

# High-Level Design (HLD)

## AI-Assisted Pull Request Review System (Human-in-the-Loop)

### 1. System Overview

The AI-Assisted Pull Request Review System augments human code reviews by providing automated, non-blocking feedback on pull requests. The system integrates with GitHub and focuses on code quality, design principles, maintainability, security patterns, and guideline adherence. The AI acts strictly as an advisory reviewer and never approves, rejects, or merges pull requests. Final decisions remain with senior human reviewers.

### 2. Actors & External Systems

**Developer:** Creates or updates pull requests and receives feedback.

**Senior Reviewer:** Reviews AI feedback and CI results and makes final decisions.

**GitHub:** Hosts repositories, emits webhook events, and enforces merge permissions.

**LLM Provider:** Generates review feedback as a stateless inference service.

### 3. High-Level Components

**Webhook Listener:** Receives and validates GitHub PR events.

**Orchestrator:** Applies rules and decides review scope and personas.

**Context Fetcher:** Collects minimal, relevant code and guideline context.

**AI Review Engine:** Generates structured, explainable feedback with confidence levels.

**Feedback Publisher:** Posts non-blocking AI comments back to GitHub.

### 4. Data Flow

1. Developer opens or updates a PR.
2. GitHub emits a webhook event.
3. Webhook Listener validates and forwards the event.
4. Orchestrator evaluates PR metadata.
5. Context Fetcher gathers minimal context.
6. AI Review Engine generates feedback.
7. Feedback is posted to GitHub.
8. Senior Reviewer makes the final decision.

### 5. Context Management Strategy

The system avoids full repository ingestion. Only PR diffs, modified files, limited related files, and repository guidelines are fetched. Dependency resolution and build validation are delegated to CI pipelines.

### 6. Human-in-the-Loop & Safety Design

AI feedback is clearly labeled and non-blocking. The AI has no merge or approval permissions. Human reviewers retain full authority, and AI failure does not block the PR workflow.

## 7. Non-Functional Requirements

**Scalability:** Supports multiple repositories and large PRs.

**Reliability:** PR workflow continues even if AI is unavailable.

**Cost Control:** Enforced token and context limits.

**Security:** Least-privilege access and no direct LLM-to-GitHub access.

## 8. Out of Scope

The system does not auto-merge or auto-approve PRs, run builds or tests, resolve dependencies, replace CI/CD pipelines, or learn autonomously from merges.

## 9. Design Philosophy

The system is built on the principle that AI should amplify human judgment, not replace it. Strict boundaries, minimal context usage, and human-in-the-loop control ensure production readiness.