

건설 현장의 돈과 사람을 다루는 백엔드 설계

건설 시공사가 매일 쓰는 ERP의 백엔드를 도메인 단위로 설계했습니다. 코드보다 먼저 정한 것은 **다른 회사의 데이터가 보이지 않게 하는 보안**, **지워지거나 과거가 바뀌지 않게 하는 안전장치**, **어디까지 함께 성공할지 정하는 트랜잭션 경계**였습니다. 그 위에 **안면인식기** 근태부터 **급여·노무비·기성·투입비**까지 현장의 규칙을 구현했습니다.

작성 하성진	역할 도메인 설계 · 백엔드 개발	스택 Spring Boot 3.4 · Java 21	기간 2025.11 ~ 현재 (V2)
-----------	--------------------------	------------------------------------	-------------------------

600

REST 엔드포인트 (컨트롤러 105)

162

JPA 엔티티 · 에러 코드 301

93%

도메인 코드 커밋 직접 작성 (699 / 754)

01 시스템 범위

인사·근태부터 급여, 원가, 자재, 공사, 안전, 전자결재까지 시공사 업무 전반을 다룹니다. 아래는 업무 영역별 API 규모입니다.

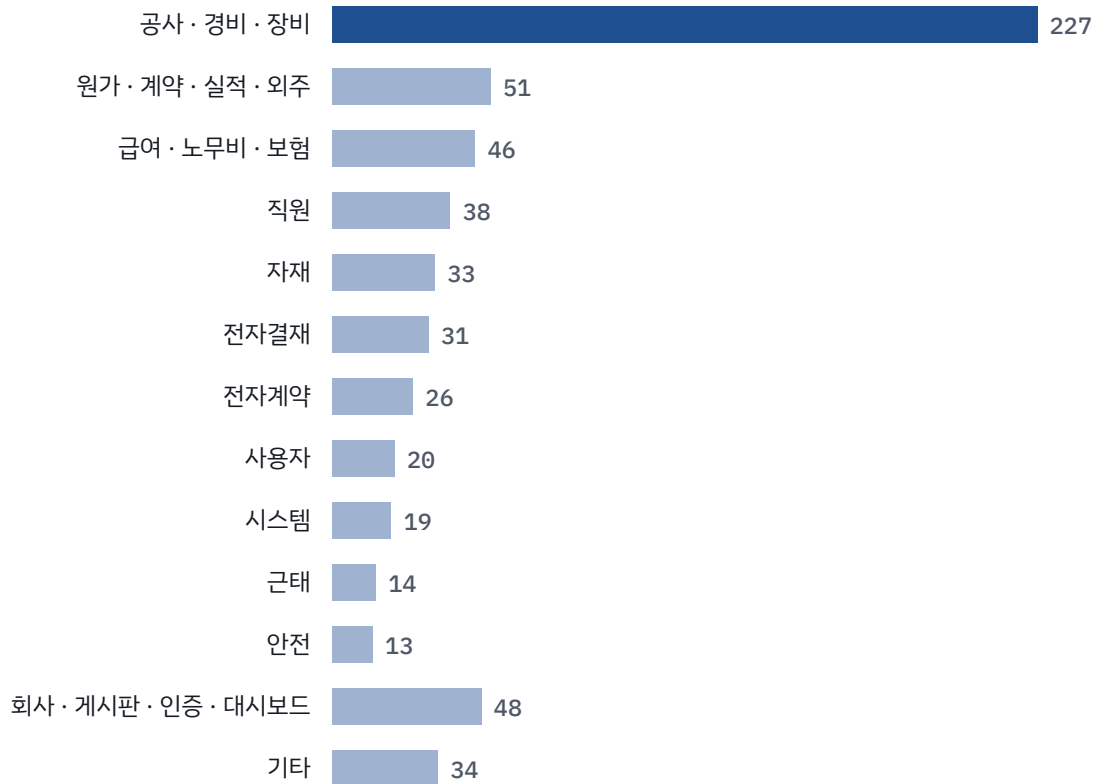


그림 1. 업무 영역(패키지)별 엔드포인트 수. 합계 600. 공사 영역에는 경비·장비·자재비 집계 포함된다.

1.1 도메인 설계 담당

업무 도메인마다 데이터 모델, API 계약, 계산 규칙, 기능명세서·아키텍처 문서를 직접 설계했습니다. 업무 도메인 26개 중 18개에서 커밋의 90% 이상을 직접 작성했습니다.

도메인	본인 / 전체	설계한 내용
원가·계약	151 / 151	기성 차수·투입비 22항목·변경계약·외주비 명세·손익 분석
공사·경비·장비	125 / 125	현장 경비·장비비 6종·레미콘 타설·자재비 월 집계
직원	140 / 146	정규·계약·상용직 3종 신원·인사발령·증명서 발급
근로자·사용자	103 / 112	근태용 통합 식별자·직원-근로자 자동 동기화
근태	80 / 90	안면인식기 실시간 등록·누락 복구·근로자 매칭 (\$05)
급여·노무비	81 / 83	근로자 급여·노무비대장·도급/직영 분류·4대보험·퇴직금
자재	63 / 65	자재 마스터·재고·렌탈 이월 청구·거래처
전자결재·전자계약	45 / 45	결재선·단계 상태 전이·보류/회수·전자계약
시스템·인증	88 / 94	권한 체계·회사 단위 격리·메뉴·문의 관리

개발 저장소 dev 브랜치, 도메인 코드 경로 기준, 머지 커밋 제외. 본인 계정 2개 합산.

02 보안 설계

여러 회사와 수많은 현장이 한 서버를 함께 씁니다. 그래서 보안의 중심은 **"다른 회사·다른 현장의 데이터가 절대 보이지 않게"**와 **"주민번호·계좌 같은 개인정보를 지키는 것"**이었습니다. 개발자가 조건 하나를 빠뜨려도 뚫리지 않도록, 요청이 지나가는 길마다 방어선을 겹겹이 두었습니다.

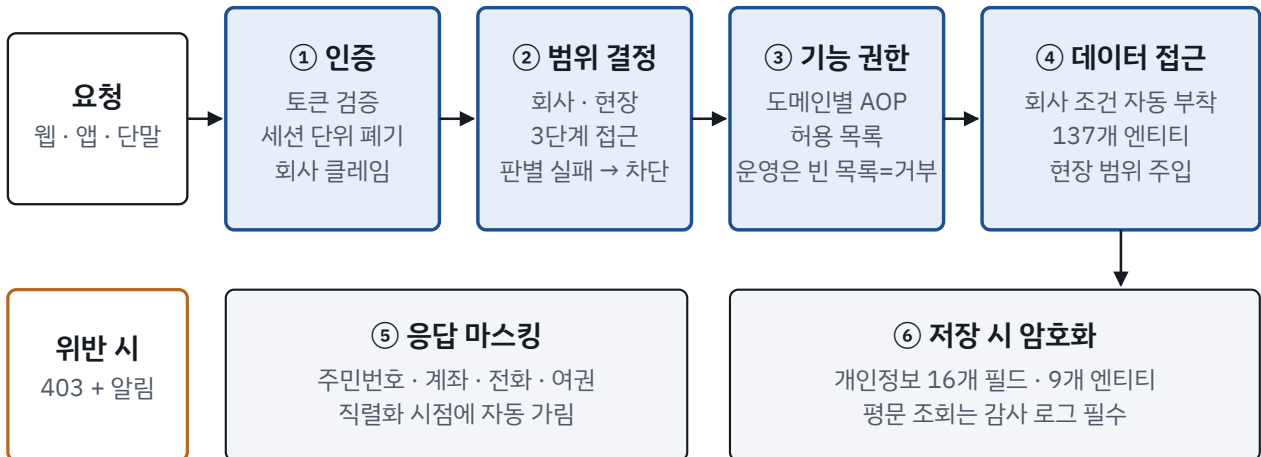


그림 2. 요청이 지나는 6개 방어선. 한 곳이 빠져도 다음 계층이 막는다. 타사 접근 시도는 차단과 동시에 장애 알림으로 보고된다.

설계 대상	막으려는 것 → 설계
세션과 토큰	리프레시 토큰이 새면 만료까지 약용된다 → 갱신할 때마다 기존 토큰을 폐기하고 새로 발급(회전). 기기별로 세션을 관리해 로그아웃은 그 기기만, 비밀번호 변경 시에는 모든 세션을 폐기 한다. 만료·폐기 토큰은 매일 정리한다.
권한 판정 위치	토큰에 적힌 권한을 믿으면 권한을 회수해도 토큰이 살아 있는 동안 유지된다 → 개발자 전체 권한은 요청마다 서버가 다시 판정 한다. 설정이 바뀌면 커밋 직후 캐시를 새로 읽는다.
3단계 접근 범위	본사는 자기 회사 전 현장, 현장 직원은 자기 현장만. 회사 경계를 넘는 요청은 403과 함께 장애 알림으로 남긴다.
도메인별 기능 권한	인사·자재·안전 등 민감 기능은 도메인 단위 어노테이션(AOP 8종)으로 허용 대상을 따로 관리한다. 운영 환경에서는 허용 목록이 비어 있으면 전부 거부(fail-closed) .
회사 간 격리	개발자가 회사 조건을 빠뜨리면 타사 데이터가 나온다 → 조회 시 회사 조건이 DB 레벨에서 자동으로 붙도록 했다(137개 엔티티). 회사 축과 현장 축을 분리해, 현장 범위가 넓어도 회사 필터는 꺼지지 않게 했다. 로그인 요청인데 회사를 판별하지 못하면 차단한다.
회사 판별 근거	사용자의 회사를 소속 현장 코드로 추측하던 방식이, 회사 미지정 현장(근로자의 다수가 소속)에서 오판을 냈다 → 로그인 시 발급하는 토큰의 회사 클레임 을 근거로 바꿨다. 저장 시 회사 코드를 채우는 엔티티 126개가 코드 수정 없이 함께 바로잡혔다.

설계 대상	막으려는 것 → 설계
개인정보	주민번호·외국인등록번호·여권·계좌 등 16개 필드를 저장 시 암호화한다. 응답에서는 직렬화 시점에 자동으로 가리고, 평문이 필요하면 전용 API로만 조회하며 조회 기록을 지울 수 없는 감사 로그 로 남긴다. 로그에는 존재 여부만 남긴다.
파일	버킷은 완전 비공개. DB에는 객체 키만 저장하고, 조회할 때마다 10분짜리 서명 URL을 발급한다. 원본 URL은 응답에 노출하지 않는다.
인증 없는 단말 경로	현장 안면인식기는 토큰을 쓸 수 없다 → 대신 실시간·누락 근태와 단건 등록 경로에서 단말 번호가 등록된 회사에 매핑돼 있는지 검사하고, 매핑이 없으면 거부하고 알린다.

03 데이터 안전 · 정합성 설계

ERP 데이터는 급여와 정산으로 이어집니다. 지워지면 안 되고, 과거가 바뀌면 안 되고, 두 번 들어가면 안 됩니다. 배포가 실패해도 서비스가 멈추면 안 됩니다.

원칙	설계
지우지 않는다	엔티티 162개 중 106개가 논리 삭제다. DB 외래키는 쓰지 않는다 — 연쇄 삭제·제한·NULL 처리 모두 논리 삭제와 충돌하기 때문. 대신 삭제 가능 여부는 서비스에서 검증한다.
누가 언제 바꿨는지 남긴다	저장·수정 시점에 등록자·수정자·시각을 자동 기록한다(144개 / 82개 엔티티). 직원·근로자·급여·결재 등 이력 테이블 21종. 직원·근로자 변경 이력은 본 작업이 커밋된 뒤 별도 트랜잭션으로 저장해, 이력 저장 실패가 본 작업을 되돌리지 않게 했다.
과거는 바꾸지 않는다	시점에 따라 달라지는 규칙은 적용 시작일이 있는 버전으로 쌓는다. 수정은 새 버전 추가, 해제는 해제 버전 기록. 사람이 직접 입력한 분류는 자동 동기화가 덮지 않는다. 퇴직금은 확정 후 퇴직일 변경을 막는다. 결재는 승인 시점의 서명을 결재선에 복사해 고정한다.
두 번 들어가지 않는다	DB 유니크 제약 13개. 논리 삭제 테이블은 유니크 대신 "활성 행 중복 검사"를 서비스에서 한다(존재 검사 81곳, 중복 에러 코드 56개). 급여 계산의 동시 요청 경합은 저장 직전 재조회로 막았다. 단말 누락 근태는 복구 직후 해당 이벤트를 닫아 재전송을 막는다.
제약 위반은 숨기지 않는다	DB가 잡은 정합성 위반은 409로 응답하면서 장애 알림을 보낸다 — 코드 검증 누락이나 경합의 신호이기 때문. 제약 이름으로 업무 에러 코드를 매핑한다.
스케줄러는 혼자서도 안전하게	도메인 예약 작업 10개 전부 실행 여부를 모니터링한다. 건별 예외 처리로 한 건 실패가 전체를 멈추지 않는다. 투입비 자동 저장은 당월만 갱신하고, 금액만 덮고 비고는 보존하며, 실제로 바뀐 경우에만 수정 기록을 남긴다.
스키마는 되돌릴 수 있게	DB 마이그레이션은 추가 방식으로만 바꾼다(삭제는 77개 중 1개). 적용된 파일은 수정하지 않는다. 다른 브랜치와 번호가 겹치면 push 단계에서 차단한다. 배포를 되돌렸을 때 이전 버전이 새 스키마 위에서도 돌아야 하기 때문.

3.1 막을지, 흘려보낼지 – 실패 시 동작을 의도적으로 정했다

실패하면 막는다 (fail-closed)	실패해도 진행한다 (fail-open)
로그인 요청인데 회사 판별 실패 → 차단	로그인 정보가 없는 스케줄러 스레드 → 필터 건너뛰기, 엔티티에 회사 명시
운영에서 기능 허용 목록이 비어 있음 → 전부 거부	변경 이력 저장 실패 → 본 작업은 유지, 알림만
조회 범위가 비어 있음 → 권한 없음 처리	계약 체결 알림 실패 → 계약은 유지, 알림만
새 버전 헬스 체크 실패 → 배포 중단, 기존 유지	근로자 정보 동기화 실패 → 다음 출근 이벤트로 자연 복구

3.2 배포 게이트 – 실패하면 기존 서비스를 유지

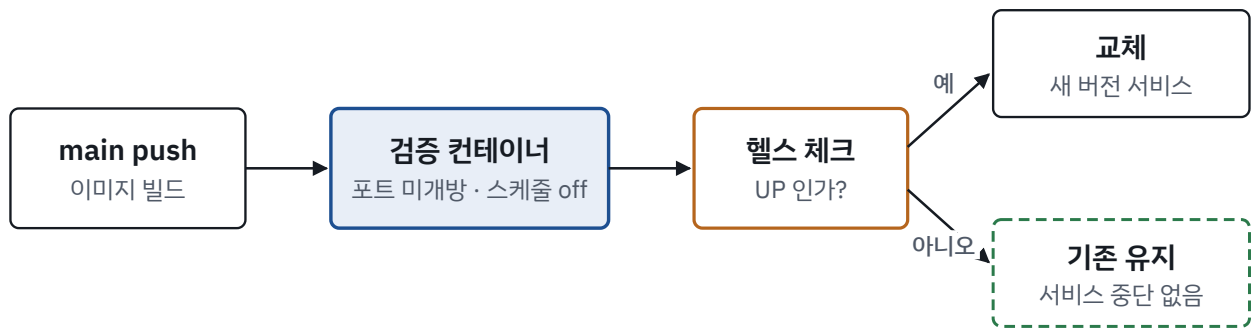


그림 3. 부팅에 실패한 이미지가 무한 재시작되고 이전 컨테이너는 이미 지워져 있던 장애를 계기로 도입했다.

04 트랜잭션 설계

트랜잭션의 경계를 "어디까지 함께 성공하고, 어디서 끊을 것인가"라는 업무 질문으로 정했습니다.

설계	내용과 이유
기본은 읽기 전용	서비스 134개 중 121개(90%)가 클래스 단위 읽기 전용이고, 쓰기 메서드만 쓰기 트랜잭션을 연다. 조회 경로에서 실수로 데이터가 바뀌는 것을 막고, 변경 감지 비용을 줄인다.
건별로 끊는 곳 (8곳)	단말 누락 근태·근로자 동기화 복구는 500건 단위로 돌되 건마다 독립 트랜잭션 이다 – 한 건 오류가 나머지 수백 건을 되돌리면 안 되기 때문. 퇴직금 일괄 산정도 근로자별로 끊는다. 공수 동기화는 독립 커밋이라, 뒤의 투입비 저장에 실패해도 공수는 남는다.
전부 함께 성공하는 곳	엑셀 일괄 등록은 반대로 한 건이라도 실패하면 전체를 되돌린다 – 일부만 반영된 상태를 남기지 않기 위해.
읽기와 쓰기 경로 분리	대시보드 조회가 공수 데이터를 지우고 다시 쓰면서 동시 조회 시 교착이 났다 → 대시보드는 읽기만, 쓰기는 노무비대장·투입비 마감 경로로만 제한했다.

설계	내용과 이유
락 전략	문서번호·계약번호 채번과 계약 상태 변경에만 비관적 락(3곳)을 건다 — 번호 중복과 상태 경합이 실제 위험인 곳. 전자계약은 본문 버전을 대조해 서명 시점에 내용이 바뀌었으면 409로 거부하고, 늦게 끝난 PDF 생성 결과는 버전이 다르면 버린다.
외부 호출은 커밋 뒤에	계약 PDF 생성(외부 함수 호출)과 변경 이력 저장은 커밋 이후 비동기로 처리한다(커밋 후 리스너 5개). 트랜잭션 안에서 외부를 부르면, 롤백돼도 외부 작업은 이미 실행된 상태가 된다.
비동기에도 회사 정보 전파	비동기 스레드에는 로그인 정보가 없어 회사 필터가 동작하지 않는다 → 작업을 넘길 때 회사·현장 범위를 함께 복사해 전달하고, 끝나면 비운다.
누계는 한 트랜잭션에서 다시 계산	기성 차수를 고치면 그 현장의 전 차수를 한 트랜잭션 안에서 순서대로 다시 합산한다. 실제로 값이 바뀐 차수만 갱신된다.

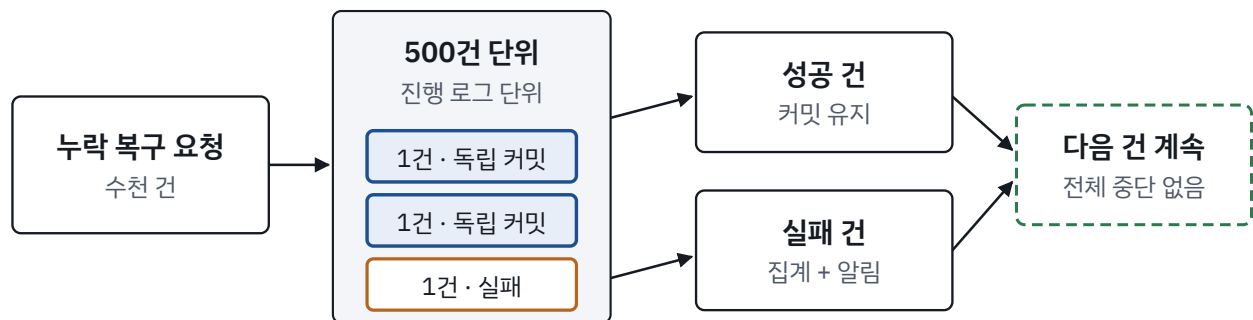


그림 4. 단말 누락 복구의 트랜잭션 경계. 청크는 진행 단위일 뿐이고, 커밋은 건마다 따로 일어난다.

05 현장 근태 — 안면인식기 연동

건설 현장의 출퇴근은 **안면인식기**로 기록됩니다. 이 기록이 공수가 되고, 공수가 급여·노무비·투입비로 이어집니다. 근태가 틀리면 그 뒤의 정산이 모두 틀립니다.

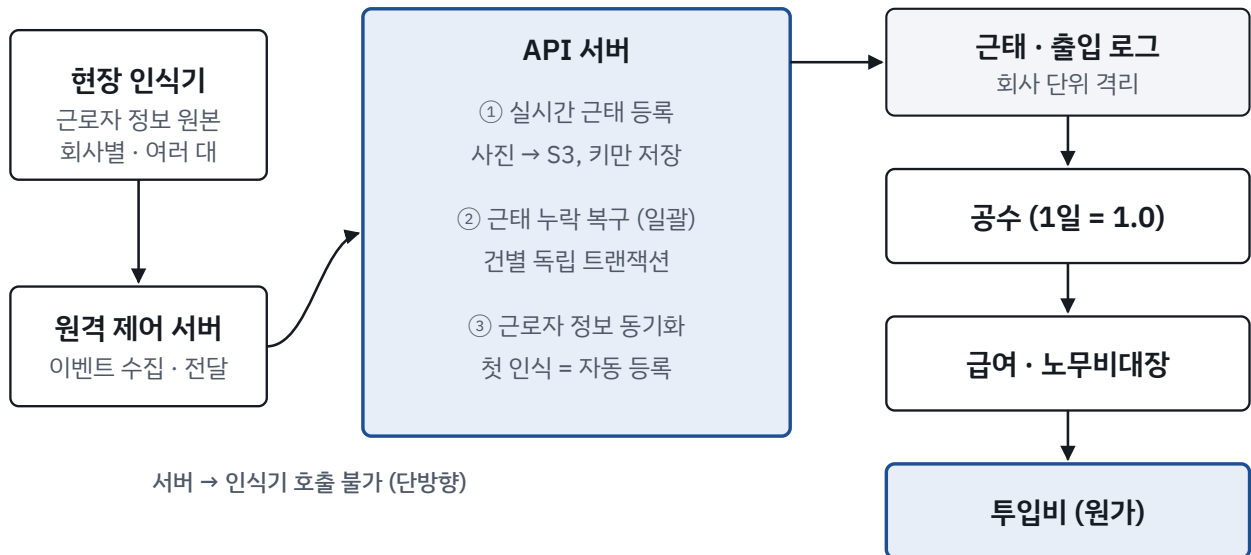


그림 5. 인식기가 원본이고 서버는 사본이다. 서버는 인식기를 호출할 수 없고, 인식기 쪽에서 오는 이벤트만 받는다.

5.1 설계 포인트

과제	설계
원본이 인식기에 있다	근로자 등록·수정은 현장 관리자가 인식기에서 한다. 인식기는 출퇴근 이벤트마다 그 근로자의 현재 정보를 함께 보낸다. 이를 이용해 첫 인식이 곧 서버 자동 등록 이 되게 했다 — 입·퇴사가 잦은 현장에서 사전 등록 없이 스스로 맞춰진다.
중간 프로그램 의존 제거	예전에는 단말 → C# 중계 프로그램 → DB 프로시저였다. 단말이 Spring Boot API를 직접 호출하도록 바꾸고, 출근 사진은 DB에 바이너리로 넣는 대신 S3에 저장하고 키만 남긴다. 조회는 만료되는 URL로 제공한다.
누락 복구는 둘로 분리	근태 기록 누락은 급여에 치명적이므로 일괄 복구 경로를 따로 두고, 건별 독립 트랜잭션으로 한 건 실패가 전체를 멈추지 않게 했다(500건 단위). 근로자 정보 누락은 다음 출근 이벤트로 자연 복구되므로 실패 시 알림만 남긴다.
식별자가 두 개	ERP의 "사람" 번호(영구 1개)와 인식기의 "등록" 번호(인식기마다 발급)를 매핑 테이블로 연결했다. 여러 인식기가 있는 현장에서 한 사람의 기록이 흩어져 급여가 누락될 수 있는 경로를 데이터 흐름 단위로 분석해 설계 문서로 정리했다.
구버전 프로시저의 함정	근로자 등록 프로시저는 분기마다 돌려주는 결과셋 수가 0~3개로 달랐다. 구버전은 순서를 고정 가정해 성공인데도 실패 코드가 나오거나, 여러 명 중 마지막 사람 결과만 남았다. 순서에 기대지 않는 응답 원칙으로 새로 설계했다.
조회 성능	근태 현황 조회가 느린 원인을 실행 계획으로 추적했다 — 이력 테이블 조인의 동적 조건이 인덱스를 무력화해 전체 스캔이 일어났다. 또 장기간 조회에 사진 데이터를 그대로 담아 응답이 무거워지는 문제를 찾아 개선 대상으로 정리했다.

06 도메인 이해 — 시공사의 돈의 흐름

시공사는 대부분 **후불제**로 일합니다. 자기 돈으로 먼저 공사하고, 진행된 만큼 발주처에 청구합니다. 그래서 "얼마를 받을 수 있나(기성)"와 "얼마를 썼나(투입비)"를 같은 달 기준으로 맞춰 보는 것이 현장 손익 관리의 중심입니다.

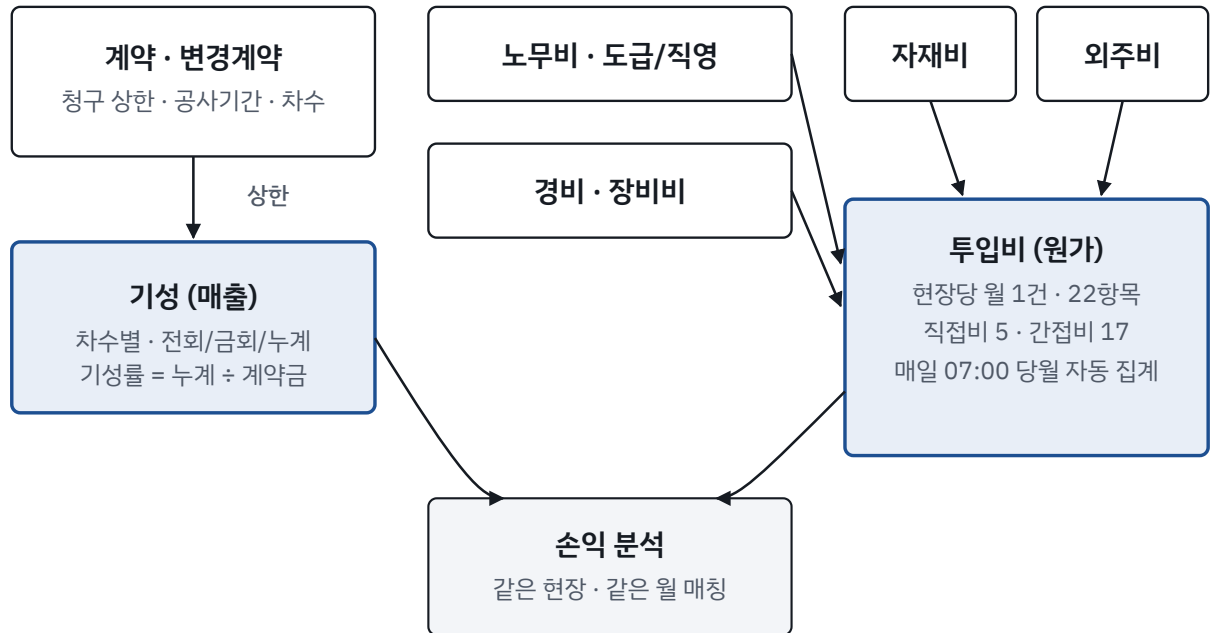


그림 6. 계약이 기성의 상한이 되고, 노무비·자재비·외주비·경비가 투입비로 모인다. 기성과 투입비는 같은 현장·같은 월로 짝지어 손익을 본다.

6.1 시스템에 옮긴 현장 개념

용어	현장에서의 뜻	서버에서 다루는 방식
기성	진행된 만큼 발주처에 청구하는 금액	차수 단위. 금회만 입력받고 전화·누계·기성률은 서버가 계산
투입비	현장에서 실제로 쓴 원가	현장당 월 1건, 고정 22항목. 노무비·자재비·외주비가 자동 연동
변경계약	공사 중 금액·기간이 바뀐 계약	차수 관리. 증감액은 직전 차수 기준, 상위 차수가 있으면 수정 불가
도급 / 직영	외부 업체 인력 / 회사가 직접 고용한 인력	일급 일용직에만 존재하는 구분. 원가에서 따로 집계
공수	하루 노동량. 8시간 = 1.0	근무 시간에서 점심을 빼고 계산, 상한 1.0. 특근은 요일·공휴일별 배율
작업반	실제 시공을 맡는 팀 단위	현장·공사·팀·도급/직영 4개 키. 엑셀 시트 1장 = 작업반 1개로 대응

용어	현장에서의 뜻	서버에서 다루는 방식
임대료 / 손료	빌린 자재 비용 / 회사 자산 사용 대가	같은 계산식 - 일별 현장 보유 수량 × 단가의 합. 반납 전까지 월 이월 청구
현장 계층	본사와 그 아래 현장들	모든 집계의 기준 단위. 본사는 전 현장, 현장 직원은 자기 현장만 조회

07 도메인 규칙을 구현한 사례

7.1 노무비를 도급과 직영으로 나눠 원가에 자동 연동

문제	구버전 시스템은 근로자별 월 노무비를 한 줄로만 계산했다. 그런데 원가(투입비)에는 도급분과 직영분을 나눠 넣어야 했다.
해결	직영분 = 월 노무비 × (직영 공수 ÷ 전체 공수)를 반올림하고, 도급분 = 전체 - 직영분 으로 계산했다. 나머지를 한쪽에 몰아 근로자별 합계가 1원도 어긋나지 않는다. 일급·시급·월급 하위 6개 값과 부모 합계 2개로 저장한다.
결과	노무비대장의 금액이 투입비의 도급·직영 노무비 항목으로 그대로 연동되고, 근로자별 합계가 항상 일치한다.

7.2 과거 정산을 바꾸지 않는 유효일 기반 분류 규칙

문제	팀이나 직종의 도급/직영 분류를 바꾸면, 이미 마감한 지난달 노무비까지 바뀌는 위험이 있었다.
해결	분류 규칙을 적용 시작일이 있는 버전 으로 관리했다. 근무일 기준으로 그 시점의 버전을 찾아 적용한다. 수정은 오늘 날짜 새 버전을 추가하고, 해제는 빈 버전(tombstone)으로 남긴다. 우선순위는 개별 기록 → 현장 규칙 → 공통 규칙 → 팀 기본값. 손으로 입력한 값은 자동 동기화가 건드리지 않는다.
결과	규칙을 언제 바꿔도 과거 근무일의 분류와 마감 금액은 그대로 유지된다.

7.3 기성 누계와 기성률을 서버 계산으로 이관

문제	프론트가 누계를 계산해 보내다 보니 계산 오류가 나고, 중간 차수를 고쳐도 이후 차수에 반영되지 않았다.
해결	요청의 누계 값은 무시하고, 현장의 전 차수를 순서대로 다시 합산한다. N차 누계 = 1~N차 금회 합, 기성률 = 누계 ÷ 계약금. 실제로 바뀐 행만 갱신된다.
결과	어느 차수를 고쳐도 이후 차수까지 일관되게 맞는다.

7.4 대시보드 동시 조회 데드락 제거

문제	당월 인건비 위젯이 조회할 때마다 공수 데이터를 지우고 다시 넣었다. 위젯 여러 개 동시에 부르면서 락이 서로 걸렸다.
해결	대시보드는 마지막 동기화 결과를 읽기만 하도록 바꾸고, 쓰기(동기화)는 노무비대장 진입·투입비 저장·스케줄러 경로로만 제한했다.
결과	조회와 쓰기가 분리되어 동시 조회에서 교착이 생기지 않는다.

7.5 근로자 급여와 4대보험 계산

문제	일급·시급·월급 근로자가 섞여 있고, 공수와 특근 배율에 따라 금액이 달라진다. 보험은 입퇴사가 아니라 적용 요건으로 상태가 바뀐다.
해결	공수 하나를 급여의 단일 출처로 삼았다. 기본급 = 일반 공수 × 단가, 연장은 공수 1을 넘는 부분. 통상시급 제수는 급여 유형별로 분리했다(일급 11.83 · 월급 332.26). 4대보험은 전월·당월 적용 여부로 취득·상실·변경·유지를 판정하고, 국민연금과 건강보험을 따로 본다.
결과	유급주휴수당은 일급 포괄 단가 관행에 맞춰 별도 지급하지 않고, 보험의 회사 부담분(50%)은 투입비에 반영된다.

08 알고 있는 한계와 다음 과제

레거시 시스템과 같은 DB를 함께 쓰면서 생긴 제약이 있습니다. 숨기지 않고 과제로 관리합니다.

항목	현재와 계획
비밀번호 해시	구버전 시스템·단말 동기화 프로시저와 같은 형식을 써야 해서 아직 레거시 해시를 쓴다. 구버전 경로를 걷어낸 뒤 강한 해시로 전환하는 것이 과제다.
개인정보 암호화 방식	구버전 프로시저와 바이트가 같아야 해서 같은 방식을 유지한다. 키를 외부 저장소로 옮기고 방식을 바꾸는 작업은 이관 완료 후 진행한다.
앱 토큰 만료	현장 앱의 사용성 때문에 만료가 없다. 리프레시 토큰 회전 구조는 이미 있어, 만료를 도입할 기반은 마련돼 있다.
단말 인증	단말은 토큰을 쓸 수 없어 회사 매핑 검사로 보완하고 있다. 일괄 동기화 경로의 매핑 검사와 API 키·서명 방식의 인증 모델을 과제로 관리한다.

09 기술과 규모

구분	내용
언어 · 프레임워크	Java 21 · Spring Boot 3.4 · Spring Security(JWT)
데이터	MySQL(MariaDB) · JPA · QueryDSL · Flyway(마이그레이션 76개)
인프라 · 배포	AWS ECR · EC2 · S3 · GitHub Actions(배포 게이트)
운영	Sentry(예외 4계층 기준 알림) · Slack
품질	테스트 파일 118개 · 코드 리뷰(Codex) · AI 에이전트 기반 검증 파이프라인
규모	컨트롤러 105 · 엔드포인트 600 · 엔티티 162 · 에러 코드 301 · Java 파일 1,568

2026.10 저장소 실측. 커밋 수는 개발 저장소 dev 브랜치 기준(머지 커밋 포함).