출입 통제 시스템: 4대 출입 인증 방식 가이드 (SDAC vs ACU 버전 비교)

본 문서는 출입 통제 시스템이 지원하는 4가지 주요 출입 인증 방식(카드, 스마트폰 태그, 지문, QR 코드)을 설명합니다.

가장 큰 특징은 단말기 계층의 선택(`SDAC 버전` 또는 `ACU 버전`)에 따라 이 4가지 인증 방식의 **작동 흐름과 판정 주체**가 완전히 달라진다는 것입니다.

1. SDAC 버전 (옵션 1: ACU 미적용 / Pull 방식)

이 모델에서 리더기는 네트워크에 직접 연결되며, 모든 판정을 중앙 서버에 실시간으로 요청합니다.

- **핵심 원칙:** "Dumb Terminal" - 리더기는 ID만 읽고, **중앙 서버(`Access Control Service`)가 모든 판정**을 수행합니다.
- **통신 흐름:** `리더기` → `API Gateway` → `Access Control Service` (실시간 판정) → `리더기` (문 열림)
- **인증:** 리더기는 `Device Token`(비밀 키)을 사용하여 `API Gateway`에 자신을 인증합니다.

A. 카드 방식 (Pull)

1. 사용자가 'IP 카드 리더기'에 '출입 카드'를 태그합니다.
2. 리더기는 `Card_ID`와 `Device Token`을 `API Gateway`로 전송합니다.
3. `Access Control Service`가 `Redis`에서 `Card_ID`를 조회하여 "GRANT"를 응답합니다.
4. 리더기가 응답을 받고 문을 엽니다.

B. 스마트폰 태그 방식 (Pull)

1. 사용자가 'NFC/BLE IP 리더기'에 스마트폰을 태그합니다.
2. 리더기는 `NFC_ID`와 `Device Token`을 `API Gateway`로 전송합니다.
3. `Access Control Service`가 `Redis`에서 `NFC_ID`를 조회하여 "GRANT"를 응답합니다.
4. 리더기가 응답을 받고 문을 엽니다.

C. 지문 방식 (Pull)

1. 사용자가 'IP 지문 리더기'에 지문을 스캔합니다.
2. 리더기 하드웨어가 로컬 템플릿과 비교하여 `User_ID`를 식별합니다.
3. 리더기는 `User_ID`와 `Device Token`을 `API Gateway`로 전송합니다.
4. `Access Control Service`가 `Redis`에서 `User_ID`의 **2차 정책(시간/구역)**을 판정하여 "GRANT"를 응답합니다.
5. 리더기가 응답을 받고 문을 엽니다.

D. QR 코드 방식 (Pull)

1. 방문객이 'IP QR 리더기'에 QR 코드를 스캔합니다.
2. 리더기는 `QR_ID`와 `Device Token`을 `API Gateway`로 전송합니다.
3. `Access Control Service`가 `Redis` / `PostgreSQL`에서 `QR_ID`의 유효성(일회성, 시간 제한)을 판정하여 "GRANT"를 응답합니다.
4. 리더기가 응답을 받고 문을 엽니다.

2. ACU 버전 (옵션 2: 하이브리드 / Push 방식)

이 모델에서 리더기는 ACU 하드웨어에 종속되며, 모든 판정은 ACU가 로컬에서 수행합니다.

- **핵심 원칙:** "Smart Controller" - `Access Control Service`가 **정책을 ACU에 미리 Push**해 두면, **ACU가 로컬에서 판정**합니다.
- **통신 흐름:** `리더기` → `ACU` (로컬 판정 및 문 열림) → `API Gateway` (사후 로그 전송)
- **인증:** ACU가 `Device Token`을 사용하여 `API Gateway`에 **로그만** 전송합니다.

A. 카드 방식 (Push)

1. **(사전 작업)** `Access Control Service`가 "카드 ID 1A-2B-3C는 문 A 통과 가능" 정책을 `ACU`에 미리 **Push**해 둡니다.
2. 사용자가 '카드 리더기'(ACU에 연결됨)에 '출입 카드'를 태그합니다.
3. `ACU`가 `Card_ID`를 받고, **로컬 메모리**의 정책을 조회하여 "GRANT"를 **즉시 판정**합니다.
4. `ACU`가 직접 문을 엽니다.
5. **(사후 전송)** `ACU`가 `API Gateway`로 "1A-2B-3C 통과 완료" 로그를 전송합니다.

B. 스마트폰 태그 방식 (Push)

1. **(사전 작업)** `Access Control Service`가 "NFC ID 9F-8E-7D는 문 A 통과 가능" 정책을 `ACU`에 미리 **Push**해 둡니다.
2. 사용자가 'NFC 리더기'(ACU에 연결됨)에 스마트폰을 태그합니다.
3. `ACU`가 `NFC_ID`를 받고, **로컬 메모리**에서 즉시 판정합니다.
4. `ACU`가 직접 문을 엽니다.
5. **(사후 전송)** `ACU`가 `API Gateway`로 로그를 전송합니다.

C. 지문 방식 (Push)

1. **(사전 작업)** `Access Control Service`가 "User ID Kim123은 문 A, 9-18시 통과 가능" 정책을 `ACU`에 미리 **Push**해 둡니다. (지문 템플릿 자체도 ACU 또는 지문 리더기에 등록되어야 함)
2. 사용자가 '지문 리더기'(ACU에 연결됨)에 지문을 스캔합니다.
3. 리더기가 `User_ID: "Kim123"`을 식별하여 `ACU`로 전달합니다.
4. `ACU`가 `User_ID`와 **현재 시간**을 **로컬 정책**과 비교하여 "GRANT"를 **즉시 판정**합니다.
5. `ACU`가 직접 문을 엽니다.
6. **(사후 전송)** `ACU`가 `API Gateway`로 로그를 전송합니다.

D. QR 코드 방식 (Push)

1. **(사전 작업)** `Access Control Service`가 "임시 QR ID Temp_ABC는 14~15시 문 A 통과 가능" 정책을 `ACU`에 **Push**합니다.
2. 방문객이 'QR 리더기'(ACU에 연결됨)에 스캔합니다.
3. `ACU`가 `QR_ID`와 **현재 시간**을 **로컬 정책**과 비교하여 "GRANT"를 **즉시 판정**합니다.
4. `ACU`가 직접 문을 엽니다.
5. **(사후 전송)** `ACU`가 `API Gateway`로 로그를 전송합니다.

3. 요약: 아키텍처 버전별 비교

인증 방식	SDAC 버전 (옵션 1)	ACU 버전 (옵션 2)
판정 주체	**중앙 서버 (Access Control Service)**	**ACU 하드웨어 (로컬 메모리)**
작동 방식	Pull (실시간 요청)	Push (사전 동기화)
서버 역할	실시간 판정 (매우 바쁨)	정책 동기화 및 로그 수신 (한가함)

인증 방식	SDAC 버전 (옵션 1)	ACU 버전 (옵션 2)
응답 속도	0.1초 미만 (Redis 캐시)	즉각적 (로컬 판정)
서버 장애 시	**시스템 중단** (문이 열리지 않음)	**정상 작동** (오프라인 안정성)
비용	ACU 비용 $0, 서버 비용 높음	ACU 비용 높음, 서버 비용 낮음