

SyncSDK™ & Core Modules : 자율 코딩 런타임 및 6대 엔터프라이즈 공통 모듈

요구사항 명세를 바탕으로 코드 초안을 만들고 컴파일 오류를 스스로 점검해 고치는 코딩 런타임과, 기업 시스템에 바로 끼워 쓰는 6가지 핵심 공통 모듈입니다.

EXECUTIVE SUMMARY · 도입 핵심 방향

복잡한 수작업과 부서 간 대기 시간을 줄이고, 현업 담당자가 쉽고 안전하게 업무를 처리할 수 있는 실무 중심 환경을 제안합니다.

01. 한눈에 보는 도입 기대 지표

Key Highlights

4-Cycle

빌드 오류 자동 점검·수정
컴파일 실패 시 로그 분석 재수정

멀티 엔진

코딩 엔진 자유 전환(SPI)
Antigravity / OpenSWE 선택

6대 모듈

검증된 공통 기능 라이브러리
AI·감사·NL2SQL·배포·RAG·코어

Git 격리

안전한 별도 브랜치 작업
개발자 코드 리뷰(PR) 후 반영

02. 도입 부서 주요 고민 및 SyncSDK™ & Core Modules 해결 방안 조건표

Requirement & Solution Mapping

업무 영역	현업에서 자주 겪는 고민	SyncSDK™ & Core Modules 해결 방법
단순 기능 수정 대기	화면에 입력 칸 하나를 추가하거나 문구를 바꾸려 해도 개발팀 일정이 밀려 1~2주를 기다려야 함	요구사항을 입력하면 관련 파일들을 찾아 수정 코드 초안과 Git PR을 자동으로 준비해 개발자 작업 시간 단축
자동 생성 코드 품질	AI가 짜준 코드를 그대로 넣으면 컴파일 에러가 나거나 기존 코드 규칙과 맞지 않을까 걱정됨	코드를 만든 뒤 실제로 빌드 테스트를 돌려보고, 에러가 나면 최대 4번까지 스스로 고친 뒤 통과한 코드만 전달
중복 기능 개발 방지	프로젝트마다 AI 연동, 감사 로그, 자연어 SQL, 문서 검색(RAG) 기능을 매번 따로 개발함	Java 25 / Spring Boot 3.5 표준으로 만들어진 6대 공통 모듈을 메이븐(Maven) 의존성 추가만으로 바로 사용
특정 AI 도구 종속 방지	특정 코딩 AI 도구 하나에만 맞춰 개발해 두면 나중에 더 좋은 엔진이 나왔을 때 바꾸기 어려움	표준 인터페이스(SPI) 구조로 설계되어 설정만 바꾸면 Antigravity, OpenSWE 등 원하는 엔진으로 전환 가능

Quick Access

온라인 문서 및 체험 안내

기술 문서 포털(<https://doc.empasy.com/sdk/>) 및 통합 포털(<https://sync.empasy.io>)에서 상세 정보를 확인하실 수 있습니다.

라이브 데모: <https://verse-admin.empasy.io>

도입 배경 및 기존 업무 방식과의 비교

현재 업무에서 반복되는 불편함을 짚어보고, 도입 후 업무 흐름이 어떻게 간편해지는지 비교합니다.

01. 기존 업무 환경에서 자주 겪는 4가지 어려움

Current Pain Points

왜 기존 방식으로는 업무 속도와 품질을 동시에 높이기 어려울까요?

- 개발팀 업무의 상당 부분이 새로운 기술 개발이 아니라 기존 화면의 필드 추가, 문구 변경, 단순 API 수정에 쓰입니다.
- 일반 코딩 보조 도구는 코드 조각만 보여줄 뿐, 프로젝트 전체에서 어떤 파일들이 함께 바뀌어야 하는지 놓치기 쉽습니다.
- 여러 하위 프로젝트에서 감사 로그, AI 호출, 벡터 검색 코드를 각자 짜다 보니 품질 차이가 나고 보안 패치도 번거롭습니다.
- AI가 작성한 코드가 운영 브랜치에 바로 섞이면 위험하기 때문에, 반드시 격리된 공간에서 검증하는 절차가 필요합니다.

02. 기존 처리 방식(AS-IS) vs SyncSDK™ & Core Modules 도입 후(TO-BE) 흐름 비교

Process Transformation

기존 수작업 중심 방식 (AS-IS)

반복 수작업 · 대기 발생

- 요구사항 티켓 접수**
현업이 화면/기능 수정 요청서 등록
- 관련 파일 수동 탐색**
개발자가 프론트·백엔드·DB 파일 위치 파악
- 수작업 코딩 및 디버깅**
코드 수정 후 로컬 빌드 오류 수동 해결
- 산출물 문서 별도 작성**
배포 후 설계서·테이블 정의서 수기 업데이트

SyncSDK™ & Core Modules 도입 후 (TO-BE)

간편 처리 · 안전 검증

- 명세 입력 및 파일 추론**
요구사항을 분석해 수정이 필요한 파일 목록 도출
- 격리 브랜치 코드 생성**
본래 코드에 영향 없는 작업 브랜치에 패치 작성
- 4-Cycle 빌드 자동 검증**
컴파일·테스트 실행 후 오류 시 스스로 수정
- 개발자 Diff 검토 및 승인**
검증 통과 코드와 SI 산출물을 확인 후 병합

03. 핵심 추진 방향 요약

Key Direction

방향 01 · 쉬운 사용성

복잡한 명령어 없이 직관적 처리

개발 지식이 없는 현업 담당자도 친숙한 웹 화면과 일상적인 언어로 원하는 작업을 바로 실행할 수 있습니다.

방향 02 · 안전한 통제

사람이 확인하고 언제나 복구

중요한 변경은 담당자가 눈으로 확인한 뒤 반영하며, 변경 직전 상태를 저장해 두어 실수가 생겨도 바로 되돌립니다.

방향 03 · 비용 효율화

반복 인건비 및 AI 요금 절약

수작업 공수를 줄이고, 캐시 재사용과 사내 경량 모델을 활용해 운영 비용 부담을 낮춥니다.

목표 시스템 구성도 (3-Layer 아키텍처)

사용자 화면층 · 애플리케이션 및 AI 엔진층 · 데이터 및 인프라층의 3개 계층으로 분리된 안정적인 구조입니다.

01. SyncSDK™ & Core Modules 3-Layer 통합 아키텍처 다이어그램

Target Architecture



02. 기존 사내 시스템과의 유연한 연동 구조

Seamless Integration

연동 구분	지원 방식 및 표준 규격	도입 고객사 이점
단독 도입 및 단계적 확장	필요한 제품 1종만 먼저 단독 도입 가능하며, 추후 타 Sync 제품군과 표준 MCP로 즉시 연결	초기 도입 부담 없이 가장 급한 업무부터 순차 적용 가능
사내 기간제(ERP/SSO/DB) 연동	표준 REST API(/api/v1/...), OAuth2/JWT 통합 인증, JDBC 표준 DB 연결 지원	기존 운영 중인 사내 시스템을 갈아엎지 않고 그대로 연결해 활용

핵심 작동 프로세스 및 4대 주요 기능

실제 업무가 어떤 순서로 처리되는지와 실무에서 바로 체감할 수 있는 4가지 핵심 기능을 안내합니다.

01. 4단계 업무 처리 프로세스 (Step-by-Step)

End-to-End Workflow

STEP 01**요구사항 명세화 및 수정 대상 파일 탐색**

소요시간: 1~2분

기능 추가나 수정 요청이 들어오면 코드 구조를 분석해 프론트엔드, 백엔드, DB 중 어디를 고쳐야 할지 찾아냅니다.

STEP 02**독립된 Git 작업 브랜치에 수정 코드 작성**

소요시간: 2~3분

운영 중인 원래 코드를 건드리지 않고 별도의 작업 공간(Git Worktree)을 만들어 전사 코딩 규칙에 맞게 코드를 작성합니다.

STEP 03**실제 빌드 실행 및 4-Cycle 자동 오류 수정**

소요시간: 자동 검증

작성된 코드를 실제로 컴파일해 보고, 만약 문법이나 타입 오류가 나면 에러 로그를 읽어 스스로 코드를 고쳐 다시 빌드합니다.

STEP 04**산출물 생성 및 개발자 코드 리뷰 후 반영**

소요시간: 개발자 확인

빌드를 통과한 코드 비교 화면(Diff)과 함께 테이블·API 설계 문서를 정리해 개발자에게 제출하고, 승인 시 반영합니다.

02. SyncSDK™ & Core Modules 4대 핵심 기능 상세

Core Capabilities

핵심 기능 01**엔진을 골라 쓰는 멀티 코딩 런타임 (Multi-Engine SPI)**

하나의 고정된 방식에 묶이지 않고 프로젝트 성격에 따라 Antigravity 엔진이나 OpenSWE 엔진을 선택·교체하여 사용할 수 있습니다.

핵심 기능 02**컴파일까지 확인하는 4-Cycle 자가치유 빌드**

그럴듯한 코드 텍스트만 만드는 것이 아니라 실제 빌드 도구를 돌려 통과 여부를 확인하므로 개발자의 디버깅 수고를 덜어줍니다.

핵심 기능 03**바로 가져다 쓰는 6대 엔터프라이즈 공통 모듈**

AI 에이전트(ai), 감사 기록(audit), 안전 SQL(nl2sql), 무중단 배포(deploy), 문서 검색(rag), 공통 코어(core)를 라이브러리로 제공합니다.

핵심 기능 04**코드 수정과 동시에 만들어지는 SI 4대 산출물**

코드만 바꾸고 문서가 옛날 버전으로 남지 않도록 요구사항정의서, 아키텍처설계서, 테이블정의서, 테스트계획서 초안을 함께 갱신합니다.

주요 화면 구성 및 기존 방식과의 비교

실무 담당자가 사용하는 주요 화면 구성과 기존 수작업·타사 도구 대비 달라지는 점을 정리했습니다.

01. 실무자용 주요 화면 및 메뉴 구성

Key Screens & Menus

화면 01

/spec-studio/engine

멀티 코딩 엔진 선택 및 실행 콘솔

작업 특성에 맞는 코딩 엔진을 선택하고 파일 변경 범위와 빌드 진행 상태를 실시간으로 확인합니다.

화면 02

/spec-studio/healing

4-Cycle 빌드 검증 및 자가수정 로그 뷰

컴파일 및 테스트 실행 결과와 오류 발생 시 자동으로 수정한 내역을 단계별로 보여줍니다.

화면 03

/spec-studio/diff

코드 Diff 비교 및 Git PR 승인 화면

수정된 모든 파일의 변경 전후 코드를 한 줄씩 비교하고 원격 저장소 반영을 승인합니다.

화면 04

/spec-studio/artifacts

SI 표준 산출물 자동 생성 및 다운로드

작업 내역을 바탕으로 생성된 설계서와 검수 문서를 내려받아 프로젝트 산출물로 활용합니다.

02. 기존 운영 방식 대비 SyncSDK™ & Core Modules 비교표

Before vs After Comparison

비교 항목	기존 수작업 · 일반 도구 환경	SyncSDK™ & Core Modules 도입 환경
일반 코딩 AI 도구와의 차이	개발자가 에디터에서 코드 조각을 복사해 수동 테스트	프로젝트 전체 문맥 파악 + 독립 브랜치 수정 + 빌드 검증까지 일괄 수행
생성 코드 오류 검증	사람이 직접 실행해 보며 컴파일 에러 수정	시스템이 자동으로 빌드를 돌려보고 에러 시 최대 4회 자가 수정
공통 기능(감사/RAG/NL2SQL) 구현	프로젝트마다 개발자가 각각 새로 구현	검증된 6대 공통 모듈(sync-module-*)을 즉시 장착해 사용
설계 문서 업데이트	바쁜 일정으로 코드만 고치고 설계 문서는 방치	코드 변경 내역에 맞춰 4대 표준 산출물 초안 자동 생성

보안 관리 대책 및 운영 비용(TCO) 절감 방안

기업 내부 데이터를 안전하게 보호하는 보안 체계와 AI-인프라 운영 비용을 합리적으로 줄이는 방안입니다.

SECURITY & COST OPTIMIZATION

사내 폐쇄망·전용 클라우드 설치로 데이터 유출 걱정을 덜고, 사내 모델·캐시 활용으로 고정비 안에서 경제적으로 운영합니다.

01. 엔터프라이즈 3대 보안 및 안전 관리 대책

Security & Governance

보안 대책 01

원본 저장소와 분리된 Git Worktree 격리 실행

모든 자동 코딩 작업은 격리된 별도 브랜치에서만 이루어지며, 권한 있는 개발자의 승인 없이는 메인 브랜치에 합쳐지지 않습니다.

보안 대책 02

전사 코딩 규칙 및 보안 가이드라인 강제

Lombok 필수 적용, 단일 API URL 규칙, 하드코딩 금지 등 사내 표준 아키텍처 규칙을 프롬프트와 검증 단계에서 함께 확인합니다.

보안 대책 03

폐쇄망 사내 Git(GitLab/Gitea) 및 내부 모델 연동

외부 클라우드로 소스코드를 보내지 않고 사내 Git 서버와 내부 GPU 코딩 모델을 연결해 핵심 지적재산권을 보호합니다.

02. 운영 비용(TCO) 및 리소스 절감 비교

Cost Optimization Analysis

비용 항목	기존 방식 (수작업 · 외부 유료 API 전면 의존)	SyncSDK™ & Core Modules 최적 운영안 (자동화 · 캐시/사내 GPU 결합)
단순 기능 수정·유지보수 인건비	작은 화면·필드 수정에도 시니어 개발자 공수 지속 투입	초안 코드 작성과 기본 빌드 검증을 자동화해 개발자 리류 시간만 소요
AI 기능 내재화 개발비	기존 시스템에 AI-RAG·NL2SQL을 붙이려면 수개월 연구개발 필요	6대 공통 모듈 적용으로 기존 Spring 프로젝트에 빠르게 AI 기능 탑재
산출물 문서 작성 공수	감리·검수 대비용 설계 문서 수작업 작성에 많은 시간 소모	코드 기반 산출물 초안 자동 생성으로 문서화 부담 경감

03. 고객사 환경에 맞춘 3가지 배포(설치) 옵션

Deployment Options

옵션 A · 사내 구축형

온프레미스 (폐쇄망)

외부 인터넷과 분리된 고객사 자체 서버 및 사내 GPU 환경에 독립 설치하여 보안 규정을 준수합니다.

옵션 B · 전용 클라우드

고객사 전용 VPC

AWS, Azure, NCP 등 고객사 전용 격리 네트워크 안에 컨테이너(Docker/K8s) 기반으로 신속하게 구축합니다.

옵션 C · 하이브리드

사내 DB + 클라우드 연동

민감한 고객 원장은 사내에 두고, 필요한 연산 기능만 암호화 통신으로 연결해 초기 구축 부담을 낮춥니다.

단계별 도입 추진 일정(WBS) 및 산출물 안내

사전 검증(PoC)으로 실효성을 먼저 확인한 뒤 본 시스템 구축과 현업 이관까지 안전하게 진행합니다.

Implementation Roadmap

01. 3단계 표준 도입 일정 및 주요 과업 (WBS)

추진 단계	소요 기간	주요 수행 과업 내용	제공 산출물
1단계: 공통 모듈 적용 및 환경 구성	1~2주차 (2주)	사내 Maven 저장소에 6대 공통 모듈 등록 · Git 작업 브랜치 및 빌드 파이프라인 연결	SDK 연동 가이드 · 환경 구성서
2단계: 사내 표준 규칙 및 MCP 도구 확장	3~5주차 (3주)	사내 코딩 컨벤션 주입 · 핵심 레거시 API의 MCP 도구 어노테이션 적용 및 테스트	커스텀 도구 명세서 · 검증 리포트
3단계: 개발팀 교육 및 실무 적용	6~8주차 (3주)	개발자·아키텍트 대상 Spec Studio 및 SDK 활용 실습 · 유지보수 티켓 시범 운영	개발자 매뉴얼 · 완료보고서

02. 단계별 품질 점검 기준 (Quality Gate)

Quality Assurance

GATE 01 · 착수/설계

요구사항·아키텍처 점검

고객사 업무 환경과 연동 대상 API, 보안 요건을 사전 확인하고 설계서를 확정합니다.

GATE 02 · 실데이터 연동

실환경 동작 검증

가짜 데이터가 아닌 실제 테스트 DB와 연동하여 기능 동작과 데이터 정합성을 확인합니다.

GATE 03 · 통합 테스트

성능·권한·복구 리허설

역할별 접근 권한, 예외 상황 시 스냅샷 복원, 동시 접속 응답 속도를 종합 점검합니다.

GATE 04 · 변화관리/이관

현업 실습 교육 및 오픈

운영자와 현업 담당자가 직접 화면을 조작해 보는 실습 교육 후 안정화 지원에 돌입합니다.

03. 전담 수행 조직 및 역할 분담

Project Organization

역할 구분	주요 담당 업무	고객사 협업 사항
PM / 솔루션 아키텍트	전체 일정·품질 관리, 사내 시스템 연동 아키텍처 및 보안 설계 총괄	도입 목표 범위 확정 및 주간 진척 점검
도메인·AI 엔지니어	SyncSDK™ & Core Modules 코어 구축, 업무 규칙 및 프롭र्ट·모델 최적화, 인터페이스 개발	연동 대상 시스템 API/DB 명세 공유
QA / 변화관리 전담	실환경 시나리오 검증, 운영자·현업 사용자 매뉴얼 작성 및 대면 실습 교육 진행	현업 시범 사용자(UAT) 참여 및 검증

도입 기대 효과(ROI), 주요 활용 사례 및 기술 지원

도입 전후의 실질적인 업무 개선 효과와 안정적인 운영을 뒷받침하는 3단계 기술 지원 체계입니다.

01. 도입 전·후 기대 효과 비교 (ROI)

Expected Benefits

측정 항목	도입 전 (기존 방식)	SyncSDK™ & Core Modules 도입 후	개선 효과
단순 화면·API 수정 리드타임	평균 1~2주 (일정 대기 및 수작업 코딩)	수십 분 내 코드 초안 및 빌드 검증 완료	개발 대기 시간 단축
신규 프로젝트 AI/RAG 기능 탑재	평균 2~3개월 (프레임워크 선정 및 자체 구현)	1~2주 내 (6대 공통 모듈 즉시 결합)	초기 개발 기간 단축
변경에 따른 설계 산출물 현행화	수작업 문서 수정에 건당 반나절 소요	코드 변경과 동시에 산출물 초안 자동 생성	문서 정합성 유지

02. 추천 도입 대상 및 주요 활용 시나리오

Use Cases & References

활용 분야 01

다수 사내 시스템을 유지보수하는 기업 IT 개발실

현업의 반복적인 기능 개선 요청을 빠르게 처리하고 개발자는 핵심 아키텍처에 집중

활용 분야 02

기존 Spring Boot 시스템에 AI 기능을 추가하려는 기업

검증된 AgentScope Java 런타임과 NL2SQL-RAG 공통 모듈을 도입해 시행착오 없이 고도화

03. 3단계 전담 기술 지원 및 유지보수 체계 (SLA)

Maintenance & Support

LEVEL 1 · 헬프데스크

15분 내 초동 응답

온라인·유선 기술 지원 창구를 통해 사용법 문의 및 설정 변경을 빠르게 안내합니다.

LEVEL 2 · 전담 엔지니어

1시간 내 원인 분석·복원

연동 이상 발생 시 로그 분석과 스냅샷 원상 복원을 통해 업무 지연을 최소화합니다.

LEVEL 3 · 코어 연구소

정기 점검 및 기능 패치

분기별 성능·보안 정기 점검 리포트를 제공하고 코어 엔진 업데이트를 지원합니다.

CONTACT & LIVE DEMO

주식회사 엠파시 (Empasy Inc.) | 살아 있는 소프트웨어는 엠파시가 만듭니다

서울특별시 금천구 가산디지털1로 205-27 · 도입 문의: poh@empasy.com / 0507-1360-8169

공식 홈페이지: <https://empasy.io>

통합 데모 포털: <https://sync.empasy.io>

기술 문서 포털: <https://doc.empasy.com>