목차

1. 과제 배경 및 목표

1.1. 과제 배경

- 개인 정보 보호 및 기밀 유지의 중요성

개인정보는 전자상거래, 고객관리, 금융거래 등 사회의 구성, 유지, 발전을 위한 필수적인 요소로서 기능하고 있다. 특히 데이터경제 시대를 맞이하여 개인정보와 같은 데이터는 기업 및 기관의 입장에서도 부가가치를 창출할 수 있는 자산적 가치로서 높게 평가되고 있다. 그러나 개인정보가 누군가에 의해 악의적인 목적으로 이용되거나 유출될 경우 개인의 사생활에 큰 피해를 줄 뿐만 아니라 개인 안전과 재산에 피해를 줄 수 있다.

지난 5월 6일, 정부24에서 다른 사람의 성적·납세증명서가 발급되는 등 오류가 발생했던 사실이 보도되었다. 해당 서류에는 이름과 주민등록번호 같은 민감한 개인 정보가 포함되어 있었다.

유출된 개인정보는 스팸메일, 불법 텔레마케팅 등에 악용되어 개인에게 원치 않는 광고성 정보가 끊임없이 전송되는 동시에 대량의 스팸메일 발송을 위한 계정 도용, 보이스 피싱 등 범죄행위에 악용될 우려가 있다. 이러한 문제점이 개인정보의 주체에게 미치는 정신적·물질적 피해 규모는 측정이 어렵다. 뿐만 아니라 한번 유출된 개인정보는 회수가 사실상 불가능하기 때문에 더욱 심각하다고 할 수 있다.

- 신기술 발달에 따른 개인 정보 보호

현대 정보통신기술의 발전에 따라 개인정보를 이용하고자 하는 기업 및 정부 등의 욕구가 커지고 있다. 한편, 정보주체인 개인의 자기정보에 대한 접근권과 통제권이 강화되어 왔다. 그럼에도 불구하고 정보통신기술의 발전은 개인정보 보호에 새로운 과제를 꾸준히 제공하고 있다.

정보통신기술은 정보에 대한 접근성을 변화시킴으로써 사생활에 영향을 미칠 뿐만 아니라 사생활 규범 자체를 변화시키기도 한다. 예를 들어, 소셜 네트워킹 사이트는 사용자가 다른 방법보다 더 많은 정보를 공유하도록 초대한다. 이러한 새로운 문제에 대처해 가지 않으면 안된다. 신기술의 홍수 속에서 그것이 개인정보와 프라이버시에 미치는 영향에 대해 정확히 인식하면서 적절한 규제와 보호가 이루어질 필요가 있다.

디지털 혁명(Digital Revolution) 시대에 물리적 공간은 덜 중요해지고, 정보는 인터넷상에서 자유롭게 획득되고, 결합하고, 유통되고 있다. 이러한 현실 속에서 개인의 프라이버시 보호는 그 중요성이 더욱 커지는 면도 있다. 개인의 프라이버시를 과도하게 침해하지 않으면서 정보공유 및 공유효율성을 달성하기 위한 접점을 찾아야 할 것이다.

- 분산 시스템의 필요성

지난해 11월, 국가 행정망 전산마비 사태가 발생했다. 17일 공무원 전용 행정전산망 및 정부24 서비스가 마비되었고 사흘만인 19일 복구되었다. 그리고 22일 주민등록시스템에서 일시 장애가 발생했고 이어서 23일, 조달청 국가종합전자조달시스템이 불통이 되는 문제가 발생했다. 뿐만 아니라

24일, 정부 전자증명서 발급 및 모바일 신분증 서비스가 마비되는 사태가 발생했다.

사고 원인으로는 대한민국의 본인 확인 절차에서 무조건 행정안전부와 경찰청 전산을 거치는 시스템의 문제였음이 밝혀졌다. 복구가 늦어진 원인은 행정전산망의 백업을 위한 데이터가 서버가 아닌 단순 저장장치인 스토리지에 담겨 있는 탓에 사람이 직접 백업 데이터를 현재 전산망에 옮겨가며 복구를 해야해 시간이 많이 걸렸다고 한다.

단일 서버 및 물리적 스토리지에 의존하는 것은 위처럼 단일 실패 지점을 만들고 이는 무중단 서비스 제공에 큰 문제가 된다. 하지만 분산 시스템의 경우, 하나의 노드에서 장애가 발생하더라도 다른 노드가 서비스를 이어서 할 수 있다. 또한 동시에 더 많은 정보를 처리할 수 있으며 시스템의 확장이 보다 간편하다는 이점을 가지고 있다.

1.2. 과제 목표

- 본 과제에서는 블록체인 상에서 증명서를 발급과 조회를 한번에 할 수 있는 블록체인 기반 증명서 발급 및 조회를 개발하고자 한다. DID를 통한 신원 인증으로 개인이 발급된 증명서를 관리하는 주체가 되도록 하고, 허가형 블록체인으로 검증된 유저만 증명서를 발급 및 조회를 할 수 있다.

- 기존 서비스는 중앙화 저장소에 모든 데이터를 의존하여 저장하고 필요한 데이터를 조회하여 응답을 보내는 방식이다. 이러한 방식은 단일 실패 지점을 만들어 사용자들은 이용에 불편함을 겪었다. 따라서 본 과제에서는 분산 시스템인 블록체인에 기반한 증명서 발급 및 조회 시스템을 통해 탈중앙화를 구현하고자 한다.

1.3. 기대 효과

개인정보 유출 방지	- DID를 통한 신원 증명 시스템으로 사용자에게 신뢰성 있는 정보를 제공할수 있다.
안정성 개선	- 단일 실패 지점을 만들지 않아 장애가 발생해도 서비스를 문제없이 계속해서 제공할 수 있다.
데이터 주권 실현	- 중앙 시스템에서 모든 데이터를 소유하고 관리하는 중앙집중형 방식에서 벗어날 수 있다.
중개 수수료 절감	- 블록체인 상에서 데이터 검증과 발급, 조회가 이루어지므로 수수료 절감이 가능하다.
투명성 제공	- 여러 노드의 합의 알고리즘으로 데이터 위·변조가 어렵고 상호 검증을 통해 투명한 정보 전달이 가능하다

1.4. 현실적 제약 사항 분석 결과 및 대책

블록체인 특성상 한번 체인에 저장된 자료는 삭제 및 수정이 불가능하다는 특징이 있다. 그러나 개인정보보호법 제21조에는 아래와 같이 명시되어 있다.

개인정보처리자는 보유기간의 경과, 개인정보의 처리 목적 달성, 가명정보의 처리 기간 경과 등 그 개인정보가 불필요하게 되었을 때에는 지체 없이 그 개인정보를 파기하여야 한다. 다만, 다른 법령에 따라 보존하여야 하는 경우에는 그러하지 아니하다.

세부 조항에는 아래와 같은 조항이 존재한다.

② 개인정보처리자가 제1항에 따라 개인정보를 파기할 때에는 복구 또는 재생되지 아니하도록 조치하여야 한다.
③ 개인정보처리자가 제1항 단서에 따라 개인정보를 파기하지 아니하고 보존하여야 하는 경우에는 해당 개인정보 또는 개인정보파일을 다른 개인정보와 분리하여서 저장·관리하여야 한다.
④ 개인정보의 파기방법 및 절차 등에 필요한 사항은 대통령령으로 정한다.

현재 법률 상으로는 개인정보 수정 및 삭제가 불가능하다는 블록체인의 특성이 한계점으로 작용한다. 하지만 부산광역시에서는 블록체인 규제자유특구로 선정되어 개인정보 파기를 오프체인으로 할 경우도 인정해주어 한시적으로 규제가 풀린 상황이다. 따라서 법률적 제약 사항 속에서 실제 서비스 상용화보다는 구현가능성, 발전가능성과 같은 방향성에 초점을 두고자 한다.

2. 요구사항 분석

2.1. 기능 요구사항

요구사항	상세 설명
회원가입 기능	- 발급기관 : 인증서를 통한 기관 및 본인인증 수행 (구현상 생략) - 신청자 : 가입 정보(아이디, 이메일, 비밀번호)와 인적사항(이름, 주소지, 전화번호 등) 등록을 통한 DID 등록, 증명서 저장 지갑 생성
로그인 기능	- 사용자가 입력한 값과 DID 체인에 저장된 값을 비교해 로그인 여부 결정 - 로그인 승인 시 JWT 토큰을 통해 기능 사용
메인페이지 기능	- 로그인 후 가장 먼저 보이는 화면 - 증명서 발급 신청, 조회, 검증 서비스 이용 가능
마이페이지 기능	- 발급기관 : 해당 기관에서 발급된 증명서 기록 제공, 등록된 증명서의 양식 수정 기능 제공, 새로운 증명서 등록 기능 제공 - 신청자 : 발급받은 증명서 기록 제공, 발급된 증명서 조회 기능 제공
증명서 발급 기능	- 발급기관에서 사전에 등록된 데이터와 신청자의 신원 정보가 일치할 경우 기존에 등록된 양식에 맞추어 증명서 발급 - 발급된 증명서는 신청자의 개인 지갑에 보관
증명서 조회 기능	- 신청자 개인 지갑에 저장되어 있는 증명서 조회 - 증명서 소유자는 발급된 증명서를 출력 가능

	- 발급된 증명서의 PDF 파일을 이메일 전송 가능
증명서 온·오프라인 진위확인 기능	- 요청이 들어온 증명서에 대한 진위 확인 기능 제공

2.2. 사용자 요구사항

사용자	상세 설명
발급기관	- 서비스 이용을 위한 회원가입 및 로그인 - 증명서 양식 등록 - 증명 데이터 등록 - 증명서 통계 관리
신청자	- 서비스 이용을 위한 회원가입 및 로그인 - 증명서 발급 신청 - 증명서 조회 - 증명서 출력 - 증명서 진위확인 - 증명서 이메일/팩스 전송 - 수수료 결제

3. 설계 문서

3.1. 사용 기술

3.1.1. 영지식 증명 기술 (ZKP, Zero-Knowledge Proof)

영지식 증명 기술은 증명자(prover)가 검증자(verifier)에게 자신의 비밀과 관련해 어떠한 정보도 노출하지 않고 비밀의 유효성을 증명하는 방법으로 사전에 정의된 연산에 대해 비밀 입력 값은 공개하지 않고 입출력 값의 관계에 해당하는 비밀 입력 값을 알고 있음을 증명하는 암호 기술이다.

증명자는 영지식 증명 기술을 통해서 해당 함수 및 입력 값과 출력 값의 관계를 증명하는 증명값을 만들고 검증자는 증명값 만을 가지고 함수의 입력과 출력의 관계가 맞는지 확인하여 검증 시 비밀 입력 값이 필요하지 않고 유출되지 않는다.

증명할 문장이 참이면 정직한 증명자는 정직한 검증자에게 이 문장이 사실임을 납득시킬 수 있는 완전성, 증명할 문장이 거짓이면 어떠한 거짓된 증명자도 정직한 검증자에게 사실이라 납득시킬 수 없는 건실성, 검증자는 증명할 문장의 참, 거짓 이외에 아무것도 알 수 없는 영지식성의 조건을 만족하여야 한다.

성인 증명에 사용할 수 있는 범위 증명(range proofs), 특정 국가의 시민임을 증명할 수 있는 집합의 원소(set membership) 증명, 신원 정보 중 성별이 남성인지 여성인지 증명할 수 있는 비교(comparison) 증명, 계산 무결성을 증명하는데 활용할 수 있다.

3.1.2. DID (Decentralized Identifier, 탈중앙화 식별자)

탈중앙화 식별자(이하 DID)는 웹 관련 표준을 개발하는 사설표준화기구 W3C의 Decentralized Identifier Working Group에 의해 2022년 7월 제정이 완료된 표준이다. 사용자에 대한 탈중앙화 디지털 신원을 구현하기 위한 아키텍처, 데이터 모델 등을 정의한다. DID는 중앙관리기구가 필요하지 않으며, DID의 식별 대상의 통제 하에 관리된다.

모든 DID는 해당 식별 대상에 대한 기본적인 정보를 포함하는 DID 문서와 연결되며, DID 문서의 신뢰성 보장 및 접근 용이성을 위해 블록체인 등 신뢰할 수 있는 저장소에 저장 관리된다. DID 문서는 식별 대상이 서명한 문서를 암호학적으로 검증 하거나 암호 통신 등을 수행하기 위한 공개키 정보와 식별 대상과 소통할 수 있는 서비스 엔드포인트(endpoint) 등의 정보로 구성된다.

3.1.3. DID 구성요소

- DID Document

DID에 의해 식별되는 주체인 DID Subject를 설명하는 JSON 형태의 데이터이다. ID, 공개키, 인증정보, 서비스 등을 담고 있으며 개인 정보는 저장되지 않는다. DID Subject는 DID Document를 통해 자신의 신원을 증명할 수 있다. DID Document에는 5가지 세부 항목이 있다.

1. ID: 해당 DID Subject에 대한 DID 식별값

2. Public Key: 인터랙션 과정에서 암호학적 증명이 필요한 경우 사용되는 키

3. Authentication: 인증 방식에 대한 정보

4. Services: DID subject와의 인터랙션을 위한 service-endpoints 정보를 확인

5. Context: 데이터가 어떤 값을 가지는지 명확한 정의를 해주는 역할이다. 통신과정에서 에러를 막기위해서 명확한 key-value 값을 정의한다. JSON-LD 문법을 사용한다.

- VC (Verifiable Credential)

VC는 운전면허증, 학위 증명서, 여권 등 현실 세계의 신분증을 디지털 세계에서 활용할 수 있는 방법을 제공한다. VC는 암호학적으로 안전하고 프라이버시를 보존하며 기계가 검증할 수 있는 방식으로 디지털 세상에서 필요한 다양한 종류의 크리덴셜을 제공할 수 있다. 이를 위해 VC는 물리적인 형태의 신분증과 같이 사용자에 대한 다양한 신원 속성(attribute)을 포함할 수 있으며, 디지털 서명을 이용해 VC 정보의 무결성을 검증할 방안을 제공한다.

VC 기반의 탈중앙화 신원증명 서비스는 발급자(issuer), 소유자(holder), 검증자 (verifier) 및 신뢰 저장소(verifiable data registry)의 역할(role)을 갖는 개체(entity)들로 구성된다. 발급자는 소유자에게 VC를 발급하고 소유자는 발급된 VC를 디지털지갑을 통해 관리하고 서비스 이용에

필요한 VP(Verifiable Presentation, 소유자가 검증자에게 전달하기 위해 VC를 이용해 생성된 데이터)를 검증자에게 제출한다. 검증자는 제출된 VP를 검증하기 위해 신뢰 저장소로부터 검증에 필요한 정보를 받아 확인하는 기본 절차를 갖는다.

- VP (Verifiable Presentation)

VP는 정보주체가 발급받은 검증 가능한 자격증명에서 검증기관이 요구하는 속성을 추출하여 생성한 제출용 증명서이다. 실생활에서 성인임을 증명할 때 우리는 주민등록증과 같은 신분증을 제출한다. 그러나 주민등록증에는 이름, 주민등록번호, 주소 등 다양한 개인정보가 포함되어 있다.

탈중앙화 신원증명 환경에서 검증 가능한 제공 ID 데이터 집합을 이용하면 주민등록증의 다양한 속성 중 주민등록번호에 기록된 생년 정보만을 추출하여 제출할 수 있다. 또한, 다수의 검증 가능한 자격증명을 결합하여 검증가능한 제공 ID 데이터 집합 형태로 표현하는 것도 가능하다. 필요한 정보만 선택하여 공개함으로써 자기주권신원을 행사할 수 있는 것이다.

3.1.4. 스마트 컨트랙트 (Smart Contract)

스마트 컨트랙트는 블록체인이 1세대에서 2세대로 넘어갈 수 있게 되는 가장 중요한 계기 중 하나로 블록체인 기술을 활용해 제3의 인증기관 없이 개인 간 계약이 이루어질 수 있도록 하는 기술이다.

계약상의 급부와 반대급부를 프로토콜화하여 소프트웨어 및 하드웨어에 미리 저장하고, 해당 계약을 이행하는 과정에서 조건 충족 여부에 대한 판단을 인간이 아닌 컴퓨터 등의 기계가 대신 실행함으로써 제3의 인증기관이 필요 없도록 하는 개념을 의미한다.

3.1.5. 하이퍼레저 패브릭 (Hyperledger Fabric)

하이퍼레저 패브릭은 다른 유명한 분산 원장이나 블록체인 플랫폼을 넘는 차별화된 핵심 능력을 전달하는 오픈 소스 엔터프라이즈급 허가형 분산 원장 기술이다. 패브릭은 높은 모듈성과 설정 가능한 아키텍처를 가지며 Java, Go 그리고 Node.js 같은 범용 목적의 프로그래밍 언어로 작성한 스마트 컨트랙트를 지원하는 최초의 분산 원장 플랫폼이다.

패브릭 플랫폼은 또한 허가형이다. 즉, 참여자들이 서로 알고 있다는 뜻이다. 이는 참여자들이 다른 누군가를 완전히 믿을 수 없더라도, 법적 동의 혹은 분쟁 조절의 틀 같은 참여자 간 어떤 형태로 존재하는 신뢰를 바탕으로 만들어진 거버넌스 모델 하에서 네트워크를 운영할 수 있다.

또다른 특징 중 하나는 플랫폼을 특정 요구에 맞춰 더 효율적으로 커스텀 할 수 있도록 하는 플러그인 가능한 합의 프로토콜이다. 예를 들어, 하나의 기업 내에 배포할 때 혹은 신뢰 기구에 의해 운영될 때, 완전 비잔틴 장애 허용 합의(Fully byzantine fault tolerant consensus)는

불필요하고, 성능과 처리량을 저하시킨다고 판단할 수 있다.

그와 같은 상황에서는, 크래시 장애 허용(CFT: Crash Fault-Tolerant) 합의 프로토콜이 더 적절하지만, 여러 단체의 탈중앙화된 유스 케이스에서는 더 전통적인 비잔틴 장애 허용(BFT: Byzantine Fault Tolerant) 합의 프로토콜을 적용할 수 있다.

3.1.6. 하이퍼레저 인디 (Hyperledger Indy)

하이퍼레저 인디는 블록체인 또는 기타 분산 원장에 기반을 둔 디지털 신원을 제공하기 위한 툴킷, 라이브러리 등을 제공한다. 사용자 신원정보 보호를 위해 선택적 공개와 영지식 증명을 하고, Idemix 프로토콜에 기반한 Indy-anoncreds를 사용하여 자격증명(Crednetial)의 주장(Claim)을 보호한다.

암호화 관련 API를 제공하는 하이퍼레저 울사(Hyperledger Ursa)의 Bulletproof 방법, Idemix 기반한 방법의 두 방식의 영지식 증명 방법을 사용할 수 있다. 보안과 암호화 관련하여 crypto, 선택적 정보제공, 대칭키 암호화 등 여러 방법을 결합하여 보안에 주의를 두고 있다.

3.2. 개발 환경

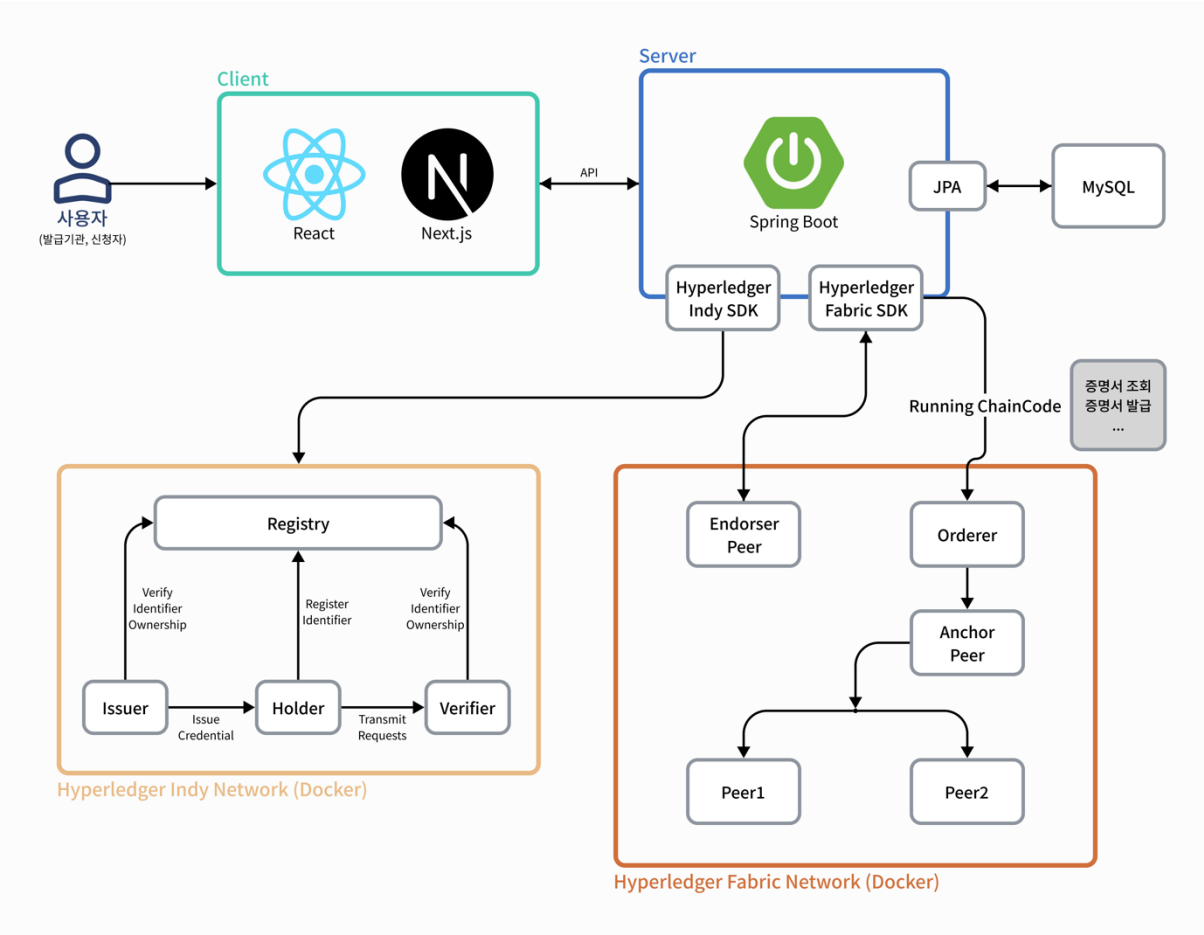
3.2.1. 개발 도구

분류	도구 이름
블록체인	- Hyperledger Indy, Hyperledger Fabric
Frontend	- React 18, Next.js 14
Backend	- Spring Boot 3.2.3
Docs	- Swagger, MkDocs

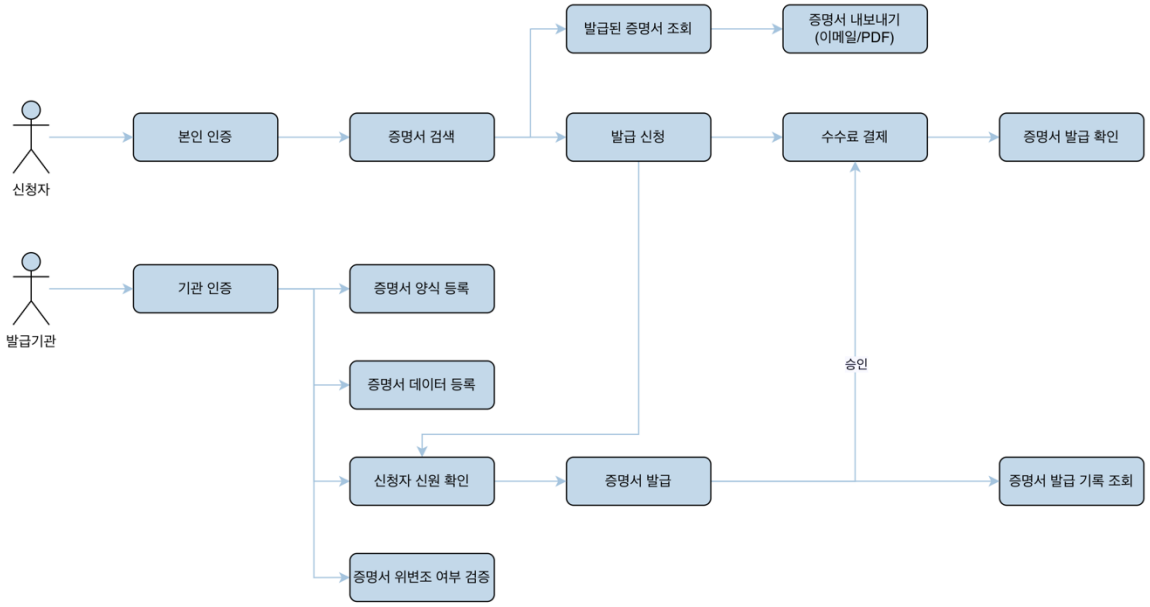
3.2.2. 소프트웨어

이름	상세 설명
Docker	- 블록체인 네트워크 구축
yarn	- 패키지 관리
MySQL	- JWT 등 웹 클라이언트 관련 데이터베이스
git	- 형상 관리

3.3. 시스템 구성


Client
Server
사용자
(발급기관, 신청자)
React
Next.js
API
Spring Boot
JPA
MySQL
Hyperledger Indy SDK
Hyperledger Fabric SDK
Running ChainCode
증명서 조회
증명서 발급
...
Registry
Verify Identifier Ownership
Register Identifier
Verify Identifier Ownership
Issuer
Issue Credential
Holder
Transmit Requests
Verifier
Hyperledger Indy Network (Docker)
Endorser Peer
Orderer
Anchor Peer
Peer1
Peer2
Hyperledger Fabric Network (Docker)

3.4. 서비스 흐름도



4. 개발 일정 및 역할 분담

4.1. 개발 일정

	5월					6월				7월				8월					9월				10월		
업무	1주	2주	3주	4주	5주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	5주	1주	2주	3주	4주	1주	2주	3주
서비스 기획																									
요구사항 분석																									
착수 보고서 작성																									
웹 클라이언트 디자인																									
블록체인 네트워크 구축																									
체인코드 개발																									
서버 환경 구축																									
웹 클라이언트 개발																									
중간 보고서 작성																									
API 연동																									
테스트 및 디버깅																									
결과 보고서 작성																									
발표 포스터 제작																									

4.2. 역할 분담

이름	역할
공통	- 체인코드 개발 - 테스트 및 디버깅, 보고서 작성
문석준	- 블록체인 네트워크 구축 - CI & CD 작업
이현빈	- 웹 클라이언트 기능 개발 - 웹 클라이언트 디자인
진예규	- 서버 환경 구축 - 백엔드 개발