

# DID를 활용한 대학교 증명 명 발급 및 제출 시스템



**대학교 증명**  
**발급 서비스**

조운형  
김영민  
이재영

# '신원(Identity)'의 정의

Today, usually just "proof" of something about ourselves

현대인들이 사용하는 신원 인증 수단 : 신분증, 면허증 (offline)  
공인인증서, id/pw, ldp (online)

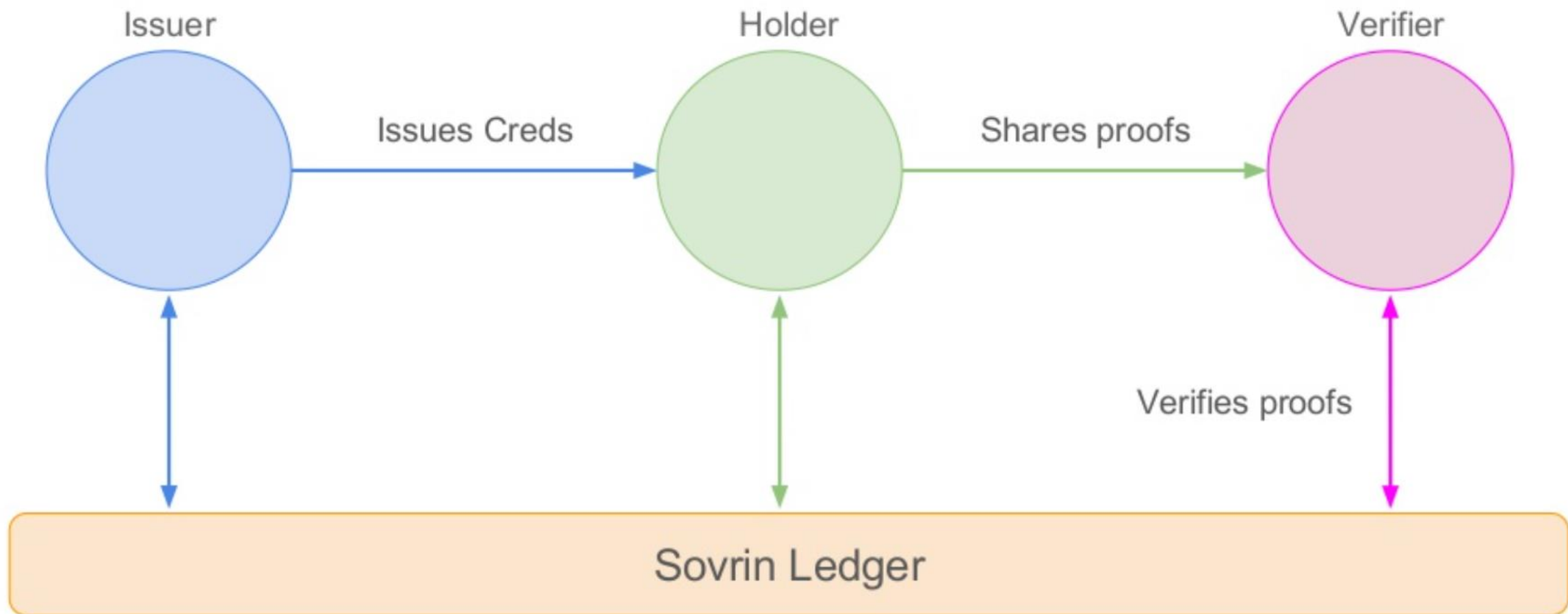
# 현재 신원 인증 방식의 문제점

- 중앙 집중된 신뢰기관에 발급 및 인증을 의존함
- 위조 및 변조가 가능함
- 발급 받는데 비용이 상대적으로 많이 들어감
- 인증 과정에서 필요한 정보보다 많은 정보를 공개하게 됨

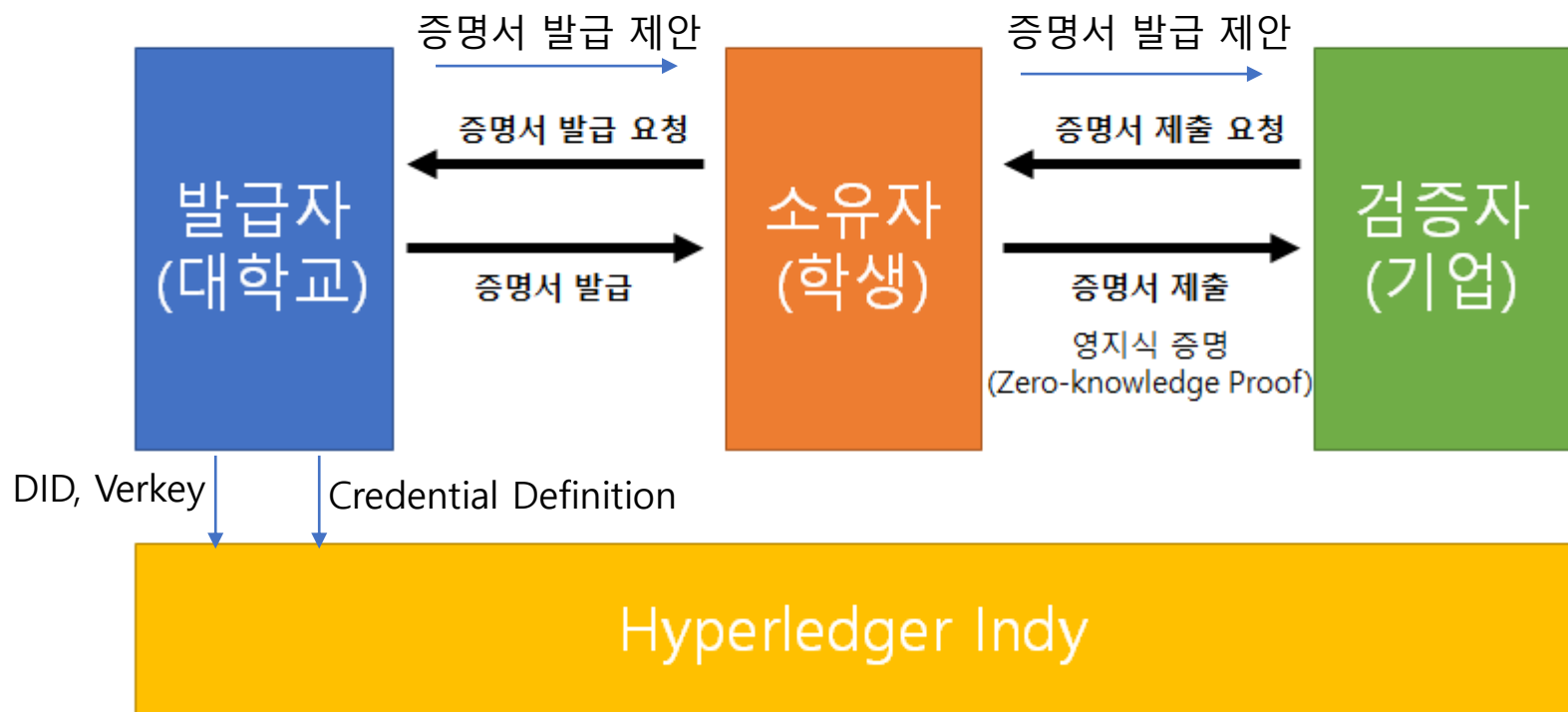
# DID의 정의

중앙 시스템에 의해 통제되지 않으며 개개인이 자신의 정보에 완전한 통제권을 갖도록 하는 기술

# DID를 활용 신원 인증 시스템 구성요소



# 프로젝트 개요



# Hyperledger Indy



- Distributed Ledger Software
- 신원인증을 위한 분산원장 개발에 특화된 Hyperledger Indy를 활용하여 ledger의 node 생성
- 개발을 위한 여러가지 도구/라이브러리/reusable components 를 제공

# Hyperledger Aries



- Library
- DID 생성, 전송 및 저장하는데 중점을 둔 재사용 가능하고 상호 운용 가능한 공유 키트 제공
- 블록체인 기반의 사용자 간의 P2P 상호 작용 지원
- Hyperledger Ursa를 이용한 암호화 기능 제공



# 프로젝트 결과물

- Hyperledger Indy 노드 구축
- 대학교의 증명서 발급 사이트 및 이용자와의 구현
- 이용자의 증명서 저장 앱 구현
- 기업의 증명서 검증 사이트 및 이용자와의 구현
- Demo Video

# Hyperledger 노드 구축

- Local machine에서 Docker을 이용해 4개의 node 생성, Hyperledger Indy 시작
- Trustee DID 생성(genesis), Steward DID 생성(genesis)
- <https://github.com/hyperledger/indy-node>

### Validator Node Status

Node1	DID: 6w6pDLhcBco0esN72qfotTgFa7cbugZpkX3Xo6pLhPhv Uptime: 14 hours, 47 minutes, 42 seconds Txns: 0 config, 17 ledger, 4 pool, 0.0106/s read, 0.000225/s write indy-node version: 1.12.3
Node2	DID: 8ECvSk179mjsjKRLWiQtssMLgp6EPHwXtaYyStWPSGAb Uptime: 14 hours, 47 minutes, 42 seconds Txns: 0 config, 17 ledger, 4 pool, 0.0108/s read, 0.000225/s write indy-node version: 1.12.3
Node3	DID: DKVx62fXTU8yT5N7hGEbXB3dfdAnYv1JczDUHpmDxya Uptime: 14 hours, 47 minutes, 42 seconds Txns: 0 config, 17 ledger, 4 pool, 0.0107/s read, 0.000225/s write indy-node version: 1.12.3
Node4	DID: 4PS3ED03dW1tci1Bp6543CfuuebJFrg36kLAUcsgfAA Uptime: 14 hours, 47 minutes, 42 seconds Txns: 0 config, 17 ledger, 4 pool, 0.0108/s read, 0.000225/s write indy-node version: 1.12.3

View detailed information about the status of the running validator nodes:

### Connect to the Network

Download the genesis transaction file to connect to the network.

[Genesis Transaction](#) JSON

### Authenticate a New DID

Easily write a new DID to the ledger for new identity owners.

☒ Register from seed ☐ Register from DID

Wallet seed (32 characters or base64)

DID (optional)

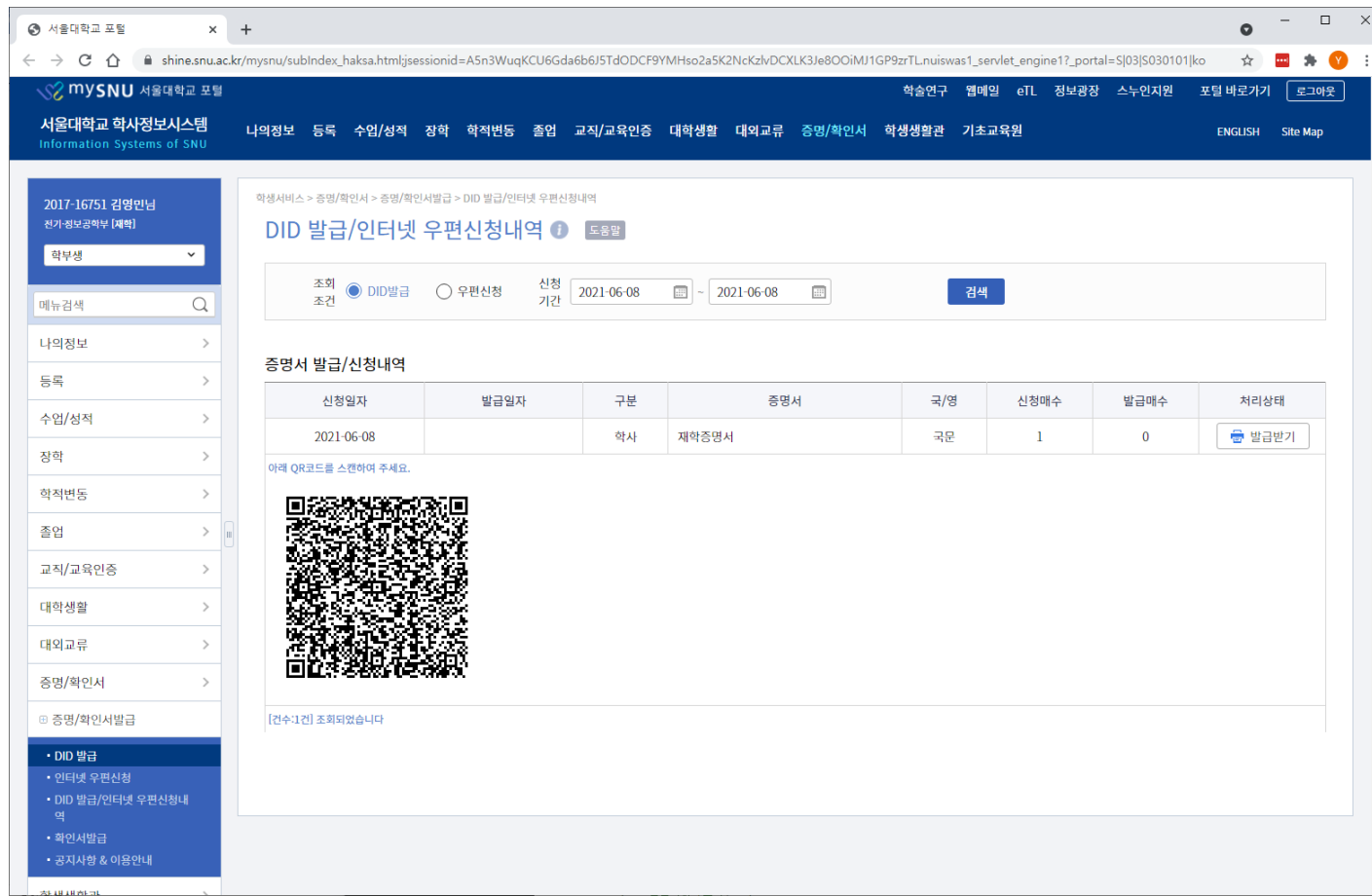
Alias (optional)

Role

Endorser

# 대학교 증명서 발급 사이트 구현

- 마이스누 증명서 발급 페이지를 직접 수정할 수 없어, 크롬 확장 프로그램 형태로 (클라이언트 사이드) 구현



- Aires cloud agent 사용
- <https://github.com/blockchain-project-f/aries-cloudagent-python>

# 이용자 증명서 앱 구현

- Aries Mobile Agent React Native 를 사용하여 구현 (Android에서 테스트 완료)
- UI는 Demo Video에서 확인 가능
- <https://github.com/blockchain-project-f/aries-mobile-agent-react-native>

# 기업의 증명서 검증 사이트 구현

- Create React App을 이용하여 구현

The screenshot shows a web browser window with the URL 'localhost:8000'. The page is titled 'NAVER Cloud Career' and has a navigation bar with links for '개발' (Development), '엔지니어' (Engineer), '사업/경영지원/디자인' (Business/Operations Support/Design), and '상시Pool' (Permanent Pool). The main content area is titled '지원서 작성하기' (Apply for Job) and contains a form for '[NAVER Cloud] Blockchain 서비스 개발 경력사원 모집' (Recruitment of Experienced Staff for Blockchain Service Development at NAVER Cloud). The form includes fields for '이름' (Name) with the value '김영민', '생년월일' (Date of Birth) with the value '1998.03.11', and '이메일' (Email) with the value 'y@y/lem.kim'. There are also dropdown menus for '성별' (Gender) and '지원경로' (Application Path), and a search bar for '연락처' (Contact Information). A QR code is displayed under the heading '재학증명서 필수' (Must provide proof of enrollment). Below the QR code, there is a note: '• 위 QR코드를 스캔하여 DID 재학증명서를 제출하여 주십시오.' (Please scan the QR code to submit your DID proof of enrollment certificate). At the bottom, there is a section for '경력사항 필수' (Must provide work experience) with a table for listing previous employers.

- Aires cloud agent 사용
- <https://github.com/blockchain-project-f/aries-cloudagent-python>

# 프로세스

(1) 1. Hyperledger Indy 시작

- Trustee DID 생성 (genesis)
- Steward DID 생성 (genesis)

(2) 2. 대학교 증명 발급

- 지갑 생성 및 DID 등록
- 재학증명서 schema 생성 및 등록
- 재학증명서 credential definition 생성 및 등록

(3) 3. 이용자 앱

- 지갑 생성 및 DID 등록

(4) 4. 제출자 사이트

- 지갑 생성 및 DID 등록

# 프로세스

## (5) 2. 대학교 증명 발급

- Invitation 생성 및 QR코드 표시 (Pairwise-unique DID 사용)

## (6) 3. 이용자 앱

- 표시된 QR코드 스캔
- WebSocket을 사용하여 연결 생성\*\*

## (7) 2. 대학교 증명 발급

- 재학증명서 제안

## (8) 3. 이용자 앱

- Ledger로부터 credential definition 확인
- 발급 요청

## (9) 2. 대학교 증명 발급

- 재학증명서 발급

## (10) 3. 이용자 앱

- 재학증명서 지갑에 저장

# 프로세스

## (11) 4. 제출자 사이트

- Invitation 생성 및 QR코드 표시 (Pairwise-unique DID 사용)

## (12) 3. 이용자 앱

- 표시된 QR코드 스캔
- WebSocket을 사용하여 연결 생성\*\*

## (13) 4. 제출자 사이트

- 서울대학교 사이트로부터 서울대학교 DID 취득\*\*
- 서울대학교 DID로부터 재학증명서 credential definition 확인
- 증명 요청 (필요한 항목 지정)

## (14) 3. 이용자 앱


- 지갑으로부터 적절한 재학증명서 선택
- 증명 생성

## (15) 4. 제출자 사이트

- 증명 검증



# Demo Video



대학교중앙도서관

## 암호

암호를 입력해주세요:

---

확인

Q&A