

REMOTE MANAGER

아키텍처 초안

항만 TOS와 Crane PLC 사이의 작업 중계, 크레인 상태 감시, 작업 시퀀스 관리 시스템

이 문서는 REMOTE MANAGER를 하나의 시스템으로 설명한다. 내부에는 TOS 연계, 작업 접수/할당, PLC 상시 통신, 크레인 상태 추적, 충돌/대기 판단, 이벤트/이력 관리 기능이 포함된다.

작성일	2026-05-18
대상	항만 TOS, REMOTE MANAGER, Crane PLC, RCS/운영자, 운영 DB
핵심 전제	Crane PLC polling과 crane status 관리는 작업 유무와 관계없이 상시 수행한다.

1. 정의

REMOTE MANAGER는 TOS의 작업 지시를 받아 크레인 PLC가 이해할 수 있는 작업 데이터로 변환하고, 크레인의 현재 위치/상태/RFID/작업 완료 신호를 상시 확인하며, 작업 진행 상태와 완료 결과를 TOS 및 RCS에 제공하는 중계/제어 시스템이다.

2. 전체 논리 아키텍처



3. REMOTE MANAGER 내부 기능

REMOTE MANAGER는 하나의 시스템으로 보이며, 내부적으로 다음 기능들을 포함한다.



4. 작업 지시 흐름

<p>1</p> <p>TOS가 REMOTE MANAGER에 작업을 지시한다.</p>	<p>2</p> <p>REMOTE MANAGER는 TOS 작업 전문을 내부 표준 작업 정보로 변환한다.</p>	<p>3</p> <p>작업 대상 크레인, 작업 위치, 작업 유형, 컨테이너/트럭/RFID 정보를 확인한다.</p>	<p>4</p> <p>동일 블록, 베이, 로우 또는 작업 영향권이 겹치는 작업이 있는지 판단한다.</p>	<p>5</p> <p>작업 가능하면 Crane PLC에 작업 데이터를 전달한다.</p>	<p>6</p> <p>PLC write 성공 시 작업 할당 이력을 저장하고 TOS에 작업 할당 결과를 보고한다.</p>	<p>7</p> <p>충돌 또는 운영 조건으로 작업이 불가능하면 작업을 대기 상태로 두고, 조건 해제 시 재평가한다.</p>
---	--	--	---	---	---	--

5. 상시 PLC Polling과 상태 관리

PLC polling과 crane status 관리는 작업 지시와 독립적으로 항상 수행된다. 크레인이 idle 상태이거나 TOS 연계가 일시적으로 불안정해도 REMOTE MANAGER는 PLC 상태를 계속 확인해야 한다.

<p>1</p> <p>REMOTE MANAGER는 크레인별로 PLC polling loop를 유지한다.</p>	<p>2</p> <p>목표 주기는 100ms 수준이며, PLC의 위치/속도/작업 상태/RFID/완료 신호를 읽는다.</p>	<p>3</p> <p>최신 상태는 현재 상태 정보로 갱신한다.</p>	<p>4</p> <p>상태가 의미 있게 변경되면 상태 변경 이력을 저장한다.</p>	<p>5</p> <p>RFID 감지, 작업 단계 완료, 작업 완료 조건을 polling data에서 감지한다.</p>	<p>6</p> <p>감지된 업무 이벤트는 TOS에 보고하고 화면에 반영한다.</p>	<p>상시</p> <p>작업 유무와 관계없이 수행한다.</p>
--	---	---	---	--	--	---

6. RFID 및 작업 완료 처리

Job Type [2, 3, 4, 5] 경우에만 RFID Verify

<p>RFID 감지</p> <p>PLC polling data에서 RFID 값을 감지하고 작업과 연결한다. TOS에 RFID 검증을 요청한다.</p>	<p>RFID 검증 성공</p> <p>PLC에 실행 승인 신호를 전달하고 이력을 저장한다. TOS에 RFID 검증 성공을 보고한다.</p>
<p>작업 단계 완료</p> <p>Move/Get/Put 등 단계 완료 조건을 감지하고 작업 상태를 갱신한다. TOS에 단계 완료를 보고한다.</p>	<p>작업 완료</p> <p>job 완료 조건 충족 시 작업을 완료 처리하고 이력을 저장한다. TOS에 작업 완료를 보고한다.</p>
<p>오류 발생</p> <p>PLC 통신 실패, TOS 보고 실패, 검증 실패를 오류 이벤트로 저장한다. 필요 시 TOS/RCS에 오류 상태를 제공한다.</p>	

7. Job Type과 초기 시퀀스

1. Shuffling Get → Move → Put, 중간 Move 단계가 여러 개일 수 있음	2. GateIn Truck/RFID 검증 → ET Container Get → Yard Put → TOS 완료 보고
3. GateOut Yard Get → Truck/RFID 검증 → ET Container Put → TOS 완료 보고	4. DischargeIn Truck/RFID 검증 → YT Container Get → Yard Put → TOS 완료 보고
5. LoadOut Yard Get → Truck/RFID 검증 → YT Container Put → TOS 완료 보고	6. Move 동일 야드 Move
7. Park 크레인 또는 장비를 지정 위치로 이동/대기	8. InternalMove 운영 내부 목적의 이동

8. 작업 상태 관리

RECEIVED TOS로부터 작업을 수신함 작업 수신	WAITING 충돌/운영 조건으로 대기 중 대기 시작	ASSIGNED PLC에 작업 데이터 전달 성공 작업 할당	RFID_WAIT RFID 검증이 필요한 단계 RFID 감지
RUNNING PLC에서 실제 작업 진행 중 작업 단계 완료	COMPLETED 작업 완료 조건 충족 작업 완료	CANCELED 작업 취소 작업 취소	FAILED PLC/TOS/업무 검증 실패 오류 발생

9. 데이터 저장 원칙

작업 정보 작업 유형, 대상 크레인, source/put/move 위치, 컨테이너/트럭 정보를 관리한다.	최신 크레인 상태 작업 판단에 사용할 현재 상태를 빠르게 조회한다.
상태 변경 이력 PLC 상태 변화, 작업 상태 변화, 장애 분석을 위한 이력을 보관한다.	TOS 송수신 전문 TOS 요청/응답 원문과 성공/실패 결과를 추적한다.
업무 이벤트 이력 작업 수신, 할당, RFID, 단계 완료, 완료, 오류 등 전체 흐름을 감사 가능하게 남긴다.	

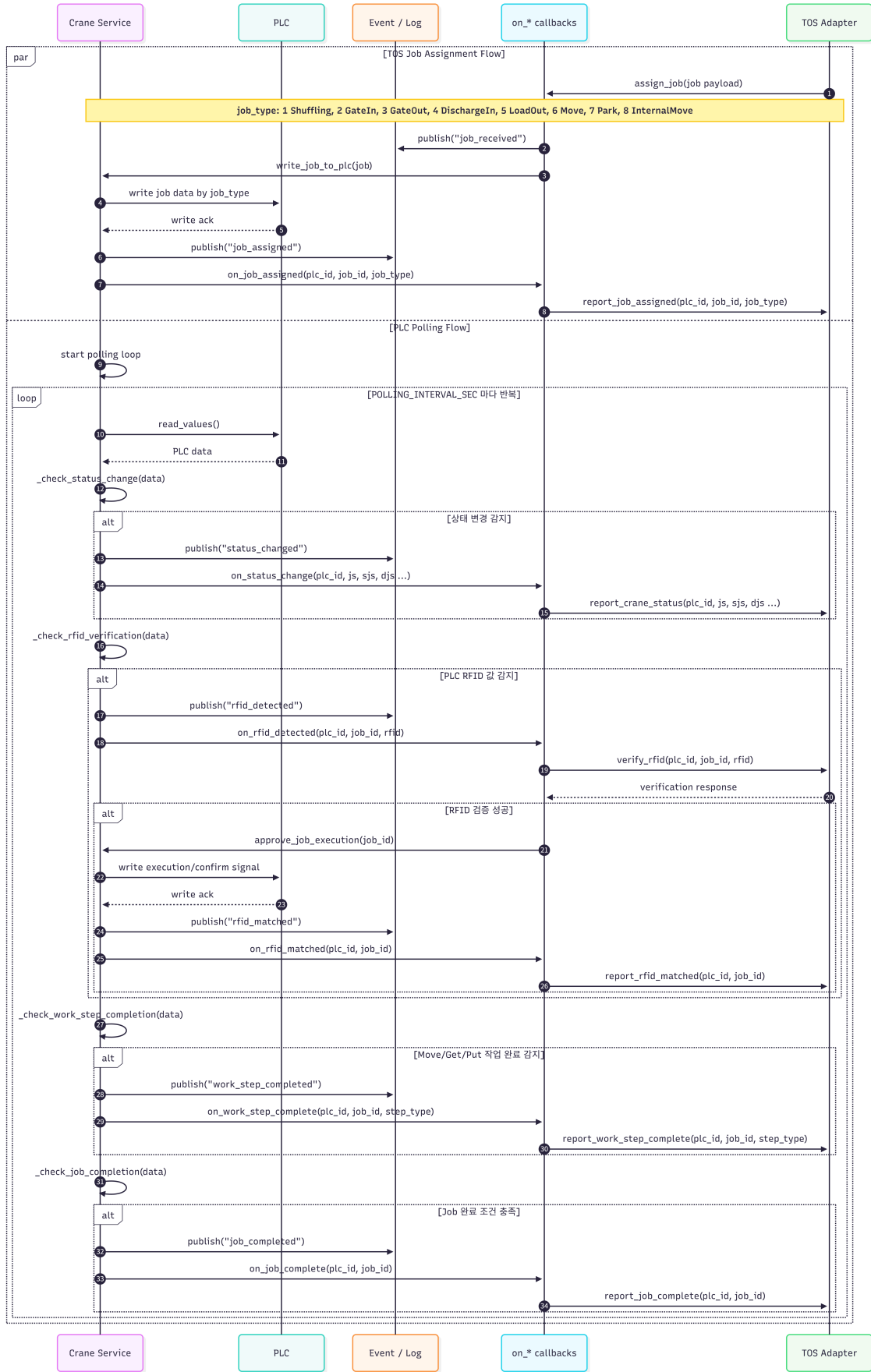
10. DB HA 반영

<p>1</p> <p>운영 DB는 HAProxy + Patroni + etcd + PostgreSQL 구조를 사용한다.</p>	<p>2</p> <p>쓰기 작업은 Primary DB로 전달한다.</p>	<p>3</p> <p>장애 발생 시 Patroni가 Primary/Replica 역할을 재조정하고 HAProxy가 새 Primary로 라우팅한다.</p>	<p>4</p> <p>장애 전환 중에도 PLC polling은 가능한 한 유지하고, DB 기록 실패 이벤트는 retry 대상으로 관리한다.</p>	<p>5</p> <p>상태 이력, TOS 전문, 업무 이벤트 이력은 시간이 지나면 커지므로 파티셔닝 또는 보관 정책이 필요하다.</p>
---	---	--	--	--

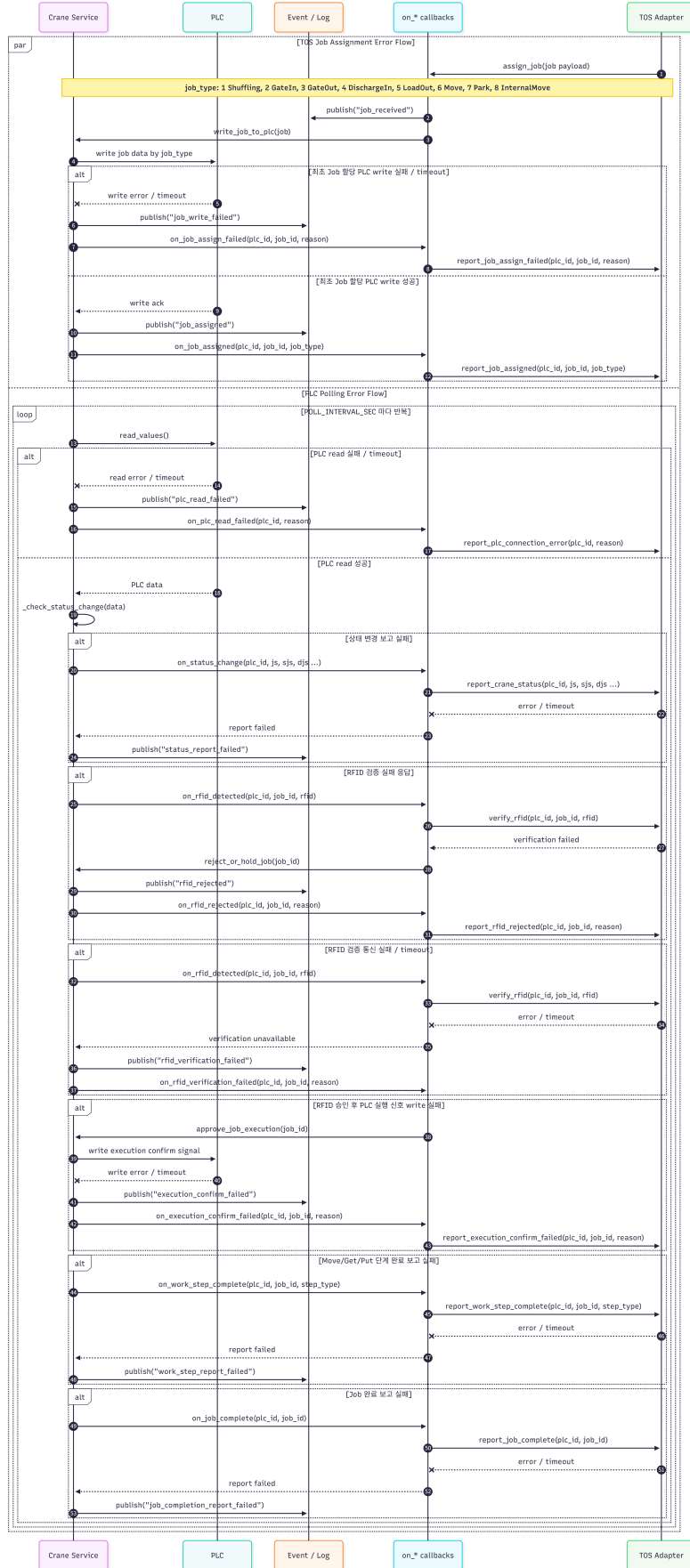
11. 구현 우선순위

<p>REMOTE MANAGER의 PLC 상시 통신과 최신 크레인 상태 갱신을 먼저 구현한다.</p>	<p>작업 단계 완료와 작업 완료 조건을 polling data 기반으로 감지한다.</p>
<p>작업 접수, 할당, 대기, 완료 상태 흐름을 구현한다.</p>	<p>RFID 감지/검증/승인 흐름을 연결한다.</p>
<p>충돌/대기 판단 정책을 block, bay, row, tier 기준으로 구체화한다.</p>	<p>TOS 작업 지시를 내부 표준 작업 정보로 변환하는 연계를 만든다.</p>
<p>TOS 보고 실패, PLC 통신 실패, DB 기록 실패에 대한 재시도/이력 정책을 구현한다.</p>	<p>PLC와의 Fault List를 맞추고 DB Interface를 공유하여 개발·운영 정책을 구체화한다.</p>

12. Job Sequence

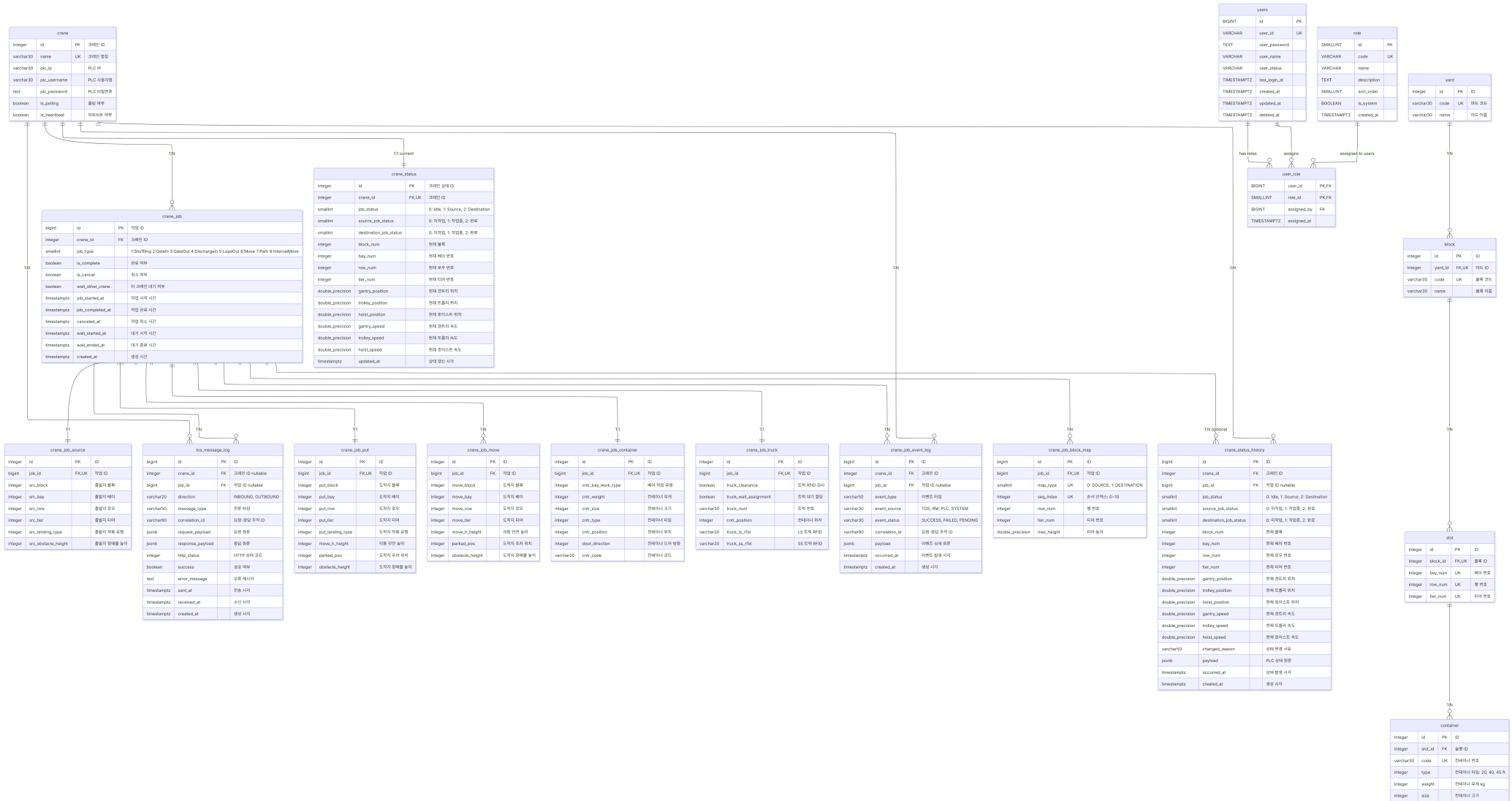


12. Job Sequence - Error Handling

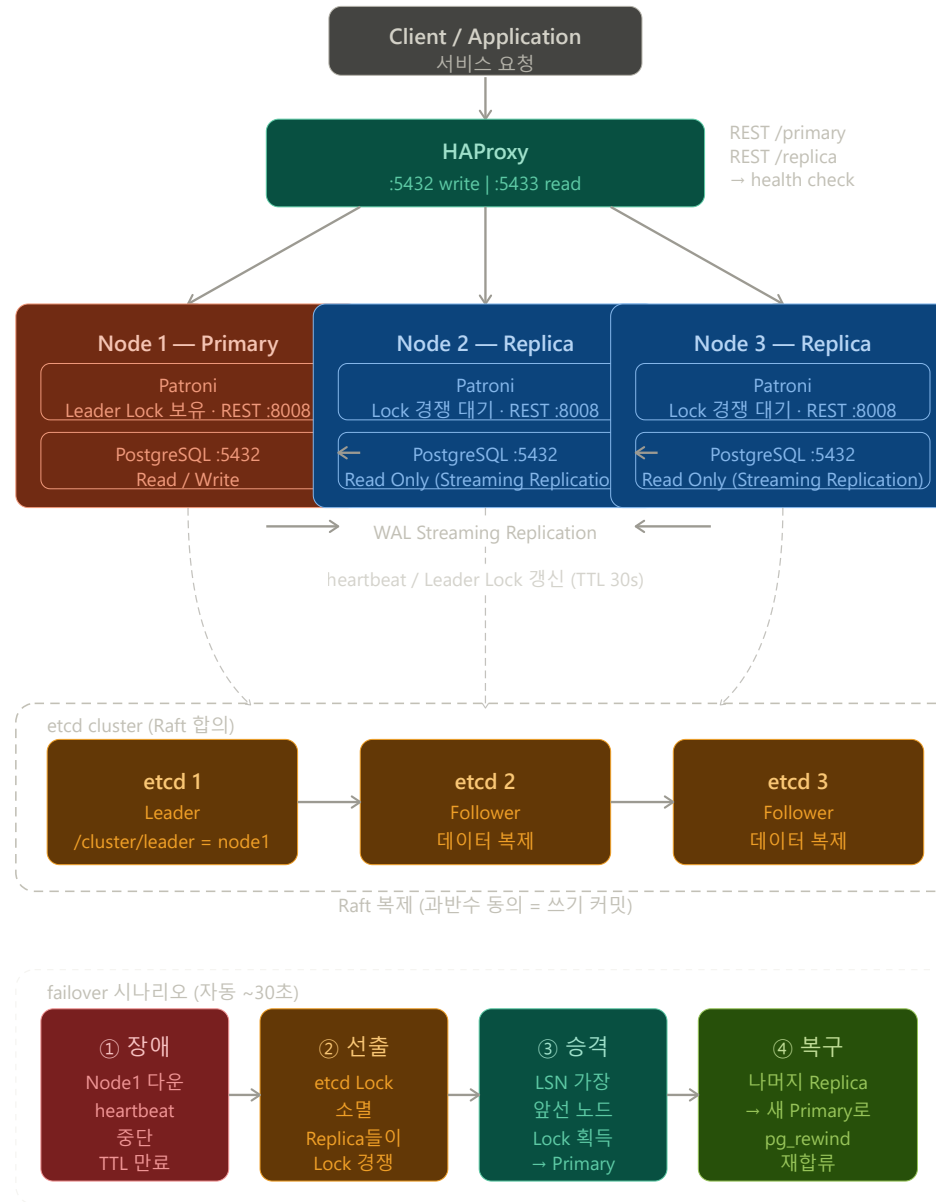


13. Data Base ERD

Crane Domain ERD



14. Data Base Architecture



15. 도입 전 기술 검증 결과

1. OPC UA 라이브러리 성능 테스트 및 개발 언어 선정 - 완료

PLC와 통신하는 애플리케이션 개발 언어로 C#을 선정하였다. C#은 Windows 환경과의 호환성이 우수하고, OPC UA와 같은 산업 표준 통신 라이브러리를 안정적으로 지원한다. 또한 WinForms 및 WPF 기반의 UI 개발이 가능하여 설비 상태 모니터링, 데이터 표시, 알람 처리 등의 기능을 효율적으로 구현할 수 있다.

PLC 통신 애플리케이션은 장기간 안정적으로 동작해야 하며, 향후 설비 추가나 데이터베이스 연동 등 확장 가능성도 고려해야 한다. C#은 객체지향 구조를 기반으로 통신 모듈, 화면 모듈, 데이터 처리 모듈을 분리하여 개발할 수 있으므로 유지보수성이 높다. 또한 .NET 환경에서 제공하는 비동기 처리, 예외 처리, 데이터베이스 연동 기능을 활용할 수 있어 산업용 애플리케이션 개발에 적합하다.

따라서 Siemens PLC와의 표준 통신, 운영 화면 개발, 데이터 수집 및 외부 시스템 연동을 종합적으로 고려하여 C#을 개발 언어로 선정하였다.

2. SNMP(Simple Network Management Protocol) - 완료

SNMP(Simple Network Management Protocol)는 네트워크 장비 및 서버의 상태를 원격에서 조회하기 위한 표준 프로토콜로, 스위치, 라우터, 방화벽, 프린터, UPS, NAS, Windows/Linux 서버 등 다양한 장비의 상태 정보를 수집할 수 있다. SNMP 통신 테스트를 수행하였으며, 이를 통해 향후 크레인 PC의 네트워크 상태, 시스템 자원 사용량(CPU, Memory), 장비 이상 여부 등을 모니터링 시스템과 연계하여 통합 관리할 수 있는 확장 가능성을 확인하였다.

3. Data Base 이중화 - 완료

Patroni 기반 PostgreSQL 고가용성(HA) 클러스터를 Docker Compose 환경에서 구축하고 동작 테스트를 진행했다.

PostgreSQL 3대, etcd 3대, HAProxy 1대로 구성하여 데이터베이스 장애 상황에서도 서비스가 지속될 수 있도록 구성했다.

Primary DB 장애를 강제로 발생시켜 Replica 서버가 자동으로 새로운 Primary로 전환되는 페일오버(Failover) 기능과, HAProxy가 클라이언트 요청을 새로운 Primary로 자동 연결하는 라우팅 전환 기능을 검증했다.

또한 자동 검증 스크립트와 부하 테스트를 통해 데이터 유실 여부, 다운타임, 장애 복구 후 기존 서버의 재합류 과정까지 정상 동작함을 확인했다.

페일오버(Failover)

현재 서비스 중인 Primary(DB 리더)가 장애로 죽었을 때, 대기 중이던 Replica 중 하나를 자동으로 새로운 Primary로 승격(promote)시키는 과정이다.

즉, "죽은 서버 대신 다른 서버가 이어서 서비스하는 것"이다.

라우팅 전환(Routing Switch)

애플리케이션이 접속하는 대상(DB 서버)을 자동으로 새 Primary로 바꿔주는 과정이다.

HAProxy가 그 역할을 한다.