

공인전자주소 기반의 모바일용 전자문서유통 클라이언트 앱 개발

2013. 6. 5

목 차

1. 과제 개요
2. 개발 내용
3. 추진 계획



I . 과제 개요

1. 과제개요
2. 과제목표
3. 과제범위
4. 추진 전략
5. 응용분야 및 활용
6. 기대효과

1. 과제 개요

과제명

공인전자주소 기반의 모바일 용 전자문서 유통 클라이언트 앱 개발

수행기간

2013.5 ~ 2013.11 (7개월)

주관기관

정보통신산업진흥원

과제수행사

토피도 (주)

과제목적

#메일 이용 확산을 위해서

- 지식경제부가 2012년 6월 개정한 “전자문서 및 전자거래기본법”에 따라 시행되는 공인전자주소 기반의 전자문서유통체계에서 제공되는 사용자 용 어플리케이션의 일환으로 모바일 앱 개발
- Android 용 공인전자주소 기반 전자문서유통(#메일) 모바일 앱(native App) 개발
- NIPA의 “전자문서 유통 클라이언트 기술규격”을 100% 준수한 제품 개발

1

연계인터페이스 클라이언트 모듈 개발

- #메일 유통 메시징서버와 연계하기 위한 연계인터페이스(로그인, 로그아웃, 메시지 조회, 메시지 전송, 스팸신고 등 15개 연계 인터페이스 지원) 개발
- 적용 기술 : #메일 유통 메시징서버와 연계하기 HTTP/S 기반의 XML-RPC 통신 기술

2

안드로이드 앱 개발

- #메일 송.수신, #메일 열람, 메시지함 관리 등을 위한 모바일 사용자용 화면(UI) 개발(안드로이드 용)
- 적용 기술 : 안드로이드 제품 개발 기술

전략

1

모든 사업의 기본적인 진행은 주관기관인 토피도(주)에서 수행함

전략

2

공개SW 커뮤니티 중 “Sourceforge.net” 을 활용하여 개발된 소스를 공개하고 관리

전략

3

본 프로젝트 기간 중 모든 소스는 기본적으로 주관기업이 수정, 보완, 업그레이드를 책임짐

전략

4

프로젝트 종료 후에도 주관기관은 커뮤니티에 Committer로서 최소 6개월 이상 주도적으로 버그수정, 업그레이드를 수행함

5. 응용분야 및 활용

- 기본적으로 개발된 모바일 용 솔루션은 일반 사용자들이 중계자 또는 자체구축 유통메시징 서버에 연계하여 #메일 서비스를 받을 수 있는 앱으로 사용
- 주관사업자는 개발된 연계 클라이언트 라이브러리를 기반으로 중계사업자 별 고유 인터페이스를 보장하여 중계사업자에게 서비스하고 이에 대한 기술지원 및 유지보수를 상용화 계획

기술적 측면

- 2013년 1월 새롭게 제정된 “전자문서 유통 클라이언트 기술규격” 기반으로 처음 구축되는 모바일 native-app 유통클라이언트임
- “전자문서 유통 클라이언트 기술규격”을 기반으로 실제 앱 개발 시 발생할 수 있는 규격의 문제점이나 기능들을 도출하고 이를 규격 개정애 반영할 수 있는 기반이 됨

경제 산업적 측면

- 정부의 전자문서 확산 정책의 일환인 온라인 등기형 메일의 확산을 도모하게 됨
- 일반 개인사용자들이 #메일을 사용할 수 있는 기반을 제공할 뿐 아니라 공개 SW로 개발되면서 다양한 사용자들에 의해 고객의 요구를 수용할 수 있는 기회를 줌으로써 사용자가 제공받을 수 있는 서비스의 품질을 향상하게 됨
- #메일 연계 기능 개발을 위해 개발자들이 중복적으로 투자하게 되는 개발 노고를 줄여줌으로써 경제적인 효율성을 제공함

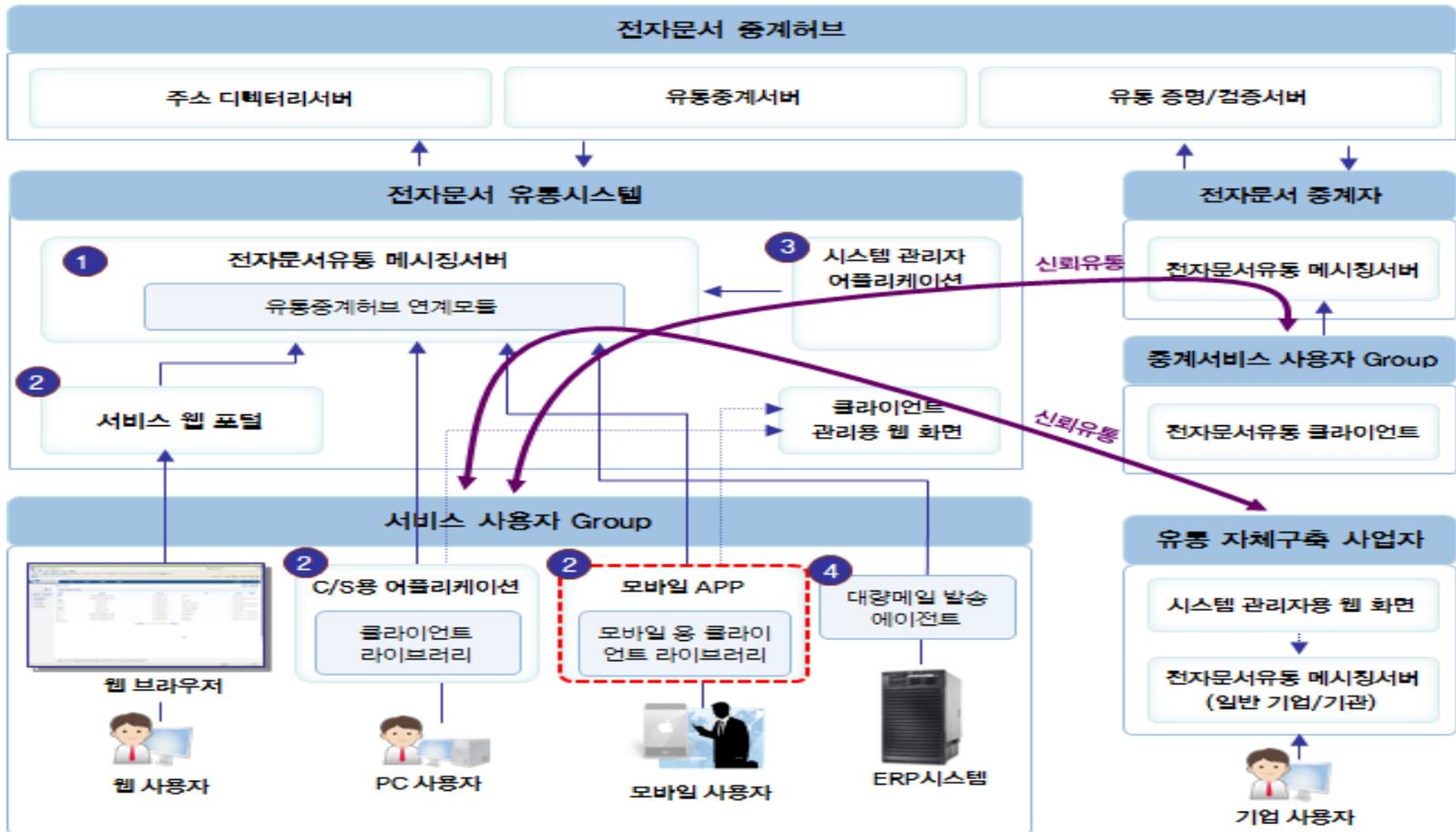


II. 개발 내용

- 1. 목표시스템 구성도**
- 2. 개발 방안**
- 3. 유통 메시징서버 연계방안**
- 4. 개발환경**

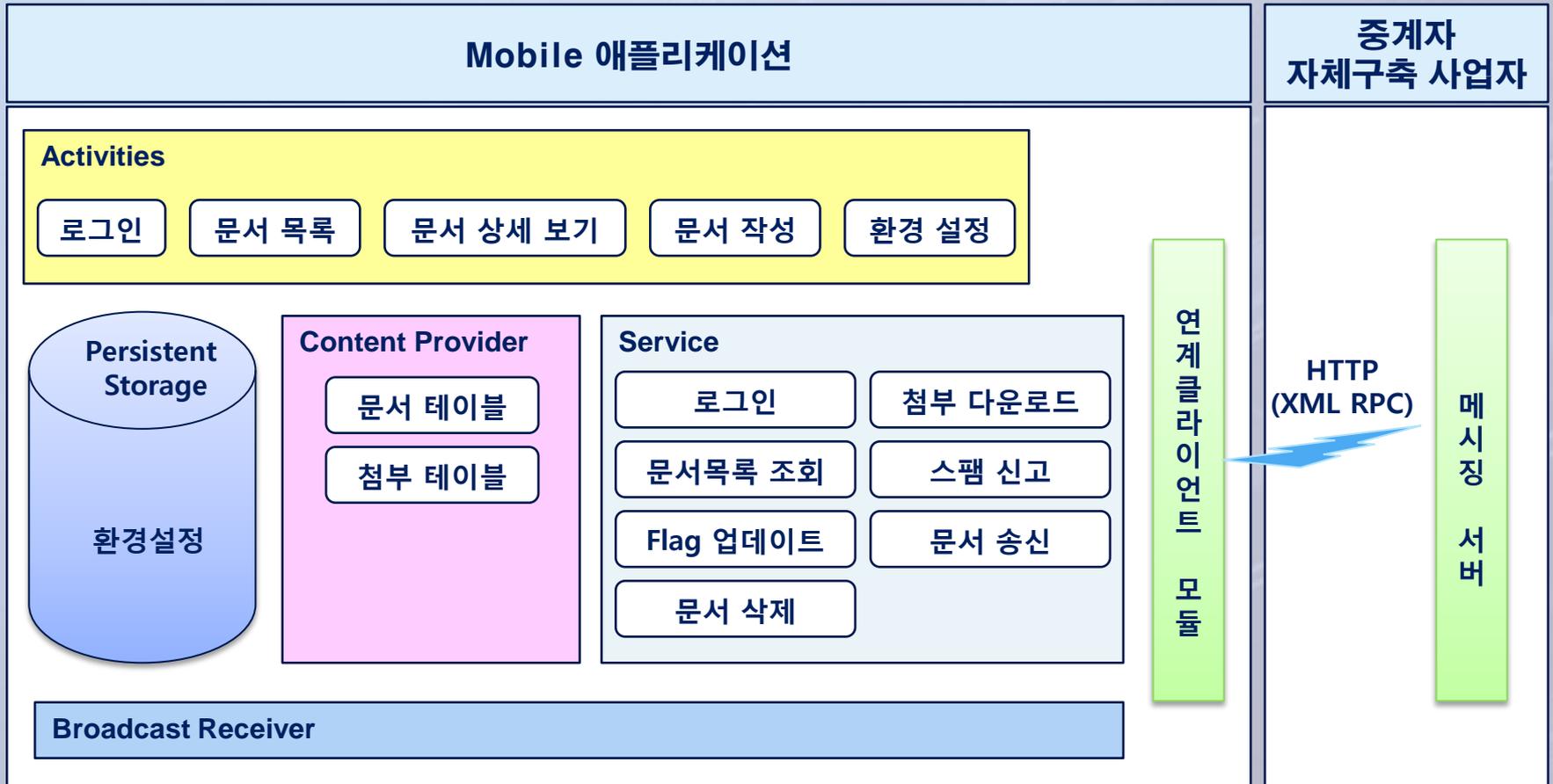
1. 목표시스템 구성도

❖ 시스템 개념도

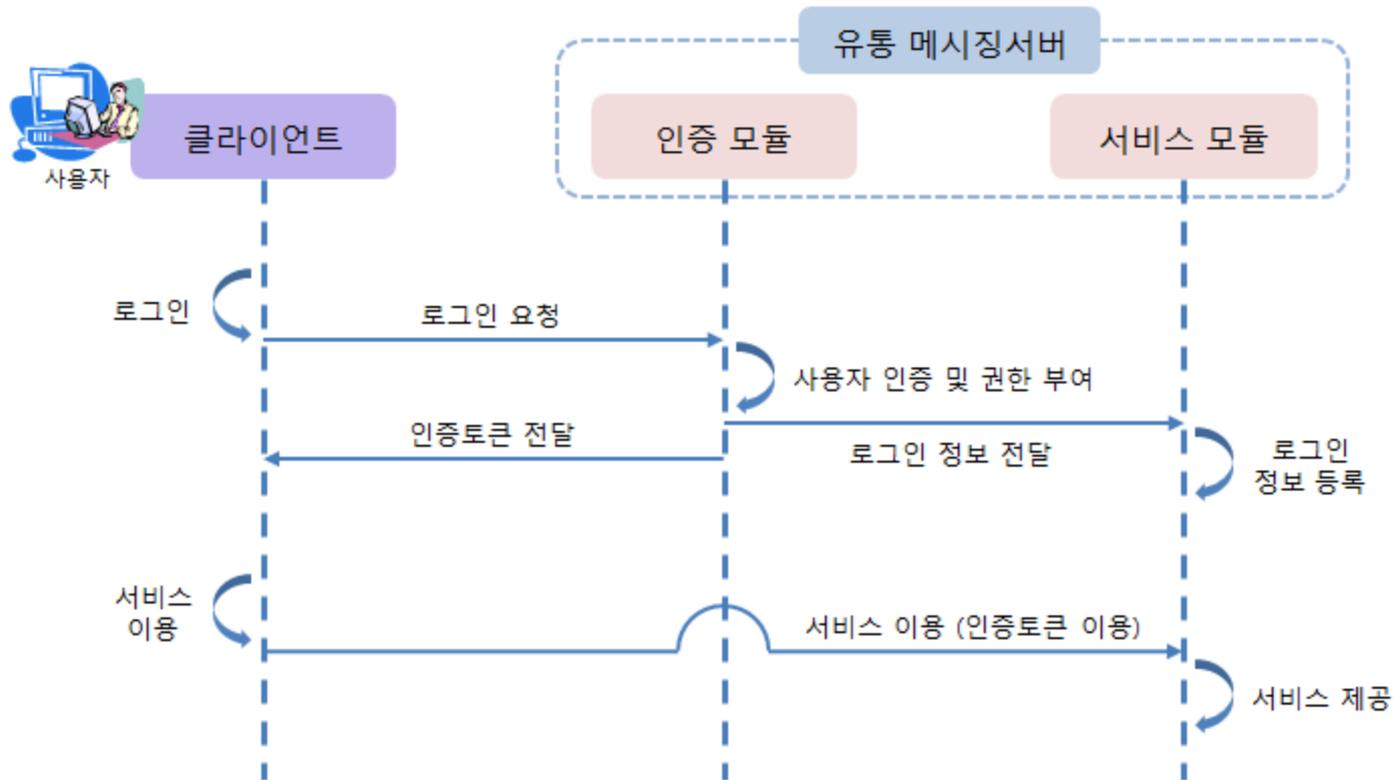


1. 목표시스템 구성도

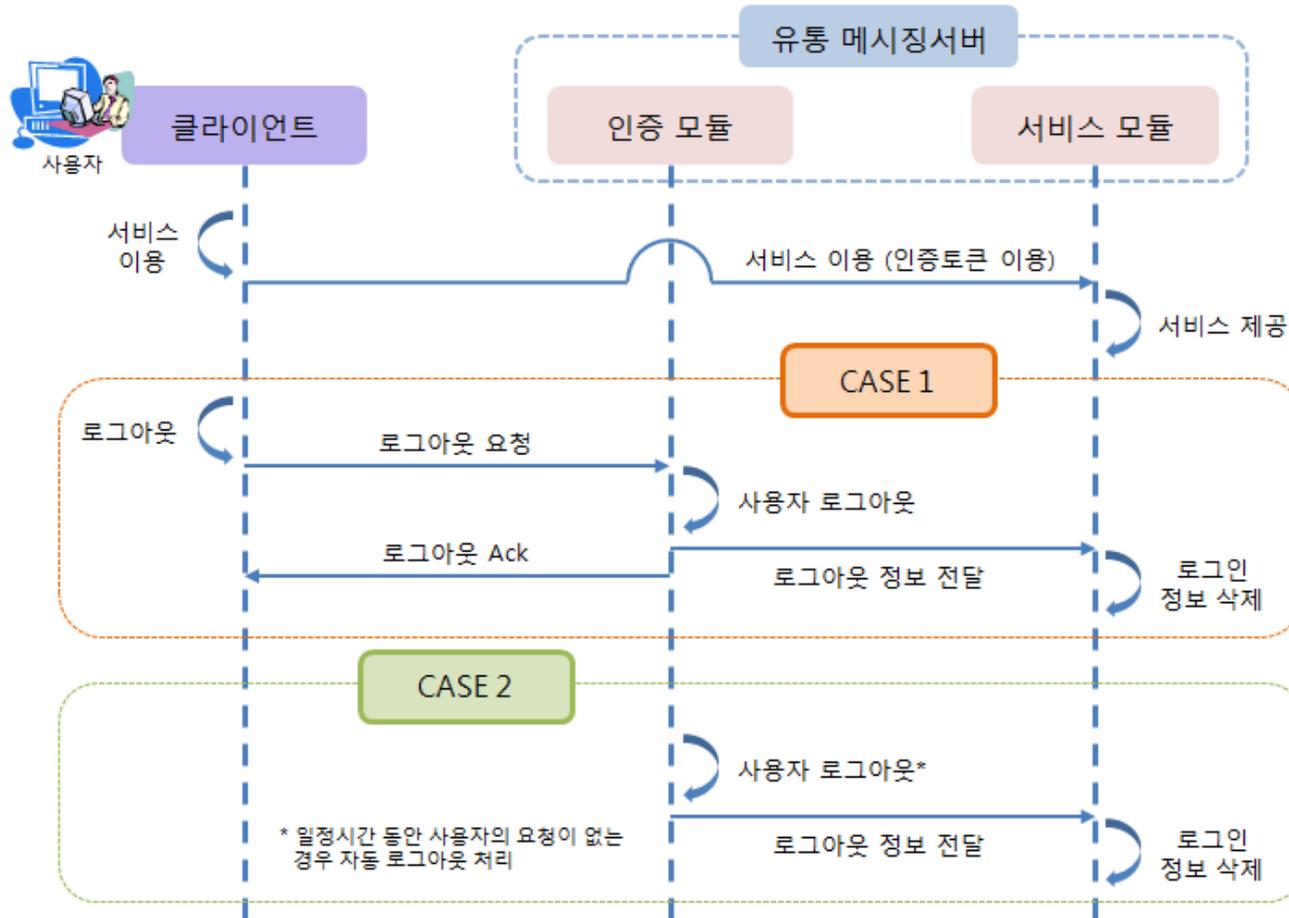
❖ 기능 개념도



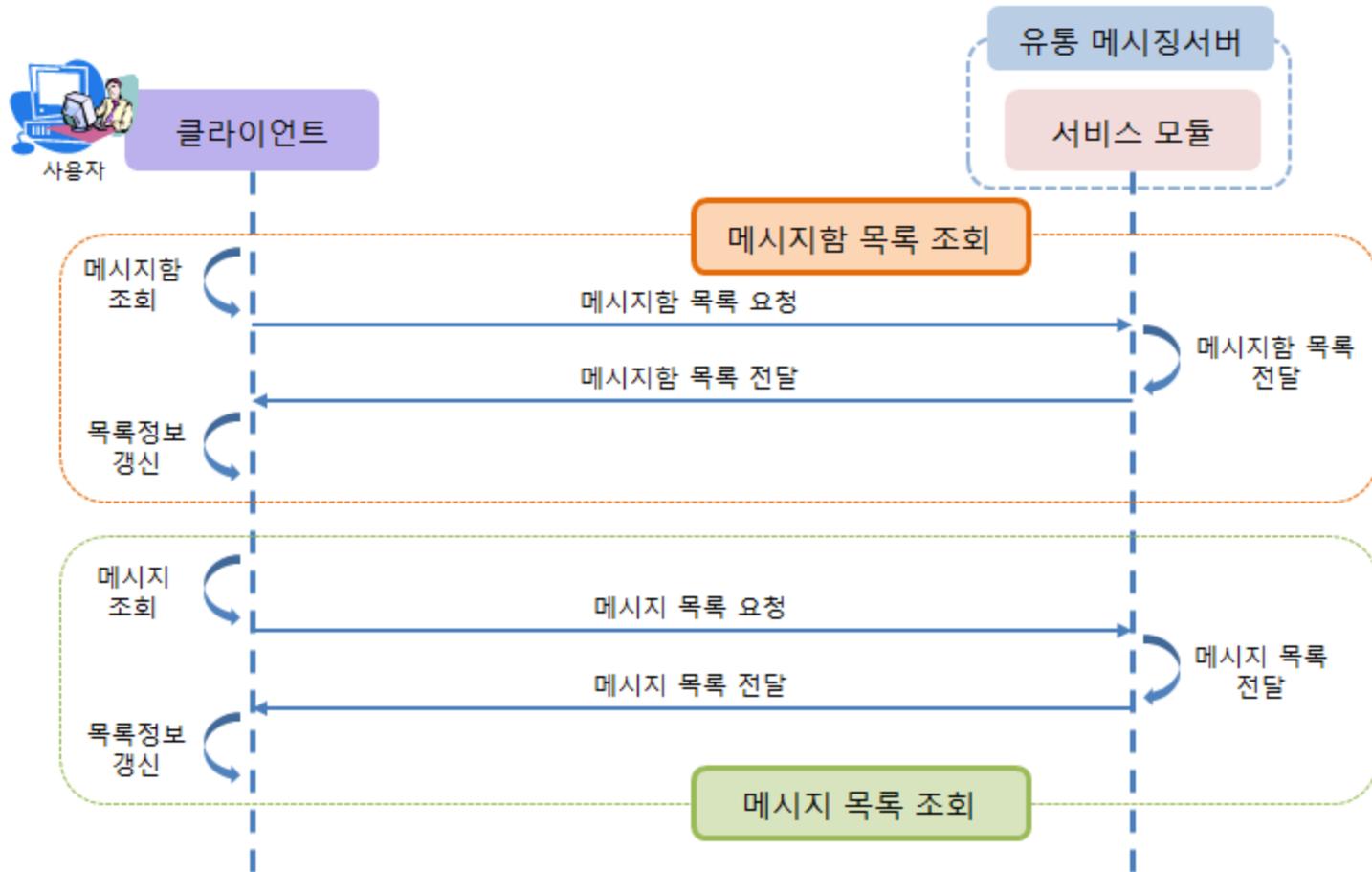
❖ 로그인 프로세스



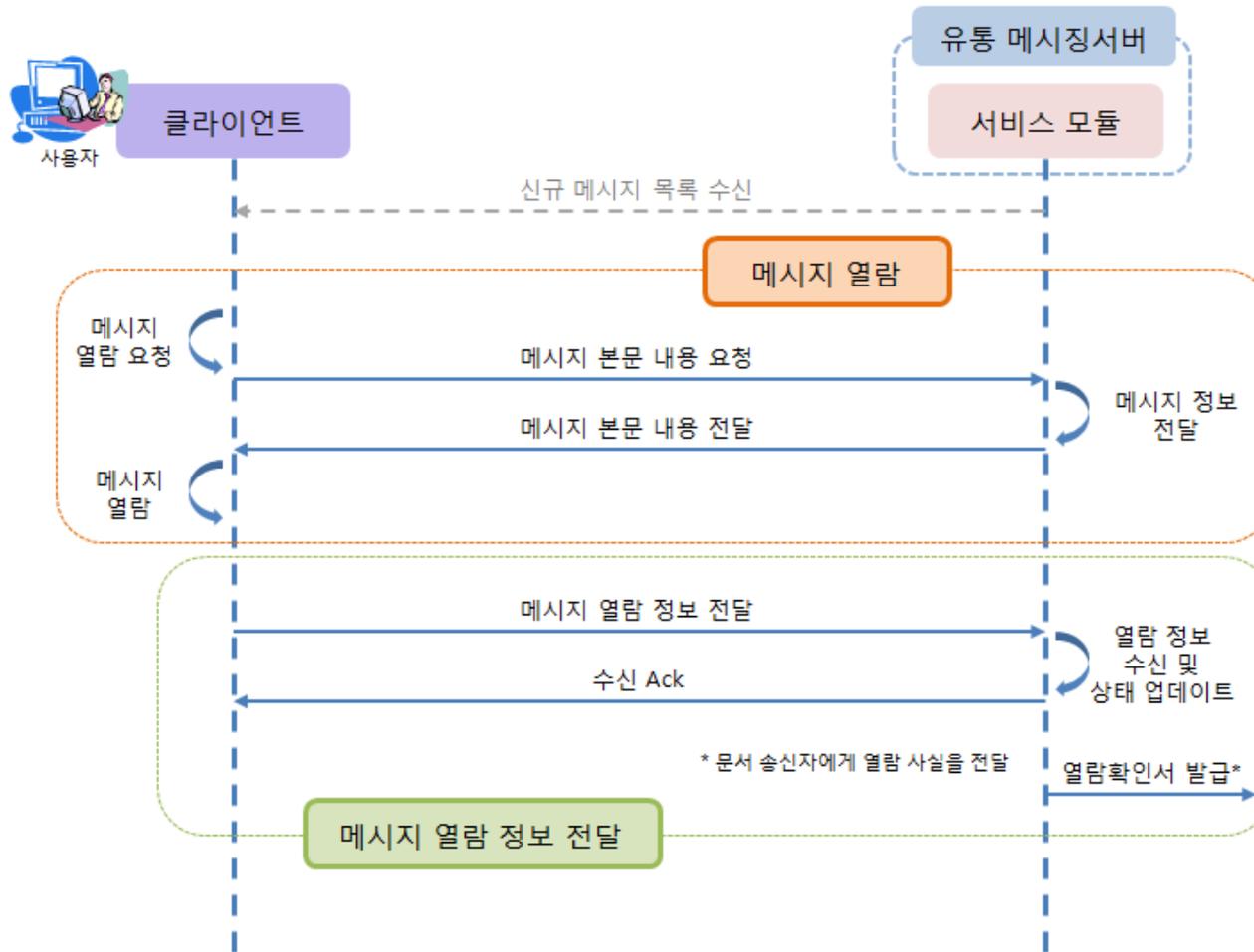
❖ 로그아웃 프로세스



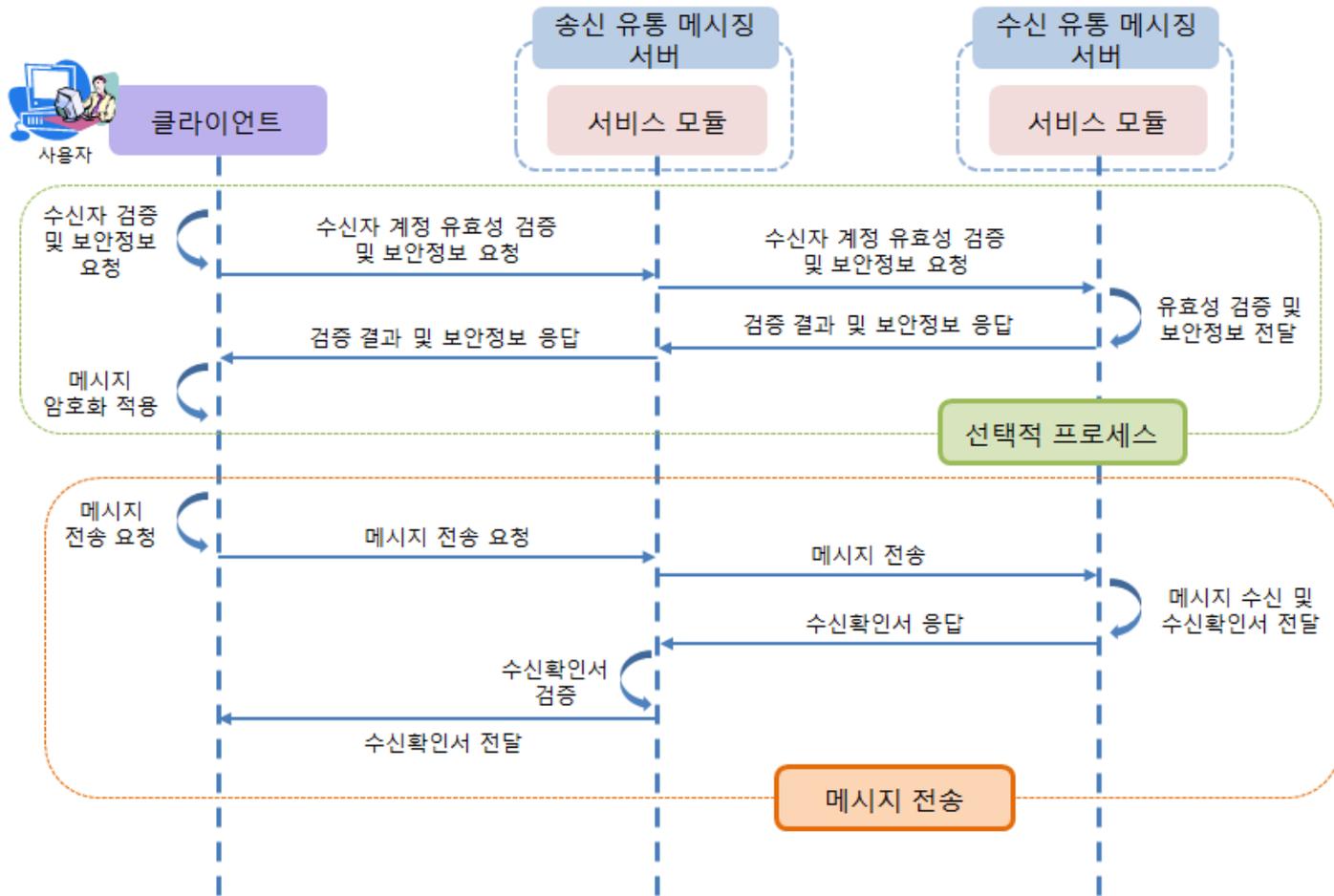
❖ 메시지함/메시지 목록 조회 프로세스



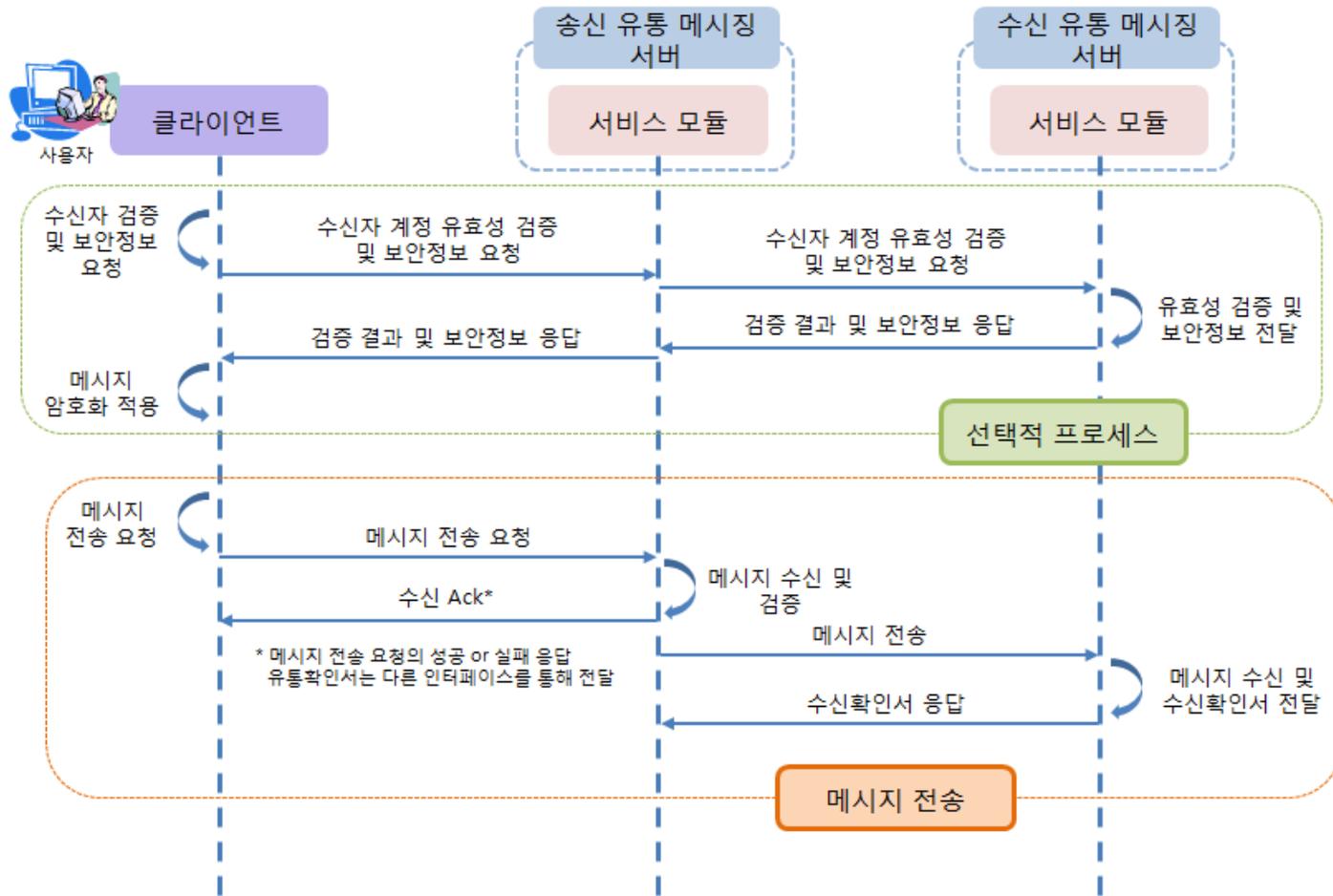
❖ 메시지 열람 프로세스



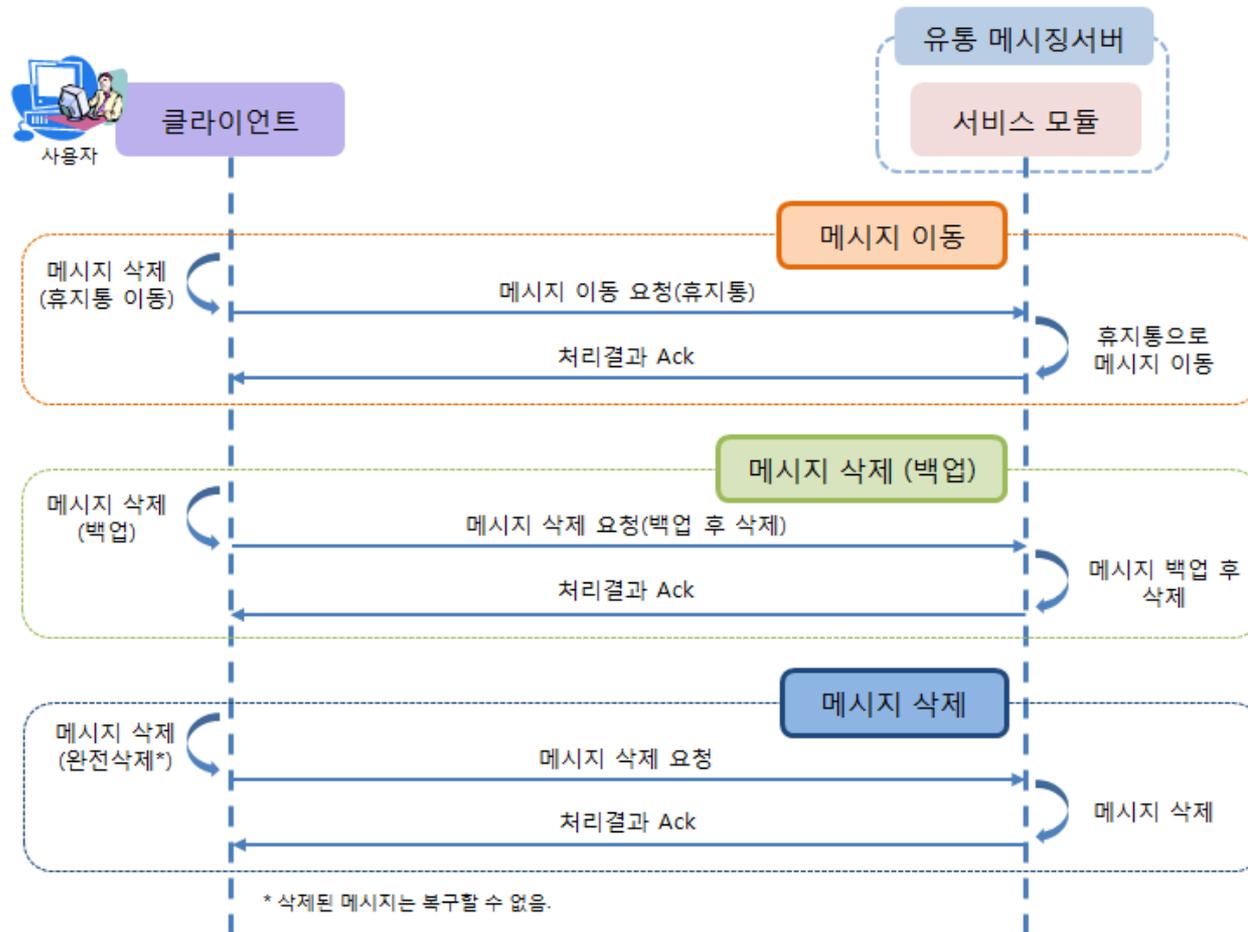
❖ 메시지 전송 프로세스 (동기식)



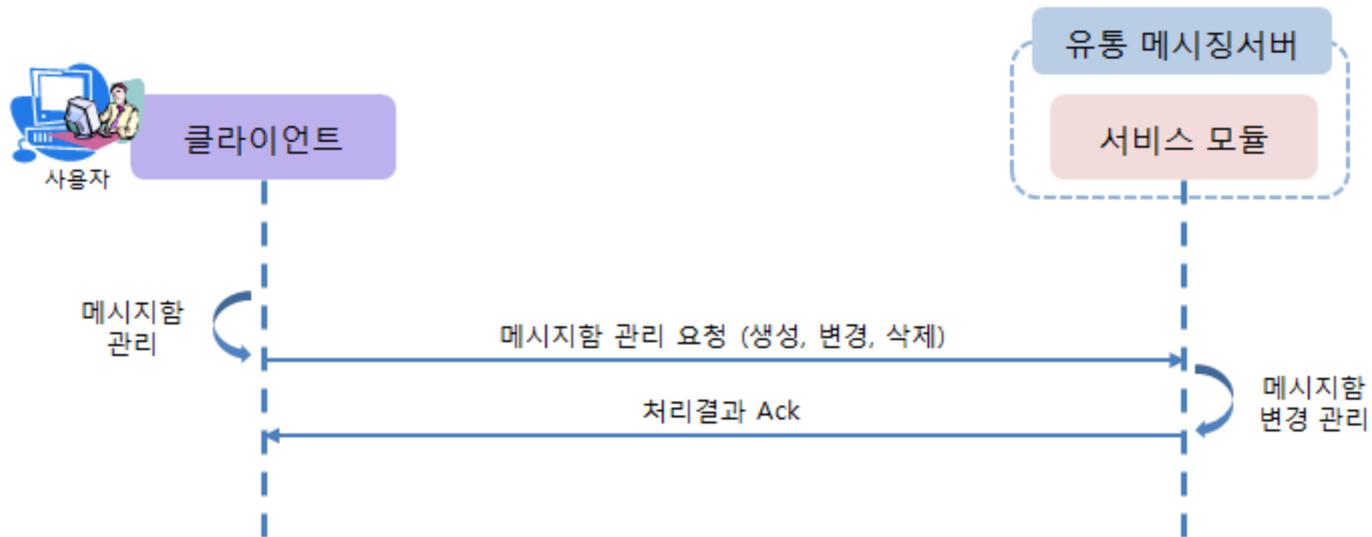
❖ 메시지 전송 프로세스 (비동기식)



❖ 메시지 삭제 프로세스



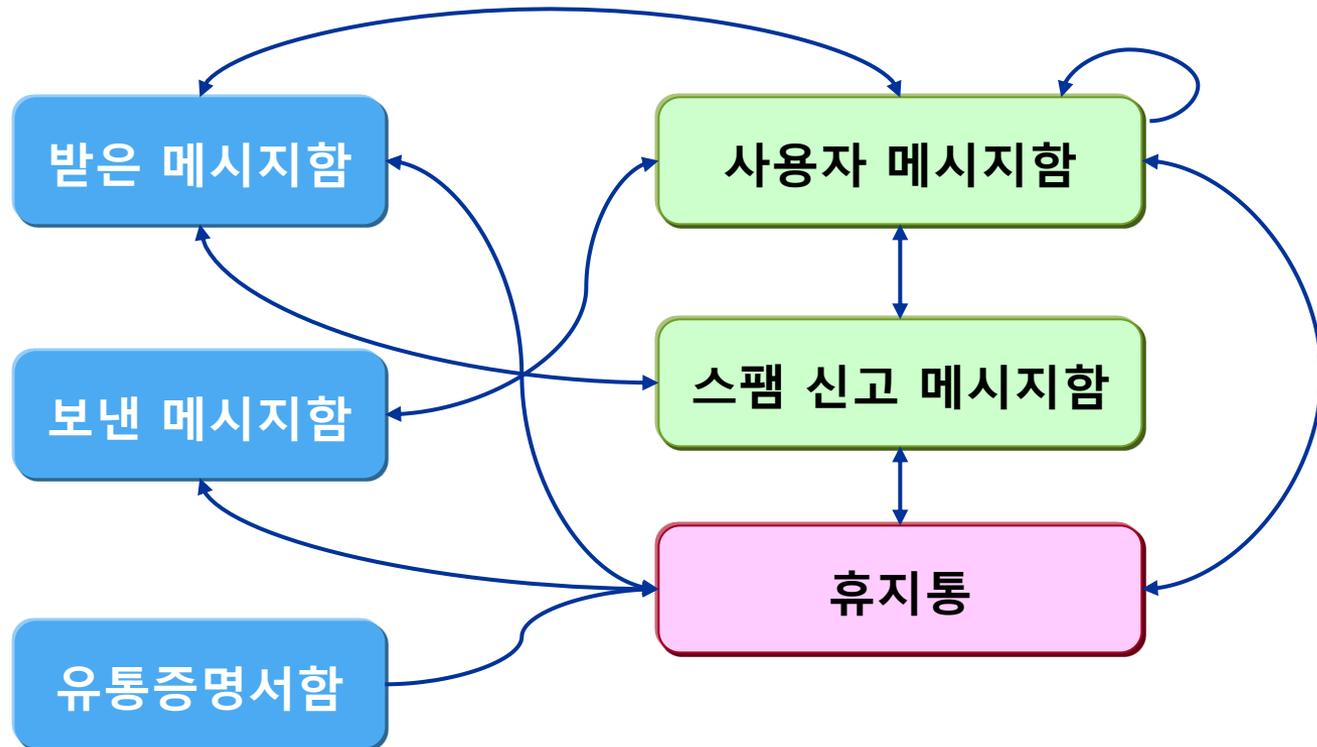
❖ 메시지함 관리 프로세스



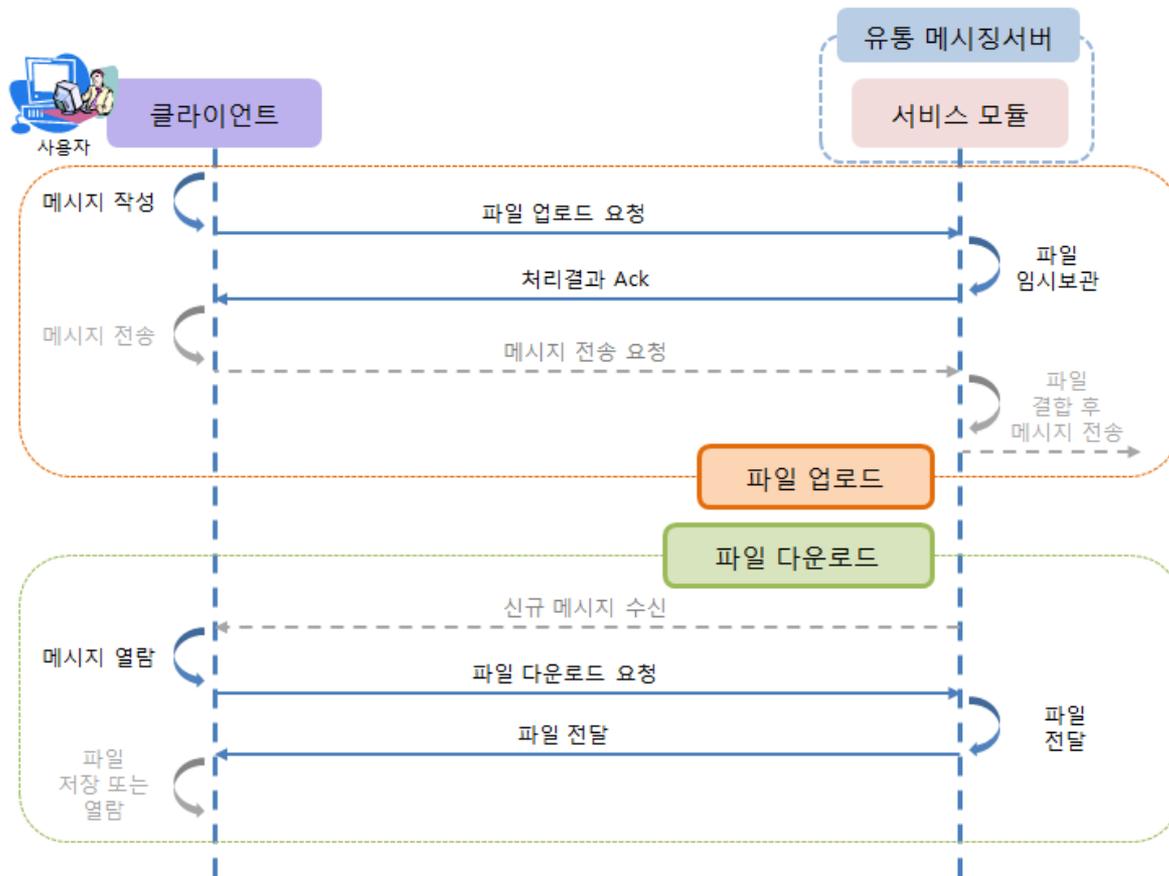
❖ 메시지 이동 프로세스



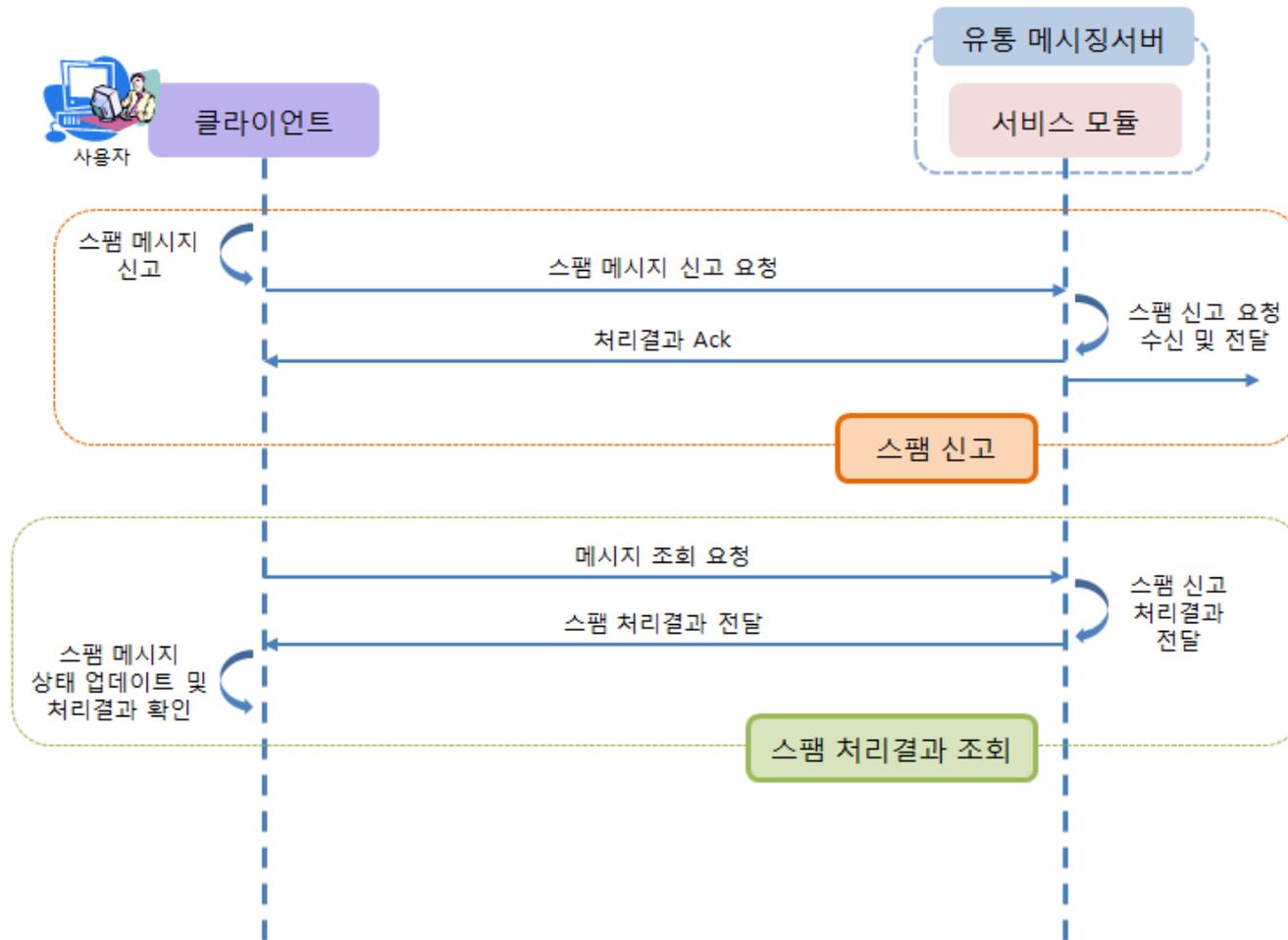
❖ 메시지 함 간 이동 법칙



❖ 첨부파일 업로드/다운로드 프로세스



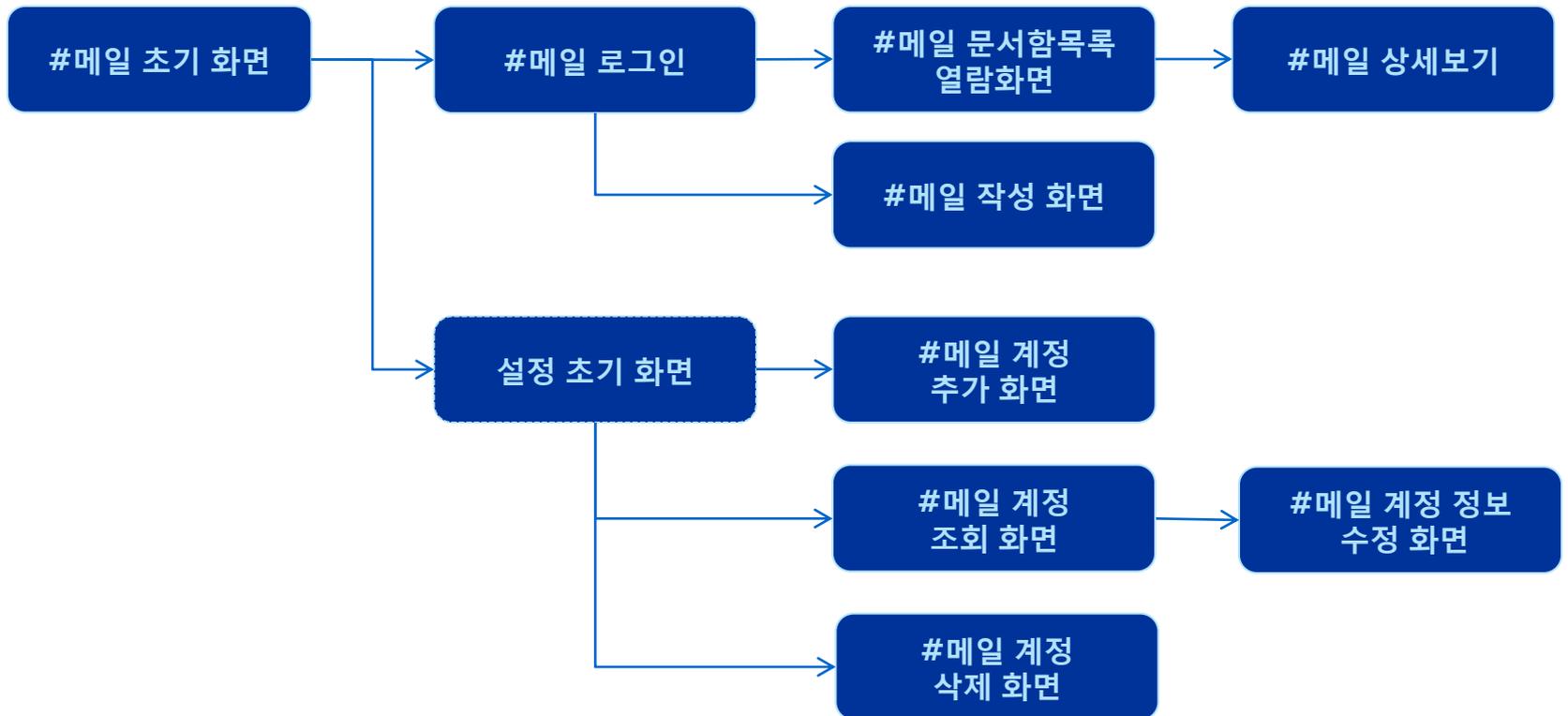
❖ 스팸신고 프로세스



❖ 유통증명서 발급 프로세스



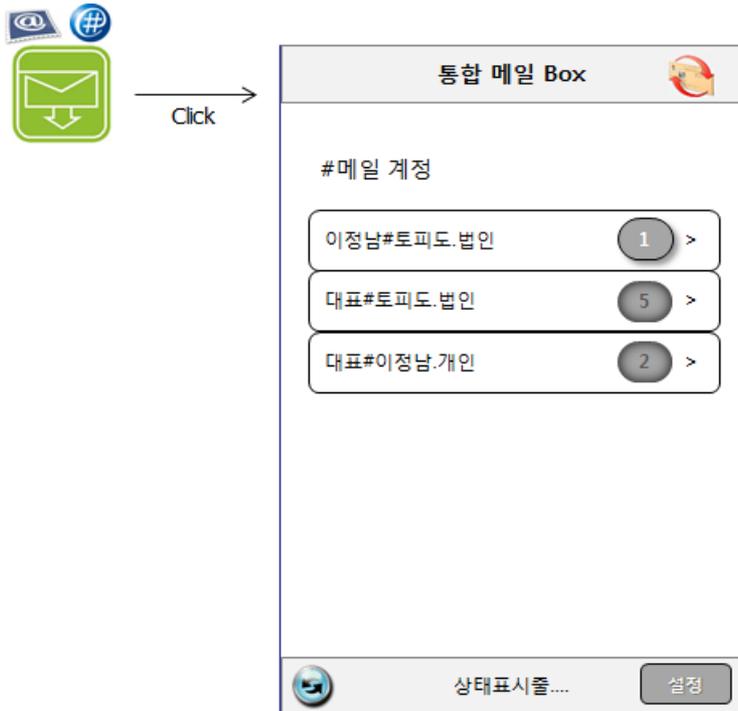
❖ 개발 화면 흐름도



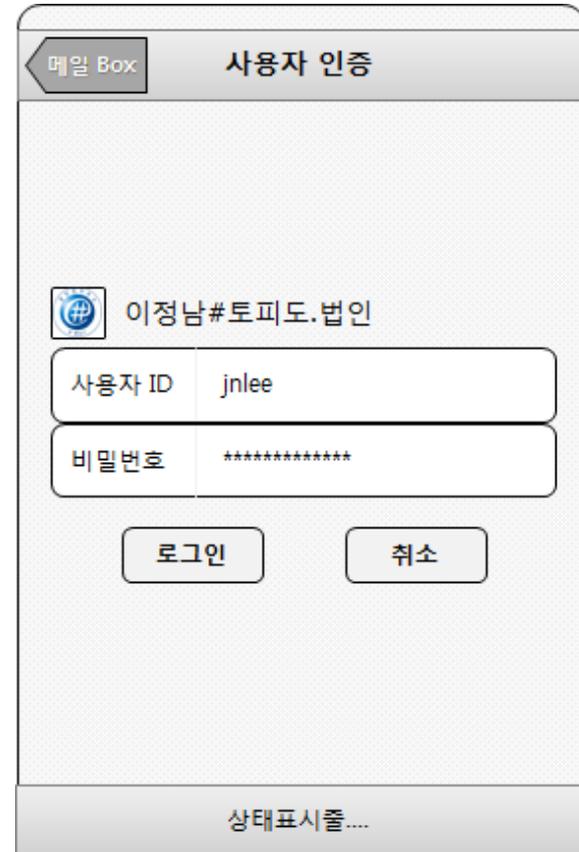
2. 개발 방안

인터페이스 구분	인터페이스 설명
로그인	유통 클라이언트가 사용자 인증을 받고 인증 토큰 값을 전달하기 위한 인터페이스
로그아웃	클라이언트가 유통 메시징서버와의 접속을 끊고 사용을 중단하기 위한 인터페이스
메시지함 조회	유통 클라이언트가 메시지함에 대한 정보를 조회하기 위한 인터페이스
메시지 조회	유통 클라이언트가 메시지함에 보관된 메시지에 대한 정보를 조회하기 위한 인터페이스
메시지 열람	조회를 통해 획득한 메시지 정보를 통해 실제 메시지의 내용을 보기 위한 인터페이스(본문과 첨부파일의 정보 획득)
메시지 열람 정보 전달	유통 클라이언트가 메시지 열람 후 사용자가 메시지를 열람했음을 유통 메시징 서버에 알리기 위한 인터페이스
메시지 전송	유통 클라이언트가 유통 메시징서버에 메시지 전송을 요청하기 위한 인터페이스
공인전자주소 검색	수신개체의 물리적 주소, 수신자의 계정 유효성 및 보안정보를 획득하기 위한 인터페이스
메시지 삭제	유통 메시징 서버의 메시지함의 메시지를 삭제하기 위한 인터페이스
메시지함 관리	유통 메시징 서버의 메시지함을 관리하기 위한 인터페이스
메시지 이동	유통클라이언트가 메시지를 메시지함 간 이동을 하고자 할 때 사용하는 인터페이스
파일 다운로드	유통 클라이언트가 유통 메시징 서버로부터 파일을 다운로드 하기 위한 인터페이스 (메시지 열람 후 첨부파일을 다운로드 하여 열람할 때 사용)
파일 업로드	유통 클라이언트가 유통 메시징 서버에 파일을 업로드 하기 위한 인터페이스 (유통 메시지 송신 시 미리 파일을 올려놓고 송신할 경우 파일 업로드를 사용)
스팸 신고	유통 클라이언트가 메시지 열람 후 스팸 메시지로 인식할 경우 유통관리시스템에 스팸을 신고하기 위한 인터페이스
유통증명서 발급 요청	송수신한 유통 메시지에 대해서 유통증명서를 발급을 하기 위한 인터페이스

❖ 초기화면



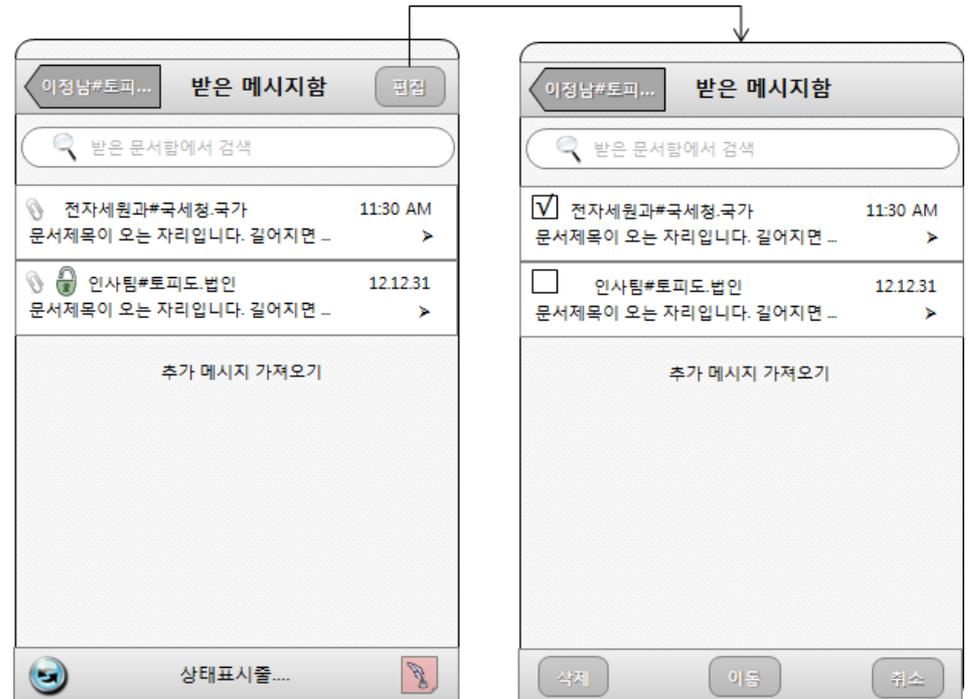
❖ 사용자 인증



❖ 메시지함 목록 조회



❖ 받은 메시지함 조회

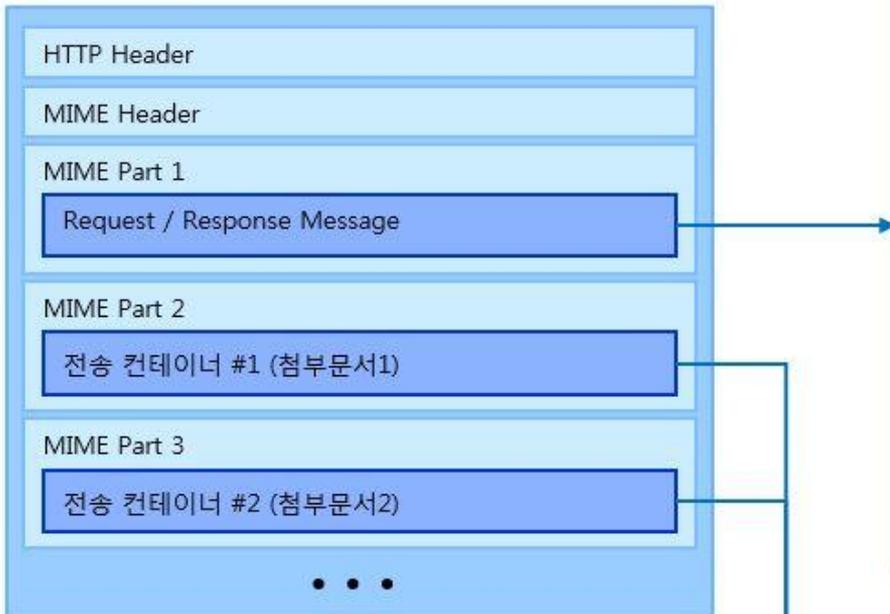


❖ 신규 메시지 작성

❖ 보낸 메시지함 조회

3. 통신 프로토콜

- 유통메시지 서버와 클라이언트 연계를 위한 프로토콜은 **XML-RPC**를 기반으로 함
- XML-RPC는 전송방식에서는 **HTTP** 프로토콜을 사용(**HTTP POST** 방식의 요청만 사용)하며 데이터 인코딩은 XML 방식을 사용



- 연계 인터페이스에서 정의하는 요청 및 응답 메시지 정보
- 예시) 메시지 전송

```
<?xml version="1.0" encoding="UTF-8" ?>
<Request>
  <SendMsgReq version="x.x">
    <Timestamp> </Timestamp>
    <Title>발송테스트</Title>
    <Sender>테스터#정보통신산업진흥원.법인</Sender>
    <Receiver>hong#홍길동.개인</Receiver>
    <ReqConfirm>1</ReqConfirm>
    <IsEncrypted>1</IsEncrypted>
    <Identifier>154868c7-8e43-49a9-8343-9425bc01c3f1</Identifier>
    <MEPType>1</MEPType>
    <Contents>
      <Content>
        <ContentId>cid:1234567890</ContentId>
      </Content>
    </Contents>
  </SendMsgReq>
</Request>
```

- 요청 및 응답 메시지 관련 추가정보 및 본문/첨부분서 등 (연계 인터페이스 유형에 따라 구성정보가 달라짐)

4. 개발환경

연구 장비

보유기관	연구시설·장비명	규격	수량	용도	활용도 및 시기
토피도(주)	ProLiant DL360P Gen8	CPU: Intel(R) Xeon(R) CPU E5-2603 0 @ 1.80GHz * 2 메모리: 16GB (4x4GB) RDIMM	1	개발 제품의 테스트 대상인 유통메시징서버 운영 장비	모바일 개발 후, 단위테스트 시점부터 활용 (프로젝트 시작부터 1년간 유지)

구분	환경
O/S	Linux
WAS	Apache Tomcat 6
JDK	1.6
DBMS	Oracle 10g

개발자 환경

구분	환경
O/S	Windows 7
개발 IDE	Eclipse Juno
개발 Plug-In	Android Development Tool
JDK	1.6

테스트 환경

구분	버전
갤럭시 S4	젤리빈
갤럭시 S3	아이스크림 샌드위치
갤럭시 S2	진저브레드
Nexus 7	젤리빈



Ⅲ. 추진계획

- 1. 개발 일정**
- 2. 수행조직 및 업무분장**
- 3. 기술개발 추진체계**
- 4. 커뮤니티 운영계획**

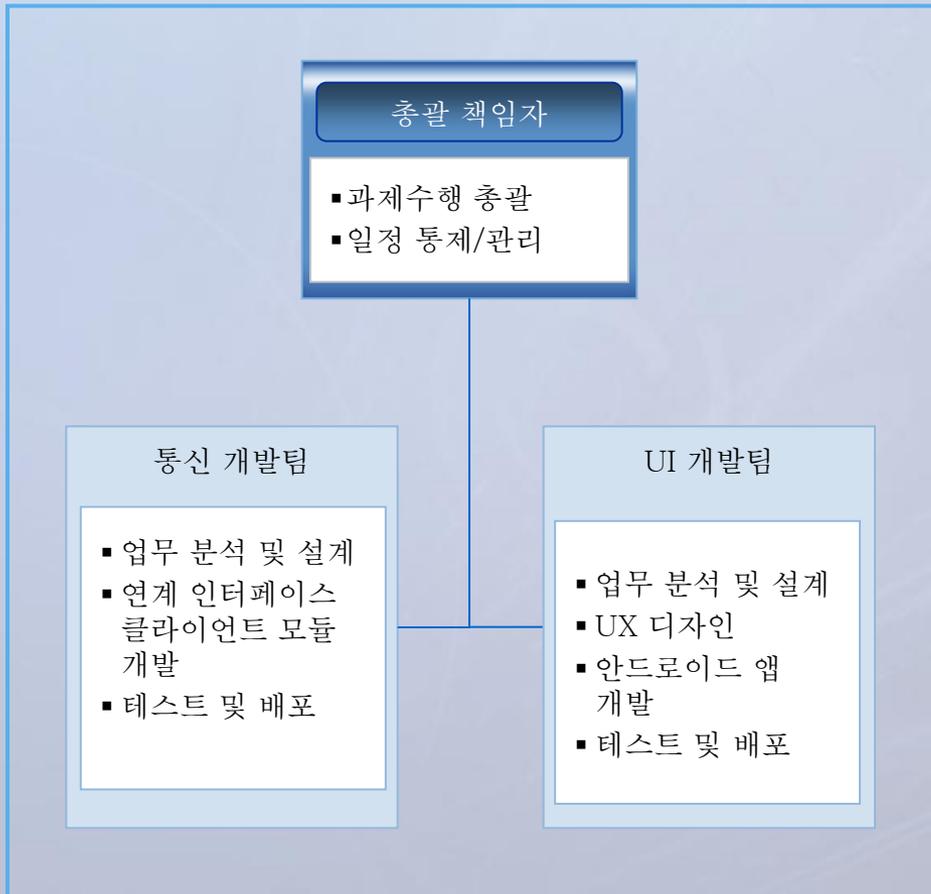
1. 개발 일정

Task	수행 기간							
	5월	6월	7월	8월	9월	10월	11월	
개발환경 Setting	←→							
요구사항 정의	←→		Kick-Off Work Shop					
App 기능 및 화면 설계	←→							
Android용 연계인터페이스 라이브러리 개발		←→						
Android용 App 개발		←→						
소스 등록 및 참여 구성원 모집					←→		완료보고	
테스트 및 보완 작업						←→		

2. 수행조직 및 업무분장

총괄 책임자를 중심으로 개발을 효율적으로 수행하도록 통신 개발팀, UI 개발팀으로 배정하여 관리하도록 하겠습니다.

수행조직 체계



수행조직 별 역할

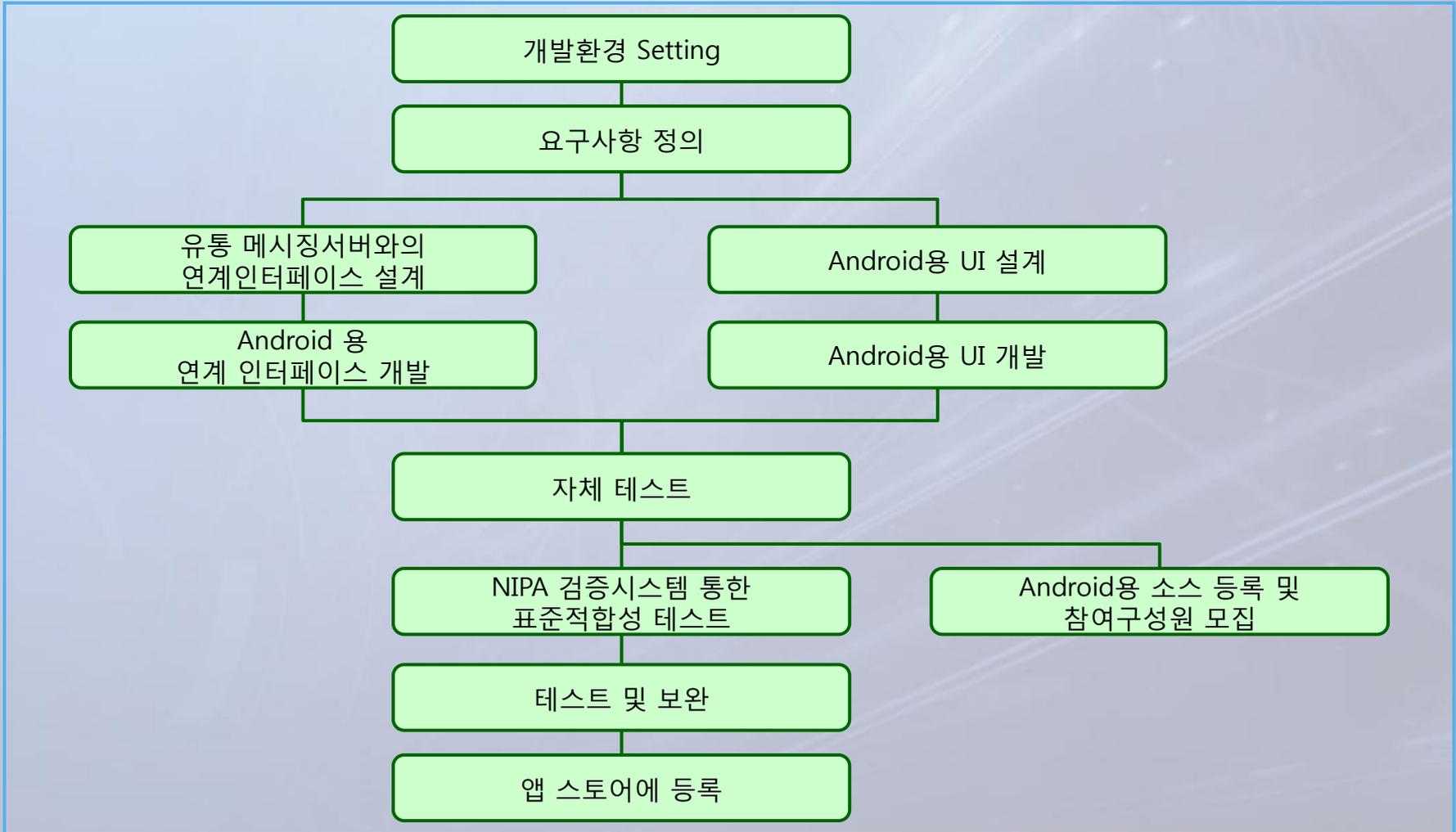
구분	주요 업무 및 역할
총괄책임자	<ul style="list-style-type: none"> 과제 추진 총괄 일정 관리 및 통제
통신 개발팀	<ul style="list-style-type: none"> 업무 분석 및 설계 연계 인터페이스 클라이언트 모듈 개발
UI 개발팀	<ul style="list-style-type: none"> 업무 분석 및 설계 UX 디자인 안드로이드 앱 개발

투입인력 계획

소속	항목	담당 분야	본 과제 참여율(%)	정부출연 연구사업 참여율(%)	전체 참여율(%)
도피도	총괄 책임자	기획	33	25	58
		통신기획	21	0	21
	통신 개발팀	통신개발	71	0	71
		UI 기획	57	0	57
	UI 개발팀	UI 개발	100	0	100
		디자인	11	0	11

3.기술개발 추진 체계

모든 업무의 수행주체는 주관기관인 토피도(주)입니다.



4. 커뮤니티 운영계획

커뮤니티 구성방법

- SourceForge 사이트 활용
- 커뮤니티는 기본적으로 온라인 중심으로 구성함
- 현재 Sourceforge 등록된 커뮤니티 URL: "<https://sourceforge.net/projects/ecfclient>"

운영 계획

- 소스는 SourceForge에서 제공되는 SVN을 통해서 업데이트 관리함
- Issue tracking tool도 SourceForge에서 제공되는 것을 사용하여 프로젝트 참여 구성원(commiter)과 issue들을 공유하고 같이 해결함
- SourceForge에서 제공되는 forum을 활용하여 committer 뿐만 아니라, downloader, user들과 의견을 공유하고 토론함
- SourceForge에서 제공되는 다운로드 통계는 오픈소스의 활동상황을 보여주는데 활용함

발전 방향

- 프로젝트 시작 시에 SourceForge에 가입하고 프로젝트를 등록함
- 프로젝트 시작 시점부터 4개월 후에 안드로이드 버전의 라이브러리와 앱을 등록하고 소스를 공개함
- 소스공개 이후 지속적으로 업그레이드 버전을 발표하고 참여자 및 사용자의 의견을 반영하여 개선함
- 오프라인 오픈소스 커뮤니티에 참여하여 프로젝트의 내용과 활용성을 다른 개발자에게 제공함
- 향후 당 프로젝트의 참여자가 많아지게 되면 별도의 독자적인 사이트를 구축하는 것을 고려함



질의응답 (QnA)





감사합니다.