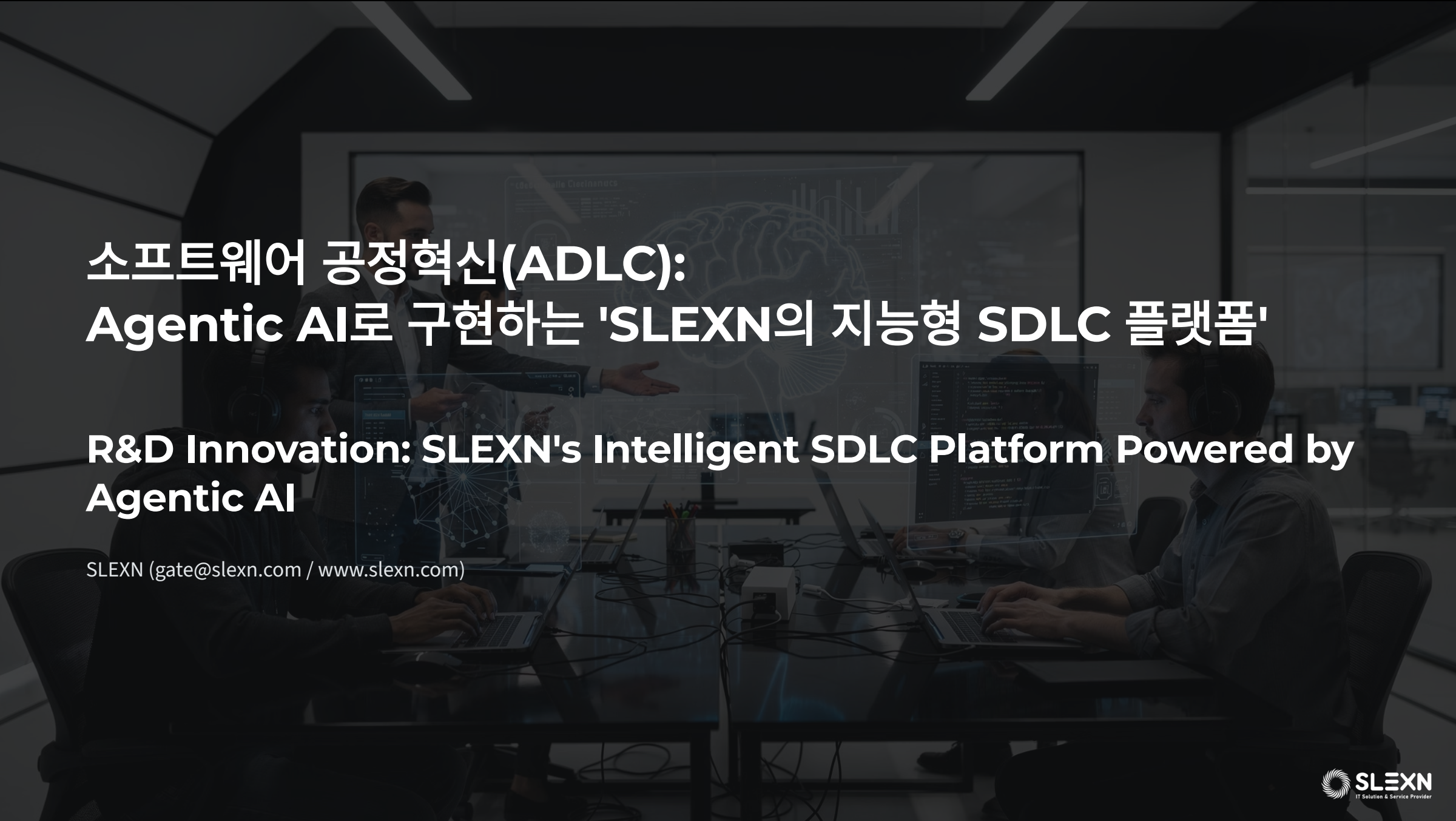


소프트웨어 공정혁신(ADLC): Agentic AI로 구현하는 'SLEXN의 지능형 SDLC 플랫폼'

R&D Innovation: SLEXN's Intelligent SDLC Platform Powered by Agentic AI

SLEXN (gate@slexn.com / www.slexn.com)

SLEXN
IT Solution & Service Provider

개발 공정의 병목 현상: 단절된 도구와 보안의 딜레마

단절된 도구와 데이터 고립 (The Silo Problem)

- **파편화된 워크플로우:** 요구사항(ALM), 코딩(Git), 테스트, 배포 도구가 서로 연결되지 않아 데이터 전송이 수동으로 이루어짐.
- **맥락 상실(Context Loss):** 단계가 넘어갈 때마다 개발 의도나 기술적 배경이 훼손되어 재작업이 발생.
- **툴체인 관리의 복잡성:** 각기 다른 툴을 통합 유지보수하는 비용이 기업의 부담으로 증가.

반복적인 수작업과 생산성 저하

- **비창조적 업무:** 요구사항 문서화, 테스트 케이스 작성, 보안 리포트 생성 등 단순 반복 업무에 개발자의 시간이 과도하게 소요됨.
- **커뮤니케이션 비용:** 이기종 툴 간의 정보 공유 부족으로 인한 불필요한 미팅 및 오해 발생.

보안(Air-gap) 환경에서의 AI 도입 장벽

- **클라우드 의존성:** 최신 AI(Copilot, GPT 등)는 대부분 클라우드 기반이라, 보안이 중요한 국방/금융/제조 분야에서 도입 불가.
- **인프라 부재:** 사내망(Air-gap) 구축을 위한 고성능 GPU 서버와 최적화된 LLM 운영 환경 조식의 어려움.

단순한 보조(Copilot)를 넘어, 자율적 행동(Agent)으로

Generative AI vs. Agentic AI

Generative AI (Chatbot/Copilot)

- 사용자가 질문하거나 명령해야만 반응.
- **역할:** 콘텐츠 생성, 제안 (Suggestion).
- **한계:** 도구를 직접 조작하거나 여러 단계의 작업을 연속으로 수행 불가.

Agentic AI (에이전트)

- **자율성(Autonomy):** 사용자의 목표를 설정하면 스스로 계획을 수립하고 실행.
- **행동(Action):** API를 통해 개발 도구(Git, IDE, Jira 등)를 직접 제어하고 코드를 수정/커밋.
- **협업(Collaboration):** '요구사항 분석 에이전트', '코딩 에이전트', '테스트 에이전트'가 서로 대화하며 작업을 주고받음.

SDLC에서의 에이전트 워크플로우

01

Perceive (인지)

요구사항 변경 사항이나 코드 커밋 로그를 실시간으로 감지.

02

Reason (추론)

변경 사항이 영향을 미치는 범위를 판단하고 수행할 작업(테스트 생성, 코드 리팩토링 등)을 결정.

03

Act (행동)

실제로 툴을 조작하여 작업을 수행하고 결과를 리포트.

보안을 타협하지 않는
완벽한 지능형 SDLC

SLEXN의 핵심 전략: "Secure & Intelligent R&D"

Air-gap Native

모든 AI 기능을 온프레미스 환경에서 구현하여 데이터 유출 리스크 제로(Zero).

End-to-End Orchestration

요구사항부터 배포까지 하나의 플랫폼에서 에이전트가 관리.

LLM Infrastructure

단순 소프트웨어를 넘어, AI 구동을 위한 하드웨어 인프라까지 제공하는 원스톱 솔루션.

SLEXN 지능형 SDLC 플랫폼의 구조

- **뇌(Brain):** Puteron AI - 사내망 구축형 LLM 서버, 안정적인 추론 및 학습 지원.
- **신경망(Connection):** CodeCenter, Trace.Space, GitOn 등 각 툴에 내장된 AI 에이전트들이 Puteron AI를 중심으로 실시간 협업.
- **결과(Result):** 개발 생산성 300% 증대 (예상). 결함 및 보안 취약점 조기 발견. 규제 준수(Compliance) 자동화.

SLEXN
IT Solution & Service Provider

AI 인프라 혁신

Puteron AI (The Brain)

Air-gap 환경을 위한 최적화된 AI 인프라 어플라이언스

Key Message: 비용 효율성과 온프레미스 보안을 동시에 해결하는 AI 서버

제품 개요

정의: 사내망(Air-gap)에 바로 설치하여 LLM(대규모 언어 모델) 및 VLM(비전 모델)을 추론(Inference)하고 학습(Fine-tuning)할 수 있는 올인원(AIO) 서버.

목표: 별도의 복잡한 설정 없이, 전원만 연결하면 바로 기업 전용 AI 환경 구축.

핵심 경쟁력 (Core Technology)

하이브리드 GPU 지원 (NVIDIA + AMD)

특정 벤더에 종속되지 않고 고객의 환경에 맞는 GPU 선택 자유성 보장. 최신 GPU 아키텍처를 지원하여 고성능 연산 수행.

혁신적인 메모리 관리: KV Cache 기술 (Cost Saver)

기존 방식의 문제: 긴 문맥(Context)을 처리하기 위해 비싼 GPU VRAM을 대량으로 구매해야 함.

Puteron AI의 해결: SSD를 활용한 KV Cache 기술 적용.

효과: GPU 메모리 용량 한계를 초월하여 긴 문맥 처리 가능, GPU 추가 구매 비용 획기적 절감.

내장형 LLM Ops

Hugging Face와 호환되는 UI/제공. 모델 다운로드, 학습(Training), 파인튜닝(Fine-tuning), 서빙(Serving)까지 원스톱 관리.

역할: SLEXN의 모든 솔루션(CodeCenter, Trace.Space 등)에 안정적이고 빠른 LLM 추론 능력을 공급하는 개발 생태계의 '심장'.

SLEXN
IT Solution & Service Provider

개발 혁신

CodeCenter

기업 보안을 준수하는 AI 코딩 어시스턴트

Key Message: 'Cursor'의 편리함에 대규모 컨텍스트 엔진과 '기업급 보안'을 더하다

제품 개요

정의: 개발자의 생산성을 극대화하는 AI 기반 코드 생성 및 어시스턴트 도구.

포지셔닝: 클라우드 기반 AI 코딩 툴(Cursor, Copilot 등)의 기능을 Air-gap 보안 환경에서 완벽하게 재현.

핵심 기능 (Features)

강력한 컨텍스트 엔진 & RAG 서버

단순히 공개된 코드만 학습하는 것이 아니라, 사내 문서, 기술 매뉴얼, 기존 소스 코드를 실시간으로 참조(RAG)하여 회사 고유의 코딩 스타일과 비즈니스 로직을 반영한 코드 제안.

MCP (Model Context Protocol) 및 Tool Chain 연동

개발 도구(IDE, Git, CI/CD)뿐만 아니라 외부 도구와 유연하게 연결되는 확장 가능한 아키텍처. Agentic AI가 코드를 작성한 후 직접 Git에 Push하거나 테스트를 실행하는 행동(Action)의 기반이 됨.

협업을 위한 AI Workspace

개발자 개인의 프롬프트나 효율적인 AI 사용법을 팀원들과 공유하여 팀 전체의 AI 활용 역량 강화. 다양한 AI 코딩 도구와의 호환성을 통해 기존 환경 보호.

보안 (Security)

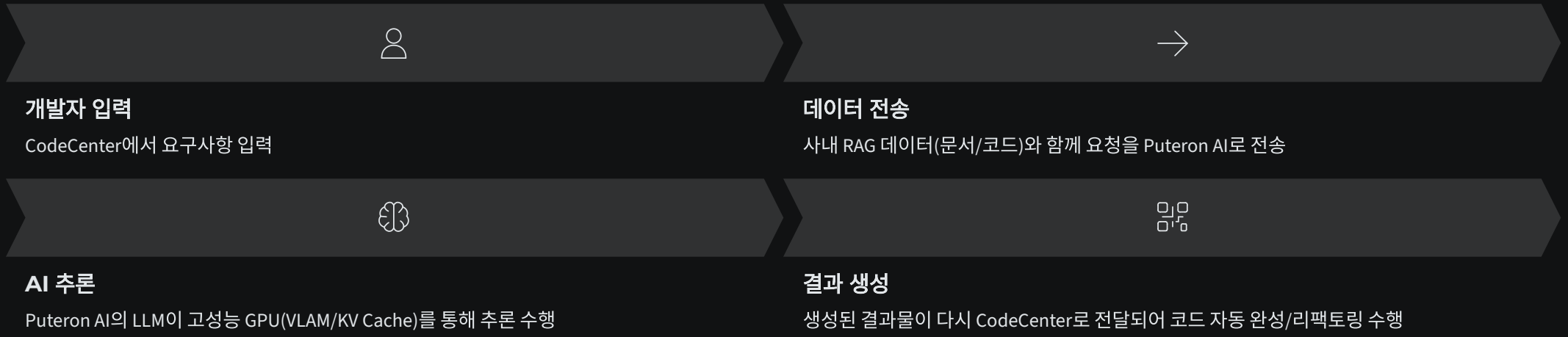
모든 코드 데이터가 사외로 유출되지 않으며, 민감한 정보를 학습 데이터에서 제거하는 필터링 기능 내장.

시너지

Puteron AI + CodeCenter = 완벽한 온프레미스 개발 환경

Key Message: 인프라와 도구의 완벽한 결합으로 실현하는 'Secure Agentic Coding'

통합 아키텍처 (Integrated Workflow)



기대 효과 (Benefits)

성능
로컬 네트워크 기반이므로 클라우드 API 호출보다 응답 속도(Latency)가 훨씬 빠름.

비용
API 사용량(토큰 수)에 따라 과금되는 클라우드 방식과 달리, 자체 장비로 평생 무제한 사용 가능 (Puteron의 효율적인 리소스 활용).

안정성
인터넷 연결 없이도 AI 기능을 24시간 안정적으로 사용 가능.

Trace.Space (AI Requirements & ALM)

완벽한 출발을 위한 AI 기반 요구사항 관리 및 ALM

Key Message: AI가 요구사항을 작성하고 검증하는 '명세 및 설계 품질'의 혁신

제품 개요

정의: 요구사항 정의부터 테스트 관리까지 전체 수명 주기를 관리하는 ALM(Application Lifecycle Management) 도구.

특징: 업계 최초로 AI가 직접 내장된 요구사항 관리 시스템으로, Air-gap 환경에서도 완벽하게 작동.

핵심 기능 및 프로세스 (Core Capabilities)

AI 기반 요구사항 작성 (Authoring)

사용자의 간단한 키워드나 아이디어를 입력하면 AI가 구조적이고 명확한 요구사항 문서를 자동으로 작성. 다국어 번역 및 스타일 정규화 지원.

INCOSE 등 Rule 기반 자동 검증 (Verification)

"요구사항이 모호하지 않은가?" INCOSE(국제 시스템 공학 학회)의 표준 규칙을 적용하여 AI가 작성된 문서를 자동으로 검증. 불완전하거나 모호한 표현을 실시간으로 수정 제안하여 초기 결함 예방.

요구사항 기반 테스트 케이스 자동 생성 (Traceability)

작성된 요구사항을 분석하여, 이를 검증할 수 있는 테스트 케이스(Test Case)와 테스트 스크립트를 자동 생성. 요구사항 변경 시 연계된 테스트 케이스도 자동으로 업데이트 제안.

기대 효과

- 요구사항 작성 시간 단축 및 품질 향상.
- 요구사항과 테스트 간의 무결점 추적성(Traceability) 확보.

SLEXN
IT Solution & Service Provider

Code Intelligence (AI Security Testing)

Agentic Fuzzing으로 조기에 발견하는 보안 취약점

Key Message: 개발 단계에서 잠재는 보안 위협, AI가 미리 찾아낸다

제품 개요

정의: LLM(대규모 언어 모델)을 기반으로 동작하는 차세대 자동화 퍼징(Fuzzing) 테스트 솔루션.

배경: 소프트웨어 결함 중 상당수는 테스트에서 누락되어 배포 후 발견됨. Code Intelligence는 개발 과정 중에 이를 조기에(Early Detection) 발견하도록 돕습니다.

핵심 기술 및 기능

LLM 기반 스마트 퍼징 (Smart Fuzzing)

기존의 랜덤한 무작위 데이터 입력이 아니라, AI가 코드의 로직과 구조를 학습하여 가장 취약할 가능성이 높은 데이터를 생성하여 입력. 사람이 상상하지 못한 엣지 케이스(Edge Case) 시나리오를 자동으로 테스트.

SW 결함 및 보안 취약점 자동 발견

버퍼 오버플로우, 메모리 누수, 크래시 유발 코드 등 심각한 SW 결함을 자동으로 탐지. Zero-day 취약점(알려지지 않은 보안 허점)을 사전에 식별.

개발 파이프라인 통합 (DevSecOps)

CI/CD 파이프라인에 연동하여, 코드가 커밋될 때마다 자동으로 퍼징 테스트를 수행하고 보안 리포트를 생성.

차별점 (Why AI Fuzzing?)

- **적은 리소스:** AI가 효율적인 테스트 경로를 찾아주므로 불필요한 테스트 반복을 줄여 시간과 리소스를 절약.
- **높은 정확도:** 거짓 양성(False Positive)을 줄이고 실제 위협에 집중.

형상관리의 진화

GitOn (AI driven DevOps)

기업 환경에 최적화된 AI 기반 Git Code 관리 시스템(8x faster vs. GitLab, GitHub)

Key Message: Air-gap 환경에서도 GitHub/GitLab 수준의 경험을, 그 이상의 성능과 보안으로 제공

제품 개요

정의: Git 기반의 소스 코드 관리(SCM) 협업 플랫폼.

포지셔닝: GitLab/GitHub의 강력한 기능을 담되, 금융/국방/제조 등 보안이 중요한 온프레미스 환경에 특화된 대안 솔루션.

핵심 특징 (Key Features)

1

압도적인 고속 처리 (High Performance)

대규모 레포지토리와 거대한 파일(LFS) 처리에 최적화된 엔진. 클론/풀/푸시 속도가 타 경쟁사 대비 훨씬 빨라, 대규모 개발 조직의 빌드 시간 단축에 기여.

2

기업용 보안 및 고가용성 (HA)

Air-gap 폐쇄망 완벽 지원. 이중화(HA) 구성을 통해 서버 장애 시에도 중단 없는 개발 환경 보장. 세밀한 권한 제어 및 감사 로그(Audit Log) 기능.

3

AI 기반 협업 강화

AI가 코드 리뷰를 보조하거나, 병합(Merge) 충돌을 예측하여 개발자의 효율성 향상.

기대 효과

- 외부 클라우드 의존도 제거로 데이터 주권 확보.
- 빠른 처리 속도로 개자들의 생산성 저하 요인 제거.

SLEXN
IT Solution & Service Provider

보안 중심의 패키지 및 바이너리 관리

OllPack (Secure Repository)

개발부터 배포까지 안전한 소프트웨어 공급망 관리

Key Message: OSS 취약점을 사전에 차단하고, 아티팩트를 안전하게 관리하는 'Curation'의 중심

제품 개요

정의: 바이너리, 패키지, 도커 이미지 등 모든 소프트웨어 아티팩트를 저장하고 관리하는 통합 리포지토리.

핵심 가치: JFrog Artifactory의 강력한 기능을 온프레미스 환경에 맞게 재구성한 'Software Supply Chain' 보안 솔루션.

핵심 기능 (Core Capabilities)

OSS Curation (선별 및 관리)

오픈소스 라이브러리를 단순히 저장하는 것을 넘어, 사용 전에 자동으로 검사하여 보안 취약점(CVE)과 라이선스 위반 여부를 판단. 안전한 라이브러리만을 승인(Approval)하여 개발자들이 사용하도록 유도.

세밀한 권한 관리

프로젝트, 팀, 개발자별로 접근 권한을 매우 세분화하여 설정 가능. 배포할 수 있는 사람(Promotion 권한)을 엄격히 통제하여 운영 사고 예방.

Air-gap 통합 관리

인터넷이 없는 환경에서도 내부 패키지와 외부에서 가져온 라이브러리를 한 곳에서 통합 관리.

보안 강점

SBOM (Software Bill of Materials): 어떤 오픈소스가 포함되었는지 명확한 목록화를 지원하여 규제 대응 용이.

aras INNOVATOR

ARAS (PLM) - 소프트웨어와 하드웨어의 완벽한 조화: ALM + PLM

Key Message: Trace.Space와 연동하여 시스템 전체의 생명주기를 통합 관리

제품 개요

정의: 제품(Product)의 전체 수명 주기를 관리하는 PLM(Product Lifecycle Management) 솔루션.

특징: PTC, Siemens, Dassault 와 달리 온프레미스 AI 연동 및 Air-gap 설치가 가능한 엔터프라이즈급 솔루션.

기대 효과

- 복잡한 시스템(예: 자율주행차, 방산 시스템) 개발 시 SW/HW 간의 연동 오류 감소.
- 전체 제품 개발 일정 최적화.

ALM과의 통합 (The Synergy)

SW + HW 통합 관리

기존 PLM은 하드웨어(기계, 부품) 위주였으나, 이제는 소프트웨어가 제품의 핵심이 되었습니다. Trace Space(SW 요구사항) 및 CodeBeamer 등 ALM 도구와의 강력한 연동 커넥터를 제공. 소프트웨어 버전과 하드웨어 버전을 매핑하여, 특정 하드웨어에 어떤 소프트웨어가 탑재되었는지 명확히 추적.

디지털 스레드 (Digital Thread)

요구사항 -> 설계 -> 제조 -> 서비스까지 데이터가 끊기지 않고 이어지는 디지털 스레드 구현.

SLEXN의 지능형 SDLC 스택(ADLC Platform)

One Brain, One Platform: 중앙 집중형 지능형 개발 아키텍처

Key Message: Puteron AI가 모든 도구의 두뇌가 되어, 통합된 지능을 공급하는 단일 플랫폼

아키텍처 구성도 (Architecture Layers)

L1. 인텔리전스 & 데이터 기반 계층 (The Brain & Foundation)

Puteron AI (Central Brain): 모든 솔루션에 LLM 추론 능력과 학습 기능을 공급하는 중앙 제어 타워.

GitOn & OllPack (Data Layer): 소스 코드와 바이너리가 안전하게 저장되는 신뢰할 수 있는 데이터 창고.

L2. 개발 및 협업 계층 (The Smart Process)

Trace.Space: 요구사항과 테스트 계획을 AI로 생성.

CodeCenter: AI 코딩 어시스턴트와 개발 워크스페이스.

Code Intelligence: 코드의 결함과 취약점을 AI로 자동 탐지.

MCP(Open Protocol)를 통해 각 도구의 AI Agent가 서로 통신

L3. 시스템 관리 계층 (The System View)

ARAS: 소프트웨어 변경 사항을 하드웨어(PLM) 정보와 연계하여 전체 시스템 충돌을 관리.

설계 철학 (Design Philosophy)

- **Security First:** 모든 계층이 Air-gap 내에서 구동되며, 데이터는 외부로 나가지 않음.
- **Open & Connected:** 폐쇄적인 개별 툴이 아니라, 표준 프로토콜을 통해 데이터가 자유롭게 흐르는 'Living Platform'.

SLEXN
IT Solution & Service Provider

Agentic AI에 의한 자동화 파이프라인

끊김 없는 흐름: 요구사항부터 배포까지 (Requirements to Release)

Key Message: AI 에이전트들이 도구 사이의 경계를 허물고 업무를 자동으로 연결한다

순환적 워크플로우 (The Agentic Loop)

① 계획 (Plan) - Trace.Space
사용자가 목표 입력 → Trace Space Agent가 요구사항 자동 작성 및 INCOSE 검증.

② 개발 (Code) - CodeCenter & Puteron AI
Puteron AI가 요구사항 분석 → CodeCenter Agent가 사내 RAG 기반으로 코드 생성 및 커밋 제안.

③ 저장 (Version) - GitOn
코드가 GitOn에 Push → 트리거(Trigger) 발생.

④ 검증 (Test) - Code Intelligence
Test Agent가 커밋을 감지 → 자동으로 Fuzzing 테스트 실행
→ 취약점 리포트 생성.

⑤ 배포 (Release) - OllPack & CONTACT
테스트 통과 시 → OllPack이 바이너리 패키징 및 OSS Curation 검사.
최종 버전 정보가 PLM 데이터베이스와 동기화

핵심 기술: 에이전트 간 협업 (Agent Collaboration)

MCP (Model Context Protocol)
도구 간의 호환성 문제를 해결하여, AI 에이전트가 A 툴의 정보를 읽어 B 툴에서 행동(Action)할 수 있게 하는 표준 인터페이스.

Real-time Sync
한 곳에서 발생한 변경 사항이 연관된 다른 도구에 즉시 반영됨 (예: 요구사항 변경 시 연관 테스트 케이스 자동 수정 제안).

기대 효과 (Impact)

- **Hand-off 제거:** 단계가 넘어갈 때 발생하는 정보 누락이나 수작업 복사/붙여넣기 제거.
- **Self-healing Pipeline:** 오류가 발생하면 AI 에이전트가 스스로 원인을 분석하고 수정안을 제안하는 자가 치유 파이프라인 구현.

Use Case: Agentic AI를 활용한 SDLC 자동화 시나리오

예. 안전성이 중요한 자동차 제어 소프트웨어 개발

수동 핸드오프(Hand-off)를 제거하고, AI 에이전트가 API 기반으로 상호 작용하는 'Self-Driving DevOps' 구현

시나리오 배경: 차량의 브레이크 제어 로직에 새로운 '비상 제동 지원(Emergency Brake Assist)' 기능을 추가하는 경우

01

요구사항 정의 및 검증 (Requirements & Planning)

Tools: Trace.Space ↔️ Puteron AI

Input: 엔지니어 입력 (예: "센서 입력 10ms 이내 비상 제동 로직 추가")

Agentic Action (Agent A):

- **자동 구체화:** Puteron LLM이 모호한 입력을 ISO 26262 표준 요구사항 포맷으로 변환 (ID: REQ-2024-001).
- **규칙 기반 검증:** INCOSE Rule 엔진이 모순(Conflict)과 모호성(Ambiguity)을 실시간 분석.
- **연계 작업 생성:** 요구사항 승인 즉시, 연동된 GitOn API를 통해 해당 기능을 위한 Feature 브랜치 자동 생성.
- **테스트 동기화:** 요구사항 ID를 기반으로 Trace.Space 내부에 Test Case(단위/통합) 자동 생성 및 테스트 큐잉.

02

지능형 코딩 및 구현 (Intelligent Coding)

Tools: CodeCenter (IDE Plugin) + Puteron AI

Input: GitOn에서 생성된 이슈 및 Trace.Space의 REQ-2024-001

Agentic Action (Agent B):

- **Context Awareness:** CodeCenter의 RAG 엔진이 사내 '코딩 표준 가이드(MISRA C 준수)' 및 '유사 레거시 코드'를 벡터 검색하여 참조.
- **코드 생성:** 단순 라인 생성을 넘어, 함수 레벨의 구조(Skeleton)와 로직을 제안.
- **자동 문서화:** 생성된 코드 블록에 대해 주석을 자동 삽입하여 가독성 확보.
- **Commit Suggestion:** 개발자가 코드를 수정하면, Commit Message를 자동으로 생성하고 GitOn Repository로 Push 제안.

03

지속적 보안 검증 및 아티팩트 관리 (Security & Compliance)

Tools: GitOn → Code Intelligence & OllPack

Trigger: GitOn Pre-commit Hook 또는 Pull Request 이벤트 발생

Agentic Action (Agent C - Security):

- **LLM 기반 퍼징 (Code Intelligence):** 새로운 코드의 함수 시그니처를 분석하여 Coverage-guided Fuzzing 테스트 케이스를 자동 생성. 버퍼 오버플로우 등 잠재적 크래시(Crash) 유발 코드를 탐지하고, GitHub/GitOn PR에 코멘트로 리포팅.
- **공급망 보안 (OllPack):** 빌드 시점에 새롭게 추가된 라이브러리의 SBOM(Software Bill of Materials)을 실시간 생성. 사전 구축된 Curation DB와 대조하여, 취약점(CVE)이 있는 OSS가 포함된 경우 빌드 파이프라인을 자동 차단(Block).

04

ALM+PLM 통합 배포 및 관리 (Release & Management)

Tools: ARAS ↔️ Trace.Space

Input: 성공한 빌드 아티팩트 (OllPack Storage) 및 테스트 리포트

Agentic Action (Agent D - Management):

- **버전 매핑:** 소프트웨어 빌드 버전(v1.2.3)을 CONTACT Software 내의 해당 ECU(전자제어유닛) 하드웨어 파트(Part No.)와 연결.
- **충돌 검증:** 현재 하드웨어 스펙(메모리, IO)과 소프트웨어 요구사항의 호환성을 AI가 교차 검증(Validation).
- **배포 승인:** 모든 검증이 통과되면 ARAS PLM에서 Change Request(CR)를 자동 승인 상태로 변경하고, 배포 패키징을 트리거.

기대 효과 및 성과 (Metrics & ROI)

DevOps 성과 지표 혁신: 생산성, 품질, 보안의 밸런스

정성적인 편의성을 넘어 정량적인 지표(Metric)로 입증되는 SDLC 혁신

생산성 (Productivity) - "Time-to-Market 단축"

개발 사이클 타임(Lead Time) 최소화

- 요구사항 작성부터 코드 생성까지의 수작업 시간 70% 이상 감소 (Trace.Space + CodeCenter).
- 에이전트 간 자동 협업으로 이메일/미팅 등 커뮤니케이션 오버헤드(Overhead) 제거.

인프라 운영 비용(OPEX) 절감

- Puteron AI의 KV Cache 기술: 동일한 Context Window 처리 시 필요한 GPU VRAM 사용량 효율화 → 추가 GPU 도입 비용 40% 절감.
- API 토큰 과금(Cloud LLM) 제거로 고정 비용화.

품질 (Quality) - "Shift-Left를 통한 결함 제로화"

조기 결함 탐지 (Early Defect Detection)

- 요구사항 단계(Trace.Space)에서 40~60%의 논리적 오류를 사전 예방.
- 코딩 단계(Code Intelligence)에서 퍼징 테스트를 통해 런타임 에러를 커밋 전 해결.

테스트 커버리지 향상

- 인간이 작성하기 힘든 엣지 케이스(Edge Case)를 LLM이 생성하여 코드 커버리지(Code Coverage) 20% 이상 향상.
- 추적성(Traceability) 자동화로 규제 심사(Audit) 시간 단축.

보안 (Security) - "Zero Trust & Compliance 준수"

완벽한 Air-gap 보안

- 모든 AI 추론이 온프레미스(Puteron AI) 내에서 완료 → 소스코드 및 학습 데이터의 사외 유출 위험 제로(Zero).
- 세밀한 RBAC(Role-Based Access Control)로 권한 없는 사용자의 AI 접근 제어.

소프트웨어 공급망 보안 (SCS)

- OllPack Curation: 취약한 오픈소스 라이선스 사전 필터링 → 법적 리스크(Legal Risk) 및 보안 패치 미적용 리스크 해소.
- SBOM 자동화를 통해 보안 가이드라인 및 사이버 보안 인증 대응.

완벽한 Agentic AI SDLC 생태계

Why SLEXN? 보안과 생산성의 두 마리 토끼를 잡는 유일한 솔루션

온프레미스 환경에서 구현 가능한 유일한 End-to-End Agentic AI 플랫폼

전방위적인 기술 포트폴리오 (Full-Stack Capabilities)

Infrastructure (Puteron AI)
클라우드 없이 구동되는 고성능 LLM 인프라로 AI 도입의 진입 장벽 해소.

Development (CodeCenter, GitOn)
엔터프라이즈 보안을 준수하는 초고속 협업 및 코딩 환경.

Prduct LM (Trace.Space, ARAS)
요구사항부터 HW/SW 통합 관리까지 디지털 스레드 구현.

Quality (Code Intelligence, OllPack)
개발 단계에서의 보안 검증과 안전한 소프트웨어 공급망 관리.

단편적인 도구(Point Solution)가 아닌, **통합된 플랫폼(Platform)**을 통해 데이터와 지능형 R&D 구축

SLEXN만의 차별화 된 경쟁력 (Competitive Advantage)

Air-gap Native
타 솔루션(클라우드 기반)과 달리, 국방/금융/제조 등 보안이 필수적인 환경에서 100% 온프레미스로 구현 가능. 데이터 주권(Data Sovereignty) 완전 보장.

Agent Ecosystem
각 제품이 단순히 AI를 탑재한 것이 아니라, 서로 대화하고 협업하는 Agent로 설계됨. MCP(Model Context Protocol) 기반의 확장 가능한 아키텍처 제공 및 커스텀 개발 서비스.

비즈니스 임팩트

비용(Cloud API 비용 제거) + 보안(유출 방지) + 생산성(AI 자동화)의 트리플 플레이(Triple Play) 달성.

SLEXN
IT Solution & Service Provider

지능형 R&D의 미래

Next Horizon:
인간의 개입이 최소화되는 자율지능 개발 환경

AI 단순 보조를 넘어, 스스로 설계하고 검증하는 자율 R&D 시스템으로 진화

Hyper-Automation (초고도 자동화)

Zero-Touch Deployment: 현재는 개발자의 승인이 필요한 배포 과정을, AI가 모니터링하는 '자율 운영 모드'로 전환. 시스템 장애 시 스스로 롤백(Rollback)하고 패치를 생성하는 Self-Healing Pipeline 구현.

Dynamic Orchestration: 프로젝트의 복잡도나 긴급도에 따라 AI가 자동으로 리소스(Puteron의 GPU 할당, 테스트 병렬 처리 수준)를 최적화

Hardware/Software Co-Design (HW/SW 통합 설계)

AI-Driven Co-Simulation: CONTACT PLM과 CodeCenter의 연계를 심화하여, 소프트웨어 변경이 하드웨어 성능(전력, 열, 메모리)에 미치는 영향을 AI가 미리 시뮬레이션(Co-Simulation). "이 코드는 하드웨어 스펙을 초과한다"는 경고를 설계 단계에서 제공.

Digital Twin Integration: 개발된 소프트웨어를 하드웨어 디지털 트윈에 실시간 탑재해 가상으로 검증하여, 실제 프로토타입 제작 비용 절감.

Advanced LLM Ops (고도화된 도메인 모델)

Domain-Specific Small Language Models (SLMs): 거대한 범용 모델(Llama, GPT 등) 대신, SLEXN이 제공하는 특정 산업(자동차, 방산, 금융)에 특화된 소형 모델(SLM)을 Puteron AI에 탑재. 효과: 더 적은 리소스로 더 정확한 전문 용어 처리 및 코딩 지원 가능.

On-Device RLHF (Reinforcement Learning from Human Feedback): 개발자가 코드를 수정하는 피드백 루프를 실시간으로 학습하여, 기업 맞춤형 코딩 스타일로 모델이 지속적으로 진화(Evolution).

SLEXN
IT Solution & Service Provider