메시지를 받은 경우에만 전송성공으로 인식함
- [558] ② 응답메시지 수신실패: 송신자가 메시지를 정상적으로 전송하였으나 수신자로부터 응답메시지를 받는 과정에서 오류가 발생한 경우이다. 송신자 입장에서는 "①요청메시지 송신실패"와 구분이 되지 않으므로 오류에 대해서 동일한 방식으로 처리하나, 수신자는 요청메시지를 정상적으로 받았기 때문에 처리방식이 다르다. 도 40에는 응답메시지 수신실패와 관련한 프로세스를 도시하였으며, 처리 절차는 아래 1) 내지 3)과 같다.
- [559] 1) 메시지가 수신자에게 정상적으로 전달되었으나, 송신자가 수신자로부터 수신확인 메시지를 받지 못한 경우
- [560] 2) 이 경우 송신자는 송신실패 오류로 인식하고 수신자에게 동일한 메시지를 동일한 MessageId로 재전송하게 됨
- [561] 3) 수신자는 수신한 문서의 MessageId가 이전 수신 메시지와 동일한 경우에는 중복수신으로 수신확인 메시지를 보낸 후 내부처리
- [562] ③ 오류메시지 수신: 송신자가 메시지를 정상적으로 전송하였으나 전송한 메시지를 수신한 수신자가 메시지를 처리하는 과정에서 오류가 발생한 경우이다. 이 경우 송신자의 처리 방식은 오류메시지의 유형에 따라 다르다. 통신 프로토콜 상의 오류유형은 상술한 "ErrorList" 항목을, 각 연계 인터페이스 별로 요청메시지에 대한 내부 처리과정에서 발생한 오류에 대해서는 각 연계 인터페이스의 메시지구조를 참조한다. 도 41에는 오류메시지 수신과 관련한 프로세스에 대하여 도시하였으며, 처리 절차는 아래 1) 내지 3)과 같다.
- [563] 1) 송신자가 수신자에게 전송한 메시지가 정확히 전달되었으나 전송메시지 자체에 오류가 있어서 오류메시지를 응답받은 경우
- [564] 2) 이 경우 송신자는 요청 메시지를 재생성한 후 메시지를 재전송하는 것이

일반적이나 오류유형에 따라 메시지 처리를 달리할 수 있음

- [565] 3) 송신자가 요청메시지를 재전송할 때, 전송하는 메시지의 MessageId는 동일할 필요는 없으며, 업무 상황에 따라 다르게 처리할 수 있음
- [566] ④ 3단계 동기식 오류: 유통 클라이언트가 유통 메시징서버를 통해 타 유통 메시징서버, 주소 디렉터리서버, 유통 중계서버와 연계되는 3 개체 간 메시지 유통은 연계 유형 중 최종적 결과를 즉각적으로 확인하기 위해 동기식으로 연계하는 방안을 지원한다. 이 과정 중 유통 메시징서버와 수신자 간 연계단계에서 오류가 발생하면 유통 메시징서버는 즉각적으로 오류를 발생시킨 후 이를 유통 메시징서버에 응답메시지로 전달하여야 한다. 도 42에는 3단계 동기식 오류와 관련한 프로세스에 대하여 도시하였으며, 처리 절차는 아래 1) 내지 3)과 같다.
- [567] 1) 유통 클라이언트가 유통 메시징서버와 연계하여 메시지 전송을 하는 단계에서는 전송성공을 하였으나, 유통 메시징서버 다음 수신자(주소 디렉터리서버, 타 유통 메시징서버, 유통 중계서버 등)에게 전송하는 과정에서 오류가 발생함
- [568] 2) 이 때 오류는 유통 메시징서버와 수신자간 동기식 전송에서 발생하는 모든 오류인 경우를 지칭함
- [569] 3) 유통 메시징서버는 오류를 인지한 시점에 유통 클라이언트를 위한 오류 메시지를 발생시키고 이를 유통 클라이언트에 응답메시지로 전달함
- [570] 유통 메시징서버가 생성하는 오류 메시지는 아래 표 62와 같은 구조로 구성된다.
- [571] 표 62

항목 명	설명	반복 회수	유형	길이
Content	■ Root 엘리먼트			
DocType	■ 유통메시지 유형 - 오류: 9	1..1	Integer	1
Sender	■ 요청메시지의 수신자 공인전자주소	1..1	String	최대 128
Receiver	■ 요청메시지의 송신자 공인전자주소	1..1	String	최대 128
RefIdentifier	■ 요청메시지의 고유 식별값	1..1	String	36
Identifier	■ UUID 형식으로 생성한 오류 메시지의 고유 식별 값	1..1	String	36
ErrorCode	■ 해당 오류 코드	1..1	Integer	4

- [572] 이하, 오류 처리 방안에 있어서 비동기식 오류처리 방안에 대하여 설명하면

다음과 같다.

- [573] 유통 클라이언트가 유통 메시징서버를 통해 다 유통 메시징서버, 주소 디렉터리서버, 유통 중계서버와 연계되는 3 개체 간 메시지 유통은 연계 유통 중 유통 클라이언트 사용자가 최종 수신자의 상황에 독립적으로 운영할 수 있도록 비동기식 방식의 연계를 지원하기도 한다.
- [574] 비동기식 전송에 대한 최종 오류는 동기식 전송과 달리, 전송자가 즉각 확인할 수가 없으므로 유통 메시징서버가 최종적으로 오류를 확인한 시점에 유통 클라이언트를 위한 오류 메시지를 발생시키고, 이를 유통 클라이언트가 수신할 수 있도록 한다.
- [575] 도 43에는 비동기식 오류처리 방안과 관련한 프로세스를 도시하였으며, 처리 방안은 아래 1) 내지 4)와 같다.
- [576] 1) 유통 클라이언트가 유통 메시징서버와 연계하여 메시지 전송을 하는 단계에서는 전송성공을 하였으나, 유통 메시징서버 다음 수신자(주소 디렉터리서버, 다 유통 메시징서버, 유통 중계서버 등)에게 전송하는 과정에서 오류가 발생함
- [577] 2) 이때 오류는 유통 메시징서버와 수신자간 동기식 전송에서 발생되는 모든 오류인 경우를 지칭함
- [578] 3) 유통 메시징서버는 재시도 이후, 최종적으로 오류를 인지한 시점에 유통 클라이언트를 위한 오류 메시지를 발생시킨 후 유통 클라이언트의 수신함에 전달함
- [579] 4) 유통 클라이언트가 유통 메시징서버에 수신메시지를 요청하는 단계에서 자신의 수신함에 수신된 오류메시지를 통해 이전 요청메시지에 대한 오류를 인지함
- [580] 유통 메시징서버가 생성하는 오류 메시지는 아래 표63과 같은 구조로 구성된다.
- [581] 표 63

항목명	설명	반복 회수	유형	길이	
Content	■ Root 엘리먼트				
	DocType	■ 유통메시지 유형 - 오류: 9	1..1	Integer	1
	Sender	■ 요청메시지의 수신자 공인전자주소	1..1	String	최대 128
	Receiver	■ 요청메시지의 송신자 공인전자주소	1..1	String	최대 128
	RefIdentifier	■ 요청메시지의 고유 식별값	1..1	String	36
	Identifier	■ UUID 형식으로 생성한 오류 메시지의 고유 식별 값	1..1	String	36
	ErrorCode	■ 해당 오류 코드	1..1	Integer	4

[582]

[유통 메시징서버와 주소 디렉터리 서버 간 연계 인터페이스]

[583]

주소 디렉터리 서버는 유통체계에서 가장 기본이 되는 공인전자주소 정보를 관리하고 있는 시스템으로 전자문서유통에서 반드시 필요한 시스템이다.

[584]

[585]

유통 메시징서버와 주소 디렉터리서버 간 연계 인터페이스는 크게 2가지로 기능이 구분된다. 첫번째는 등록대행기관과의 공인전자주소 등록 등 업무에 관한 인터페이스이며, 두번째는 유통 메시징서버와의 물리적 주소 질의/응답, 스캔 신고 등 업무에 관한 인터페이스이다.

[586]

위 등록대행기관과의 공인전자주소 등록 등 업무에 관한 인터페이스는 따로 구분할 수 있지만, 등록대행기관을 전자문서중계자 또는 제3자 보관기관 사업자가 하기 때문에 유통 메시징서버 내에 인터페이스 기능을 삽입하였다. 단, 송수신 개체에 설치되는 유통 메시징서버에는 공인전자주소 등록 등과 관련된 연계 인터페이스는 들어가지 않는다.

[587]

[588]

유통 메시징서버와 주소 디렉터리 서버 간 인터페이스 기능은 아래 표 64와 같다.

[589]

표 64

인터페이스 구분		인터페이스 설명	비고
주소 관리	공인전자주소 등록(공인 송수신자의 주소를 등록하는 경우)	☐ 전자문서 중계자를 통해 문서를 송수신하는 공인 송·수신자의 공인전자주소 정보를 주소 디렉터리서버에 등록하기 위한 인터페이스 ☐ 요청한 공인전자주소가 주소 디렉터리서버에서 unique하지 않은 경우에는 등록에 실패함	요청자는 전자문서 중계자임
	공인전자주소 정보 변경	☐ 공인전자주소에 대한 관련정보들(ex: 보안정보, ID에 대해 변경이 발생한 경우 주소 디렉터리서버에 변경요청한 후 그 결과를 받는 인터페이스	
	공인전자주소 삭제	☐ 주소 디렉터리서버에 등록된 공인전자주소를 더 이상 사용하지 않는 경우 주소 디렉터리서버에 삭제요청을 한 후 그 결과를 받는 인터페이스	
주소 검색	물리적 주소정보 검색	☐ 주소 디렉터리서버에 공인전자주소 정보에 해당하는 사용자의 보안정보(공인인증서)와 물리적 주소정보를 검색요청한 후 그 결과를 받는 인터페이스	요청자는 전자문서 중계자와 송수신 개체임
블랙리스 트 관리	스팸메시지 신고	☐ 주소 디렉터리서버에 스팸메시지를 신고하고 접수여부를 결과로 받는 인터페이스 ☐ 주소 디렉터리서버는 스팸 신고된 메시지에 대한 최종 처리결과(스팸메시지로 확정되었는지 여부)를 신고자 및 스팸발송자에게 "메시지 전송 인터페이스"를 사용하여 통지함	

통보	화이트 리스트 통보	☐ 주소 디렉터리서버에서 송수신 개체로 화이트리스트 전달하는 인터페이스	
	블랙 리스트 통보	☐ 주소 디렉터리서버에서 송수신 개체로 블랙리스트 전달하는 인터페이스	

- [590] 이와 같은 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스의 상세 내용에 대하여 설명하면 다음과 같다.
- [591] 먼저, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서 공통 사항에 대하여 설명하면 다음 ① 및 ②와 같다.
- [592] ① 요청메시지의 MessageHeader 확장
- [593] 송신 개체의 유통 메시징서버가 주소 디렉터리서버에 송신하는 요청메시지의 첫 번째 MIME Part의 SOAP 메시지 내에는 송신 개체의 전자서명정보가 포함되어 전달되어야 하며, 주소 디렉터리서버가 SOAP 메시지의 전자서명에 사용된 인증서의 소유자가 해당 송신 개체와 일치하는지를 검증(VID 검증)하는데 필요한 추가적인 송신 개체의 정보(CorpNum, RValue) 또한 포함되어 전달되어야 한다.
- [594] 추가적인 송신 개체의 정보는 요청메시지의 SOAP 메시지 내 MessageHeader 요소 하위에 확장 요소(any ##other 위치)로서 위치해야 한다.
- [595] 확장 요소 구조는 아래 표 65와 같으며, 확장 요소 예시는 아래 표 66과 같다.
- [596] 표 65

항목 명	설명	반복 회수	유형	길이
Extension	■ 확장 요소 엘리먼트	1..1		
CorpNum	■ 중계자 혹은 송수신개체 사업자등록번호	1..1	String	10
RValue	■ 중계자 혹은 송수신개체 공인인증서 개인키에서 추출한 RValue ■ RValue를 Base64로 인코딩 하여 입력해야 함	1..1	String	28

- [597] 표 66

```

<eb:MessageHeader SOAP:mustUnderstand="1" eb:id="MessageHeader"
eb:version="2.0"> <eb:From> <eb:PartyId
eb:type="ecf_cd">123456789</eb:PartyId> <eb:Role>sender</eb:Role> </eb:From>
<eb:To> <eb:PartyId eb:type="ecf_cd">ads</eb:PartyId>
<eb:Role>receiver</eb:Role> </eb:To>
<eb:CPAId>urn:ads-and-ecm-cpa</eb:CPAId>
<eb:ConversationId>20001209-133003-28572</eb:ConversationId>
<eb:Service>>urn:ads-service</eb:Service> <eb:Action>request</eb:Action>
<eb:MessageData><eb:MessageId>20110210-170644Z-00057@127.0.0.18d1f96bf-9
cd6-4049-9fdb-a6c0ed9af46 7</eb:MessageId>
<eb:Timestamp>2011-02-10T08:06:44.810Z</eb:Timestamp> </eb:MessageData>
<eb:DuplicateElimination></eb:DuplicateElimination> <Extention>
<CorpNum>2208203228</CorpNum> <RValue>asdfasdf</RValue>
</Extention></eb:MessageHeader>

```

- [598] ㉔ 전체 메시지 구조
- [599] 유통 메시징서버와 주소 디렉터리서버 간 연계 인터페이스는 메시지의 첫 번째 MIME Part에는 SOAP 메시지가 위치하고, 두 번째 MIME Part에는 해당 요청 및 응답에 대한 유통메시지가 위치한다.
- [600] 유통 메세징서버와 주소 디렉터리서버 간 SOAP 구조는 도 44와 같다.
- [601] 이하, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서 공인전자주소 등록에 대하여 설명하면 다음과 같다.
- [602] 공인전자주소 등록처리와 관련한 메시지 교환 흐름은 도 45와 같다.
- [603] 요청 유통메시지 구조는 아래 표 67과 같으며, 메시지 예제는 아래 표 68과 같다.
- [604] 표 67

항목 명	설명	반복회수	유형	길이
Request	■ 요청 Root 엘리먼트			
Request	■ 공인 송수신자 공인전자주소 등록 요청 엘리먼트			
	PeerCorpNum	■ 송수신 개체의 사업자 등록번호	1..1	String
	PeerRegNum	■ 송수신 개체의 인증번호	0..1	String
	Name	■ 회원 명	1..1	String
	Type	■ 회원 유형 - 개인: U - 사업자: C	1..1	String
	IDN	■ 회원 식별번호 - 개인: 주민등록번호 - 사업자: 사업자등록번호	1..1	String
	RAddress	■ 공인전자주소	1..1	String
	Cert	■ 공인인증서	0..1	Base64
	Representative	■ 사업자인 경우 대표자명	0..1	String
	Addr	■ 개인 또는 사업자 주소	0..1	String
	Tel	■ 개인 또는 사업자 전화번호(- 없이 입력)	0..1	Integer
	Fax	■ 개인 또는 사업자 팩스 번호	0..1	Integer
	Mobile	■ 개인 또는 사업자 휴대폰 번호(- 없이 입력)	0..1	Integer
	EMail	■ 개인 또는 사업자 이메일	0..1	String
	RegDate	■ 공인전자주소 등록일	0..1	Long
	EndDate	■ 공인전자주소 만료일	0..1	Long
	ManagerName	■ 공인전자주소 책임자 명	0..1	String
	ManagerAddr	■ 공인전자주소 책임자 주소	0..1	String
	ManagerEMail	■ 공인전자주소 책임자 이메일	0..1	String

ManagerTel	■ 공인전자주소 책임자 전화번호	0..1	Integer
ManagerMobile	■ 공인전자주소 책임자 휴대폰 번호	0..1	Integer

[605] 표 68

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation"> <RegAddReq>
<PeerCorpNum>1234567890</PeerCorpNum>
<PeerRegNum>5555555555</PeerRegNum> <Name>홍길동</Name>
<Type>U</Type> <IDN>111111222222</IDN>
<RAddress>#000-0000-0000</RAddress> <Cert>MIDJAHjhh46dhkfsjsfj...</Cert>
<Addr>서울시</Addr> <Tel>021113333</Tel> <Mobile>0101112222</Mobile>
</RegAddReq></Request>

```

[606] 응답 유통메시지 구조는 아래 표 69와 같으며, 메시지 예제는 아래 표 70과 같다.

[607] 표 69

항목명	설명	반복회수	유형	길이
Response	■ 응답 Root 엘리먼트			
RegistrationResponse	■ 등록대행기관(전자문서중계자) 회원 등록 응답 엘리먼트			
	ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean
	ErrorCode	■ 오류 코드(처리 결과가 실패(0)인 경우에만 해당 오류 코드 입력)	0..1	String

[608] 표 70

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation"> <RegAddRes>
<ResultCode>1</ResultCode> </RegAddRes></Response>

```

[609] 이하, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서 공인전자주소 정보 변경 인터페이스에 대하여 설명하면 다음과 같다.

[610] 공인전자주소 정보 변경 인터페이스는 전자문서 중계자가 주소 디렉터리서버에 등록된 공인 송수신자의 공인전자주소 정보에 대한 변경을

요청하여 응답을 받는 인터페이스로서, 변경하고자 하는 사용자 정보 및 공인전자주소 정보를 요청메시지에 포함하여 전송한 후, 주소 디렉터리서버의 변경 처리 결과를 응답메시지로 수신받는다.

[611] 공인전자주소 정보 변경처리와 관련한 메시지 교환 흐름은 도 46과 같다.

[612] 요청 유통메시지 구조는 아래 표 71과 같으며, 메시지 예제는 아래 표 72와 같다.

[613] 표 71

항목 명	설명	반복회수	유형	길이
Request	■ 요청 Root 엘리먼트			
Mod A dd rR eq	■ 공인 송수신자 공인전자주소 등록 요청 엘리먼트			
	PeerCorpNum	■ 송수신 개체의 사업자 등록번호	1..1	String
	PeerRegNum	■ 송수신 개체의 인증번호	0..1	String
	Name	■ 회원 명	0..1	String
	Type	■ 회원 유형 - 개인: U - 사업자: C	0..1	String
	IDN	■ 회원 식별번호 - 개인: 주민등록번호 - 사업자: 사업자등록번호	0..1	String
	RAddress	■ 공인전자주소	1..1	String
	Cert	■ 공인인증서	0..1	Base64
	Representative	■ 사업자인 경우 대표자명	0..1	String
	Addr	■ 개인 또는 사업자 주소	0..1	String
	Tel	■ 개인 또는 사업자 전화번호(- 없이 입력)	0..1	Integer
	Fax	■ 개인 또는 사업자 팩스 번호	0..1	Integer
	Mobile	■ 개인 또는 사업자 휴대폰 번호(- 없이 입력)	0..1	Integer
	EMail	■ 개인 또는 사업자 이메일	0..1	String
	RegDate	■ 공인전자주소 등록일	0..1	Long
	EndDate	■ 공인전자주소 만료일	0..1	Long
	ManagerName	■ 공인전자주소 책임자 명	0..1	String
	ManagerAddr	■ 공인전자주소 책임자 주소	0..1	String
	ManagerEMail	■ 공인전자주소 책임자 이메일	0..1	String

ManagerTel	■ 공인전자주소 책임자 전화번호	0..1	Integer
ManagerMobile	■ 공인전자주소 책임자 휴대폰 번호	0..1	Integer

[614] 표 72

```
<Request xmlns="http://www.nipa.kr/eDocument_Circulation"> <ModAddrReq>
<PeerCorpNum>1234567890</PeerCorpNum>
<PeerRegNum>5555555555</PeerRegNum> <Name>홍길동</Name>
<Type>U</Type> <IDN>111112222222</IDN>
<RAddress>#000-0000-0000</RAddress> <Cert>MIDJAHjhh46dhkfsjsfsj...</Cert>
<Addr>서울시</Addr> <Tel>021113333</Tel> <Mobile>0101112222</Mobile>
</ModAddrReq></Request>
```

[615] 응답 유통메시지 구조는 아래 표 73과 같으며, 메시지 예제는 아래 표 74와 같다.

[616] 표 73

항목 명	설명	반복회수	유형	길이
Response	■ 응답 Root 엘리먼트			
ModAddrReq	■ 전자문서 중계자 회원 수정 응답 엘리먼트			
	ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean
	ErrorCode	■ 오류 코드(처리 결과가 실패(0)인 경우에만 해당 오류 코드 입력)	0..1	String

[617] * 도 73에서, ErrorCode는 ResultCode가 실패(0)로 입력된 경우, 오류 원인에 해당하는 오류 코드를 입력

[618] 표 74

```
<Response xmlns="http://www.nipa.kr/eDocument_Circulation"> <ModAddrRes>
<ResultCode>1</ResultCode> </ModAddrRes></Response>
```

[619] 이하, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서

공인전자주소 삭제 인터페이스에 대하여 설명하면 다음과 같다.

[620] 공인전자주소 삭제 인터페이스는 전자문서 중계자가 주소 디렉터리서버에 등록된 공인 송수신자의 공인전자주소 정보의 공인전자주소 정보에 대한 삭제를 요청하여 응답을 받는 인터페이스로서, 삭제하고자 하는 사용자 정보 및 공인전자주소 정보를 요청메시지에 포함하여 전송한 후, 주소 디렉터리서버의 삭제 처리 결과를 응답메시지로 수신받는다.

[621] 공인전자주소 삭제처리와 관련한 메시지 교환 흐름은 도 47과 같다.

[622] 요청 유통메시지 구조는 아래 표 75과 같으며, 메시지 예제는 아래 표 76와 같다.

[623] 표 75

항목 명	설명	반복회수	유형	길이
Request	■ 요청 Root 엘리먼트			
DeIA dd rR eq	■ 회원 공인전자주소 삭제 요청 엘리먼트			
	Name	■ 회원 명	1..1	String
	IDN	■ 회원 식별번호 - 개인: 주민등록번호 - 사업자: 사업자등록번호	1..1	String
	RAddress	■ 공인전자주소	1..1	String

[624] 표 76

```
<Request xmlns="http://www.nipa.kr/eDocument_Circulation"> <DelAddrReq>
<Name>홍길동</Name> <IDN>111112222222</IDN>
<RAddress>#000-0000-0000</RAddress> </DelAddrReq></Request>
```

[625] 응답 유통메시지 구조는 아래 표 77과 같으며, 메시지 예제는 아래 표 78과 같다.

[626] 표 77

항목 명	설명	반복회수	유형	길이
Response	■ 응답 Root 엘리먼트			
DelAddrRes	■ 전자문서 중계자 회원 삭제 응답 엘리먼트			
	ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean
	RAddress	■ 공인전자주소	0..∞	String
	ErrorCode	■ 오류 코드(처리 결과가 실패(0)인 경우에만 해당 오류 코드 입력)	0..1	String

[627] * 표 77에서, ErrorCode는 ResultCode가 실패(0)로 입력된 경우, 오류 원인에 해당하는 오류 코드를 입력.

[628] 표 78

```
<Response xmlns="http://www.nipa.kr/eDocument_Circulation"> <DelAddrRes>
<ResultCode>1</ResultCode> </DelAddrRes></Response>
```

[629] 이하, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서 물리적 주소정보 검색 인터페이스에 대하여 설명하면 다음과 같다.

[630] 물리적 주소정보 검색 인터페이스는 전자문서 중계자 또는 송수신 개체가 주소 디렉터리서버에 전자문서 수신자의 공인전자주소에 해당하는 물리적 주소 정보와 메시지 보안처리를 위한 공인인증서 정보를 요청하여 응답을 받는 인터페이스로서, 전자문서 수신자의 공인전자주소 및 공인인증서 요청 여부를 요청메시지에 포함하여 전송한 후, 주소 디렉터리서버로부터 전자문서 수신자의 물리적 주소 정보(IP 주소 또는 Domain 주소) 및 공인인증서 정보를 응답메시지로 수신받는다.

[631] 물리적 주소정보 검색처리와 관련한 메시지 교환 흐름은 도 48과 같다.

[632] 요청 유통메시지 구조는 아래 표 79와 같으며, 메시지 예제는 아래 표 80과 같다.

[633] 표 79

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
SchAddrReq	■ 공인전자주소 조회 요청 엘리먼트			
ReqInfo	■ 요청 공인전자주소 정보 엘리먼트	1..∞		
RAddress	■ 공인전자주소	1..1	String	최대 128
IsCert	■ 공인인증서 요청 여부 - 요청: 1 - 요청 안함: 0	1..1	Integer	1

[634] 표 80

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SchAddrReq>
    <ReqInfo>
      <RAddress>#000-0000-0000</RAddress>
      <IsCert>0</IsCert>
    </ReqInfo>
  </SchAddrReq>
</Request>

```

[635] 응답 유통메시지 구조는 아래 표 81과 같으며, 메시지 예제는 아래 표 82와 같다.

[636] 표 81

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
SchAddrRes	■ 공인전자주소 조회 응답 엘리먼트			
ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean	-
ResultData	■ 결과 목록	0..∞		
RAddress	■ 공인전자주소	0..∞	String	최소 1 최대 128
IsExist	■ 주소 정보 존재 여부(Attribute) - 존재: 1 - 미존재: 0	1..1	Integer	1
Endpoint	■ 공인전자주소의 물리적 주소	0..1	String	최대 256
PeerRegNum	■ 송수신 개체의 인증번호	0..1	String	10
Cert	■ 수신자 공개키	0..1	Base64	-
PeerCert	■ 송수신 개체의 공개키	0..1	Base64	-
ErrorCode	■ 오류 코드(처리 결과가 실패(0) 인 경우에만 해당 오류 코드 입력)	0..1	String	256

[637] 표 82

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SchAddrRes>
    <ResultCode>1</ResultCode>
    <ResultData>
      <RAddress IsExist="1">#000-0000-0000</RAddress>
      <Endpoint>http://111.111.111.111:8080/imxs/msh</Endpoint>
      <Cert>MFJIFDjfkdsfjsl...</Cert>
    </ResultData>
  </SchAddrRes>
</Response>

```

[638] 이하, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서 스팸메시지 신고 접수 인터페이스에 대하여 설명하면 다음과 같다.

[639] 스팸 메시지 신고 접수 인터페이스는 전자문서 중계자 또는 송수신 개체가 주소 디렉터리서버에 스팸메시지를 신고하는 인터페이스로서, 스팸 송신자의 공인전자주소와 스팸 메시지 정보를 요청메시지에 포함하여 전송한 후, 주소 디렉터리서버로부터 스팸신고 접수여부를 응답메시지로 수신 받는다. 주소

디렉터리서버는 신고 접수된 스팸메시지에 대한 스팸 여부 판단이 완료되면, 유통 메시징서버 상호 간 연계 인터페이스의 “메시지 전송” 인터페이스를 사용하여 처리 결과를 통지한다.

[640] 스팸메시지 신고 접수처리와 관련한 메시지 교환 흐름은 도 49과 같다.

[641] 요청 유통메시지 구조는 아래 표 83과 같으며, 메시지 예제는 아래 표 84와 같다.

[642] 표 83

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
ReportSpamReq	■ 스팸 신고 요청 엘리먼트			
ReportRAddress	■ 신고자 공인전자주소	1..1	String	최대 128
SpamRAddress	■ 스팸 송신자 공인전자주소	1..1	String	최대 128
ContentsPid	■ 스팸 송신자가 보낸 Content 파 일 참조 아이디(스팸 신고 메시지의 MIME Part cid)	1..1	String	최대 256
AttacheFileInfo	■ 스팸 송신자가 보낸 첨부 문서 정보	0..*		
FilePid	■ 스팸 송신자가 보낸 첨부문서 참 조 아이디(스팸 신고 메시지의 MIME Part cid)	1..1	String	최대 256
FileName	■ 스팸 송신자가 보낸 첨부문서 참 조명	1..1	String	최대 256
SpamPeerCorpNum	■ 스팸 사용자가 이용하는 유통 메 시징서버 운영자의 사업자 번호	1..1	String	최대 10

[643] 표 84

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <ReportSpamReq>
    <ReportRAddress>#000-0000-0000</ReportRAddress>
    <SpamRAddress>#000-0000-0000</SpamRAddress>
    <ContentsPid>cid-1</ContentsPid>
    <AttacheFileInfo>
      <FilePid>cid-2</FilePid>
      <FileName>License.txt</FileName>
    </AttacheFileInfo>
    <SpamPeerCorpNum>1234567890</SpamPeerCorpNum>
  </ReportSpamReq>
</Request>

```

[644] 응답 유통메시지 구조는 아래 표 85와 같으며, 메시지 예제는 아래 표 86과 같다.

[645] 표 85

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
ReportSpamRes	■ 스팸 신고 응답 엘리먼트			
ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean	-
RAddress	■ 스팸 송신자 공인전자주소	0..1	String	최소 1 최대 128
ErrorCode	■ 오류 코드(처리 결과가 실패(0)인 경우에만 해당 오류 코드 입력)	0..1	String	256

[646] * 표 85에서, ResultCode는 스팸 신고 메시지에 대한 단순 접수 처리 결과임에 주의.

[647] 표 86

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <ReportSpamRes>
    <ResultCode>1</ResultCode>
    <RAddress>#스팸머.개인</RAddress>
  </ReportSpamRes>
</Response>

```

[648] 이하, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서 화이트 리스트 통보 인터페이스에 대하여 설명하면 다음과 같다.

[649] 화이트 리스트 통보 인터페이스는 송수신 개체에 화이트 리스트(유통체계에 참여하는 송수신 개체 및 송수신자의 공인전자주소 목록)를 통보해주기 위한 인터페이스이다.

[650] 화이트 리스트 통보와 관련한 메시지 교환 흐름은 도 50과 같다.

[651] 요청 유통메시지 구조는 아래 표 87과 같으며, 메시지 예제는 아래 표 88와 같다.

[652] 표 87

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
BroadcastWlistReq	■ 화이트 리스트 통보 요청 엘리먼트			
PeerInfo	■ 송수신 개체 정보	1..∞		
Name	■ 등록자실명(개인:이름 기관:사업자명)	1..1	String	최대 128
PeerCorpNum	■ 소속 유통 메시징 서버의 운영자의 사업자번호	1..1	String	최대 128
CorpType	■ (일반기업:C, ASP사업자:A)	1..1	String	최대 256
RAddress	■ 공인전자주소	1..∞	String	최소 1 최대 128
Tel	■ 전화번호 ('-' 없이 입력함)	0..1	Integer	최소 9 최대 12

[653] 표 88

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <BroadcastWlistReq>
    <PeerInfo>
      <Name>홍길동</Name>
      <PeerCorpNum>22432456</PeerCorpNum>
      <CorpType>C</CorpType>
      <RAddress>#000-0000-0000</RAddress>
    </PeerInfo>
  </BroadcastWlistReq>
</Request>

```

[654] 응답 유통메시지 구조는 아래 표 89와 같으며, 메시지 예제는 아래 표 90과 같다.

[655] 표 89

항목명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
BroadcastWlistRes	■ 화이트 리스트 통보 응답 엘리먼트			
ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean	-
ErrorCode	■ 오류 코드(처리 결과가 실패(0)인 경우에만 해당 오류 코드 입력)	0..1	String	256

[656] * 표 89에서, ResultCode는 스팸 신고 메시지에 대한 단순 접수 처리 결과임에 주의

[657] 표 90

```
<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <BroadcastWlistRes>
    <ResultCode>1</ResultCode>
  </BroadcastWlistRes>
</Response>
```

[658] 이하, 유통 메시징서버와 주소 디렉터리 서버 간 인터페이스에 있어서 블랙 리스트 통보 인터페이스에 대하여 설명하면 다음과 같다.

[659] 블랙리스트 통보 인터페이스는 송수신 개체에 블랙 리스트(수신 거부 리스트)를 통보해주기 위한 인터페이스이다. 통보된 블랙 리스트는 송수신 개체에 의해 블랙리스트 관리로 쓰인다.

[660] 블랙 리스트 통보와 관련한 메시지 교환 흐름은 도 51과 같다.

[661] 요청 유통메시지 구조는 아래 표 91과 같으며, 메시지 예제는 아래 표 92와 같다.

[662] 표 91

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
BroadcastBlistReq	■ 블랙 리스트 통보 요청 엘리먼트			
UserInfo	■ 블랙 리스트 엘리먼트	1..∞		
SpamPeerCorpNum	■ 스팸을 송신한 유통 메시징 서버의 운영자의 사업자번호	1..1	String	최대 128
Name	■ 등록자실명(개인:이름 기관:사업자명)	0..1	String	최대 128
RAddress	■ 공인전자주소	1..∞	String	최소 1 최대 128
Tel	■ 전화번호 ('-' 없이 입력함)	0..1	Integer	최소 9 최대 12

[663] 표 92

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <BroadcastBlistReq>
    <UserInfo>
      <SpamPeerCorpNum>32345633</SpamPeerCorpNum>
      <Name>박인수</Name>
      <RAddress>#000-0000-0000</RAddress>
    </UserInfo>
  </BroadcastBlistReq>
</Request>

```

[664] 응답 유통메시지 구조는 아래 표 93과 같으며, 메시지 예제는 아래 표 94와 같다.

[665] 표 93

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
BroadcastBlistResponses	■ 스팸 신고 응답 엘리먼트			
ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean	-
ErrorCode	■ 오류 코드(처리 결과가 실패(0)인 경우에만 해당 오류 코드 입력)	0..1	String	256

[666] 표 94

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <BroadcastBlistRes>
    <ResultCode>1</ResultCode>
  </BroadcastBlistRes>
</Response>

```

- [667] [유통 메시징서버 상호 간 연계 인터페이스]
- [668] 유통 메시징서버는 기본적으로 다른 송수신 개체 또는 전자문서 중계자가 구축한 유통 메시징서버와 메시지 송수신을 위해 연계를 하여야 한다.
- [669] 이러한 기본 기능 외에, 유통증명서를 제3자 보관기관에 보관하기 위하여 제3자 보관기관 사업자의 유통 메시징서버와 타 메시징서버 간에는 유통증명서 전달 연계 기능도 추가적으로 제공되어야 한다.
- [670] 유통 메시징서버 상호 간 연계 인터페이스는 메시징서버 상호 간 메시지 및 유통증명서를 송수신하기 위한 프로토콜로서, 아래 표 95와 같은 인터페이스로 구분된다.
- [671] 표 95

인터페이스 구분		인터페이스 설명
유통메시징 서버 상호 간 연계	메시지 전송	☐ 송신자의 유통 메시징서버가 수신자의 유통 메시징 서버로 메시지를 전송하기 위한 인터페이스 ☐ 메시지 수신 후, 수신 유통 메시징서버는 수신증명서 또는 오류메시지를 응답메시지로 송신 유통 메시징 서버에 되돌려 줌
	유통증명서 전달	☐ 메시지 수신자의 유통 메시징서버가 메시지 송신자의 유통 메시징서버에 열람증명서를 전송하기 위한 인터페이스 ☐ 메시지 수신자가 메시지를 열람하였을 때 메시지 수신 유통 메시징서버는 열람증명서를 메시지 송신 메시징서버에 송신하여야 함 ☐ 유통증명서를 수신한 유통 메시징 서버는 수신확인 ACK 또는 오류메시지를 응답메시지를 되돌려 줌
	유통증명서 보관요청	☐ 일반 유통 메시징서버가 제3자 보관기관 사업자의 유통 메시징서버에 유통증명서를 보관 요청하는 인터페이스 ☐ 제3자 보관기관 사업자가 구축한 유통 메시징서버는 일반 유통 메시징서버로부터 유통증명서를 전달받아 제3자 보관기관에 보관하는 서비스를 반드시 구축하여야 함 ☐ 제3자 보관기관 사업자가 아닌 일반 기업/기관/개인의 유통 메시징서버에서는 이 서비스는 구축 대상이 아님
	제3자 보관기관 보관결과 전달	☐ 제3자 보관기관 사업자의 유통 메시징서버가 유통증명서/유통문서를 보관한 후 보관결과(최초등록증명서)를 보관 요청 유통 메시징서버에 전달하는 인터페이스

[672] 이와 같은 유통 메시징서버 상호 간 인터페이스의 상세 내용에 대하여 설명하면 다음과 같다.

[673] 먼저, 유통 메시징서버 상호 간 인터페이스에 있어서 **공통 사항**에 대하여 설명하면 다음 ①과 같다.

[674] ① 요청 및 응답메시지의 MessageHeader 확장

[675] 유통 메시징서버 상호 간 연계 인터페이스 메시지의 첫 번째 MIME Part인 SOAP 메시지 내에는 송신자의 전자서명정보가 포함되어 전달되어야 하며, 수신자가 SOAP 메시지의 전자서명에 사용된 인증서의 소유자가 해당 송신자와 일치하는지를 검증(VID 검증)하는데 필요한 추가적인 송신자의 정보(CorpNum, RValue) 또한 포함되어 전달되어야 한다.

[676] 추가적인 송신자의 정보는 요청 및 응답메시지의 SOAP 메시지 내 MessageHeader 요소 하위에 확장 요소(any ##other 위치)로서 위치해야 한다.

[677] 확장 요소 구조는 아래 표 96과 같으며, 스키마 구조는 아래 표 97과 같다.

[678] 표 96

항목 명	설명	반복 회수	유형	길이
Extension	■ 확장 요소 엘리먼트	1..1		
CorpNum	■ 송신자 사업자등록번호	1..1	String	10
RValue	■ 송신자 공인인증서 개인키에서 추출한 RValue ■ RValue를 Base64로 인코딩하여 입력해야 함	1..1	String	28

[679] 표 97

```

<eb:MessageHeader SOAP:mustUnderstand="1" eb:id="MessageHeader"
eb:version="2.0">
  <eb:From>
    <eb:PartyId eb:type="ecf_cd">123456789</eb:PartyId>
    <eb:Role>sender</eb:Role>
  </eb:From>
  <eb:To>
    <eb:PartyId eb:type="ecf_cd">567890123</eb:PartyId>
    <eb:Role>receiver</eb:Role>
  </eb:To>
  <eb:CPAId>urn:ecm-and-ecm-cpa</eb:CPAId>
  <eb:ConversationId>20001209-133003-28572</eb:ConversationId>
  <eb:Service>>urn:ecm-service</eb:Service>
  <eb:Action>request</eb:Action>
  <eb:MessageData>

<eb:MessageId>20110210-170644Z-00057@127.0.0.18d1f96bf-9cd6-4049-9fdb-
a6c0ed9af46
7</eb:MessageId>
  <eb:Timestamp>2011-02-10T08:06:44.810Z</eb:Timestamp>
</eb:MessageData>
  <eb:DuplicateElimination></eb:DuplicateElimination>
  <Extention>
    <CorpNum>2208203228</CorpNum>
    <RValue>asdfasdf</RValue>
  </Extention>
</eb:MessageHeader>

```

[680] 이하, 유통 메시징서버 상호 간 인터페이스에 있어서 메시지 전송 인터페이스에 대하여 설명하면 다음과 같다.

[681] 메시지 전송 인터페이스는 유통 메시징서버가 타 유통 메시징서버에 메시지를 전송할 시에 사용된다.

[682] 메시지 전송에 있어서 메시지 교환 흐름은 도 52와 같다.

[683] 메시지 교환 시에 요청 포맷은 도 53과 같으며, 도 53과 같은 전체 메시지 구조에 있어서 첫번째 MIME Part에는 SOAP 메시지, 두번째 MIME Part에는 요청 유통메시지, 그리고 사용자가 첨부한 문서가 있는 경우에는 세번째 MIME Part부터 위치한다.

[684] 요청 유통메시지의 구조는 아래 표 98과 같으며, 실제 예시는 다음과 같다.

[685] 표 98

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
SendMsgReq	■ 메시지 전송 요청 엘리먼트			
DocType	■ 유통메시지 유형 - 문서: 0	1..1	Integer	1
Title	■ 메시지 제목	1..1	String	최대 256
Text	■ 메시지 본문 - 송신자에 의해 수신자의 인증 서로 암호화 될 수 있음	0..1	String	-
Sender	■ 송신자 공인전자주소	1..1	String	최대 128
Receiver	■ 수신자 공인전자주소	1..1	String	최대 128
ReqConfirm	■ 열람증명서 요청 - 미요청: 0 - 요청: 1	1..1	Integer	1
IsEncrypted	■ 메시지 암호화 여부 - 평문 : 0 - 암호화문 : 1	1..1	Integer	1
Identifier	■ 본 요청 유통메시지의 고유 식 별값 (UUID)	1..1	String	36

[686] * 표 98에 있어서, 문서 전달 목적으로 본문이 필요 없는 경우 Text 생략 가능

[687] 표 99

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SendMsgReq>
    <DocType>0</DocType>
    <Title>발송 문서</Title>
    <Text>발송 문서 본문</Text>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <ReqConfirm>1</ReqConfirm>
    <IsEncrypted>0</IsEncrypted>
    <Identifier>b366ff65-16c8-4d9d-a0ba-d76a2cc95ad2</Identifier>
  </SendMsgReq>
</Request>

```

[688] 메시지 교환 시에 응답 포맷은 도 54와 같으며, 도 54와 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 응답 유통메시지, 그리고 세 번째 MIME Part에는 수신증명서가 위치한다. 만약 요청메시지에 대한 처리과정에서 오류가 발생하였다면, 세 번째 MIME Part는 생성하지 않는다.

[689] 응답 유통메시지의 구조는 아래 표 100과 같으며, 실제 예시는 아래 표 101과 같다.

[690] 표 100

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
SendMsgRes	■ 메시지 전송 응답 엘리먼트			
DocType	■ 유통메시지 유형 - 수신증명서 : 1 - 오류 : 9	1..1	Integer	1
RefIdentifier	■ 본 응답 유통메시지에 대응하는 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36
ErrorCode	■ 오류 코드(유통메시지 유형이 오류(9)인 경우에만 해당 오류 코드 입력)	0..1	String	256

[691] * 표 100에서, DocType이 오류(9)인 경우는 수신증명서가 위치하게 되는 MIME Part3을 생성하지 않음.

[692] 표 101

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SendMsgRes>
    <DocType>1</DocType>
    <RefIdentifier>b366ff65-16c8-4d9d-a0ba-
d76a2cc95ad2</Identifier>
  </SendMsgRes>
</Response>

```

- [693] 이하, 유통 메시징서버 상호 간 인터페이스에 있어서 유통증명서 전달 인터페이스에 대하여 설명하면 다음과 같다.
- [694] 유통증명서 전달 인터페이스는 유통 메시징서버가 타 유통 메시징서버에 열람증명서를 전송할 때 사용된다. 또한 유통 중계서버가 전자문서 전송을 의뢰받아 수신 유통 메시징서버에 송신한 후 응답메시지로 수신한 수신증명서를 전송의뢰 유통 메시징서버에 전송할 때도 사용된다.
- [695] 유통증명서 전달 처리와 관련한 메시지 교환 흐름은 도 55와 같다.
- [696] 유통증명서 전달 요청 포맷은 도 56과 같으며, 도 56과 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지, 그리고 세 번째 MIME Part에는 유통증명서가 위치한다.
- [697] 요청 유통메시지 구조는 아래 표 102와 같으며, 실제 예시는 아래 표 103과 같다.
- [698] 표 102

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
SendCertReq	■ 유통증명서 전달 요청 엘리먼트			
DocType	■ 유통메시지 유형 - 수신증명서: 1 - 송신증명서: 2 - 열람증명서: 3	1..1	Integer	1
Sender	■ 증명서 송신자의 공인전자주소	1..1	String	최대 128
Receiver	■ 수신자의 공인전자주소	1..1	String	최대 128
Identifier	■ 본 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36
TargetIdentifier	■ 증명서 발급의 대상이 되는 메시지 전송 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36

- [699] 표 103

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SendCertReq>
    <DocType>1</DocType>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <Identifier>5347146a-3528-4469-8ef7-9c346ab36d54</Identifier>
    <TargetIdentifier>b366ff65-16c8-4d9d-a0ba-
d76a2cc95ad2</TargetIdentifier>
  </SendCertReq>
</Request>

```

[700] 유통증명서 전달 응답 포맷은 도 57(57a는 성공인 경우, 57b는 오류인 경우)과 같으며, 도 57과 같은 전체 메시지 구조에 있어서, 요청메시지에 대한 처리가 성공인 경우 첫 번째 MIME Part에 수신확인 Acknowledgement SOAP 메시지만 위치하며, 오류인 경우는 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 오류 응답 유통메시지가 위치한다.

[701] 응답 유통메시지 구조는 아래 표 104와 같으며, 표 104는 처리 결과가 오류인 경우에만 해당된다.

[702] 표 104

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
SendCertRes	■ 유통증명서 전달 응답 엘리먼트			
DocType	■ 전송 메시지 유형 - 오류 : 9	1..1	Integer	1
RefIdentifier	■ 본 응답 유통메시지에 대응하는 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36
ErrorCode	■ 오류 코드	1..1	String	256

[703] 이하, 유통 메시징서버 상호 간 인터페이스에 있어서 유통증명서 보관요청 인터페이스에 대하여 설명하면 다음과 같다.

[704] 유통증명서 보관요청 인터페이스는 송수신 개체의 유통 메시징서버가 유통증명서를 제3자 보관기관에 보관하기 위하여, 제3자 보관기관 사업자의 유통 메시징 서버에 유통증명서에 대한 보관 요청을 할 때 사용된다. 본 인터페이스 상의 응답메시지에는 수신확인 정보만 포함되며, 유통증명서를 제3자 보관기관에 보관한 결과로서 발급받은 최초등록증명서는 후술하는 "제3자 보관기관 보관결과 전달 인터페이스"를 사용하여 보관요청 유통 메시징서버에 전달한다.

[705] 유통증명서 보관요청 처리와 관련한 메시지 교환 흐름은 도 58과 같다.

- [706] 유통증명서 보관 요청 포맷은 도 59와 같으며, 도 59와 같은 전체 메시지 구조에 있어서, 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지, 그리고 세 번째 MIME Part에는 유통증명서가 위치한다.
- [707] 요청 유통메시지 구조는 아래 표 105와 같으며, 실제 예시는 아래 표 106과 같다.
- [708] 표 105

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
RegCertReq	■ 유통증명서 보관 요청 엘리먼트			
DocType	■ 유통메시지 유형 - 수신증명서: 1 - 송신증명서: 2 - 열람증명서: 3	1..1	Integer	1
Sender	■ 송신자의 공인전자주소	1..1	String	최대 128
Receiver	■ 수신자의 공인전자주소	1..1	String	최대 128
Identifier	■ 본 요청 유통메시지의 고유 식별 값 (UUID)	1..1	String	36

- [709] 표 106

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <RegCertReq>
    <DocType>1</DocType>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <Identifier>5347146a-3528-4469-8ef7-9c346ab36d54</Identifier>
  </RegCertReq>
</Request>

```

- [710] 유통증명서 보관 응답 포맷은 도 60(60a는 성공인 경우, 60b는 오류인 경우)과 같으며, 도 60과 같은 전체 메시지 구조에 있어서, 요청메시지에 대한 처리가 성공인 경우 첫 번째 MIME Part에 수신확인 Acknowledgement SOAP 메시지만 위치하며, 오류인 경우는 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 오류 응답 유통메시지가 위치한다.
- [711] 응답 유통메시지 구조는 아래 표 107과 같으며, 표 107은 처리 결과가 오류인 경우만 해당된다.
- [712] 표 107

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
RegCertRes	■ 유통증명서 보관 응답 엘리먼트			
DocType	■ 유통메시지 유형 - 오류 : 9	1..1	Integer	1
RefIdentifier	■ 본 응답 유통메시지에 대응하는 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36
ErrorCode	■ 오류 코드	1..1	String	256

[713] 이하, 유통 메시징서버 상호 간 인터페이스에 있어서 제3자 보관기관 보관결과 전달 인터페이스에 대하여 설명하면 다음과 같다.

[714] 제3자 보관기관 보관결과 전달 인터페이스는 제3자 보관기관 사업자의 유통 메시징서버가 제3자 보관기관에 유통증명서를 보관한 후 해당 결과로서 수신한 최초등록증명서를 유통증명서 보관을 요청한 유통 메시징서버에 송신할 때 사용된다.

[715] 제3자 보관기관 보관결과 전달 처리와 관련한 메시지 교환 흐름은 도 61과 같다.

[716] 제3자 보관기관 보관결과 전달 포맷은 도 62와 같으며, 도 62와 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지 그리고 세 번째 MIME Part에는 최초등록증명서가 위치한다. 만약 유통증명서를 제3자 보관기관에 보관하는 과정에서 오류가 발생하였다면, 세 번째 MIME Part 는 생성하지 않는다.

[717] 요청 유통메시지 구조는 아래 표 108과 같으며, 실제 예시는 아래 표 109와 같다.

[718] 표 108

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
RegResultReq	■ 제3자 보관기관 보관 결과 전달 처리 요청 엘리먼트			
DocType	■ 유통메시지 유형 - 최초등록증명서: 4 - 오류: 9	1..1	Integer	1
Sender	■ 송신자의 공인전자주소	1..1	String	최대 128
Receiver	■ 수신자의 공인전자주소	1..1	String	최대 128
Identifier	■ 본 요청 유통메시지의 고유 식별값 (UUID)	0..1	String	최대 128
TargetIdentifier	■ 본 요청 유통메시지의 대상이 되는 유통증명서 보관 요청 유통메시지의 고유 식별 값 (UUID)	1..1	String	최대 128
ErrorCode	■ 오류 코드(유통메시지 유형이 오류(9)인 경우에만 해당 오류 코드 입력)	0..1	String	최대 256

[719] * 표 108에 있어서, DocType이 오류(9) 인 경우는 최초등록증명서가 위치하게 되는 MIME Part 3을 생성하지 않음

[720] 표 109

— 성공

```
<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <ReqResultReq>
    <DocType>4</DocType>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <Identifier>dd27e2e2-4731-4da1-8043-a250dcc8690c</Identifier>
    <TargetIdentifier>5347146a-3528-4469-8ef7-
9c346ab36d54</RefIdentifier>
  </regResultReq>
</Request>
```

— 오류

```
<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <RegResultReq>
    <DocType>9</DocType>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <TargetIdentifier>5347146a-3528-4469-8ef7-
9c346ab36d54</RefIdentifier>
    <ErrorCode>ERR-01-0001</ErrorCode>
  </RegResultReq>
</Request>
```

[721] 제3자 보관기관 보관결과 응답 포맷은 도 63(63a는 성공인 경우, 63b는 오류인 경우)과 같으며, 도 63과 같은 전체 메시지 구조에 있어서, 요청메시지에 대한 처리가 성공인 경우는 첫 번째 MIME Part에 수신확인 Acknowledgement SOAP 메시지만 위치하며, 오류인 경우는 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 오류 응답 유통메시지가 위치한다.

[722] 응답 유통메시지 구조는 아래 표 110과 같으며, 표 110은 처리 결과가 오류인 경우만 해당된다.

[723] 표 110

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
RegResultRes	■ 제3자 보관기관 보관 결과 전달 처리 결과 엘리먼트			
DocType	■ 유통메시지 유형 - 오류 : 9	1..1	Integer	1
RefIdentifier	■ 본 응답 유통메시지에 대응하는 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36
ErrorCode	■ 오류 코드	1..1	String	256

[724] [유통 클라이언트와 유통 메시징서버 간 연계 인터페이스]

[725] 유통 메시징서버는 실제 전자문서 유통을 요청하는 사용자(내부 송수신자 또는 공인 송수신자)를 위한 시스템(유통 클라이언트)과 연계하여 사용자에게 문서 송수신 기본 기능을 제공하여야 한다.

[726] 유통 클라이언트와 유통 메시징서버 간 연계 인터페이스는 유통 클라이언트가 전자문서를 송신하고 수신하기 위하여 일차적으로 유통 메시징 서버와 통신하기 위한 프로토콜로서 아래 표 111과 같은 인터페이스로 구분된다.

[727] 표 111

인터페이스 구분		인터페이스 설명
유통 클라이언트와 유통 메시징 서버 간 연계 인터페이스	메시지 전송요청	☐ 유통 클라이언트가 유통 메시징서버에 메시지 전송을 요청하기 위한 인터페이스
	메시지 목록 요청	☐ 유통 메시징서버에 수신된 수신 메시지의 목록을 요청하기 위한 인터페이스 ☐ 유통 메시징서버는 유통 클라이언트 사용자를 인증한 후, 해당 사용자에게 수신된 메시지목록을 전달함
	메시지 상세정보 요청	☐ 유통 클라이언트가 유통 메시징서버에 사용자에게 수신된 특정 메시지의 전체 정보를 요청하기 위한 인터페이스 ☐ 유통 메시징서버는 유통 클라이언트 사용자를 인증한 후, 해당 사용자가 요청한 메시지의 전체 정보를 전달함
	스팸 메시지 신고	☐ 유통 클라이언트가 스팸 메시지를 신고하기 위한 인터페이스 ☐ 유통 메시징 서버는 유통 클라이언트 사용자를 인증한 후, 해당 사용자가 신고한 내용을 주소 디렉터리 서버로 전달함
	물리적 주소정보 검색	☐ 유통 클라이언트가 물리적 주소 정보를 검색하기 위한 인터페이스 ☐ 유통 메시징 서버는 유통 클라이언트 사용자를 인증한후, 주소 디렉터리 서버로 검색 요청후 결과를 전달함

- [728] 유통클라이언트와 유통 메시징서버 간 연계 인터페이스의 상세 내용을 설명하면 다음과 같다.
- [729] 먼저, 유통클라이언트와 유통 메시징서버 간 연계 인터페이스의 공통 사항에 대하여 설명하면 다음 ①과 같다.
- [730] ① 요청메시지의 MessageHeader 확장
- [731] 유통 클라이언트가 유통 메시징 서버에 송신하는 요청메시지의 첫 번째 MIME Part인 SOAP 메시지 내에는 사용자의 전자서명 정보가 포함되어 전달되어야 하며, 유통 메시징 서버가 SOAP 메시지의 전자서명에 사용된 인증서의 소유자가 해당 사용자와 일치하는지를 검증(VID 검증)하는데 필요한 추가적인 사용자의 정보(IDN, RValue) 또한 포함되어 전달되어야 한다.
- [732] 해당 정보는 요청메시지의 SOAP 메시지 내 MessageHeader 요소 하위에 확장 요소(any ##other 위치)로서 위치해야 한다.
- [733] 또한 동일 인증서를 사용하는 복수의 내부 사용자들에 대한 개별인증정보가 추가될 수 있다.
- [734] 확장 요소 구조는 아래 표 112와 같으며, 확장 요소 예시는 아래 표113과 같다.
- [735] 표 112

항목 명		설명	반복 회수	유형	길이
User Info		■ 확장 요소 엘리먼트	1..1		
	IDN	■ 사용자 식별번호 - 개인: 주민등록번호 - 사업자: 사업자등록번호	1..1	String	10
	RValue	■ 사용자 공인인증서 개인키에서 추출한 RValue ■ RValue를 Base64로 인코딩 하여 입력해야 함	1..1	String	28
	Id	■ 유통 메시징 서버에 등록된 사용자 아이디	0..1	String	최대 20
	Password	■ 유통 메시징 서버에 등록된 사용자 패스워드	0..1	String	8
	AuthType	■ 복수의 내부 사용자 인증 방식 ■ 아이디, 패스워드 방식은 기본적으로 0으로 설정 ■ 아이디, 패스워드 방식 외 다른 방식으로 사용자를 인증하고자 할 경우 AuthType 값 및 아래 확장 요소를 자체적으로 정의하여 사용	1..1	Integer	1
	Any Extension	■ 아이디, 패스워드 방식 외 다른 방식으로 사용자를 인증하고자 할 경우 새로운 요소를 자체적으로 정의하여 사용 ■ ex> Token, Certificate	0..1	Any	-

[736] 표 113

```

<eb:MessageHeader      SOAP:mustUnderstand="1"      eb:id="MessageHeader"
eb:version="2.0">
  <eb:From>
    <eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-
type:duns">openAPI
_Sender</eb:PartyId>
    <eb:Role>http://www.rosettanet.org/processes/3A4.xml#Buyer</eb:Role>
  </eb:From>
  <eb:To>
    <eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-
type:duns">openAPI_
Receiver</eb:PartyId>

    <eb:Role>http://www.rosettanet.org/processes/3A4.xml#seller</eb:Role>
  </eb:To>
  <eb:CPAId>uri:openapi-and-openapi-cpa_mxs</eb:CPAId>
    <eb:ConversationId>uri:openapi-and-openapi-
cpa_mxs:0210050643</eb:ConversationId>
  <eb:Service>bpid:icann:rosettanet.org:3A4$2.0</eb:Service>
  <eb:Action>RequestRelaySend</eb:Action>
  <eb:MessageData>

  <eb:MessageId>20110210-170644Z-00057@127.0.0.18d1f96bf-9cd6-4049-9fdb-
a6c0ed9af
467</eb:MessageId>
    <eb:Timestamp>2011-02-10T08:06:44.810Z</eb:Timestamp>
  </eb:MessageData>
  <eb:DuplicateElimination></eb:DuplicateElimination>
  <UserInfo>
    <IDN>2208203228</CorpNum>
    <RValue>asdfasdf</RValue>
    <Id>tester1</Id>
    <Password>test</Password>
    <AuthType>0</AuthType>
  </UserInfo>
</eb:MessageHeader>

```

[737] 이하, 유통 클라이언트와 유통 메시징서버 간 연계 인터페이스에 있어서 메시

전송 요청 인터페이스에 대하여 설명하면 다음과 같다.

- [738] 메시지 전송 요청 인터페이스는 유통 클라이언트가 유통 메시징 서버를 통해 메시지를 전송하기 위하여 유통 메시징서버에 메시지를 전송할 때 사용된다.
- [739] 유통 클라이언트의 메시지 전송 처리 흐름은 도 64와 같다.
- [740] 유통 클라이언트의 메시지 전송 요청 포맷은 도 65와 같으며, 도 65와 같은 전체 메시지의 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통 메시지가 위치한다. 그리고 사용자가 첨부한 문서가 있는 경우 세 번째 MIME Part부터 위치한다.
- [741] 요청 유통메시지 구조는 아래 표 114와 같으며, 실제 예시는 아래 표 115와 같다.

[742] 표 114

항목명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
SendMsgReq	■ 메시지 전송 요청 엘리먼트			
DocType	■ 유통메시지 유형 - 문서: 0	1..1	Integer	1
Title	■ 메시지 제목	1..1	String	최대 256
Text	■ 메시지 본문 - 송신자에 의해 수신자의 인 증서로 암호화 될 수 있음	0..1	String	-
Sender	■ 송신자 공인처주소	1..1	String	최대 128
Receiver	■ 수신자 공인처주소	1..1	String	최대 128
ReqConfir	■ 열람증명서 요청 - 미요청: 0 - 요청: 1	1..1	Integer	1
IsEncrypted	■ 메시지 암호화 여부 - 평문: 0 - 암호화문: 1	1..1	Integer	1
Identifier	■ 본 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36

[743] * 표 114에 있어서, 문서 전달 목적으로 본문이 필요없는 경우 Text 생략 가능함.

[744] 표 115

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SendMsgReq>
    <DocType>0</DocType>
    <Title>발송 문서</Title>
    <Text>발송 문서 본문</Text>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <ReqConfirm>1</ReqConfirm>
    <IsEncrypted>1</IsEncrypted>
    <Identifier>b366ff65-16c8-4d9d-a0ba-d76a2cc95ad2</Identifier>
  </SendMsgReq>
</Request>

```

- [745] 유통 클라이언트 메시지 전송 응답 포맷은 도 66(66a는 성공인 경우, 66b는 오류인 경우)과 같으며, 도 66과 같은 전체 메시지 구조에 있어서, 요청메시지에 대한 처리가 성공인 경우는 첫 번째 MIME Part에 수신확인 Acknowledgement SOAP 메시지만 위치하며, 오류인 경우는 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 오류 응답 유통메시지가 위치한다.
- [746] 응답 유통메시지 구조는 표 116과 같으며, 표 116은 처리 결과가 오류인 경우만 해당된다.
- [747] 표 116

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
SendMsgRes	■ 메시지 전송 응답 엘리먼트			
DocType	■ 유통메시지 유형 - 오류 : 9	1..1	Integer	1
RefIdentifier	■ 본 응답 유통메시지에 대응하는 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36
ErrorCode	■ 오류 코드	1..1	String	256

- [748] 이하, 유통 클라이언트와 유통 메시징서버 간 연계 인터페이스에 있어서 메시지 목록 요청 인터페이스에 대하여 설명하면 다음과 같다.
- [749] 메시지 목록 요청 인터페이스는 유통 클라이언트가 유통 메시징서버에 수신된 메시지의 목록을 요청할 때 사용된다.
- [750] 유통 클라이언트의 메시지 목록 처리 흐름은 도 67과 같다.
- [751] 유통 클라이언트의 메시지 목록 요청 포맷은 도 68과 같으며, 도 68과 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지가 위치한다.
- [752] 요청 유통메시지 구조는 아래 표 117과 같으며, 실제 예시는 아래 표 118과

같다.

[753] 표 117

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
MsgListReq	■ 메시지 목록 요청 엘리먼트			
Requester	■ 요청자의 공인전자주소	1..1	String	최대 128
MsgSize	■ 유통메시지 목록의 개수	1..1	Integer	최대 100

[754] 표 118

```
<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <MsgListReq>
    <Requester>#000-0000-0000</Requester>
    <MsgSize>100</MsgSize>
  </MsgListReq>
</Request>
```

[755] 유통 클라이언트의 메시지 목록 응답 포맷은 표 67과 같으며, 표 67과 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 응답 유통메시지(유통 메시징서버에 수신된 유통메시지 목록)이 위치한다.

[756] 응답 유통메시지 구조는 아래 표 119와 같으며, 실제 예시는 아래 표 120과 같다.

[757] 표 119

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
MsgListRes	■ 메시지 목록 응답 엘리먼트			
ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean	-
ErrorCode	■ 오류 코드 (ResultCode가 실패(0) 일 경우에만 해당 오류 코드 입력)	0..1	Integer	1
List	■ 수신 유통메시지 리스트	0..∞		
DocType	■ 유통메시지 유형 - 문서: 0 - 수신 증명서 : 1 - 발신 증명서 : 2 - 열람 증명서 : 3 - 보관 증명서 : 4 - 오류: 9	1..1	Integer	최대 100
Title	■ 메시지 제목	0..1	String	최대 256
Sender	■ 송신자 공인전자주소	1..1	String	최대 128
Identifier	■ 유통메시지의 고유 식별값 (UUID)	0..1	String	36
TargetIdentifier	■ 유통메시지의 유형이 유통증명서인 경우, 증명서 발급대상이 되는 유통메시지의 고유 식별값 (UUID)	0..1	String	36
IsExistPayloads	■ 첨부 파일 존재 여부 - 없음: 0 - 있음: 1	1..1	Integer	1
IsEncrypted	■ 메시지 암호화 여부 - 평문 : 0 - 암호화문 : 1	1..1	Integer	1
SendDate	■ 문서 송신 시간	0..1	Long	-
ReceiveDate	■ 문서 수신 시간	0..1	Long	-

[758] 표 120

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <MsgListRes>
    <ResultCode>1</ResultCode>
    <List>
      <DocType>0</DocType>
      <Title>문서 제목</Title>
      <Sender>#000-0000-0000</Sender>
      <Identifier>e82fae92-981e-4d36-80b8-ec16cb6b7993</Identifier>
      <IsExistPayloads>0</IsExistPayloads>
      <IsEncrypted>0</IsEncrypted>
      <SendDate>1286263498929</SendDate>
      <ReceiveDate>1286273498929</ReceiveDate>
    </List>
    <List>
      <DocType>9</DocType>
      <Sender>#000-0000-0000</Sender>
      <RefIdentifier>e82fae92-981e-4d36-80b8-ec16cb6b7993</RefIdentifier>
      <IsExistPayloads>0</IsExistPayloads>
      <IsEncrypted>0</IsEncrypted>
    </List>
  </MsgListRes>
</Response>

```

- [759] 이하, 유통 클라이언트와 유통 메시징서버 간 연계 인터페이스에 있어서 메시지 상세정보 요청 인터페이스에 대하여 설명하면 다음과 같다.
- [760] 메시지 상세정보 요청 인터페이스는 유통 클라이언트가 유통 메시징서버에 수신된 특정 메시지와 첨부문서를 요청할 때 사용된다.
- [761] 유통 클라이언트의 상세정보 요청 처리 흐름은 도 69와 같다.
- [762] 유통 클라이언트의 메시지 상세정보 요청 포맷은 도 70과 같으며, 도 70과 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지 본문이 위치한다.
- [763] 요청 유통메시지 구조는 아래 표 121과 같으며, 실제 예시는 아래 표 122와 같다.
- [764] 표 121

항목명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
MsgDetailReq	■ 메시지 상세 정보 요청 엘리먼트			
Requester	■ 요청자의 공인전자주소	1..1	String	최대 128
RefIdentifier	■ 요청 대상 유통메시지의 고유 식별값 (UUID)	1..1	String	36

[765] 표 122

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <MsgDetailReq>
    <Requester>#000-0000-0000</Requester>
    <RefIdentifier>5f6d8126-a691-452b-b17a-
e3a8b8ce3ac5</RefIdentifier>
  </MsgDetailReq>
</Request>

```

[766] 유통 클라이언트의 메시지 상세정보 응답 포맷은 도 72와 같으며, 도 72와 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 응답 유통메시지(유통메시지 상세 정보) 그리고 첨부 문서가 있는 경우 세 번째 MIME Part부터 순서대로 위치한다.

[767] 응답 유통메시지 구조는 아래 표 123과 같으며, 실제 예시는 아래 표 124와 같다.

[768] 표 123

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
MsgDetailRes	■ 메시지 상세 정보 응답 엘리먼트			
	■ 유통메시지 유형 - 문서: 0 - 수신, 발송, 열람 증명서 : 1, 2, 3 - 보관 증명서 : 4 - 오류: 9	1..1	Integer	최대 100
DocType				
Title	■ 메시지 제목	0..1	String	최대 256
Text	■ 메시지 본문	0..1	String	-
Sender	■ 발송자 공인전자주소	1..1	String	최대 128
Receiver	■ 수신자 공인전자주소	0..1	String	최대 128
ReqConfirm	■ 열람증명서 요청 - 미요청: 0 - 요청: 1	0..1	Integer	1
IsEncrypted	■ 메시지 암호화 여부 - 평문 : 0 - 암호화문 : 1	1..1	Integer	1
Identifier	■ 유통메시지 고유 식별값 (UUID)	0..1	String	36
TargetIdentifier	■ 유통메시지의 유형이 유통증명서인 경우, 증명서 발급대상이 되는 유통메시지의 고유 식별값 (UUID)	0..1	String	36
ErrorCode	■ 오류 코드 (유통메시지 유형이 오류(9)일 경우에만 해당 오류 코드 입력)	0..1	Integer	1

[769] 표 124

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <MsgDetailRes>
    <DocType>0</DocType>
    <Title>문서 제목</Title>
    <Text>문서 본문</Text>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <ReqConfirm>1</ReqConfirm>
    <IsEncrypted>0</IsEncrypted>
    <Identifier>e82fae92-981e-4d36-80b8-ec16cb6b7993</Identifier>
  </MsgDetailRes>
</Content>

```

- [770] 이하, 유통 클라이언트와 유통 메시징서버 간 연계 인터페이스에 있어서 스팸 메시지 신고 인터페이스에 대하여 설명하면 다음과 같다.
- [771] 스팸 메시지 신고 인터페이스는 유통 클라이언트가 유통 메시징 서버에 스팸 메시지를 신고할 때 사용된다. 유통 메시징 서버는 주소 디렉터리 서버로 스팸 메시지를 신고 후 결과를 유통 클라이언트로 전달해 준다.
- [772] 유통 클라이언트의 스팸 메시지 신고 처리 흐름은 도 73과 같다.
- [773] 유통 클라이언트의 스팸 메시지 신고 포맷은 도 74와 같으며, 도 74와 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지가 위치한다.
- [774] 요청 유통메시지의 구조는 아래 표 125와 같으며, 메시지 예제는 아래 표 126과 같다.
- [775] 표 125

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
ReportSpamReq	■ 스팸 신고 요청 엘리먼트			
ReportRAdderss	■ 신고자 공인전자주소	1..1	String	최대 128
SpamRAdderss	■ 스팸 송신자 공인전자주소	1..1	String	최대 128
ContentsPid	■ 스팸 송신자가 보낸 Content 파일 참조 아이디(스팸 신고 메시지의 MIME Part cid)	1..1	String	최대 256
AttacheFileIn fo	■ 스팸 송신자가 보낸 첨부 문서 정 보			
FilePid	■ 스팸 송신자가 보낸 첨부문서 참 조 아이디(스팸 신고 메시지의 MIME Part cid)	1..1	String	최대 256

[776] 표 126

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <ReportSpamReq>
    <ReportRAddress>#000-0000-0000</ReportRAddress>
    <SpamRAddress>#000-0000-0000</SpamRAddress>
    <ContentsPid>cid-1</ContentsPid>
    <AttacheFileInfo>
      <FilePid>cid-2</FilePid>
      <FileName>License.txt</FileName>
    </AttacheFileInfo>
    <SpamPeerCorpNum>1234567890</SpamPeerCorpNum>
  </ReportSpamReq>
</Request>

```

[777] 유통 클라이언트의 스팸 메시지 응답 포맷은 도 75와 같으며, 도 75와 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 응답 유통메시지(유통 메시징서버에 수신된 유통메시지 목록)이 위치한다.

[778] 응답 유통메시지 구조는 아래 표 127과 같으며, 메시지 예제는 아래 표 128과 같다.

[779] 표 127

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
ReportSpamRes	■ 스팸 신고 응답 엘리먼트	1..1		
ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean	-
RAddress	■ 스팸 송신자 공인전자주소	0..1	String	최소 1 최대 128
ErrorCode	■ 오류 코드(처리 결과가 실패(0) 인 경우에만 해당 오류 코드 입력)	0..1	String	256

[780] * 표 127에 있어서, ResultCode는 스팸 신고 메시지에 대한 단순 접수 처리 결과임에 주의

[781] 표 128

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <ReportSpamRes>
    <ResultCode>1</ResultCode>
    <Address>#000-0000-0000</Address>
  </ReportSpamRes>
</Response>

```

- [782] 이하, 유통 클라이언트와 유통 메시징서버 간 연계 인터페이스에 있어서 물리적 주소 검색 인터페이스에 대하여 설명하면 다음과 같다.
- [783] 물리적 주소 검색 인터페이스는 유통 클라이언트가 유통 메시징 서버에 물리적 주소 검색을 요청할 때 사용한다. 유통 메시징 서버는 주소 디렉터리 서버로 물리적 주소를 검색후 결과를 전달해 준다.
- [784] 물리적 주소 검색 처리와 관련하여 메시지 교환 흐름은 도 76과 같다.
- [785] 물리적 주소 검색 요청 메시지 포맷은 도 77과 같으며, 도 77과 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지가 위치한다.
- [786] 요청 유통메시지 구조는 아래 표 129와 같으며, 메시지 예제는 아래 표 130과 같다.
- [787] 표 129

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
SchAddrReq	■ 회원 공인전자주소 조회 요청 엘리먼트			
ReqInfo	■ 요청 공인전자주소 정보 엘리먼트	1..∞		
Address	■ 공인전자주소	1..1	String	최대 128
IsCert	■ 공인인증서 요청 여부 - 요청: 1 - 요청 안함: 0	1..1	Integer	1

- [788] 표 130

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SchAddrReq>
    <ReqInfo>
      <Address>#000-0000-0000</Address>
      <IsCert>0</IsCert>
    </ReqInfo>
  </SchAddrReq>
</Request>

```

- [789] 물리적 주소 검색 응답 메시지 포맷은 도 78과 같으며, 도 78과 같은 전체

메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 응답 유통메시지(유통 메시징서버에 수신된 유통메시지 목록)이 위치한다.

[790] 응답 유통메시지 구조는 아래 표 131과 같으며, 실제 예시는 아래 표 132와 같다.

[791] 표 131

항목명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
SchAddrRes	■ 공인전자주소 조회 응답 엘리먼트			
ResultCode	■ 처리 결과 - 성공: 1 - 실패: 0	1..1	Boolean	-
ResultData	■ 결과 목록	0..∞		
RAddress	■ 공인전자주소	1..1	String	최소 1 최대 128
IsExist	■ 주소 정보 존재 여부(Attribute) - 존재: 1 - 미존재: 0	1..1	Integer	1
Endpoint	■ 공인전자주소의 물리적 주소	0..1	String	최대 256
PeerRegNu	■ 송수신 개체의 인증번호	0..1	String	10
Cert	■ 수신자 공인인증서	0..1	Base64	-
PeerCert	■ 송수신 개체의 공개키	0..1	Base64	-
ErrorCode	■ 오류 코드(처리 결과가 실패(0)인 경우에만 해당 오류 코드 입력)	0..1	String	256

[792] * 복수개의 RAddress 중 일부 또는 전체 주소에 대하여 검색은 정상적으로 수행되었으나 주소 부재의 오류가 발생하는 경우, 다른 인터페이스와는 달리 ResultCode를 성공(1)으로 입력함에 주의

[793] * RAddress는 ResultCode의 성공(1)/실패(0) 여부에 상관없이 기술하도록 하며, 속성정보인 IsExist에 각 RAddress에 대한 존재여부를 입력

[794] * Endpoint와 Cert는 IsExist의 값이 존재(1)인 경우에 입력

[795] * 요청메시지의 오류로 RAddress를 파싱하지 못한 경우, RAddress는 생략가능함(Endpoint 및 Cert도 결과적으로 생략됨)

[796] * ErrorCode는 ResultCode가 실패(0)로 입력된 경우, 즉 주소 부재의 오류를 제외한 타 오류가 발생한 경우, 오류 원인에 해당하는 오류 코드를 입력

[797] 표 132

```

<Response xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SchAddrRes>
    <ResultCode>1</ResultCode>
    <ResultData>
      <RAddress IsExist="1">#000-0000-0000</RAddress>
      <Endpoint>http://111.111.111.111:8080/imxs/msh</Endpoint>
      <Cert>MFJIFDjfkdfjsl...</Cert>
    </ResultData>
  </SchAddrRes>
</Response>

```

- [798] [유통 메시징서버와 유통 중계서버 간 연계 인터페이스]
- [799] 유통 중계서버는 전자문서유통 체계에서 유통 메시징서버 간 직접 전자문서를 전송하는 과정에서 오류가 발생하여 전송이 실패한 경우, 송신 유통 메시징서버를 대행하여 전자문서 전송을 수행하는 시스템이다.
- [800] 유통 중계서버는 정보통신산업진흥원이 관리하고 있으며, 모든 유통 메시징서버는 유통 중계서버와 연계하여 P2P 유통과정에서의 오류 시 지원을 받을 수 있다.
- [801] 유통 메시징서버와 유통 중계서버 간 연계 인터페이스는 유통 메시징서버가 유통 중계서버에게 전자문서 전송을 의뢰하기 위한 프로토콜로서, 아래 표 133과 같은 인터페이스로 구분된다.
- [802] 표 133

인터페이스 구분		인터페이스 설명
전송대행	메시지 전송의뢰	☐ 송신자의 유통 메시징서버가 메시지를 전송과정에서 수신자의 시스템 및 네트워크 환경 등에 의해 오류가 발생된 경우, 유통 중계서버에 전송 메시지를 대행 전송하여 줄 것을 요청하는 인터페이스 ☐ 유통 중계서버는 응답메시지로 송신자의 유통 메시징서버가 송신을 시도하였음을 증명하는 송신증명서를 돌려 줌

- [803] 먼저, 유통 메시징서버와 유통 중계서버 간 연계 인터페이스의 공통사항에 대하여 설명하면 다음 ①과 같다.
- [804] ① 요청메시지의 MessageHeader 확장
- [805] 유통 메시징 서버와 유통 중계서버 간 연계 인터페이스 메시지의 첫 번째 MIME Part인 SOAP 메시지 내에는 유통 메시징 서버의 전자서명정보가 포함되어 전달되어야 하며, 유통 중계서버가 SOAP 메시지의 전자서명에 사용된 인증서의 소유자가 해당 유통 메시징서버와 일치하는지를 검증(VID 검증)하는데 필요한 추가적인 유통 메시징서버의 정보(CorpNum, RValue) 또한 포함되어 전달되어야 한다.

- [806] 추가적인 유통 메시징서버의 정보는 SOAP 메시지 내 MessageHeader 요소 하위에 확장 요소(any ##other 위치)로서 위치해야 한다.
- [807] 확장 요소 구조는 아래 표 134와 같으며, 확장 요소 예시는 아래 표 135와 같다.
- [808] 표 134

항목 명	설명	반복 회수	유형	길이
Extension	■ 확장 요소 엘리먼트	1..1		
CorpNum	■ 송신자 사업자등록번호	1..1	String	10
RValue	■ 송신자 공인인증서 개인키에서 추출한 RValue ■ RValue를 Base64로 인코딩 하여 입력해야 함	1..1	String	28

- [809] 표 135

```

<eb:MessageHeader      SOAP:mustUnderstand="1"      eb:id="MessageHeader"
eb:version="2.0">
  <eb:From>
    <eb:PartyId eb:type="ecf_cd">123456789</eb:PartyId>
    <eb:Role>sender</eb:Role>
  </eb:From>
  <eb:To>
    <eb:PartyId eb:type="ecf_cd">ech</eb:PartyId>
    <eb:Role>receiver</eb:Role>
  </eb:To>
  <eb:CPAId>urn:ech-and-ecm-cpa</eb:CPAId>
  <eb:ConversationId>20001209-133003-28572</eb:ConversationId>
  <eb:Service>>urn:ech-service</eb:Service>
  <eb:Action>request</eb:Action>
<eb:MessageData>

<eb:MessageId>20110210-170644Z-00057@127.0.0.18d1f96bf-9cd6-4049-9fdb-
a6c0ed9af46
7</eb:MessageId>
  <eb:Timestamp>2011-02-10T08:06:44.810Z</eb:Timestamp>
</eb:MessageData>
<eb:DuplicateElimination></eb:DuplicateElimination>
<Extention>
  <CorpNum>2208203228</CorpNum>
  <RValue>asdfasdf</RValue>
</Extention>
</eb:MessageHeader>

```

- [810] 이하, 유통 메시징서버와 유통 중계서버 간 연계 인터페이스의 메시지 전송의뢰 인터페이스에 대하여 설명하면 다음과 같다.
- [811] 메시지 전송의뢰 인터페이스는 유통 메시징서버가 타 유통 메시징서버에 메시지를 전송하는 과정에서 타 유통 메시징서버 측의 수신 오류가 발생한 경우, 유통 중계서버에 메시지 전송을 의뢰하고 송신 증명서를 발급받을 때 사용된다. 유통 중계서버는 유통 메시징서버의 메시지 전송의뢰에 대한 접수결과만을 즉시 리턴하며, 수신 유통 메시징서버에 메시지를 전송한 후 수신한 수신증명서는 상술한 “유통증명서 전달 인터페이스”를 사용하여 전송의뢰 유통 메시징서버에 전송한다.
- [812] 메시지 중계 처리 흐름은 도 79와 같다.

[813] 메시지 중계 요청 메시지 포맷은 도 80과 같으며, 도 80과 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 요청 유통메시지가 위치한다. 그리고 사용자가 첨부한 문서가 있는 경우 세 번째 MIME Part부터 위치한다.

[814] 요청 유통메시지 구조는 아래 표 136과 같으며, 실제 예시는 아래 표 137과 같다.

[815] 표 136

항목 명	설명	반복 회수	유형	길이
Request	■ 요청 Root 엘리먼트			
SendMsgReq	■ 메시지 전송 의뢰 엘리먼트			
DocType	■ 유통메시지 유형 - 문서: 0	1..1	Integer	1
Title	■ 메시지 제목	1..1	String	최대 256
Text	■ 메시지 본문 - 송신자에 의해 수신자의 인증 서로 암호화 될 수 있음	0..1	String	-
Sender	■ 송신자 공인전자주소	1..1	String	최대 128
Receiver	■ 수신자 공인전자주소	1..1	String	최대 128
ReqConfir m	■ 열람증명서 요청 - 미요청: 0 - 요청: 1	1..1	Integer	1
IsEncrypt ed	■ 메시지 암호화 여부 - 평문 : 0 - 암호화문 : 1	1..1	Integer	1
Identifie r	■ 본 요청 유통메시지의 고유 식 별값 (UUID)	1..1	String	36

[816] * 표 136에 있어서, 문서 전달 목적으로 본문이 필요없는 경우 Text 생략 가능함.

[817] 표 137

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SendMsgReq>
    <DocType>0</DocType>
    <Title>발송 문서</Title>
    <Text>발송 문서 본문</Text>
    <Sender>#000-0000-0000</Sender>
    <Receiver>#000-0000-0000</Receiver>
    <ReqConfirm>1</ReqConfirm>
    <IsEncrypted>0</IsEncrypted>
    <Identifier>b366ff65-16c8-4d9d-a0ba-d76a2cc95ad2</Identifier>
  </SendMsgReq>
</Request>

```

[818] 메시지 중계 응답 메시지 포맷은 도 81과 같으며, 도 81과 같은 전체 메시지 구조에 있어서 첫 번째 MIME Part에는 SOAP 메시지, 두 번째 MIME Part에는 응답 유통메시지, 그리고 세 번째 MIME Part에는 송신증명서가 위치한다. 만약 요청메시지에 대한 처리과정에서 오류가 발생하였다면, 세 번째 MIME Part는 생성하지 않는다.

[819] 응답 유통메시지 구조는 아래 표 138과 같으며, 실제 예시는 아래 표 139와 같다.

[820] 표 138

항목 명	설명	반복 회수	유형	길이
Response	■ 응답 Root 엘리먼트			
SendMsgRes	■ 메시지 전송 의뢰 응답			
DocType	■ 유통메시지 유형 - 송신증명서 : 2 - 오류 : 9	1..1	Integer	1
RefIdentifier	■ 본 응답 유통메시지에 대응하는 요청 유통메시지의 고유 식별값 (UUID)	1..1	String	36
ErrorCode	■ 오류 코드(유통메시지 유형이 오류(9)인 경우에만 해당 오류 코 드 입력)	0..1	String	256

[821] * 표 138에 있어서, DocType이 오류(9)인 경우는 송신증명서가 위치하게 되는 MIME Part 3을 생성하지 않음

[822] 표 139

```

<Request xmlns="http://www.nipa.kr/eDocument_Circulation">
  <SendMsgRes>
    <DocType>2</DocType>
    <RefIdentifier>b366ff65-16c8-4d9d-a0ba-d76a2cc95ad2</Identifier>
  </SendMsgRes>
</Request>

```

- [823] 이하, 상술한 바와 같은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템 및 이를 이용한 전자문서 유통 방법의 다른 실시예에 대하여 상세히 설명하면 다음과 같다.
- [824]
- [825] [전자문서 유통 시스템의 구조 및 전자문서 유통 프로세스]
- [826] 전자문서 유통은 신뢰유통을 위한 규격을 준수하는 기업/기관이 직접적으로 서로 전자문서를 주고받는 P2P 통신을 기본으로 한다. 이러한 P2P 통신을 수행하기 위한 본 발명에 따른 전자문서 유통 시스템의 기본 요소는 주소정보를 관리하고 있는 주소 디렉터리 서버와 각 송수신 개체들 간 유통을 할 수 있도록 지원하는 표준규격 기반의 유통 메시징서버 시스템이다. 이와 같은 주소 디렉터리 서버와 유통 메시징서버 시스템만 있으면 기업이나 기관이 전자문서를 유통할 수 있는 기본구조는 갖춰진 상태이며, 여기에 송수신자 간 문서유통을 증빙하기 위해 유통증명서를 발급하고, 동시에 이를 제3자 보관기관(공전소, 공인전자문서보관소)에 보관함으로써 유통에 대한 법적 근거를 확보할 수 있다.
- [827] 본 발명에 따른 전자문서 유통 시스템은 상기 기본 요소 외에도, 일반 사용자(기업/기관, 개인)들이 쉽게 전자문서를 유통할 수 있도록 하기 위해서는 문서 송수신 기능에 대한 사용자 인터페이스를 제공하는 유통 클라이언트 어플리케이션(APP), 문서 작성의 편의성을 향상시키기 위해 표준문서양식을 제공하는 전자문서 서식 등록기, 행정기관과 전자문서를 중계하기 위한 공공부문 연계게이트웨이 등이 추가 구성 요소로서 구비된다.
- [828] 상기와 같은 전자문서 유통 시스템에서 발생하는 기본적인 프로세스는 아래의 표 140과 같다.
- [829] 표 140

구분	프로세스	비고
주소정보 등록 및 관리	주소정보 등록	기본 프로세스
	주소정보 변경	기본 프로세스
	주소정보 검색	기본 프로세스
송수신개체들 간 P2P 문서유통	송수신 개체 간 전자문서 전송 및 수신확인	기본 프로세스
	실 사용자와 송수신 개체의 유통 메시징서버 시스템 간 전자문서 송수신 요청	기본 프로세스
	유통증명서 발급 및 공인전자문서보관소에 보관을 요청	기본 프로세스
	유통 클라이언트 APP와 전자문서 서식 등록 기 간 전자문서 Form 검색 및 다운로드	선택적 프로세스
	송/수신 문서에 대해 공인전자문서보관소에 보관을 요청	선택적 프로세스
공공/행정부문과 기업/개인사용간 전자문서 유통	공공연계 게이트웨이를 경유한 문서유통	공공/행정기관에서 유통 메시징서버 규격에 따라 유통시스템을 구축하기 힘든 경우 공공연계 게이트웨이를 경유하여 문서를 유통함

[830]

[831] [전자문서 유통 시스템의 각 구성 요소]

[832] 전자문서유통 시스템을 구성하는 요소를 좀 더 체계적으로 살펴보면 다음과 같다.

[833] 전자문서유통이 이루어지기 위해서는 먼저 유통의 주체가 되는 “① 송수신 개체”가 존재하여야 하며, 각 송수신 개체는 문서를 유통하기 위해 유통 메시징서버 규격을 준수하는 “② 유통 메시징서버 시스템”을 보유하여야 한다. 또한, 전자문서유통의 기본구성요소로 각 송수신 개체 및 사용자의 공인전자주소를 등록, 관리하는 “③ 주소디렉터리 서버”가 존재하여야 한다.

[834] 이러한 기본 구성요소를 바탕으로 사용자에게 유통의 편의성을 제공하기 위해 “④ 유통 클라이언트 APP”가 제공되어야 하며, 행정/공공기관 연계를 지원하는 “⑤ 공공부문 연계 게이트웨이”와 문서의 서식을 관리하는 “⑥ 전자문서 서식 등록기”가 부가적으로 제공되어야 한다.

[835] 상기와 같은 각 구성 요소에 대하여 차례로 상세히 설명하면 다음과 같다.

[836] ① 송수신 개체

[837] 전자문서유통의 기반 인프라 구성요소 중 유통의 기준이 되는 단위가 송수신 개체인데, 송수신 개체는 유통에 참여하는 역할에 따라 송신자(Sender) 또는 수신자(Receiver)을 같이 수행하게 되며, 이 개체들은 유통 메시징서버 시스템을 통해 유통 프로토콜 규격에 따라 문서(정보)를 유통한다.

[838] 유통에 참여하는 모든 송수신 개체는 유통메시징 규격에 따라 문서를 송수신할 수 있는 유통 메시징서버 시스템을 구축한 후, 유통 메시징서버 시스템의 물리적 주소정보를 주소디렉터리 서버에 등록함으로써 전자문서유통에 참여할 수 있는 기반을 만들게 된다. 이때, 각 송수신 개체들은 하위에 1개 이상의 공인전자주소를 갖는 실 유통 사용자를 갖게 된다.

[839] 전자문서유통에서 송수신 개체로서 인정받을 수 있는 개체는 메시징서버 규격을 준수한 시스템을 구축한 후, 정보통신산업진흥원에 의해 표준적합성과 상호운용성을 인증받은 개체로 한정되며, 유통을 증명하기 위해서는 ① 인증 받은 송수신 개체를 통해 전자문서가 유통된 후, ② 표준규격에 맞게 유통증명서를 발급하여 제3자 보관기관에 보관하여야 한다.

[840] 이때, 송수신 개체는 전자문서에 대한 법적 소유자 및 책임자로서 직접 전자문서 전송을 책임지는 개체와, 유통되는 전자문서의 실 소유자이며 책임자인 사용자를 위해 전자문서를 대행해 주는 개체로 구분된다. 전자문서의 소유자가 직접 전자문서를 전송하는 송수신 개체인 경우에는 유통 메시징서버 시스템의 표준적합성과 상호운용성을 인증받고, 유통증명서를 안전하게 제3자 보관기관에 보관하는 것만으로도 송수신 개체에 참여할 수 있다.

[841] 그러나 전자문서 소유자(사용자)를 대행해서 3자 전송을 책임져야 하는 송수신 개체인 경우에는 송수신 개체가 안전하고 신뢰성 있는 방법으로 전송메시지를 관리하고, 사용자 정보를 관리하고 인증하는지에 대한 부분까지 입증하여야 한다. 3자 유통의 안전성 및 신뢰성을 보장하기 위해 한시적으로 이러한 3자 유통이 가능한 송수신 개체로는 제3자 보관기관 사업자만으로 한정한다.

[842] ② 유통 메시징서버 시스템

[843] 유통 메시징서버 시스템은 유통 메시징서버 규격을 바탕으로 전자문서(정보)를 유통하기 위해 메시징 송수신기능과 주소 디렉터리 서버와 연계하여 수신자에 대한 주소정보 및 보안 관련한 정보를 검색하는 기능을 반드시 구축하여야 한다. 유통 메시징서버 시스템은 물리적으로 하나의 전자주소(IP Address)를 가지나 하위의 사용자를 위해 복수의 사용자 계정을 발급하고 관리할 수 있으며, 사용자 계정은 각각 하나의 공인전자주소를 갖게 된다.

[844] 유통 메시징서버 시스템은 각 사용자 계정을 관리하기 위해 사용자 계정별 전자문서 사서함을 관리하여야 하며, 유통 메시징서버 시스템은 이 사용자 계정을 대표하여 안전하고 신뢰성 있는 전자문서유통 책임을 갖게 된다.

[845] 이와 같은 유통 메시징서버 시스템이 전자문서유통 내에 송수신 개체로서 참여하기 위해서는 본 발명에 따른 요건에 적합하게 구현이 되어 있는지, 타 솔루션과 상호 운용에는 문제가 없는지를 인증 받는 과정을 거쳐야 한다.

[846] 유통 메시징서버 시스템에 대한 표준 적합성 및 상호운용성을 인증해 주는 인증 시스템은 인증된 송수신 개체를 관리하여야 하며, 주소 디렉터리 서버가 공인전자주소를 등록하는 과정에 인증 통과 여부 확인을 요청하면 그 결과를

돌려주어야 한다.

- [847] 유통 메시징서버 시스템이 인증을 받아 공인전자주소로서 등록을 하기 위해서는 도 82 및 아래와 같은 절차를 따라야 한다.
- [848] 먼저, 송수신 개체가 되고자 하는 기업/기관 또는 개인사용자는 본 기술규격에 적합하게 유통 메시징서버 시스템을 구축한다.
- [849] 다음으로, 인증 테스트 베드가 제공하는 자동화된 검증도구를 통해 구축된 유통 메시징서버 시스템의 표준적합성 및 상호운용성을 검증한다.
- [850] 다음으로, 자체 검증을 모두 완료한 송수신 개체는 인증테스트베드에 인증시험을 요청한다.
- [851] 다음으로, 인증 테스트 베드의 테스트 절차에 따라 시스템에 대한 인증을 마친 후 결과가 “통과”로 나오면 송수신 개체는 공인전자주소 등록을 위한 다음 절차를 준비한다.
- [852] 다음으로, 인증 테스트 베드는 인증심사를 통과한 송수신 개체에 대한 정보를 주소 디렉터리 서버에 전달하고, 주소 디렉터리 서버는 이 정보를 주소 등록의 조건으로 활용한다.
- [853] 다음으로, 송수신 개체는 인증 통과된 유통 메시징서버 시스템을 등록하기 위해 주소 디렉터리 서버에 고유의 ID 발급을 신청한다.
- [854] 다음으로, 유통 메시징서버 시스템이 주소디렉터리 서버에 등록이 완료되면, 유통 메시징서버 시스템은 전자문서유통에 참여할 수 있게 된다.
- [855] 다음으로, 유통 메시징서버 시스템의 인증이 완료되면 사용자 계정을 개설하고 대표 공인전자주소인 경우에 사용자 계정은 공인전자주소에 등록요청을 한다.
- [856] **③ 주소디렉터리 서버**
- [857] 신뢰할 수 있는 전자문서유통에 참여하기 위해서 모든 사용자는 고유의 전자주소를 발급받아야 한다.
- [858] **④ 유통 클라이언트 APP**
- [859] 유통 클라이언트 APP는 문서유통에 참여하는 사용자들을 위해 문서 송신 및 수신, 수신문서 열람 및 관리 등의 UI를 제공하는 어플리케이션을 지칭한다. 유통 클라이언트 APP는 독자적으로 문서를 송수신할 수 없으며 반드시 유통 메시징서버 시스템과 연계되어야 한다.
- [860] 유통 클라이언트 APP에서 작성되거나 첨부된 문서는 유통 메시징서버 시스템에 전달되어 전송을 요청하게 되며, 유통 메시징서버 시스템을 통해 수신된 문서를 조회하게 된다. 유통 메시징서버 시스템이 사용자 계정을 통해 송수신 사서함을 관리하는 경우라면 유통 클라이언트 APP는 수신문서 중 사용자 계정정보 확인을 통해 해당 문서에만 접근이 가능하다.
- [861] 유통 클라이언트 APP는 사용자의 요구에 의해 C/S 형태의 어플리케이션으로 구현될 수도, 웹 형태의 화면으로 구현될 수도 있다.
- [862] **⑤ 공공부문 연계 게이트웨이**
- [863] 전자문서유통을 수용하기 어려운 행정 및 공공기관의 경우 공공부문 연계

게이트웨이를 통해 행정, 공공기관과 전자문서유통 체계 하의 민간기업, 기관, 개인들 간 문서를 중계하는 역할을 수행한다.

[864] **㉔ 전자문서 서식 등록기**

[865] 전자문서 서식 등록기는 유통 메시징서버 시스템을 이용해서 전자문서를 전송하고자 하는 사용자들이 Office 도구를 이용해서 직접 전송문서를 작성할 수도 있으나, 사용자가 좀 더 쉽게 전자문서 생성할 수 있도록 지원하기 위해 표준문서 양식을 등록하고 관리하면서 유통 클라이언트 APP와 같은 사용자용 어플리케이션이 이용할 수 있도록 문서 양식 등록 및 관리, 문서 양식 검색, 열람 및 다운로드, 문서 양식 삭제 등의 관리를 지원하는 시스템이다.

[866] 전자문서 서식 등록기는 문서표준양식을 관리하는 서버 엔진과 클라이언트 어플리케이션(APP)이 이를 검색하고, 다운로드 받은 후에 내부 프로그램에 플러그인(Plug-In)하여 사용할 수 있도록 하는 표준인터페이스를 제공한다.

[867]

[868] **[전자문서 유통 방법]**

[869] 전자문서유통에서 전자문서를 유통하기 위한 전체 프로세스는 “① 유통 전 사전 준비단계”, “② 전자문서 유통 단계”, “③ 유통을 위한 증빙 단계”의 3단계로 크게 구분해서 볼 수 있다. 이하에서는 상기 세 단계에 대한 상세한 설명과 함께, “문서 송수신 방안”과 “유통증명 방안”과 “스팸 메시지 처리 방안”에 대하여 상세히 설명하도록 한다.

[870] **① 유통 전 사전 준비단계**

[871] - 전자문서 서식 등록기 관리자는 전자문서 서식 등록기를 이용해서 사용할 표준문서 서식을 등록한다.

[872] - 송수신 참여자는 자체적으로 신뢰유통을 위한 유통 메시징서버 시스템을 구축할 지, 기 구축된 유통 메시징서버 시스템에 사용자 계정을 개설하고 사용할 지 여부를 결정한다. 자체적으로 신뢰유통을 위한 유통 메시징서버 시스템을 구축하는 경우에는, 전자문서 송수신을 위한 유통 메시징서버 시스템을 구축한 후에, 인증기관을 통해 유통 메시징서버 시스템의 표준적합성, 상호운용성에 대한 인증테스트를 수행한 후에, 주소 디렉터리 서버에 접속한 후 인증된 유통 메시징서버 시스템을 위한 송수신 개체 ID를 신청하고 발급받은 후에, 내부의 실 사용자를 위해 자체적으로 내부 구분자를 등록하고 관리하며, 표준문서 서식 기반의 문서 작성 기능을 이용하기 위해 일반사용자를 위한 클라이언트 어플리케이션에 표준문서 서식 작성기능을 플러그인(Plug-In)한다(선택적 사항). 반대로, 3자 유통이 가능한 유통 메시징서버 시스템을 보유한 송수신 개체를 이용하는 경우에는, 유통 메시징서버 시스템을 통해 기업/기관/개인을 위한 사용자 계정 개설을 요청한 후에, 사용자 계정에 대한 공인전자주소 정보를 주소 디렉터리 서버에 등록한 후에, 표준문서 서식 기반의 문서 작성기능을 이용하기 위해 일반사용자를 위한 클라이언트 어플리케이션에 표준문서양식 작성기능을 플러그인(Plug-In)한다(선택적 사항).

[873] **② 전자문서 유통 단계**

[874] ○ 문서 송신자

[875] - 문서 송신자는 유통할 문서를 선택하거나 문서작성기를 통해 전송할 문서를 작성한다.

[876] - 문서 수신 상대방 주소정보 및 전달 문서, 문서 암호화여부 및 전자서명 여부를 선택한다. (암호화 및 전자 서명은 전송 메시지가 아닌 첨부하는 전달 문서를 대상으로 하며, 이 절차는 선택적 사항임)

[877] - 유통 클라이언트 APP는 주소 디렉터리 서버에 연계하여 수신 상대방의 공인 전자주소를 기반으로 물리적 주소정보 및 암호화를 위한 공개키 정보를 획득한다. (선택적 사항으로, 유통 클라이언트 APP가 물리주소 획득을 하지 않은 경우 유통 메시징서버가 이 작업을 수행함)

[878] - 유통 클라이언트 APP는 유통 메시징서버에 수신자의 주소정보를 기반으로 전송요청을 한다. (물리적 주소정보 또는 공인 전자주소 모두 가능)

[879] ○ 송신자의 유통 메시징 서버

[880] - 유통 클라이언트 APP에서 요청한 전송요청 메시지가 수신자에 대한 물리적 주소정보가 아닌 경우, 유통 메시징서버는 공인전자주소를 기반으로 수신자의 송수신 개체에 대한 물리적 주소정보를 주소 디렉터리 서버에 질의한다.

[881] - 전자문서를 유통 프로토콜 규격에서 정의한 메시지 구조체로 패키징한다.

[882] - 송신자 유통 메시징서버의 공인인증서 기반으로 메시지에 전자서명을 한다.

[883] - 수신자의 물리적 주소정보로 메시지를 전송한다.

[884] ○ 수신자의 유통 메시징 서버

[885] - 메시지 수신 후, 수신메시지 검증하고 메시지에서 문서를 추출한다.

[886] - 동기식 응답으로 수신증명서를 포함한 메시지를 송신자에게 전송한다.

[887] **③ 유통을 위한 중빙 단계**

[888] - 수신자는 문서 수신사실 확인을 위한 문서 수신시점에 “수신증명서”를 생성한 후 송신자에게 전달하여야 하며, 이를 수신한 문서 송신자는 “수신증명서”를 제3자 보관기관에 보관한다.

[889] - 송신자가 요구할 경우, 수신자는 수신문서를 실 문서 담당자(사용자)에게 전달한 후, 담당자가 수신문서를 확인한 시점에 “열람증명서”를 생성하여 송신자에게 전달하며, “열람증명서”를 수신한 문서 송신자는 “열람증명서”를 제3자 보관기관에 보관한다. (열람증명서의 발급은 송신자의 요청이 있을 경우에만 적용됨)

[890] - 송신자가 수신자에게 문서전달을 시도하였으나 실패한 경우에는 송신 시도에 대해 중빙을 하고자 객관적 3자인 전자문서 유통허브에 문서 전송을 의뢰하며, 전송의뢰를 받은 전자문서 유통허브는 전송의뢰를 받았음을 입증하고자 “송신증명서”를 발급하여 송신의뢰자에게 전달하고 이를 수신한 송신의뢰자는 “송신증명서”를 제3자 보관기관에 보관한다.

[891] **○ 문서 송수신 방안**

- [892] 유통 메시징서버 시스템을 통해 송신자와 수신자는 문서를 전자적으로 유통합니다. 유통 메시징서버 시스템은 유통 프로토콜에 따라 전자문서를 송수신하며, 신뢰메시지 유통을 위해서 모든 메시지는 전송과 수신확인(또는 수신증명서) 메시지의 조합으로 이루어지며, 수신자에 대한 물리적 주소정보는 주소 디렉터리 서버를 통해 획득하게 된다.
- [893] ○ 유통증명 방안
- [894] "유통증명"이라 함은 전자문서 유통과 관련된 송신, 수신, 열람에 대하여 해당 사실을 신뢰성 있는 방법으로 증명하는 것을 말하는 것으로, 이때 전자문서 유통과 관련된 행위에 대하여 발급하는 증명서를 통칭하여 "유통증명서"라고 한다.
- [895] 유통 메시징서버 시스템은 송신, 수신에 대한 행위 입증을 위해 송신 및 수신 시점에 유통증명서를 발급하고, 발급한 유통증명서를 공인 전자문서 제3자 보관기관에 보관함으로써 유통 행위에 대한 증빙 자료로서 활용할 수 있도록 한다.
- [896] 유통 메시징서버 시스템은 전자문서 송신, 수신, 열람에 대한 사실을 증명하며 각 사실에 대한 유통증명서를 생성하며, 유통증명서는 유통증명서 식별정보, 유통증명서 생성시각 및 만료시각, 유통증명서 정책 및 유통증명 대상을 포함한다.
- [897] 전자문서 송신에 대한 유통증명서는 전자문서 유통허브가 생성하며 유통증명 대상에 발신자 식별정보, 수신자 식별정보, 유통 식별정보, 문서 식별정보, 전자문서 송신의뢰 시각을 포함한다.
- [898] 전자문서 수신에 대한 유통증명서는 전자문서를 수신한 수신자가 생성하며 유통증명 대상에 발신자 식별정보, 수신자 식별정보, 유통 식별정보, 문서 식별정보, 전자문서 송신 시각, 전자문서 수신 시각을 포함한다.
- [899] 전자문서 열람에 대한 유통증명서는 전자문서를 수신 확인한 사용자가 생성하며 유통증명 대상에 발신자 식별정보, 수신자 식별정보, 유통 식별정보, 문서 식별정보, 전자문서 송신 시각, 전자문서 수신 시각, 전자문서 수신확인 시각을 포함하여야 한다.
- [900] 이와 같이 생성된 유통증명서는 NPKI 또는 GPKI 인증서로 전자서명 되어야 하고, 생성된 유통증명서는 전자문서 송신자에게 전달되어야 하며, 모든 유통증명서는 제3자 보관기관에 보관되는 것이 바람직하다.
- [901] ○ 스팸 메시지 처리 방안
- [902] 전자문서유통은 기본적으로 송신자가 인증된 유통 메시징서버 시스템을 통해 전송을 하고, 수신자 역시 이를 기본으로 수신하므로 스팸을 발송하였을 때 전송자에게 책임을 물을 수 있는 기반구조를 가진다. 그러나, 스팸 발송자가 유통 메시징서버 시스템에 사용자 계정을 개설하고 이를 이용해서 전송을 하는 경우가 있을 수 있다. 또한, 현재의 인증방식이 시스템의 기술적 내용에 대한 인증만을 대상으로 하고 있어서, 스팸발송자가 유통 메시징서버 시스템을

- 구축하고 이를 기술적으로 인증한 후 스팸 발송수단으로 사용하였을 때, 초기에 원천적으로 봉쇄하기가 쉽지 않은 상황이다.
- [903] 따라서, 이와 같은 문제점을 해결하기 위해 본 발명에 따른 표준문서 유통 인프라에서는 인증목록 관리 기반의 화이트리스트, 사용자 신고방식에 의한 스팸대상 목록 관리 기반의 블랙리스트를 체계를 제공하며, 이러한 체계에 의해 수신자가 수신거부를 할 수 있는 프로세스를 적용하여 스팸메시지를 방지할 수 있도록 한다.
- [904] 스팸메시지 신고 및 송신 상대방에 대한 확인을 위한 기능은 필수 기능으로 모든 유통 메시징서버는 이 기능을 반드시 구축하여야 한다.
- [905] 수신자는 수신한 메시지가 스팸 메시지라고 판단이 되면, 도 83에 도시한 바와 같은 프로세스에 의해 스팸메시지를 전자문서 유통허브의 주소 디렉터리 서버에 신고하며, 이와 관련한 처리 절차는 다음과 같다.
- [906] 먼저, 수신자가 메시지를 수신한 시점에서 스팸메시지라고 판단되면, 수신자는 유통 메시징서버 시스템을 통해 주소 디렉터리 서버에 해당 메시지를 수신 메시지로 신고한다.
- [907] 다음으로, 유통 메시징서버 시스템으로부터 스팸메시지 신고를 접수한 주소 디렉터리 서버는 접수되었음에 대한 확인메시지를 되돌려준다.
- [908] 다음으로, 주소 디렉터리 서버를 관리하는 주체인 정보통신산업진흥원은 해당 메시지를 분석하고, 송신자에 대한 조사를 통해 송신자의 공인전자주소에 대해 블랙리스트 추가 여부를 심사하고 판단한다.
- [909] 다음으로, 최종적으로 블랙리스트 대상자로 확정이 되면, 주소 디렉터리 서버는 해당 공인전자주소를 블랙리스트에 추가한 후, 송신자에게 블랙리스트 추가에 대한 내용을 통지한다.
- [910] 다음으로, 주소 디렉터리 서버는 스팸메시지 요청에 대한 처리 결과를 스팸신고자(수신자)에게 전달한다.
- [911] 상기와 같은 처리 절차에 있어서 화이트리스트는 송신 유통 메시징서버 시스템이 인증을 받고 정식으로 등록된 메시징서버 시스템에 대한 정보만 기록되고, 블랙리스트는 전송자의 주소가 스팸발송자로 등록된 경우에 등록되며, 동일한 유통 메시징서버 시스템을 통해 블랙리스트에 등록되는 스팸 주소가 중복 발생되는 경우에는 전자문서 유통허브에서 해당 유통 메시징서버 시스템에 대한 인증취소 여부를 판단한 후, 인증을 취소하고 화이트리스트에서 삭제할 수 있다.
- [912] 수신자는 메시지 수신 시, 송신 상대방이 신뢰할만한 정당한 사용자인지 여부를 확인하기 위해 주소디렉터리 서버의 화이트리스트와 블랙리스트를 확인한 후, 수신거부를 할지의 여부를 결정한다. 송신자에 대한 확인은 수신시점에 실시간 확인을 하거나, 수신자의 유통 메시징서버 시스템에 Cache 형태로 관리하고 있는 목록을 통해 확인하는 주기적 확인 방법이 있다.
- [913] 실시간으로 송신자에 대한 확인을 수행하는 프로세스는, 도 84에 도시한 바와

같이 수신자가 메시지를 수신하는 시점에 주소 디렉터리 서버에 송신자의 주소가 화이트리스트, 블랙리스트에 등록되었는지의 여부를 판단한 후에 메시지에 대한 수신거부 여부를 결정하며, 이와 같이 실시간으로 송신자에 대한 확인을 수행하는 프로세스의 상세한 처리절차는 다음과 같다.

- [914] 먼저, 수신자의 유통 메시징서버 시스템은 메시지를 수신하면 주소 디렉터리 서버에 정당한 사용자인지 여부를 확인하기 위해 확인요청 메시지를 전달한다.
- [915] 다음으로, 주소 디렉터리 서버는 요청받은 사용자의 주소정보가 화이트리스트에 포함되어 있는지 여부를 확인한다.
- [916] 다음으로, 해당 주소가 화이트리스트에 없으면 주소 디렉터리 서버는 바로 확인 요청자에게 등록되지 않은 사용자임을 결과메시지로 돌려주고, 화이트리스트에 있으면 다시 해당주소가 블랙리스트에 등록된 주소인지 여부를 확인한다.
- [917] 다음으로, 주소 디렉터리 서버는 확인 요청자에게 블랙리스트 등록 여부에 대한 결과 메시지를 돌려준다.
- [918] 다음으로, 수신자는 주소 디렉터리 서버로부터 송신자가 정당한 사용자가 아니라는(화이트리스트에 없거나 블랙리스트에 등록된 경우) 결과메시지를 받은 경우에는 수신메시지를 자체적으로 스팸메시지로 처리한 후, 주소디렉터리 서버로부터 받은 처리결과 메시지와 스팸메시지 수신이력을 기록하고 보관한다.
- [919] 다음으로, 스팸메시지에 대한 처리 이력은 반드시 한 달 이상 보관함으로써, 해당 송신자에 대한 수신거부의 정당성을 확인할 수 있도록 한다.
- [920] 그리고, 주기적으로 송신자에 대한 확인을 수행하는 프로세스는 도 85에 도시한 바와 같이 수신자는 사전에 주소 디렉터리 서버로부터 화이트리스트와 블랙리스트를 받아서 자체적으로 관리하고 이를 기반으로 송신자의 주소가 화이트리스트, 블랙리스트에 등록되었는지 여부를 판단한 후 메시지에 대한 수신거부 여부를 결정하며, 이와 같이 주기적으로 송신자에 대한 확인을 수행하는 프로세스의 상세한 처리절차는 다음과 같다.
- [921] 먼저, 수신자의 유통 메시징서버 시스템은 미리 주소 디렉터리 서버로부터 최신의 화이트리스트와 블랙리스트를 요청한 후 자체적으로 관리한다. 이때, 리스트의 변동사항 발생 시 자동 통지 요청 여부를 같이 전달한다. 이와 같은 변동사항 발생 자동통지를 요청한 경우에도 주소 디렉터리 서버에 최신 리스트를 가져오기 위한 요청은 주기적으로 함으로써 리스트 정보가 최대한 1일 이상 차이가 나지 않도록 한다.
- [922] 다음으로, 주소 디렉터리 서버는 화이트리스트 및 블랙리스트에 변동사항이 발생하면 변동 통지요청을 한 사용자에게 변동내역을 브로드캐스팅(broadcasting)한다.
- [923] 다음으로, 리스트에 대한 변동사항을 받은 사용자 유통 메시징서버 시스템은 자체 관리하는 리스트의 정보를 수정함으로써 동기화를 시켜준다.

- [924] 다음으로, 수신자는 메시지를 수신하면 주소 디렉터리 서버에 정당한 사용자인지 여부를 확인하기 위해 자체 관리하는 리스트를 확인한다.
- [925] 다음으로, 수신자는 자체 관리하는 리스트 점검결과 송신자가 정당한 사용자가 아니라고(화이트리스트에 없거나 블랙리스트에 등록된 경우) 판단한 경우에는 수신메시지를 자체적으로 스팸메시지로 처리한 후, 스팸메시지 수신 이력을 기록하고 보관한다.
- [926] 다음으로, 스팸메시지에 대한 처리 이력은 반드시 한 달 이상 보관함으로써, 해당 송신자에 대한 수신거부의 정당성을 확인할 수 있도록 한다.
- [927]
- [928] **[유통 메시징서버 시스템]**
- [929] 이하에서는, 상술한 바와 같은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템의 유통 메시징서버 시스템과 관련하여 상세히 설명하도록 한다.
- [930] 유통 메시징서버 시스템은 크게 메시지 송신, 메시지 수신, 수신 메시지에 대한 사서함 관리, 메시지 보안(사용자 인증, 문서 암호/복호화 등), 송수신 이력관리, 주소디렉터리 서버 연계, 메시지 검증, 내부시스템 연계인터페이스, 유통증명서 발급 및 관리, 제3자 보관기관 연계 등으로 구성된다.
- [931] 도 86에는 유통 메시징서버 시스템의 구조를 도시하였으며, 이와 같은 도 86을 참조하여 유통 메시징서버 시스템의 구성 요소 각각(①~⑨)에 대하여 상세히 설명하면 다음과 같다.
- [932] **① 메시지 송수신**
- [933] - 유통 프로토콜에 따라 메시지를 송신하고 수신한다.
- [934] **② 사용자 별 계정(사서함) 관리**
- [935] - 송수신한 메시지를 사용자 계정 또는 내부구분자에 따라 계정별 사서함에 보관한다.
- [936] - 사서함에 보관한 송신문서에 대하여 "송신 중", "송신 완료", "송신 실패", "담당자 수신 완료"를 포함하는 4단계의 상태정보를 관리한다. 이때, "송신 중" 상태는 문서 전송 후 수신자로부터 아무런 응답을 받지 못한 상태이며, "송신 완료" 상태는 수신자의 유통 메시징서버 시스템으로부터 "수신증명서"를 받은 상태이고, "송신 실패" 상태는 수신 유통 메시징서버 시스템 내부에서 오류가 발생하여 SOAP Fault 메시지를 리턴하거나, 송수신 과정에서 네트워크 오류가 발생한 경우이며, "담당자 수신 완료" 상태는 송신 유통 메시징서버 시스템이 수신자로부터 담당자의 문서를 확인했음을 증명하는 "열람증명서"를 받은 경우이다.
- [937] - 사용자 계정별 사서함에 보관된 수신문서에 대하여 "검증오류", "수신 확인 전", "수신 확인", "열람 확인"을 포함하는 4단계의 상태정보를 관리한다. 이때, "검증오류" 상태는 수신한 메시지에 대한 기본구조 검증에서 오류 발생한 상태이며, "수신 확인 전" 상태는 수신문서 담당자가 사서함의 수신 문서 목록을 읽기 전이고, "수신 확인" 상태는 수신문서 담당자가 사서함의 수신 문서 목록을

읽은 상태이며, "열람 확인" 상태는 수신문서 담당자가 수신문서에 대한 상세내용을 열람한 상태로서 이 시점에 수신자의 유통 메시징서버 시스템은 "열람증명서"를 발급한 후에 송신자에게 전달한다.

- [938] - 수신사용자에 의해 삭제 요청이 오면 해당 수신문서를 물리적으로 삭제 처리한다.
- [939] - 사서함에서 송신문서, 송신에 대한 수신확인 메시지, 수신담당자의 수신확인 메시지는 서로 연관될 수 있도록 연관정보를 가진다.
- [940] ③ 주소 디렉터리 서버 연계
- [941] - 주소 디렉터리 서버가 제공하는 주소정보 등록 및 검색 프로세스에 따라 주소정보를 관리한다.
- [942] - 주소 디렉터리 서버가 제공하는 서비스를 호출할 수 있는 클라이언트 기능을 포함한다. 즉, 주소 디렉터리 서버가 제공하는 주소정보 등록, 검색, 수정, 삭제 기능을 원격에서 호출하는 서비스 클라이언트 기능을 제공한다.
- [943] ④ 메시지 보안(사용자 인증, 문서 암호/복호화 등)
- [944] - 유통 프로토콜에서 제시하는 메시지 보안 기능(메시지 전자서명, 서명 검증)을 기본적으로 수행한다.
- [945] ⑤ 송수신 이력 관리
- [946] - 유통 메시징서버 시스템은 송수신 이력에 대해 최소 1년 이상은 반드시 보관/관리한다.
- [947] - 보관하는 송수신 이력에 대한 정보는 송신 이력과 수신 이력이며, 송신 이력은 메시지 id, 연관메시지 id, 송신자(사용자 계정 포함), 수신자, 송신시간, 송신 문서에 대한 해쉬 값을 포함하며, 수신이력은 송신자, 수신자(사용자 계정 포함), 수신시간, 수신 문서에 대한 해쉬 값을 포함한다.
- [948] ⑥ 유통증명서 발급 및 관리
- [949] - 유통 메시징서버 시스템은 문서 송수신 사실에 대한 내용을 증빙할 수 있도록 유통증명서를 발급하고 관리한다.
- [950] - 발급된 유통증명서는 전달받자마자 제3자 보관기관에 보관 의뢰함으로써 그 신뢰성을 보장받는다.
- [951] - 발급된 후 제3자 보관기관에 보관된 유통증명서 이력을 관리하며, 유통증명서 발급 이력은 유통증명서 id, 유통증명서 발급 시각, 관련 메시지 id, 유통증명서 원본(선택적), 제3자 보관기관 보관 후 수신한 보관-key정보를 포함한다.
- [952] ⑦ 메시지 패키지 처리(Packaging, Parsing, Extracting)
- [953] - 송신 문서를 전송 전에 유통 프로토콜에 정의된 메시지구조로 패키징(Packaging) 한다.
- [954] - 수신한 문서를 유통 프로토콜에 정의된 메시지구조에 의해 파싱(Parsing, 구문해석)하고 필요한 정보를 추출(Extracting)한다.
- [955] ⑧ 유통증명서 보관요청

- [956] - 일반 송수신 개체가 유통증명서의 보관요청을 위해서는 제3자 보관기관의 유통 메시징서버 시스템에 제3자 보관기관 보관요청메시지를 전송한다.(원격 보관요청)
- [957] - 제3자 보관기관의 유통 메시징서버 시스템은 공인전자문서보관소 보관요청 메시지를 수신하면, 제3자 보관기관에 유통증명서 보관을 위한 보관요청 Client를 호출한다.
- [958] - 제3자 보관기관의 유통 메시징서버 시스템이 직접 유통증명서를 생성한 경우에는 생성시점에 제3자 보관기관 보관요청 Client를 직접 호출한다.(로컬 보관요청)
- [959] - 유통증명서 보관요청을 위한 Client는 제3자 보관기관의 송수신 연계인터페이스 규격에 따라 제3자 보관기관에 보관을 요청한다.
- [960] ⑩부가 서비스
- [961] - 유통 클라이언트 APP 관리의 배포, 버전 관리 등을 수행한다.
- [962] - 메시지 유통관리(이력, 통계정보 등)를 수행한다.
- [963] - 시스템 관리(시스템 모니터링, 환경정보 등)를 수행한다.
- [964] - 문서양식(Form) 관리를 수행한다.
- [965]
- [966] 상기와 같은 ①-⑩의 구성 요소를 가지는 본 발명에 따른 유통 메시징서버 시스템을 도 87과 같이 제3자 보관기관에 적용한 경우에는, 유통증명서를 보관할 시에 유통증명서 보관요청 모듈은 제3자 보관기관 연계인터페이스 규격에 따라 개발된 연계인터페이스 클라이언트를 호출하여 보관요청을 한다.
- [967] 상기와 같은 ①-⑩의 구성 요소를 가지는 본 발명에 따른 유통 메시징서버 시스템을 도 88과 같이 일반 송수신 개체(일반 사업자)에 적용한 경우에는, 유통증명서를 보관할 시에 제3자 보관기관 사업자의 유통 메시징서버 시스템에 유통증명서 보관을 요청하는 메시지를 전송하고 처리 결과를 받아오는 방식으로 처리한다.
- [968]
- [969] 상기와 같은 ①-⑩의 구성 요소를 가지는 본 발명에 따른 유통 메시징서버 시스템을 이용하여 송신자와 수신자 간에 직접 메시지를 유통하는 프로세스는, "①수신자에 대한 물리적 주소 및 보안정보 획득", "②메시지 전송 및 전송확인", "③업무수신자의 수신확인", "④유통증명서 발급 및 보관"을 포함하는 4단계로 이루어지는데, 이와 같은 4단계와 관련하여 도 89를 참조하여 상세히 설명하면 다음과 같다.
- [970] ①수신자에 대한 물리적 주소 및 보안정보 획득
- [971] - 송신자의 시스템은 상대방에 대한 주소 정보를 바탕으로 실제 메시지가 전달되어야 할 물리적 주소 정보 및 보안정보(송신메시지에 대한 수신암호를 필요로 하는 경우)를 주소 디렉터리 서버에 요청함으로써 이를 획득한다.
- [972] - 수신자에 대한 물리적 주소 및 보안 정보는 유통 클라이언트 APP가 주소

디렉터리 서버에 요청한 후에 수신자의 물리적 주소 정보를 유통 메시징 서버에 전달하도록 한다.

- [973] - 사용자에 대한 id(예: 주민등록번호, 사업자 등록번호 등)만으로 수신자에 대한 주소정보를 획득할 수도 있으나, 이 경우에는 수신자가 송신자에게 id 기반의 주소정보 검색을 허용한 경우에만 가능하다.

- [974] * 송신자가 수신자의 물리적 주소 정보 및 보안 정보를 이미 알고 있는 경우에 이 절차는 생략이 가능하다.

[975] **② 메시지 전송 및 전송 확인**

- [976] - 송신자는 메시지를 유통 프로토콜 규격에 맞게 패키징 한 후, 유통 메시징서버 시스템의 공인인증서를 기반으로 전자서명을 수행한다.

- [977] - 유통 메시징서버 시스템은 앞서 획득한 물리적 주소에 패키징하고 전자서명된 메시지를 전송한다.

- [978] - 메시지를 수신한 수신 유통 메시징서버 시스템은 메시지의 기본 패키징 구조, 전자서명에 대한 유효성, 송신자에 대한 적합성(검증에 대한 상세 내용은 "2.4.6 메시지 검증"부분 참조)을 검증한 후, 수신확인을 위한 수신증명서 또는 오류메시지를 생성한다.

- [979] - 수신 유통 메시징서버 시스템은 생성한 응답메시지를 송신자에게 전송한다.
- [980] - 전송과 전송확인 과정은 동기식 메시지 처리로 이루어진다.

[981] **③ 업무수신자의 수신 확인**

- [982] - 송신자가 메시지 전송시점에 업무 수신자의 담당자 열람확인 메시지를 요청한 경우에는 수신자는 메시지에 대한 업무적 수신확인 시점에 송신자에게 반드시 담당자 열람확인을 증명할 수 있는 열람증명서를 생성하여 이를 전송하여야 한다.

- [983] - 수신자가 메시지 송신자에게 담당자 열람확인을 위한 열람증명서 메시지를 보내면 원 메시지 송신자를 이에 대한 수신확인 메시지를 동기식으로 보내준다.

[984] **④ 유통증명서 발급 및 보관**

- [985] - 각 단계별로 유통에 대한 증빙을 받고자 하는 경우 송신자는 각 단계에 따라 수신, 열람, 송신에 대한 증명서를 발급하고 이를 제3자 보관기관에 보관함으로써 유통에 대한 법적 증빙의 근거를 확보한다.

[986]

본 발명에 따른 유통 메시징서버 시스템을 이용하여 송신자와 수신자 간에 직접 메시지를 유통하는 프로세스는 상기와 같은 "①수신자에 대한 물리적 주소 및 보안정보 획득", "②메시지 전송 및 전송확인", "③업무수신자의 수신확인", "④유통증명서 발급 및 보관"외에 "⑤오류 처리"도 수행하는데, 도 90 내지 도 92를 참조하여 오류 처리 기능과 관련하여 상세히 설명하면 다음과 같다.

[988] **⑤ 오류 처리 기능**

- [989] 유통 메시징서버 시스템의 모든 메시지 송수신 프로세스는 동기식 처리를 기본으로 한다. 따라서, 전송에 대한 모든 오류는 전송자가 확인 가능하므로

- 재전송하는 것을 기본으로 하며, 동일한 메시지 전송은 동일한 MessageId 값을 설정하여 다시 보내도록 함으로써 수신자가 수신 성공 후 수신확인 메시지 전송과정에서의 오류에 대해서도 중복메시지 수신을 감지할 수 있도록 한다.
- [990] 유통 메시징서버 시스템은 요청메시지 송신에 실패한 경우에도 90과 같은 처리 흐름도를 따른다. 즉, 메시지 송신자가 전송하는 과정에서 네트워크 오류 등에 의해 전송오류가 발생한 경우에 송신자는 HTTP 오류와 같은 오류 메시지를 받게 되면 동일한 메시지를 다시 재전송 요청하며, 송신자는 수신자에게 수신확인 메시지를 받은 경우에만 전송성공으로 인식한다.
- [991] 유통 메시징서버 시스템은 응답메시지 수신에 실패한 경우에도 91과 같은 처리 흐름도를 따른다. 즉, 메시지가 수신자에게 정상적으로 전달되었으나 송신자가 수신자로부터 수신확인 메시지를 받지 못한 경우에 송신자는 송신실패 오류로 인식하고 수신자에게 동일한 메시지를 동일한 MessageId로 재전송하게 되며, 수신자는 수신한 문서의 MessageId가 이전 수신 메시지와 동일한 경우에는 중복 수신으로 수신확인 메시지를 보낸 후 내부 처리를 한다.
- [992] 유통 메시징서버 시스템은 오류메시지 수신에 실패한 경우에도 92와 같은 처리 흐름도를 따른다. 즉, 송신자가 수신자에게 전송한 메시지가 정확히 전달되었으나 전송메시지 자체에 오류가 있어서 오류메시지를 응답받은 경우에 송신자는 오류 유형에 따라 메시지 처리를 달리하며, 재요청 시 전송하는 메시지의 MessageId는 동일할 필요는 없으며 업무 상황에 따라 다르게 처리할 수 있다.
- [993]
- [994] 상술한 바와 같은 본 발명에 따른 유통 메시징서버 시스템에 있어서 필수로 요구되는 기능인 "① 메시지 송수신", "② 수신 메시지 사서함 관리", "③ 메시지 보안", "④ 송수신 이력 관리", "⑤ 주소 디렉터리 서버 연계", "⑥ 메시지 검증", "⑦ 내부시스템 연계 인터페이스", "⑧ 유통증명서 발급 및 관리" 기능들에 대하여 상세히 설명하면 다음과 같다.
- [995]
- [996] **① 메시지 송수신**
- [997] 유통 메시징서버 시스템이 메시지를 송수신하는 기본 프로세스는 상술한 본 발명에 따른 [전자문서 유통 방법]의 "문서 송수신 방안"을 따른다. 메시지 송수신을 위한 기본이 되는 메시지 교환 유형은 메시지 유통 프로토콜의 동기식 응답을 기본으로 하며, 송신메시지와 수신확인 메시지, 송신 메시지와 수신오류 메시지, 송신 메시지와 비즈니스적 응답메시지(수신확인 메시지의 의미를 포함)의 구성을 이룰 수 있다.
- [998] 메시지 송수신 유형으로는 송신과 수신확인 응답 메시지의 조합과, 송신과 비즈니스 응답메시지의 조합을 포함하는 2가지 유형이 있다.
- [999] 메시지 송수신 유형이 송신과 수신확인 응답 메시지의 조합인 경우에 처리 흐름은 도 93과 같으며, 송신 메시지와 수신확인(또는 수신오류) 메시지는

SOAP(Simple Object Access Protocol) Request-Response의 조합으로 이루어지며, 송신 메시지와 그에 대한 응답 메시지는 송신 메시지의 MessageId를 응답 메시지의 RefToMessageId에 넣어서 보냄으로써 연관관계를 맺어주는데, 이와 관련한 상세한 설명은 후술하는 [유통 프로토콜]을 참조한다.

[1000] 메시지 송수신 유형이 송신과 비즈니스 응답메시지의 조합인 경우에 처리 흐름은 도 94와 같으며, 송신 메시지와 수신확인(또는 수신오류) 메시지를 포함한 응답메시지는 SOAP Request-Response의 조합으로 이루어지며, 수신자가 메시지를 수신한 후 내부시스템에 실시간 연계하여 비즈니스 처리한 응답문서를 생성한 후에 응답문서와 수신확인 ACK 메시지를 함께 응답메시지에 실어서 송신자에게 전달하며, 송신메시지와 그에 대한 응답메시지는 송신메시지의 MessageId를 응답메시지의 RefToMessageId에 넣어서 보냄으로써 연관관계를 맺어주는데, 이와 관련한 상세한 설명은 후술하는 [유통 프로토콜]을 참조한다.

[1001] 송수신되는 메시지의 구조는 도 95에 도시한 바와 같이 MultiPart-MIME 구조를 가지며, 첫 번째 MIME 부분에는 SOAP 메시지가, 2번째 MIME부터는 전송하고자 하는 문서가 들어간다.

[1002] 첫 번째 MIME은 SOAP 헤더와 SOAP Body로 구성된 SOAP Envelope이 들어가며, SOAP 헤더는 메시지 송수신을 위한 메시지 헤더 정보, 전자서명, 수신확인 메시지, 동기식 전송 표시, 오류 메시지 등이 들어간다. 그리고, 두 번째 MIME은 메시지 수신자에게 전달하는 문서(정보)가 들어가며, 담당자 수신확인 메시지를 전달하는 경우에 이 위치에 들어간다. 그리고, 세 번째 MIME은 메시지 수신자에게 전달하는 문서(정보)가 2개 이상인 경우 세 번째 MIME부터 순차적으로 들어간다.

[1003] ② 수신 메시지 사서함 관리

[1004] 유통 메시징서버 시스템은 메시지를 수신하면 수신 메시지를 계정 별로 사서함에 저장한다. 수신 메시지 사서함은 하나 이상의 사용자 계정 별로 구분되어 메시지를 저장 관리하며, 사용자의 요청(새 수신메시지 존재 여부, 수신메시지 열람, 수신메시지 다운로드, 수신메시지 삭제 등)에 따라 필요한 처리 후 결과를 돌려주는 인터페이스를 반드시 표준화된 방식으로 제공하여야 한다.

[1005] 유통 메시징서버 시스템이 관리하는 사용자 계정이 전자문서유통에 포함되는 공인전자주소로서 자격을 갖기 위해서는 유통 메시징서버 시스템은 신뢰 사용자 계정을 갖기 위한 인증요건(차후 별도의 평가지침을 통해 요건을 정의할 것이며, 현 시점에서는 제3자 보관기관만이 이 인증요건을 충족한 것으로 인정함)을 통과하여야 한다.

[1006] 따라서, 개인 또는 기업(기관)이 전자문서유통에서 공인전자주소를 획득하기 위한 방안으로는 다음과 같은 2가지 방안이 있다. 첫 번째 방안은, 자체적으로 유통 메시징서버 시스템을 구축하고 이를 인증 받은 후 획득한 송수신 개체 ID를

주소 디렉터리 서버에 등록하는 것이며, 두 번째 방안은, 인증 받은 유통 메시징서버 시스템 중 신뢰 사용자 계정을 갖는 요건을 추가적으로 충족한 송수신 개체에 사서함을 개설하여 사용자 ID를 발급 받은 후 이를 주소 디렉터리 서버에 등록하는 것이다.

[1007] **③ 메시지 보안**

[1008] 전송 메시지에 대한 보안은 무결성 보장을 위한 전자서명과 기밀성 보장을 위한 암호/복호화로 나눌 수 있다. 유통 메시징서버 시스템을 통해 전송되는 메시지는 SOAP 메시지와 첨부문서로 나뉜다. 이때, 첨부문서는 유통 클라이언트 APP에서 이미 암호화되어 있는 상태이고 SOAP Envelope에는 단지 메시지 송수신을 위한 헤더 정보만 포함되므로 유통 메시징서버 시스템에서는 추가적인 암호화 과정을 거치지 않으며, 메시지 송수신 과정에서 위변조 방지를 위한 전자서명 과정은 수행한다. 전자서명 방식 및 세부 절차는 후술하는 [유통 프로토콜]을 참조한다.

[1009] **④ 송수신 이력 관리**

[1010] 유통 메시징서버 시스템은 향후 송수신에 관련한 분쟁이 발생하거나 이슈가 제기되었을 때, 이를 확인하기 위해 송수신에 대한 이력정보를 관리하여야 한다. 이력정보는 송수신 행위에 대한 정보뿐 아니라 실제 송수신한 문서에 대한 정보도 관리하여야 하는데, 실 문서를 제3자 보관기관에 보관한 경우에는 문서의 원본이 아닌 제3자 보관기관으로부터 받은 등록증명서만을 보관하는 것도 가능하다.

[1011] **⑤ 주소 디렉터리 서버 연계**

[1012] 유통 메시징서버 시스템은 주소 디렉터리 서버가 제공하는 서비스 연계 인터페이스를 사용하여 주소 디렉터리 서버와 연계를 한다. 주소 디렉터리 서버는 2종류의 주소 검색 서비스와 주소 등록 서비스, 주소 변경 서비스를 제공하는데 유통 메시징서버 시스템은 주소 검색 서비스 중 “공인전자주소에 대한 물리적 주소 검색” 서비스 연계 기능은 필수적으로 제공한다.

[1013] 검색 서비스 외에 주소 등록 및 주소 변경 서비스는 유통 메시징서버 시스템이 하위에 등록/관리하는 사용자 계정을 기업이나 개인의 공인전자주소로 사용할 수 있는지에 따라 사용 여부가 결정된다. 유통 메시징서버 시스템이 등록/관리하는 사용자 계정이 공인전자주소로 등록이 가능하도록 인증을 받은 경우에는 공인전자주소에 대한 등록 및 변경서비스를 대행하여야 하므로 주소 디렉터리 서버의 해당 서비스를 연계한다.

[1014] **⑥ 메시지 검증**

[1015] - 유통 메시징서버 시스템이 메시지를 송수신할 때 수신자는 메시지 수신시점에 메시지 유효성에 대한 검증을 수행하며, 도 96에 도시한 프로세스와 같이 수신자는 메시지 유효성 검증을 한 후 검증에 통과한 경우에만 메시지 수신확인 메시지를 송신자에게 전달하고, 그렇지 않은 경우에는 수신 메시지에 대한 오류 메시지를 전송한다.

[1016] - 검증대상: 수신 메시지의 스키마 검증(수신 메시지가 유통 프로토콜에 따라 정확히 패키징되었는지를 검증), 메시지의 무결성 검증(수신한 메시지의 전자서명 값을 검증함으로써, 메시지의 위변조가 발생하지 않고 무결한지를 검증), 메시지 송신자 검증(메시지에 전자서명을 한 송신자가 메시지에 표기된 송신자가 맞는지를 인증하기 위해 전자서명에 사용된 인증서와 메시지의 송신자가 동일한지를 검증)

[1017] ② 내부시스템 연계 인터페이스

[1018] 유통 메시지정서버 시스템은 내부 시스템이 유통 메시지정서버 시스템을 통해 문서를 전송하고 수신할 수 있도록 송수신을 위한 표준화된 인터페이스를 제공하여야 한다. 이 인터페이스에 대한 상세한 내용은 후술하는 [유통 클라이언트 APP]를 참조한다.

[1019] ③ 유통증명서 발급 및 관리

[1020] 유통증명서의 기본요건은, ① 유통증명서는 발신 및 수신 유통 메시지정서버 시스템이 생성한다는 것과, ② 유통증명서는 GPKI 및 NPKI 인증서를 기반으로 전자서명하여 생성된다는 것과, ③ 유통증명서는 전자문서 유통행위를 기준으로 생성(이때, 한 번의 전자문서 유통 시 하나 이상의 전자문서가 전달되는 경우 하나의 유통증명서를 생성하고, 하나의 전자문서 유통을 위해서는 반드시 해당 유통을 식별할 수 있는 ID가 부여되며 이를 기준으로 유통으로 유통증명서를 생성함)된다는 것이다.

[1021] 유통증명서 발급 시점에 고려하여야 할 내용은, ① 유통증명서의 일련번호는 개별 송수신 개체가 생성하므로 유일성 부여를 위하여 기존 증명서의 규격과는 달리 20byte 난수를 사용하는 것과, ② 유통증명의 갱신 및 폐지는 정의하지 않는다는 것과, ③ 유통증명서를 생성하는 유통 메시지정서버 시스템 및 유통 클라이언트 APP의 시스템 시각은 항상 현재 시각을 유지해야한다는 것과, ④ 유통증명서 정책은 기술규격에서 정의한 OID 및 명칭만을 사용한다.

[1022] 유통증명서 발급 프로세스는 도 97에 도시한 바와 같으며, 유통증명서의 유형 및 생성에 필요한 필수 정보는 아래 표 141과 같으며, 유통증명서의 필수 정보의 획득 방법은 아래 표 142과 같다.

[1023] 표 141

유형	생성주체/시점	목적	필수정보
수신 증명서	수신 유통 메시징서 버 시스템 /수신 직후	수신자의 메시지 수신사 실에 대한 부인방지	문서정보, 송신자, 수신자, 송신자 송신시각, 수신자 수신시각
열람 증명서	수신 유통 메시징서 버 시스템 /담당자 열람 직후	수신자의 수신메시지 열 람사실에 대한 부인방지	문서정보, 송신자, 수신자, 송신자 송신시각, 수신자 수신시각, 수신자 열람시 각
송신 증명서	전자문서 유통허브 /송신의뢰메시지 수 신 직후	송신자의 송신시도에 대 한 증명	문서정보, 송신자, 수신자, 송신자 송신의뢰시각

[1024] 표 142

유형	필수정보	생성방법
수신 증명서	문서정보, 송신자, 수신 자, 송신자 송신시각	송신자가 전송한 유통 메시지 내 필드값 이용
	수신자 수신시각	수신 유통 메시징서버 시스템의 시각 이용
열람 증명서	문서정보, 송신자, 수신 자, 송신자 송신시각	송신자가 전송한 유통 메시지 내 필드값 이용
	수신자 수신시각, 수신자 열람시각	수신 유통 메시징서버 시스템의 시각 이용
송신 증명서	문서정보, 송신자, 수신 자	송신자가 전송한 유통 메시지 내 필드값 이용
	송신자 송신의뢰시각	전자문서 유통허브 시스템의 시각 이용

[1025] 유통증명서는 송수신 개체가 생성하며 송수신 개체의 NP키 및 GP키 인증서를 이용하여 전자서명하며, 유통증명서의 기본 구조는 CMS 표준의 SignedData 구조를 사용하며 증명서와 동일한 콘텐츠 식별자를 사용한다.

[1026] 유통증명서의 contentType 은 다음과 같다.

[1027] id-kiec-arcCertReseponse OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiek(200032) certificate(2) 2 }

[1028] ARCCertResponse ::= CHOICE {

[1029] arcCertInfo [0] EXPLICIT ARCCertInfo,

[1030] arcErrorNotice [1] EXPLICIT ARCCertErrorNotice }

[1031] 유통증명서의 기본필드는 다음과 같다.

[1032] ARCCertInfo ::= SEQUENCE {

[1033] version [0] EXPLICIT ARCVersion DEFAULT v1,

[1034] serialNumber SerialNumber,

[1035] issuer GeneralNames,

- [1036] dateOfIssue GeneralizedTime,
- [1037] dateOfExpire DateOfExpiration,
- [1038] policy ARCCertificatePolicies,
- [1039] requestInfo RequestInfo,
- [1040] target TargetToCertify,
- [1041] extnsions [1] EXPLICIT Extensions OPTIONAL
- [1042] }
- [1043] 상기와 같은 유통증명서의 기본필드에 대하여 상세히 설명하면 다음과 같다.
- [1044] ① version, 버전
- [1045] - 유통증명서 구조의 버전을 나타낸다. 유통증명서를 위해서는 v2로 설정해야하며 target 필드에 dataHash를 사용한다.
- [1046] ARCVersion ::= INTEGER {v1(1), v2(2)}
- [1047] ② Serial Number, 일련번호
- [1048] - 유통증명서의 식별정보를 나타낸다. 유통증명서는 전자문서를 수신한 송수신 개체가 생성하므로 일련번호 방식의 식별번호는 의미가 없다. 또한, 송수신 개체의 유통클라이언트가 재설치되는 등의 경우에는 일련번호의 유지가 불가능하다.
- [1049] 따라서 유통증명서의 식별정보는 20byte 난수를 사용한다. 유통증명서를 처리하기 위해서는 20byte 난수를 처리할 수 있어야 한다.
- [1050] SerialNumber ::= INTEGER
- [1051] ③ issuer, 증명서 발급자
- [1052] - 유통증명서를 발급하는 발급자의 인증서 식별값을 넣는다. 본 필드의 값은 유통증명서를 전자서명한 서명자의 인증서 내 SubjectName 필드과 동일한 값을 가져야 한다.
- [1053] ④ dateOfIssue, 증명서 발급일
- [1054] - 발급자가 유통증명서를 발급한 시점을 나타낸다.
- [1055] ⑤ dateOfExpire, 증명서 효력 만기일
- [1056] - 유통증명서의 만료 시점을 나타낸다.
- [1057] ⑥ policy, 증명서 정책
- [1058] - 유통증명서 정책을 나타낸다. 모든 유통증명서 내의 정책 OID는 증명서 종류에 따라 다르며 기술규격에서 저장한 값만을 사용하여야 한다.
- [1059] - 유통증명서는 증명서 종류에 따라 일괄적으로 하나의 OID를 가진다.
- [1060] - Qualifier 값으로는 UserNotice > ExplicitText > DisplyText 에 UTF8String 형식으로 나타내며 지정된 문장을 사용한다.
- [1061] - 유통증명서 유형에 따라 아래의 표 143와 같은 정책 정보를 사용하여야 한다.
- [1062] 표 143

증명서 종류	정책 OID	Qualifier
발신 증명서	1.2.410.200032.2.?.1	발신 증명서
수신 증명서	1.2.410.200032.2.?.2	수신 증명서
수신확인 증명서	1.2.410.200032.2.?.3	수신확인 증명서

- [1063] ⑦ requestInfo, 증명서 요청 메시지 정보
- [1064] - 본 필드는 null로 설정한다.
- [1065] RequestInfo ::= CHOICE {
- [1066] arcCertRequest ARCCertRequest,
- [1067] null NULL }
- [1068] ⑧ target, 증명 대상
- [1069] - 유통된 전체 전자문서의 해쉬 값을 지정한다. 본 필드는 반드시 distributionInfos 방식을 사용하여야 한다. opRecord 및 orgAndIssued, dataHash 필드에 대한 구조는 제3자 보관기관의 '증명서 포맷 및 운용절차 기술규격'을 참조한다.
- [1070] - 유통되는 전자문서에 대한 정보는 DistributionInfos 필드에 포함된다.
- [1071] TargetToCertify ::= CHOICE {
- [1072] opRecord [0] EXPLICIT OperationRecord,
- [1073] orgAndIssued [1] EXPLICIT OriginalAndIssuedDocumentInfo,
- [1074] dataHash [2] EXPLICIT HashedDataInfo
- [1075] distributionInfos [10] EXPLICIT DistributionInfos}
- [1076] DistributionInfos ::= SEQUENCE OF DistributionInfo
- [1077] DistributionInfo ::= SEQUENCE {
- [1078] senderAdd GeneralNames,
- [1079] receiverAdd GeneralNames,
- [1080] dateOfSend GeneralizedTime,
- [1081] dateOfReceive [0] EXPLICIT GeneralizedTime OPTIONAL,
- [1082] dateOfReceiveConfirm [1] EXPLICIT GeneralizedTime OPTIONAL,
- [1083] distributionId INTEGER,
- [1084] numberOfFiles INTEGER,
- [1085] distributedFileInfos DistributedFileInfos}
- [1086] ①-① senderAdd, 공인 전자주소
- [1087] - 송신자의 공인 전자주소를 나타낸다.
- [1088] ①-② receiverAdd, 수신자 공인 전자주소
- [1089] - 수신자의 공인 전자주소를 나타낸다.
- [1090] ①-③ dateOfSend, 송신 일시
- [1091] - 송신자가 전자문서를 발송한 시점을 나타낸다.
- [1092] - 송신증명서의 경우 송신자가 전자문서 유통허브에 송신 의뢰한 시각을 지정한다.

- [1093] - 송신증명서는 본 필드만 포함하여야 하며 dateOfReceive 및 dateOfReceiveConfirm는 포함하지 않아야 한다.
- [1094] ①-④ dateOfReceive, 수신 일시
- [1095] - 수신자가 전자문서를 수신한 시점을 나타낸다. 해당 시점은 증명서를 생성한 시점보다 같거나 이전이어야 한다. 수신 증명서 및 열람 증명서는 반드시 본 필드를 포함하여야 한다. 송신증명서는 본 필드를 포함하지 않아야 한다.
- [1096] ①-⑥ dateOfReceiveConfirm, 열람 일시
- [1097] - 수신자가 전자문서를 수신하여 확인한 시점을 나타낸다. 해당 시점은 수신 일시보다 같거나 이후이어야 하며 증명서를 생성한 시점보다 같거나 이전이어야 한다. 열람 증명서는 반드시 본 필드를 포함하여야 한다. 송신 증명서 및 수신 증명서는 반드시 본 필드를 포함하지 않아야 한다.
- [1098] ①-⑥ distributionId, 유통식별값
- [1099] - 전자문서의 유통 건에 대한 식별 값을 나타낸다. 본 필드의 생성을 위하여 20byte 난수를 생성하여 사용한다. 본 필드값은 전자문서 유통에 대하여 유통메시지에 부여되는 식별값을 의미한다.
- [1100] ①-⑦ numberOfFiles, 유통파일 개수
- [1101] - 유통 시 하나 이상의 전자문서가 전달될 수 있으며, 본 필드는 한 번의 유통에 전달되는 파일의 개수를 나타낸다.
- [1102] ①-⑧ distributedFileInfos, 유통문서정보
- [1103] - 유통 시 하나 이상의 전자문서가 전달될 수 있으며, 본 필드에는 전달되는 모든 문서에 대한 정보가 포함되어야 한다.
- [1104] DistributedFileInfos ::= SEQUENCE OF DistributedFile
- [1105] DistributedFile ::= SEQUENCE {
- [1106] fileHashedData HashedDataInfo,
- [1107] fileId [0] UTF8String OPTIONAL,
- [1108] fileName [1] UTF8String OPTIONAL
- [1109] }
- [1110] ①-⑧-① fileHashedData, 파일 해쉬 정보
- [1111] - 본 필드는 유통되어 전달된 전자문서에 대한 해쉬 값을 나타낸다.
- [1112] ①-⑧-② fileId, 파일 식별값
- [1113] - 유통되는 전자문서에 식별 값을 부여한 경우 해당 문서에 대한 식별 값을 지정한다. 파일 식별 값은 송신자가 생성하며 전자문서를 수신자에게 전달할 때 함께 전달되어야 한다. 수신자는 전달받은 파일 식별 값을 이용하여 본 필드에 적용하여야 한다.
- [1114] - 송신자는 본 필드의 생성을 위하여 uuid 방식으로 생성하여야 한다.
- [1115] - 본 필드는 선택적으로 사용될 수 있으나 fileName 필드가 사용되지 않는 경우에는 반드시 사용되어야 하며 field 필드의 사용을 권고한다.
- [1116] ①-⑧-③ fileName, 파일명

- [1117] - 유통되는 전자문서에 대한 파일명을 나타낸다. 파일명은 송신자가 지정하며 전자문서를 수신자에게 전달할 때 함께 전달되어야 한다. 수신자는 전달받은 파일 식별 값을 이용하여 본 필드에 적용하여야 한다.
- [1118] - 본 필드는 선택적으로 사용될 수 있으나 fileID 필드가 사용되지 않는 경우에는 반드시 사용되어야 한다.
- [1119] 상술한 바와 같은 유통증명서의 시각 정보관련 정합성 기준은 아래 표144와 같다.
- [1120] 표 144

번호	필드	내용
1	dateOfSend	송신 일시/송신의뢰 일시
2	dateOfReceive	수신 일시
3	dateOfReceiveConfirm	열람 일시
4	dateOfIssue	증명서 발급일
5	dateOfExpire	증명서 효력 만기일

- [1121] 시각정보 순서는 발송일시 < 수신 일시 ≤ 열람 일시 ≤ 증명서 발급일 < 증명서 효력 만기일이며, 유통증명서 검증 시에 시각 정보가 상기의 순서에 맞는지 확인해야 한다.
- [1122] 유통증명서의 검증은 증명서 구조 검증, 증명서 전자서명 검증, 증명서 주요 필드 확인, 증명서 시각 정보 정합성 검증을 포함한다.
- [1123] 증명서 구조 검증은 증명서가 ASN.1 정의된 바와 동일한 지 검증하는 과정이다.
- [1124] 증명서 전자서명 검증은 유통증명서에 적용된 전자서명을 검증하는 과정이다.
- [1125] 증명서 주요 필드 확인은 version 필드 값이 v2인지 확인하는 버전필드 확인, target 필드가 hashData 인지 확인하는 target 필드 확인, 전자서명에 사용된 인증서의 DN과 증명서 기본필드의 DN이 동일한지 검증하는 발급자 정보 검증, requestInfo 필드가 Null 인지 확인하는 requestInfo 필드 확인, distributionInfos 확장필드가 존재하는지 확인하고 critical이 TRUE인지 확인하는 확장필드 확인, numberOfFiles 필드의 값과 distributionInfos 확장필드 내 DistributedFile의 개수가 동일한 지 확인하는 파일 개수 확인, target 필드의 해쉬 값과 distributionInfos 확장필드의 해시 값이 동일한 지 확인하는 target 필드 해시 값 확인, 유통증명서 시각 정보의 정합성 검증 기준에 따라 검증하는 시각 정보 정합성 검증을 포함한다.
- [1126] 증명서 시각 정보 정합성 검증은 유통증명서 시각 정보의 정합성 검증 기준에 따라 검증한다.
- [1127] 한편, 유통증명은 전자문서의 유통과정에서 발생한 발신, 수신, 수신확인에 대한 사실에 대하여 신뢰성 있는 방식으로 증명하는 행위를 말한다. 유통증명은 별도의 응용프로그램에서 수행하며 유통증명서 뷰어 및 유통증명API에서 수행하지 않는다. 유통증명은 유통증명서 검증에 추가적으로 수행하고자 하는

경우 다음의 내용을 수행한다.

[1128] - 유통증명서 검증: 유통증명서의 검증을 수행한다.

[1129] - 유통증명서 정책 확인: 발신, 수신, 수신확인에 대한 유통증명서 정책 OID 및 Qualifier 값을 확인한다.

[1130] - 송신자 주소 확인: 전자문서를 발송한 송수신 개체의 주소가 정확한지 확인한다.

[1131] - 수신자 주소 확인: 전자문서를 수신한 송수신 개체의 주소가 정확한지 확인한다.

[1132] - 발송 일시 확인: 송신자가 전자문서를 발송한 시각이 정확한지 확인한다.

[1133] - 수신 일시 확인: 수신자가 전자문서를 수신한 시각이 정확한지 확인한다.

[1134] - 수신확인 일시 확인: 수신자가 전자문서를 수신확인한 시각이 정확한지 확인한다.

[1135] - 유통 ID 확인: 유통 개별 건에 부여된 유통 ID가 정확한지 확인한다. 송신자 및 수신자가 유통 ID를 별도로 보관하여 관리하는 경우 이를 비교하여 관리할 수 있다.

[1136] - 유통 파일의 식별자 또는 파일명 확인: 유통되는 파일의 ID 또는 파일명이 정확한지 확인한다. 송신자 및 수신자가 파일 ID 및 파일명을 별도로 보관하여 관리하는 경우 이를 비교하여 관리할 수 있다.

[1137] - 유통 파일의 해쉬 값 확인: 유통 대상이 되는 파일들의 각각의 해쉬 값과 확장필드의 Distributedfile 필드의 값이 동일한지 확인한다. 이때 사용하는 해쉬 알고리즘은 유통증명서 내에 지정된 것으로 수행하여 비교한다.

[1138] 한편, 유통증명서 프로파일은 아래 표 145과 같으며, 고려할 사항은 전자서명은 RSA 2048bit 및 SHA256 알고리즘을 적용한다는 것과, signedData 구조에서 인증서는 반드시 포함되어야 한다는 것과, signerInfos 필드에는 하나의 signerInfo만 포함된다는 것이다.

[1139] 표 145

기본필드	내용	특이사항
version	버전	v3
serialNumber	일련번호	20byte 난수
dateOfIssue	발급 일시	GeneralizedTime
dateOfExpire	증명서 효력 만기일시	GeneralizedTime
policy	증명서 정책	OID : 1.2.410.200032.2.7.7
requestInfo	증명서 요청 메시지 정보	null
target	증명대상	distributionInfos 구조 사용
senderAdd	송신자 공인전자주소	
receiverAdd	수신자 공인전자주소	
dateOfSend	송신일시	GeneralizedTime, 필수
dateOfReceive	수신일시	GeneralizedTime, 선택
dateOfReceiveConfirm	수신확인일시	GeneralizedTime, 선택
distributionId	유통식별자	20byte 난수
numberOfFiles	전송파일 개수	
distributedFileInfos	전송파일 정보	하나 이상의 DistributedFile
DistributedFile		
fileHashedData	파일 해쉬값	SHA256
fileId	파일 ID	fileId와 fileName 두 개 필수
fileName	파일명	중 하나는 필수

- [1140] 상술한 바와 같은 유통증명서를 제3자 보관기관과 연계하는 방안은 유통증명서를 발급함과 동시에 제3자 보관기관에 저장(보관) 의뢰함으로써 발급된 유통증명서의 신뢰성을 보장받는 것이다.
- [1141] 제3자 보관기관 사업자의 유통 메시징서버 시스템의 경우에 유통증명서 보관 프로세스는 도 98과 같으며, 유통 메시징서버 시스템에서 발급한 유통증명서를 직접 제3자 보관기관 연계모듈을 통해 제3자 보관기관 내부에 보관요청을 하며, 제3자 보관기관 연계모듈은 유통증명서 보관요청모듈과 제3자 보관기관 연계인터페이스 클라이언트 모듈로 구성되며, 기존의 제3자 보관기관 연계인터페이스 규격에 따라 제3자 보관기관에 보관된다.
- [1142] 일반 송수신 개체의 유통 메시징서버 시스템의 경우에 유통증명서 보관 프로세스는 도 99와 같으며, 발급된 유통증명서 보관을 위해 제3자 보관기관 사업자들에게 요청을 하기 위해 제3자 보관기관 사업자의 유통 메시징서버 시스템에 요청메시지를 전달하며, 외부에서 유통증명서 보관요청을 받은 제3자 보관기관 사업자 유통 메시징서버 시스템은 제3자 보관기관 연계모듈을 통해 제3자 보관기관 내부에 보관 요청을 하며 제3자 보관기관 연계모듈은 기존의 제3자 보관기관 연계 인터페이스 규격에 따라 제3자 보관기관에 보관 요청을 한다.
- [1143] 송수신 개체가 유통증명서를 제3자 보관기관에 보관하는 상세 처리는 도 100에 도시한 바와 같은 절차로 이루어지며, 상세한 설명은 다음과 같다.
- [1144] ○ 유통증명서 등록자
- [1145] - 제3자 보관기관에 유통증명서를 보관할 때 제3자 보관기관 사업자와 송수신 개체 간 계약에 의해 보관대행자를 지정할 수 있으며, 제3자 보관기관 사업자는

보관대행자가 등록자가 되어 보관대행자의 공인인증서를 기반으로
유통증명서를 보관하게 됨

[1146] ○ 제3자 보관기관 보관요청 프로세스 유형

[1147] - 제3자 보관기관 사업자는 동기식 또는 비동기식 처리 중 하나이상
제공하여야 하며 유통 메시징서버는 연계하고자 하는 제3자 보관기관 사업자가
제공하는 방식에 따라 연계됨

[1148] - Case 1: 동기식 처리 프로세스 (송신자가 유통증명서 보관요청 시 제3자
보관기관에 등록이 완료된 후 등록증명서가 발급되는 모든 프로세스가
동기식으로 이루어짐으로써 송신자의 유통 메시징서버는 동기식 응답 메시지로
등록증명서를 전달 받음. 요청에 대한 응답 메시지가 최종적인 제3자 보관기관
등록결과이므로 보관요청에 대한 오류 발생 시 재처리하는 송신자의 유통
메시징서버가 수행하여야 함)

[1149] - Case 2: 비동기식 처리 프로세스 (송신자가 제3자 보관기관 유통
메시징서버에 유통증명서 보관요청을 하면, 제3자 보관기관 유통 메시징서버가
먼저 요청메시지의 유효성을 검증한 후, 보관요청을 접수함. 제3자 보관기관
사업자는 보관요청 메시지에 따라 제3자 보관기관에 등록하고 발급 받은
등록증명서를 최초 보관요청자인 송신자의 유통 메시징서버에 전달하여야 함.
접수한 보관요청에 대해서 제3자 보관기관에 등록하는 것은 제3자 보관기관
사업자의 책임이므로, 보관오류 발생 시 재처리 역시 제3자 보관기관 사업자가
수행하여야 함)

[1150]

[1151] [유통 프로토콜]

[1152] 이하에서는, 상술한 바와 같은 본 발명의 바람직한 실시예에 따른 전자문서
유통 시스템 및 방법에 적용되는 유통 프로토콜에 대하여 상세히 설명하도록
한다.

[1153] 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템 및 방법에 적용되는
유통 프로토콜을 설명함에 있어서, "① 메시지 패키징", "② 메시지 봉투 구성", "③
HTTP 바인딩"에 대하여 차례로 설명하도록 한다.

[1154] ① 메시지 패키징

[1155] 유통 프로토콜의 메시지 구조는 ebMS V2.0 규격을 준용하며, 두 개의 논리적인
MIME 파트를 가진다.

[1156] 첫 번째 MIME 파트는 SOAP 메시지를 포함하며 헤더 컨테이너라고 불리고,
SOAP 메시지는 Header와 Body로 구성되며, 두 번째 MIME 파트는 0개 이상의
추가적인 MIME 파트로서 페이로드 컨테이너라고 불리는데 어플리케이션
레벨의 첨부 문서를 포함한다.

[1157] 이와 같은 유통 메시지의 기본적인 구조는 도 101과 같으며, Simple Object
Access Protocol(SOAP) 1.1 및, SOAP Messages with Attachment와 같은 표준
규격을 준수한다.

- [1158] 유통 메시지 패키지의 모든 MIME Header 요소들은 SOAP Messages with Attachments 규격을 준수한다. 추가로, 메시지 패키지 내의 Content-Type MIME Header는 반드시 SOAP 메시지 문서를 포함하는 MIME Body 부분의 MIME 미디어 유형과 동일한 type 속성을 가진다. SOAP 규격에 따르면 SOAP 메시지의 MIME 미디어 유형은 "text/xml" 값을 가져야 한다고 되어 있다.
- [1159] 루트 부분은 [RFC2045]에 준하는 구조를 가지는 Content-ID MIME 헤더를 포함하고 Multipart/Related 미디어 유형에 대한 필수적인 파라미터에 추가하여, start 파라미터([RFC2387]에서는 선택사항)가 항상 존재해야 한다. multipart/related 메시지 패키지의 MIME 헤더 예제는 다음 표 146과 같다.
- [1160] 표 146

```
Content-Type: multipart/related; type="text/xml"; boundary="boundaryValue";
start=messagepackage-123@example.com

--boundaryValue
Content-ID: <messagepackage-123@example.com>
```

- [1161] 이하에서 본 발명에 따른 유통 메시지에 대하여 설명함에 있어서 메시지 패키지의 루트 Body 부분을 Header(헤더) 컨테이너라고 정의한다. Header 컨테이너는 MIME 몸체 부분으로 SOAP Messages with Attachment명세에서 정의한 것과 같이 하나의 SOAP 메시지를 포함한다.
- [1162] 헤더 컨테이너의 MIME Content-Type header는 SOAP 규격에 따라 "text/xml" 값을 가져야 한다. Content-Type 헤더는 "charset" 속성을 포함할 수도 있으며, 예제는 다음 표 147과 같다.
- [1163] 표 147
-
- ```
Content-Type: text/xml; charset="UTF-8"
```
- 
- [1164] MIME charset 속성은 SOAP 메시지를 생성하는데 사용되는 문자 군을 식별하기 위해 사용된다. 이 속성의 의미론은 [XMLMedia]에 명시되어 있는 text/xml의 "charset parameter / encoding consideration"에 설명되어 있다. 유효한 값의 목록은 <http://www.iana.org>에서 찾을 수 있다.
- [1165] 만약에 두 가지가 모두 포함되어 있다면, MIME charset 속성은 SOAP 메시지의 인코딩 선언부와 동일해야 한다. 만약에 제공되어 있다면 MIME charset 속성은 SOAP 메시지를 생성할 때 인코딩과 상충되는 값을 포함하고 있어서는 안 된다.
- [1166] 이 문서를 인코딩할 때는 최대한의 호환성을 위하여 반드시 [UTF-8]을 사용하여야 한다. text/xml[XMLMedia]에서 도출해 온 미디어 유형들을 위해 정의된 처리 규칙 때문에 이 MIME 속성은 기본 값을 가지지 않는다.
- [1167] 헤더 컨테이너의 예제는 아래 표 148와 같다.
- [1168] 표 148

|                                                        |      |              |
|--------------------------------------------------------|------|--------------|
| Content-ID: <messagepackage-123@example.com>           | ---- | Header       |
| Content-Type: text/xml; charset="UTF-8"                |      |              |
|                                                        |      |              |
| <SOAP:Envelope                                         | --   | SOAP Message |
| xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/" |      |              |
| <SOAP:Header>                                          |      |              |
| ...                                                    |      |              |
| </SOAP:Header>                                         |      |              |
| <SOAP:Body>                                            |      |              |
| ...                                                    |      |              |
| </SOAP:Body>                                           |      |              |
| </SOAP:Envelope>                                       | --   |              |
|                                                        |      |              |
| --boundaryValue                                        | ---- |              |

- [1169] SOAP Messages with Attachments 규격에 따라 메시지 패키지 내에는 0개 이상의 페이로드 컨테이너가 포함될 수 있다. 만약 메시지 패키지가 어플리케이션 페이로드를 포함하고 있다면, 이는 반드시 페이로드 컨테이너에 포함되어 있어야 한다.
- [1170] 만약, 메시지 패키지가 어플리케이션 페이로드를 포함하고 있지 않다면 페이로드 컨테이너를 표시하지 말아야 한다. 각 페이로드 컨테이너의 내용물은 SOAP Body 내의 ebXML 메시지 Manifest 요소에 의해 식별되어야만 한다.
- [1171] ebXML 메시지 서비스 명세는 어플리케이션 페이로드의 구조와 내용물에 대하여 어떠한 규정도 어떤 식의 제약도 정해놓고 있지 않다. 페이로드는 simple-plain-text 객체 또는 복잡하게 중첩된 여러 부분의 객체도 될 수 있다. 페이로드 객체의 구조와 구성에 대한 명세는 ebXML 메시지 서비스를 사용하는 업무 프로세스나 정보 교환을 어떻게 정의하는지에 따라 달라질 수 있다. 페이로드 컨테이너의 예제는 다음 표 149와 같다.

[1172] 표 149

|                                      |       |            |           |
|--------------------------------------|-------|------------|-----------|
| Content-ID: <domainname.example.com> | ----- | ebXML MIME |           |
| Content-Type: application/xml        | ----- |            |           |
|                                      |       |            | Payload   |
| <Invoice>                            | ----- |            | Container |
| <Invoicedata>                        |       | Payload    |           |
| ...                                  |       |            |           |
| </Invoicedata>                       |       |            |           |
| </Invoice>                           | ----- |            |           |

- [1173] 본 발명에 따른 유통 메시지의 모든 MIME 부분은 [RFC2045] 규격에 준하는 추가적인 MIME 헤더들을 포함할 수 있다. 구현 시에는 이 발명에서 정의되지 않은 MIME 헤더들을 무시할 수도 있으며, 식별할 수 없는 MIME 헤더들은

반드시 무시해야 한다. 예를 들어, 구현 시에 content-length를 메시지에 포함할 수 있지만, content-length가 나타나 있는 메시지의 수령자는 이를 무시할 수도 있다.

[1174] **② 메시지 봉투 구성**

[1175] SOAP 규격에 준하여 모든 확장 요소 내용은 유효한 네임스페이스로 한정되어야 한다. 본 발명에서 정의된 모든 ebXML SOAP 확장 요소 내용은 ebXML SOAP Envelope 확장 네임스페이스로 한정되어야 한다. 네임스페이스 선언부들은 SOAP Envelope, Header 또는 Body 요소들에 포함되어 있거나, 각 SOAP 확장 요소들에 직접 포함되어 있을 수 있다.

[1176] SOAP Envelope은 SOAP 메시지의 Root 항목으로 SOAP 메시지 내의 각종 Namespace 들을 선언한다. 선언해야 할 Namespace는 다음 표 150과 같다.

[1177] 표 150

| 항목                | Namespace URL                             |
|-------------------|-------------------------------------------|
| SOAP              | http://schemas.xmlsoap.org/soap/envelope/ |
| Digital Signature | http://www.w3.org/2000/09/xmldsig#        |
| xlink             | http://www.w3.org/1999/xlink              |
| xs1               | http://www.w3.org/2001/XMLSchema-instance |

[1178] 메시지 봉투의 스키마 구조는 도 102와 같으며, 메시지 봉투의 예제는 아래 표 151와 같다.

[1179] 표 151

```

<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
 http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
 <SOAP:Header
 xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
 xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
 http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
 <eb:MessageHeader ...>
 ""
 </eb:MessageHeader>
 </SOAP:Header>
 <SOAP:Body
 xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
 xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
 http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
 <eb:Manifest eb:version="2.0">
 ""
 </eb:Manifest>
 </SOAP:Body>
</SOAP:Envelope>

```

[1180] SOAP Envelope 요소의 자식 요소인 SOAP Header 요소와 SOAP Body 요소에 대하여 차례로 상세히 설명하면 다음과 같다.

[1181] SOAP Header 요소는 SOAP Envelope 요소의 첫 번째 자식 요소이며, MessageHeader, SyncReply, Signature, ErrorList와 같은 확장 요소를 포함한다.

[1182] MessageHeader는 메시지의 라우팅 정보(To/From, 등)와 메시지에 관한 다른 문맥정보를 포함한 필수 요소이며, SyncReply는 다음 SOAP 노드로 가는 필수 전송 상태를 가리키는 요소이고, Signature는 메시지와 연관된 데이터를 서명하는 [XMLDSIG]에 준하는 전자서명을 표시하는 요소이며, ErrorList는 이전의 메시지를 대상으로 보고되어진 오류의 목록을 담은 요소이며 이전의 메시지에 대한 오류를 보고할 때에만 사용되어지는데, 이와 같은 MessageHeader의 요소들 각각에 대하여 상세히 설명하면 다음과 같다.

[1183] MessageHeader 요소는 모든 ebXML 메시지에 표현되어야 하는 필수 요소로서 반드시 SOAP Header 요소의 자식 요소로 표현되어야 한다. MessageHeader 요소는 다음과 같은 하위 요소들로 구성된 복합 요소이며, MessageHeader의 element 구조는 아래 표 152과 같으며, MessageHeader의 스키마 구조는 도 103과 같다.

[1184] 표 152

| 항목 명               | 설명                                                                     | 반복<br>회수 | 유<br>형 | 길이     |
|--------------------|------------------------------------------------------------------------|----------|--------|--------|
| From               | • 메시지 송신 송수신 개체 정보                                                     | 1..1     |        |        |
|                    | PartyId    ▪ 송신자를 식별하는 코드                                              | 1..1     | S      | 13     |
|                    | Role        ▪ 송신자 역할                                                   | 1..1     | S      | 최대 256 |
| To                 | • 메시지 수신 송수신 개체 정보                                                     | 1..1     |        |        |
|                    | PartyId    ▪ 수신자를 식별하는 코드                                              | 1..1     | S      | 13     |
|                    | Role        ▪ 수신자 역할                                                   | 1..1     | S      | 최대 256 |
| CPAID              | • 거래협업정의서 아이디                                                          | 1..1     | S      | 최대 256 |
| Conversatio<br>nId | • 송수신 트랜잭션 구분자                                                         | 1..1     | S      | 최대 256 |
| Service            | • CPA에 정의된 메시지 서비스                                                     | 1..1     | S      | 최대 256 |
| Action             | • Service 내의 특정 업무 프로세스 구분자<br>• Service 내에서 유일한 값                     | 1..1     | S      | 최대 256 |
| MessageData        | • 메시지를 식별하기 위한 데이터                                                     | 1..1     |        |        |
|                    | MessageId    ▪ 하나의 메시지가 가지는 유일한 식별자                                    | 1..1     | S      | 최대 256 |
|                    | Timestamp    ▪ 메시지 생성 시간<br>▪ UTC 형식<br>▪ ex> 2008-07-31T06:29:39.724Z | 1..1     | S      | 24     |
|                    | RefToMessageId    ▪ 응답 메시지에만 존재<br>▪ 요청 메시지의 MessageId                 | 0..1     | S      | 최대 256 |

[1185] SyncReply 요소는 동기식 송신을 의미하는 것으로서 id 속성, version 속성, SOAP actor 속성(반드시 "http://schemas.xmlsoap.org/soap/actor/next" 값을 가져야 함), SOAP mustUnderstand 속성값을 가지며, SyncReply 요소의 예제는 다음 표 153와 같다.

[1186] 표 153

```
<eb:SyncReply eb:id="3833kkj9" eb:version="2.0" SOAP:mustUnderstand="1"
 SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
```

[1187] Signature 요소는 SOAP Header의 자식요소로 반드시 존재해야 하는데, 이는 유통 메시지는 위에서 언급한 위험 요소에 대응하기 위하여 반드시 전자적으로 서명되어야 하기 때문이다.

[1188] [XMLDSIG] 규격에 따라 전자서명을 수행하는 과정은 다음과 같다.

- [1189] 먼저, SOAP Envelope에 SignatureMethod, CanonicalizationMethod, Reference 요소를 가진 SignedInfo 요소와 필수 페이로드 객체를 [XMLDSIG]에 규정된 대로 생성한다.
- [1190] 다음으로, 정규화 한 뒤 [XMLDSIG]에 지정된 대로 SignedInfo에 지정된 알고리즘을 기준으로 SignedInfo의 SignatureValue를 산출한다.
- [1191] 다음으로, [XMLDSIG]에 지정된 대로 SignedInfo, KeyInfo(권고사항), SignatureValue 요소를 포함하는 Signature 요소를 생성한다.
- [1192] 다음으로, SOAP Header의 Signature 요소를 SOAP Header 요소에 포함한다.
- [1193] 상술한 바와 같은 전자서명 시에 사용되는 알고리즘 정보는 다음과 같다. 알고리즘은 W3C "XML-Signature Syntax and Processing" (RFC3275)의 알고리즘 부분(6.0 Algorithms)을 기본적으로 따른다. 또한, 국내 고유 알고리즘을 지원하기 위해 TTAS.IF-RFC3075 "확장성 생성 언어 전자서명 구문과 처리 (XML-Signature Syntax and Processing)" (한국정보통신기술협회, 2004년)에서 정의된 알고리즘을 이용한다.
- [1194] 본 발명에 따른 유통 프로토콜에서 이용하는 알고리즘 목록은 전자서명 Namespace, 해쉬 (Digest), 전자서명(Signature), 정규화(Canonicalization), 변환(Transform)을 포함한다. 메시지 송수신 시 전자서명의 생성 및 검증 과정에서 모호성을 최소화하기 위해서 다음 목록 이외의 알고리즘은 사용하지 않는 것이 바람직하다.
- [1195] 전자서명의 Namespace의 예제는 다음 표 154와 같다.
- [1196] 표 154

---

```
<... xmlns:ds="http://www.w3.org/2000/09/xmldsig#" ... >
```

---

- [1197] 데이터를 축약하는데 이용하는 알고리즘으로 SHA1과 SHA256을 이용할 수 있으며, 예제는 다음 표 155와 같다. 단, HA1은 '공인인증서 암호체계 고도화'가 완전히 적용되는 시점인 2012년부터는 사용이 제한된다.
- [1198] 표 155

---

```
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
```

또는

```
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
```

---

- [1199] 메시지 전자서명 시 사용되는 알고리즘은 RSAwithSHA1, RSAwithSHA256로서, 예제는 다음 표 156과 같다. 단, RSAwithSHA1을 이용하는 경우는 '공인인증서 암호체계 고도화'가 완전히 적용되는 시점인 2012년부터는 사용이 제한된다.
- [1200] 표 156

---

```
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
또는
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig#more#rsa-s-sha256"/>
```

---

- [1201] 논리적으로 동일한 문서에 대해 물리적으로 여러 가지 표현이 가능 방식한 XML의 특성으로 인해 같은 문서에 대해 전자서명 값이 다르게 나올 수 있으므로, 이러한 현상을 방지하기 위해 반드시 정규화(Canonicalization) 과정을 거쳐야 하며, 예제는 다음 표 157과 같다. 정규화는 주석 없는 정규 XML(Canonical XML, omits comments)을 사용한다.

[1202] 표 157

---

```
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xmldsig-core1-20010315"/>
```

---

- [1203] 전체 XML 데이터 중에서 실제 서명 대상이 되는 데이터를 가공하고 선택하는 과정을 거치는 알고리즘으로 다양한 변환 알고리즘이 존재하나 그 중에서 3가지만을 이용하도록 한다. 첫 번째는 전자서명이 서명 대상 내에 포함되는 형식을 따르므로 Enveloped Signature 변환이고, 두 번째는 상기 설명한 정규화(Canonicalization), 그리고 세 번째는 서명 대상 정보를 선택하는 XPath 필터링(XPath Filtering)이며, 예제는 다음 표 158와 같다.

[1204] 표 158

---

```
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
및
<ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xmldsig-core1-20010315"/>
및
<ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
<ds:XPath>not(ancestor-or-self::node()@SOAP:actor="urn:oasis:names:tc:ebxml-rs-ga:actor:rest"|
| ancestor-or-self::node()@SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"))
</ds:XPath>
</ds:Transform>
```

---

- [1205] 전자서명 구문의 구조는 도 104와 같으며, 상술한 방식대로 전자서명이 수행된 메시지 예제는 다음 표 159와 같다.

[1206] 표 159

---

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
 <SOAP:Header>
 <eb:MessageHeader eb:id="..." eb:version="2.0" SOAP:mustUnderstand="1">
 ...
 </eb:MessageHeader>
 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
 <CanonicalizationMethod
 Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
 <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
 <Reference URI="">
```

[1207]

```

<Transforms>
 <Transform
 Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
 <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
 <XPath> not(ancestor-or-self::node()[@SOAP:actor=
 "urn:oasis:names:tc:ebxml-msg:actor:nextMSH"]
 | ancestor-or-self::node()[@SOAP:actor=
 "http://schemas.xmlsoap.org/soap/actor/next"])
 </XPath>
 </Transform>
 <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>...</DigestValue>
</Reference>
<Reference URI="cid://blahblahblah/">
 <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
 <DigestValue>...</DigestValue>

<Transforms>
 <Transform
 Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
 <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
 <XPath> not(ancestor-or-self::node()[@SOAP:actor=
 "urn:oasis:names:tc:ebxml-msg:actor:nextMSH"]
 | ancestor-or-self::node()[@SOAP:actor=
 "http://schemas.xmlsoap.org/soap/actor/next"])
 </XPath>
 </Transform>
 <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>...</DigestValue>
</Reference>
<Reference URI="cid://blahblahblah/">
 <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
 <DigestValue>...</DigestValue>

```

- [1208] Errorlist 요소는 메시지를 수신하여 처리 과정을 수행할 때 오류가 발생하는 경우에만 Header 하위에 위치한다. ErrorList 요소가 생성되는 경우에는 반드시 MessageHeader 요소 내에 RefToMessageId가 존재해야 하며 RefToMessageId는 오류가 발생한 메시지의 MessageId를 지칭해야 한다. ErrorList 요소는 id 속성, OAP mustUnderstand 속성, version 속성, highestSeverity 속성, 한 개 이상의 Error 요소와 같은 속성들을 가지며, ErrorList의 구조는 도 105와 같다. 이때, 보고될

오류가 없으면 `ErrorList` 요소는 존재해서는 안 된다.

- [1209] `highestSeverity` 속성은 모든 `Error` 요소들의 가장 심각한 상태를 표시한다. 특히, 어떤 `Error` 요소가 `severity`를 `Error` 라고 설정되어 있으면, `highestSeverity`는 `Error`로 설정되어야 하며, 그렇지 않은 경우에는 `highestSeverity`를 `Warning`으로 설정해야 한다.
- [1210] `Error` 요소는 `id` 속성, `codeContext` 속성, `errorCode` 속성, `severity` 속성, `location` 속성, `Description` 속성을 가진다.
- [1211] `id` 속성은 문서 내에서 `ErrorList` 요소를 유일하게 식별하는 역할을 한다.
- [1212] `codeContext` 속성은 `errorCodes`의 네임스페이스 또는 스키마를 나타낸다. 이는 반드시 URI이어야 한다. 이 속성의 기본값은 `urn:oasis:names:tc:ebxml-msg:service:errors`이다. 이 속성에 기본 값이 없으면, 그 명세의 구현은 `errorCodes`를 사용한다는 것을 가리킨다.
- [1213] 필수 속성인 `errorCode` 속성은 오류를 가지고 있는 메시지의 오류가 지닌 본질을 지시한다. `errorCode`의 유효한 값들과 코드의 의미는 아래에서 설명하도록 한다.
- [1214] 필수 속성인 `severity` 속성은 오류의 심각성을 나타내는 값으로서, 유효한 값들은 `Warning`, `Error`가 있는데, `Warning`은 오류가 존재하지만 대화 중의 다른 메시지들은 정상적으로 생성되고 있다는 것을 가리키며, `Error`는 복구 불가능한 오류가 메시지에 존재하고 대화 중 더이상 다른 메시지들은 생성되지 않는다는 것을 가리킨다.
- [1215] `location` 속성은 오류가 존재하는 메시지 부분을 가리킨다. 만약에 오류가 ebXML 요소 내에 존재하고 요소가 "well-formed"라면, `location` 속성의 내용은 `[Xpointer]`이어야 한다.
- [1216] `Description` 속성의 내용은 `xml:lang` 속성에서 정의된 언어로 오류의 서술적인 설명을 제공한다. 보통, 이것은 XML 파서나 메시지를 검증하는 소프트웨어가 생성한 메시지가 된다. 이 의미는 이 내용은 `Error` 요소를 생성했던 소프트웨어의 판매자나 개발자에 의해 정의된다는 것을 의미한다.
- [1217] `ErrorList`의 예제는 다음 표 160과 같다.
- [1218] 표 160

---

```

<eb:ErrorList eb:id="3490sdo", eb:highestSeverity="error" eb:version="2.0"
SOAP:mustUnderstand="1">
 <eb:Error eb:errorCode="SecurityFailure" eb:severity="Error" eb:location="URI_of_ds:Signature">
 <eb:Description xml:lang="en-US">Validation of signature failed</eb:Description>
 </eb:Error>
 <eb:Error ...> ... </eb:Error>
</eb:ErrorList>

```

---

- [1219] 유통 프로토콜을 기반으로 메시지를 송수신하는 과정에서 오류가 발생하면 오류를 인지한 송수신 개체는 상대방에게 오류 내용을 보고해야 하며, 보고해야

할 오류는 메시지 구조 오류, 신뢰메시징 오류, 보안 오류를 포함한다.

[1220] 본 발명에서 정의하는 유통 프로토콜보다 하위 레이어에 속하는 HTTP 및 Socket과 같은 데이터 통신 프로토콜과 관련된 오류들은 데이터 통신 프로토콜에서 지원하는 표준 메커니즘에 의해 발견되고 보고되어야 하며, 본 발명에서 정의하는 오류 보고 메커니즘을 사용하지 않는다.

[1221] 오류코드는 오류 대상 및 유형 별로 구분되며, 자세한 내용은 다음 표 161과 같습니다.

[1222] 표 161

| 오류 코드              | 내용                                 | 상세 설명                                                                                                                         |
|--------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| ValueNotRecognized | 요소 내용이나 속성 값이 인식되지 않는다.            | 비록 문서가 well formed이고 유효하지만 요소/속성의 값이 인식할 수 없는 값이고 그러므로 ebXML 메시지 서비스에 의해 사용되어질 수 없는 값을 포함한다.                                  |
| NotSupported       | 요소나 속성이 지원되지 않는다.                  | 비록 문서가 well formed이고 유효하고 요소나 속성이 이 명세의 규칙이나 제약을 따르고 있지만 메시지를 처리할 수 있는 ebXML 메시지 서비스에 의해 지원되지 않는다.                            |
| Inconsistent       | 요소 내용이나 속성 값이 또 다른 요소나 속성에 불일치 한다. | 비록 문서가 well formed이고 유효하고 이 명세의 규칙과 제약을 따르지만 요소와 속성의 내용이 다른 요소나 그들의 속성에 일치하지 않는다.                                             |
| OtherXml           | 요소 내용이나 속성 값 속에서의 또 다른 오류          | 비록 문서가 well formed이고 유효하지만 그 요소 내용이나 속성 값이 이 명세 내의 규칙과 제약에 따르지 않고 다른 오류 코드에 속하지 않는다.<br>Error 요소의 내용은 문제의 본질을 가리키는데 사용되어야 한다. |
| DeliveryFailure    | 메시지 전송 실패                          | 수신된 메시지가 대개 혹은 확실히 다음 목적지에 보내지지 못했다. 만약에 severity가 Warning으로 설정되어 있으면 메시지가 배달될 가능성은 적다.                                       |
| TimeToLiveExpired  | 메시지가 존재 할 수 있는 시간이 초과됨             | 메시지가 수신되었으나 MessageHeader 요소의 TimeToLive 요소가 제약한 시간을 초과한 시각에 수신되었다.                                                           |
| SecurityFailure    | 메시지의 보안 검사에 실패                     | 메시지를 보낸 당사자의 서명의 검증 또는 권한 또는 실명 검사에 실패했다.                                                                                     |
| Unknown            | 알 수 없는 오류                          | 어떤 오류 종류에도 속하지 않는 오류가 발생함을 뜻한다. Error 요소의 내용이 문제의 본질을 가리키는데 사용되어야 한다.                                                         |

- [1223] 한편, SOAP Body 요소는 SOAP Envelope 요소의 두 번째 자식 요소이며, Manifest와 같은 확장 요소를 포함하며, Manifest는 페이로드 컨테이너 또는 웹과 같이 다른 장소에 위치한 데이터를 가리키는 요소이다.
- [1224] Manifest 요소는 한 개 이상의 Reference 요소들로 구성된 복합 요소이다. 각 Reference 요소는 페이로드 컨테이너에 담긴 페이로드 문서(들)의 일부로 포함되거나 URL로 접근 가능한 원거리의 자원인 메시지에 관련된 데이터를 식별한다. SOAP Body에는 페이로드 데이터를 실지 않는 것이 바람직하며, Manifest의 목적은 XML 메시지와 관련된 특정한 페이로드를 쉽게 직접적으로 접근할 수 있게 하는 것과, 파싱 작업 없이도 어플리케이션이 페이로드를 처리할 수 있는지 판단할 수 있도록 하는 것이다.
- [1225] Manifest 요소는 다음과 같은 한 개의 id 속성, 한 개의 version 속성 및 한 개 이상의 Reference 요소로 구성되어 있다.
- [1226] Reference 요소는 0개 이상의 Schema 요소 및 0개 이상의 Description 요소를 포함하는 하위 요소들로 구성된 복합 요소이다. 이때, 0개 이상의 Schema 요소는 부모 Reference 요소에서 식별된 인스턴스 문서를 정의하는 스키마(들)에 대한 정보이며, 0개 이상의 Description 요소: 부모 참조 요소에 의해 Reference된 페이로드 객체에 대한 설명이다.
- [1227] Reference 요소는 그 자체가 [XLINK]의 단순 링크이다. XLINK 프로세서 또는 엔진의 사용이 필수적이지는 않지만 구현 요구 사항에 따라 유용할 수 있다. Reference 요소는 위에서 제공된 요소의 내용과 더불어 id, xlink-type, xlink:href, xlink:role와 같은 속성 내용을 포함하고 있으며, 이 외에 다른 유효 네임스페이스인 속성들이 존재할 수 있으며, 수신 MSH는 위에서 정의한 것 외에 외부의 네임스페이스 속성들은 무시할 수 있다. 이때, id는 Reference 요소에 대한 XML ID이며, xlink-type은 XLINK 단순 링크로 요소를 정의하며 "simple"이라는 고정된 값을 가지며, xlink:href는 참조된 페이로드 객체의 URI 값이며 [XLINK] 명세의 단순 링크에 준하는 것이어야 한다. 그리고, xlink:role는 페이로드 객체나 그의 목적을 설명하는 자원을 식별하는 것으로서, 존재한다면 [XLINK] 명세에 준하는 유효한 URI 값을 가져야 한다.
- [1228] Schema 요소는 참조하는 항목이 그것을 기술하는 스키마(들)를 가지고 있다면 (예, XML Schema, DTD, 또는 Database Schema), 그 Schema 요소는 Reference 요소의 자식 요소로 존재해야 한다. 이는 스키마와 버전을 식별하는 방법으로 사용되며, 부모 Reference 요소에 의해 식별되는 페이로드 객체를 정의한다. Schema 요소는 location 및 version과 같은 속성을 가지고 있다. 이때, location은 스키마의 필수 URI이며, version은 스키마의 버전 식별자이다.
- [1229] xlink:href 속성이 content id(URI scheme "cid")인 URI를 포함하고 있으면 그 content-id를 가지는 MIME은 메시지의 페이로드 컨테이너에 표현되어 있거나, 그렇지 않으면 errorCode를 MimeProblem으로, severity를 Error로 하는 오류를 발신당사자에게 전달해야 한다. xml:href 속성이 content id(URI scheme "cid")인

URI를 포함하고 있지 않으면 URI는 해석되지 못하고, 구현에 따라 오류를 전달해야 할지 말아야 할지 결정해야 한다. 오류가 전달되어야 한다고 결정되면 errorCode를 MimeProblem으로, severity를 Error로 하는 오류를 발신당사자에게 전달해야 한다.

[1230] 아래의 표 162은 전형적인 한 개의 페이로드 MIME Body 부분을 가지는 메시지의 Manifest를 나타낸다.

[1231] 표 162

---

```
<eb:Manifest eb:id="Manifest" eb:version="2.0">
 <eb:Reference eb:id="pay01" xlink:href="cdpay/cad-1" xlink:role="http://regrep.org/gcl/purchaseOrder">
 <eb:Schema eb:location="http://regrep.org/gcl/purchaseOrder/po.xsd" eb:version="2.0"/>
 <eb:Description xml:lang="en-US">Purchase Order for 100,000 widgets</eb:Description>
 </eb:Reference>
</eb:Manifest>
```

---

[1232] **③ HTTP 바인딩**

[1233] HTTP를 통해 메시지를 전송하는 방안에 있어서 HTTP 바인딩 예제는 다음 표 163와 같다.

[1234] 표 163

---

```
POST /servlet/ebXMLhandler HTTP/1.1
Host: www.example2.com
SOAPAction: "ebXML"
Content-type: multipart/related; boundary="Boundary"; type="text/xml";
start="<ebxhmheader111@example.com>"

--Boundary
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

[1235] xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
 xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
 http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
 http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
 http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<SOAP:Header>
 <eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
 <eb:From>
 <eb:PartyId>urn:duns:123456789</eb:PartyId>
 </eb:From>
 <eb:To>
 <eb:PartyId>urn:duns:912345678</eb:PartyId>
 </eb:To>
 <eb:CPAId>20001209-133003-28572</eb:CPAId>
 <eb:ConversationId>20001209-133003-28572</eb:ConversationId>
 <eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
 <eb:Action>NewOrder</eb:Action>
 <eb:MessageData>
 <eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
 <eb:Timestamp>2001-02-15T11:12:12</eb:Timestamp>
 </eb:MessageData>
 </eb:MessageHeader>
</SOAP:Header>
<SOAP:Body>
 <eb:Manifest eb:version="2.0">
 <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
 xlinkrole="XLinkRole" xlink:type="simple">
 <eb:Description xml:lang="en-US">Purchase Order 1</eb:Description>
 </eb:Reference>
 </eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>

```

[1236]

--Boundary

Content-ID: &lt;elxmlpayload111@example.com&gt;

Content-Type: text/xml

&lt;?xml version="1.0" encoding="UTF-8"?&gt;

&lt;purchase\_order&gt;

&lt;po\_number&gt;1&lt;/po\_number&gt;

&lt;part\_number&gt;123&lt;/part\_number&gt;

&lt;price currency="USD"&gt;500.00&lt;/price&gt;

&lt;/purchase\_order&gt;

--Boundary--

[1237] 본 발명에 따른 유통 프로토콜에서 HTTP 레벨의 응답 코드를 반환하기 위해서 [RFC2616]에 정의된 HTTP 응답 코드를 이용해야 하며, 주요 응답 코드는 다음표 164와 같습니다.

[1238] 표 164

| 상태 코드 | 연관 메시지                | 의미                                                    |
|-------|-----------------------|-------------------------------------------------------|
| 200   | OK                    | 요청이 성공적으로 처리됨                                         |
| 400   | Bad Request           | 요청에 문법적으로 잘못된 부분이 있음                                  |
| 401   | Unauthorized          | 클라이언트가 올바른 허가를 받지 않고 허가가 필요한 페이지에 접근하려 함              |
| 404   | Not Found             | 이 주소에서는 어떤 내용도 발견할 수 없음                               |
| 500   | Internal Server Error | 서버 내부의 오류로 인해 요청을 정상적으로 처리할 수 없음                      |
| 503   | Service Unavailable   | 처리할 수 있는 한계를 벗어나 과도하게 요청이 들어와서 서버가 현재 해당 요청을 처리할 수 없음 |

[1239]

[1240] [전자문서 서식 등록기]

[1241] 이하에서는, 상술한 바와 같은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템의 전자문서 서식 등록기와 관련하여 상세히 설명하도록 한다.

[1242] 전자문서 서식 등록기는 전자문서유통에서 송수신 개체가 문서를 유통하기 위해 필요한 서식을 생성, 등록, 관리할 수 있는 시스템이다.

[1243] 전자문서 서식 등록기는 서식생성기, 서식등록기, 서식관리기 및 표준연계모듈로 구성된다.

[1244] 서식생성기는 PDF 변환모듈과 PDF Form Designer로 이루어지며, PDF

변환모듈은 일반 서식을 PDF로 변환하는 기능(예: 표준 PDF-A로 생성)을 제공하며, PDF Form Designer는 입력가능한 Form PDF를 생성할 수 있는 기능을 제공하고 2차원바코드 및 복사방지마크등의 문서보안기능을 제공한다.

- [1245] 서식등록기는 사용자가 서식(예: 아래 한글, MS-Word등의 일반 서식)을 등록할 수 있는 기능을 제공한다.
- [1246] 서식관리기는 서식 관리자가 서식을 등록, 관리할 수 있는 기능을 제공하며, 카테고리별 등록, 버전별 이력관리기능 제공하고, 서식별 열람기간, 열람횟수, 인쇄횟수제어 등 설정기능 제공한다.
- [1247] 표준연계모듈은 유통 클라이언트 어플리케이션과 연계할 수 있는 기능을 제공하며, 서식 리스트, 검색기능 제공하고, 파일 다운로드 기능을 제공한다.
- [1248] 본 발명에 따른 전자문서 서식 등록기의 서식 등록 프로세스는 도 106과 같다.
- [1249] 표준전자문서는 Form designer를 이용하여 FormPDF로 생성하며, 필수 요건은 아래 표 165와 같다.
- [1250] 표 165

| 구 분       | 설 명                                                      |
|-----------|----------------------------------------------------------|
| 데이터 입력 필드 | 내용입력을 위한 데이터 입력 필드                                       |
| 2차원 바코드   | 인쇄된 문서의 원본 증명을 위하여 원본문서 및 원본문서의 전자 서명 데이터가 삽입됨.          |
| 복사방지마크    | 인쇄된 문서의 복사를 방지하기위한 기능이며 망실소거패턴을 이용하여 복사시 원본표시가 사라지는 형태임. |
| 안내문구      | 문서의 유효기간 및 검증방법등의 내용 수록.                                 |

- [1251] 표준전자문서의 구조는 문서 하단에 5cm 정도의 공간 확보가 필요하며 바코드 크기는 데이터량에 따라 가변적이며, 복사방지마크 크기는 3 > 1.3로 하며 위치는 서식 모양에 따라 적절한 위치로 배치하며, 일 예는 도 107과 같다.
- [1252] 표준연계모듈(표준인터페이스)은 유통 클라이언트 어플리케이션에서 사용자가 서식을 검색하고 다운로드하여 서식을 생성할 수 있도록 하며, Web UI(User Interface)로 제공하여 유통 클라이언트 어플리케이션에서 포함될 수 있도록 하고, 카테고리별 검색, 서식 목록, 파일 다운로드 등의 기능을 제공한다.
- [1253]
- [1254] [전자문서 패키징]
- [1255] 이하에서는, 상술한 바와 같은 본 발명의 바람직한 실시예에 따른 전자문서 유통 시스템 및 방법에 적용되는 전자문서 패키징과 관련하여 상세히 설명하도록 한다.
- [1256] 전자문서 패키징은 전자문서유통에서 송수신 개체가 문서를 유통하기 위해 필요한 메시징 시스템 규격이다.
- [1257] 전자문서 패키징은 표준전자문서, 첨부문서로 구성되어있으며 표준전자문서에 대한 메타데이터로 구성되어있다. 표준전자문서는 PDF-A를

기반으로 생성되며 메타데이터는 문서보안기능 등의 정보로 구성되어 있다.  
첨부분서는 PDF로 변환하지 않고 원본 그대로 패키징한다.

[1258] 표준전자문서는 사용자의 공인인증서로 전자서명하며 패키징 후 전자문서 패키지를 전자서명하여 패키지에 포함한다.

[1259] 도 108에는 전자문서 패키지 구조를 도시하였는데, 이와 같은 도 108을 참조하면 본 발명에 따른 전자문서 패키지 구조는 패키지 헤더, 메타 데이터, 표준전자문서, 첨부분서, 전자서명 데이터를 포함하며, 각각의 세부 구성 요소는 아래 표 27 내지 표 31과 같다.

[1260] 패키지 헤더는 패키지 전체 구조정보를 포함하며, 메타데이터는 표준전자문서의 문서보안기능의 정보를 포함하고 문서 열람횟수, 인쇄횟수, 2차원 바코드 정보 등의 정보를 포함하며, 표준전자문서는 표준 PDF-A 형식으로 구성되고 PDF 파일내 이미지 영역에 2D 바코드 데이터를 포함하며 표준 PDF Signed Data 영역에 전자서명데이터 포함하고, 첨부분서는 표준전자문서가 아닌 비정형문서이므로 문서보안기능 적용대상에서 제외하며 개별적 전자서명은 제외하며, 전자서명 데이터는 패키징된 데이터를 사용자의 공인인증서를 사용하여 전자서명하여 패키징에 포함한다.

[1261] 표 166

| 번호 | 구 성 요 소   | 비 고 |
|----|-----------|-----|
| 1  | 전체 파일 크기  |     |
| 2  | 메타데이터 크기  |     |
| 3  | 표준전자문서 크기 |     |
| 4  | 첨부파일 개수   |     |
| 5  | 첨부파일 크기   |     |

[1262] 표 167

| 번호 | 구 성 요 소       | 비 고 |
|----|---------------|-----|
| 1  | 열람횟수          |     |
| 2  | 인쇄횟수          |     |
| 3  | 저장기능          |     |
| 4  | 텍스트 추출기능      |     |
| 5  | 임시저장/불러오기 기능  |     |
| 6  | 문서 형태         |     |
| 7  | 2D 바코드 크기(가로) |     |

[1263] 표 168

|    |        |    |
|----|--------|----|
| 번호 | 구설요소   | 비고 |
| 1  | PDF 파일 |    |

[1264] 표 169

|    |       |    |
|----|-------|----|
| 번호 | 구설요소  | 비고 |
| 1  | 첨부 파일 |    |

[1265] 표 170

|    |          |    |
|----|----------|----|
| 번호 | 구설요소     | 비고 |
| 1  | 전자서명 데이터 |    |

[1266] 전자문서 폐키징을 검증하는 방안으로는 ① 전자문서 폐키징 전자서명 검증 방안, ② 표준전자문서 검증 방안, ③ 인쇄된 전자문서 검증 방안이 있으며, 각 방안에 대하여 설명하면 다음과 같다.

[1267]  
[1268] - 클라이언트 App는 전자서명 폐키징을 처리할 때 전자서명을 검증하며  
검증을 성공하였을 경우에만 표준전자문서를 전자문서부여로 전달한다.  
[1269] - 또한, 수동 전자서명 검증 기능을 지원하여 분쟁 발생 시 전자서명 검증을 할 수 있도록 한다.

[1270] ② 표준전자문서 검증 방안

- 전자문서부여는 표준전자문서를 열람할 때 전자서명검증을 수행하고  
성공하였을 경우에만 전자문서부여에서 파일을 열람할 수 있도록 한다.

[1272] ③ 인쇄된 전자문서 검증 방안

[1273] - 별도 제공되는 검증 프로그램과 평판 스캐너를 이용하여 2차원 바코드내  
전자서명 데이터 검증 및 원문문서 내용과 인쇄된 문서의 내용을 육안 비교한다.  
[1274] 한편, 복사방지마크는 전자문서를 최종으로 받는 사용자의 프린터 패턴에 따라  
생성해야하므로, 폐키징에는 포함되지 않는다.

[1275]

[1276] [유동 클라이언트 어플리케이션]

[1277] 이하에서는, 상술한 바와 같은 본 발명의 바람직한 실시예에 따른 전자문서  
유동 시스템의 유동 클라이언트 어플리케이션과 관련하여 상세히 설명하도록  
한다.

[1278] 기업 또는 개인들이 공인전자주소를 바탕으로 문서(정보)를 송수신하기  
위해서는 이를 지원하기 위한 사용자 인터페이스(User Interface, 이하 UI라 함)  
제공하는 어플리케이션이 필요하다. 유동 메시징서버 시스템이 메시지를  
주고받기 위한 메시징엔진으로서 이메일과 비교했을 때 메일서버와 같은  
역할을 한다면, 유동 클라이언트 어플리케이션(이하, APP라 함)은 사용자가 이  
메일서버와 연계하여 이메일을 주고받기 위해 제공되는 메일 클라이언트와

같은 사용자용 어플리케이션의 역할을 한다.

- [1279] 도 109에는 유통 클라이언트 APP의 구조도를 도시하였으며, 이와 같은 도 109를 참조하면 유통 클라이언트 APP는 유통 메시징서버 시스템을 이용해서 문서를 교환하고자 하는 일반 사용자를 위한 UI환경의 어플리케이션으로서 기본적으로 "① 사용자 인증", "② 메시지 작성", "③ 메시지 목록조회 및 상세내용 열람 기능", "④ 유통 메시징서버 시스템 연계"로 구성된다. 클라이언트 APP는 이와 같은 기본기능 외에 추가적으로 메시지 송수신 및 어플리케이션 관리를 위한 "⑤ 기본정보와 환경정보 관리", "⑥ 메시지 폴더 관리", "⑦ 문서 서식 관리", "⑧ 문서 작성기" 등의 기능을 제공할 수 있으나, 이는 어플리케이션 개발자에 의해 선택적으로 제공되는 기능이다.

[1280] ① 사용자 인증

- [1281] - 유통 클라이언트 APP이 유통 메시징서버 시스템과 연계되기 전에 유통 메시징서버 시스템은 사용자 계정을 확인한 후 로그인 세션 정보를 받아야 한다.
- [1282] - 유통 클라이언트 APP가 사용자 인증을 받기 위한 방법으로는 인증서(공인 또는 사설 모두 허용)를 기반으로 한 사용자 인증 또는 ID/PW를 기반으로 한 사용자 인증 등이 있다.

[1283] ② 메시지 작성 기능

- [1284] - 유통 클라이언트 APP는 신규 메시지를 작성할 수 있는 사용자 인터페이스를 제공하여야 하며, 작성된 문서를 유통 메시징서버 시스템과 연계하여 수신 상대방에게 전달하여야 한다.
- [1285] - 메시지 작성 기능은 메시지 전송하기 위해 유통 메시징서버 시스템의 전송인터페이스를 호출할 때, 필요한 기본정보들 중 환경정보에 의해 기 설정된 값이 아닌 항목은 입력을 할 수 있도록 제공하여야 한다.

[1286] ③ 메시지 목록조회 및 상세내용 열람 기능

- [1287] - 유통 메시징서버 시스템은 메시지를 송신메시지, 수신메시지로 구분하여 관리를 합니다. 유통 클라이언트 APP는 로그인된 사용자 계정을 기반으로 유통 메시징서버 시스템과 연계하여, 사용자 계정에 해당하는 각 메시지의 목록을 조회하는 기능과 메시지의 상세내용을 보고자 할 때 첨부문서를 포함하여 메시지의 상세정보를 모두 열람할 수 있는 기능을 반드시 제공하여야 합니다.

[1288] ④ 유통 메시징서버 시스템 연계

- [1289] - 유통 클라이언트 APP의 가장 중요한 기능이 유통 메시징서버 시스템과 연계하여 메시지를 송수신하는 기능이다. 유통 클라이언트 APP는 유통 메시징서버 시스템이 제공하는 메시지 송신기능과 수신메시지 읽음 인터페이스를 통해서 로그인한 계정을 기반으로 메시지를 전송하고 수신한다.

[1290] ⑤ 기본정보와 환경정보 관리

- [1291] - 클라이언트 APP는 메시지 전송 시 기본적으로 필요한 환경정보를 관리하는 기능을 제공하여야 한다. 유통 클라이언트 APP는 독립적으로 존재하는 어플리케이션이 아니므로 반드시 유통 메시징서버 시스템과의 연계를 통해

유통 기반인프라에 참여가 가능하다. 따라서 기본적으로 유통 메시징서버 시스템과의 연계를 위해 필요한 유통 메시징서버 시스템 연계정보(유통 메시징서버 시스템 주소정보)를 기본적으로 설정하고 관리하여야 한다.

- [1292] - 그 외에 문서 서식 등록기와의 연계를 위한 등록기 서버 정보 관리나, 유통 클라이언트 APP의 시스템 환경에 대한 부가정보를 관리하는 어플리케이션의 개발범위에 따라 정의하여 제공하면 된다.

[1293] **㉔ 메시지 폴더 관리**

- [1294] - 유통 메시징서버 시스템이 관리하는 메시지는 송신, 수신메시지를 기본으로 분류하여 관리하며, 송수신 메시지는 각각 처리 상태에 따라 상태정보를 관리한다. 각 메시지의 상태정보로, 송신메시지는 송신 전, 송신완료, 송신실패, 담당자 수신완료의 상태를, 수신메시지는 검증오류, 수신 확인 전, 수신확인, 열람확인의 상태를 관리하며, 유통 클라이언트 APP는 유통 메시징서버 시스템이 제공하는 기본 상태정보를 바탕으로 메시지 폴더를 관리하여 사용자에게 제공한다.

- [1295] - 유통 클라이언트 APP는 송수신 폴더를 기준으로 송신과 수신 메시지를 구분하여 유통 메시징서버 시스템이 제공하는 상태정보에 따라 사용자에게 각 메시지의 상태를 알려주는 것을 기본으로 제공하여야 한다. 그러나, 그 외에 임시보관함, 휴지통과 같은 지운 메시지함을 제공하거나 사용자가 직접 폴더를 정의하고 관리할 수 있도록 하는 기능을 제공하는 것은 어플리케이션 개발자의 선택사항이므로 본 발명의 설명에서는 생략한다.

[1296] **㉕ 문서 서식 관리 기능**

- [1297] - 유통 메시징서버 시스템은 전송하는 메시지에 첨부되는 문서의 양식을 제한하지 않으므로 송수신 대상 문서로는 일반 텍스트파일부터, 오피스 파일, XML 문서, 멀티미디어 파일 등 어떠한 종류의 파일도 모두 가능하다. 그러나, XML 문서, 멀티미디어 파일 등 어떠한 종류의 파일도 모두 가능하기 위해 사용자들이 유통 클라이언트 APP를 업무에 활용하도록 편의를 제공하기 위해 기본적인 문서에 대해서는 서식 기반으로 문서 작성을 지원하는 기능을 부가적으로 제공하는 것이 가능하다. 유통 클라이언트 APP는 문서 서식 등록기에서 제공하는 문서서식을 등록기의 표준 인터페이스를 통해 검색하여 다운로드한 후, 다운로드한 문서서식을 기반으로 문서를 작성하고 이를 메시지에 첨부하여 보낼 수 있는 기능을 제공할 수 있다.

- [1298] - 유통 클라이언트 APP는 전자문서 유통허브에서 제공하는 문서 서식 등록기와 연계하여 해당 문서 서식 등록기와 연계하여 문서서식을 관리할 수도 있으며, 독자적으로 문서 서식 관리체계를 구축하고 이 체제와 연계하여 문서 서식을 관리하는 방법이 있다. 전자문서 유통허브에서 제공하는 문서 서식 등록기와 연계하여 서식을 검색하고 다운로드 하는 방법에 대한 상세한 내용은 상술한 [전자문서 서식 등록기]에 대한 설명을 참조한다.

[1299] **㉖ 문서 작성기**

- [1300] - 문서 작성기는 문서 서식 관리 기능을 통해 유통 클라이언트 APP가

다운로드한 서식을 기반으로 사용자들이 문서를 작성할 수 있도록 지원하는 작성기이다. 문서작성기는 전자문서 유통허브가 제공하는 문서 서식 등록기를 이용하는 경우에는 상술한 [전자문서 서식 등록기]에 대한 설명을 참조하여 설계하면 되며, 자체적으로 문서 서식 관리체계를 구축한 경우에는 해당 서식 관리체계에 따라 문서 작성기를 설계하면 된다.

[1301]

[1302] 상기와 같은 유통 클라이언트 APP의 가장 기본이 되는 프로세스로는 "① 문서 전송 프로세스", "② 문서 수신 프로세스"가 있으며, 부가적 프로세스로는 "③ 전자문서 서식 다운로드 프로세스"가 있다. 문서 전송 및 문서 수신을 위해 유통 클라이언트 APP는 서버가 되는 유통 메시징서버 시스템과 연계되며, 표준문서양식 등록을 위해서는 전자문서 서식 등록기 서버와 연계가 된다.

[1303] ① 문서 전송 프로세스

[1304] 유통 클라이언트 APP가 연계된 유통 메시징서버 시스템을 통해 타 "송수신 개체"에게 전자문서를 전송하는 단계는 도 110과 같으며, 처리 절차는 아래와 같다.

[1305] 먼저, 유통 클라이언트 APP를 통해 수신자에게 전송할 메시지를 생성. 이 때 메시지는 이미 송신자가 작성한 문서를 첨부하거나, 유통 클라이언트 APP가 제공하는 문서작성기를 통해 작성한 문서를 첨부하여 수신자를 지정한 후 메시지를 생성한다.

[1306] 다음으로, 수신자 주소정보를 입력한 후, 유통 메시징서버 시스템의 전송 인터페이스를 호출함으로써 메시지 전송을 요청한다.

[1307] 다음으로, 전송자의 유통 메시징서버 시스템은 전송 프로세스에 따라 수신자에게 메시지를 전송한 후, 수신자로부터 수신에 대한 응답메시지(수신증명서 또는 수신오류)를 수신한다.

[1308] 다음으로, 전송자의 유통 메시징서버 시스템은 수신에 대한 응답메시지를 수신한 후 유통 클라이언트 APP에 전송에 대한 응답으로 전달한다.

[1309] 여기서, 첫번째~네번째 단계는 필수 절차이며, 두번째~네번째의 단계는 반드시 동기식으로 이루어져야 한다.

[1310] 다음으로, 전송자의 유통 메시징서버 시스템이 수신자로부터 수신담당자의 열람을 확인 하는 열람증명서를 포함한 메시지를 받으면, 전송 유통 메시징서버 시스템은 수신에 대한 응답메시지를 돌려주고 수신 메시지를 해당 사용자의 사서함에 보관한다.

[1311] 다음으로, 최초 전송자의 유통 클라이언트 APP은 연계된 유통 메시징서버 시스템에 수신문서를 요청한다.

[1312] 다음으로, 전송자의 유통 메시징서버 시스템은 사서함에 보관된 수신 문서목록을 수신문서를 요청한 사용자의 유통 클라이언트 APP에 전달한다.

[1313] 여기서, 다섯번째~일곱번째의 단계는 선택적 사항으로 최초 메시지 전송 시에 수신담당자의 열람 확인을 요청한 경우에만 발생하는 선택적 절차이다.

[1314]    **② 문서 수신 프로세스**

- [1315]    유통 클라이언트 APP가 타 "송수신 개체"로부터 전자문서를 수신하는 프로세스는 도 111과 같으며, 처리 절차는 다음과 같다.
- [1316]    먼저, 수신자의 유통 메시징서버 시스템은 메시지를 수신하면, 수신한 메시지에 대한 수신 응답메시지를 수신자에게 돌려주고, 수신 메시지를 해당 사용자의 사서함에 보관한다.
- [1317]    다음으로, 수신자의 유통 클라이언트 APP은 연계된 유통 메시징서버 시스템에 로그인을 한 후 수신문서를 요청한다.
- [1318]    다음으로, 수신자의 유통 메시징서버 시스템은 수신문서를 요청한 사용자의 사서함에 보관된 수신 문서목록을 전달한다.
- [1319]    여기서, 두번째~세번째 단계는 동기식이다.
- [1320]    다음으로, 수신자가 수신메시지의 목록에서 메시지에 대한 상세정보 보기를 요청하면, 유통 클라이언트 APP은 유통 메시징서버 시스템에 해당 메시지의 첨부문서를 포함한 상세정보를 전달한다.
- [1321]    다음으로, 최초 전송자가 수신 담당자의 열람확인을 요청한 경우, 수신자의 유통 메시징서버 시스템은 사용자가 수신문서에 대한 상세정보요청을 한 시점에 해당 메시지의 송신자에게 열람증명서를 포함한 메시지를 전송한다.
- [1322]    다음으로, 수신자의 유통 메시징서버 시스템은 다섯 번째 단계에서 전송된 담당자 열람확인 메시지(열람증명서)에 대한 수신 응답메시지를 수신한다.

[1323]    **③ 전자문서 서식 다운로드 프로세스**

- [1324]    유통 클라이언트 APP가 전자문서 서식을 다운로드하는 프로세스는 도 112와 같으며, 처리 절차는 다음과 같다.
- [1325]    먼저, 유통 클라이언트 APP는 전자문서 서식 등록기 서버에 직접 연계하여 문서서식에 대한 검색을 요청한다. 이때, 전자문서 서식 등록기 서버가 제공하는 표준 연계인터페이스 기반으로 연계한다.
- [1326]    다음으로, 전자문서 서식 등록기 서버는 검색된 문서 서식에 대한 정보를 결과로 돌려 준다.
- [1327]    이때, 첫번째~두번째 단계는 동기식이다.
- [1328]    다음으로, 유통 클라이언트 APP는 검색된 서식 목록을 사용자에게 보여줌으로써 사용자가 서식을 선택할 수 있도록 한다.
- [1329]    다음으로, 유통 클라이언트 APP는 전자문서 서식 등록기 서버에 선택된 전자문서 서식에 대한 다운로드를 요청한다.
- [1330]    다음으로, 전자문서 서식 등록기 서버는 요청 받은 서식을 유통 클라이언트 APP에 되돌려 준다.
- [1331]    다음으로, 유통 클라이언트 APP는 다운로드 받은 전자문서 서식을 등록하여 문서 작성기에서 사용할 수 있도록 Plug-In한다.
- [1332]
- [1333]    상기와 같은 유통 클라이언트 APP를 위해 유통 메시징서버 시스템이 제공하는

인터페이스 유형으로는 사용자 인증(로그인), 로그아웃, 메시지 전송요청, 수신메시지 Get, 메시지 상세정보 요청, 메시지 삭제가 있다.

[1334] 유통 클라이언트 APP와 유통 메시징서버 시스템의 연계 방안(①~⑤)에 대하여 설명하면 다음과 같다.

[1335] ① 유통 메시징서버 시스템의 연계 프로토콜

[1336] 유통 메시징서버 시스템이 유통 클라이언트 APP를 위해 제공하는 연계 인터페이스는 유통 메시징서버 시스템의 송수신 프로토콜과 동일한 프로토콜을 기반으로 한다. 다만, 유통 메시징서버 시스템들 간에 송수신하는 경우와 달리, 유통 클라이언트 APP와 유통 메시징서버 시스템은 도 113과 같이 단방향(One-Way) 동기식 통신만을 제공하며, 둘 간에는 메시지에 대한 전자서명 인증 또는 사용자 인증 방식을 사용한다.

[1337] 전송메시지는 유통 메시징서버 시스템의 메시지 구조를 그대로 활용하되, 사용자 정보 및 요청과 응답메시지는 도 114와 같은 구조로 구성되며, 상세한 설명은 다음과 같다.

[1338] - SOAP Header : 유통 클라이언트 APP 및 유통 메시징서버 시스템이 업무 유형에 따라 송신자 또는 수신자가 되어 상술한 [유통 프로토콜]에 따라 구성되며, messageHeader 및 Signature 정보로 구성된다.

[1339] - SOAP Body : 상술한 [유통 프로토콜]에서 정의한 Manifest 요소 정보 및 사용자 로그인 정보가 들어간다.

[1340] - 전송문서 컨테이너 #1 : 메시지 전송 요청, 수신 메시지 Get, 메시지 상세정보 수신인 경우 본문 문서(Contents)가 들어간다.

[1341] - 전송문서 컨테이너 #2 : 메시지 전송 요청, 메시지 상세정보 수신인 경우 첨부 문서가 #2부터 순차적으로 들어간다.

[1342] SOAP Header의 구조는 아래 표171과 같으며, MessageHeader의 구조는 아래 표172와 같고, SOAP Body의 구조는 아래 표173과 같으며, 본문 메시지의 구조는 아래 표174와 같다.

[1343] 표 171

| 항목명           | 설명                                | 반복<br>회수 | 유형 | 길이 |
|---------------|-----------------------------------|----------|----|----|
| MessageHeader | ▪ 아래 MessageHeader 구조 참조          | 1..1     |    |    |
| SyncReply     | ▪ 동기식 송신을 의미<br>▪ "유통 프로토콜 규격" 참조 | 1..1     |    |    |
| Signature     | ▪ 전자서명 요소<br>▪ "유통 프로토콜 규격" 참조    | 1..1     |    |    |

[1344] 표 172

| 항목명  | 설명         |                                                                                            | 반복<br>회수 | 유형 | 길이 |
|------|------------|--------------------------------------------------------------------------------------------|----------|----|----|
| From | 메시지 송신자 정보 |                                                                                            | 1..1     |    |    |
|      | PartyId    | 고정값: "clientapp"                                                                           | 1..1     | S  | 9  |
|      | Role       | <ul style="list-style-type: none"> <li>• 송신자 역할</li> <li>• 사용 시 고정값: "Sender"</li> </ul>   | 0..1     | S  | 6  |
| To   | 메시지 수신자 정보 |                                                                                            | 1..1     |    |    |
|      | PartyId    | 고정값: "openapi"                                                                             | 1..1     | S  | 7  |
|      | Role       | <ul style="list-style-type: none"> <li>• 수신자 역할</li> <li>• 사용 시 고정값: "Receiver"</li> </ul> | 0..1     | S  | 8  |

[1345]

|                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                    |                                                                                                                                                                    |        |        |      |      |        |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------|------|------|--------|-----|
| CPAID                                                                                                                                                                                            | <ul style="list-style-type: none"> <li>거래협업정의서 아이디</li> <li>clientapp-openapi 값을 반드시 사용</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                          | 1..1                                                                                                                                                               | S                                                                                                                                                                  | 17     |        |      |      |        |     |
| ConversationId                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>송수신 트랜잭션 구분자</li> <li>영문을 이용하여 길이 256 이하로 임의의 값을 설정</li> <li>ex&gt; 2e2c2b1-4603-4919-9bef-ee435bc56cb3</li> </ul>                                                                                                                                                                                                                                                                                                                                   | 1..1                                                                                                                                                               | S                                                                                                                                                                  | 최대 256 |        |      |      |        |     |
| Service                                                                                                                                                                                          | <ul style="list-style-type: none"> <li>CPA에 정의된 메시지 서비스</li> <li>urn:ebxml:nipa:ClientTo0a 값을 반드시 사용</li> </ul>                                                                                                                                                                                                                                                                                                                                                                             | 1..1                                                                                                                                                               | S                                                                                                                                                                  | 25     |        |      |      |        |     |
| Action                                                                                                                                                                                           | <ul style="list-style-type: none"> <li>Service 내의 특정 업무 프로세스 구분자 <ul style="list-style-type: none"> <li>- 문서 송신 요청의 경우: RequestSend</li> <li>- 문서 송신 요청에 대한 응답의 경우: RespondSend</li> <li>- 문서 수신 요청의 경우: RequestReceive</li> <li>- 문서 수신 요청에 대한 응답의 경우: RespondReceive</li> <li>- 문서 상세 정보 수신 요청의 경우: RequestDetailInfo</li> <li>- 문서 상세 정보 수신 요청에 대한 응답의 경우: RespondDetailInfo</li> <li>- 문서 삭제 요청의 경우: RequestDelete</li> <li>- 문서 삭제 요청에 대한 응답의 경우: RespondDelete</li> </ul> </li> </ul> | 1..1                                                                                                                                                               | S                                                                                                                                                                  | 11     |        |      |      |        |     |
| MessageData                                                                                                                                                                                      | <ul style="list-style-type: none"> <li>메시지를 식별하기 위한 데이터</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                          | 1..1                                                                                                                                                               |                                                                                                                                                                    |        |        |      |      |        |     |
|                                                                                                                                                                                                  | <table> <tr> <td>MessageId</td> <td> <ul style="list-style-type: none"> <li>하나의 메시지가 가지는 유일한 식별자</li> <li>영문을 이용하여 길이 256 이하로 임의의 값을 설정</li> <li>ex&gt; 42e2c2b1-4603-4919-9bef-ee435bc56cb3</li> </ul> </td> <td>1..1</td> <td>S</td> <td>256</td> </tr> </table>                                                                                                                                                                                                                          | MessageId                                                                                                                                                          | <ul style="list-style-type: none"> <li>하나의 메시지가 가지는 유일한 식별자</li> <li>영문을 이용하여 길이 256 이하로 임의의 값을 설정</li> <li>ex&gt; 42e2c2b1-4603-4919-9bef-ee435bc56cb3</li> </ul> | 1..1   | S      | 256  | 1..1 | S      | 256 |
|                                                                                                                                                                                                  | MessageId                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>하나의 메시지가 가지는 유일한 식별자</li> <li>영문을 이용하여 길이 256 이하로 임의의 값을 설정</li> <li>ex&gt; 42e2c2b1-4603-4919-9bef-ee435bc56cb3</li> </ul> | 1..1                                                                                                                                                               | S      | 256    |      |      |        |     |
|                                                                                                                                                                                                  | <table> <tr> <td>TimeStamp</td> <td> <ul style="list-style-type: none"> <li>메시지 생성 시간</li> <li>UTC 형식</li> <li>ex&gt; 2008-07-31T06:29:39.724Z</li> </ul> </td> <td>1..1</td> <td>S</td> <td>24</td> </tr> </table>                                                                                                                                                                                                                                                                         | TimeStamp                                                                                                                                                          | <ul style="list-style-type: none"> <li>메시지 생성 시간</li> <li>UTC 형식</li> <li>ex&gt; 2008-07-31T06:29:39.724Z</li> </ul>                                               | 1..1   | S      | 24   | 1..1 | S      | 24  |
| TimeStamp                                                                                                                                                                                        | <ul style="list-style-type: none"> <li>메시지 생성 시간</li> <li>UTC 형식</li> <li>ex&gt; 2008-07-31T06:29:39.724Z</li> </ul>                                                                                                                                                                                                                                                                                                                                                                        | 1..1                                                                                                                                                               | S                                                                                                                                                                  | 24     |        |      |      |        |     |
| <table> <tr> <td>RefToMessageId</td> <td> <ul style="list-style-type: none"> <li>응답 메시지에만 존재</li> <li>요청 메시지의 MessageId</li> </ul> </td> <td>1..1</td> <td>S</td> <td>최대 256</td> </tr> </table> | RefToMessageId                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <ul style="list-style-type: none"> <li>응답 메시지에만 존재</li> <li>요청 메시지의 MessageId</li> </ul>                                                                           | 1..1                                                                                                                                                               | S      | 최대 256 | 1..1 | S    | 최대 256 |     |
| RefToMessageId                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>응답 메시지에만 존재</li> <li>요청 메시지의 MessageId</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                    | 1..1                                                                                                                                                               | S                                                                                                                                                                  | 최대 256 |        |      |      |        |     |

[1346] 표 173

| 항목명       | 설명                                                                                                                                                                                                                         | 반복<br>회수  | 유형                                                                                                                       | 길이   |  |  |  |  |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------|------|--|--|--|--|
| Manifest  | <ul style="list-style-type: none"> <li>페이로드 정보</li> <li>첨부 문서가 있는 경우에만 존재(메시지 전송 요청, 수신 메시지 Get, 메시지 상세정보 수신)</li> </ul>                                                                                                   | 0..1      |                                                                                                                          |      |  |  |  |  |
|           | <table border="1"> <tr> <td>Reference</td><td> <ul style="list-style-type: none"> <li>Manifest가 있는 경우에는 반드시 존재</li> <li>페이로드 수만을 반복</li> <li>"유형 프로토콜 규칙 참조"</li> </ul> </td><td>0..∞</td><td></td><td></td></tr> </table> | Reference | <ul style="list-style-type: none"> <li>Manifest가 있는 경우에는 반드시 존재</li> <li>페이로드 수만을 반복</li> <li>"유형 프로토콜 규칙 참조"</li> </ul> | 0..∞ |  |  |  |  |
| Reference | <ul style="list-style-type: none"> <li>Manifest가 있는 경우에는 반드시 존재</li> <li>페이로드 수만을 반복</li> <li>"유형 프로토콜 규칙 참조"</li> </ul>                                                                                                   | 0..∞      |                                                                                                                          |      |  |  |  |  |
| UserInfo  | <ul style="list-style-type: none"> <li>사용자 정보</li> <li>메시지 전송 요청, 수신 메시지 Get, 메시지 상세정보 수신의 경우에만 반드시 존재</li> </ul>                                                                                                          | 0..1      |                                                                                                                          |      |  |  |  |  |
|           | Id                                                                                                                                                                                                                         | 0..1      | S                                                                                                                        | 20   |  |  |  |  |
|           | Password                                                                                                                                                                                                                   | 0..1      | S                                                                                                                        | 15   |  |  |  |  |

[1347] 표 174

| 항목명             | 설명                                                                                                         | 반복<br>회수 | 유형 | 길이   |
|-----------------|------------------------------------------------------------------------------------------------------------|----------|----|------|
| Content         | <ul style="list-style-type: none"> <li>문서 상세 정보</li> <li>문서 Get의 경우 문서 목록 갯수 만큼 존재</li> </ul>              | 1..∞     | S  | 200  |
| Title           | <ul style="list-style-type: none"> <li>문서 제목</li> <li>문서 송신 요청, 문서 Get 응답, 문서 상세 정보 응답에만 존재</li> </ul>     | 0..1     | S  | 200  |
| Text            | <ul style="list-style-type: none"> <li>사용자 커멘트</li> <li>문서 송신 요청, 문서 Get 응답, 문서 상세 정보 응답에만 존재</li> </ul>   | 0..1     | S  | 5000 |
| Sender          | <ul style="list-style-type: none"> <li>송신자 신뢰 주소</li> <li>문서 송신 요청, 문서 Get 응답, 문서 상세 정보 응답에만 존재</li> </ul> | 0..1     | S  | 256  |
| Receiver        | <ul style="list-style-type: none"> <li>수신자 신뢰주소</li> <li>문서 송신 요청, 문서 Get 응답, 문서 상세 정보 응답에만 존재</li> </ul>  | 0..1     | S  | 256  |
| GetMessageId    | <ul style="list-style-type: none"> <li>상세 정보 요청 MessageId</li> <li>상세 정보 요청의 경우에만 존재</li> </ul>            | 0..1     | S  | 256  |
| DeleteMessageId | <ul style="list-style-type: none"> <li>삭제 요청 MessageId</li> <li>삭제 요청의 경우에만 존재</li> </ul>                  | 0..1     | S  | 256  |

## [1348] ② 메시지 전송요청

[1349] 메시지 전송 시에 유통 클라이언트 APP가 유통 메시징서버 시스템에 전달해야 하는 기본정보는 다음과 같다. 전송 완료된 후 사서함에 보관된 송신문서는 아래 표 175와 같이 4단계의 상태정보를 가진다.

[1350] 표 175

| 상태    | 설명                                                                            |
|-------|-------------------------------------------------------------------------------|
| 송신 중  | 문서 전송 후 수신자로부터 아무런 응답을 받지 못한 상태                                               |
| 송신 완료 | 수신자로부터 수신확인 위한 응답메시지(수신증명서)를 받은 상태                                            |
| 송신 실패 | 수신 유통 메시징서버 시스템 내부에서 오류가 발생하여 SOAP Fault 메시지를 리턴하거나, 송수신 과정에서 네트워크 오류가 발생한 경우 |
| 접수 완료 | 수신사용자가 수신문서에 대한 상세내용(첨부파일 포함)을 열람했는지 여부                                       |

[1351] 요청 메시지의 예제는 아래 표 176과 같다.

[1352] 표 176

---

Content-type: multipart/related; boundary="Boundary"; type="text/xml";  
start="<ebxhmheader111@example.com>"

--Boundary

Content-ID: <ebxhmheader111@example.com>

Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>

<SOAP:Envelope ...>

<SOAP:Header>

<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">

<eb:From>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

clientapp

</eb:PartyId>

</eb:From>

<eb:To>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

openapi

</eb:PartyId>

```

[1353] </eb:From>
 <eb:To>
 <eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">
 openapi
 </eb:PartyId>
 </eb:To>
 <eb:CPAId>clientapp-openapi</eb:CPAId>
 <eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>
 <eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>
 <eb:Action>RequestSend</eb:Action>
 <eb:MessageData>
 <eb:MessageId>20100209-133003-28572@example.com</eb:MessageId>
 <eb:Timestamp>2010-02-09T13:30:03</eb:Timestamp>
 </eb:MessageData>
 </eb:MessageHeader>
 <eb:SyncReply eb:id="3833kig9" eb:version="2.0" SOAP:mustUnderstand="1"
 SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
 <Signature ...> ... </Signature>
 </SOAP:Header>
 <SOAP:Body>
 <eb:Manifest eb:version="2.0">
 <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
 xlinkrole="XLinkRole" xlinktype="simple">
 </eb:Reference>
 <eb:Reference xlink:href="cid:ebxmlpayload222@example.com"
 xlinkrole="XLinkRole" xlinktype="simple">
 </eb:Reference>
 </eb:Manifest>
 <UserInfo>
 <Id>gazuo</Id>
 <password>FJDHS6IDFJFMCD</passwd>
 </UserInfo>
 </SOAP:Body>
 </SOAP:Envelope>

```

[1354]

--Boundary

Content-ID: &lt;ebxmlpayload111@example.com&gt;

Content-Type: text/xml

&lt;Contents&gt;

&lt;Content&gt;

&lt;Title&gt; 테스트 &lt;/Title&gt;

&lt;Text&gt;테스트 메시지 입니다.&lt;/Text&gt;

&lt;Sender&gt;gazuo#ceda001&lt;/Sender&gt;

&lt;Receiver&gt;jnlee#ceda002&lt;/Receiver&gt;

&lt;/Content&gt;

&lt;/Contents&gt;

--Boundary

Content-ID: &lt;ebxmlpayload222@example.com&gt;

Content-Type: application/octet-stream

실제 첨부 문서

--Boundary--

[1355] 응답 메시지의 예제는 아래 표 177과 같다.

[1356] 표 177

---

```

Content-type: multipart/related; boundary="Boundary"; type="text/xml";
start="<ebxhmheader111@example.com>"

--Boundary
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope ...>
 <SOAP:Header>
 <eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
 <eb:From>
 <eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">
 openapi
 </eb:PartyId>
 </eb:From>
 <eb:To>
 <eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">
 clientapp
 </eb:PartyId>
 </eb:To>
 <eb:CPAid>clientapp-openapi</eb:CPAid>
 <eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>
 <eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>
 <eb:Action>RespondSend</eb:Action>
 <eb:MessageData>
 <eb:MessageId>20100209-133004-38572@example.com</eb:MessageId>
 <eb:Timestamp>2010-02-09T13:31:03</eb:Timestamp>
 </eb:MessageData>
 <eb:RefToMessageId>20100209-133003-28572@example.com</eb:RefToMessageId>
 </eb:MessageHeader>
 <eb:SyncReply eb:id="3833kkg9" eb:version="2.0" SOAP:mustUnderstand="1"
 SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
 <Signature ...> ... </Signature>
 </SOAP:Header>
 <SOAP:Body/>
</SOAP:Envelope>

--Boundary--

```

---

[1357]     **③ 수신 메시지 Get**

- [1358]     유통 클라이언트 APP가 유통 메시징서버 시스템과 연계하여 로그인 한 사용자 계정으로 수신메시지를 읽어오는 행위와 유통 메시징서버 시스템에서 메시지를 삭제하는 행위는 분리가 되어 있다. 메시지 수신의 각 프로세스에 따라 다음과

같은 2단계의 상대정보를 관리해야 한다.

- [1359] - 수신사용자가 사서함의 수신 문서 목록을 열람했는지 여부
- [1360] - 수신사용자가 수신문서에 대한 상세내용을 열람했는지 여부
- [1361] 요청 메시지의 예제는 아래 표 178과 같다.
- [1362] 표 178

```

 <eb:To>
 <eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">
 openapi
 </eb:PartyId>
 </eb:To>
 <eb:CPAid>clientapp-openapi</eb:CPAid>
 <eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>
 <eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>
 <eb:Action>RequestReceive</eb:Action>
 <eb:MessageData>
 <eb:MessageId>20100209-133003-28572@example.com</eb:MessageId>
 <eb:Timestamp>2010-02-09T13:30:03</eb:Timestamp>
 </eb:MessageData>
 </eb:MessageHeader>
 <eb:SyncReply eb:id="3833kkj9" eb:version="2.0" SOAP:mustUnderstand="1"
 SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
 <Signature ...> ... </Signature>
 </SOAP:Header>
 <SOAP:Body>
 <UserInfo>
 <Id>gazio</Id>
 <password>FJDHS6IDFJFMCD</passwd>
 </UserInfo>
 </SOAP:Body>
</SOAP:Envelope>

```

--Boundary--

[1363]

```

<eb:To>
 <eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">
 openapi
 </eb:PartyId>
</eb:To>
<eb:CPAId>clientapp-openapi</eb:CPAId>
<eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>
<eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>
<eb:Action>RequestReceive</eb:Action>
<eb:MessageData>
 <eb:MessageId>20100209-133003-28572@example.com</eb:MessageId>
 <eb:Timestamp>2010-02-09T13:30:03</eb:Timestamp>
</eb:MessageData>
</eb:MessageHeader>
<eb:SyncReply eb:id="3833ldq9" eb:version="2.0" SOAP:mustUnderstand="1"
 SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
 <Signature ...> ... </Signature>
</SOAP:Header>
<SOAP:Body>
 <UserInfo>
 <Id>gazin</Id>
 <password>FJDH56IDFJFMCD</password>
 </UserInfo>
</SOAP:Body>
</SOAP:Envelope>

--Boundary--

```

[1364] 응답 메시지의 예제는 아래 표 179와 같다.

[1365] 표 179

---

```

Content-type: multipart/related; boundary="Boundary"; type="text/xml";
start="<ebxhmheader111@example.com>"

```

```

--Boundary
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml

```

[1366]

--Boundary

Content-ID: &lt;ebxmhheader111@example.com&gt;

Content-Type: text/xml

&lt;?xml version='1.0' encoding="UTF-8"?&gt;

&lt;SOAP:Envelope ...&gt;

&lt;SOAP:Header&gt;

&lt;eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0"&gt;

&lt;eb:From&gt;

&lt;eb:Partyid eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns"&gt;

openapi

&lt;/eb:Partyid&gt;

&lt;/eb:From&gt;

&lt;eb:To&gt;

&lt;eb:Partyid eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns"&gt;

clientapp

&lt;/eb:Partyid&gt;

&lt;/eb:To&gt;

&lt;eb:CPAId&gt;clientapp-openapi&lt;/eb:CPAId&gt;

&lt;eb:ConversationId&gt;42e2c2b1-4603-4919-9bef-ee435bc56cb3&lt;/eb:ConversationId&gt;

&lt;eb:Service&gt;urn:ebxml:nipa:ClientToOa&lt;/eb:Service&gt;

&lt;eb:Action&gt;RespondReceive&lt;/eb:Action&gt;

&lt;eb:MessageData&gt;

&lt;eb:MessageId&gt;20100209-133004-38572@example.com&lt;/eb:MessageId&gt;

&lt;eb:Timestamp&gt;2010-02-09T13:31:03&lt;/eb:Timestamp&gt;

&lt;/eb:MessageData&gt;

&lt;eb:RefToMessageId&gt;20100209-133003-28572@example.com&lt;/eb:RefToMessageId&gt;

&lt;/eb:MessageHeader&gt;

&lt;eb:SyncReply eb:id="3833kkj9" eb:version="2.0" SOAP:mustUnderstand="1"

SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/&gt;

&lt;Signature ...&gt; ... &lt;/Signature&gt;

&lt;/SOAP:Header&gt;

&lt;SOAP:Body&gt;

&lt;eb:Manifest eb:version="2.0"&gt;

&lt;eb:Reference xlink:href="cid:ebxmlpayload111@example.com"

xlink:role="XLinkRole" xlink:type="simple"&gt;

&lt;/eb:Reference&gt;

```

[1367] </SOAP:Body>
 </SOAP:Envelope>

 --Boundary
 Content-ID: <ebxmlpayload111@example.com>
 Content-Type: text/xml

 <Contents>
 <Content>
 <Title>테스트</Title>
 <Text>테스트 메시지입니다.</Text>
 <Sender>gazuo#ceda001</Sender>
 <Receiver>jnlee#ceda002</Receiver>
 </Content>
 <Content>
 <Title>테스트2</Title>
 <Text>테스트2 메시지입니다.</Text>
 <Sender>gazuo#ceda001</Sender>
 <Receiver>jnlee#ceda002</Receiver>
 </Content>
 </Contents>

 --Boundary--

```

[1368]    ④ 메시지 상세정보 요청

[1369]    수신한 문서목록을 기반으로 사용자가 상세내용을 열람하고자 하는 경우에 유통 클라이언트 APP는 유통 메시징서버 시스템의 메시지 상세정보를 요청한다. 상세정보를 요청 받은 유통 메시징서버 시스템은 메시지의 상세 속성정보와 해당 메시지의 첨부문서 등 모든 메시지 내용을 유통 클라이언트 APP에 응답메시지로 전달한다.

[1370]    요청 메시지의 예제는 아래 표 180과 같다.

[1371]    표 180

---

Content-type: multipart/related; boundary="BoundaryY"; type="text/xml";  
start="<ebxhmheader111@example.com>"

--BoundaryY

Content-ID: <ebxhmheader111@example.com>

Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>

<SOAP:Envelope ...>

<SOAP:Header>

<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">

<eb:From>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

clientapp

</eb:PartyId>

</eb:From>

<eb:To>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

openapi

</eb:PartyId>

</eb:To>

<eb:CPAId>clientapp-openapi</eb:CPAId>

<eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>

<eb:Service>urn:eb:ml:nipa:ClientToOa</eb:Service>

<eb:Action>RequestDetailInfo</eb:Action>

<eb:MessageData>

<eb:MessageId>20100209-133003-28572@example.com</eb:MessageId>

<eb:Timestamp>2010-02-09T13:30:03</eb:Timestamp>

</eb:MessageData>

</eb:MessageHeader>

<eb:SyncReply eb:id="3833kdj9" eb:version="2.0" SOAP:mustUnderstand="1"

SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>

<Signature ...> ... </Signature>

</SOAP:Header>

[1372]     <SOAP:Body>  
             <UserInfo>  
                 <Id>gazuo</Id>  
                 <Password>FJDHS6IDFJFMCD</Password>  
                 <GetMessageId>20100209-133003-28572@example.com</GetMessageId>  
             </UserInfo>  
         </SOAP:Body>  
     </SOAP:Envelope>

~Boundary~

[1373]     응답 메시지의 예제는 아래 표 181과 같다

[1374]     표 181

Content-type: multipart/related; boundary="Boundary"; type="text/xml";  
 start="<ebxmhheader111@example.com>"

~Boundary

Content-ID: <ebxmhheader111@example.com>

Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>

<SOAP:Envelope ...>

<SOAP:Header>

<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">

<eb:From>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

openapi

</eb:PartyId>

</eb:From>

<eb:To>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

clientapp

</eb:PartyId>

</eb:To>

<eb:CPAid>clientapp-openapi</eb:CPAid>

<eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>

<eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>

<eb:Action>RespondDetailInfo</eb:Action>

[1375]

Content-type: multipart/related; boundary="Boundary"; type="text/xml";  
start="<ebxmheader111@example.com>"

--Boundary

Content-ID: <ebxmheader111@example.com>

Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>

<SOAP:Envelope ...>

<SOAP:Header>

<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">

<eb:From>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

openapi

</eb:PartyId>

</eb:From>

<eb:To>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">

clientapp

</eb:PartyId>

</eb:To>

<eb:CPAId>clientapp-openapi</eb:CPAId>

<eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>

<eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>

<eb:Action>RespondDetailInfo</eb:Action>

[1376]

```

<eb:MessageData>
 <eb:MessageId>20100209-133004-38572@example.com</eb:MessageId>
 <eb:Timestamp>2010-02-09T13:31:03</eb:Timestamp>
</eb:MessageData>
 <eb:RefToMessageId>20100209-133003-28572@example.com</eb:RefToMessageId>
</eb:MessageHeader>
 <eb:SyncReply eb:id="3833kg9" eb:version="2.0" SOAP:mustUnderstand="1"
 SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
 <Signature ...> ... </Signature>
</SOAP:Header>
<SOAP:Body>
 <eb:Manifest eb:version="2.0">
 <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
 xlinkrole="XLinkRole" xlinktype="simple">
 </eb:Reference>
 <eb:Reference xlink:href="cid:ebxmlpayload222@example.com"
 xlinkrole="XLinkRole" xlinktype="simple">
 </eb:Reference>
 </eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>

--Boundary
Content-ID: <ebxmlpayload111@example.com>
Content-Type: text/xml

<Contents>
 <Content>
 <Title>테스트</Title>
 <Text>테스트 메시지입니다.</Text>
 <Sender>gazuo#ceda001</Sender>
 <Receiver>jnllee#ceda002</Receiver>
 </Content>
</Contents>

```

[1377]     <Contents>  
             <Content>  
                 <Title> 테스트 </Title>  
                 <Text> 테스트 메시지입니다. </Text>  
                 <Sender> gazuo#ceda001 </Sender>  
                 <Receiver> jnlee#ceda002 </Receiver>  
             </Content>  
           </Contents>  
  
           --BoundaryY  
           Content-ID: <ebxmlpayload222@example.com>  
           Content-Type: application/octet-stream  
  
           실제 첨부 문서  
  
           --BoundaryY--

---

[1378]     ⑤ 메시지 삭제

[1379]     유통 클라이언트 APP는 사용자가 삭제요청을 하는 경우에 유통 메시징서버 시스템에 해당 문서에 대한 삭제요청을 전달하고 그 결과를 사용자에게 알려주어야 한다. 사용자의 삭제 시 휴지통 개념의 임시 삭제 기능 부여 여부는 실제 서버상에서의 행위가 아니라 유통 클라이언트 APP의 부가기능이므로 유통 클라이언트 APP 개발자가 제공여부를 결정할 수 있으나, 최종적으로 유통 메시징서버 시스템에 삭제 요청을 하는 기능은 반드시 제공되어야 한다.

[1380]     요청 메시지의 예제는 아래 표 182와 같다.

[1381]     표 182

---

Content-type: multipart/related; boundary="Boundary"; type="text/xml";  
start="<ebxmhheader111@example.com>"

~Boundary

Content-ID: <ebxmhheader111@example.com>

Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>

<SOAP:Envelope ...>

<SOAP:Header>

<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">

<eb:From>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">  
clientapp

</eb:PartyId>

</eb:From>

<eb:To>

<eb:PartyId eb:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">  
openapi

</eb:PartyId>

</eb:To>

<eb:CPAId>clientapp-openapi</eb:CPAId>

<eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>

<eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>

<eb:Action>RequestDelete</eb:Action>

<eb:MessageData>

<eb:MessageId>20100209-133003-28572@example.com</eb:MessageId>

<eb:Timestamp>2010-02-09T13:30:03</eb:Timestamp>

</eb:MessageData>

</eb:MessageHeader>

<eb:SyncReply eb:id="3833kg9" eb:version="2.0" SOAP:mustUnderstand="1"

[1382] SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>  
 <Signature ...> ... </Signature>  
 </SOAP:Header>  
 <SOAP:Body>  
 <UserInfo>  
 <Id>gazuo</Id>  
 <Password>FJDH56IDFJFMCD</Passwd>  
 <DeleteMessageId>20100209-133003-28572@example.com</DeleteMessageId>  
 </UserInfo>  
 </SOAP:Body>  
 </SOAP:Envelope>  
 --Boundary--

[1383] 응답 메시지의 예제는 아래 표 183과 같다.

[1384] 표 183

Content-type: multipart/related: boundary="Boundary"; type="text/xml";  
 start="<ebxhmheader111@example.com>"

--Boundary

Content-ID: <ebxhmheader111@example.com>

Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>

<SOAP:Envelope ...>

<SOAP:Header>

<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">

<eb:From>

<eb:PartyId eb:type="urn:casis:names:tc:ebxml-cppa:partyid-type:duns">

openapi

</eb:PartyId>

</eb:From>

<eb:To>

<eb:PartyId eb:type="urn:casis:names:tc:ebxml-cppa:partyid-type:duns">

dientapp

</eb:PartyId>

[1385]

```

</eb:To>
<eb:CPAid>clientapp-openapi</eb:CPAid>
<eb:ConversationId>42e2c2b1-4603-4919-9bef-ee435bc56cb3</eb:ConversationId>
<eb:Service>urn:ebxml:nipa:ClientToOa</eb:Service>
<eb:Action>RespondDelete</eb:Action>
<eb:MessageData>
 <eb:MessageId>20100209-133004-38572@example.com</eb:MessageId>
 <eb:Timestamp>2010-02-09T13:31:03</eb:Timestamp>
</eb:MessageData>
<eb:RefToMessageId>20100209-133003-28572@example.com</eb:RefToMessageId>
</eb:MessageHeader>
<eb:SyncReply eb:id="3833kg9" eb:version="2.0" SOAP:mustUnderstand="1"
 SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
<Signature ...> ... </Signature>
</SOAP:Header>
<SOAP:Body/>
</SOAP:Envelope>

--Boundary--

```

---

[1386]

[1387] [기록매체]

[1388] 한편, 상술한 본 발명에 따른 전자문서 유통 방법은 컴퓨터에서 실행될 수 있는 프로그램으로 작성 가능하고, 컴퓨터로 읽을 수 있는 기록 매체를 이용하여 프로그램을 동작시키는 범용 디지털 컴퓨터에서 구현될 수 있다. 컴퓨터로 읽을 수 있는 기록 매체는 마그네틱 저장 매체(예: ROM, 플로피 디스크, 하드 디스크, 자기 테이프 등), 광학적 판독 매체(예: CD-ROM, DVD, 광데이터 저장 장치 등) 및 캐리어 웨이브(예를 들면, 인터넷을 통한 전송)와 같은 저장 매체를 포함한다.

[1389]

[1390] 이하, 상술한 바와 같은 본 발명에 있어서 주소디렉터리 서버와 관련하여 또다른 실시예에 대하여 설명하면 다음과 같다.

[1391]

[1392] [주소 디렉터리 서버]

[1393] 신뢰할 수 있는 전자문서유통에 참여하기 위해서 모든 사용자는 고유의 전자주소를 발급받아야 한다.

[1394] 전자주소는 다음과 같은 구조로 표현된다.

[1395] 전자주소: 내부 구분자 + 구분 기호 + 고유등록주소

[1396] 이와 같은 전자주소의 일 예로서 "gdhong#nipa.kr" 이 있다.

[1397] 상기 전자주소의 내부 구분자는 고유등록주소의 소유자가 내부적 처리의

- 편의를 위해 선택적으로 추가하는 정보이며, 필요에 따라 생략이 가능하다.
- [1398] 상기 전자주소의 구분 기호는 고유등록주소의 앞에 위치하거나, 내부 구분자와 고유등록주소의 사이에 존재하는 기호이며, 일 예로서 "#"이 가능하며, 필요에 따라 다른 기호를 사용할 수 있다.
- [1399] 상기 전자주소의 고유등록주소는 기업/기관/개인이 발급 요청한 고유의 ID 값이며, 구분 기호 뒤에 존재하는 고유등록주소 단위가 송수신에 대한 법적 책임 단위가 된다. 이와 같은 고유등록주소는 송수신 개체가 유통 메시징서버를 자체적으로 구축한 후에 발급받거나 또는 송수신 개체가 전자문서 3자 유통 대행 기관을 통해 발급받은 고유등록주소로서, 전자 주소의 필수 구성 요소이다.
- [1400] 상기 송수신 개체는 자신이 보유한 유통 메시징서버 시스템에 대한 실제 물리적 주소(IP Address)를 가지는데, 이러한 물리적 주소와 상기 전자주소는 연관 관계가 없으며, 물리적 주소와 전자주소는 1:N의 관계를 가진다. 하나의 전자주소가 여러 개의 물리적 주소를 갖는 경우는 존재하지 않는다.
- [1401] 전자주소에 대한 정보(전자 문서)의 법적 수신책임은 구분 기호 뒤에 존재하는 기업/기관/개인이 가져야 하며, 내부 구분자에 의한 배부는 기업/기관/개인이 편의를 위해 구분한 것이므로 자체적으로 책임을 져야 한다.
- [1402] 전자문서 유통 시스템 내에서 전자주소가 가지는 의미는 도 3에 도시한 전자문서 유통 시스템 참여자 관계도와 같이 표현될 수 있다.
- [1403] 이와 같이 내부 구분자와 고유등록주소를 가지는 전자주소에 대하여 한번 더 정리하면 다음과 같다.
- [1404] ① 내부 구분자
- [1405] - 내부 구분자는 주소 디렉터리 서버와 관계없이 각 송수신 개체가 자체적으로 발급하고 관리한다.
- [1406] - 내부 구분자는 송수신 개체 내에서는 고유한 값이어야 하며, 생략이 가능한 정보다.
- [1407] - 내부 구분자에 대한 부여방식은 각 기업/기관/개인이 책임을 지는 것을 기본으로 하며, 내부 구분자에 의한 전자문서의 배부 역시 전자문서유통 기반인프라 체계 하에서 공식적인 의미를 갖지는 않는다
- [1408] - 고유등록주소가 수신에 대한 책임을 질 수 있는 정부/공공/법인/기관/단체/개인이 3자 유통 가능한 송수신 개체에 계정을 개설하고 공식적으로 주소 디렉터리 서버에 등록하여 사용하는 개체라면, 내부 구분자는 기업의 업무편의를 위해 전자문서를 분배하기 위한 용도로 사용되며, 주소 디렉터리 서버에 등록하지 않고 기업내부의 정보로서만 사용한다.
- [1409] ② 고유등록주소
- [1410] - 전자문서 유통 시스템에 참여하여 전자문서를 유통하고자 하는 정부/공공/법인/기관/단체/개인은 유통 메시징서버 시스템을 자체적으로 구축한 후에 송수신 개체로서 고유등록주소를 발급받거나, 3자 유통(유통 대행)기관을 통해 고유등록주소를 발급받아야 한다.

- [1411] - 고유등록주소는 발급 시점에 주소디렉터리 서버에 고유등록주소의 유일성(uniqueness)을 확인함으로써 중복되어 발급되지 않도록 관리된다.
- [1412] - 정부/법인/기관/단체/개인의 고유등록주소의 구성 방식은 공인전자주소 관리 총괄의 정책에 의해 결정된다.
- [1413]
- [1414] 상술한 바와 같은 전자주소는 기본적으로 2-level에 의해 관리가 된다. 공인전자주소의 최상단에는 주소 디렉터리 서버를 관리하는 공인전자주소 관리 총괄(예: 정보통신산업진흥원)이 있으며, 공인전자주소 관리 총괄은 하위 송수신 개체에 대한 고유등록주소를 발급해주고 이를 관리한다. 공인전자주소 관리 총괄의 하위 송수신 개체 중에 3차 유통(유통 대행)이 가능한 송수신 개체는 3차 유통을 원하는 사용자에게 대한 등록주소를 개설한 후에 이에 대한 주소정보를 주소 디렉터리 서버에 등록한다. 이때, 사용자 고유등록주소 값의 유일성(uniqueness)을 보장하기 위해 반드시 주소디렉터리 서버에 중복 여부를 확인하여야 한다.
- [1415] 전자주소 중 공식적인 사용자가 아니라 내부에서 업무 편의를 위해 발급하고 사용하는 내부 구분자는 주소 디렉터리 서버와 관계 없이 각 송수신 개체가 자체적으로 발급하고 관리한다.
- [1416] 전자 주소를 발급하는 체계는 도 4와 같으며, 도 4에 나타난 각 구성 요소의 역할은 아래 표 184와 같다.
- [1417] 표 184

| 구성요소                                  | 역할                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 정보통신산업진흥원<br>(공인전자주소 관리 총괄)           | <ul style="list-style-type: none"> <li>- 정보통신산업진흥원은 공인전자주소의 최상위 관리주체로서 모든 공인전자주소 정보를 관리함</li> <li>- 송수신 개체에 대한 고유 ID를 발급해 줌</li> <li>- 사용자 계정에 대한 공인전자주소 등록요청 시, 신규 등록인지, 기존 주소 변경인지 여부를 알려 줌</li> <li>- 사용자 계정의 신규등록 시 계정 ID가 고유한지 여부를 알려 줌</li> <li>- 사용자의 물리적 주소정보 검색요청에 대해 결과 값 전달</li> </ul>                                                                        |
| 송수신 개체                                | <ul style="list-style-type: none"> <li>- 물리적 공인전자주소의 가장 기본이 되는 단위임</li> <li>- 하나의 공인전자주소 하위에 복수 사용자를 위한 사용자 계정 또는 내부 구분자 정보를 발급, 관리하며, 하나의 공인전자주소 내에서는 사용자 계정 또는 내부 구분자의 유일성을 보장</li> </ul>                                                                                                                                                                              |
| 사용자(개인, 기업, 기관 등) 계정 또는 내부 구분자를 위한 ID | <ul style="list-style-type: none"> <li>- 전자문서유통에 참여하는 실 사용자로서, 동일한 송수신 개체에서 ID를 발급 받은 경우 물리적 공인전자주소는 동일하나 신뢰 유통의 실질적인 기반이 되는 단위임</li> <li>- 사용자 계정은 3자 유통이 가능한 송수신 개체를 통해 개설한 전자주소로서 법적 책임을 가진 송수신 단위이며, 반드시 전자주소 디렉터리 서버에 등록하여야 함</li> <li>- 내부 구분자는 송수신에 대한 법적 책임을 가진 송수신 개체가 내부관리의 편의를 위해 관리하는 정보로서 법적 책임은 내부 구분자 상위의 송수신 개체가 지게 되며, 전자주소 디렉터리 서버에 등록되지 않음</li> </ul> |

- [1418] 전자주소를 발급하는 프로세스는 도 5와 같으며, 사용자(기업)가 직접 주소 디렉터리 서버가 제공하는 화면에 접속하여 주소를 등록하거나 수정하는 방법과, 공인전자주소를 대행 발급해 주는 유통 메시징 서버 시스템(시스템이 제공하는 웹 사이트)을 통해 발급받는 방법이 있다.
- [1419] 유통에 참여하는 사용자는 상대방에게 메시지를 전송하기 전에 전자주소 정보를 바탕으로 물리적인 실 주소 정보를 반드시 알아야 하며, 부가적으로 첨부하는 문서를 암호화하기 위해서는 수신자의 공개키 정보도 획득하여야 한다.
- [1420] 전자문서 유통프로세스에서 전자주소의 물리적 주소를 획득하는 절차는 필수 단계로서, 송신자는 수신자의 주소정보를 기준으로 수신 상대방에 대한 물리적 주소정보 및 보안정보 획득을 위해 주소 디렉터리 서버에 질의를 던지게 된다. 이 물리적 주소를 기준으로 송신자가 수신자에게 전송문서를 전달하면 수신자의 유통 메시징서버 시스템은 이를 받아서 수신자주소정보를 기반으로 사용자 계정 또는 내부 구분자에 따라 수신문서를 내부적으로 분배하게 된다.
- [1421] 전자주소의 물리적 주소 및 보안정보 획득의 프로세스는 도 6과 같다. 전자문서 유통에서 공인전자주소를 기반으로 수신자에게 문서를 전송하기 위해서는, ① 유통 클라이언트 APP가 수신 상대방의 주소정보를 입력하는 시점에 주소

디렉터리 서버에 연계하여 필요정보를 획득한 후 검색된 실 물리주소정보를 바탕으로 유통 메시징서버에 전송요청을 하는 방법과, ② 유통 클라이언트 APP가 수신자에 대한 공인전자주소를 기반으로 유통 메시징서버에 전송요청을 하고, 유통 메시징서버가 전송 전에 주소 디렉터리 서버에 물리적 주소 및 보안정보를 획득한 후, 수신자에게 문서를 전송하는 방법이 있다. 이와 같은 두 방법에 대한 절차는 도 7에 도시한 바와 같다.

[1422] 주소 디렉터리 서버는 유통 메시징서버 시스템이 주소 정보를 검색하거나 주소 발급을 대행할 수 있도록 원격서비스를 제공한다. 주소 디렉터리 서버가 제공하는 서비스는 주소검색 서비스, 주소 등록 서비스, 주소 변경 서비스가 있으며, 유통 프로토콜 규격을 기반으로 다음과 같은 서비스 인터페이스를 제공한다.

[1423] 주소 검색 서비스는 주소 디렉터리 서버가 공인 전자주소에 해당하는 물리적 주소 정보(예 : IP 주소, Domain 주소)와 공개키 정보를 검색 요청자에게 되돌려주는 서비스로서, 일반적으로 송신자가 문서를 전송하기 전에 수신자의 실제 주소 정보와 암호화를 위한 보안 정보를 획득하기 위해 사용한다. 이때, 요청 메시지와 응답 메시지의 역할은 아래 표 185와 같다.

[1424] 표 185

| 구성요소   | 역할                                                                        |
|--------|---------------------------------------------------------------------------|
| 요청 메시지 | 수신자가 개인인지 기업/기관인지 여부, 수신자의 고유 ID(주민등록번호, 사업자 등록번호, 외국인 등록번호 등), 공개키 요청 여부 |
| 응답 메시지 | 수신자 고유 ID, 수신자 주소, 공개키(요청여부에 따라 선택적)                                      |

[1425] 주소 등록 서비스는 주소 디렉터리 서버가 제공하는 UI를 통해서 뿐만 아니라 원격에서도 사용자의 공인 전자주소를 등록할 수 있도록 제공하는 서비스로서, 사용자 정보 및 공인 전자주소 정보를 요청 메시지로 받아서 주소 디렉터리 서버가 등록 처리한 후에 이에 대한 결과를 응답 메시지로 수신받는다. 주소 등록 서비스에 대한 요청 메시지는 반드시 요청자에 대한 전자서명 정보가 포함되어 전달되어야 하며, 주소 디렉터리 서버는 요청 메시지에 포함된 사용자 정보와 전자서명에 사용된 인증서정보가 동일한지를 검증하여야 한다. 이때, 요청 메시지와 응답 메시지의 역할은 아래 표 186과 같다.

[1426] 표 186

| 구성요소   | 역할                                                                                                       |
|--------|----------------------------------------------------------------------------------------------------------|
| 요청 메시지 | 사용자 고유 id, 사용자 부가정보 구조체(사용자 명, 연락처 정보 등), 사용자가 개인인지 기업/기관인지 여부, 사용자 주소정보, 공개키 정보(선택적), 주소정보 공개범위 정의한 구조체 |
| 응답 메시지 | 사용자 고유 id, 등록결과 값(성공, 실패-요청정보 오류, 실패-기 등록, 실패-인증된 송수신 개체가 아님, 주소 디렉터리 서버 오류 등)                           |

[1427] 주소 변경 서비스는 등록된 사용자에게 대한 주소정보를 원격에서 직접 사용자가 변경할 수 있도록 하는 기능을 제공하는 원격 서비스로 변경하여야 할 정보를 포함하여 주소 디렉터리 서버에 변경요청 메시지를 전송하고 이에 대한 결과를 응답메시지로 수신받는다. 주소 변경서비스에 대한 요청 메시지는 반드시 요청자에 대한 전자서명정보가 포함되어 전달되어야 하며, 주소 디렉터리 서버는 요청 메시지에 포함된 사용자 정보와 전자서명에 사용된 인증서 정보가 동일한지를 검증하여야 한다. 이때, 요청 메시지의 응답 메시지의 역할은 아래 표 187과 같다.

[1428] 표 187

| 구성요소   | 역할                                                                                                               |
|--------|------------------------------------------------------------------------------------------------------------------|
| 요청 메시지 | 사용자 고유 id, 사용자 부가정보 구조체(사용자 명, 연락처 정보 등)중 변경된 정보만, 사용자가 개인인지 기업/기관인지 여부, 사용자 주소정보(공개키 정보(선택적), 주소정보 공개범위 정의한 구조체 |
| 응답 메시지 | 사용자 고유 id, 정보변경결과 값(성공, 실패-요청정보 오류, 실패-등록되지 않은 사용자, 실패-변경권한 없음, 주소 디렉터리 서버 오류 등)                                 |

## 청구범위

### [청구항 1]

전자문서를 유통하는 시스템에 있어서,  
전자주소를 기반으로 메시지를 송수신하고 메시지 송수신에 대한  
유통증명서를 발급 및 관리하는 유통 메시징서버를 통해  
전자문서를 유통하는 송수신 개체;  
상기 송수신 개체의 전자주소를 등록/관리하며, 상기 송수신 개체  
간의 전자문서 유통 경로를 설정하고, 상기 송수신 개체에게  
전자문서의 표준서식을 제공하며, 송수신 개체 간의 전자문서  
유통 과정에서 오류가 발생하였을 시에 메시지 전송을 대행하고  
유통증명서를 발급하는 유통 허브; 및  
유통증명서를 전달받아 보관하며, 신뢰할 수 있는 제3자 보관기관;  
을 포함하는 것을 특징으로 하는 전자문서 유통 시스템.

### [청구항 2]

제 1 항에 있어서, 상기 송수신 개체의 유통 메시징서버는,  
송수신한 메시지는 사용자별로 상태 정보를 포함하여 메시지함에  
보관하고,  
메시지 송수신 이력을 편집 및 삭제가 불가능한 매체에 소정 기간  
동안 보관하며,  
메시지 송수신에 대한 유통증명서를 발급하여 상기 제3자  
보관기관에 보관을 의뢰하고,  
상기 유통허브의 주소 디렉터리 서버와의 연계를 통해 상기  
송수신 개체에게 전자주소 등록 및 검색, 수정, 삭제를 포함하는  
기능을 사용할 수 있도록 하며,  
소정 기간 이상 보관된 메시지들을 외부 저장장치에 이관하여  
보관하는 것을 특징으로 하는 전자문서 유통 시스템.

### [청구항 3]

제 1 항에 있어서, 상기 전자주소는,  
상기 송수신 개체가 상기 유통허브의 주소 디렉터리 서버를 통해  
발급받은 이용자 식별기호;  
상기 송수신 개체가 필요한 경우에 자체적으로 부여하는 고유한  
값이며, 해당 송수신 개체 내에서 고유한 값인 추가 식별기호;  
상기 이용자 식별기호와 추가 식별기호 사이에 위치하는 구분  
기호;  
를 포함하는 것을 특징으로 하는 전자문서 유통 시스템.

### [청구항 4]

제 3 항에 있어서, 전자문서 유통에 대한 주체는 고유등록주소를  
발급받은 사용자인 것을 특징으로 하는 전자문서 유통 시스템.

### [청구항 5]

제 3 항에 있어서, 상기 구분 기호는 '#'인 것을 특징으로 하는  
전자문서 유통 시스템.

### [청구항 6]

제 1 항에 있어서, 상기 유통허브는 전자문서 서식 등록기를

- 구비하며,  
 상기 전자문서 서식 등록기는 전자문서 표준서식의 등록, 삭제 및 정보 수정을 포함하는 관리를 수행하며, 전자문서 표준서식을 문맥(context)에 따라 추가로 분류하고, 전자문서 표준서식이 사용될 수 있는 문맥(context)에 대한 등록, 수정을 포함하는 관리를 수행하는 것을 특징으로 하는 전자문서 유통 시스템.
- [청구항 7] 제 6 항에 있어서, 상기 전자문서 서식 등록기는, 문서 양식을 관리하는 서버 엔진; 및 송수신 개체를 사용하는 사용자가 문서 양식을 검색하고 다운로드 받아서 사용할 수 있도록 하는 표준인터페이스;를 포함하며,  
 상기 송수신 개체는 송수신 개체를 사용하는 사용자가 유통 메시징서버를 통해 메시지를 송수신할 수 있도록 하는 사용자 인터페이스인 유통 클라이언트 어플리케이션을 추가로 구비하며,  
 상기 유통 클라이언트 어플리케이션을 사용하는 사용자는 상기 전자문서 서식 등록기의 표준인터페이스를 통해 문서 서식을 검색하고 다운로드 받은 후에, 해당 문서 서식을 이용하여 전자문서를 생성하는 것을 특징으로 하는 전자문서 유통 시스템.
- [청구항 8] 제 1 항에 있어서, 상기 유통허브는 송수신 개체 간의 전자문서 유통 과정에서 오류가 발생하였을 시에 메시지 전송을 대행하고 유통증명서를 발급하는 유통 중계서버를 구비하며,  
 상기 유통 중계서버는 송수신 개체로부터 메시지 전송을 의뢰받으면 메시지 전송을 대행한 후에 메시지 전송을 의뢰한 송수신 개체에게 송신증명서를 발급하며, 의뢰받은 메시지 전송을 실패하였을 시에는 메시지 전송을 의뢰한 송수신 개체에게 오류 메시지를 전송하는 것을 특징으로 하는 전자문서 유통 시스템.
- [청구항 9] 제 1 항에 있어서, 상기 유통허브는 외부 시스템과의 연계를 위한 외부 연계 게이트웨이 서버를 구비하며,  
 상기 외부 연계 게이트웨이 서버는 전자주소를 기반으로 메시지를 송수신하는 유통 메시징서버를 구비하며, 연계된 외부 시스템과 전자문서 유통 시스템 간의 송수신 전자주소의 검증/변환 기능과, 연계된 외부 시스템과 전자문서 유통 시스템 간의 메시지 검증/변환 기능, 연계된 외부 시스템과 전자문서 유통 시스템 간의 전자문서에 적용된 보안의 검증/변환 기능, 연계된 외부 시스템과 전자문서 유통 시스템 간의 전자문서의 적합성을 검증하고 상호간 변환하는 기능을 제공하는 것을 특징으로 하는 전자문서 유통 시스템.
- [청구항 10] 제 1 항에 있어서, 전자주소 등록대행기관이 주소 디렉터리서버에 전자주소를 등록 요청하여 응답을 받는데 이용되는

제1인터페이스; 전자주소 등록대행기관이 주소 디렉터리서버에 등록된 전자주소 정보에 대한 변경을 요청하여 응답을 받는 데 이용되는 제2인터페이스; 및 전자주소 등록대행기관이 주소 디렉터리서버에 등록된 전자주소 정보의 삭제를 요청하여 응답을 받는 제3인터페이스;가 구비되며,

상기 전자주소 등록대행기관은 제1인터페이스를 통해 전자주소 신청자 정보 및 전자주소 정보를 요청메시지에 포함하여 전송한 후에 주소 디렉터리서버의 등록 처리 결과를 응답메시지로 수신받으며, 상기 전자주소 등록대행기관은 제2인터페이스를 통해 변경하고자 하는 사용자 정보 및 전자주소 정보를

요청메시지에 포함하여 전송한 후에 주소 디렉터리서버의 변경 처리 결과를 응답메시지로 수신받고, 상기 전자주소 등록대행기관은 제3인터페이스를 통해 삭제하고자 하는 사용자 정보 및 전자주소 정보를 요청메시지에 포함하여 전송한 후에 주소 디렉터리서버의 삭제 처리 결과를 응답 메시지로 수신받는 것을 특징으로 하는 전자문서 유통 시스템.

#### [청구항 11]

제 10 항에 있어서, 상기 전자주소 등록대행기관 또는 송수신 개체가 주소 디렉터리서버에 전자문서 수신자의 전자주소에 해당하는 물리적 주소 정보와 메시지 보안처리를 위한 공인인증서 정보를 요청하여 응답을 받는 데 이용되는 제4인터페이스가 구비되며,

전자주소 등록대행기관 또는 송수신개체의 유통 메시지정서버는 전자문서 수신자의 전자주소 및 공인인증서 요청 여부를 요청메시지에 포함하여 전송한 후에 주소 디렉터리서버로부터 전자문서 수신자의 물리적 주소 정보 및 공인인증서 정보를 응답메시지로 수신받는 것을 특징으로 하는 전자문서 유통 시스템.

#### [청구항 12]

제 1 항에 있어서, 송수신 개체의 유통 메시지정서버 또는 전자주소 등록대행기관의 유통 메시지정서버는 메시지 전송, 유통증명서 전달, 유통증명서 보관요청 및 제3자 보관기관 보관결과 전달에 이용하는 제5인터페이스를 구비하는 것을 특징으로 하는 전자문서 유통 시스템.

#### [청구항 13]

제 1 항에 있어서, 송수신 개체 내의 사용자는 사용자 인터페이스인 유통 클라이언트 어플리케이션을 구비하고, 송수신 개체의 유통 메시지정서버는 전자문서 유통을 요청하는 사용자를 위한 유통 클라이언트 어플리케이션과 연계하여 사용자에게 문서 송수신 기능을 제공하는 제6인터페이스를 구비하며,

상기 제6인터페이스는 메시지 전송요청, 메시지 목록 요청, 메시지 상세정보 요청, 스팸 메시지 신고 및 물리적 주소정보 검색 기능을 유통 클라이언트 사용자에게 제공하는 것을 특징으로 하는 전자문서 유통 시스템.

[청구항 14]

송수신 개체와 유통허브를 포함하는 전자문서 유통 시스템에서 전자문서를 유통하는 방법에 있어서,

- (a) 송신 개체는 수신 개체의 주소 정보에 대응되는 물리적 주소 정보를 유통허브를 통해 획득한 후에, 전자문서를 첨부한 메시지를 상기 물리적 주소로 전송하는 단계;
- (b) 메시지를 수신한 수신 개체는 수신 메시지 및 송신 개체에 대한 적합성 검증 결과에 따라 수신증명서 또는 오류증명서를 발급하여 송신 개체에게 전달하는 단계; 및
- (c) 수신 개체에게 메시지를 전송하였으나 실패한 송신 개체는 유통허브에 메시지 전송 대행을 의뢰하며, 메시지 전송 대행 의뢰를 받은 유통허브는 송신증명서를 발급하여 송신 개체에게 전달하고 수신 개체에게 메시지를 전송한 후에 상기 (b)단계를 수행하는 단계;

를 포함하는 전자문서 유통 방법.

[청구항 15]

제 14 항에 있어서, 상기 (a)단계는,

- (a1) 송신 개체는 수신 개체의 전자주소 정보를 기준으로 수신 개체에 대한 물리적 주소 정보 및 보안 정보를 유통허브의 주소 디렉터리 서버에 질의하는 단계;
- (a2) 상기 주소 디렉터리 서버는 송신 개체의 질의를 수신하여 검증한 후에 전자주소가 화이트리스트에 있는 경우에 전자주소에 대응되는 물리적 주소 정보를 송신 개체에게 제공하는 단계; 및
- (a3) 상기 송신 개체는 전자문서를 첨부한 메시지를 상기 주소 디렉터리 서버로부터 제공받은 물리적 주소 정보를 기준으로 경로 설정을 하여 수신 개체에게 전송하는 단계;

를 포함하는 것을 특징으로 하는 전자문서 유통 방법.

[청구항 16]

제 14 항에 있어서, 상기 (b)단계 이후에, 수신 증명서를 수신한 송신 개체는 수신 증명서의 적합성을 검증하고 검증된 정보를 수신증명서에 첨부한 후에 수신증명서를 자체 보관함과 동시에 제3자 보관기관에 보관 의뢰하는 것을 특징으로 하는 전자문서 유통 방법.

[청구항 17]

제 14 항에 있어서, 상기 (c)단계는,

- (c1) 수신 개체에게 메시지를 전송하였으나 실패한 송신 개체는 유통허브에 메시지 전송 대행을 의뢰하는 단계;
- (c2) 유통허브는 메시지 전송을 시작하되, 전송 실패 시에는 소정

시간 간격으로 재시도하며, 메시지 전송에 최종 실패한 경우에는 송신 개체에게 전송실패 메시지를 전달하는 단계;

(c3) 메시지를 정상적으로 수신한 수신 개체는 수신증명서를 발급하여 유통허브에 전송하는 단계; 및

(c4) 전자문서의 수신자가 전자문서를 열람한 경우에 수신 개체는 열람증명서를 발급하여 유통허브를 거치지 않고 송신 개체에 직접 전송하는 단계;

송신자 또는 수신자의 역할을 하는 송수신 개체와 전자문서 유통허브를 포함하는 전자문서 유통 시스템에서 전자문서를 유통하는 방법에 있어서,

(a) 송신자는 수신자의 전자주소 정보를 획득하는 단계;

(b) 송신자는 송신할 문서를 미리 정해진 메시지 구조체로 패키징한 메시지를 생성한 후에 수신자의 전자주소로 전송하는 단계;

(c) 상기 (b)단계에서 전송 실패한 경우에, 송신자는 전자문서 유통허브에 메시지 전송을 의뢰하며, 메시지 전송 의뢰를 받은 전자문서 유통허브는 송신증명서를 발급하여 송신자에게 전달한 후에 메시지 전송을 대행하는 단계;

(d) 수신자는 송신자 또는 전자문서 유통허브로부터 수신한 메시지를 검증하여 검증이 통과되면 수신한 메시지에서 문서를 추출하고 수신증명서를 발급하여 송신자에게 전달하는 단계;

(e) 송신자는 전달받은 수신증명서를 신뢰할 수 있는 제3자 보관기관을 활용하여 보관하는 단계; 및

(f) 수신자는 추출한 문서를 문서 담당자에게 전달하는 단계; 를 포함하는 전자문서 유통 방법.

#### [청구항 19]

제 18 항에 있어서, 상기 (a)단계 전에는,

(g) 송신자 및 수신자는 문서 유통을 위한 유통 메시지정서버를 구축할지, 3자 유통이 가능한 유통 메시지정서버를 보유한 송수신 개체를 이용할지에 대하여 결정하는 단계;

(h) 상기 (g)단계에서 문서 유통을 위한 유통 메시지정서버를 구축할 것이라고 결정한 경우에는, 송신자 및 수신자는 문서 유통을 위한 유통 메시지정서버를 구축한 후에, 인증 기관을 통해 유통 메시지정서버의 인증테스트를 수행하고, 전자문서 유통허브의 주소 디렉터리 서버에 접속한 후 송수신 개체로서의 전자주소로 발급받은 후에, 내부의 실 사용자를 위해 자체적으로 내부 구분자를 등록하고 관리하는 단계;

(i) 상기 (g) 단계에서 3자 유통이 가능한 유통 메시지정서버를 보유한 송수신 개체를 이용할 것이라고 결정한 경우에는, 송신자

및 수신자는 3차 유통이 가능한 유통메시징서버를 통해 전자주소 개설을 요청한 후에, 전자주소 정보를 전자문서 유통허브의 주소 디렉터리 서버에 등록하는 단계; 및

(j) 상기 (h)단계 또는 (i)단계 후에 상기 송신자는 전자문서 유통허브의 전자문서 서식 등록기에서 문서 서식을 검색하고 다운로드 받아서 등록하는 단계; 를 포함하며,

상기 (b)단계 전에는, 송신할 문서를 선택하거나 또는 상기

(i)단계에서 등록한 문서 서식을 이용하여 송신할 문서를 작성하는 하는 (k)단계가 추가로 수행되는 것을 특징으로 하는 전자문서 유통 방법.

[청구항 20]

컴퓨터로 관독 가능한 기록매체에 있어서, 제 14 항 내지 제 19 항 중 어느 한 항에 따른 방법을 구현하는 프로그램이 저장되는 기록매체.