모바일신분증을 활용한

2025 블록체인 & AI 해커톤 가이드북

2025.04.30.

Alpha

INDEX

01 개요

1-1. 배경 및 목적
1-2. 대회 일정
1-3. 심사 기준
1-4. 상금 및 특전
1-5. 참고 프로젝트

1-1. 배경 및 목적

추진배경

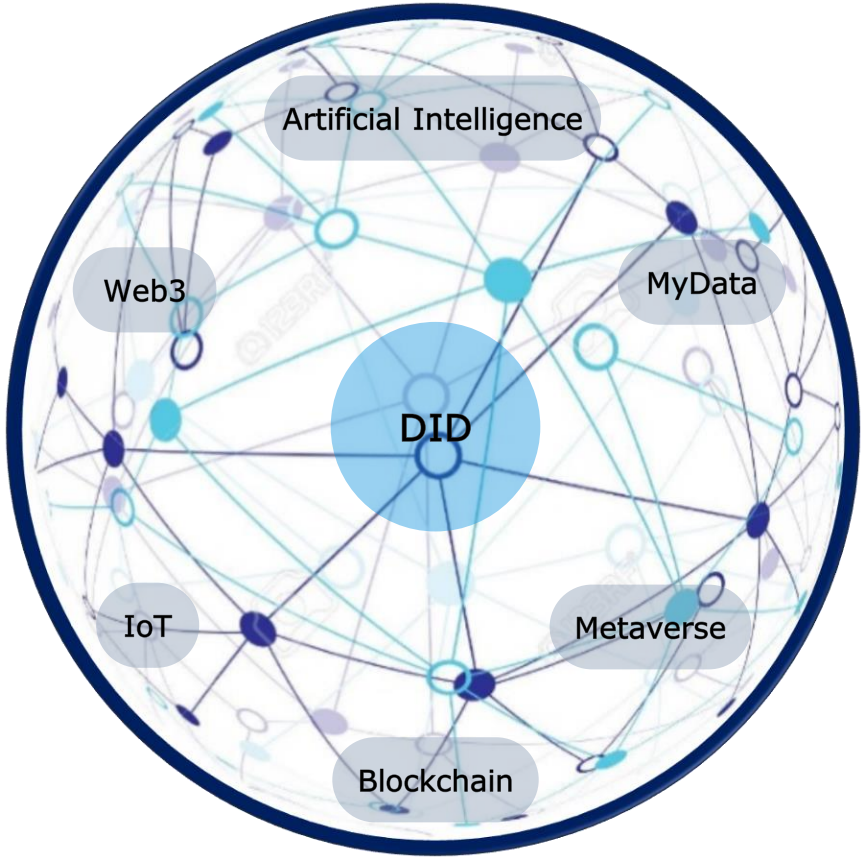
블록체인 & AI 생태계 확장을 위한 비즈니스 아이디어 및 스타트업 발굴, 육성

추진목표

- 모바일주민증의 대중화 및 사용처 확대
- 블록체인&AI 비즈니스 생태계 확장
- 스타트업 발굴 및 글로벌 진출 지원

추진방향

- 대학(원)생 및 창업 희망자 대상 해커톤 추진
- 모바일 신분증 기반 서비스와 비즈니스 모델 발굴
- 최소 MVP* 모델 구현 및 서비스화
- 입상팀 대상 창업 및 글로벌 진출 지원



※ MVP: Minimum Viable Product, 최소한의 기능을 구현한 제품

1-2. 대회 일정

01 모집 25.05.01(목) ~ 05.28(수)

- 해커톤 참가를 위해 **트랙(Track1~2)을 선택** 후 지원서 구글폼 제출

  ※ 링크: https://forms.gle/U7GpEDTRB25emhm8A

필수조건	모바일 주민등록증을 활용한 제품 / 서비스 제안
Track 1	**비즈니스 아이디어** → 제품 / 서비스 아이디어를 도출, 발표
Track 2	**서비스 모델 개발** → 도출된 아이디어를 고도화, MVP 모델 개발, 데모 시연

※ Open DID(Appendix1. 참조) 활용 또는 옴니원메인넷(Appendix2. 참조) 활용 시 각 5% 가산점 부여

- **팀 리더 1명과 최대 4명의 팀원**으로 팀 구성 가능(팀당 5명)

  ※ 시리즈 투자 유치 실적 없는 창업 3년 이내 스타트업 또는 개인자격 참여 가능

- 이해도를 높이기 위한 오프라인 기술 설명회 제공 예정

  ※ 25.05.15.(목) 16:00 ~ 17:40, 여의도 파크원 타워2

02 예선 25.06.01(월) ~ 06.13(금)

- 설명회 이후 아이디어 고도화
- 산출물 온라인 제출 (PPT 양식 PDF 포맷 10매 내외)
- 예선 심사 대상 20팀 선정 및 **온라인/개별 발표(06.30.(월))**

03 예선심사 25.06.25(수) ~ 06.26(목)

- 예선 대상 20팀 대면 심사 진행 (여의도 파크원 타워2, 48F)
- 팀당 10분 발표(프레젠테이션)
- 발표 자료 PPT 양식 10매 내외
- **결선 진출 대상 10팀 선정 및 온라인/개별 발표(06.30(월))**

04 결선 25.07.03(목) ~ 09.16(화)

- 오리엔테이션(7월 3일) 및 기술지원 멘토링 운영
- 아이디어 고도화 및 MVP* 개발 산출물 온라인 제출

※ MVP: Minimum Viable Product, 최소한의 기능을 구현한 제품

05 결선심사 및 시상 25.09.23(화)

- 결선 대상 10팀 대면 심사 진행 (코엑스 컨퍼런스룸, 3F)
- 팀당 15분 발표(프레젠테이션, Track 2: 데모시연 포함)
- 입상팀 시상(시큐업세미나 폐막식)

1-3. 심사 기준

01

창의성

아이디어의 독창성,
참신성, 차별성

02

실현 가능성

규제(프라이버시)를
고려한 기술적 구현
가능성, 운영 지속 가능성

03

사업성

시장 경쟁력,
사업화 가능성,
기획의 타당성과 논리성

04

완성도 및 협업도

성실성, 문제 해결을
위한 능동적인 자세
및 협업 능력

1-4. 상금 및 특전

상금

순위	상명	상금
대상(1팀)	행정안전부 장관상	1,500 만원
최우수상(1팀)	한국조폐공사 사장상	600 만원
우수상(3팀)	한국지능정보사회진흥원(NIA) 원장상, 정보통신산업진흥원(NIPA) 원장상, 한국인터넷진흥원(KISA) 원장상	300 만원
총 5팀		**3,000 만원**

임상팀 특전

창업지원금 10억 지원

입상팀 대상으로 별도 심사를 거쳐
창업지원금 지원

창업 패키지 지원

창업 공간, 경영 컨설팅, 인프라 제공 등을 통해
초기 스타트업 안정적 정착 유도

글로벌 시장 진출 지원

해외 투자 유치 기회 제공 및
해외 네트워크 및 파트너십 구축 지원

1-5. 참고 프로젝트

2019 민간 시범사업

탈중앙화 기부 플랫폼

(수행기업) (주)이포넷, 두나무 주식회사, 어린이재단
(주)이노블록

#시범사업 #사회복지 #민간
#데이터 이력관리 및 위 · 변조 방지

2020 민간 시범사업

DID 기반 디지털 화물 운송장 플랫폼

(수행기업)네이버시스템(주), LG CNS,
사단법인 대한교통학회, 화물복지재단

#시범사업 #물류 · 유통 #민간
#DID신원인증 · 자격증명 #데이터이력관리 및 위변조방지

2021 민간 시범사업

DID 기반 전자주총 서비스

(수행기업) 한국전자투표, 코스톰

#시범사업 #법률 · 회계 · 계약 #민간
#DID신원인증 · 자격증명

2023 민간 시범사업

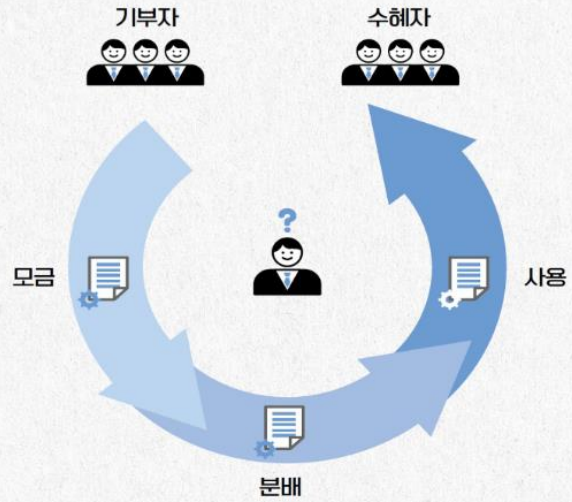
WEB3 신원인증 기반 NFT 발행사업

(수행기업) NICE평가정보, 시큐차트글로벌(주),
핵슬란트

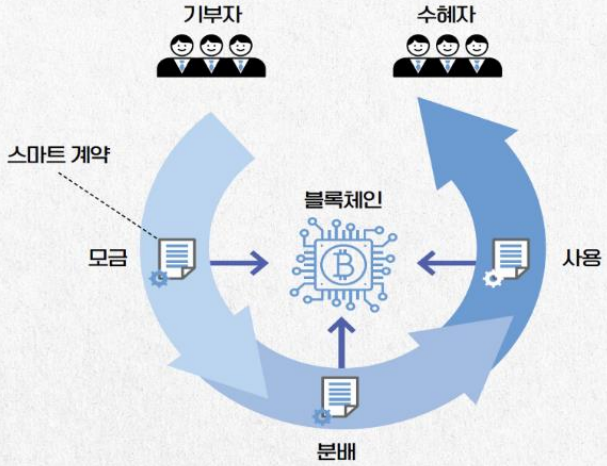
#시범사업 #물류유통 #민간 #DID신원인증 · 자격증명
#데이터 이력관리 및 위변조방지

1-5. 참고 프로젝트 ① 탈중앙화 기부 플랫폼

AS-IS



TO-BE

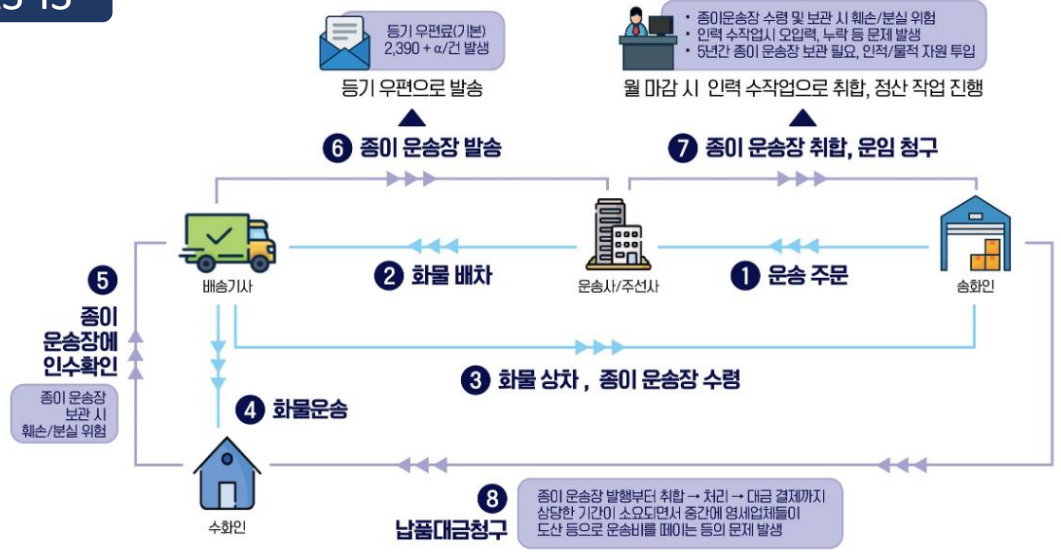


블록체인 도입 이유

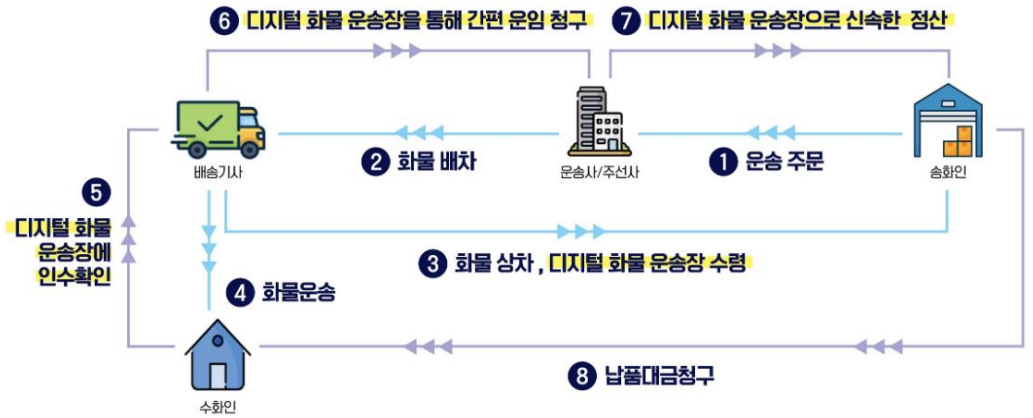
- (기부자) 기부하여 획득한 영향 지수를 토대로 기부 단체 평가 및 기부금 집행 여부 결정 등 기부금 운영에 중요한 역할 수행 가능
- (기부단체) 스마트계약을 활용하여 목적한 바에 맞는 올바른 기부금 집행으로 많은 수혜자에게 혜택 제공 가능한 서비스 운영 가능
- (수혜자) 수혜자는 기부자가 자신의 기부내역을 확인 할 수 있도록 수령인 및 사용 내역을 블록체인에 기록하여 투명한 기부금 집행 보장

1-5. 참고 프로젝트 ② DID 기반 디지털 화물 운송장 플랫폼

AS-IS



TO-BE



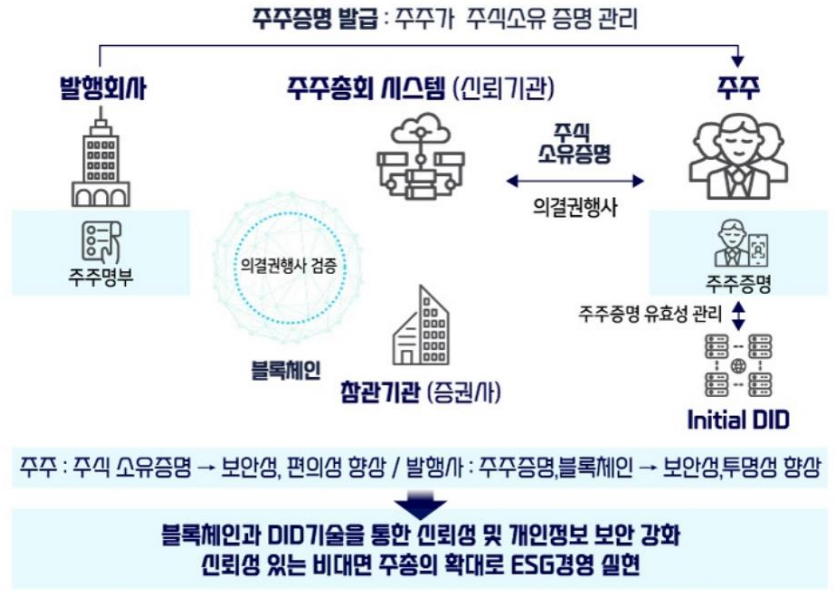
블록체인 도입 이유

- 배차-상차-운송-정산 과정을 블록체인에 기록하고 블록체인 원장을 각 운송 주체별로 공유함으로 화물운송 중 문제가 발생하였을 때 책임 소재를 분명하게 할 수 있음
- 전자운송장에 DID 서명으로 화물의 명확한 인수인계 증빙 가능

1-5. 참고 프로젝트 ③ DID 기반 전자주총 서비스

AS-IS

TO-BE

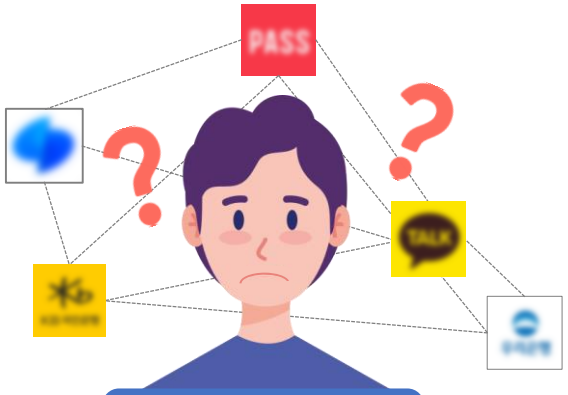


블록체인 도입 이유

- 주주는직접 주주총회 소집 통지서를 들고 주주총회 장소에 가지 않고도 개인정보를 강화한 DID 주주 증명을 사용하여 온라인으로 의결권 행사 가능
- 발행회사는 주주를 대상으로 전자 주주총회 진행과 전자 위임장 권유가 가능하며, 온라인 의결권 행사 시 DID 주주 증명과 블록체인을 활용함으로써 보안성 강화와 신뢰성 강화 가능

1-5. 참고 프로젝트 ④ WEB3 신원인증 기반 NFT 발행사업

AS-IS



Web2 환경

분산된 개인정보,
각기 다른 ID, Password
개인의 정보 침해 가능성▲

TO-BE

필요한 개인정보만 선택적 공유 가능

Web3를 이용한 인증
"NFT" 로 컨트롤

Web3 환경

개인정보의 주권을 가진 개인,
각기 다른 ID, Password 인증해결
개인의 정보 침해 가능성▼

블록체인 도입 이유

- 중앙 시스템에 의존하지 않고 개인이 자신의 정보를 관리할 수 있는 탈중앙화 기술로 디지털 월렛에 개인정보를 담아 필요할 때 개인키를 입력해 본인의 신원을 증명
- 개인정보 오남용, 유출 ,노출 방지, 프라이버시 강화, 디지털 신분증 형태의 온 · 오프라인을 통합하여 신원자격 / 자격증 / 사원증 등의 신원증명을 간편하게 구축 가능

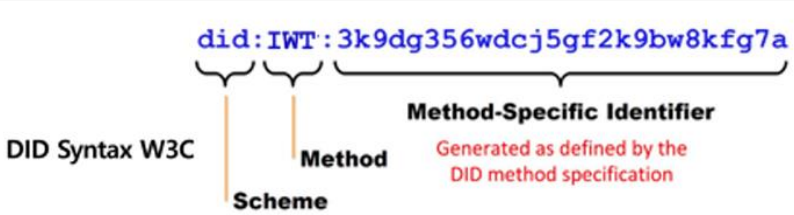
02 DID 기초 개념

2-1. Decentralized Identifier & DID Document
2-2. Verifiable Credentials & Verifiable Presentation
2-3. eWallet & Trust Repository

2-1. Decentralized Identifier & DID Document

DID (Decentralized Identity, 분산 ID)

- 탈중앙화된 신원 정보
- 자기 주권 신원 (SSI, Self-Sovereign Identity) - 이용자가 스스로 개인의 정보를 통제
- DPKI (분산 PKI) 기반, 엔티티는 DID로 식별되며 증명을 통해 인증될 수 있다.
- DID Syntax W3C



- DID Document

2-2. Verifiable Credentials & Verifiable Presentation

VC (Verifiable Claim, Verifiable Credentials)

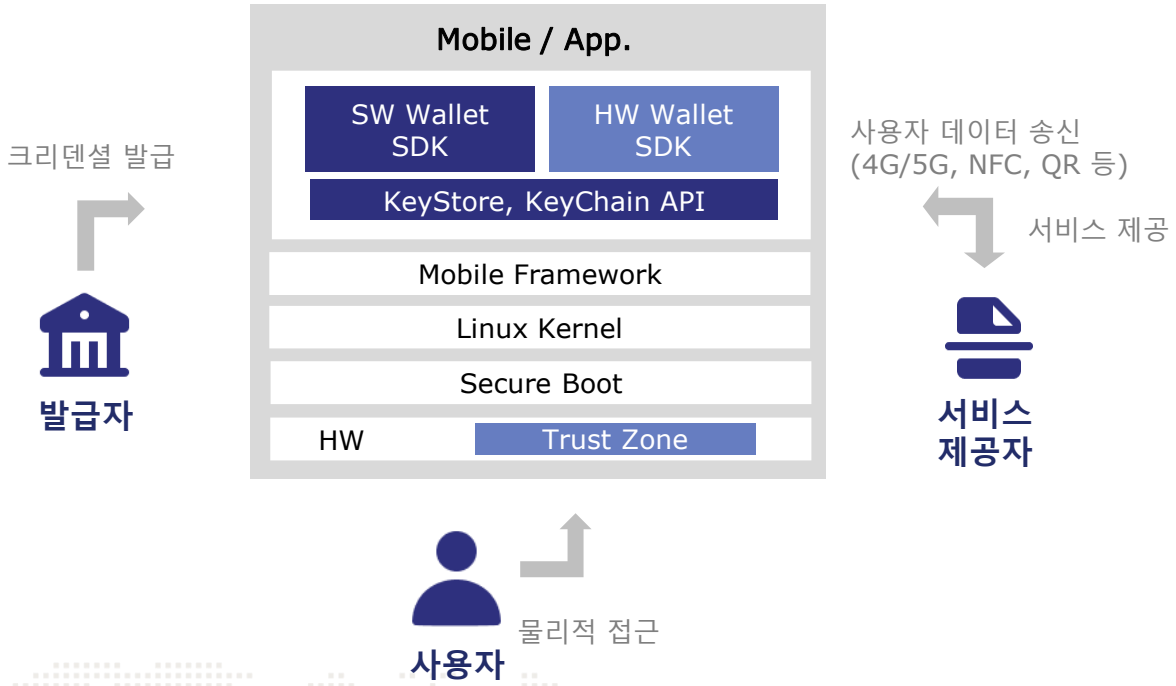
- **검증 가능한 자격 증명**
- **ID 데이터 또는 클레임들과 발급자를 암호학 적으로 검증할 수 있는 메타 데이터의 집합**
- ISSUER 가 발급한 User 에 대한 설명, 암호학적 검증 ex) 신분증, 증명서 등
- 해당 자격의 위변조 및 발급기관 발행 여부 등을 블록체인을 통해 검증 가능하도록 만든 형태
- 공개키 인증서를 이용하여 "자격 증명 검증 " 을 가능하게 함

VP (Verifiable Presentation)

- **검증 가능한 제출 정보**
- **신분 또는 자격을 설명하는 정보에 대한 일부분의 조합**
- 검증자가 VP 를 생성한 소유자(Holder) 를 검증할 수 있는 방식으로 구성
- VC를 바로 제출 또는 VC로부터 일부만 추출하여 제출

수준 1	수준 2		필수	형식	값	설명
이름	인덱스	이름				
service_code			Y	string	"${code-service}"	서비스 코드
service_name			Y	string	"${name-service}"	서비스 이름
service_desc			Y	string	"${desc-service}"	서비스 설명
sp_did			Y	DID	"${did-verifier}"	검증기관의 DID
req_claims			Y	[ ]		제출 요청 정보
	[0]		N	string	"${claim}"	
	[...]		N	string	"${claim}"	
	[n]		N	string	"${claim}"	
allowed_vctype_codes			Y	[ ]		제출 가능한 VC 유형
	[0]		N	string	"${vctype}"	
	[...]		N	string	"${vctype}"	
	[n]		N	string	"${vctype}"	
allowed_issuer_dids			Y	[ ]		제출 가능한 VC 발급기관
	[0]		N	DID	"${did-issuer}"	
	[...]		N	DID	"${did-issuer}"	
	[n]		N	DID	"${did-issuer}"	

2-3. eWallet & Trust Repository



구분	전용월렛	안전지갑(지갑24)
단말유형	• 비삼성 AOS 휴대폰 • 안전지갑 미지원 삼성전자 휴대폰 • iOS 아이폰	안전지갑 지원 삼성전자 휴대폰
VC 저장 위치	• SW 안전영역	HW 안전영역(TEE)
Key 저장 위치	• KeyStore, KeyChain	HW 안전영역(TEE)

03 모바일 신분증

3-1. 모바일 신분증 소개

3-2. 모바일 신분증 구조

- Wallet 및 CA, SP Provider 역할별 구성
- 제출 모드에 따른 인터페이스
- 제출모드별 참조 API

3-3. 모바일 신분증 활용 방법 : OmniOne CX 서비스 연계

- ① 표준인증창 호출방식
- ② Restful API 연동방식
- 방식별 검증 데이터(JWT) 파싱
- 적용사례 | 정부24

3-1. 모바일 신분증 소개

급속하게 변화하는 디지털 환경에 발맞춰 정부에서 구축한 국가 디지털 신분증 플랫폼

CONCEPT

국가 신분증이
개인의 스마트폰에
암호화되어 **안전하게 저장**

VALIDITY

어느 곳이라도
실물 신분증과 같은
법적효력으로 **편리하게 사용**

CHARACTERISTICS

DID 및 블록체인 등
최신 ICT기술을 통해
신뢰할 수 있는 서비스를 보장

3-1. 모바일 신분증 소개

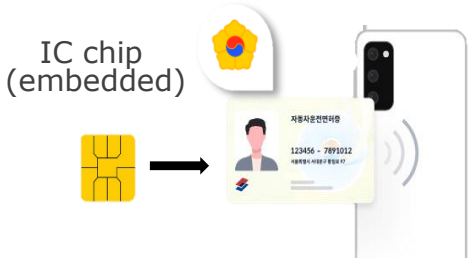
모바일 신분증은 1인 1단말 1디지털지갑으로 사용자별 1개의 DID만 발급하고
1VC 정책으로 모바일 디바이스별 1개의 모바일 신분증만 발급 및 유효

대면 신원확인

발급기관 담당자가
발급 신청자의 신원을
대면으로 확인

모바일 신분증 발급절차

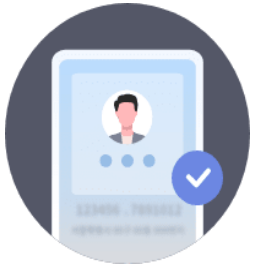
01



실물 IC신분증 확인

발급 신청자의
실물 신분증이 맞는지
1차 확인

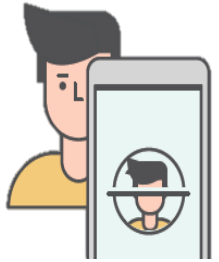
02



개인정보 확인

주민등록번호와
전화번호 등 개인정보
2차 확인

03

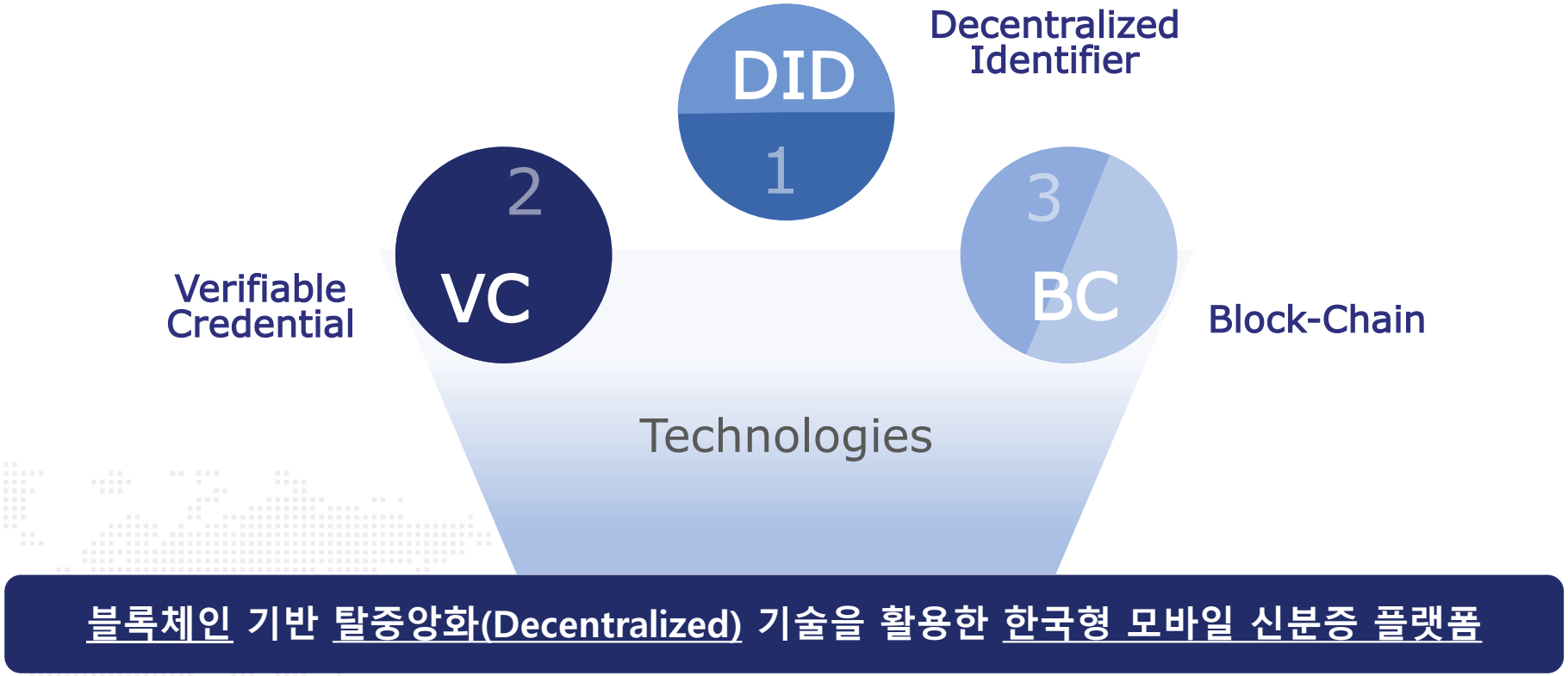


생체정보 확인

생체인식 정보를
통해 본인여부
3차 확인

3-1. 모바일 신분증 소개

모바일 디지털지갑에 국가ID 신분증(VC)를 암호화하여 저장하며,
이용처 제출 시 정부 DID 블록체인을 통하여 신원증명이 가능한 자기주권 신원증명체계



3-1. 모바일 신분증 소개

22년 모바일 운전면허증을 시작으로,
25년 현재, 모바일 주민등록증, 외국인등록증 등 발급 진행 중

모바일 운전면허증 (2022)

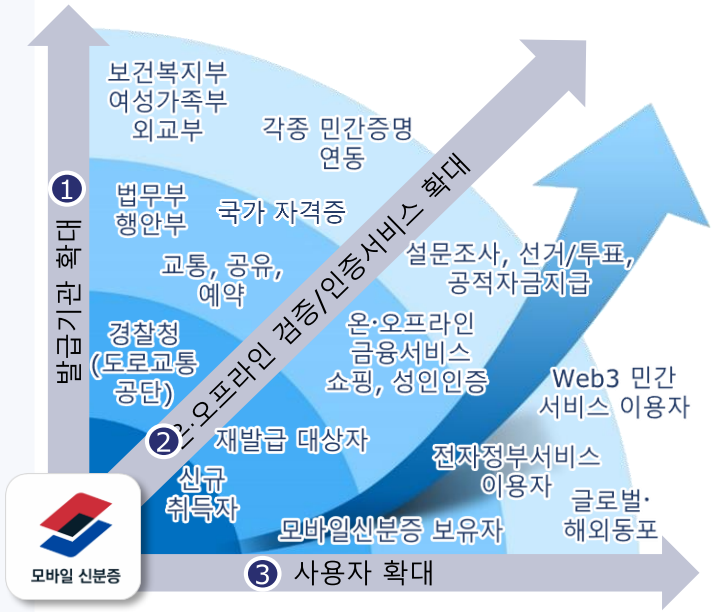
모바일 국가보훈등록증 (2023)

재외국민신원확인증 (2024)

모바일 주민등록증 (2024)

모바일 외국인등록증, 국내거소신고증, 영주증 (2025)

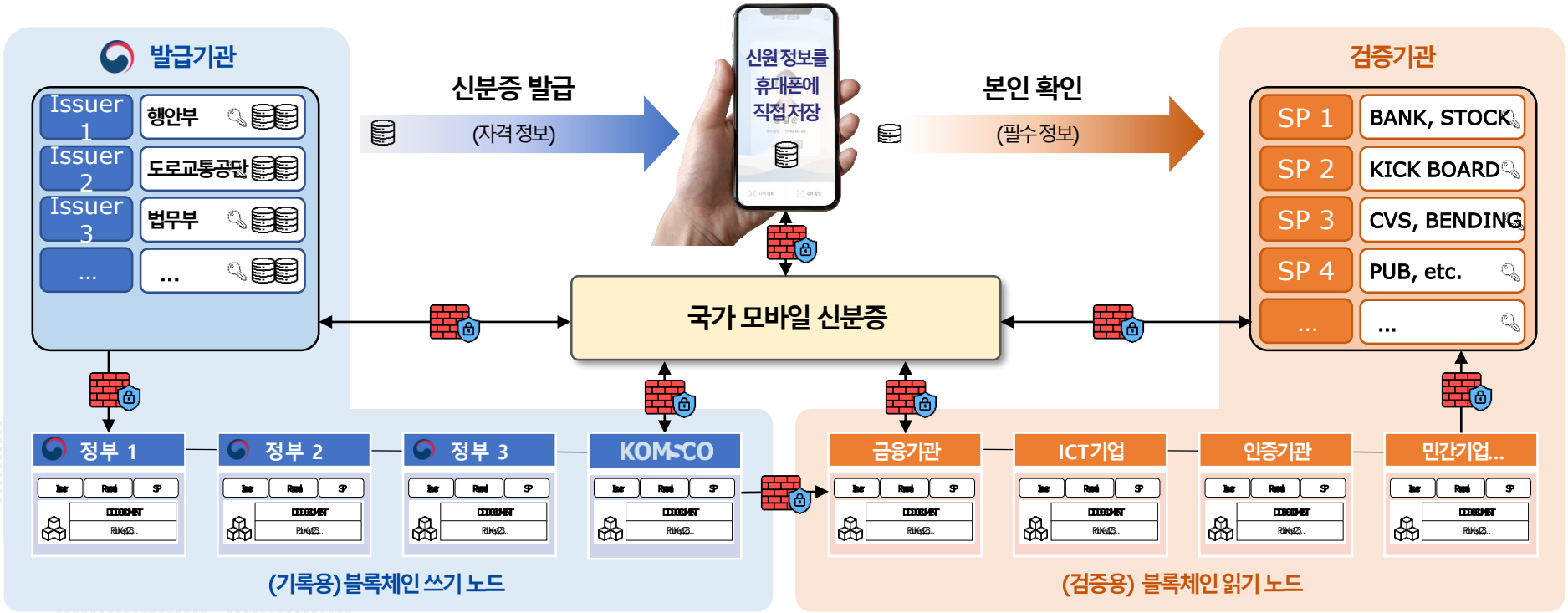
모바일 국가기술자격증 (2025)



* 모바일 장애인등록증 2025년 하반기 서비스 예정

3-1. 모바일 신분증 소개

모바일 신분증은 실물 플라스틱 신분증과 법적 효력이 동일,
온·오프라인 통합 신원인증 수단



3-1. 모바일 신분증 소개

01 DID Decentralized Identifier 기술 특징

발급 시

발급기관으로부터
개인정보를 개인 스마트폰에 저장

활용 시

CID* 와 달리 중앙서버 접근 없이
스마트폰에서 정보제공하여 개인신원 확인

* CID (Centralized ID, 중앙집중식 신원증명)
온라인 상에서 중앙서버에 연결하여 신원 확인

3-1. 모바일 신분증 소개

02 VC Verifiable Credential 기술 특징

발급 시

데이터를 옮기는
컨테이너(VC)에 자격정보를 담아
스마트폰에 저장

활용 시

VC의 필수 자격 정보만 컨테이너(VP)에
담아 SP(서비스 제공자)에 제출

* Self - Sovereign Identity (SSI): 개인이 신분증 발급을 요청하고 자신의 개인정보를 직접 관리하는 개념으로, 이는 DID(분산 신원 증명) 및 VC(검증 가능한 자격 증명) 기술로 구현

3-1. 모바일 신분증 소개

03 BC Blockchain 기술 특징

발급 시

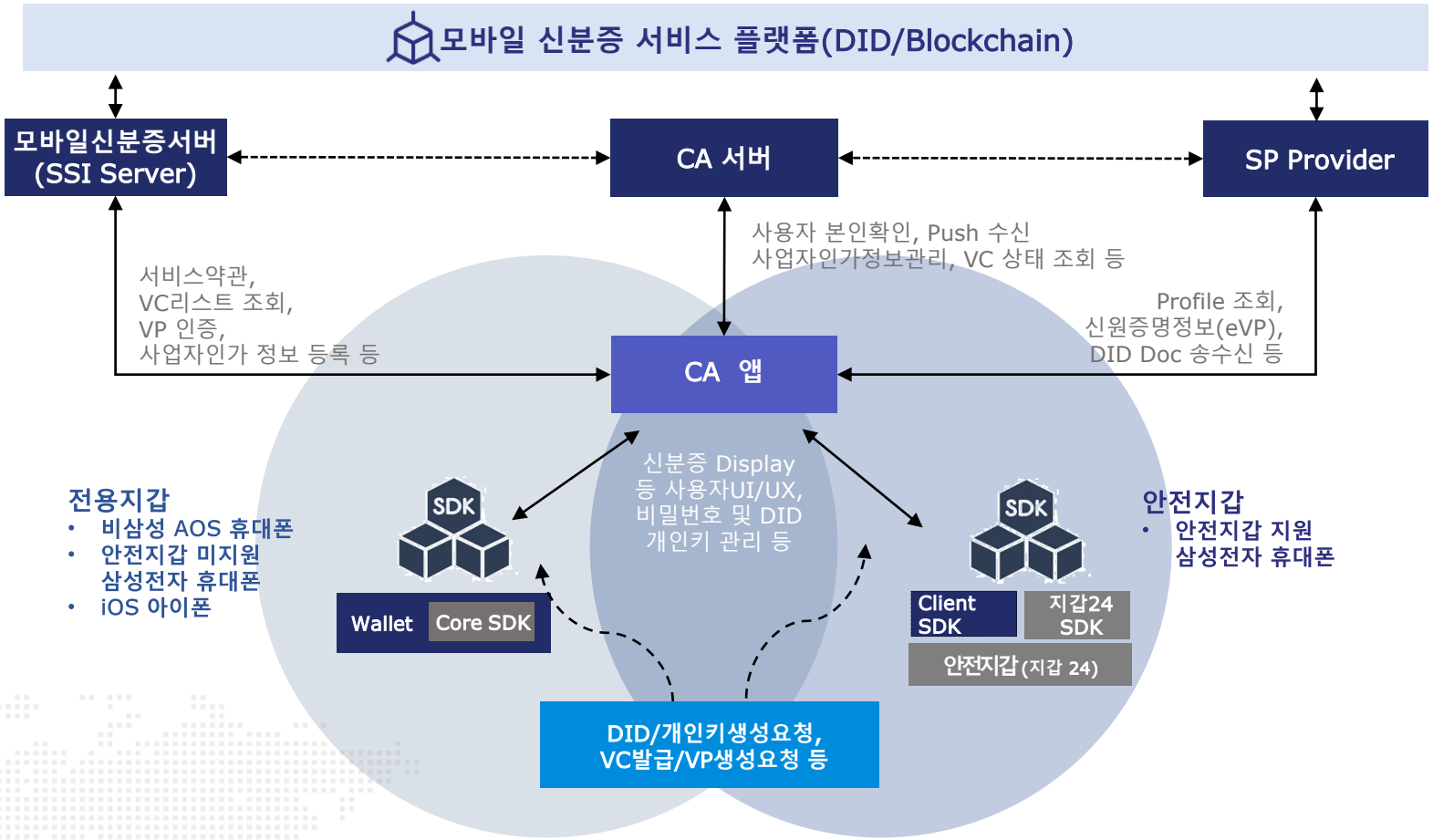
발급이력 등 최소 정보(개인정보 제외)를 여러 대의 쓰기 노드(서버)에 기록

활용 시

여러 대의 읽기 노드(서버)를 제공, SP(서비스 제공자)가 원하는 노드 선택

* 모바일신분증 블록체인 노드 : 블록체인에 개인식별정보를 저장하지 않으며, 공개키가 포함된 DID Document, 발급기관별 VC스키마 및 VC정의, 발급증명 및 상태관리를 위한 VC 메타데이터 만 저장함

3-2. 모바일 신분증 구조 : Wallet 및 CA, SP Provider



*CA 사용자가 모바일 신분증을 발급 및 사용하기 위한 앱과 서버를 제공하는 민간 기관, 평가기준을 통과하여 인가 받은 App 사업자

*SP VC를 검증하고 서비스를 제공하는 서비스제공자

3-2. 모바일 신분증 구조 : 제출모드에 따른 인터페이스

QR-MPM*

모바일에서 QR촬영

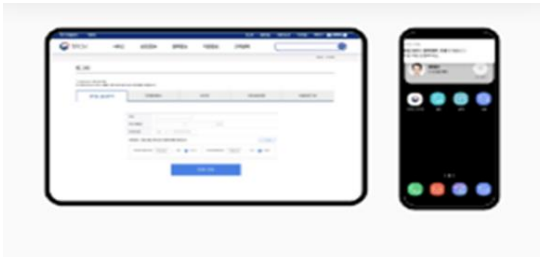
QR-CPM*

모바일에서 QR을 스캔

APP to APP

앱에서 모바일 신분증 앱을 호출

Web to APP



웹에서 모바일 신분증 앱을 호출

BLE* / NFC Tag

모바일 BLE/NFC 활용

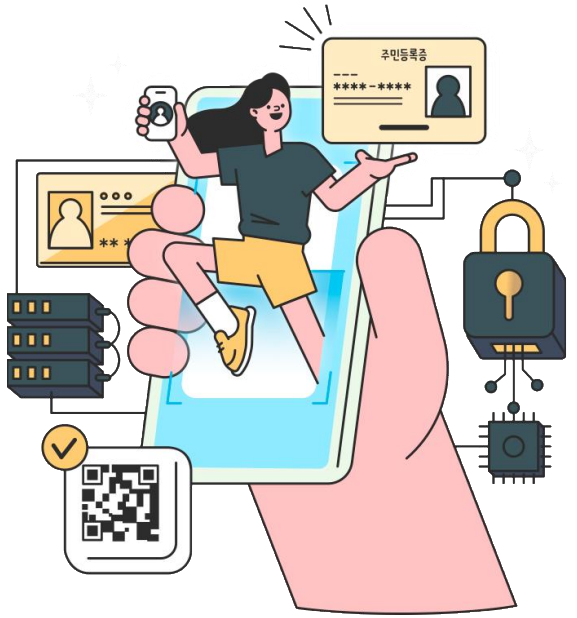
육안 검증

육안으로 모바일 신분증 앱 확인

※ 1) MPM(Merchant Presented Mode): 인증자가 QR코드 제시 → 제출자가 스캔
2) CPM(Customer Presented Mode): 제출자가 QR코드 생성 → 인증자가 QR코드 스캔
3) BLE(Bluetooth Low Energy): 저전력 장치 간 데이터 통신을 위해 사용되는 근거리 무선 통신 기술

3-2. 모바일 신분증 구조 : 제출모드별 참조 API

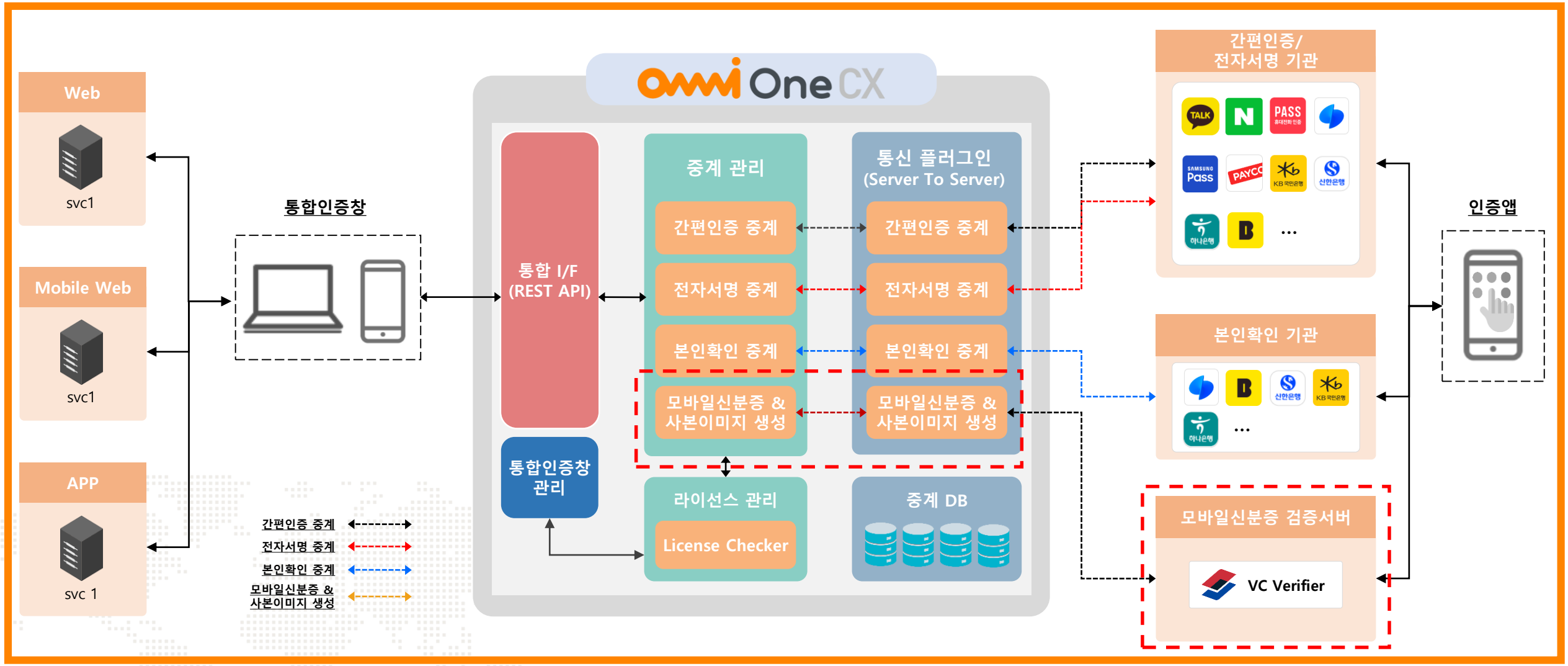
분류	설명
QR-MPM	검증자가 QR을 표출하고 이용자가 QR촬영하여 신원 및 자격을 검증하는 방식
APP 2 APP	이용기관 서비스 앱과 모바일 신분증 앱을 연계하여 신원 및 자격을 검증하는 방식
QR-CPM	이용자가 QR을 표출하고 검증자가 QR촬영하여 신원 및 자격을 검증하는 방식
PUSH	PUSH 메세지를 통해 신원 및 자격을 검증하는 방식
NFC	검증기관의 기기가 NFC 통신을 통하여 모바일 신분증 앱에 정보를 전달하여 검증하는 방식으로, 이용자도 NFC 통신을 통해 신원 및 자격정보를 전달



3-2. 모바일 신분증 구조 : 제출모드별 참조 API

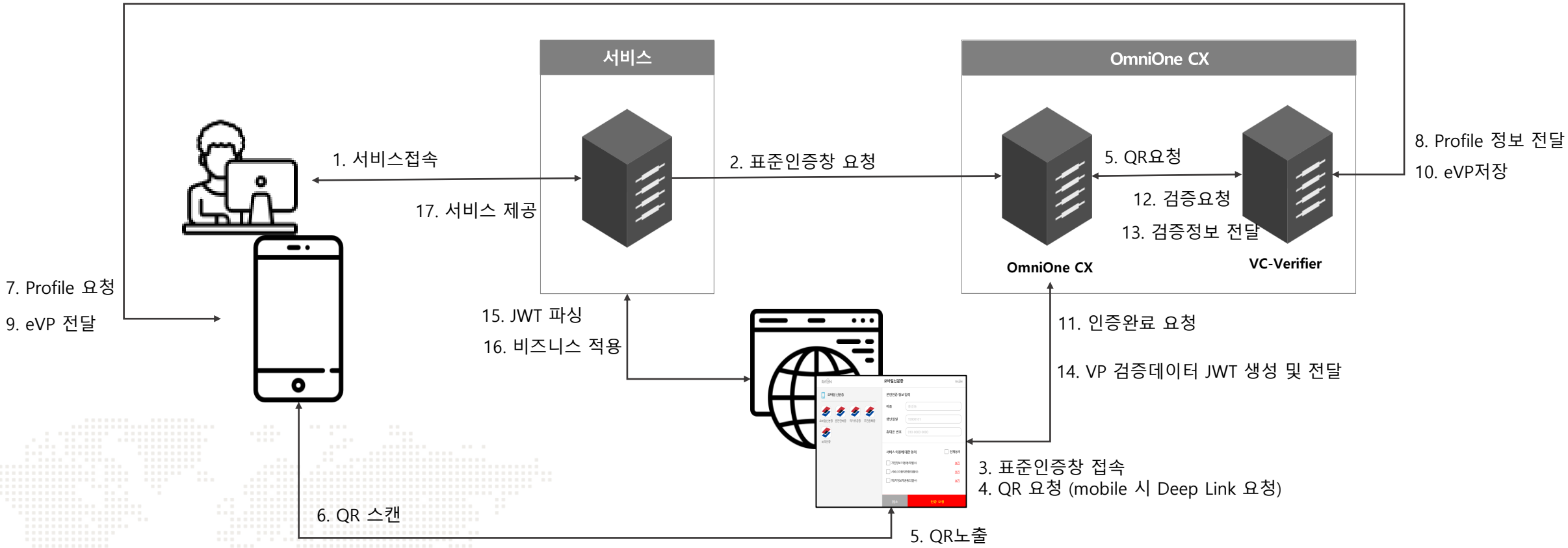
NO	분류	사용자 모바일 인터페이스				모드			
		QR		입력	APP	Indirect (응대장치)	Direct (SP서버)	Proxy (중계서버)	P2P
		표출	스캔	키보드	APP호출				
01	QR-MPM direct mode		O				O		
02	QR-MPM proxy mode		O					O	
03	APP 2 APP direct mode				O		O		
04	App 2 APP indirect mode				O	O			
05	QR-CPM proxy mode	O						O	
06	PUSH			O					
07	NFC				O	O			

3-3. 모바일 신분증 활용 방법 : OmniOne CX 서비스 연계



3-3. 모바일 신분증 활용 방법 : OmniOne CX 서비스 연계 | ① 표준인증창 호출방식

OmniOne CX 표준인증창 연동 흐름도



8.

3-3. 모바일 신분증 활용 방법 : OmniOne CX 서비스 연계 | ① 표준인증창 호출방식

OmniOne CX 표준인증창 호출 방법

```
<script defer="defer" src="https://cx.raonsecure.co.kr:17543/ent/esign/oacx-vendor.js"></script>
<script defer="defer" src="https://cx.raonsecure.co.kr:17543/ent/esign/oacx-ux.js"></script>
<link href="https://cx.raonsecure.co.kr:17543/ent/esign/oacx-ux.css" rel="stylesheet">
```

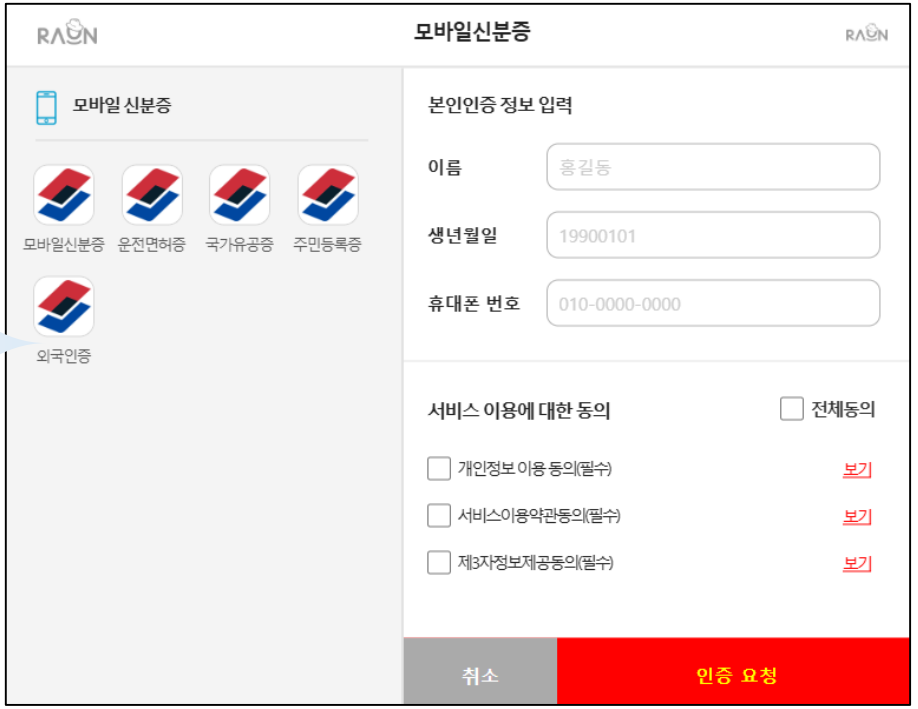
OmniOne CX 표준 인증창을 호출하려는 html 페이지 의 <head> 태그 내 코드 삽입

```
<div id="oacxDiv"><div>
```

OmniOne CX 표준 인증창을 노출할 영역 정의 Div 태그 id 는 oacxDiv 로 고정

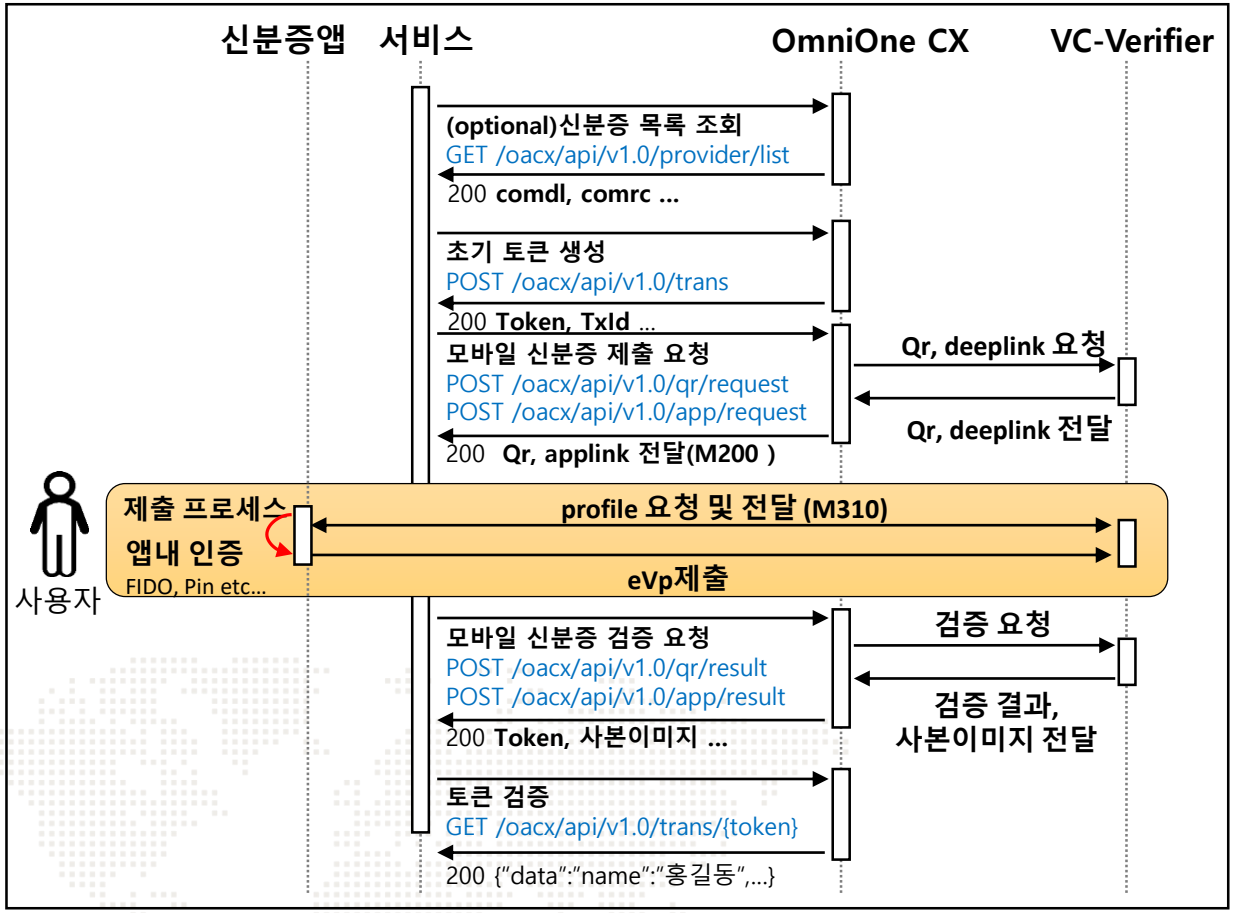
```
document.getElementById("mobileBtn").onclick = function() {
    var json = {
      contentInfo: {
        signType: "ENT_MID"
      },
      compareCI: false
    }
    OACX.LOAD_MODULE("https://cx.raonsecure.co.kr:17543/ent/esign/config/config.mid.json", json,
function(res) {
      //Success handler
      console.log("모바일신분증: ", res)
    })
```

OmniOne CX 표준 인증창을 호출하려는 Element(button 등) 이벤트에 코드 적용



3-3. 모바일 신분증 활용 방법 : OmniOne CX 서비스 연계 | ② Restful API 연동 방식

OmniOne CX Restful API 연동



종류	API ENTRY	Method	사용환경
신분증 목록	/oacx/api/v1.0/provider/list	GET	공통
접근 토큰 요청	/oacx/api/v1.0/trans	POST	공통
QR 요청	/oacx/api/v1.0/authen/qr/request	POST	PC
앱링크요청	/oacx/api/v1.0/authen/app/request	POST	Mobile
QR 검증 요청	/oacx/api/v1.0/authen/qr/result	POST	PC
앱 검증요청	/oacx/api/v1.0/authen/app/result	POST	Mobile
토큰 검증	/oacx/api/v1.0/trans/{token}	GET	공통

※ API의 상세 정보는 API 문서를 참고하세요. https://www.didalliance.org/hackathon/2025/2025_블록체인_AI_해커톤_OmniOne_CX-VC-Verifier_v1.0_API_매뉴얼.pdf

3-3. 모바일 신분증 활용 방법 : OmniOne CX 서비스 연계 | 검증 데이터(JWT) 파싱

방식별 OmniOne CX 검증 데이터(JWT) 파싱

① 표준인증창 호출방식

```
OACX.LOAD_MODULE("./esign/config/config.mid.json", json, function(res) {
//성공 callback function
   var token = res.token;
   var token = new XMLHttpRequest();
   var url = 'https://cx.raonsecure.co.kr:18543/oacx/api/v1.0/trans?token=' + encodeURIComponent(token);

   xhr.open('GET', url, true);
   xhr.onreadystatechange = function() {
      if (xhr.readyState === XMLHttpRequest.DONE) {
        if (xhr.status === 200) {
          console.log('성공:', xhr.responseText);//신분증 정보를 활용하여 처리
        } else {
          console.error('에러 발생:', xhr.status, xhr.statusText);
        }
      }
   };
   xhr.send();
})
```

OACX.LOAD_MODULE에 성공 callback 지정
callback에 전달되는 토큰을 파싱해 서비스에 활용
토큰파싱은 토큰 검증API에 결과 토큰을 파라메터로 요청

② RestApi 연동방식

POST https://cx.raonsecure.co.kr:18543/oacx/api/v1.0/authen/qr/result
Request Body:

```
{
  "provider":"comdl_v1.5,
  "token":"eyJhbGciOiJIUzI1NiJ9.eyJjeElkIjoiMjAyNTA0 ....",
   "txId":"202504231316340046E66BF2C",
   "cxId":"202504231316340046E66BF2CQR",
   "contentInfo":{"signType":"ENT_MID"}
}
```

Response Body:

```
{
   "token": "eyJhbGciOiJIUzI1NiJ9.eyJsb2NwYW5tIjoi7ISc7Jq47Yq567OE7Iuc6rK97LCw7LKDsh .....",
   "data"{
       "converterimage": "iVBORw0KGgoAAAANSUhEUgAAA/sAAAKYCAIAAA...."
    }
}
```

검증 API 성공 시, 토큰 형태로 제출한 claim이 수신됨

data.convertimage로 신분증 사본 이미지 (Base64)전달.

GET https://cx.raonsecure.co.kr:18543/oacx/api/v1.0/trans/eyJhbGciOiJIUzI1NiJ9.eyJ....
Response Body:

```
{
 "data": {
        "locpanm": "서울특별시경찰청장",
        "vcTypeCode": "mdriverlic",
         .....
   }
}
```

토큰 파싱 API를 호출하여, 토큰을 검증하고, 토큰에 담긴 claim을 수신
신분증별 claim은 API문서 참고

- **Token parsing 이후 모바일신분증 Claim의 key 및 value 는 API 가이드 문서 참조**

3.3. 모바일 신분증 활용 방법 : 적용 사례 | 정부24

정부24, 모바일 신분증을 이용한 로그인

모바일 신분증 인증 프로세스

1. 간편인증 로그인

2. 인증창 호출

3. 인증 요청

4. 인증 완료

5. 로그인 완료

Appendix 1.
Open DID

1-1. Open DID 소개(배경 및 목적)

1-2. Open DID 범위

Github 오픈소스 공개 범위

SDK와 애플리케이션 구성 및 역활

1-3. Open DID SDK 구조

- 클라이언트 월렛 SDK 설명
- 서버용 SDK 설명
- Smart contract 설명
- 각 SDK용 API Document 링크

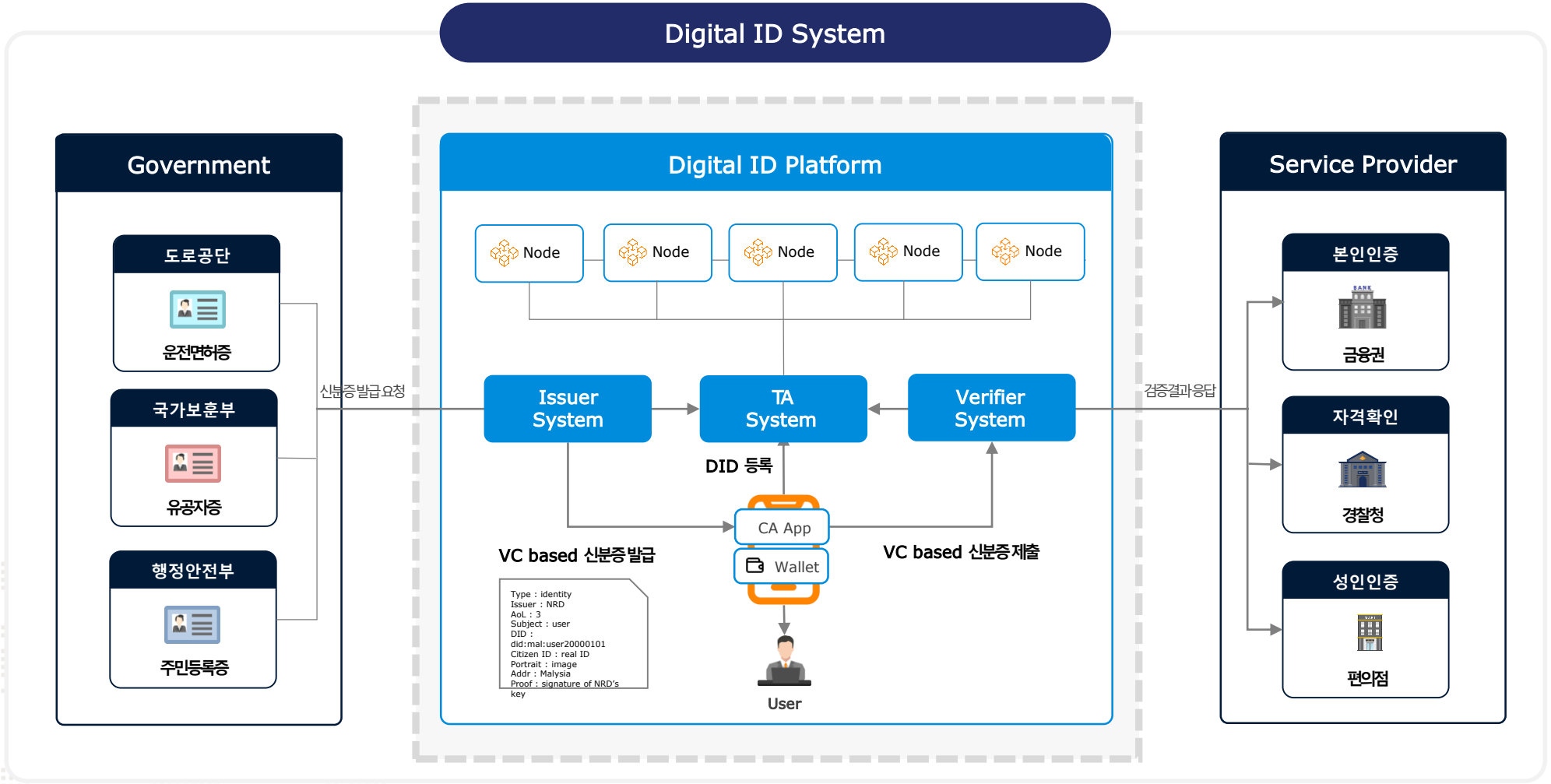
1-4. Open DID 애플리케이션 구조

- Trust Agent
- Issuer
- Verifier
- Mobile Application
- 각 샘플 소스 링크
- Demo Source 링크

1-5. Open DID 활용 방법

- 서비스 개발 방법
- 적용 방법

1.1. Open DID 소개



1.2. Open DID 범위 – Github 오픈소스 공개 범위

TAS 사업자(TAS Provider)

- Entity 등록: 모든 Entity의 신원을 인증한 후 OPEN DID 시스템에 등록하고, 해당 Entity에 대하여 적절한 권한을 부여한다.
- 보안 정책 관리: OPEN DID 시스템의 Entity들이 신뢰관계를 구축하기 위한 보안 정책을 관리하고 적용한다.

발급 사업자(Issuer Provider)

- 디지털 신분증 발급: 사용자의 신분 혹은 자격 정보를 소유하고 있으며, 사용자에게 디지털 신분증을 발급할 수 있다.
- 디지털 신분증 갱신: 사용자의 신원 정보가 변경되거나 신분증의 유효 기간
  만료 시 새로운 신분증을 생성하거나 갱신하여 업데이트 할 수 있다.
- 디지털 신분증 폐기: 사용자의 취소나 신청 철회 시 디지털 신분증을 폐기할
  수 있다.

검증 사업자(Verifier Provider)

- 디지털 신분증 검증: 사용자의 디지털 신분증을 요청하고 검증한 후, 사용자가 서비스를 이용할 권한이 있는지 확인한다.
- 서비스 제공: 유효한 디지털 신분증을 가진 사용자에게 서비스를 제공한다.

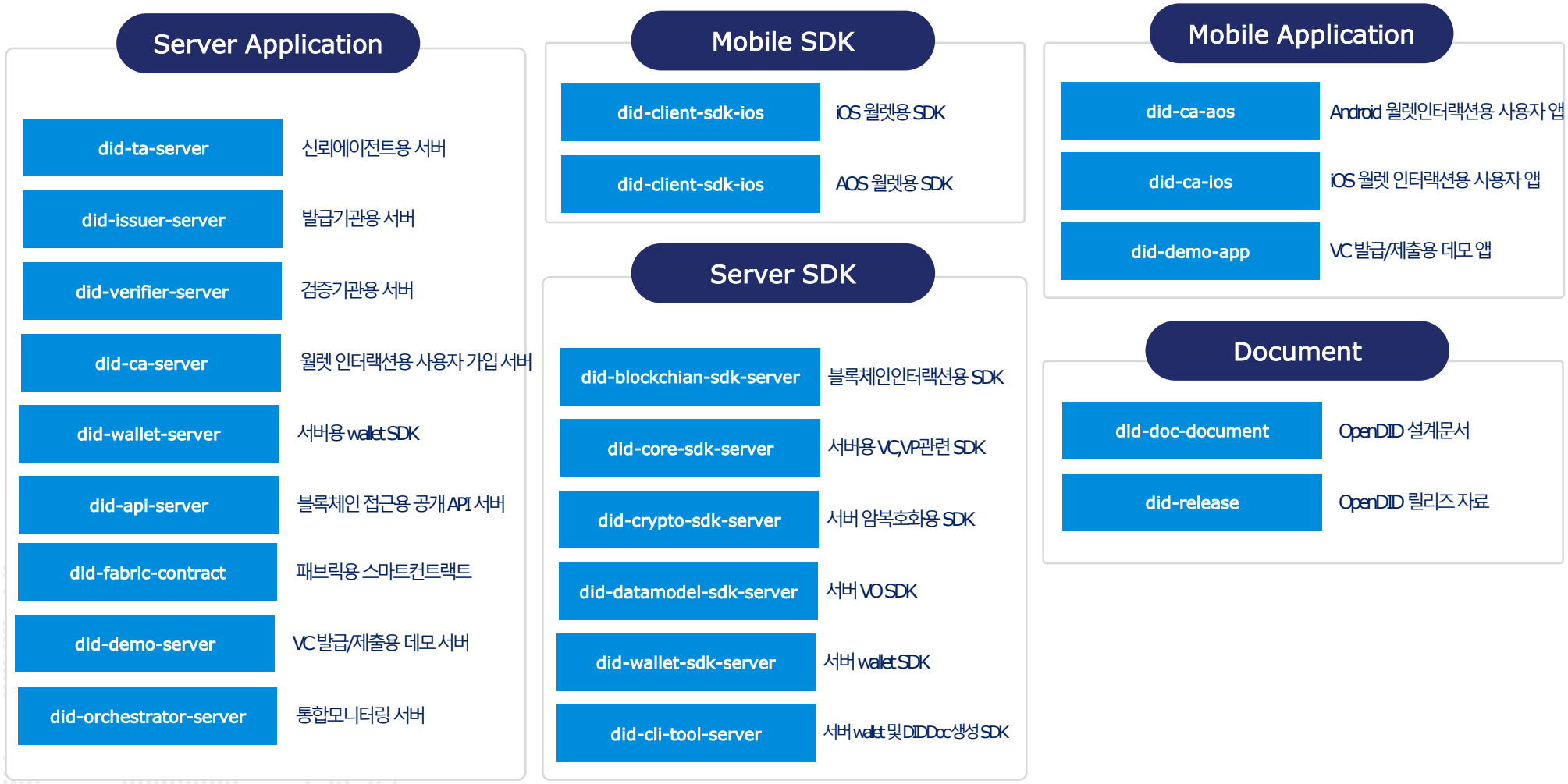
인가앱 사업자(Certificate Provider)

- 앱 상태 확인: 사용자의 앱이 정상적인 상태인지 확인하고, 신뢰할 수 있는 앱인지 확인한다.
- 월렛 신뢰정보 제공: 사용자가 요청한 앱에 대한 신뢰성 정보를 제공한다.

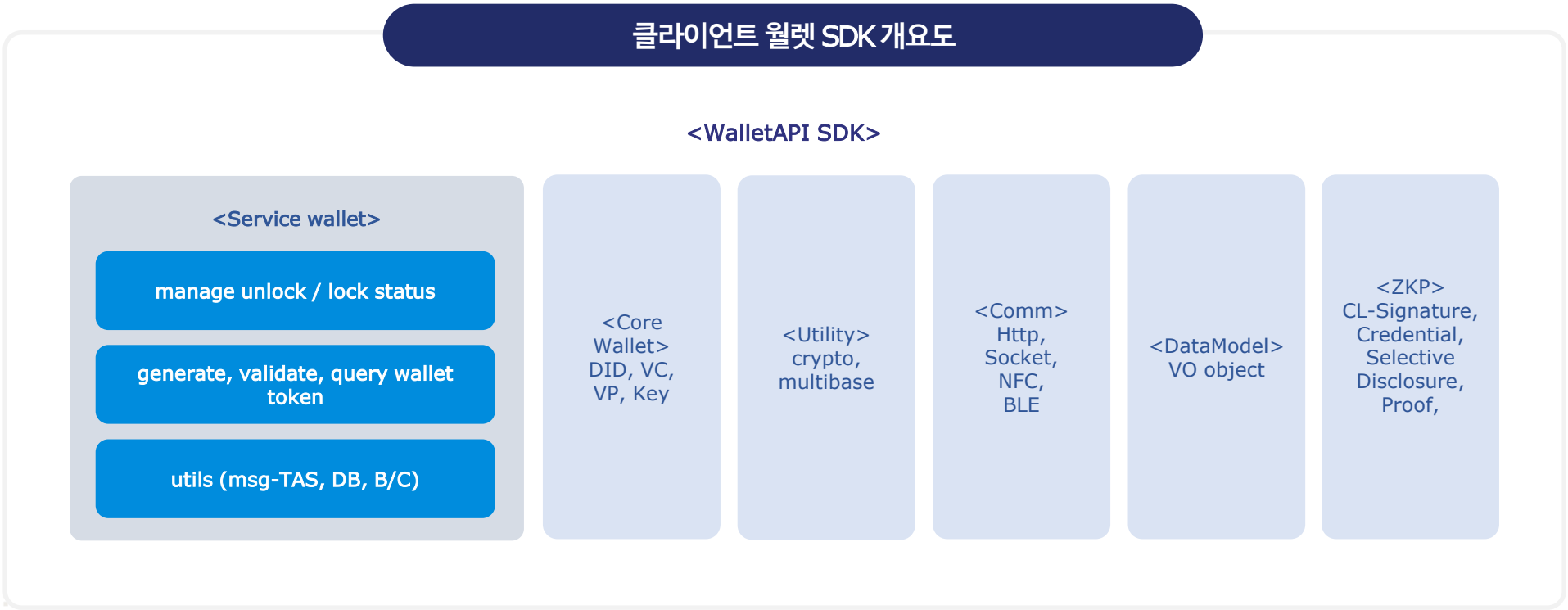
Wallet 사업자(Wallet Provider)

- 키 저장소 연동: 타입에 맞는(디바이스 / 클라우드) 월렛 키를 저장할 저장소(HSM, File, Vault)와 연동하는 인터페이스를 제공한다.
- 월렛 상태 확인: 사용자의 월렛이 정상적인 상태인지 확인하고, 신뢰할 수 있는 월렛인지 확인한다.
- 월렛 신뢰정보 제공: 사용자가 요청한 월렛에 대한 신뢰성 정보를 제공한다.
- Crypto 기능 제공 : 암호화/복호화, 서명/서명검증, 인코딩/디코딩 기능을 제공한다.

1.2. Open DID 범위 – SDK와 애플리케이션 구성 및 역할



1.3. Open DID SDK 구조 - 클라이언트 월렛 SDK 설명



Wallet API
- 월렛 접근을 위한 external API

Service wallet
- 서비스 통신을 위한 토큰 관리 및 메시지 생성

Core wallet
- DID/ VC 등을 관리

Utility
- 암호 알고리즘 제공

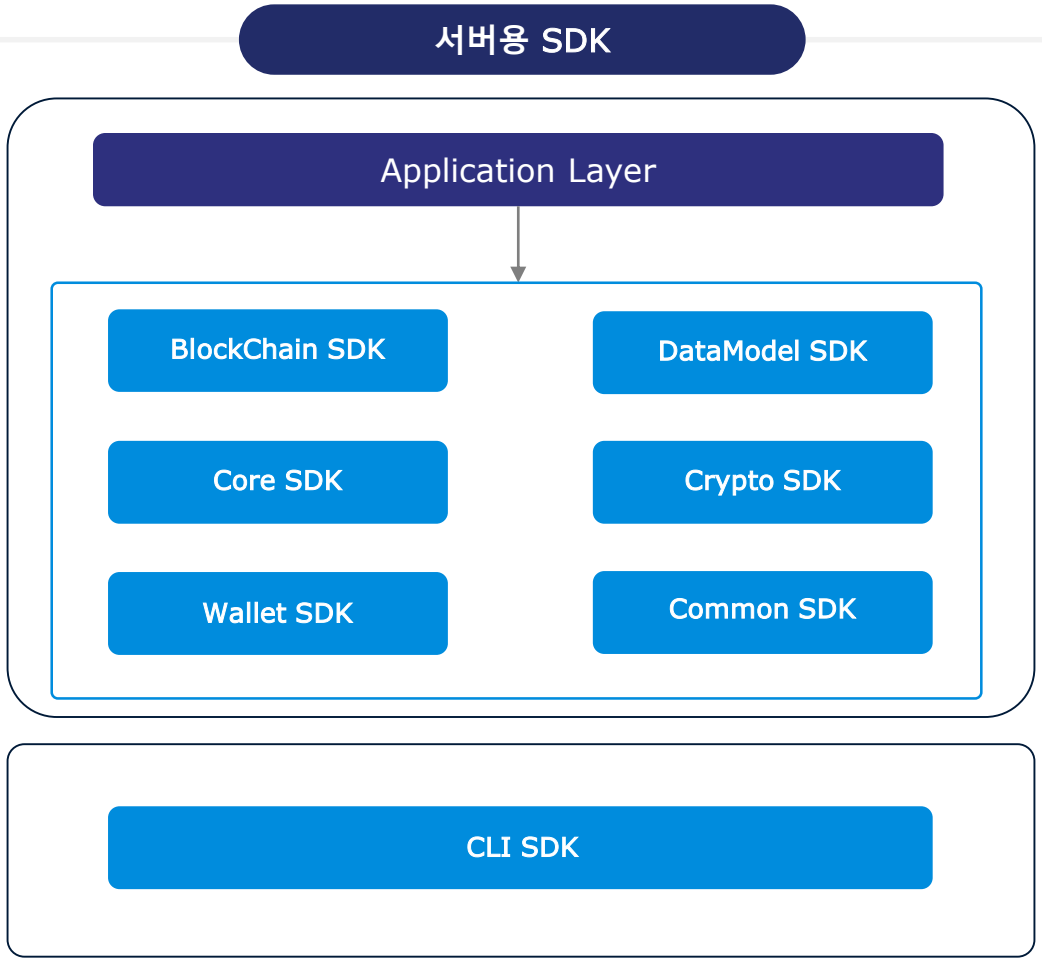
Comm
- 통신 관리

Data model
- 앱/월렛 SDK에서 사용하는 데이터 모델 정의

ZKP
- 영지식 증명

Wallet Type
- Local Wallet(File based wallet)

1.3. Open DID SDK 구조 - 서버용 SDK 설명



BlockChain SDK
- DID 및 VC에 대한 정보를 블록체인에 등록, 조회, 상태 변경

Core SDK
- DID Document, 검증 가능한 자격 증명(VC), 프레젠테이션(VP) 데이터를 생성, 관리, 검증

Wallet SDK
- 디지털 월렛을 생성, 관리, 암호화, 서명 및 검증하는 기능을 제공

DataModel SDK
- 서버에서 공통적으로 사용하는 데이터 모델 정의

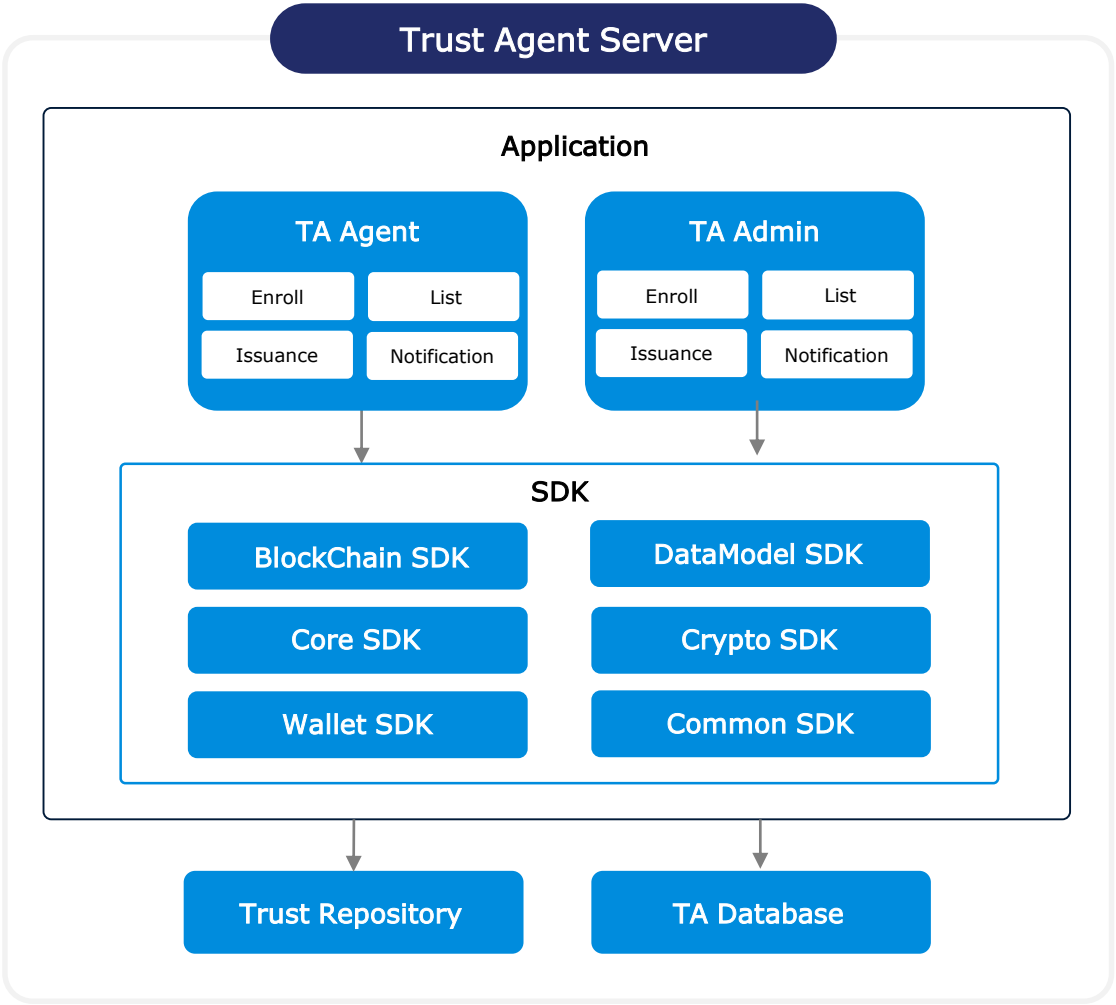
Crypto SDK
- 암호화/복호화, 전자서명/검증, 해싱 및 키 관리 기능을 제공

Common SDK
- 서버에서 공통으로 사용하는 유틸리티 라이브러리

CLI SDK
- 서버 월렛, DID Document 생성 기능 제공

1.3. Open DID SDK 구조 – SDK용 API Document 링크

1.4. Open DID 샘플 애플리케이션 – Trust Agent



Open DID에서 제공하는 SDK를 이용하여 TA를 구현한 서버

신뢰 체인 형성

- Digital Identity Committee (이하 위원회)로부터 다음의 권한을 위임받아 신뢰체인을 구축하고 운영

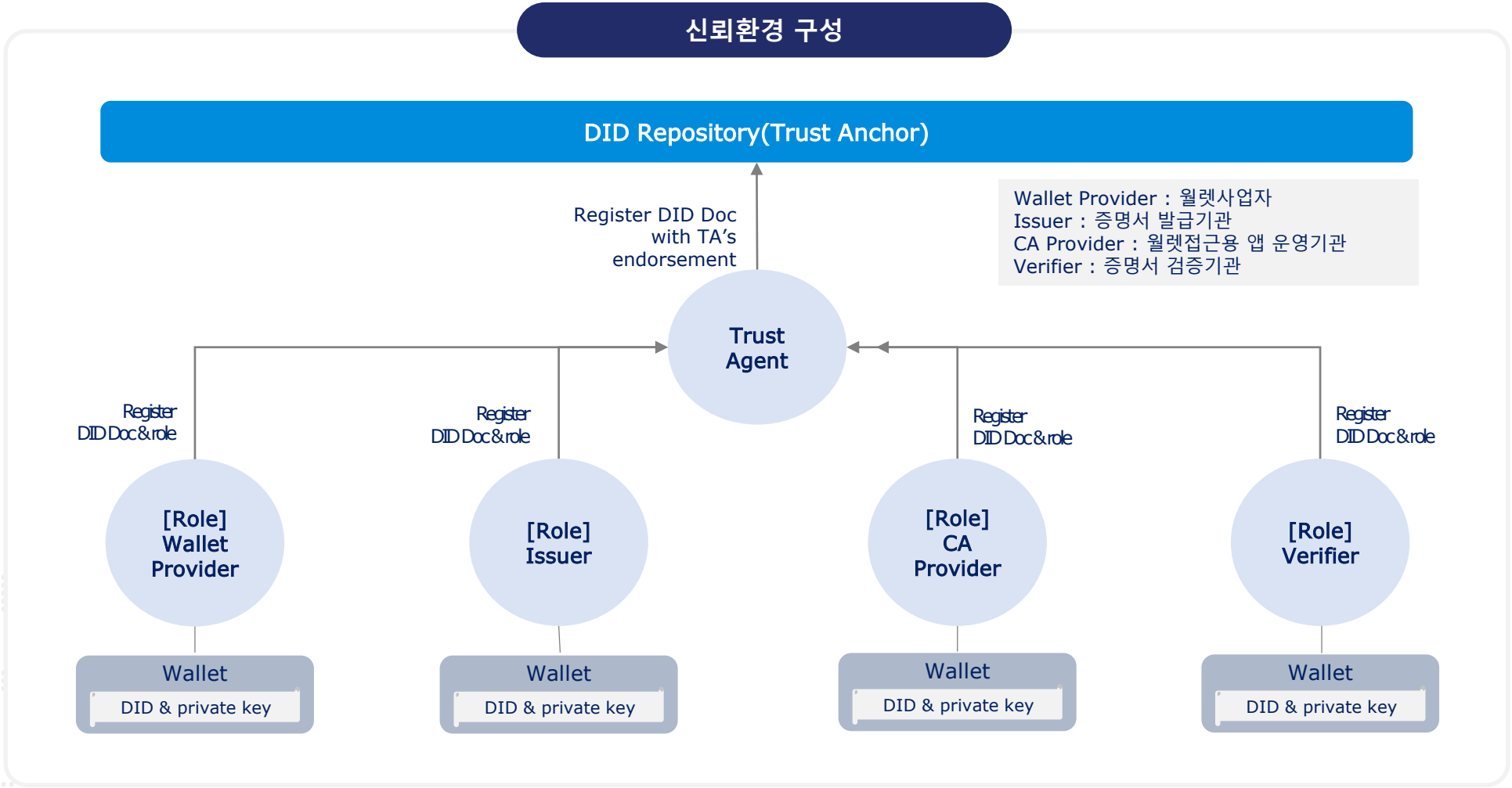
분산 식별자 문서(DID Document) 등록

Entity(Issuer, Verifier and other providers) 등록 및 가입증명서(Certificate VC) 발급

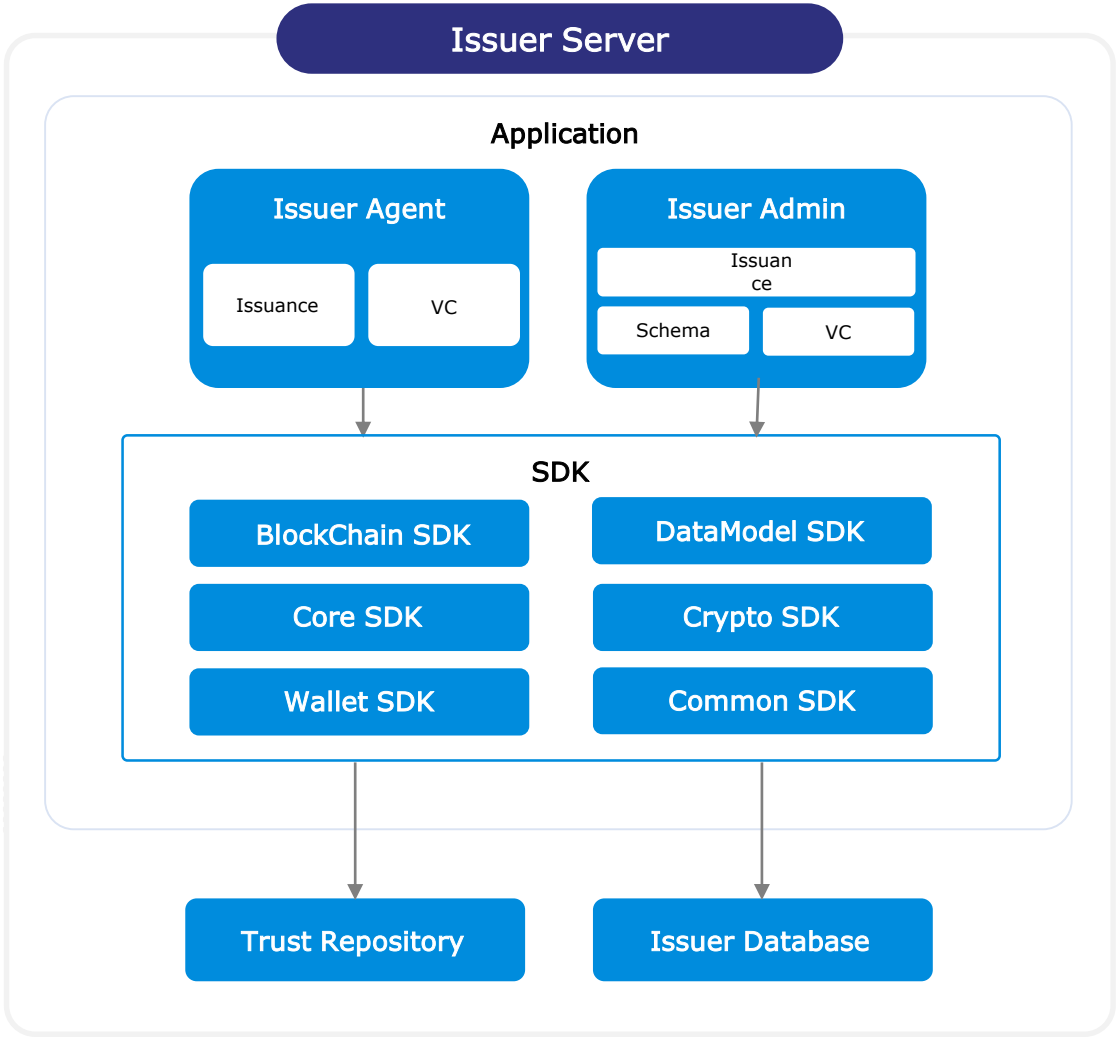
Client Wallet 등록

사용자 등록/탈퇴

1.4. Open DID 샘플 애플리케이션 – Trust Agent



1.4. Open DID 샘플 애플리케이션 – Issuer

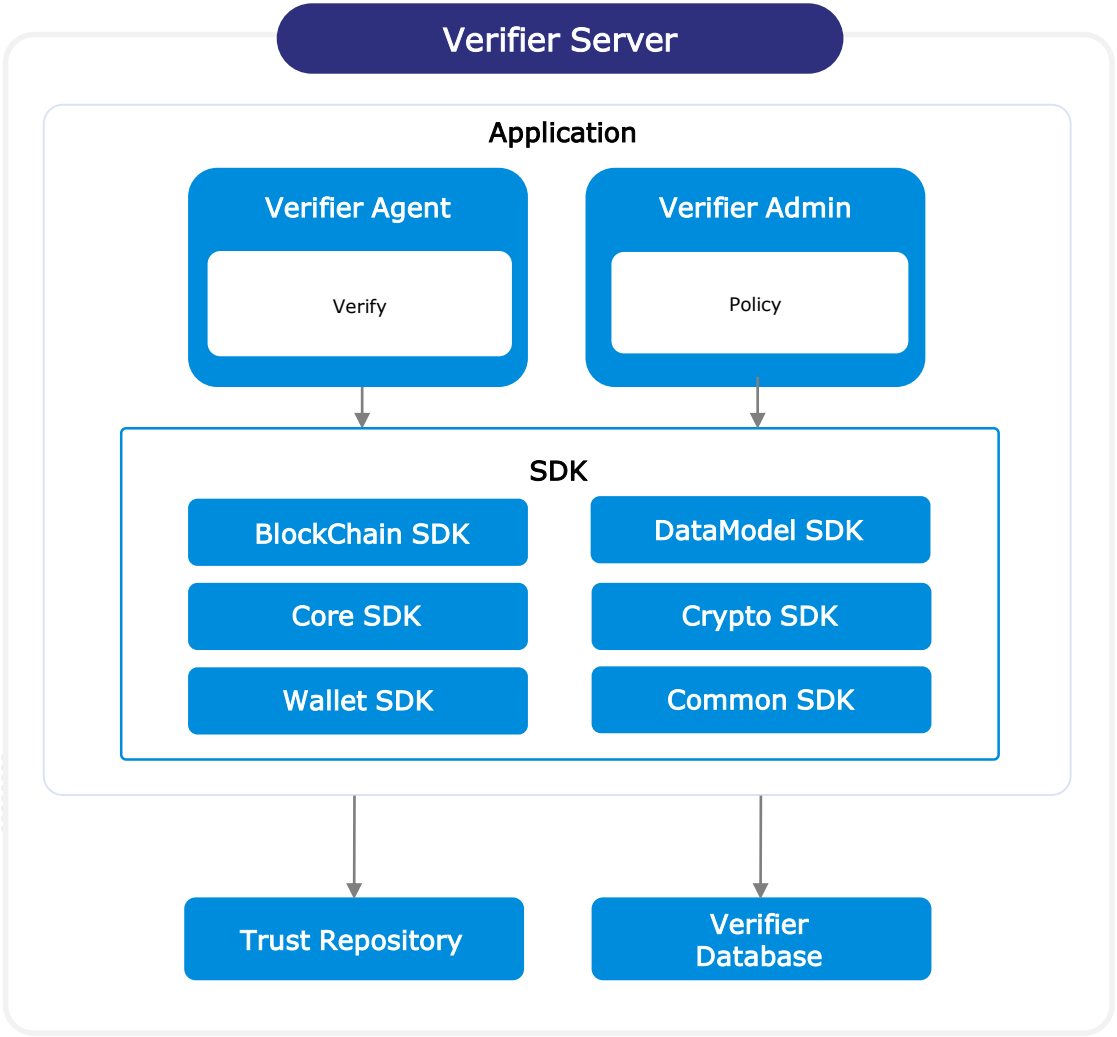


Open DID에서 제공하는 SDK를 이용하여 Issuer를 구현한 서버

검증 가능한 크리덴셜 발급

검증 가능한 크리덴셜 상태 관리

1.4. Open DID 샘플 애플리케이션 – Verifier



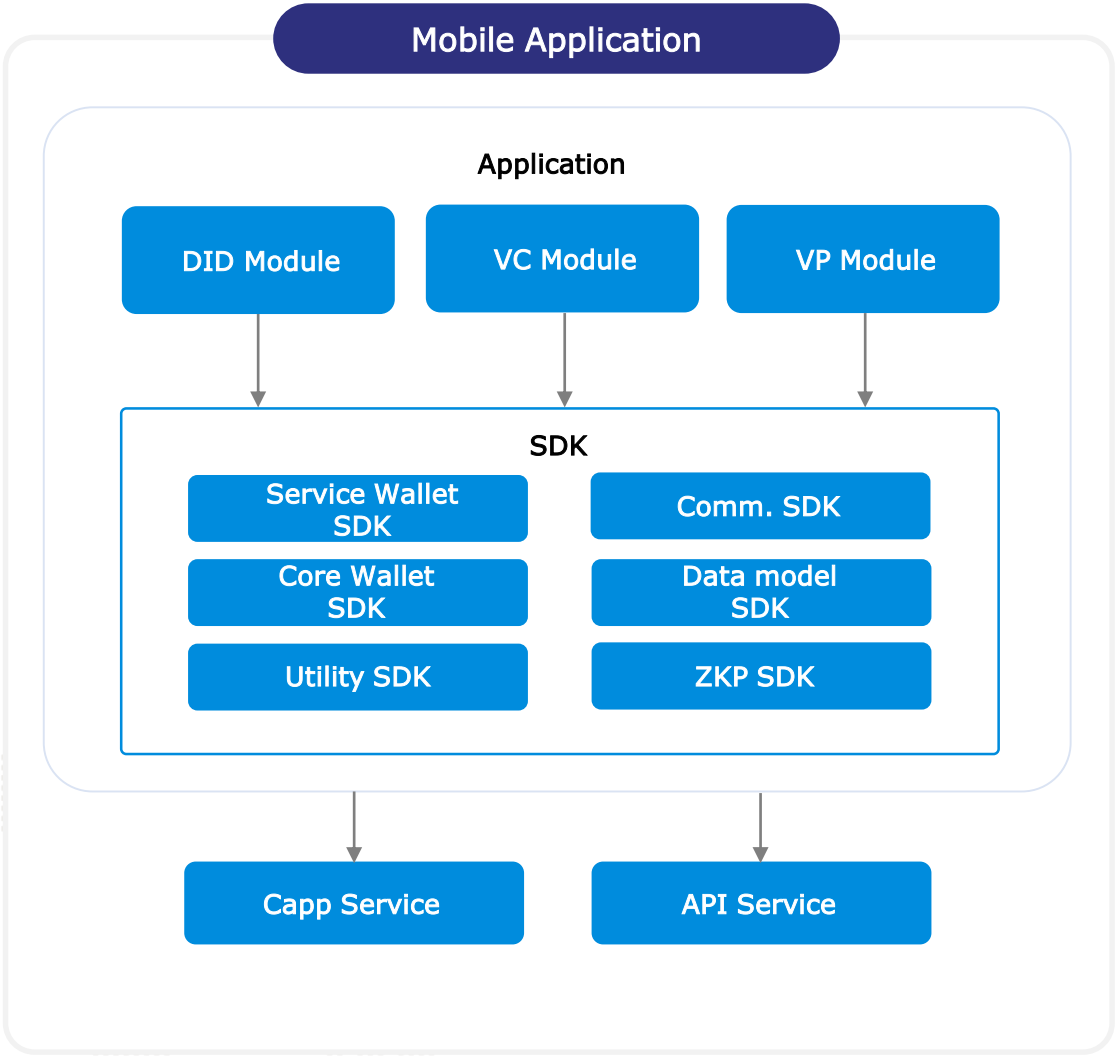
Open DID에서 제공하는 SDK를 이용하여 Verifier를 구현한 서버

VP 제출 정책 설정

VP 검증

- 사용자가 제출한 VP를 검토하여 데이터의 진위성과 무결성을 확인

1.4. Open DID 샘플 애플리케이션 – Mobile Application



Open DID에서 제공하는 SDK(AOS, iOS)를 이용하여 구현 모바일 응용 프로그램

DID 모듈

- Wallet 생성, 잠금/잠금해제 (lock/unlock) 및 등록
- 사용자 DID 생성, 사용자 등록/회원 탈퇴
- DID 키 사용을 위한 사용자 인증 (PIN, BIO 인증)
- DID 상태변경

VC 모듈

- VC 발급, 삭제, 재발급
- VC 조회, 상태변경

VP 모듈

- VP 생성/제출

1.4. Open DID 샘플 애플리케이션 – 소스 링크

Server Application

did-ta-server	신뢰에이전트용 서버	https://github.com/OmniOneID/did-ta-server
did-issuer-server	발급기관용 서버	https://github.com/OmniOneID/did-issuer-server
did-verifier-server	검증기관용 서버	https://github.com/OmniOneID/did-verifier-server
did-ca-server	월렛 인터랙션용 사용자 가입 서버	https://github.com/OmniOneID/did-ca-server
did-wallet-server	서버용 wallet SDK	https://github.com/OmniOneID/did-wallet-server
did-api-server	블록체인 접근용 공개 API 서버	https://github.com/OmniOneID/did-api-server
did-fabric-contract	패브릭용 스마트컨트랙트	https://github.com/OmniOneID/did-fabric-contract
did-demo-server	VC 발급/제출용 데모 서버	https://github.com/OmniOneID/did-demo-server
did-orchestrator-server	통합모니터링 서버	https://github.com/OmniOneID/did-orchestrator-server

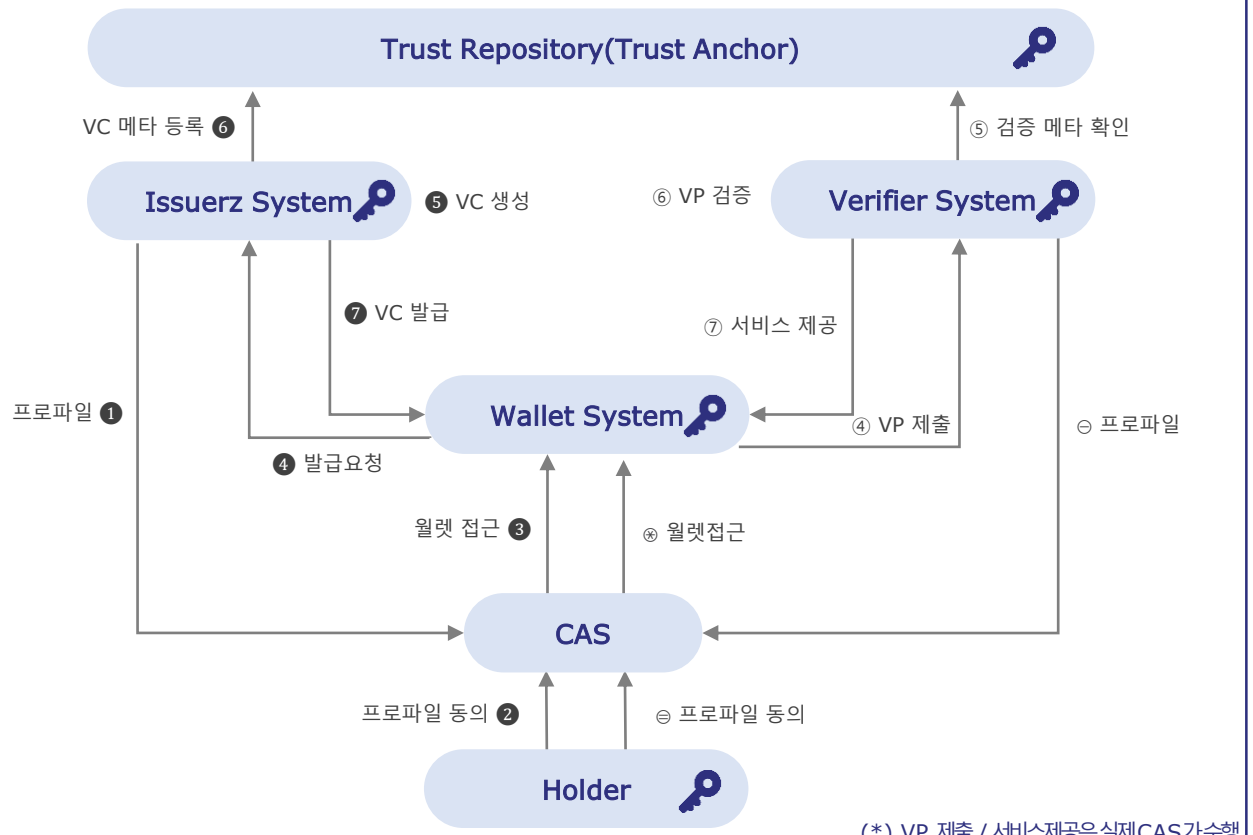
1.4. Open DID 샘플 애플리케이션 – 소스 링크

Mobile Application

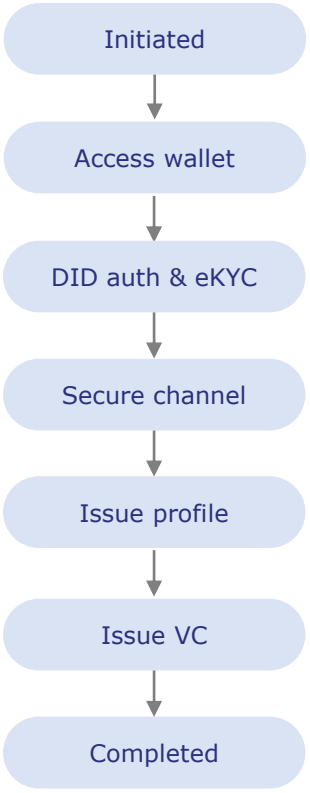
did-ca-aos	AOS 월렛인터랙션용 사용자 앱	https://github.com/OmniOneID/did-ca-aos
did-ca-ios	iOS 월렛 인터랙션용 사용자 앱	https://github.com/OmniOneID/did-ca-ios

1.4. Open DID 샘플 애플리케이션 – 데모

발급 / 제출 개요도



발급 라이프사이클

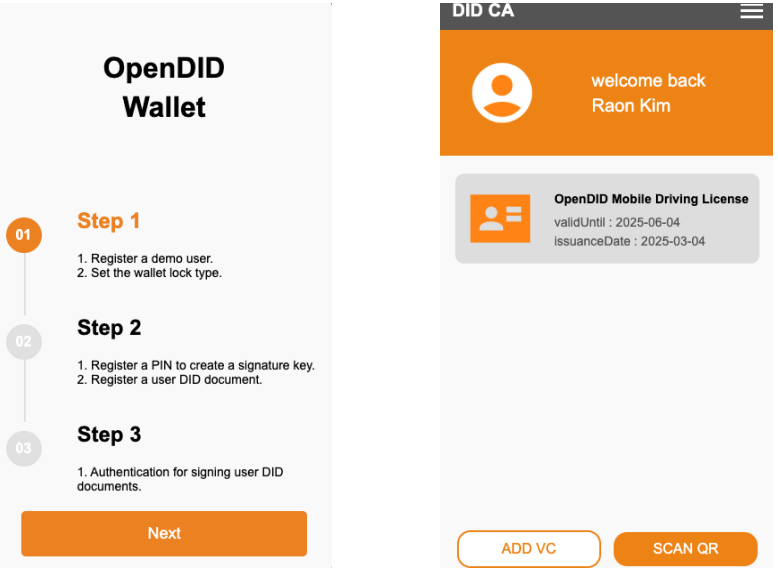


1.4. Open DID 샘플 애플리케이션 – 데모

Demo App

did-demo-app VC 발급/제출용 데모 앱

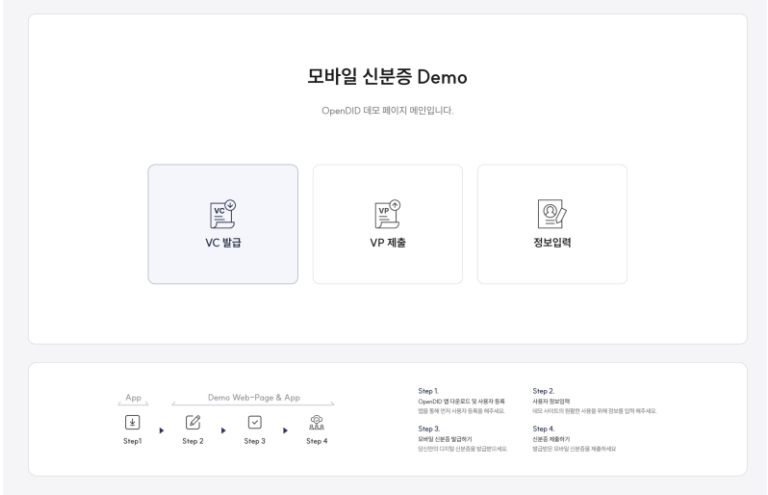
https://github.com/OmniOneID/did-demo-app



Demo Server

did-demo-server VC 발급/제출용 데모 서버

https://github.com/OmniOneID/did-demo-server



1.5. Open DID 활용방법 – 서비스 개발 방법

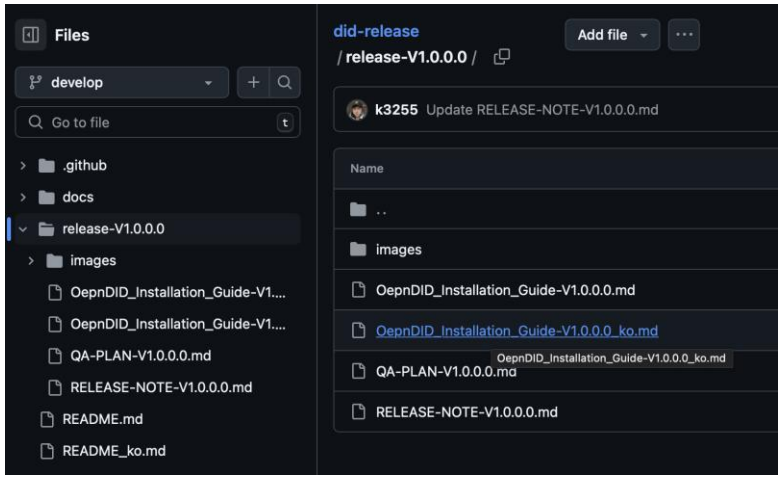
01. 서비스 시나리오 확정

02. 데모 통한 동작 확인

03. SDK, Sample Application 소스 참조

04. 샘플 애플리케이션 참조, 서비스 개발

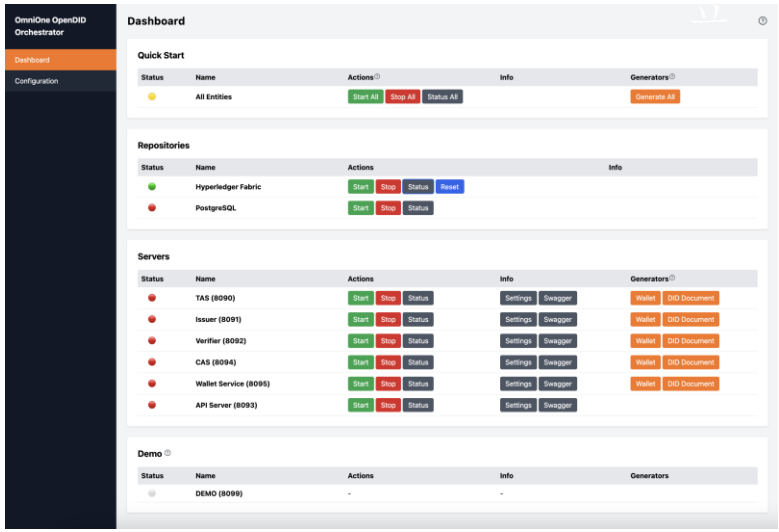
05. 다양한 증명서 발급 및 제출



did-release

OpenDID 릴리즈 자료

https://github.com/OmniOneID/did-release



did-orchestrator

통합모니터링 서버

https://github.com/OmniOneID/did-orchestrator

1.5. Open DID 활용방법 – 적용방법

Case 1. DID 기반 학생증(VC) 발급 및 출석체크

샘플 애플리케이션 소스 확인 → 학생증 발급 서버 개발 → 모바일 어플리케이션 개발 → 학생증 검증 서버 개발 → 컨트랙트 개발 → 컨트랙트 등로

Case 2. DID 기반 투표시스템

샘플 애플리케이션 소스 확인 → DID 등록 서버 개발 → 투표 서버 포탈 개발 → Zkp 앱 개발 → Zkp 검증 서버 개발 → 컨트랙트 등로

Case 3. DID 기반 익명 메신저

샘플 애플리케이션 소스 확인 → 메신저 중계 서버 개발 → 메신저 앱 개발 → 앱내 클라이언트 월렛 SDK 적용 → DID 등록 서버 개발 → 컨트랙트 등록

Appendix 2.
BESU 메인넷

2-1. 메인넷 기반 Web3 서비스 소개
2-2. Web3 API 활용 개발 방법
2-3. 메인넷 기반 서비스 적용 방법

2.1. BESU 메인넷 : 메인넷 기반 Web3 서비스 소개

BESU 메인넷 Web3 서비스

- Web3 기반의 다양한 블록체인 서비스 제공 가능
- 개발자들이 간편한 Web3 전환 서비스를 개발할 수 있는 API 제공

스마트 컨트랙트 배포 서비스

- 스마트 컨트랙트의 솔리디티 파일과 컴파일하여 나오는 ABI + Bytecode를 손쉽게 등록하도록 지원
- 배포한 스마트 컨트랙트를 이용한 안정적인 트랜잭션 운영 및 처리 가능

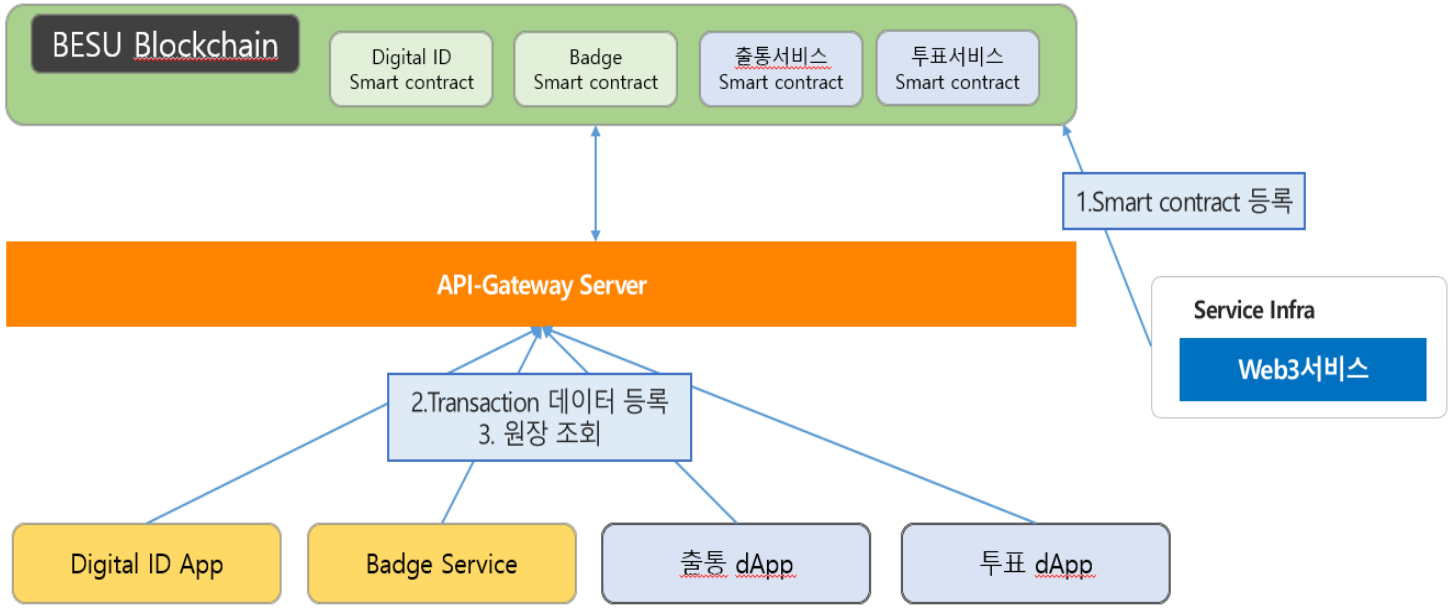
환경 계정 제공 서비스

- 개발 편의성을 제공하기 위해 계정을 관리하도록 지원
- EOA (Externally Owned Account) 타입의 외부 소유 계정 지원
- 사용자가 개인키를 직접 관리하는 로컬 월렛 방식
- 사용자는 EOA 개인키를 사용하여 트랜잭션을 서명하고 전송함으로써 해당 계정의 소유를 증명하고 Web3 서비스를 이용

2.1. BESU 메인넷 : 메인넷 기반 Web3 서비스 소개

BESU Web3 서비스 구성도

- dApp 용 월렛 생성, 컨트랙트 등록, 트랜잭션 API 등을 제공



dApp 등록용 Web3 서비스 기능

01. 메인넷 서비스 가입 및 사용자 계정 생성
02. API-Key 발급
03. 월렛 생성
04. 컨트랙트 등록 및 배포
05.트랜잭션 발송
06. 블록체인 원장 조회

2.2. BESU 메인넷 : Web3 API 활용 개발 방법

서비스 가입 사용자 계정 생성

- 사용자의 계정 생성
- Google 로그인 지원

환경 계정 생성

- 서비스를 진행하기 위한 환경 계정 생성
- 환경 계정 생성 후 개인키 발급

API Key 생성

- API 호출 시 필요한 API Key 발급
- API Key 의 목적 선택

스마트 컨트랙트 등록

- 스마트 컨트랙트를 등록하여 Besu에 배포
- 타입은 Solidity 파일 혹은 ABI + Bytecode 업로드

등록된 스마트 컨트랙트 배포

- 배포할 등록된 컨트랙트 정보 입력

Transaction 및 App 개발

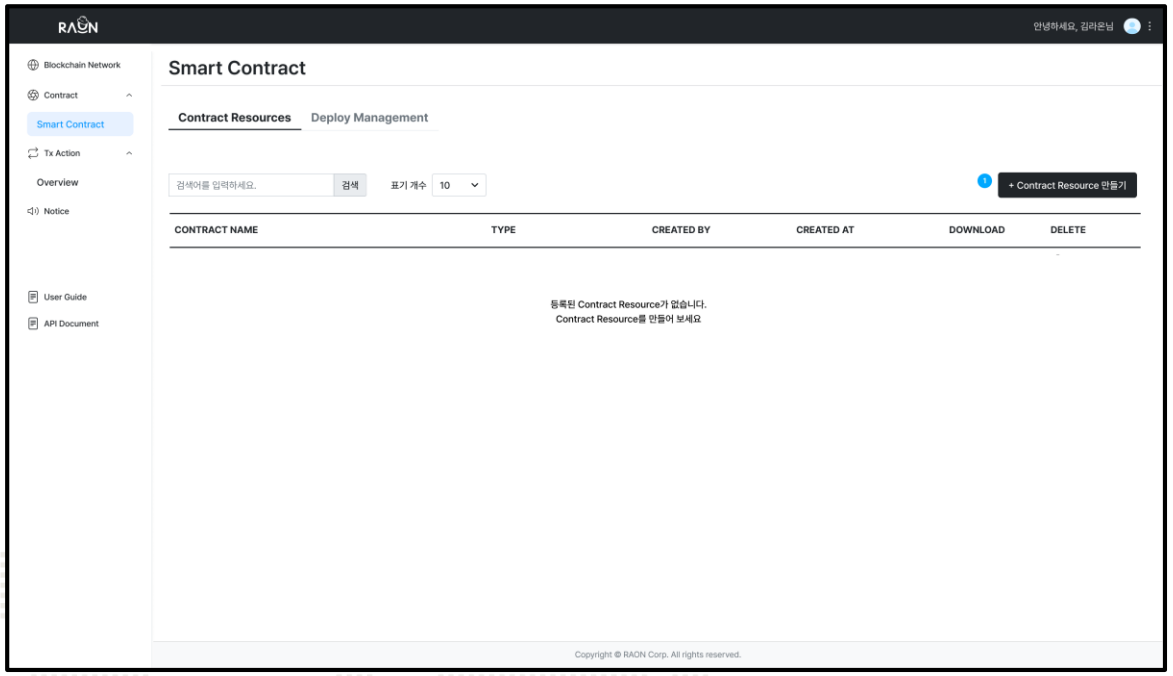
- 등록된 컨트랙트에 Transaction 기록
- dApp 개발

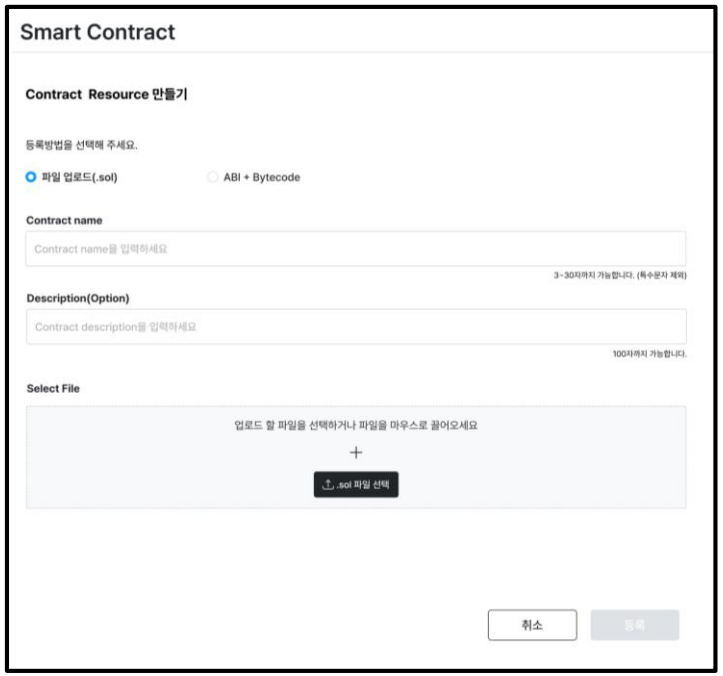
※ OmniOne Web3 이용방법은 유저 가이드 문서를 참고하세요. https://www.didalliance.org/hackathon/2025/2025_블록체인_AI_해커톤_OmniOne_Web3_User_Guide.pdf

2.2. BESU 메인넷 : Web3 API 활용 개발 방법

스마트 컨트랙트 등록

- 스마트 컨트랙트를 등록하여 Besu에 배포
- 타입은 Solidity 파일 혹은 ABI + Bytecode 업로드

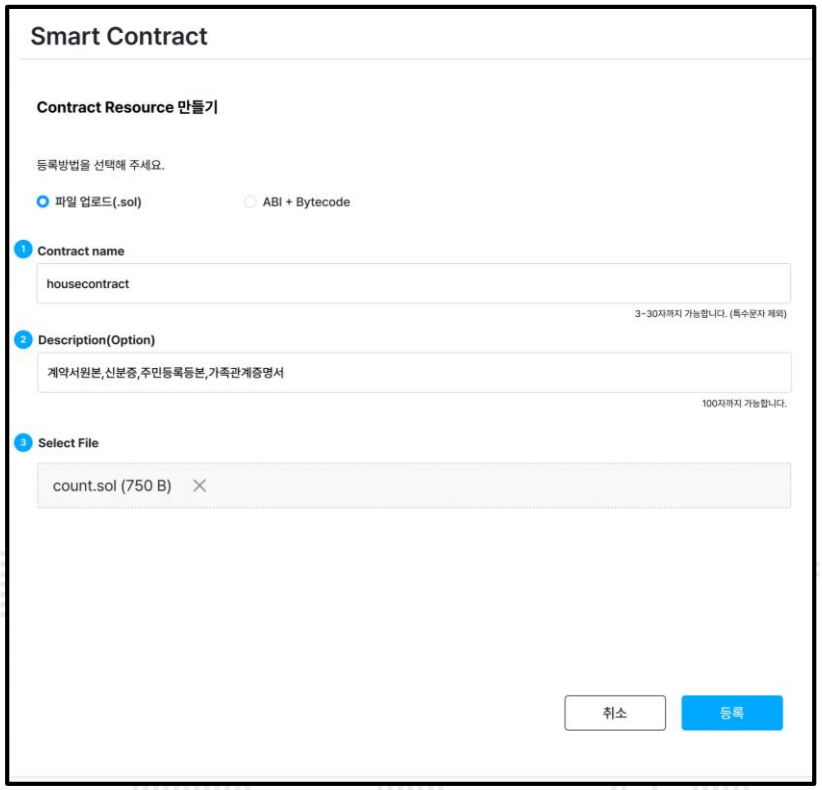


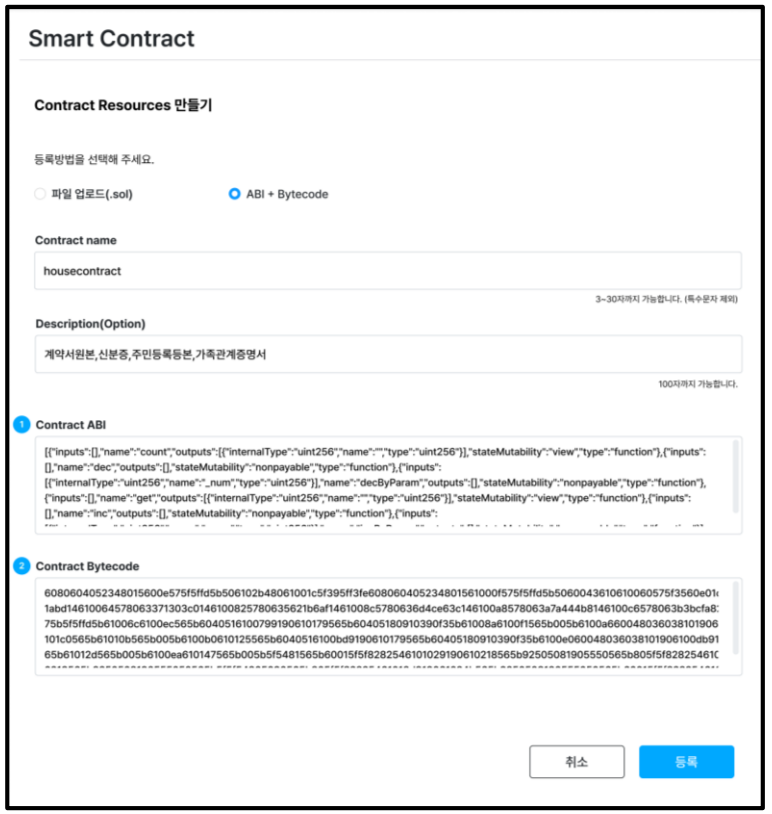


2.2. BESU 메인넷 : Web3 API 활용 개발 방법

스마트 컨트랙트 등록

- 등록할 컨트랙트 이름과 설명 작성
- Solidity 파일 또는 컴파일 된 ABI + Bytecode 업로드

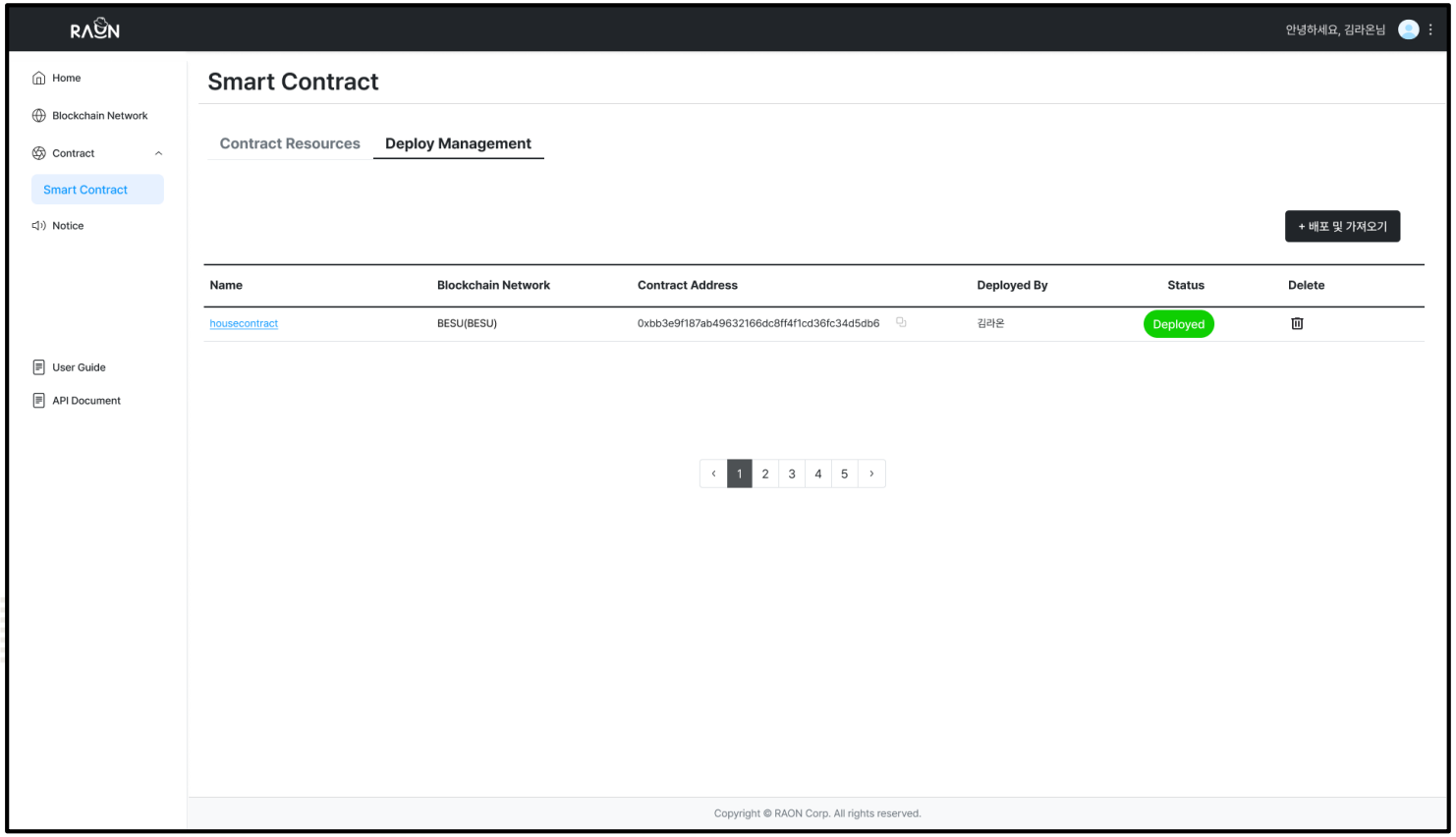




2.2. BESU 메인넷 : Web3 API 활용 개발 방법

스마트 컨트랙트 배포

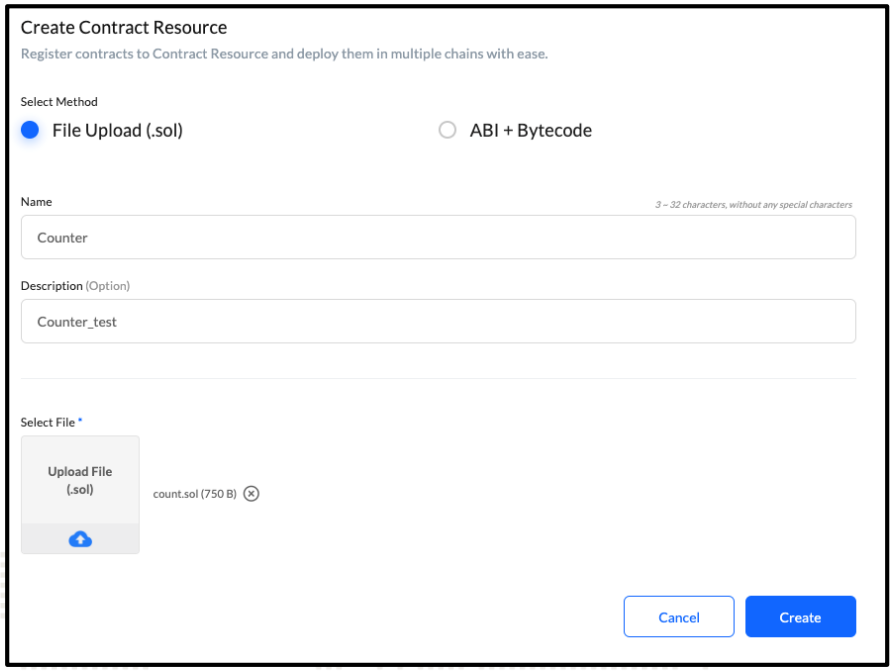
- 등록된 컨트랙트 배포

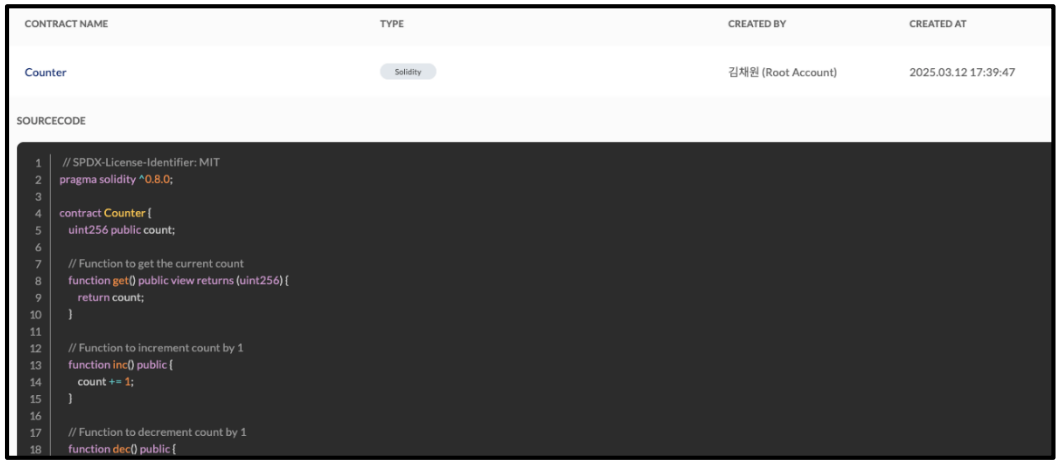


2.3. BESU 메인넷 : 메인넷 기반 서비스 적용 방법

예시) Counter - Counter.sol 의 스마트 컨트랙트 등록

- Counter.sol 이라는 Counter 코드를 구현한 후 스마트 컨트랙트에 등록
- sol 파일 혹은 sol파일을 컴파일한 ABI+Bytecode 업로드

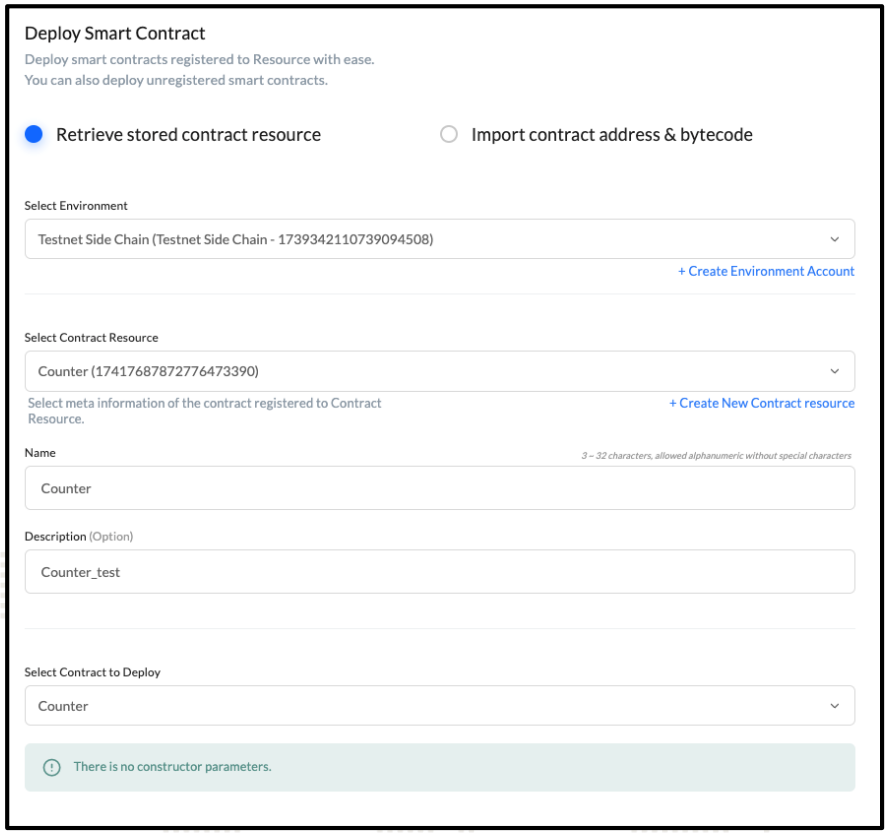


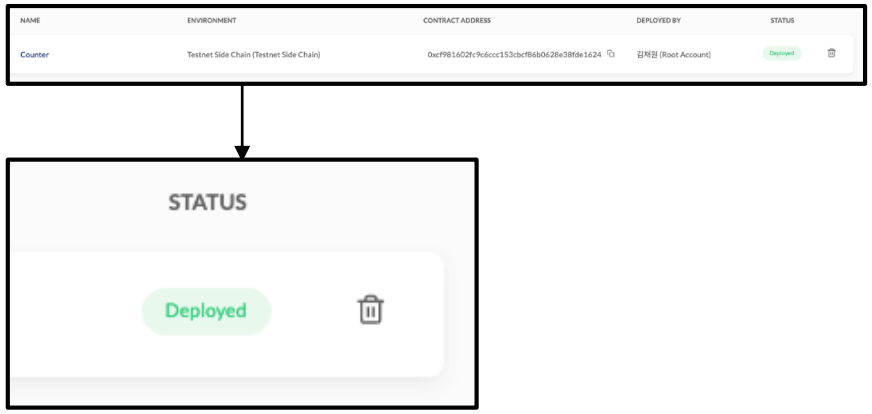


2.3. BESU 메인넷 : 메인넷 기반 서비스 적용 방법

예시) Counter - 등록된 스마트 컨트랙트 배포

- Counter.sol 이라는 Counter 코드를 구현한 후 스마트 컨트랙트에 등록
- sol 파일 혹은 sol파일을 컴파일한 ABI+Bytecode 업로드





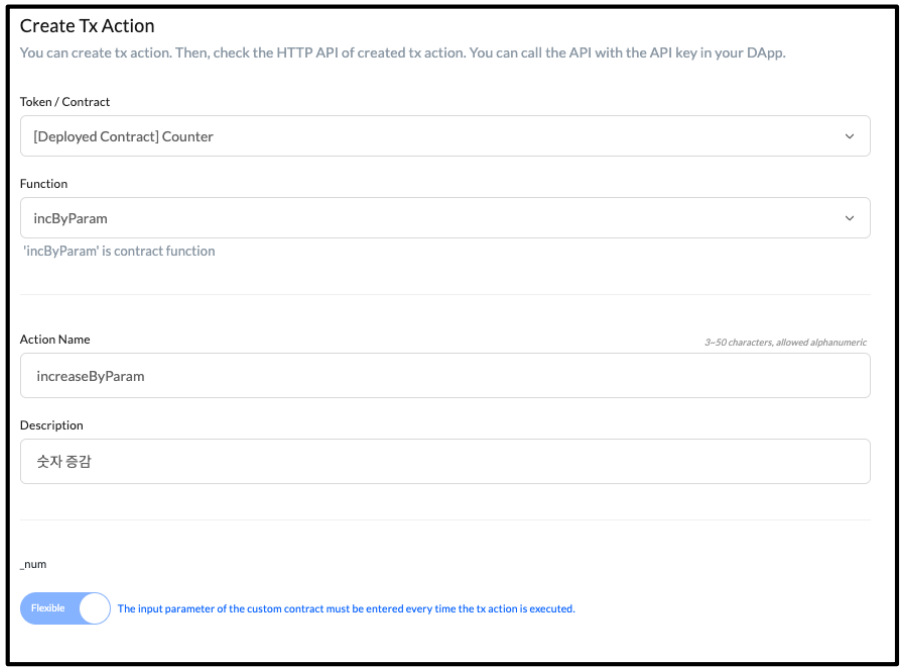
해당 컨트랙트 상태값 배포 완료

2.3. BESU 메인넷 : 메인넷 기반 서비스 적용 방법

예시) Counter - 배포된 스마트 컨트랙트로 트랜잭션 발행

- Counter 의 현재 숫자를 가져오는 getCount 발행

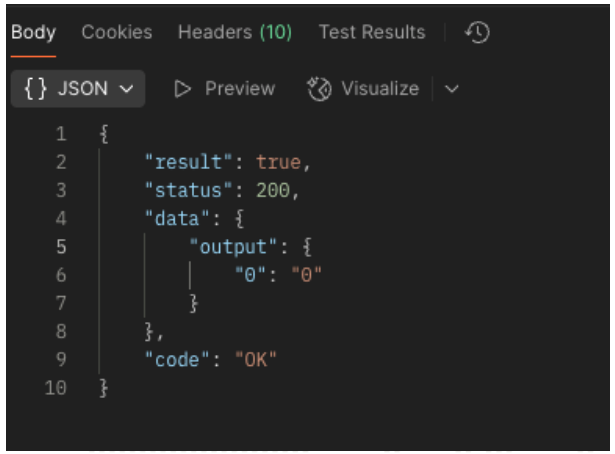
- Counter 의 숫자를 증가시키는 increaseByParam 발행



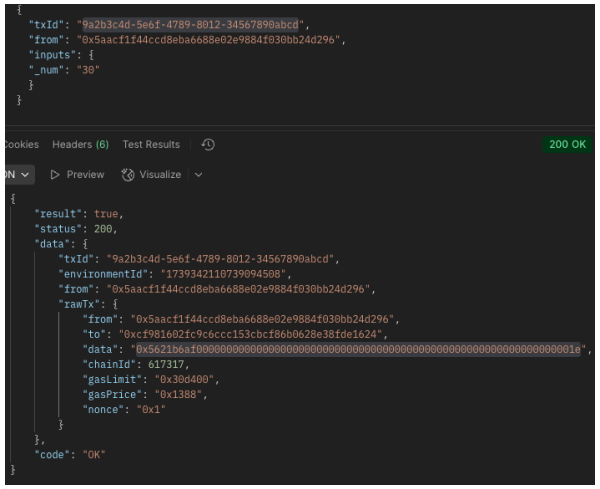
2.3. BESU 메인넷 : 메인넷 기반 서비스 적용 방법

예시) Counter - 트랜잭션 호출

- Api Key 로 AuthToken 을 발급받아 헤더에 등록
- RESTAPI 로 Tx-Action에 맞는 해당 request 데이터 값을 Url로 호출 후 데이터 확인



getCount 로 현재 숫자 확인



increaseByParam 으로 트랜잭션 발행
받은 rawTx 를 EOA키로 서명

request의 서명한 rawTx를 전송하여
응답값 확인 (0 => 30)

감사합니다.

Contact Us

contact@didalliance.org

www.didalliance.org