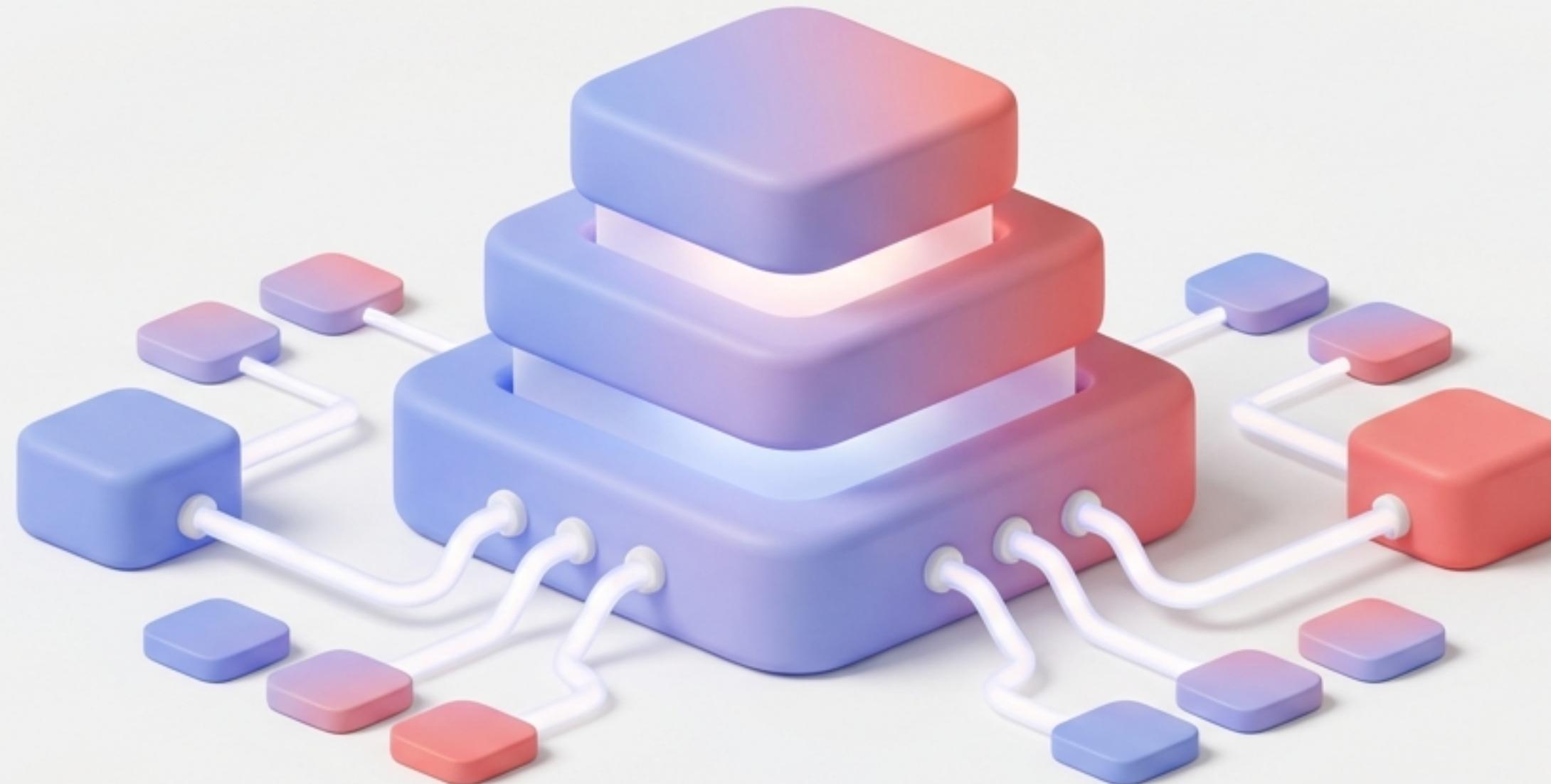


ACTIVO: 차세대 로컬 기반 코드 품질 분석 에이전트

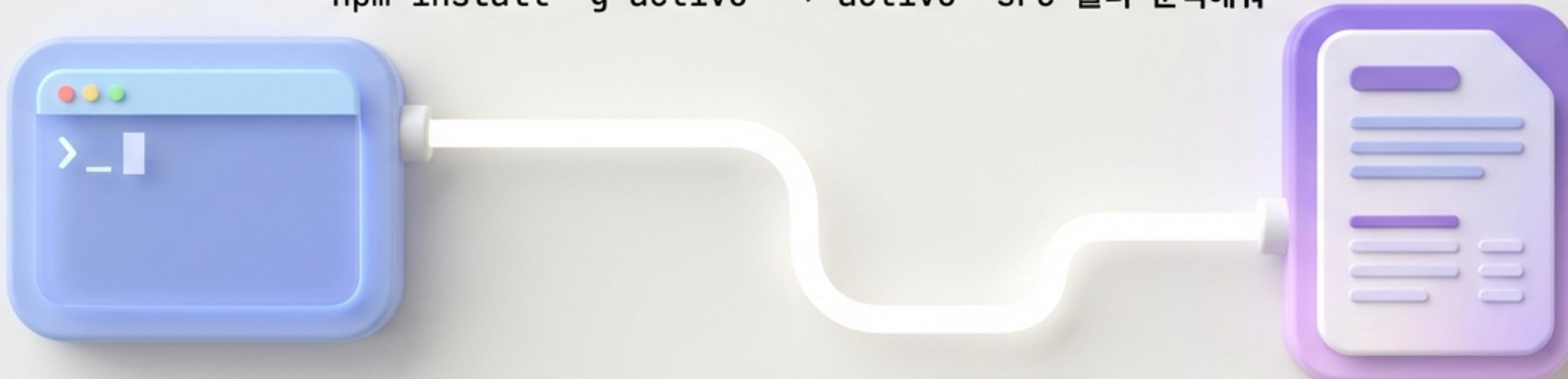
보안 걱정 없는 100% 로컬 LLM 환경에서, 단 하나의 명령어로 멀티스택 프로젝트를 진단합니다.



Ollama + Tool Calling + MCP 기술 기반

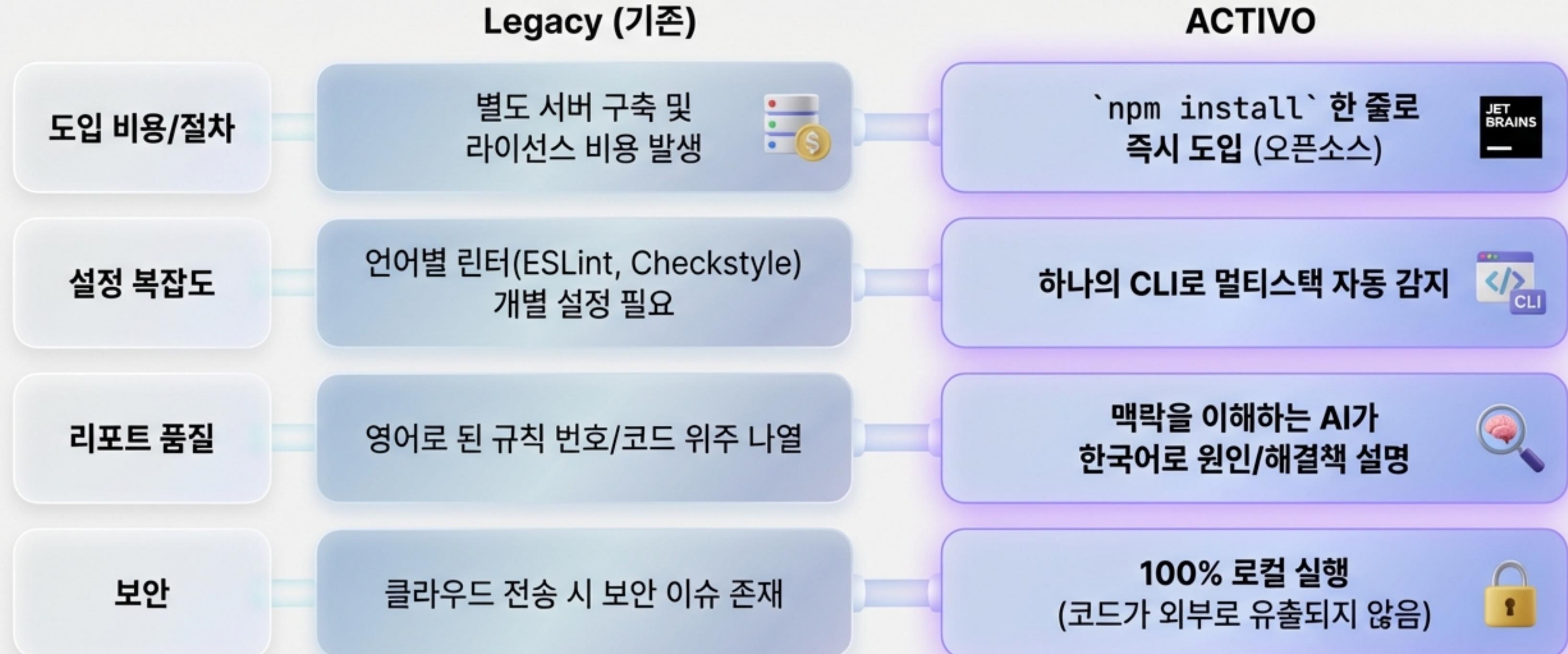
설치부터 분석까지, 복잡한 설정 없이 즉시 시작

`npm install -g activo` → activo 'src 폴더 분석해줘'`



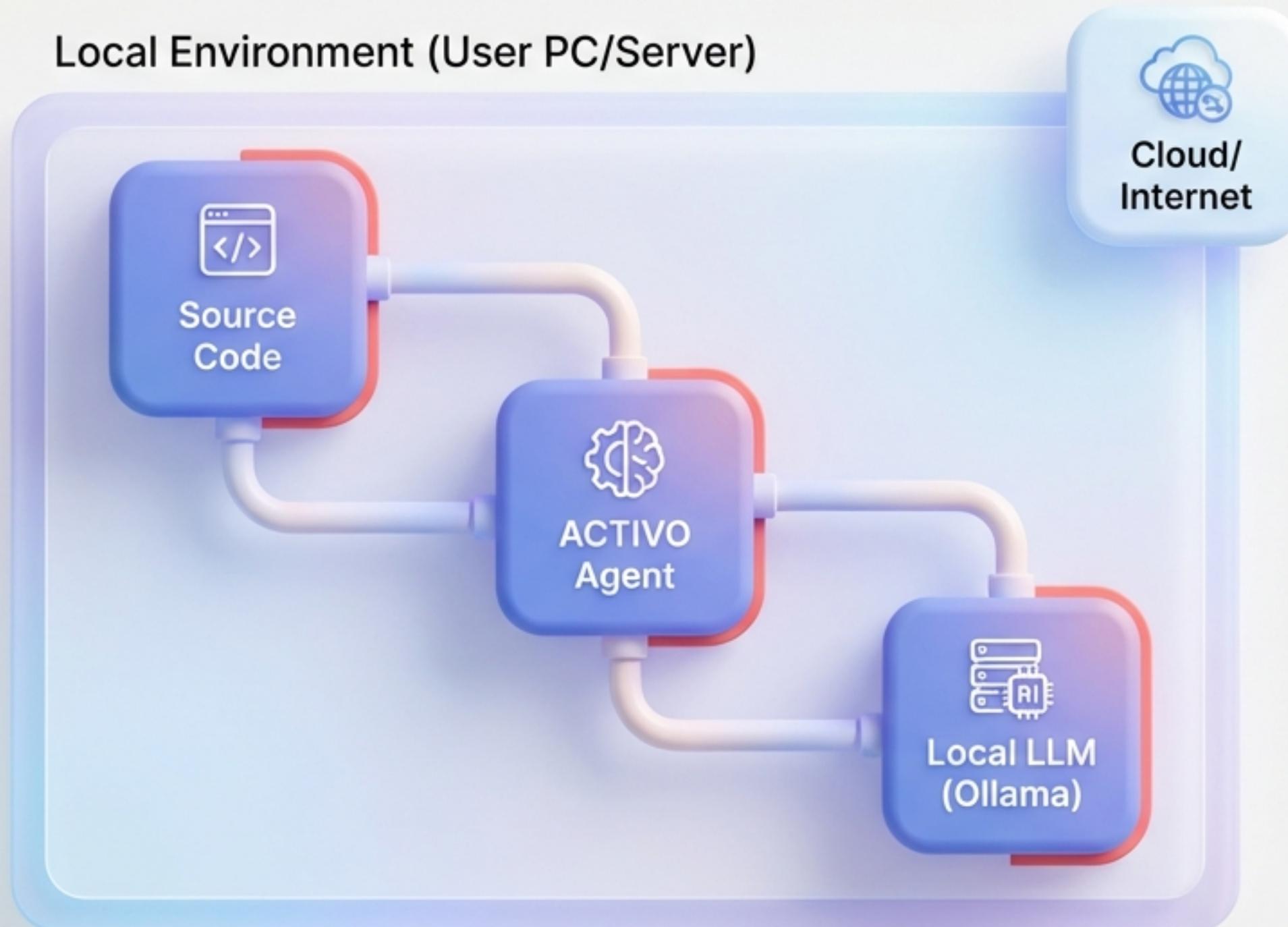
단 한 줄의 명령어로 Java, Python, TS, SQL 등 다양한 언어를 자동 감지하여 분석합니다.
단순한 오류 나열이 아닌, '어디가 문제이고 어떻게 고쳐야 하는지' 한국어로 친절하게 설명합니다.

기존 정적 분석 도구 vs ACTIVO



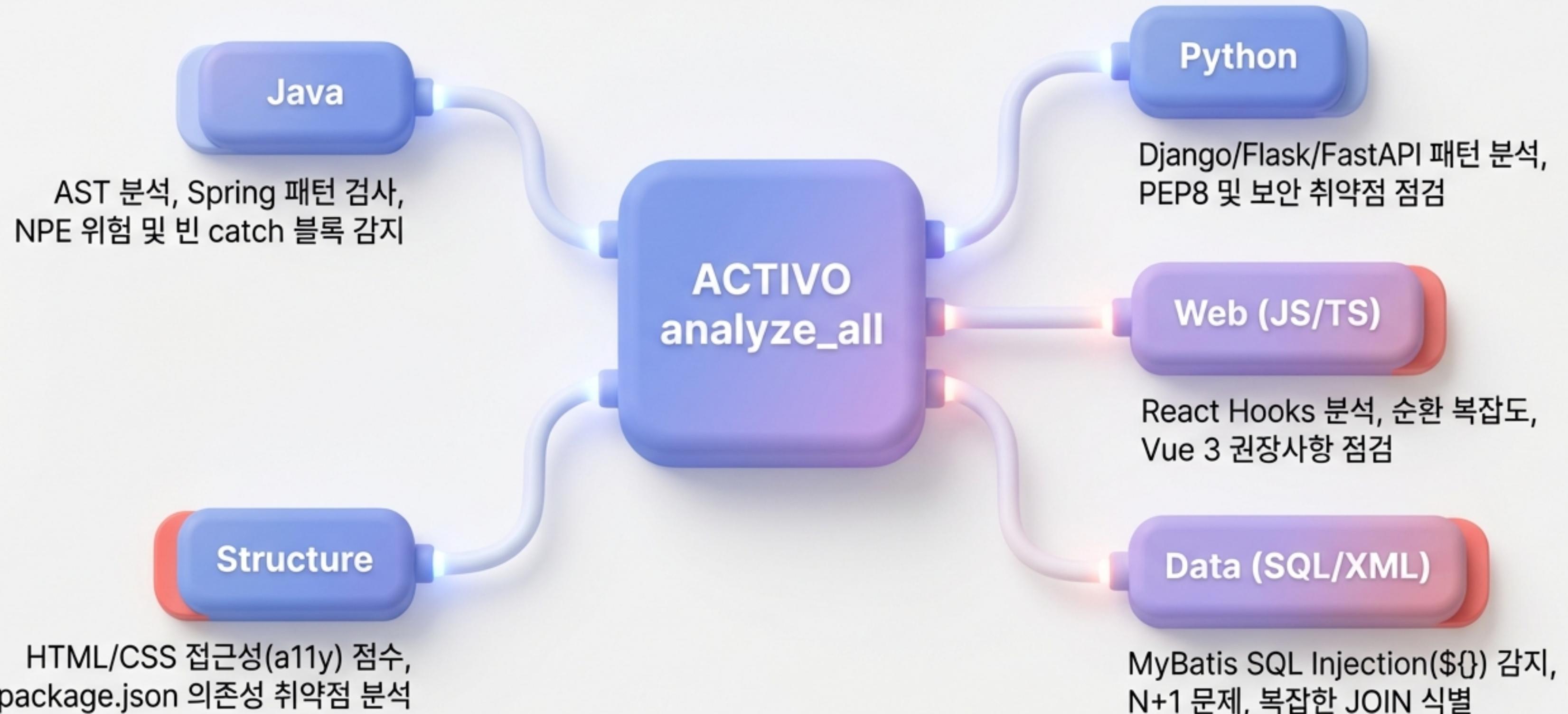
데이터 유출 제로, 완벽한 100% 로컬 아키텍처

Local Environment (User PC/Server)



- **Secure Infrastructure:** 코드가 외부 API나 클라우드로 전송되지 않습니다. 폐쇄망 환경에서도 안전하게 작동합니다.
- **Local Intelligence:** Ollama와 로컬 LLM(Mistral, Qwen 등)을 활용하여 내부망에서 모든 추론과 분석이 완료됩니다.
- **Architecture Principle:** 분석은 전용 도구가 정확하게 수행하고, LLM은 요약과 설명만 담당하여 환각 현상을 최소화합니다.

`analyze_all`: 프로젝트 내 모든 기술 스택 자동 감지



Static Analysis

```
2023-05-28 12:54:28.7388752+009 | org.uneuEochnucapdata.coenr+s+oe,sslog_cvsto...  
2022-06-28 12:54:28.7388752+009 | org.nodoteneccorSamboseen.ercooyasexs,contoorog...  
2023-05-28 12:54:28.7388752+008 | org.oosaaeoortdkoocnssee,arcors,contiesitonnes...  
2023-05-28 12:54:28.7388752+008 | org.oeeo2Sensoo4adGettoe,errrera.ceee.Uoo,obne...  
2022-06-28 12:54:28.7388752+008 | org.necesnotonitmentore,errenra.confiectionPte...  
2023-05-28 12:54:28.7388752+008 | org.bosacecostAloedehaca,ertrorescoes,stetcooper...  
2022-06-28 12:54:28.7388752+009 | org.coro2sensentaddeexconarxormesexa,atetCohnet...  
2022-06-28 12:54:28.7388752+009 | org.nondecomoeokeberctee,errrera.contioorieneara...  
2023-05-28 12:54:28.7388752+009 | org.nosadcoos.osendss,cooorrrestors.appizon35...  
2022-06-28 12:54:28.7388752+009 | org.coccesenerokodenses,arroressexva,stot2onde...  
2023-05-28 12:54:28.7388752+008 | org.herdampnortoetvacecessors.nuttibetekanak...  
[8000R] NutlPo2m1s+Eseoption a5 llee (3  
[E000A] NutlPainterE3captten.ton dt llee 42  
[8000] Secority warning: RRsing validation  
[8000] Secority warning: RRsing valzdetien  
[E000A] Code seeli: Righ cycloatts coaptexity  
[E000A] Code seeli: Righ cycloatic cooptexity  
[ERR0R] Secority warning: RRsing validation (s betozen NuttFoskierException 8 ...  
  
static class Code {  
    public static tomcostatopen() {  
        cesotsode = oct1;  
        return ariopspasicot&etoctingucknis;  
    }  
    public static foocaeet&aeet() {  
        cbeta.serateade + ecoosoesS&niroe();  
        Ss7log eseces + os.oyonkeearIndee.setEctentProkes();  
        S2sesem.oetodatss.getE088821();  
        setcm correevete.&bstoe(oo());  
    }  
}  
  
Event  
| union Syslog  
| where EventLevelName == "Error"
```

Conversational Insight

단순 규칙 검사를 넘어, 코드의 '맥락'을 이해하는 리포트



- **Contextual Priority:** 정적 규칙만 적용하는 것이 아니라, AI가 전체 맥락을 파악하여 수정 우선순위를 판단합니다.



- **Korean Explanation:** '왜 이 코드가 문제가 되는지', '어떻게 수정하면 더 효율적인지'를 개발자에게 한국어로 친절하게 설명합니다.



- **Learning Tool:** 신입 개발자에게는 코드 리뷰어이자 학습 도구로서의 역할을 수행합니다.

RAG 기반 조직 전용 개발 표준 자동 점검



- **Document Integration:** 조직 내부의 개발 표준 문서(HWP, PDF)를 AI가 직접 읽고 학습합니다.
- **Auto-Conversion:** HWP/PDF 문서를 마크다운으로 자동 변환하고 벡터 임베딩(nomic-embed-text)을 수행합니다.
- **Compliance Check:** 개발자가 작성한 코드가 우리 조직의 표준 규칙을 준수하는지 실시간으로 대조하고 피드백을 제공합니다.

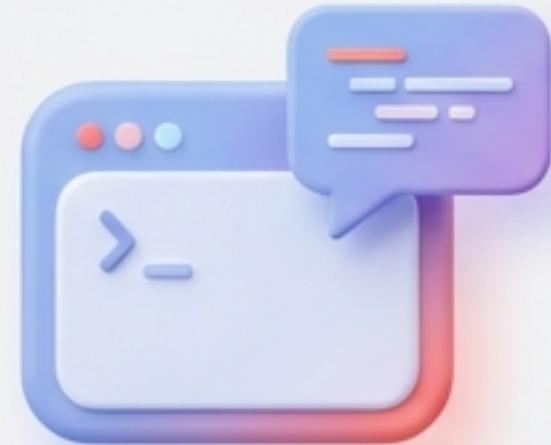
시맨틱 검색을 통한 레거시 코드 및 의존성 분석



- **Semantic Search:** 단순히 키워드(변수명)로 찾는 것이 아니라, '로그인 처리 로직 어디에 있어?'와 같이 자연어 의미 기반으로 코드를 검색합니다.
- **Structure:** 함수/클래스 단위로 코드를 청킹(Chunking)하여 벡터 인덱싱을 수행, 정확도를 높였습니다.
- **Session Memory:** 프로젝트의 기술 스택과 주요 파일 맥락을 `.activo/memory/`에 저장하여, 반복 설명 없이 연속적인 분석이 가능합니다.

개발자 로컬부터 CI/CD 파이프라인까지 유연한 대응

Interactive (TUI)



터미널 대화형 모드.
채팅하듯 코드를 분석하고
질문합니다.

```
'activo'
```

One-Shot



프롬프트 명령 한 번으로
특정 폴더나 기능을 즉시 분석합니다.

```
'activo 'src 분석해줘'
```

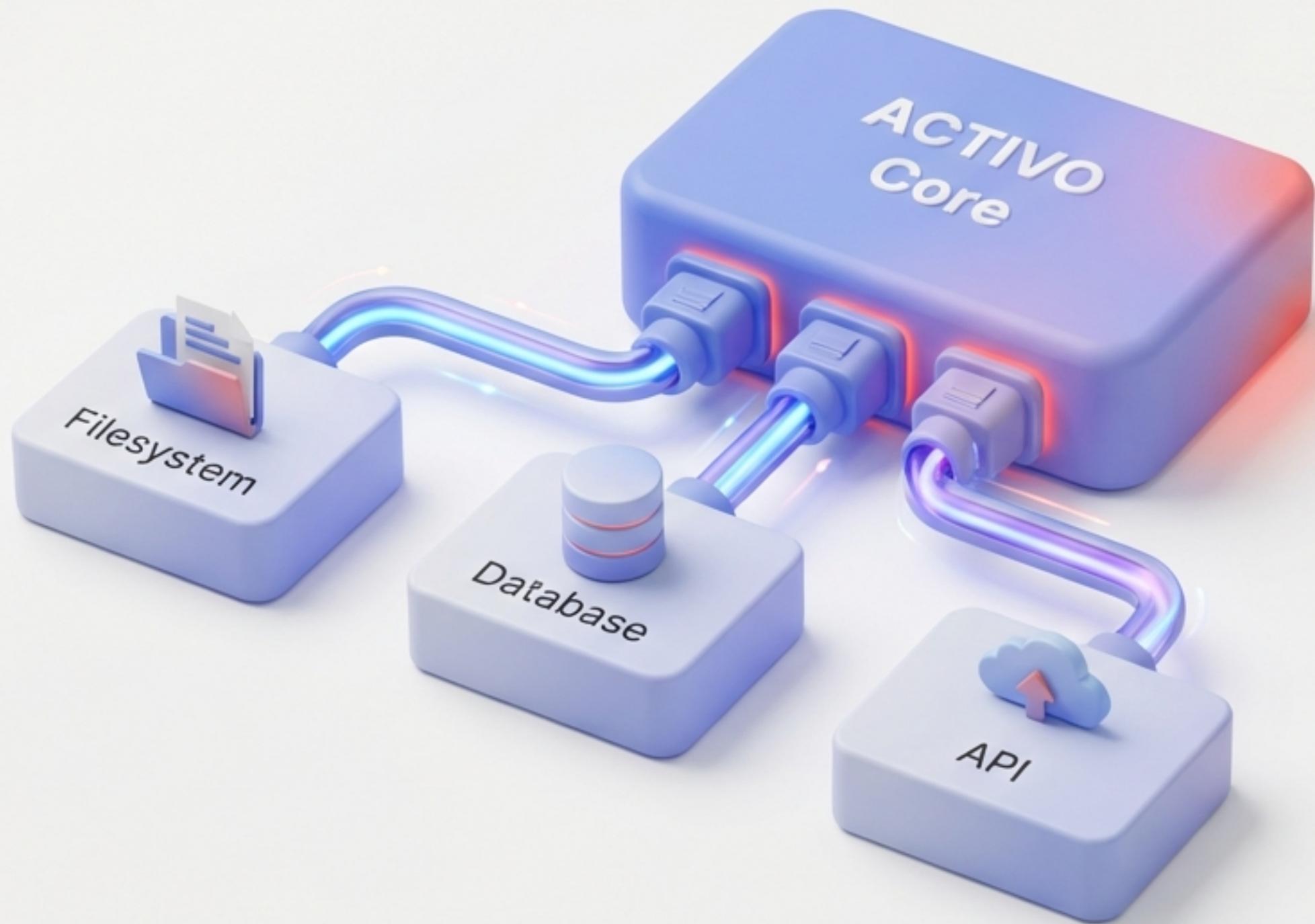
Automation (Headless)



CI/CD 파이프라인 연동.
Github Actions/Jenkins 등에서
품질 게이트로 활용합니다.

```
'activo --headless 'analyze_all'
```

확장성을 위한 MCP (Model Context Protocol) 지원

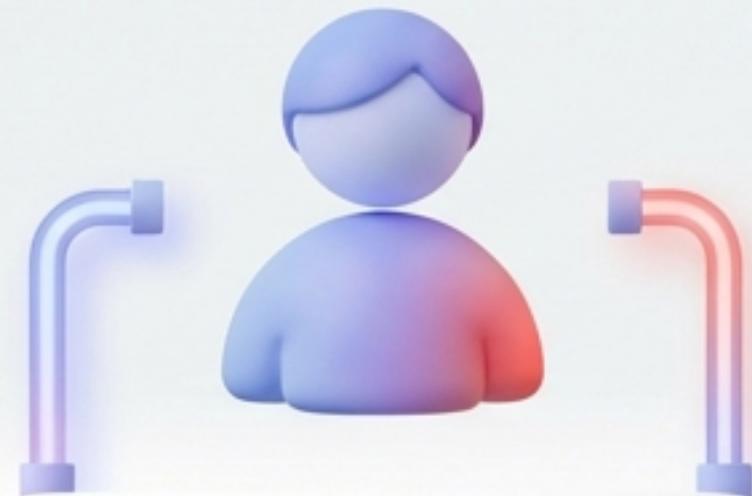


- **Standard Protocol:** MCP 서버를 등록하면 ACTIVO가 파일시스템, 데이터베이스, API 등 외부 도구를 자동으로 호출하여 사용할 수 있습니다.
- **Efficient Routing:** 'Intent Router'가 사용자의 의도를 파악하여 LLM 호출 없이 적절한 도구를 선택, 속도는 높이를 사용도를 눈려' , 속도는 높이고 VRAM 소모는 줄였습니다.

가볍고 효율적인 시스템 요구사항

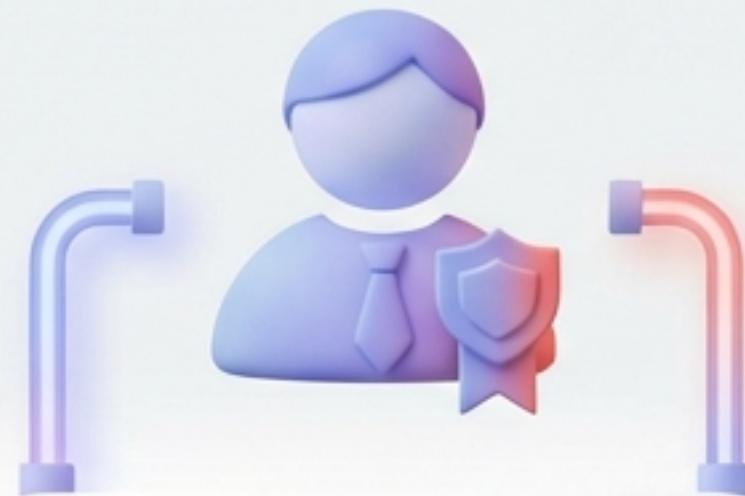
- OS/Runtime: Node.js 18 이상 (TypeScript 기반)
- LLM Engine: Ollama (최신 버전 권장)
- Recommended Model: Mistral (8GB VRAM) 또는 Qwen2.5-Coder (4GB VRAM)
- Hardware: 일반적인 개발자 랩탑 수준 (8GB ~ 16GB RAM)에서 원활하게 구동
- License: MIT 라이선스 (상업적 이용 가능, 무료)

도입 기대 효과



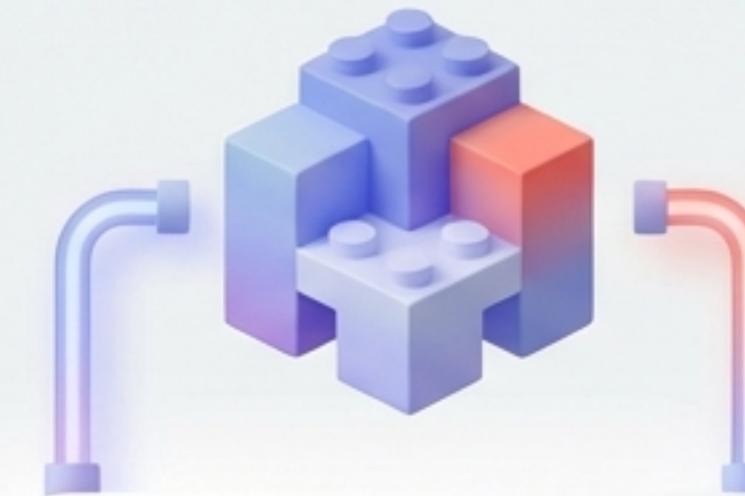
Developer (개발자)

- 커밋 전 headless 모드로 셀프 코드 리뷰 수행
- 복잡한 레거시 코드의 로직을 대화형으로 빠르게 파악



Team Leader (리더/QA)

- 조직 내부 개발 표준 문서 (PDF/HWP) 기반 자동 감수
- Java, SQL, Web 등 파편화된 스택을 통합 리포트로 관리



Organization (조직)

- 외부 데이터 전송 없는 강력한 보안 환경 구축
- 오픈소스 및 로컬 LLM 활용으로 라이선스 비용 절감

ACTIVO + CQC: 상호보완적 품질 관리 시스템



ACTIVO로 문제를 탐색/학습하고, CQC로 최종 품질 게이트를 통과시키는 하이브리드 전략.



가장 안전하고 똑똑한 코드 품질 파트너, ACTIVO

지금 바로 시작하세요.

```
npm install -g activo
```

<https://github.com/mhb8436/activo>