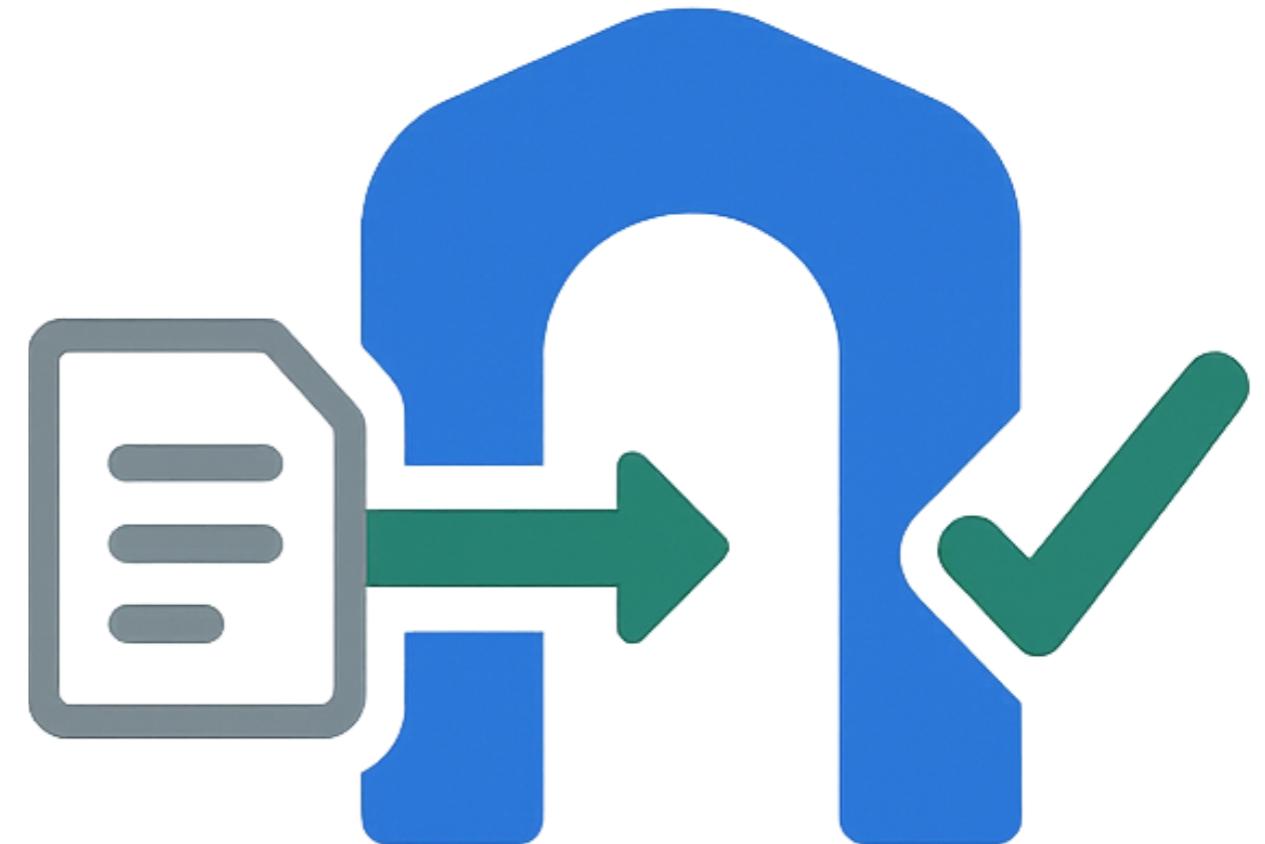


SpecGate

설계와 구현을 연결하는 문



조윤영

2025.09.16 클테코

목차

1. 문제 인식
2. 솔루션: SpecGate
3. 핵심 기능: 3-Phase 문의 작동 원리
4. 아키텍처
5. 화면 Mockup
6. 기대효과
7. 로드맵
8. Q&A

문제 인식

현재 상황



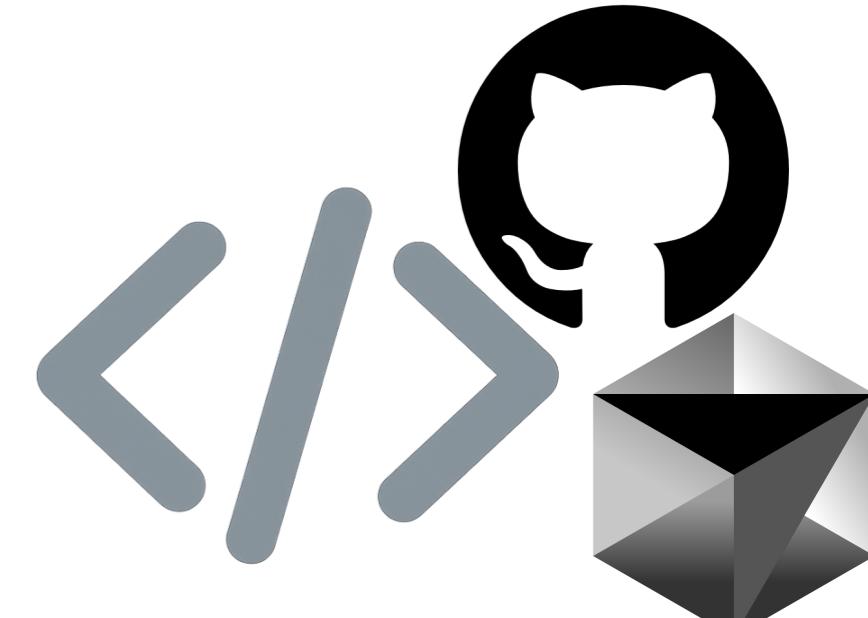
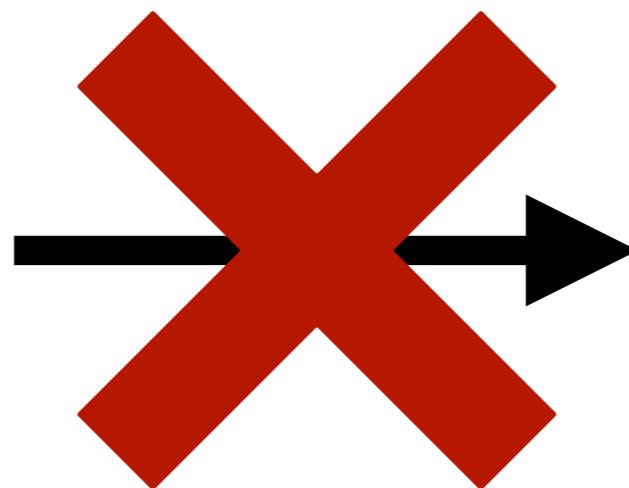
Confluence 설계 문서

API 설계서

아키텍처 문서

ERD 다이어그램

...

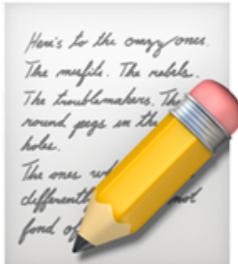


실제 코드

시간이 지날수록 설계와 코드의 불일치 증가

문제 인식

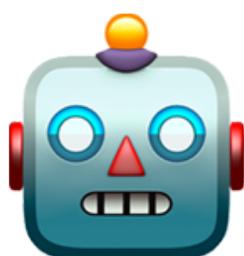
기존 방식의 한계



문서 작성만으로는 지속적 준수 어려움



코드 리뷰에서 수동 지적 → 비용 크고 누락 찾음



Cursor만으로는 설계 밖 코드 생성 위험

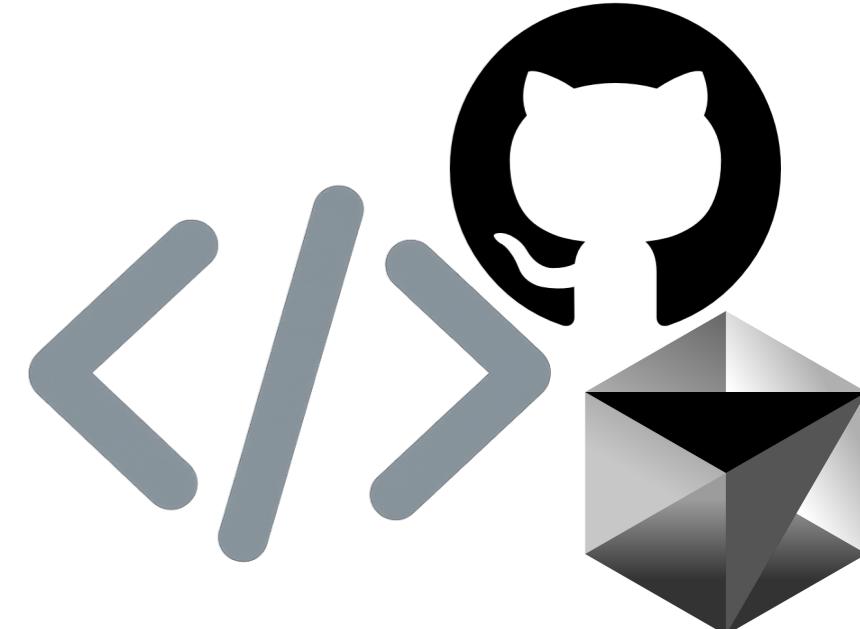
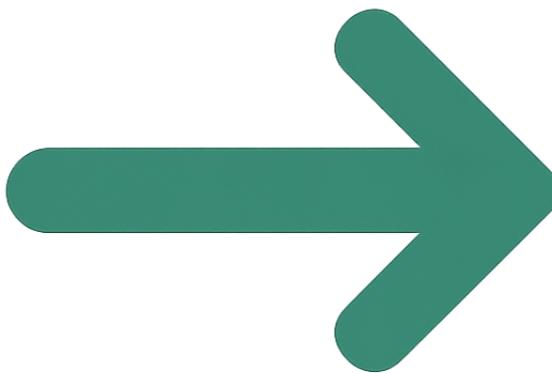
솔루션: SpecGate



Confluence 설계 문서

- API 설계서
- 아키텍처 문서
- ERD 다이어그램

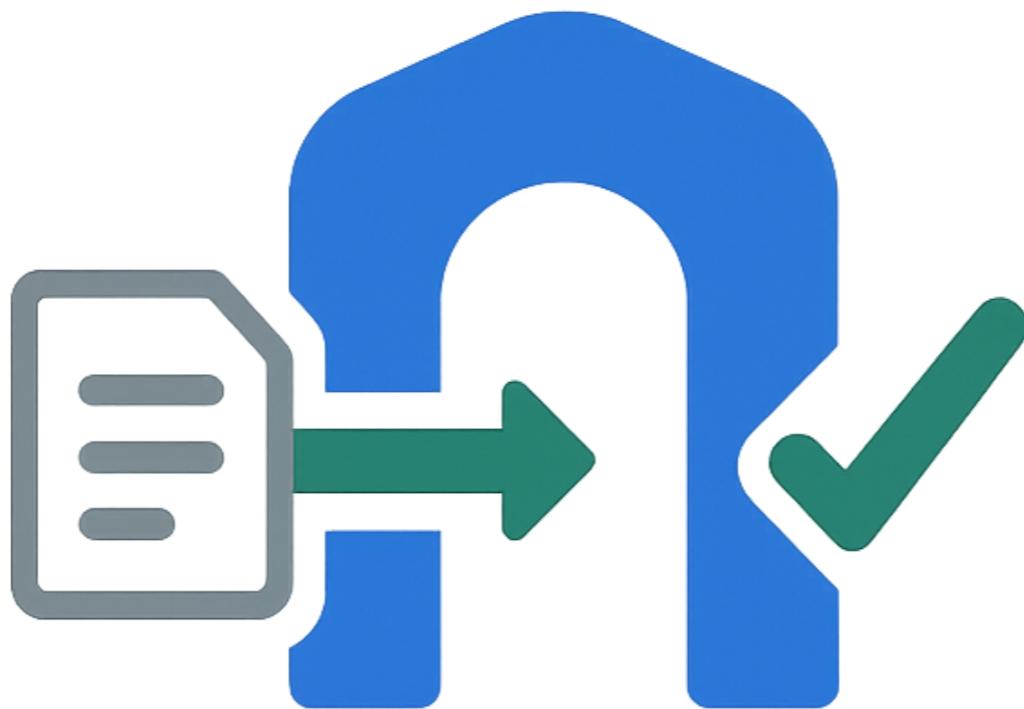
...



실제 코드

설계 문서와 코드를 연결하는 문

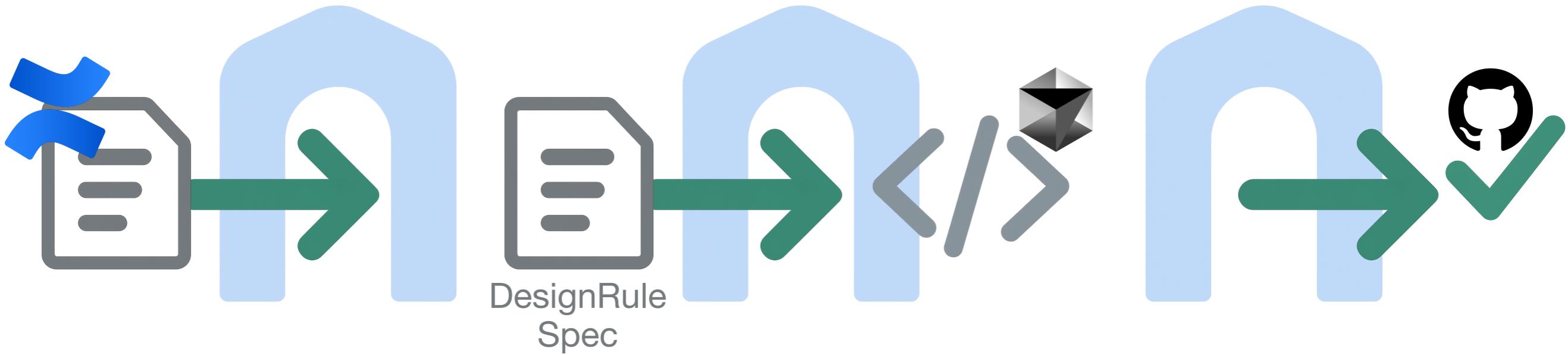
솔루션: SpecGate



SpecGate

솔루션: SpecGate

3-Phase 구조



1) 문서 정규화

Confluence API로 설계 문서 자동 수집
SpecLint로 품질 검사 (0-100점)
HTML→MD 변환으로 정규화

2) 개발 가이드

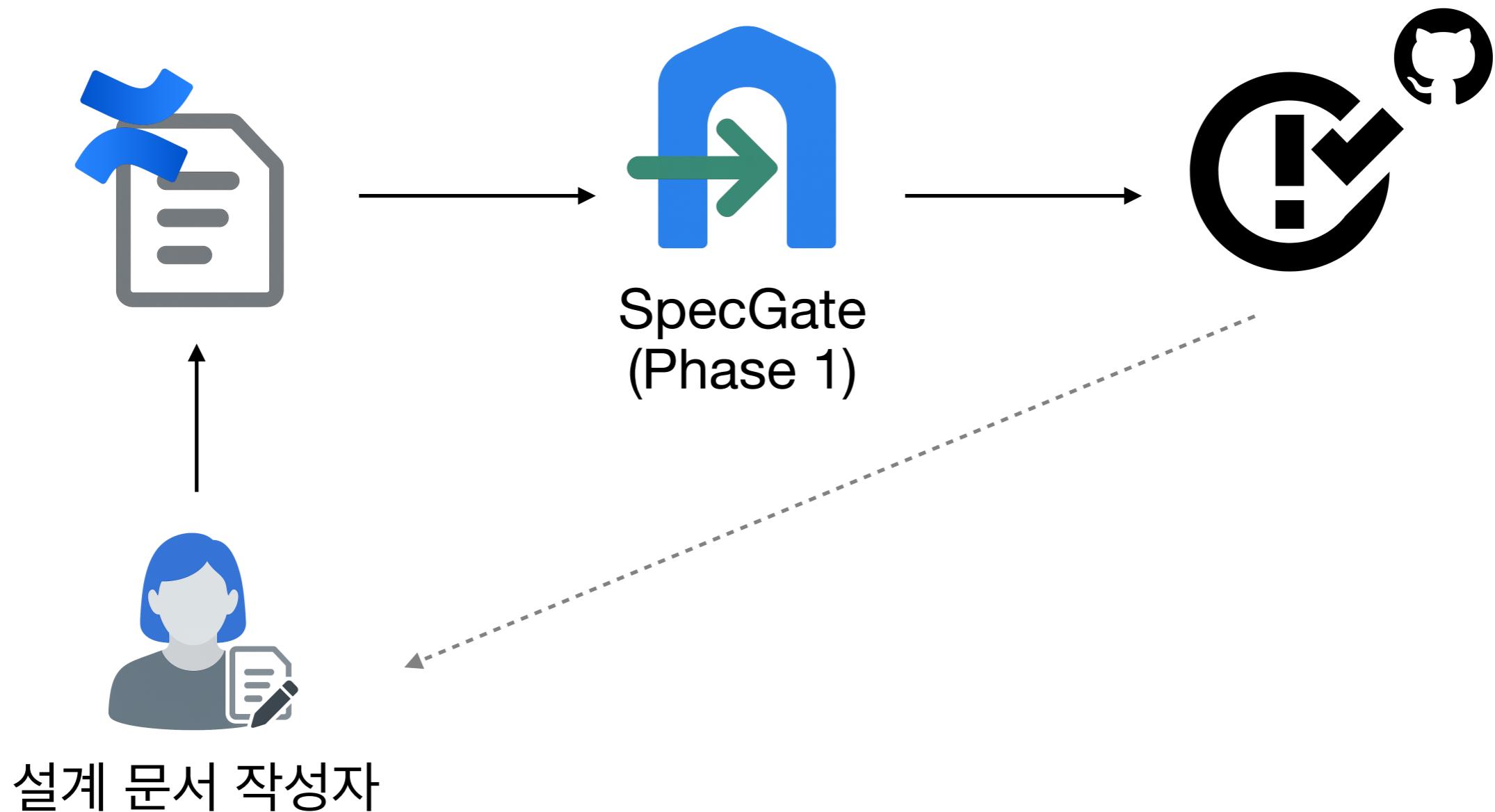
DesignRuleSpec 중간 표현 DSL 생성
Cursor IDE용 .mdc 를 파일 자동 생성
실시간 스니펫, 가이드, 위반 감지

3) 검증 확인

ArchUnit, Semgrep, OpenAPI Diff
Drift Score로 설계-코드 괴리 정도 측정
PR 차단 및 Drift Radar 시각적 피드백

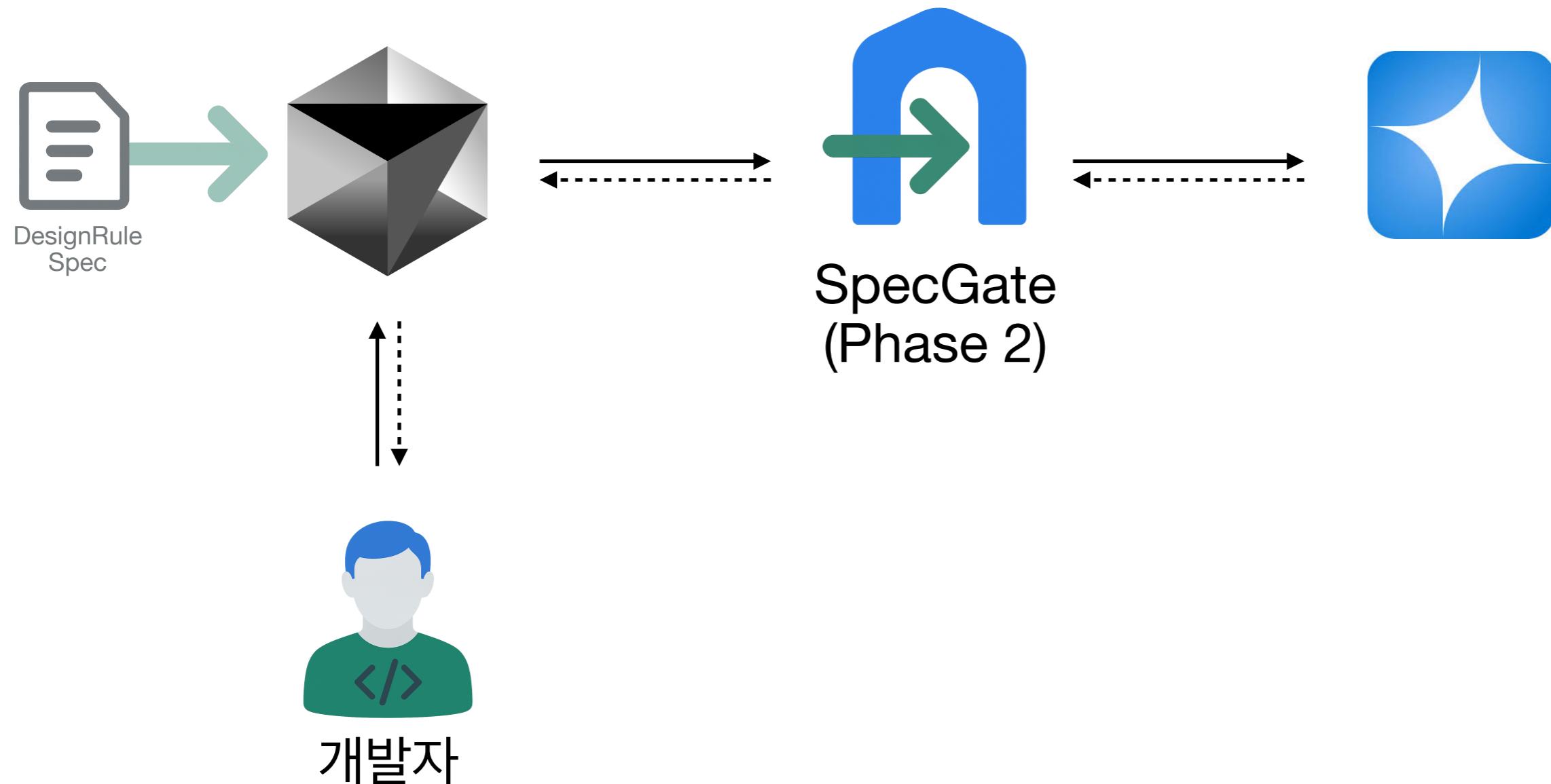
3-Phase 문의 작동 원리

Phase 1 : 문서 수집 및 정규화



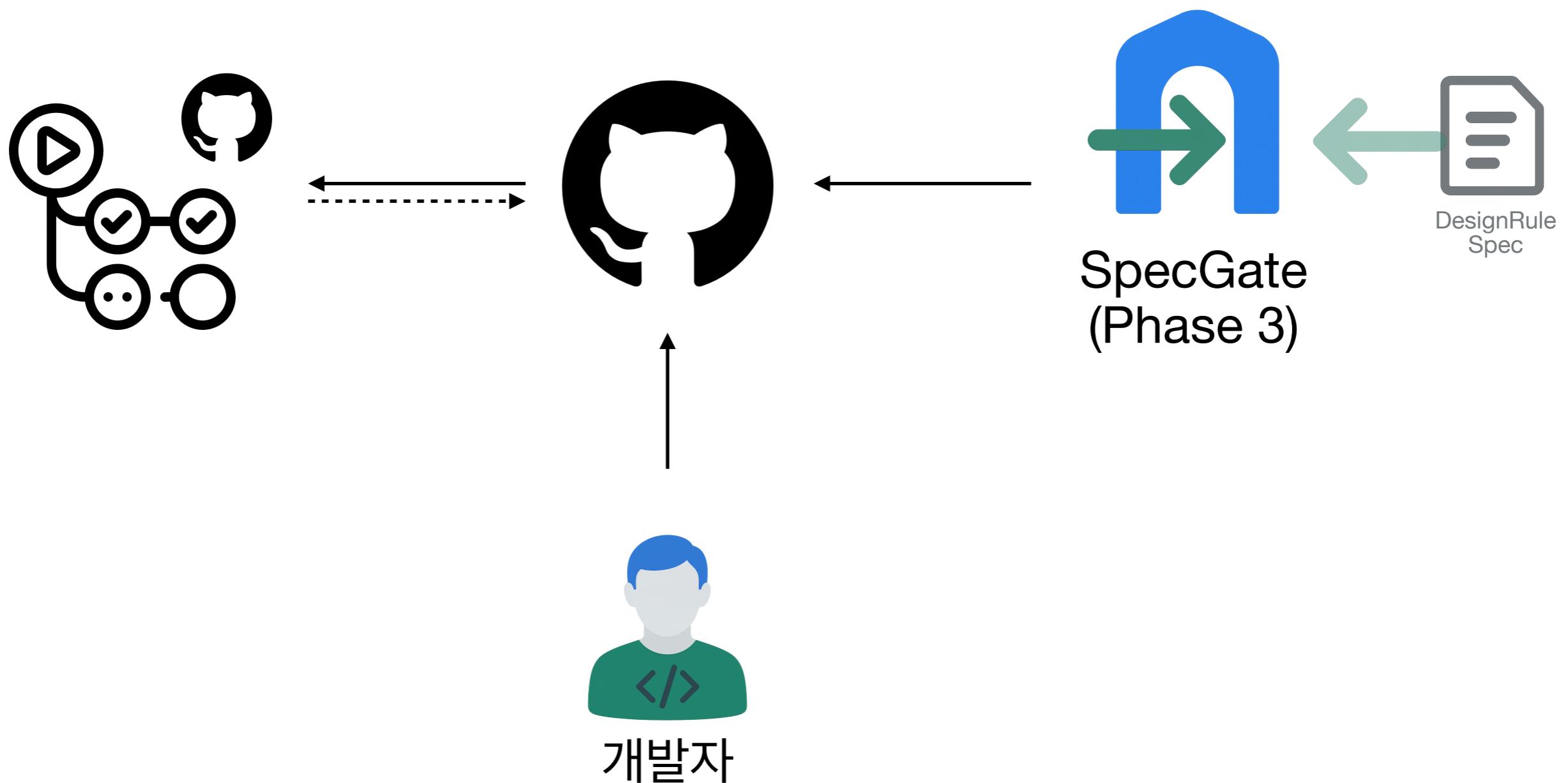
3-Phase 문의 작동 원리

Phase 2 : 규칙 추출 및 IDE 를 생성



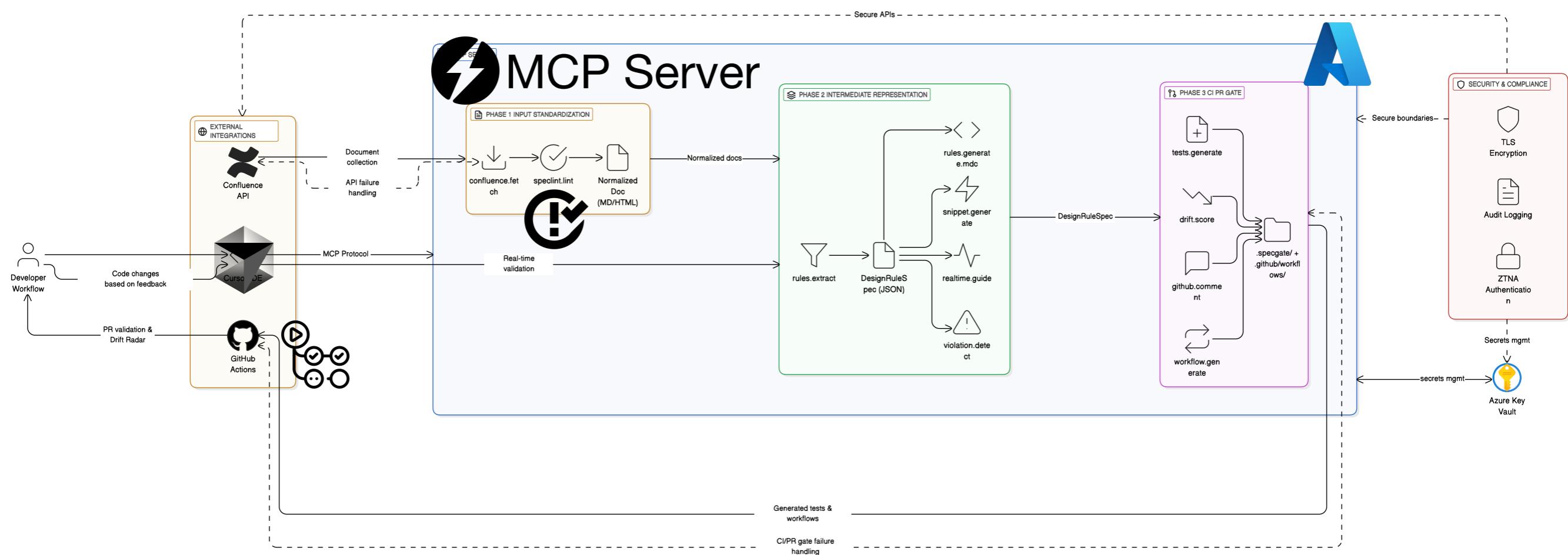
3-Phase 문의 작동 원리

Phase 3 : 자동 테스트 및 CI 검증



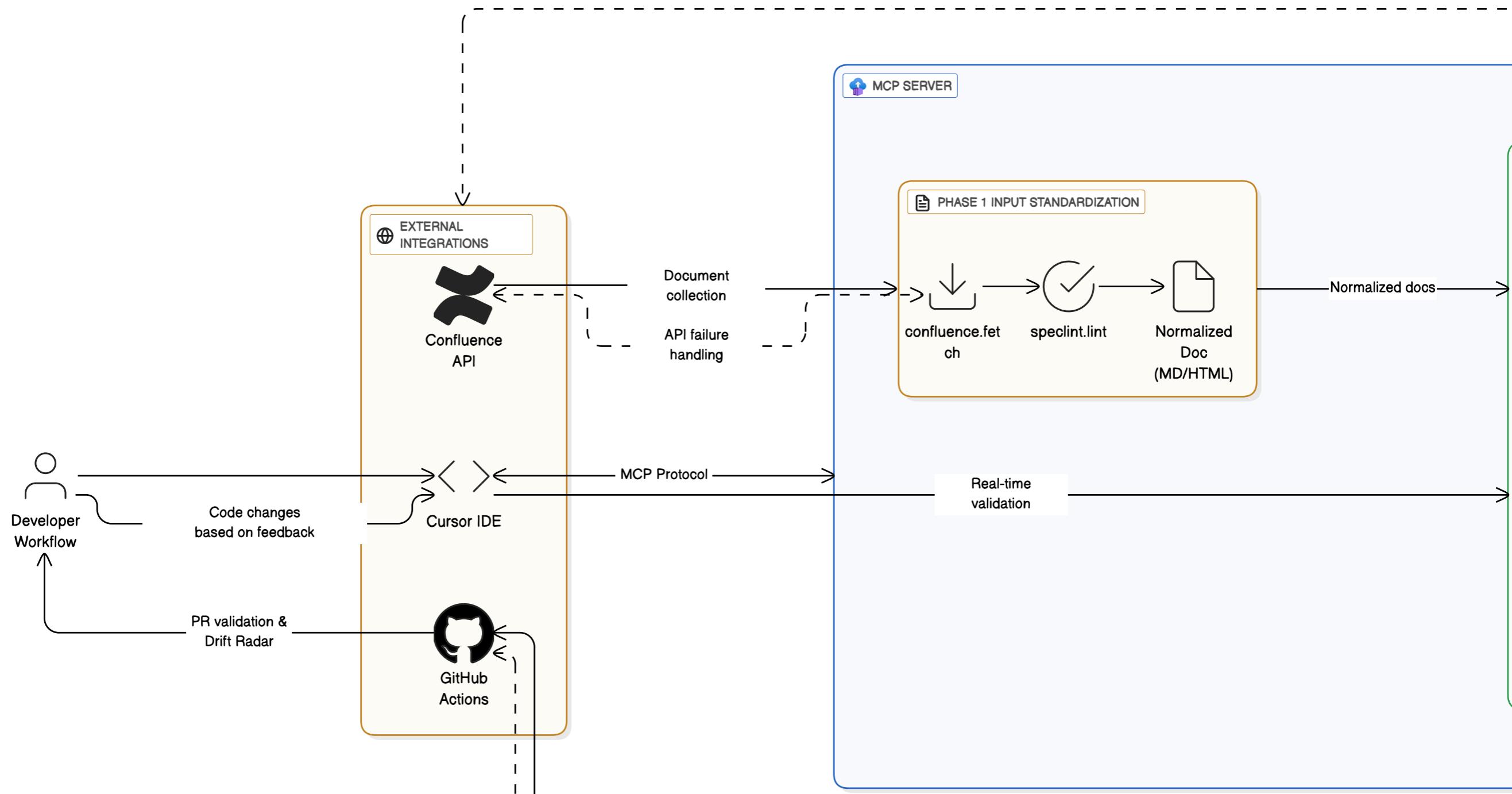
아키텍처

Tech Stack



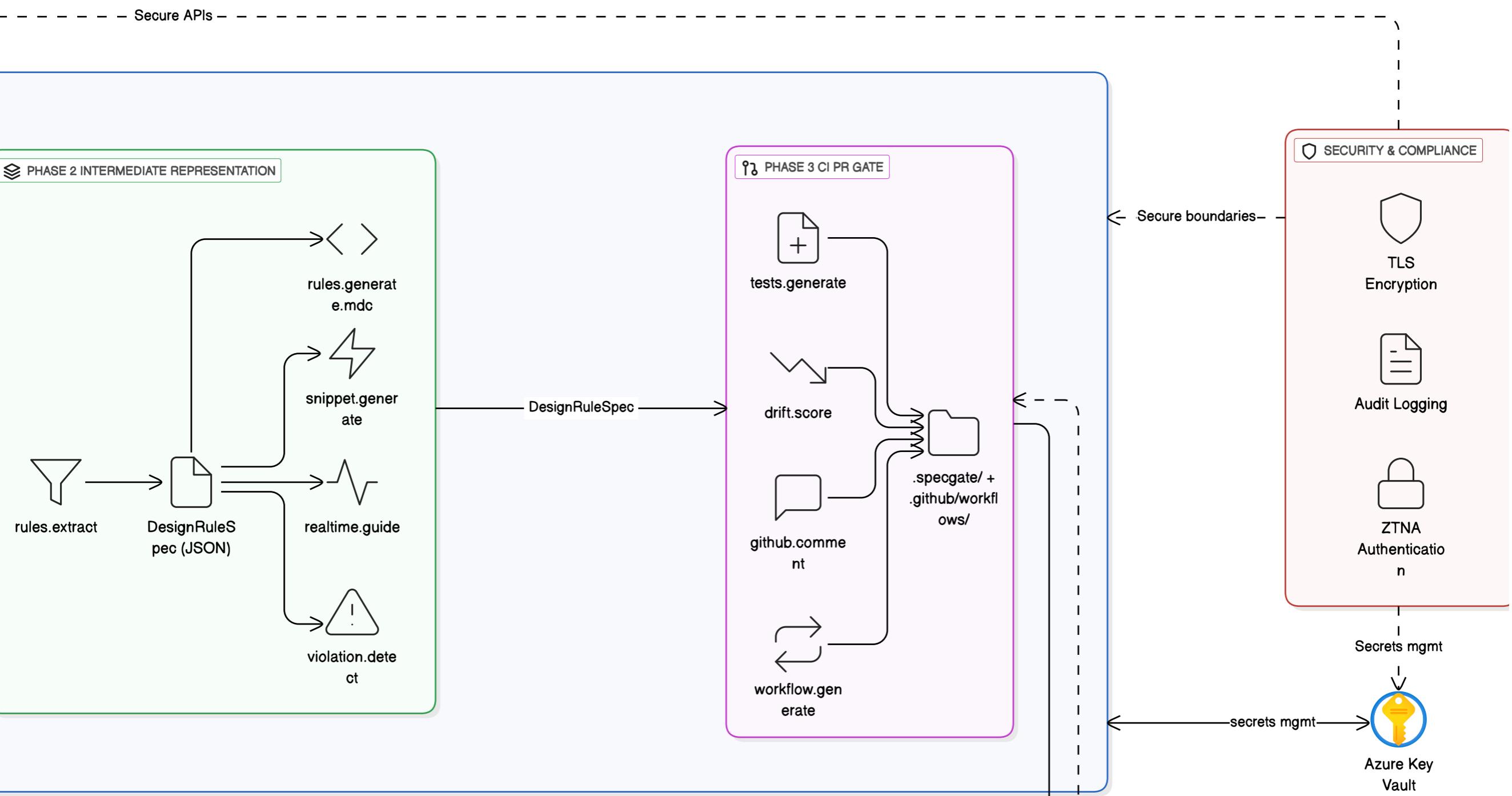
아키텍처

Phase 1



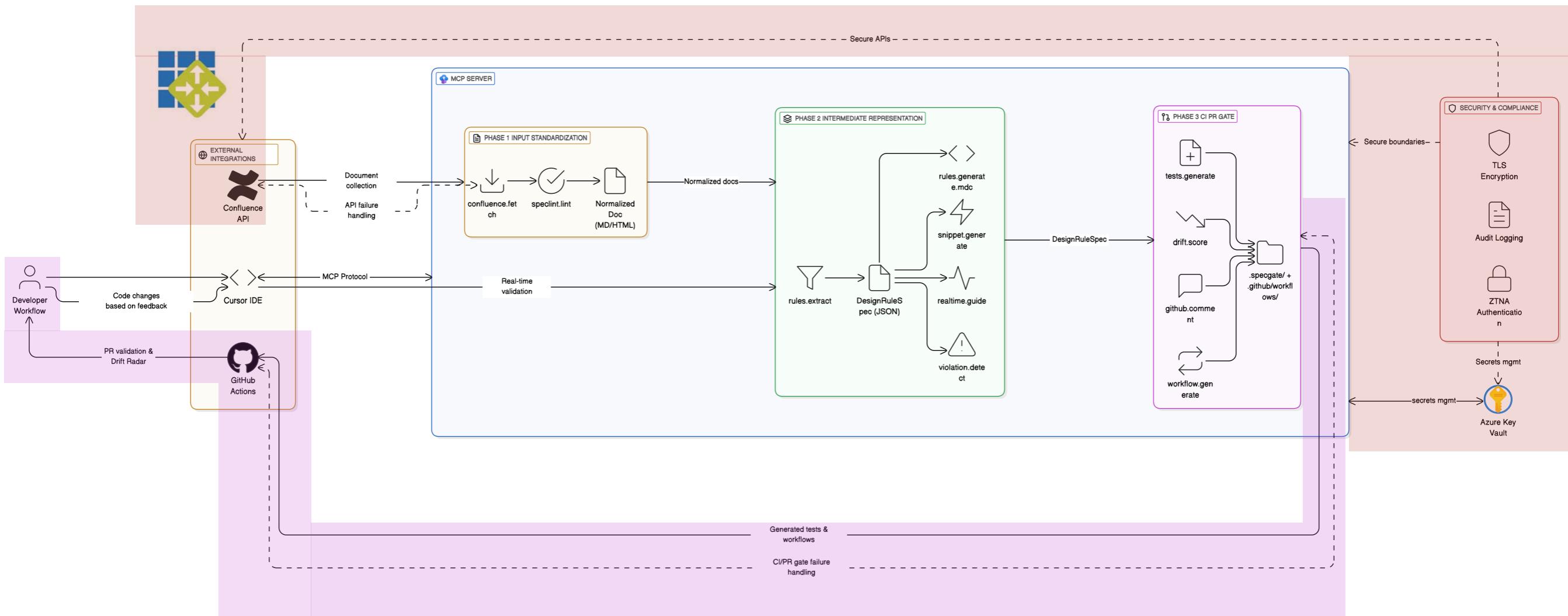
아키텍처

Phase 2 & Phase 3



아키텍처

Phase 3 & 보안



화면 Mockup

문서 정규화 단계

- SpecLint

[SpecGate] Confluence 문서 품질 개선 필요 #4

 Open

Edit

New issue

younyoungieo opened now

[SpecGate] Confluence 문서 품질 개선 필요

SpecGate 문서 품질 검사 결과

문서 품질 점수: 65/100 (개선 필요)

검사 대상 문서

- [API 설계 가이드](#) - 65점
- [아키텍처 설계서](#) - 70점
- [데이터베이스 설계서](#) - 60점

주요 문제점

문서	점수	문제점	개선 방안
API 설계 가이드	65점	RESTful 원칙 설명 부족	구체적인 예시 코드 추가
아키텍처 설계서	70점	계층 구조 다이어그램 누락	아키텍처 다이어그램 추가
데이터베이스 설계서	60점	ERD 불완전	완전한 ERD 작성

SpecGate 자동 수정 제안

- API 설계 가이드 개선**
 - RESTful 원칙 섹션 추가
 - 예시 코드 블록 삽입
 - HTTP 상태 코드 가이드 추가
- 아키텍처 설계서 개선**
 - Mermaid 다이어그램 추가
 - 계층별 책임 명시
 - 의존성 관계 표시
- 데이터베이스 설계서 개선**
 - 완전한 ERD 작성
 - 테이블 관계 명시
 - 인덱스 설계 가이드 추가

Assignees	
No one - Assign yourself	
Labels	
No labels	
Projects	
No projects	
Milestone	
No milestone	
Relationships	
None yet	
Development	
 Code with agent mode	
Create a branch	for this issue or link a pull request.
Notifications	Customize
 Unsubscribe	
You're receiving notifications because you're subscribed to this thread.	
Participants	
→ Transfer issue	
Duplicate issue	
Lock conversation	
Pin issue	
Delete issue	

화면 Mockup

개발 단계

- 실시간 개발 지원

커서 를 적용

스니펫 자동 완성

> Called SpecGate ✓

SpecGate MCP 서버가 연결되었습니다!
Confluence에서 5개 설계 문서를 수집했습니다
Phase 1 완료: 문서 정규화 및 품질 검사 (평균 85점)
Phase 2 완료: DesignRuleSpec 생성 (총 12개 규칙)
Phase 3 준비: 테스트 및 CI 검증 준비 완료
SpecGate가 준비되었습니다! 어떤 개발을 도와드릴까요?

UserController를 만들어야 해

UserController 개발을 시작하겠습니다!

적용된 룰:

- RULE-API-001 (MUST): RESTful 원칙 준수
- RULE-API-002 (MUST): 표준 응답 형식 사용
- RULE-ARCH-001 (MUST): Service 계층을 통한 호출

어떤 메서드부터 구현하시겠어요?

@RestController

> Called SpecGate ✓

@RestController 어노테이션을 추가했습니다!

J java

```
@RestController
@RequestMapping("/api/v1/users")
public class UserController {
    // SpecGate가 제안한 기본 구조
}
```

SpecGate 를 적용 확인:

- RULE-API-001 (MUST): RESTful 원칙 준수 - Controller 계층으로 올바르게 설정됨
- RULE-ARCH-001 (MUST): Service 계층을 통한 호출 - 준비됨

다음으로 어떤 메서드를 구현하시겠어요?

> 5 Files Undo All ⌘⌫ Keep All ⌘↩

@ Add Context 32%

Plan, search, build anything

∞ Agent ⌘I Auto ⌘

화면 Mockup

개발 단계

- 위반 감지 및 수정 가이드

자동 수정

SpecGate 를 적용 확인:

- RULE-API-001 (MUST): RESTful 원칙 준수 - GET 메서드 사용
- RULE-API-002 (MUST): 표준 응답 형식 사용 - ResponseEntity 사용
- RULE-ARCH-001 (MUST): Service 계층을 통한 호출 - userService 호출

SpecGate 실시간 가이드:

- RULE-API-001 (MUST): RESTful API는 리소스명을 동사로 사용하지 않고, 행위는 HTTP 메서드로 표현합니다.
- `getUserData()` 는 RESTful 원칙에 맞지 않습니다. `getUser()` 또는 `findUserById()` 를 권장합니다.

자동 수정해줘

규칙 위반 감지

> Called SpecGate ✓

SpecGate 자동 수정을 실행합니다!

J java

```
@RestController  
@RequestMapping("/api/v1/users")  
public class UserController {  
  
    @GetMapping("/{id}")  
    public ResponseEntity<User> getUser(@PathVariable Long id) {  
        // SpecGate 스나펫 제안: return userService.findById(id);  
        return userService.findById(id);  
    }  
}
```

자동 수정 완료:

- 메서드명: `getUserData()` → `getUser()`
- URI: `/users/{id}/data` → `/{id}`
- RULE-API-001 (MUST): RESTful 원칙 준수 완료

SpecGate 를 적용 확인:

- RULE-API-001 (MUST): RESTful 원칙 준수 - 완료
- RULE-API-002 (MUST): 표준 응답 형식 사용 - 완료
- RULE-ARCH-001 (MUST): Service 계층을 통한 호출 - 완료

다음으로 어떤 메서드를 구현하시겠어요?

> 5 Files

Undo All ⌘U Keep All ⌘C

@ Add Context

32%

Plan, search, build anything

∞ Agent ⌘I Auto ^



화면 Mockup

PR/CI 단계

- Drift Radar

[SpecGate] UserController RESTful API 구현 #3
younyoungieo wants to merge 1 commit into `main` from `test`

younyoungieo commented 1 minute ago

SpecGate Drift Radar

설계 준수도	점수	상태
전체 Drift Score	90/100	⚠️ PR 승인 가능 (개선 권장)
RULE-API-001 (RESTful 원칙)	100점	✅ 완료
RULE-API-002 (표준 응답 형식)	85점	⚠️ 개선 필요
RULE-ARCH-001 (Service 계층 호출)	100점	✅ 완료

GitHub Actions Status

- ✅ ArchUnit 테스트: 아키텍처 규칙 검증 완료
- ✅ Semgrep 스캔: 보안 취약점 없음
- ⚠️ OpenAPI Diff: API 일관성 부분 개선 필요

SpecGate 자동 코멘트

설계 준수도 90점! 대부분의 룰을 준수했지만, 일부 개선이 필요합니다.

개선 권장사항:

- ResponseEntity 사용 시 HTTP 상태 코드 명시 권장
- API 응답 형식 표준화 필요

변경사항:

- `getUserData()` → `getUser()` (RESTful 원칙 준수)
- URI: `/users/{id}/data` → `/{id}` (RESTful 원칙 준수)
- Service 계층을 통한 호출 구조 적용

PR 승인 가능하지만, 개선 후 승인을 권장합니다. ⚠️

변경사항

- UserController RESTful API 구현 완료
- SpecGate 를 적용: RULE-API-001, RULE-API-002, RULE-ARCH-001
- 자동 수정: `getUserData()` → `getUser()` (RESTful 원칙 준수)
- 개선 필요: API 응답 형식 표준화

Reviewers

Suggestions

Copilot

Request

Still in progress? [Convert to draft](#)

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Subscribe

You're not receiving notifications from this thread.

1 participant

Lock conversation

기대효과

개발자 경험

설계 문서를 일일이 찾아볼 필요 없이 실시간으로 올바른 코드 작성



실시간 가이드로 개발 속도 향상

- Cursor IDE에서 실시간 스니펫 제안
- 설계 규칙을 기억할 필요 없이 자동 가이드



설계 준수 코드 자동 작성 편의

- DesignRuleSpec 기반 자동 스니펫 생성
- 위반 코드 작성 시 즉시 알림 및 수정 가이드



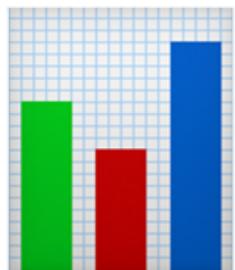
리뷰 효율성 증대

- Drift Radar로 위반 사항 시각화
- 구체적인 수정 방법과 예시 코드 제공

기대효과

품질 관리

설계와 코드의 일관성을 자동으로 보장하여 품질 관리



설계 준수도를 객관적 수치로 측정

- Drift Score로 설계-코드 괴리 정도 측정
- 시간에 따른 품질 변화 추적 가능



자동 검증으로 위반 사항 사전 차단

- PR 단계에서 자동 테스트 실행
- 임계치 초과 시 PR 자동 차단
- 프로덕션 배포 전 설계 위반 사전 차단

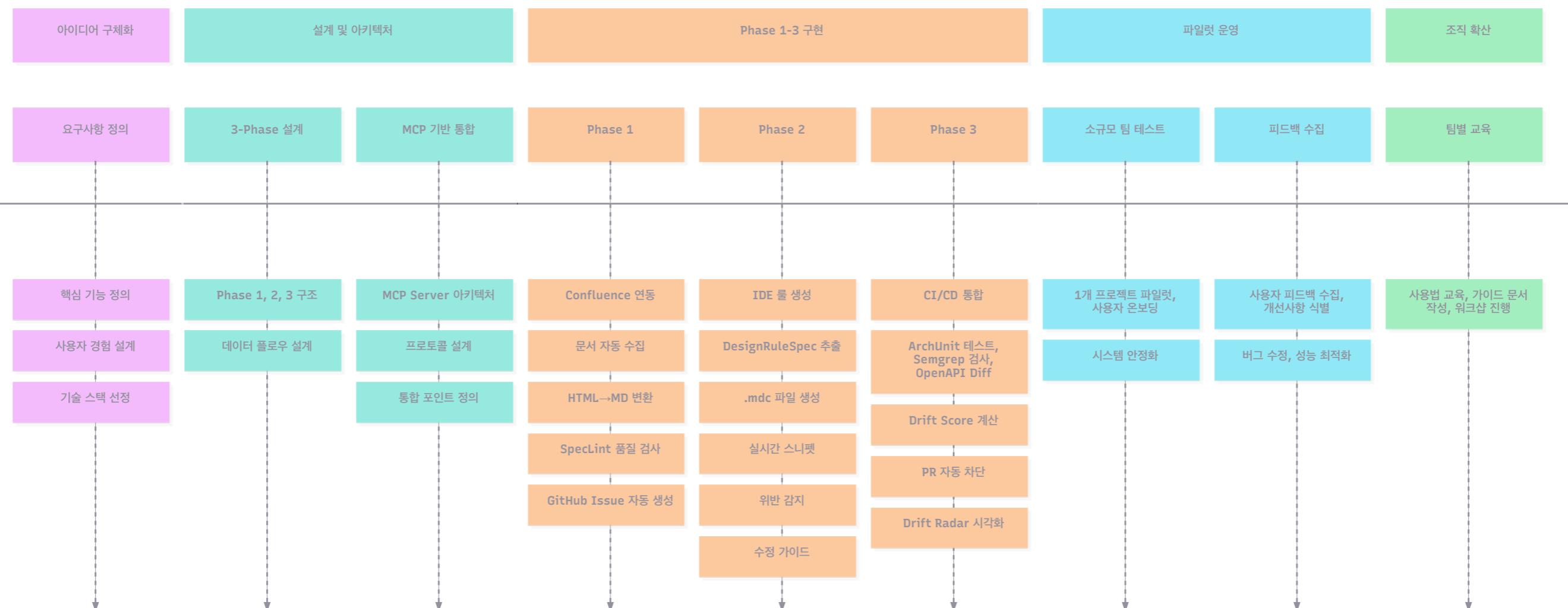


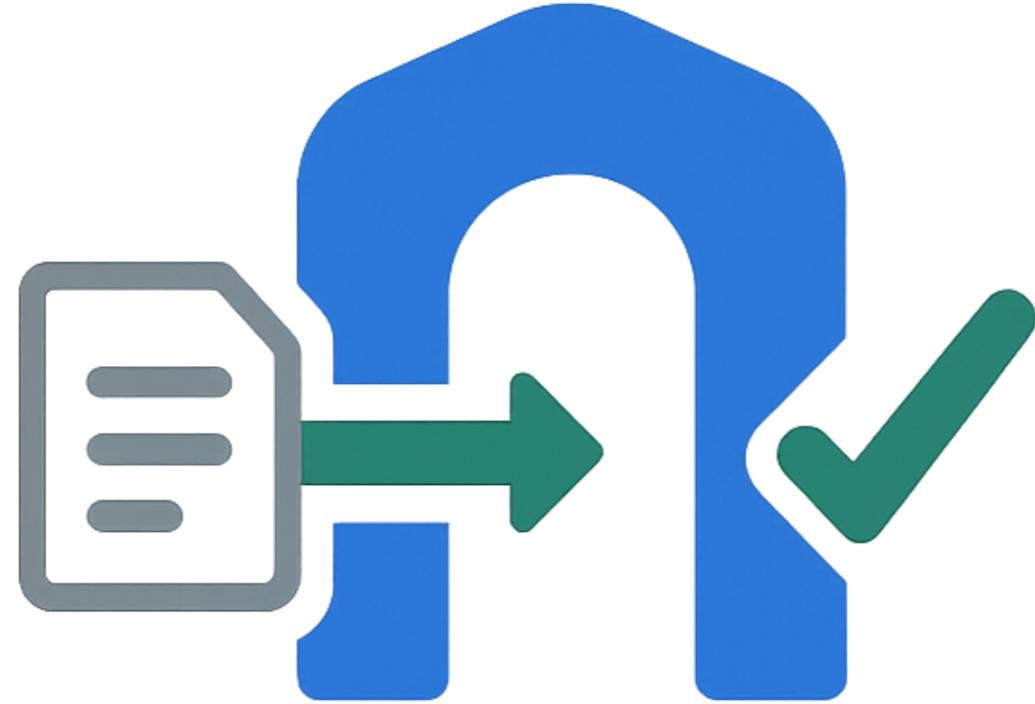
신규 투입자 온보딩 속도 향상

- 실시간 가이드로 프로젝트 설계 규칙 학습 가능
- 설계 문서를 찾아볼 필요 없이 IDE에서 바로 확인

로드맵

SpecGate 구현 로드맵





SpecGate

설계와 구현을 연결하는 문

Q&A

Q1: SpecGate 도입 시 기존 개발 프로세스를 바꿔야 하나요?

Q2: 다른 비슷한 도구들과의 차별점은 무엇인가요?

Q3: 사내망 보안과 배포는 어떻게 처리하나요?

Q4: Drift Score가 너무 엄격해서 개발 속도가 느려질 수 있나요?

Q5: Cursor 외에 다른 IDE나 버전 관리 시스템도 지원하나요?