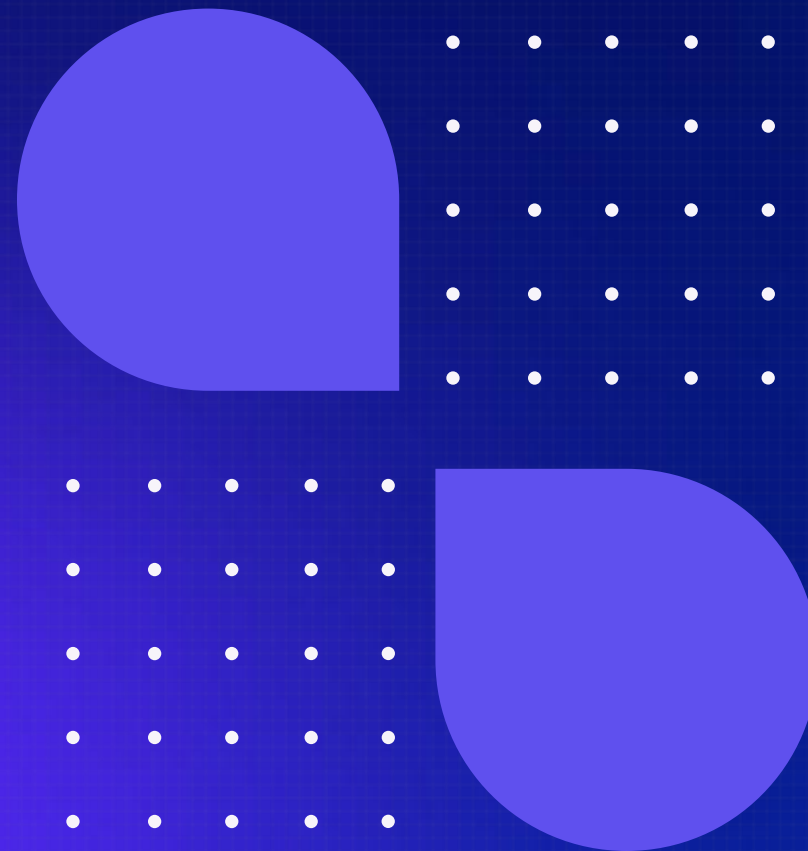


Athena Code Assistant

보안·비용·감사 대응까지
On-Premise 개발 환경을 지원합니다.



Uracle is : 안정성 및 성장성

유라클은 모바일 플랫폼을 비롯한 다양한 업종과 분야에서 시스템 구축 사업을 수행하고 역량을 축적해 온 25년차 전문기업입니다.

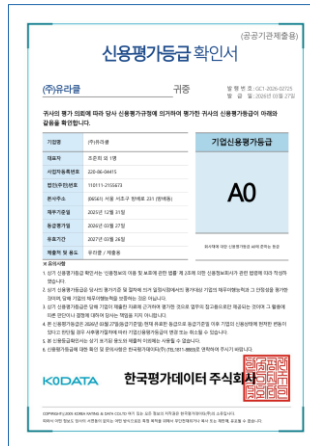


uracle

AI 플랫폼·LLM·자체 개발 솔루션 보유

회사명	(주)유라클	대표자	조준희, 권태일
주소	서울시 서초구 방배로 231 (방배동)		
사업분야	AI 솔루션 개발, 시스템 통합 구축·유지, 모바일 시스템 구축·개발		
설립년도 / 사업기간	2001.01 ~ 2026.03 (25년 2개월)		
보유인력현황	총 254명 (개발:218/지원인력:36)		

'25 신용평가등급
A0



최근 3년 평균 매출
350억+



2001 ~ 2009

2010 ~ 2019

2020 ~ 2025

- '08 벤처기업대상 중소기업청장표창 수상
- '07 (주)유라클 상호변경
- '07 정보통신부 핵심동력개발사업 차세대기술선정
- '01 '모바일로' 금융감독원 금융 신상품 대상
- '01 (주)아이엠넷피아 설립

- '19 SW산업발전유공자 대통령 표창 수상
- '19 대한민국 소프트웨어 기업경쟁력 대상
- '16 SW기업경쟁력 대상 IT솔루션 분야 최우수상
- '15 모바일 전자정부 행정자치부 장관 표창
- '11 스마트앱어워드2011 금융서비스 최우수상

- '25 생성형 AI 플랫폼 출시 (Athena AI 플랫폼)
- '25 AI 코드어시스턴트 제품 출시 (Athena Code)
- '24 고려대 HI AI 공동연구소 설립 및 LLM 개발
- '24 (주)유라클 KOSDAQ 상장 KOSDAQ
- '21 한국소프트웨어산업협회 회장 취임 (조준희 회장)

사업 분야

모바일·AI 플랫폼 등 SW, 시스템 구축, 운영 및 유지보수

주요 제품

AI Product Suite

Mobile Product Suite

주요 고객사

• 삼성



• 현대



• Major Financial Institutions



• Other Industries



Uracle is : 고객사 현황

1,000여개의 산업 별 전문 지식과 경험으로 유라클이 AI 도입 여정의 새로운 파트너가 되겠습니다.

공공

Public Sector



금융

Finance



제조

Manufacturing



서비스

Service



기타

Other Industries

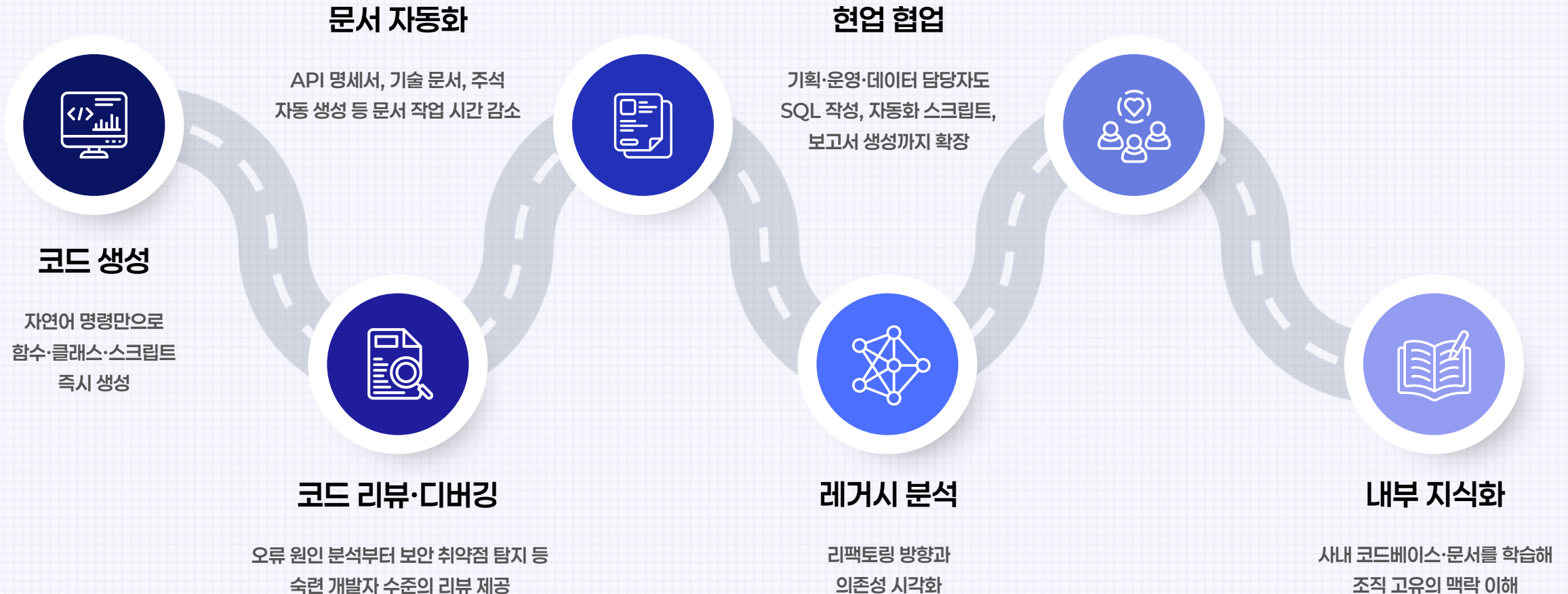


Uracle is : AI 사업 수행 실적

2025년 1월부터 AI사업 시작 후 상반기 20건의 PoC를 거쳐, 상용 레퍼런스를 빠르게 확장하고 있습니다.

사업명	발주처	투입 S/W	주요내용
AI 개발플랫폼 구축	KODATA(한국평가데이터)	AI Platform, AI Code Assistant	AI 개발플랫폼(LLMOps) 및 지식관리 시스템 구축
현대힐스테이트 AI 분양상담사 구축	현대건설	AI Platform	AI 분양상담 챗봇 대고객서비스 구축, 하반기 확산 사업 예정
협회 통합정보시스템 구축	대한의사협회	AI Platform	AI 플랫폼 기반 민원인·직원·광고심의 신청자용 AI 챗봇 구축
AI기반 FDS 검사시스템 구축	우리은행	AI Platform	과거 사례 시나리오 기반 FDS 검사시스템 구축
AI 코드 어시스턴트 도입	KT 스카이라이프	AI Code Assistant	코드어시스턴트를 통한 IE/JAVA기반 구형 웹서비스 전환
AI 서비스 플랫폼 구축	KCB(코리야크레딧뷰로)	AI Platform	No code 기반의 전사 AI 서비스 구축용 플랫폼 구축
AIOps 플랫폼 및 고객 지원 챗봇 개발	인텔리안테크	AI Platform	생성형 AI 플랫폼 및 고객 지원용 RAG 챗봇 구축
AI 서비스 내재화 플랫폼 구축	현대건설	AI Platform	전사 생성형 AI 플랫폼 구축
新 모바일 워크플레이스 구축	GS건설	AI Platform	전 임직원용 AI 비서(기사 요약, 앱 검색) 구축
생성형 AI 시범과제 구축 영역	메리츠증권	AI Platform	실시간 문서 번역/요약 Agent 구축
AI기반 차세대 안전신고 및 위험징후 분석 서비스 기술 개발	행정안전부	LG AI 연구원 EXAONE LLM	신고 Agent 개발 및 차세대 안전신문고 시스템 연동
지식 검색용 RAG 챗봇 PoC	한국과학기술단체총연합회	AI Platform	정책 보고서 지식 검색 RAG 구축
홈페이지 고객용 RAG 챗봇 PoC	한국인공지능·SW산업협회	AI Platform	고객용 지식 검색 RAG 구축
경력관리 지식검색용 챗봇 PoC	한국건설기술인협회	AI Platform	경력 관리용 RAG 검색 테스트
학사행정 지식검색용 RAG 챗봇 PoC	서울과학기술대학교	AI Platform	학사행정 지식검색 RAG 테스트
지식 검색용 RAG 챗봇 PoC	한국투자저축은행	AI Platform	내부직원용 지식 검색 RAG 구축
내부 지식 검색용 RAG 챗봇 PoC	NH저축은행	AI Platform	사내 문서 지식검색 RAG 구축
ERP 매뉴얼 지식검색 PoC	두산퓨얼셀	Athena GenAI Platform	ERP 매뉴얼 검색 용 RAG 구축
지식 검색용 RAG 챗봇 PoC	다이소	Athena GenAI Platform	직원용 매뉴얼 검색 RAG 테스트
품질검사 AI Assistant PoC	두산에너빌리티	AI Platform	검사원 선정 업무 자동화, 검사체크리스트 생성

코딩을 넘어, 조직 전체를 위한 AI 도구



| 기존 AI 코드 어시스턴트가 엔터프라이즈 현장에서 멈추는 이유

보안망내 외부 AI 서비스 접속 차단

클라우드 기반 AI 코드 어시스턴트는 외부 API와 상시 연결되는 구조로, 보안망 내부 환경에서는 접속 자체가 허용되지 않습니다. 특히 망분리가 요구되는 환경에서는 도입 검토 단계에서부터 제외됩니다.

팀·조직 단위 AI 사용 통제 불가

개인 계정 기반 SaaS 도구는 조직 단위의 접근 권한 설정, 사용량 제한, 모델 통제가 구조적으로 어렵습니다. 누가 어떤 모델로 무엇을 했는지 관리자가 파악하고 통제할 수 있는 수단이 없습니다.

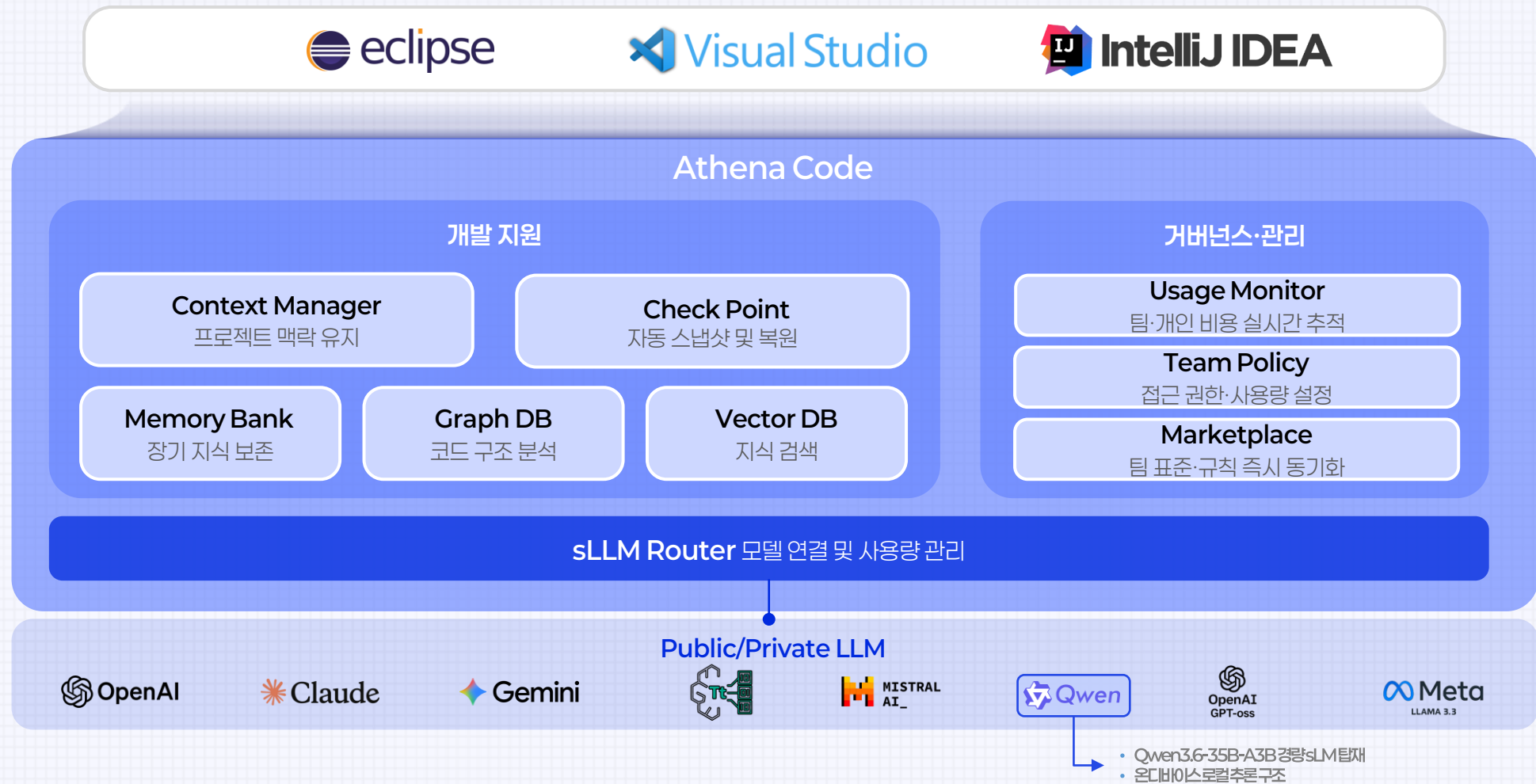
AI 생성 코드의 품질·보안 신뢰성 확보 어려움

AI가 생성한 코드에 대한 검증 체계 없이 그대로 활용할 경우, 코드 품질 편차와 보안 취약점이 발생합니다. 특히 금융·공공 환경에서는 AI 생성 코드의 감사 추적과 품질 통제가 규정상 필수적으로 요구됩니다.

대규모 레거시 코드베이스 파악의 한계

SI 프로젝트는 다양한 모듈·오픈소스·레거시 라이브러리가 복잡하게 얽혀 있습니다. 파일 단위로만 동작하는 기존 AI 도구는 전체 시스템 구조와 의존성을 파악하지 못해, 코드 수정 시 예상치 못한 영향 범위가 발생합니다.

엔터프라이즈를 위한 아테나 코드 어시스턴트



| 기존 AI 코드 어시스턴트의 한계 해결



엔터프라이즈
보안 완전 충족

완전한 폐쇄망 운영

주요 FOCUS

- 외부 인터넷 없이 내부망 독립 작동
- Air-Gapped 환경 즉시 사용 가능

적용 기술

- Public/Private LLM 선택적 연동
- sLM Router 기반 모델 관리

감사 대응 자동화

주요 FOCUS

- 모든 AI 사용 이력 자동 로깅 및 Export
- 관리자 대시보드를 통한 관리 및 관제

적용 기술

- 감사 로그 자동화
- Team Policy 기반 권한 관리



조직 전체
개발 역량 강화

레거시 코드 맥락 이해

주요 FOCUS

- 전체 시스템 구조 의존성을 통합 파악
- 코드 수정 시 영향 범위 즉시 확인 가능

적용 기술

- Graph DB 기반 코드 관계 시각화
- Memory Bank·Vector DB 통합 지식 검색

AI 사용 가시성 확보

주요 FOCUS

- 팀·개인별 사용량·비용 실시간 모니터링
- 사용자별 모델 접근 권한 및 한도 설정

적용 기술

- Usage Monitor 대시보드
- 사용자별 모델 접근 한도 설정

개발자는 더 빠르게, 조직은 더 안전하게

01

반복 작업 자동화

맥락과 이력이 유지되는 개발 환경

컨텍스트 설정

체크 포인트

/ 명령어

02

레거시·대규모 코드 분석

수십만 줄 코드베이스의 즉각적인 구조 파악

그래프 DB

메모리 뱅크

03

보안·감사 대응

모든 AI 사용 이력의 완전한 추적과 증빙

감사 로그

컴플라이언스

04

비용·사용량 통제

팀·사용자 단위의 실시간 비용 가시성 확보

sLLM 라우터

AI 사용 모니터링

비용 한도 설정

05

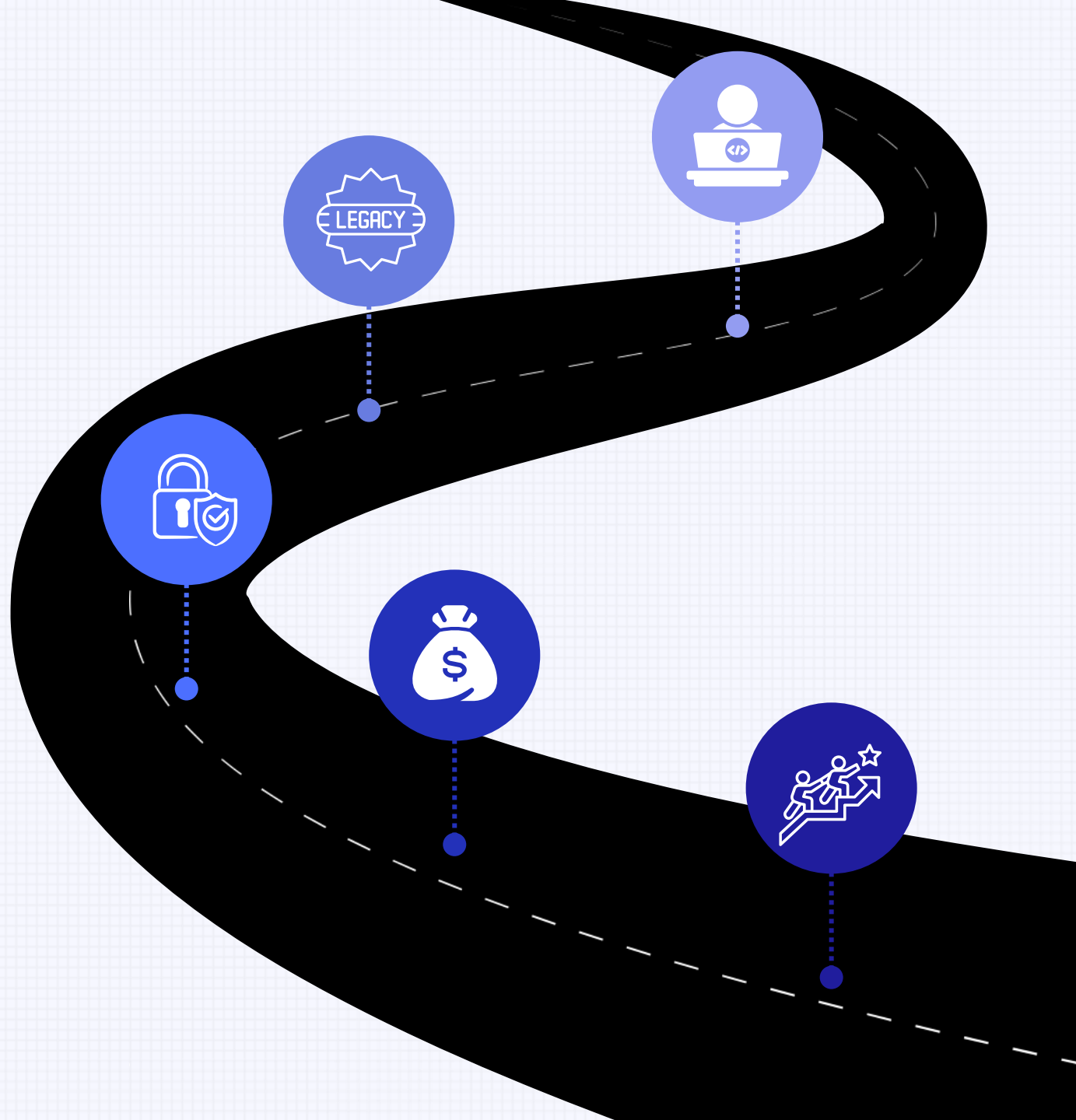
조직 환경 맞춤화

조직 개발 방식 그대로의 AI 환경 구성

Rules

MCP

마켓 플레이스



작업모드

목적에 따라 AI의 동작 범위를 구분하여 불필요한 파일 접근과 코드 변경을 방지합니다.

Code 모드

코드 생성·수정·리팩토링에
최적화된 구현 전용 모드로,
파일 편집과 버그 수정까지
수행하여 개발 속도 향상

Architect 모드

시스템 설계·기술 스택
선정·프로젝트 구조 계획에
최적화된 설계 전용 모드로,
구현 전 설계 완성도 확보

Ask 모드

개념 설명·기술 질의에
최적화된 답변 전용 모드로,
읽기와 분석만 수행하여
불필요한 코드 변경 방지

모드

Athena의 행동을 맞춤화하는 전문화된 페르소나.
Ctrl + . 다음 모드용, Ctrl + Shift + . 이전 모드용

Architect

Plan and design before implementation

Code

Write, modify, and refactor code

Ask

Get answers and explanations



으로 코드를 생성, 리팩터링 및 디버깅합니다. 자세한 내용은 문서를 확인하세요.

계획, 설계, 전략 수립

코드 작성, 수정, 리팩토링

설명, 문서화, 기술적인 질문에 대한 답변

컨텍스트 설정

AI의 컨텍스트 창에 포함되는 정보를 제어하여 token 사용량과 응답 품질에 영향을 미칩니다.

컨텍스트 범위 세밀 조정

열린 탭·작업 공간 파일·동시 읽기 개수 조정으로 비용과 응답 품질, 최적의 성능 확보

보안 영역 자동 차단

민감 파일·디렉토리를 AI 접근 및 검색에서 원천 배제하여 보안 거버넌스 투명성 확보

장시간 작업 안정성 확보

컨텍스트 사용량 임계치 도달 시 이전 대화 자동 요약·압축으로 끊김 없는 작업 지속

컨텍스트 범위 직접 조정

필요한 정보만 AI에 전달해 불필요한 토큰 낭비 방지

민감 파일 AI 참조 제외

보안 파일·기밀 코드가 컨텍스트에 포함되지 않도록 차단

컨텍스트 자동 압축

임계값 조정으로 긴 대화에서도 응답 품질 유지, 토큰 비용 절감

열린 탭 컨텍스트 제한

20

컨텍스트에 포함할 VSCode 열린 탭의 최대 수. 높은 값은 더 많은 컨텍스트를 제공하지만 token 사용량이 증가합니다.

작업 공간 파일 컨텍스트 제한

200

현재 작업 디렉토리 세부 정보에 포함할 파일의 최대 수. 높은 값은 더 많은 컨텍스트를 제공하지만 token 사용량이 증가합니다.

동시 파일 읽기 제한

5

read_file 도구가 동시에 처리할 수 있는 최대 파일 수입니다. 높은 값은 여러 작은 파일을 읽는 속도를 높일 수 있지만 메모리 사용량이 증가합니다.

☒ 목록 및 검색에서 .athenignore 파일 표시

활성화되면 .athenignore의 패턴과 일치하는 파일이 잠금 기호와 함께 목록에 표시됩니다. 비활성화되면 이러한 파일은 파일 목록 및 검색에서 완전히 숨겨집니다.

파일 읽기 자동 축소 임계값

-1 줄 ☒ 항상 전체 파일 읽기

☒ 지능적 컨텍스트 압축 자동 트리거

⚙️ 압축 트리거 임계값

글로벌 기본값 (모든 프로필)

90%

컨텍스트가 이 비율에 도달하면 사용자 정의 설정이 없는 한 모든 프로필에 대해 자동으로 압축됩니다

체크포인트

파일 수정 전 자동으로 스냅샷을 생성하여 언제든지 이전 상태로 안전하게 복원합니다.

자동 스냅샷 생성

파일 수정이 시작되기 전 Git 기반 스냅샷을 자동 생성하여 모든 변경 이력 보존

즉각적인 상태 복원

원하는 시점의 체크포인트로 되돌아가 실수나 잘못된 수정으로 인한 손실 방지

작업 흐름 중단 없는 안정성

별도 설정 없이 작업 중 자동으로 관리되어 개발 흐름을 끊지 않고 안정적인 작업 환경 유지

체크포인트

☒ 자동 체크포인트 활성화

활성화되면 Athena는 작업 실행 중에 자동으로 체크포인트를 생성하여 변경 사항을 검토하거나 이전 상태로 되돌리기 쉽게 합니다.

✓ API 요청

초기 체크포인트 현재

자동 스냅샷 생성·복원

작업 시작 시 초기 체크포인트 자동 생성

💡 생각 중

> 작업: Help improve the code, file, or behavior described by the user's request. If the target is clear, inspect the relevant files as needed and suggest concrete...
14.4k 128.0k 5/5

- Todo list
- 현재 코드 분석
- 입력 유효성 검사 추가
- 에러 처리 방식 개선 (Error 객체 사용)
- NaN/Infinity 처리 추가
- JSDoc 타입 힌트 보완

체크포인트

✓ API 요청

Athena가 이 파일을 편집하고 싶어합니다.

calculator.js

체크포인트

✓ API 요청

차이점 보기

체크포인트 복원

파일 복원

프로젝트 파일을 이 시점에 찍힌 스냅샷으로 복원합니다.

파일 및 작업 복원

프로젝트 파일을 이 시점에 찍힌 스냅샷으로 복원하고 이 지점 이후의 모든 메시지를 삭제합니다.

컨텍스트 멘션 & 슬래시 명령어 @ 멘션으로 AI가 참조할 대상을 지정하고, 명령어로 반복 작업을 즉시 실행합니다.

다양한 컨텍스트 소스 연결

파일·폴더·터미널·Git·URL
을 @ 기호 하나로 참조하여
AI에게 필요한 정보 전달

반복 작업의 단축 실행

자주 쓰는 작업 패턴을
/(슬래시) 명령어로 등록하여
프롬프트 작성 없이 즉시 실행

팀 표준 명령어 공유

팀에서 정의한
Rules·Commands를
마켓플레이스로 공유하여
모든 팀원 IDE에 즉시 반영

슬래시 명령

사용자 지정 워크플로와 작업을 신속하게 실행하기 위해 슬래시 명령을 관리합니다.

전역 명령

새 전역 명령...

Slash Commands

사용자 지정 워크플로와 작업을 신속하게 실행하기 위해 슬래시 명령을 관리합니다.

Commands

/init
코드베이스를 분석하여 AI 어시스턴트를 위한 간결한 AGENTS.md 파일을 생성합니다

/create-command
사용자 정의 슬래시 커맨드를 생성하고 전문가 수준의 프롬프트로 변환합니다

/create-rules
사용자 정의 규칙을 생성하고 전역 또는 프로젝트별 디렉토리에 저장합니다

/improve
현재 코드 선택에 대한 개선 사항을 제안합니다

/explain
현재 코드 선택이 어떤 동작을 하는지 설명합니다

커스텀 명령어 즉시 실행

/ 입력 시 등록된
Commands 목록 자동 표시

push_test_data.json - 다양한 푸시
push_test_data.csv - 대량 테스트를
push_test_data_multiplex.csv - D



Athena가 새 파일을 만들고 싶어합니다:

컨텍스트 즉시 연결

@ 입력 시 파일·폴더 등
소스 목록 자동 표시

지정

Morpheus MADP Client | URACLE

@

메모리 뱅크

중요한 결정·작업·패턴을 구조화하여 세션이 달라져도 일관된 이해를 제공합니다.

프로젝트 맥락의 영구 보존

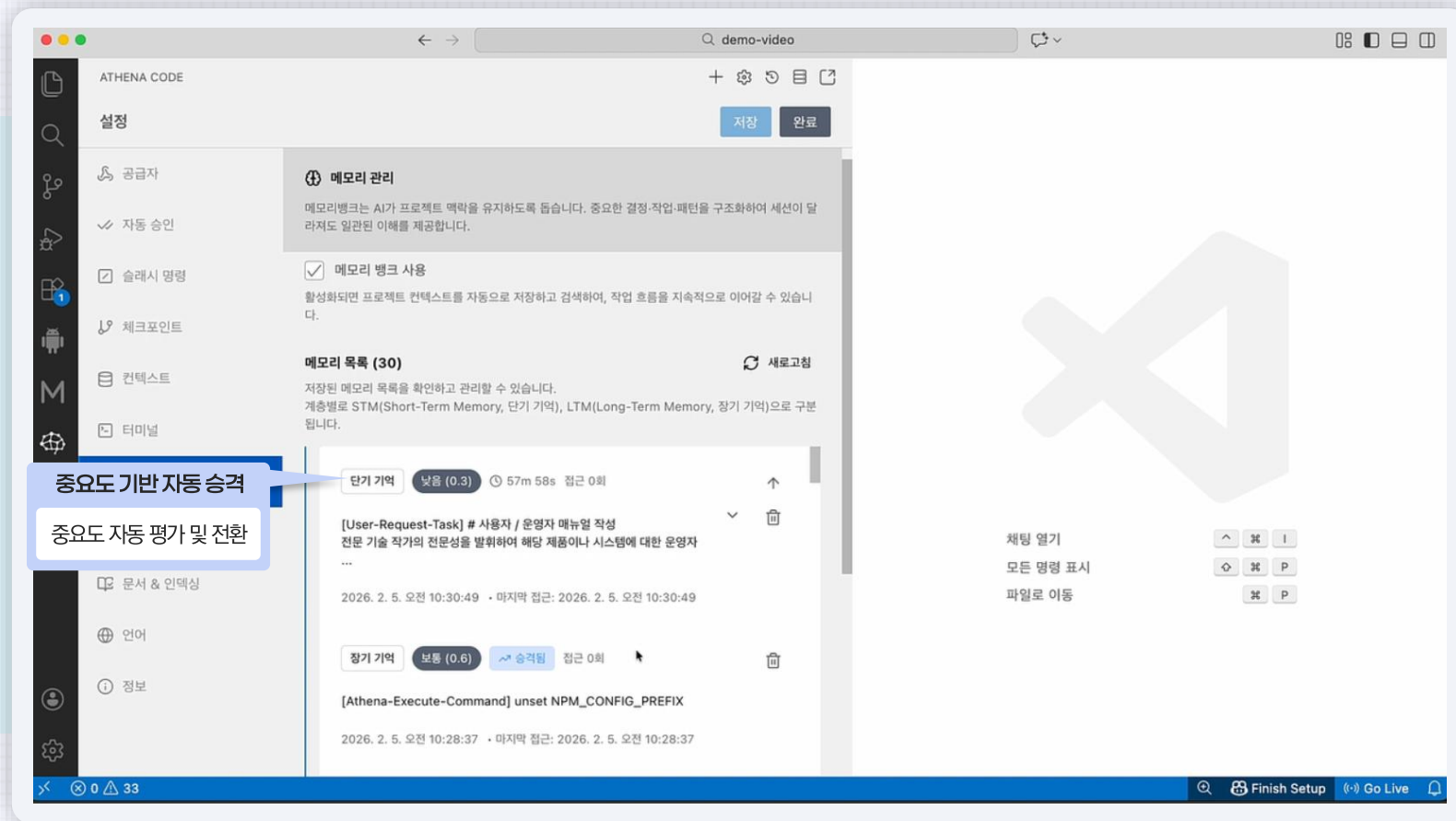
세션이 종료되어도
STM(단기 기억)·LTM
(장기 기억) 계층 구조로 설계
결과와 업무 이력을 지속 보존

중복 없는 지식 무결성

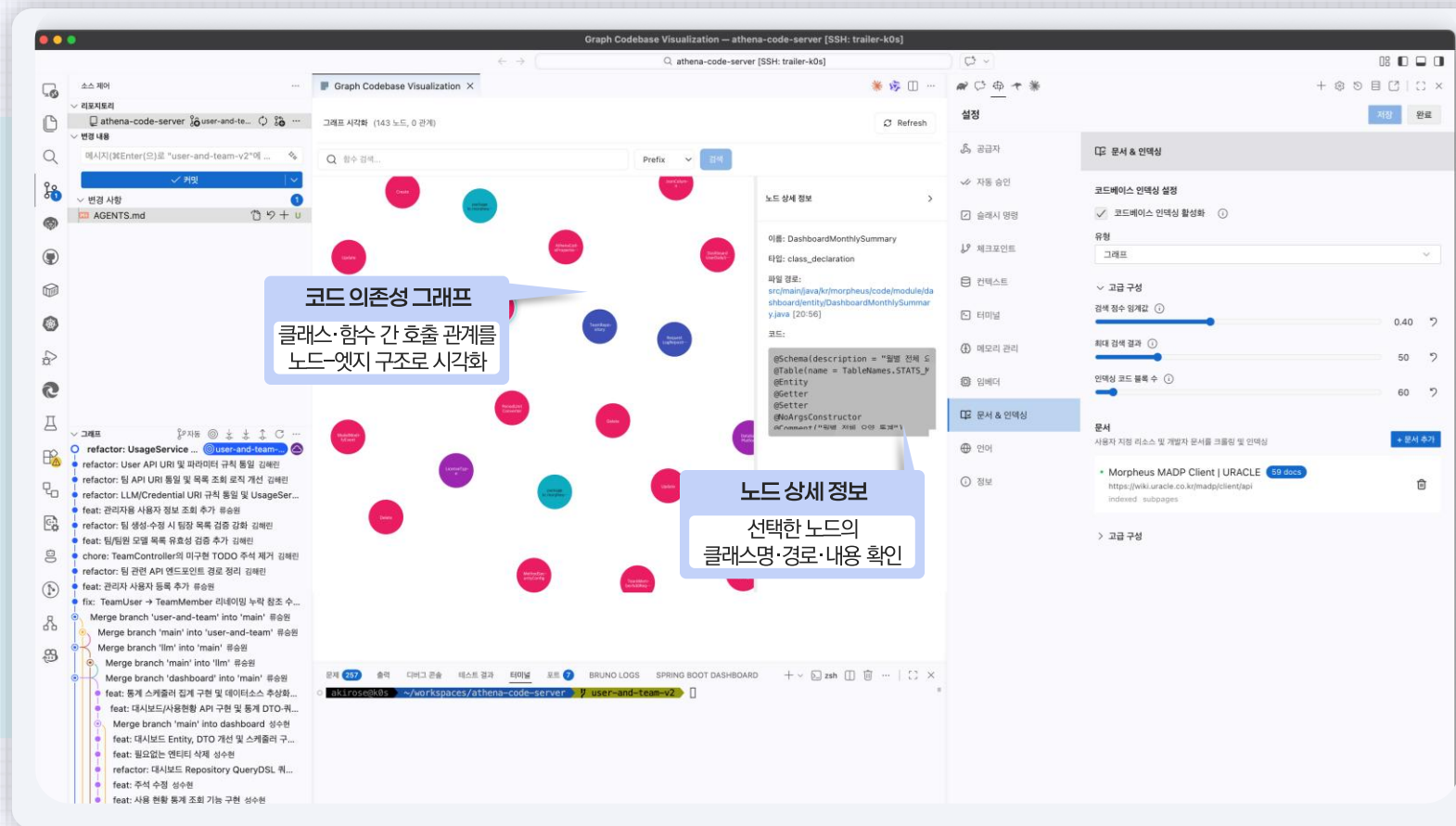
벡터 유사도 기반 중복 감지로
기존 내용과 충돌하는 정보를
자동 정리하여 최신 지식
무결성 유지

즉각적인 맥락 검색

프로젝트별 독립된 지식
베이스에서 관련 맥락을 즉시
탐색하여 작업 흐름 중단 없이
연속성 유지



대규모 레거시 코드도 의존성
그래프로 자동 인덱싱하여
신규 투입 개발자의 파악 시간
대폭 단축



감사 로그 & 컴플라이언스

팀 전체의 AI 사용 이력을 실시간으로 추적하고, 요청별 상세 내역을 언제든 조회·Export합니다.

전체 요청 이력 추적

일시·팀·사용자·모델·토큰·비용·상태를 기준으로 모든 AI 요청 이력을 실시간 기록 및 조회

요청별 상세 내역 확인

개별 요청의 Request·Response·Trackback 전문을 열람하여 응답 품질 및 이상 여부 검토

감사 대응 Export

기간·팀·사용자·모델별 필터링 후 로그를 Export하여 내부 감사 및 컴플라이언스 대응

요청이력

요청 이력 전체 조회

일시·사용자·토큰·비용
항목별 필터링 및 이력 표시

2026.05.21



일주일

30일

90일

이름

팀 검색

모델 선택

상태 선택

검색결과 219

일시	팀	사용자	모델	Token	비용(\$)	상태	상세보기
2026-05-28 08:49	-	yongbum7186@uracle.co.kr	qwen3	53,641	0.016	성공	상세
2026-05-28 08:48	-	yongbum7186@uracle.co.kr	qwen3	51,888	0.014	성공	상세
2026-05-28 08:48	-	yongbum7186@uracle.co.kr	qwen3	11,889	0.004	성공	상세
2026-05-28 08:48	-	yongbum7186@uracle.co.kr	qwen3	10,021	0.003	성공	상세
2026-05-28 08:48	-	yongbum7186@uracle.co.kr	qwen3	27,792	0.008	취소	상세
2026-05-28 08:48	-	yongbum7186@uracle.co.kr	qwen3	9,886	0.003	성공	상세
2026-05-27 15:46	-	yongbum7186@uracle.co.kr	qwen3	57,570	0.017	성공	상세
2026-05-27 15:45	-	yongbum7186@uracle.co.kr	qwen3	51,035	0.015	성공	상세
2026-05-27 15:45	-	yongbum7186@uracle.co.kr	qwen3	31,103	0.009	성공	상세
2026-05-27 15:45	-	yongbum7186@uracle.co.kr	qwen3	30,484	0.009	성공	상세

요청이력 상세

요청 ID chatlog-4470a741-c76d-470e-a297-f8ba8861a947
Model qwen3.5-35b
Model ID 49
Start Time 2026-05-28 08:49:52
End Time 2026-05-28 08:50:01
Tokens 53,641
Cost 0

Request

Response

Trackback

사용자·팀 단위로 비용 한도를
설정하여 예산 초과 전 사전
차단 및 낭비 방지

Total 0,47\$ / 1,72M

Rules

팀의 코딩 규칙을 문서로 정의하면, 시가 모든 개발자의 IDE에서 자동으로 적용합니다.

코딩 표준 중앙 배포

팀장·시니어가 정의한 규칙을
전체 팀원 IDE에 즉시
반영하여 코드 품질의 일관성
확보

실시간 규칙 적용

코드 작성 시점에 규칙 위반을
즉시 감지하고 자동 수정
제안으로 리뷰 부담 최소화

프로젝트 단위 분리 적용

전체 프로젝트 공통 규칙과
개별 프로젝트 전용 규칙을
분리하여 유연한 표준 관리

ATHENA CODE

> 작업: 이 규칙 기반 사용자 로그인 함수 만들어줘

15.4k

제공 기능

함수	설명
<code>registerUser()</code>	사용자 등록 (ID, 비밀번호, 추가 정보)
<code>authenticateUser()</code>	사용자 로그인 인증
<code>removeUser()</code>	사용자 삭제
<code>getAllUsers()</code>	전체 사용자 목록 조회

팀 코딩 규칙 준수 사항

- ✓ `console.log` 대신 `logger.info()` 및 `logger.error()` 사용
- ✓ 함수명은 동사로 시작 (`registerUser`, `authenticateUser`, `removeUser`, `getAllUsers`)
- ✓ 변수명은 camelCase 사용 (`userId`, `password`, `userDatabase`, `additionalInfo`)

테스트 결과

```
[INFO] 사용자 등록 완료: admin, user1, user2
[INFO] 로그인 성공: admin, user1
[ERROR] 잘못된 비밀번호 시 적절한 오류 처리
[ERROR] 존재하지 않는 사용자 시 적절한 오류 처리
```

markdown

팀 코딩 규칙

- `console.log` 사용 금지. 반드시 `logger.info()` 또는 `logger.error()` 사용
- 함수명은 반드시 동사로 시작 (`getData`, `fetchUser`, `processOrder` 등)
- 변수명은 camelCase 사용

코드 리뷰

```
function login() {
  console.log('로그인 시도')
  var user_name = 'test'
}
```

발견된 문제점

항목	문제	수정 제안
<code>console.log()</code>	팀 규칙 위반: <code>console.log</code> 사용 금지	<code>logger.info()</code> 또는 <code>logger.error()</code> 사용
<code>var</code>	구식 변수 선언	<code>const</code> 또는 <code>let</code>
<code>user_name</code>	snake_case 표기	camelCase 사용
함수 로직	실제 로그인 로직 없음	실제 인증 로직 추가

수정된 코드

```
const logger = {
  info: (message) => console.log(`[INFO] ${message}`),
  error: (message) => console.error(`[ERROR] ${message}`)
};
```

핵심 기능 비교

각 제품은 서로 다른 환경과 목적에 최적화되어 있습니다. 공개된 자료를 기반으로 주요 기능 지원 여부를 정리한 참고 자료입니다.

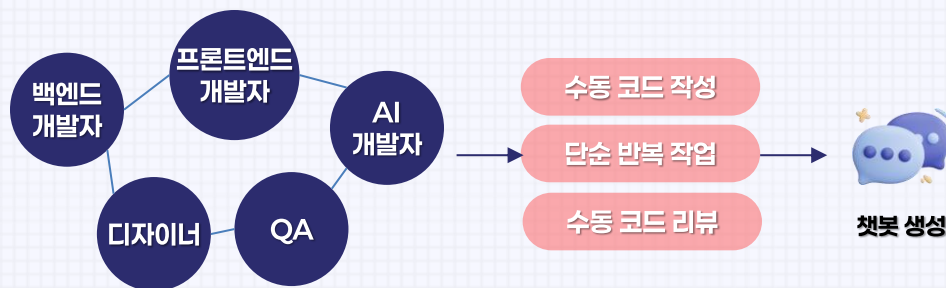
※ 기준일: 2026년 6월 / 공식문서 및 발표 자료 기반

기능	 CURSOR	 Claude	 Copilot	아테나 코드
코드 자동완성	✓	✓	✓	✓
코드베이스 전체 맥락 이해	✓	✓	✓	✓
IDE 지원 (VS Code)	✓	✓	✓	✓
IDE 지원 (IntelliJ·JetBrains)	✓	✓	✓	✓
IDE 지원 (Eclipse)	X	X	✓	✓
장기 맥락 보존 (세션 간)	✓	✓	X	✓
완전한 폐쇄망·Air-Gapped 운영	X	X	X	✓
온프레미스 배포	△ (코드 실행만, AI 추론은 클라우드 경유)	X	X	✓
팀 사용량·비용 실시간 모니터링	△	△	△	✓
감사 로그 자동화·Export	△	△	△	✓
팀 코딩 표준 동기화	△	X	X	✓
Graph DB 코드 구조 시각화	외부 툴 별도 설치 연동 필요			✓

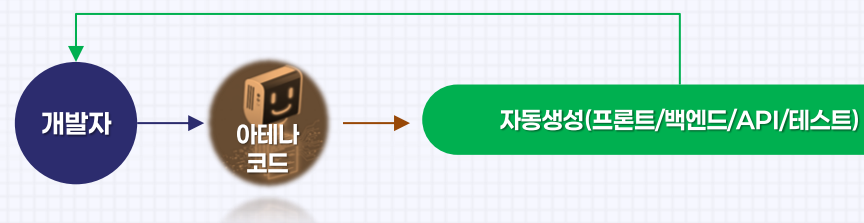
△: 제한적 지원 또는 상위 플랜(Enterprise)에서만 가능

내부 챗봇 개발 적용 예시

기존 운영 방식



향후 운영방식 — 아테나 코드 도입 후



다수 전문 역할(기획·UI/UX·FE·BE·AI·QA 등) 10명 **분절 운영**



프론트엔드·백엔드·API·테스트 코드 100% **수동 작성**



외부 SaaS AI 도구 **접속 불가**



PM·AI 엔지니어·풀스택·QA 4인 **고효율** 팀으로 전환



프론트엔드·백엔드·API 반복 코드의 **70% 자동 생성** 및
테스트 코드 스캐폴딩부터 API 문서·화면 플로우 초안 자동 생성

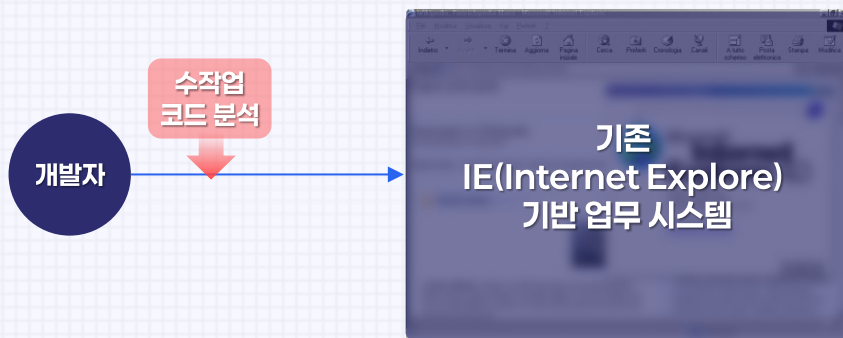


개발망 폐쇄 환경에서 온프레미스로 **안전**하게 운영

내부 PoC 기준 개발 공수 약 78% 절감 및 기간 50% 단축

레거시 웹 전환 적용 예시

기존 운영 방식



향후 운영방식 — 아테나 코드 도입 후



- Ⓔ IE 기반 JSP 구조로 운영 중인 주요 업무 시스템
- 🔧 화면 분석·리팩터링 전 과정을 개발자가 **수작업**으로 진행
- 🚫 개발망 인터넷 차단으로 SaaS 기반 AI 솔루션 **사용 불가**

- 🔄 IDE 내에서 AI가 JSP 로직·UI **자동 분석 및 변환** 지원
- 🌐 Chrome/Spring 기반 최신 구조로 점진적 **자동 변환·리팩터링** 수행
- 🌍 개발망 폐쇄 환경에서도 안전하게 운영, 비용·기간 **대폭 절감**

개발망 폐쇄 환경에서 레거시 전환 비용·기간·리스크 대폭 축소

다양한 규모의 코드 어시스턴트 환경 지원

대규모 환경 뿐만 아니라, GPU 제약이 있는 중·소규모 환경까지 지원 가능합니다.

※ Qwen3.6-35B-A3B 경량 sLM 기준

구분	디바이스	가용 메모리	AI 모델 메모리 사용량	권장 사용 인원	특징
개인용	AMD AI Max 395+	128GB 공유	~20GB	1인 전용	고성능 노트북
	MacBook M5 Pro	64GB 공유			개인 노트북 환경 최적화
팀 공용 서버	Mac Studio M3 Ultra	96GB 공유		3인	소규모 팀 공용, 다중 세션 최적화
	RTX PRO 6000	96GB 전용 VRAM		6~10인	중규모 팀 운영 최적화, 최대 성능

💡 코드 어시스턴트는 사용자가 타이핑하는 동안 GPU를 점유하지 않습니다. 실제 GPU 사용 시간은 전체의 10~20% 수준으로, RTX PRO 6000 1대로 6~10인 팀을 실질적으로 지원할 수 있습니다.

감사합니다.

H. uracle.co.kr

E. sales@uracle.co.kr

T. 02-3479-4400