2016/10/19

캡스톤 설계

암호화 메신저 시스템

컴퓨터소프트웨어공학과

201321283 고현서

201321286 김미옥

201321300 명소희

201321319 이민영

목차

개요

- **배경**
  - 중앙 집중 형 서버를 이용한 메신저나 거래시스템에서 발생하는 문제
    - 데이터 해킹, 위조 문제
    - 서버 과부하에 따르는 트래픽 발생
  - 기프티콘과 같은 쿠폰을 한번에 모아서 관리하기 어려움

개요

- **개발 목표**

  - 데이터 해킹, 위조 문제 / 서버 과부하에 따르는 트래픽 발생

    → 중앙서버 없이 사용자끼리 안전하게 메시지를 주고 받을 수 있는 커뮤니티 서비스 구현

  - 기프티콘과 같은 쿠폰을 한번에 모아서 관리하기 어려움

    → 쿠폰내역 서비스 구현(거래를 마친 쿠폰을 사용 혹은 선물할 때, 쿠폰에 대한 정보를 암호화하여 전송)

기존 서비스와 비교

- **기존 서비스와 비교표**

종류 / 특징	Telegram	KakaoTalk	Bleep	Sock-Dark
P2P방식	X	X	O	O
모바일 쿠폰함	**X**	**카카오 서비스만** 가능	X	다양한 모바일 쿠폰 서비스 보관 가능
시스템 단점	사용량 증가로 인한 **서버 과부하**로 실행 속도 느려짐	- 서버에 저장된 **대화 내용이 노출**될 수 있음 - 사용량 증가로 인한 **서버 과부하**로 실행속도 느려짐	알파버전이기 때문에 **기능 부실**	기술 발전단계라 보완 작업이 필요
종단간 암호화	O	수동(비밀채팅) 선택	O	O

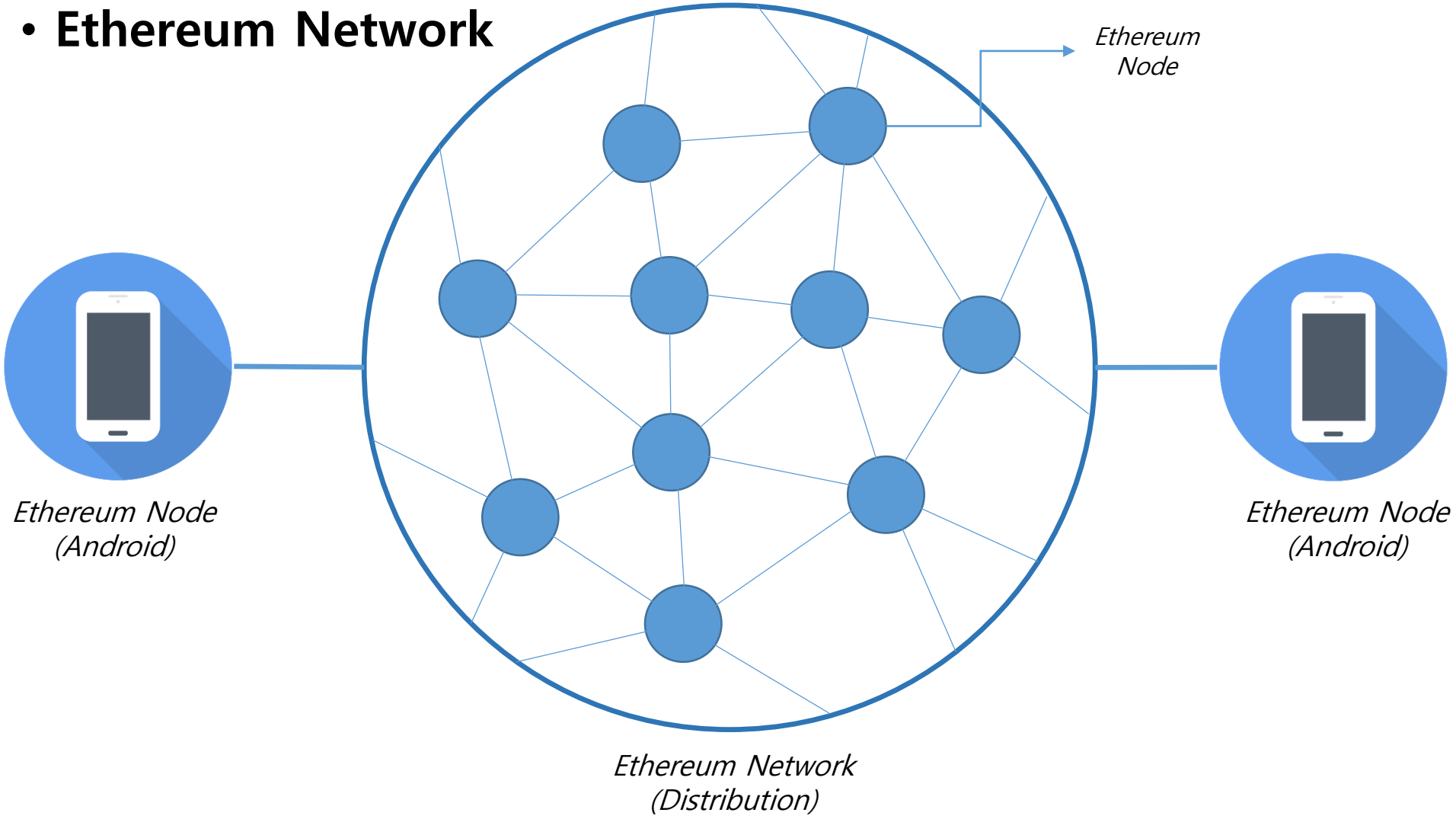
특징

- **Sock-Dark 장점과 단점**

구분	특징
장점	중앙 서버 없이 P2P네트워크를 이용하여해킹에 대한 **위험성을 최소화**
	디지털 서명과 암호화를 통해 **메시지를 보호**
	서버가 없으므로 서버의 **과부하가 일어나지 않음**
단점	국내에 본격적으로 도입되기 위해서는 법적/기술적 **보완이 필요**함

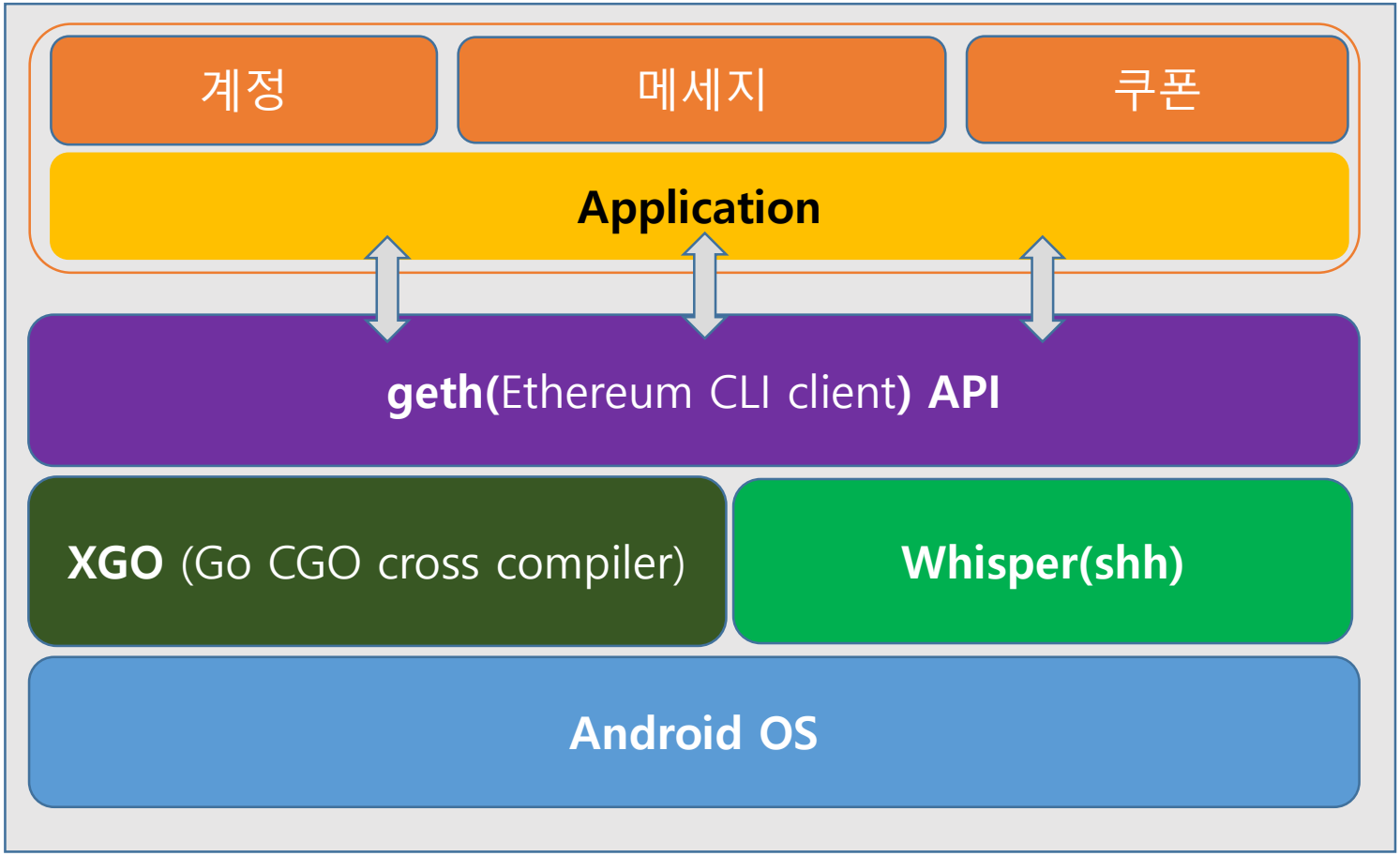
Network Topology

- **Ethereum Network**

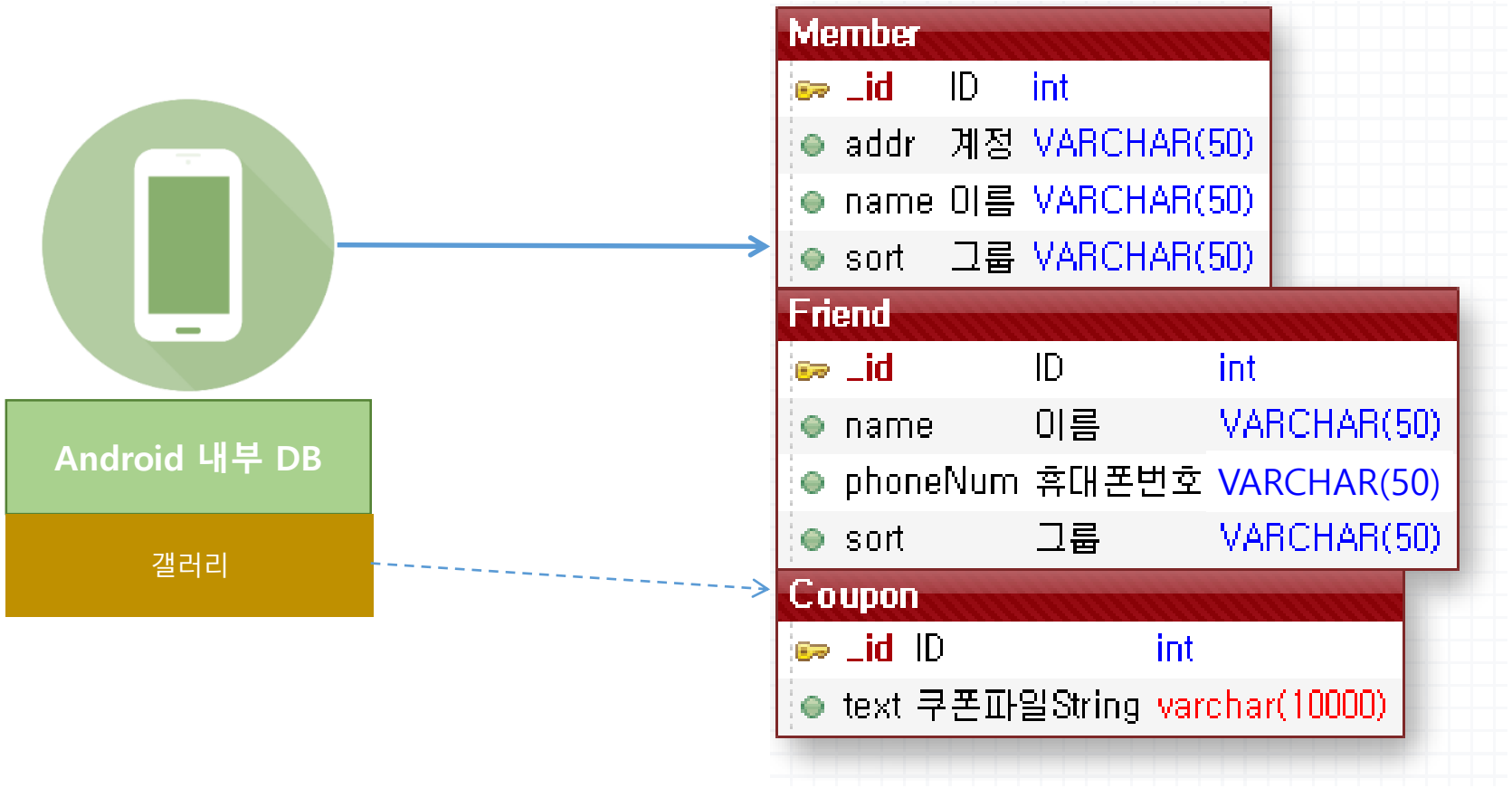


System Architecture

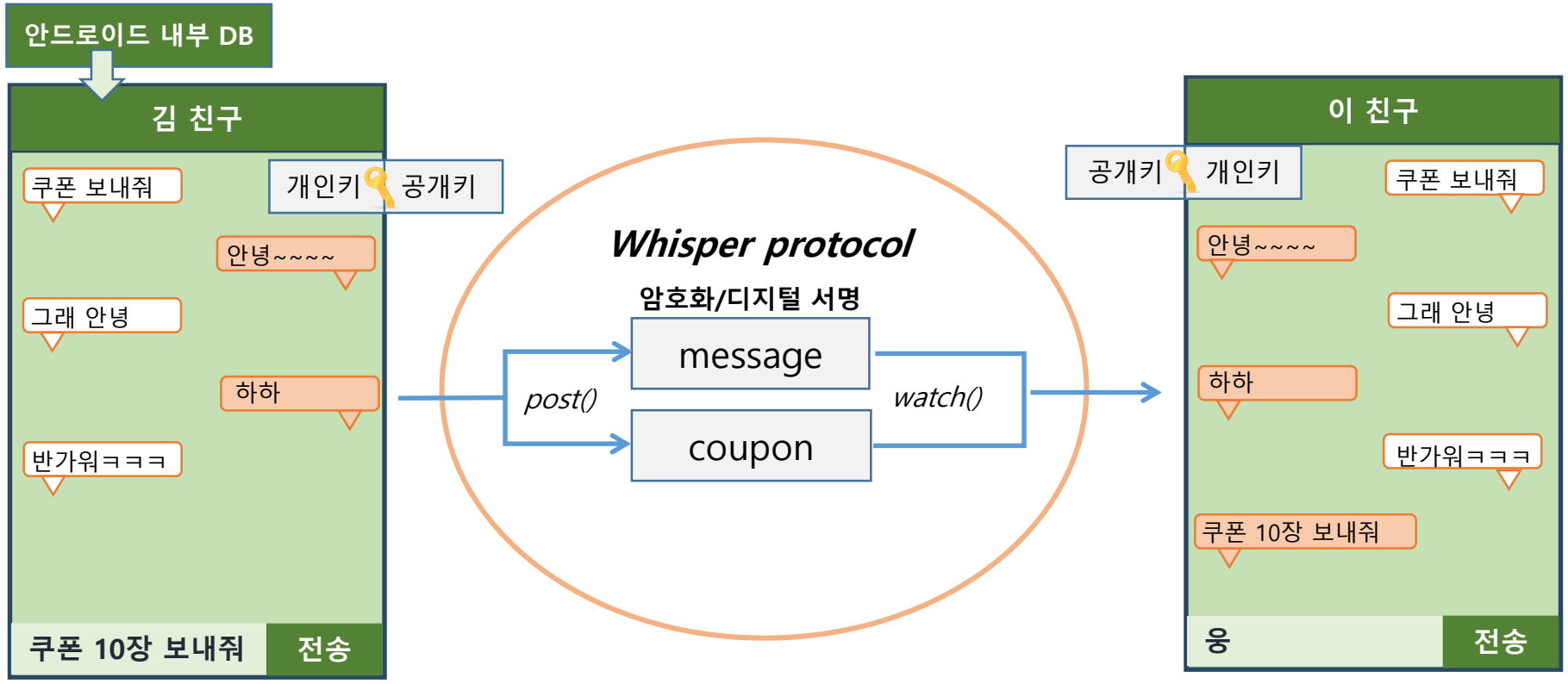
- **Software Architecture**



스마트폰 Peer 구성(DB 중심)



전송방식



디지털 서명/암호화 기술을 통한 메시지 전송

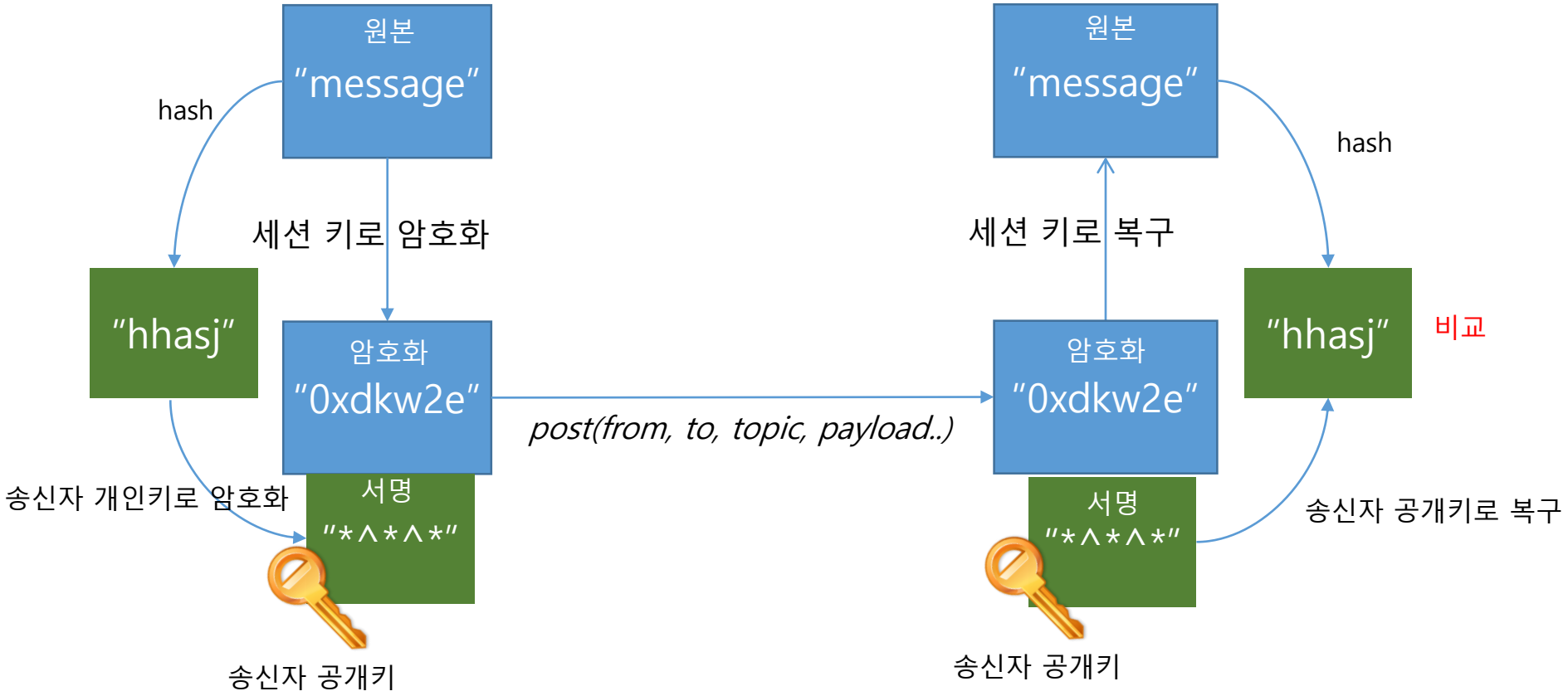
암호화 방식

❖ **메시지 / 쿠폰 송·수신** : 디지털 서명 + 암호화

<table>
<tr><th>기능</th><th>데이터 타입</th><th>암호화 방식
(디지털 서명)</th><th>암호화 방식
(암호화)</th></tr>
<tr><td>메시지
송·수신</td><td>text</td><td rowspan="2">• 공개 키 암호화 방식
1. 개인 키로 암호화 한 후 공개키로 복호화
2. 송신자가 개인키로 암호화된 서명과 송신자의 공개 키를 함께 전송
• 데이터 무결성
• 송·수신자 인증
• 부인 방지</td><td rowspan="2">• 대칭 키(세션 키) 암호화 방식
1. 메시지 전송 시 일부 정보를 공유하고 세션 키 생성
2. 송신자와 수신자가 세션 키로 메시지를 송·수신
• 쿠폰 데이터 보안(기밀성)</td></tr>
<tr><td>쿠폰
송·수신</td><td>file
↓
text</td></tr>
</table>

암호화 방식

- **메시지 / 쿠폰 전송 방식 (디지털 서명+암호화 과정)**



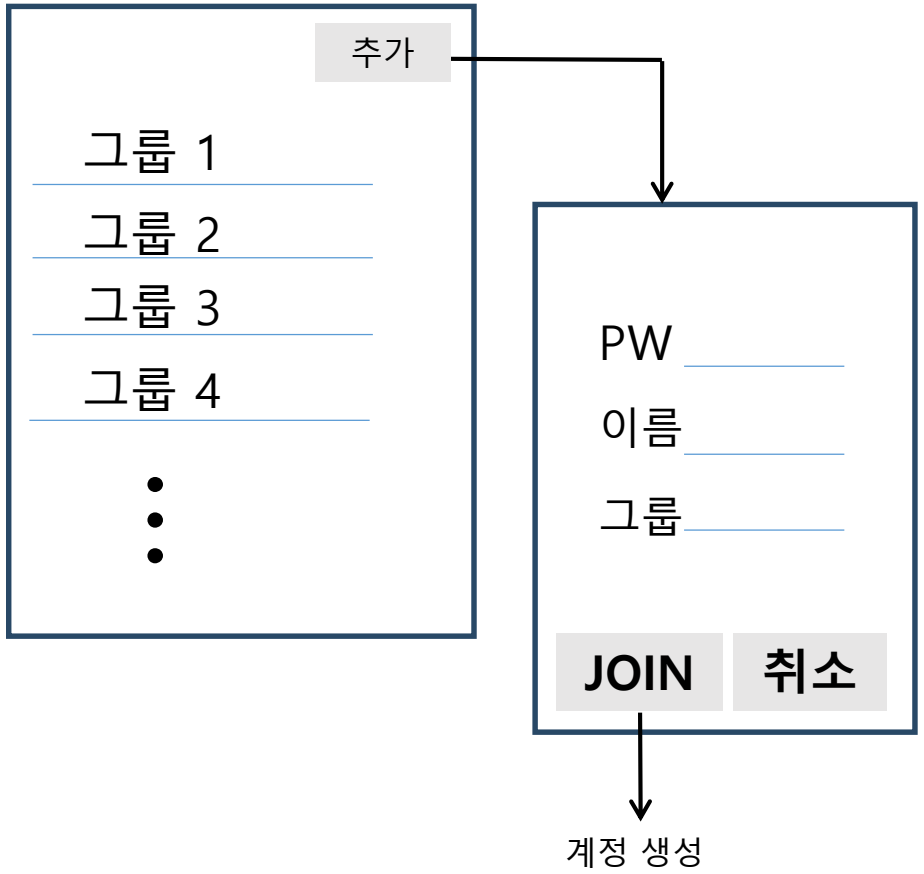
시스템 기능(개요)

- **상세 기능 설명**

기능	상세기능		상세설명
계정	계정생성		시스템 사용을 위한 계정을 생성
	친구관리	그룹화	그룹을 지정하여 그룹별로 친구 관리
		등록	친구의 이름과 휴대폰 번호로 친구등록
		삭제	친구목록 삭제
메시지	메시지 송·수신		디지털서명과 암호화 기술을 이용한 메시지 송·수신
쿠폰	쿠폰관리	쿠폰등록	갤러리 내 제공된 쿠폰을 등록
		쿠폰삭제	등록된 쿠폰을 삭제
	쿠폰 송·수신		디지털서명과 암호화 기술을 이용한 쿠폰 송·수신

시스템 기능(상세)

- **계정 관리 ① 계정 생성 (친구 그룹화)**



① 추가 클릭

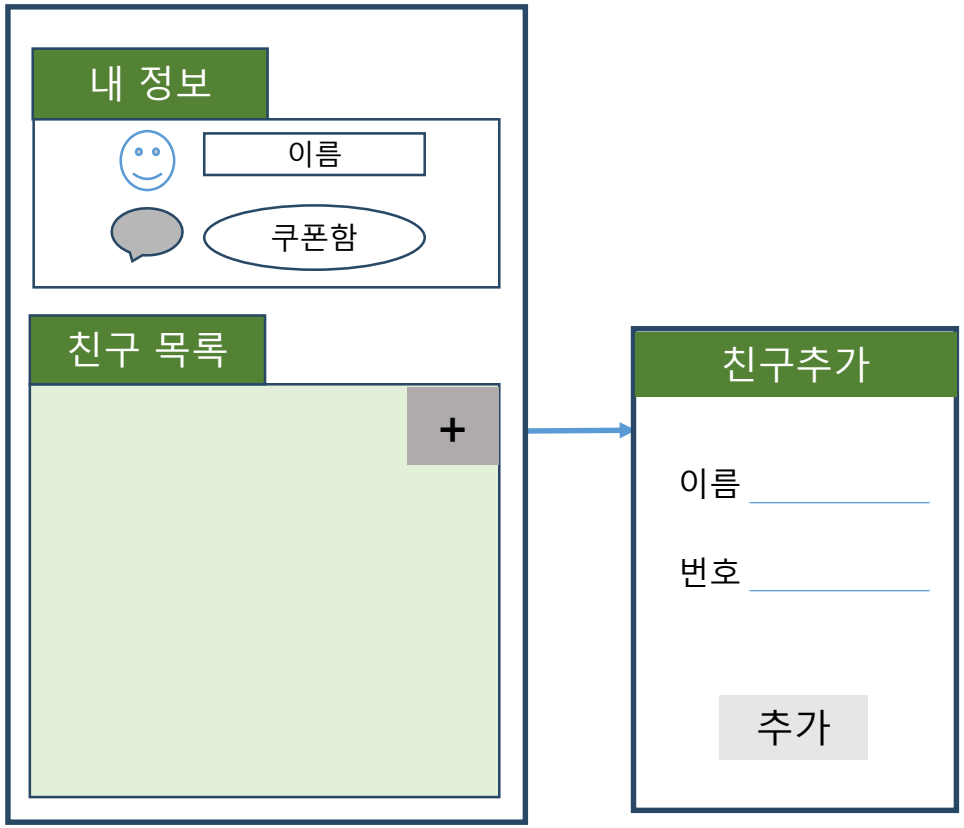
② 정보를 입력

- geth 서비스에 연결

③ 계정, 그룹 생성

시스템 기능(상세)

- **계정 관리 ① 친구 추가**



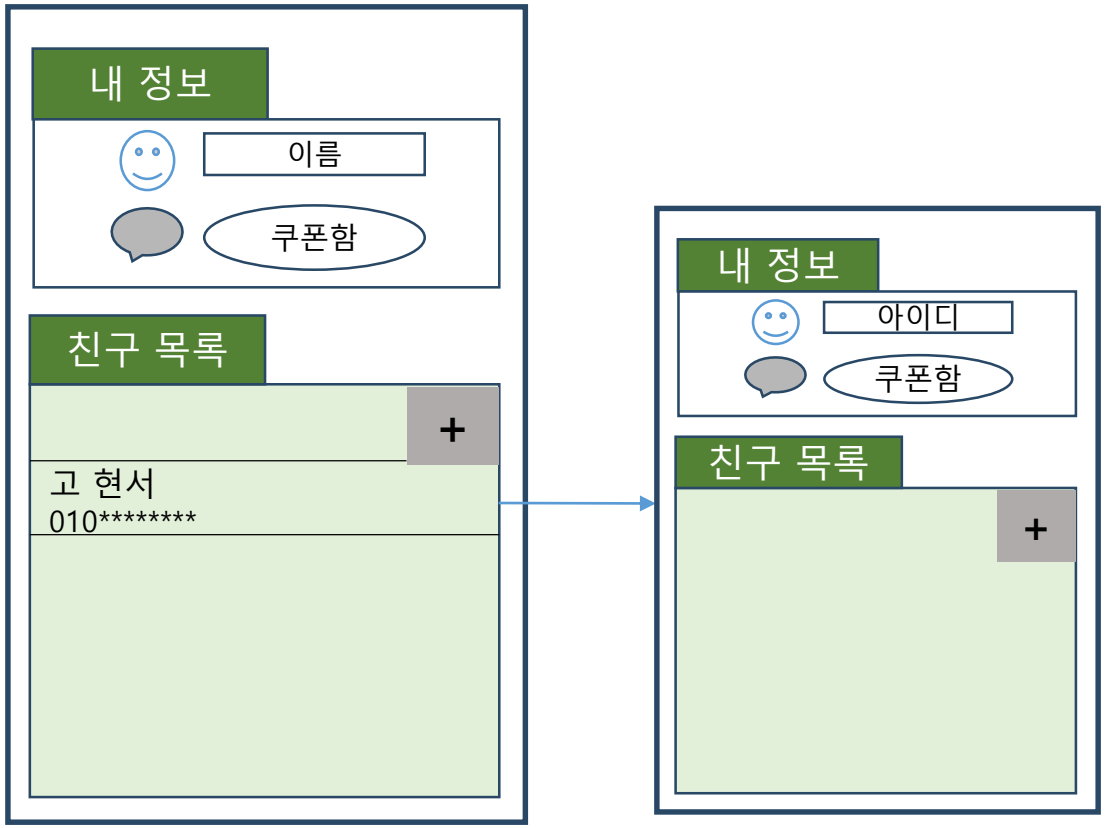
① 친구 추가 클릭

② 친구의 정보 입력

③ 추가완료

시스템 기능(상세)

- **계정 관리 ② 친구 삭제**

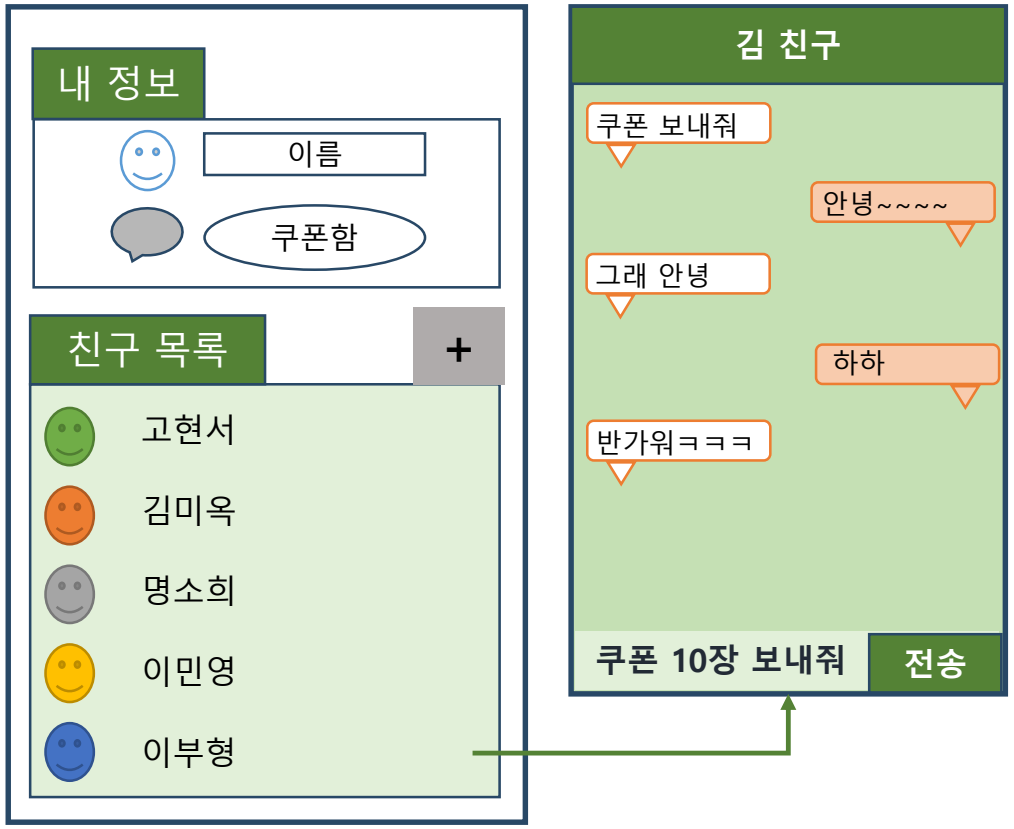


① 친구 목록 길게 클릭

② 친구 삭제

시스템 기능(상세)

- **메시지 전송 ① 메시지 송 • 수신**



① 메시지를 받을 친구를 클릭하여 메시지를 입력

web3.shh.newIdentity(); →공개키 생성

② 메시지와 함께 생성된 세션 키로 암호화 한 뒤 친구에게 전송됨

web3.shh.post(); →메시지 전송

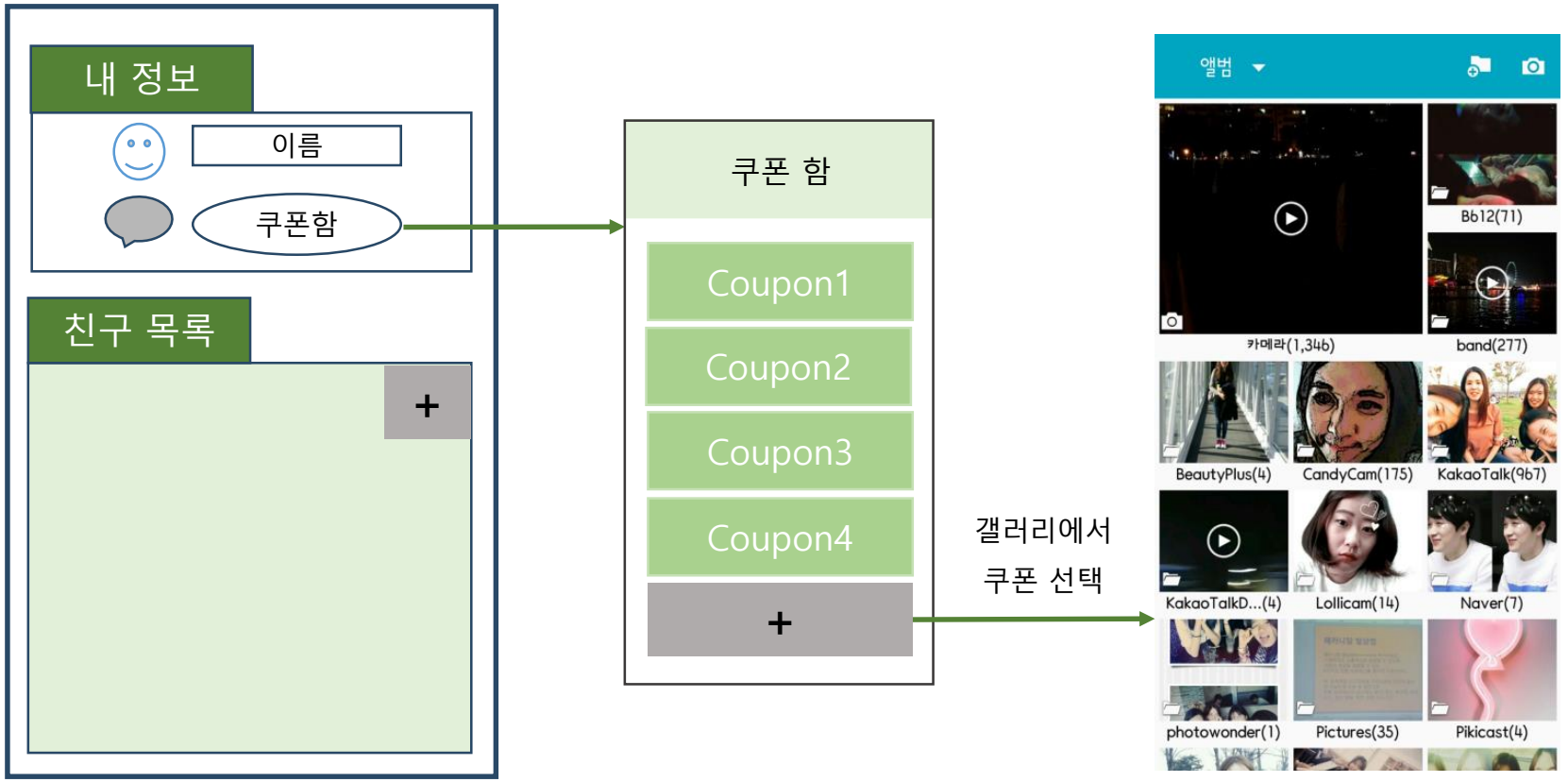
③ 메시지를 받은 친구는 세션 키로 메시지를 확인

web3.shh.filter.watch(); →메시지 수신

④ 전송된 메시지는 디지털 서명의 과정(나의 공개키)을 통해 유효함을 입증

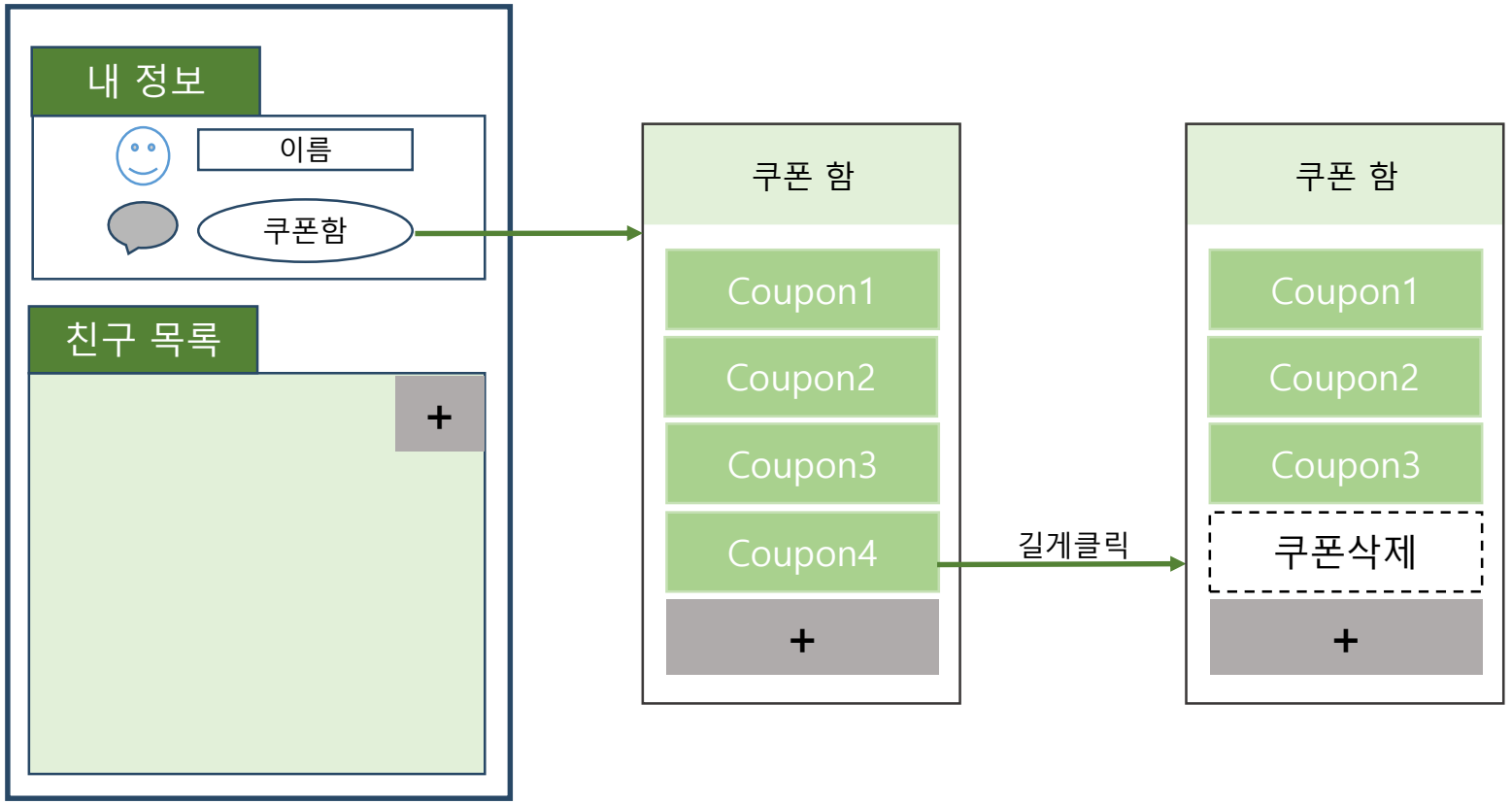
시스템 기능(상세)

- **쿠폰 관리 ① 쿠폰 등록**



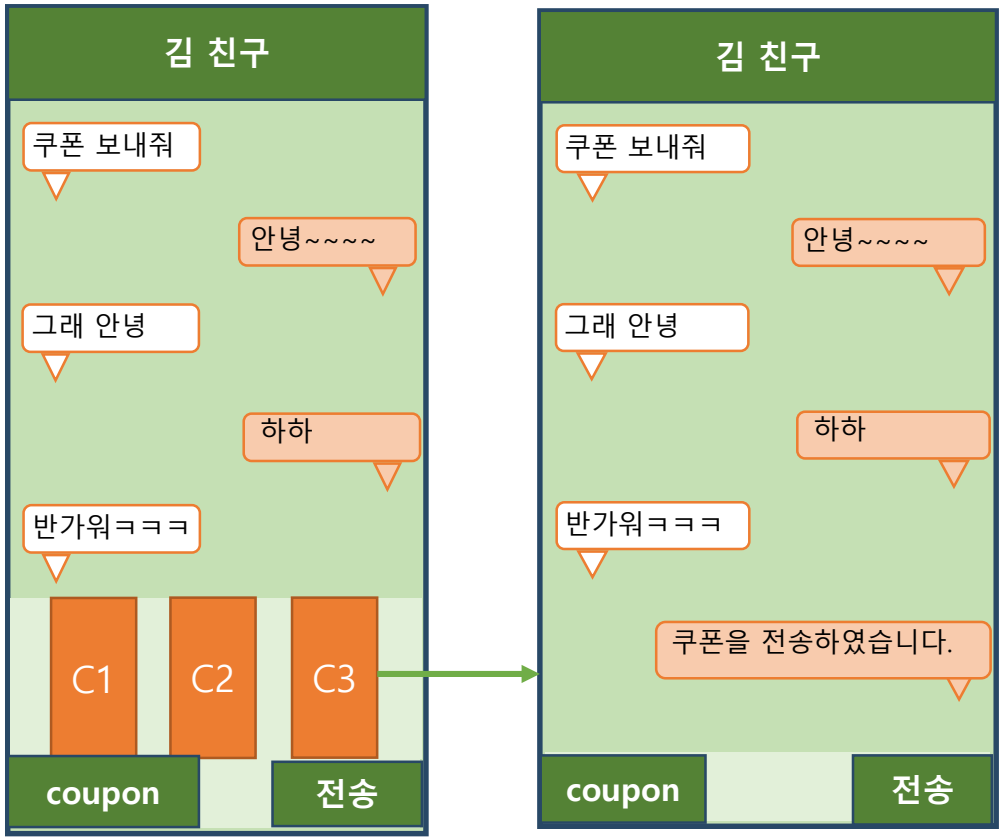
시스템 기능(상세)

- **쿠폰 관리 ② 쿠폰 삭제**



시스템 기능(상세)

- **쿠폰 전송 ③ 쿠폰 송 • 수신**



① 메시지를 받을 친구를 클릭하여 쿠폰 선택

② 선택된 쿠폰과 함께 생성된 세션 키로 암호화 한 뒤 친구에게 전송됨

web3.shh.post(); →메시지 전송

③ 쿠폰을 받은 친구는 세션 키로 메시지를 확인

web3.shh.filter.watch(); →메시지 수신

④ 전송된 쿠폰은 디지털 서명의 과정(나의 공개키)을 통해 유효함을 입증

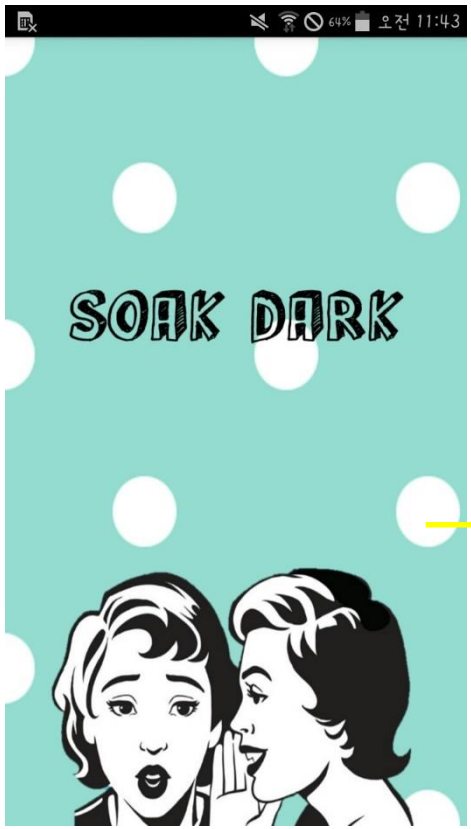
개발 진행 사항

- **Android 와 geth (go-ethereum console client) 연결**

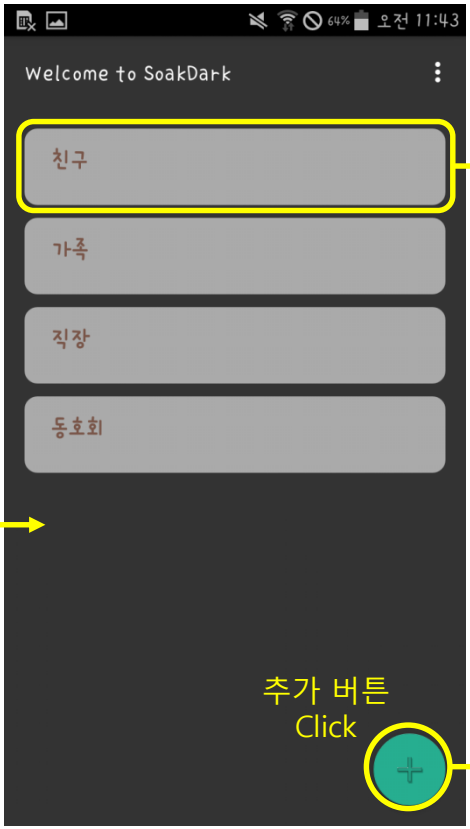
▲ https://github.com/Seedstars/go-ethereum-android 에서 제공하는 API

개발 진행 사항

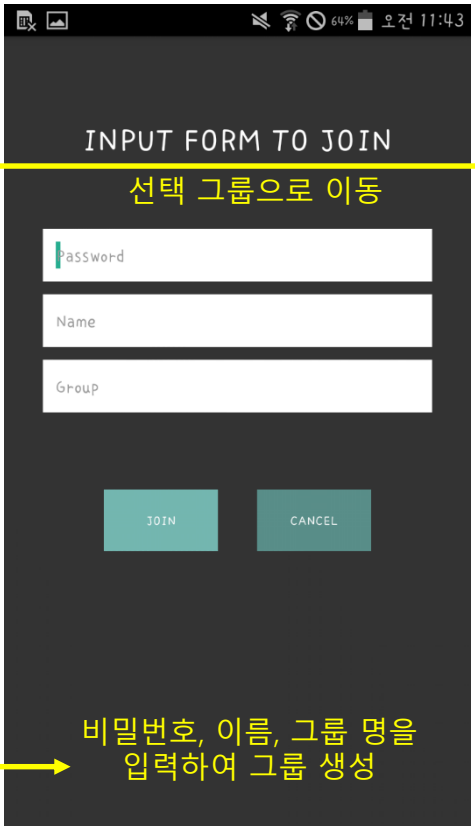
- **계정**



<App이 켜질 때 로딩 화면>



<그룹 목록>



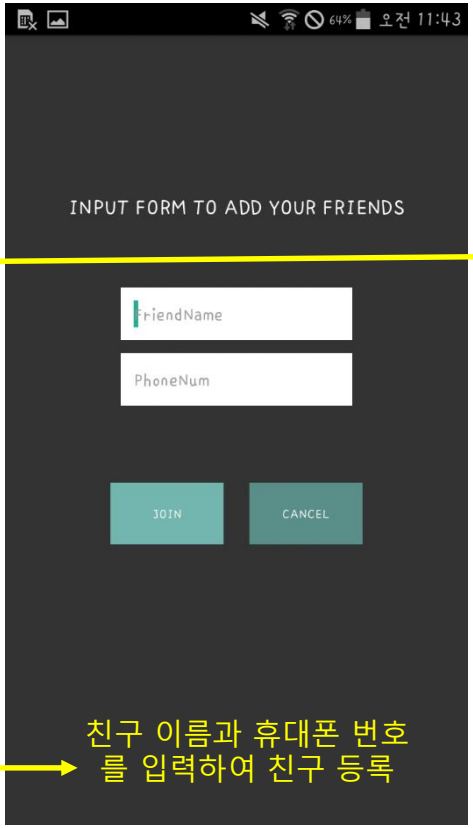
<그룹 추가 화면>

<친구목록 화면>

개발 진행 사항

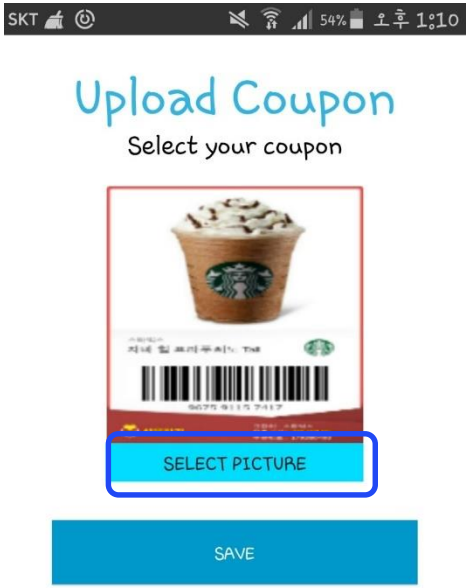
- **친구/쿠폰**

<친구 목록 화면>



<친구 추가 목록>

<쿠폰함 화면>



<쿠폰 선택화면>

개발 진행 사항

- **메시지/ 쿠폰 송·수신**

서로의 Identity가 교환이 되면 '대화가 준비되었습니다 ' 라고 문구가 바뀜

<친구 목록 화면> <선택 친구 대화 화면> <대화 쿠폰 함 화면> <상대방 대화 화면>

기대 효과

- **편리성 향상** 발행된 쿠폰을 찾을 필요 없이 한 곳에 모아 확인 가능

- **보안강화** 데이터베이스에 따로 데이터가 저장되지 않아 유출 방지를 할 수 있으며 Ethereum의 기술 중 디지털 서명과 암호화 과정을 통해 보안이 강화됨

- **비용 절감** 중앙 서버가 없으므로 관리비용이 줄어듦

감사합니다!

김미옥 고현서

명소희 이민영