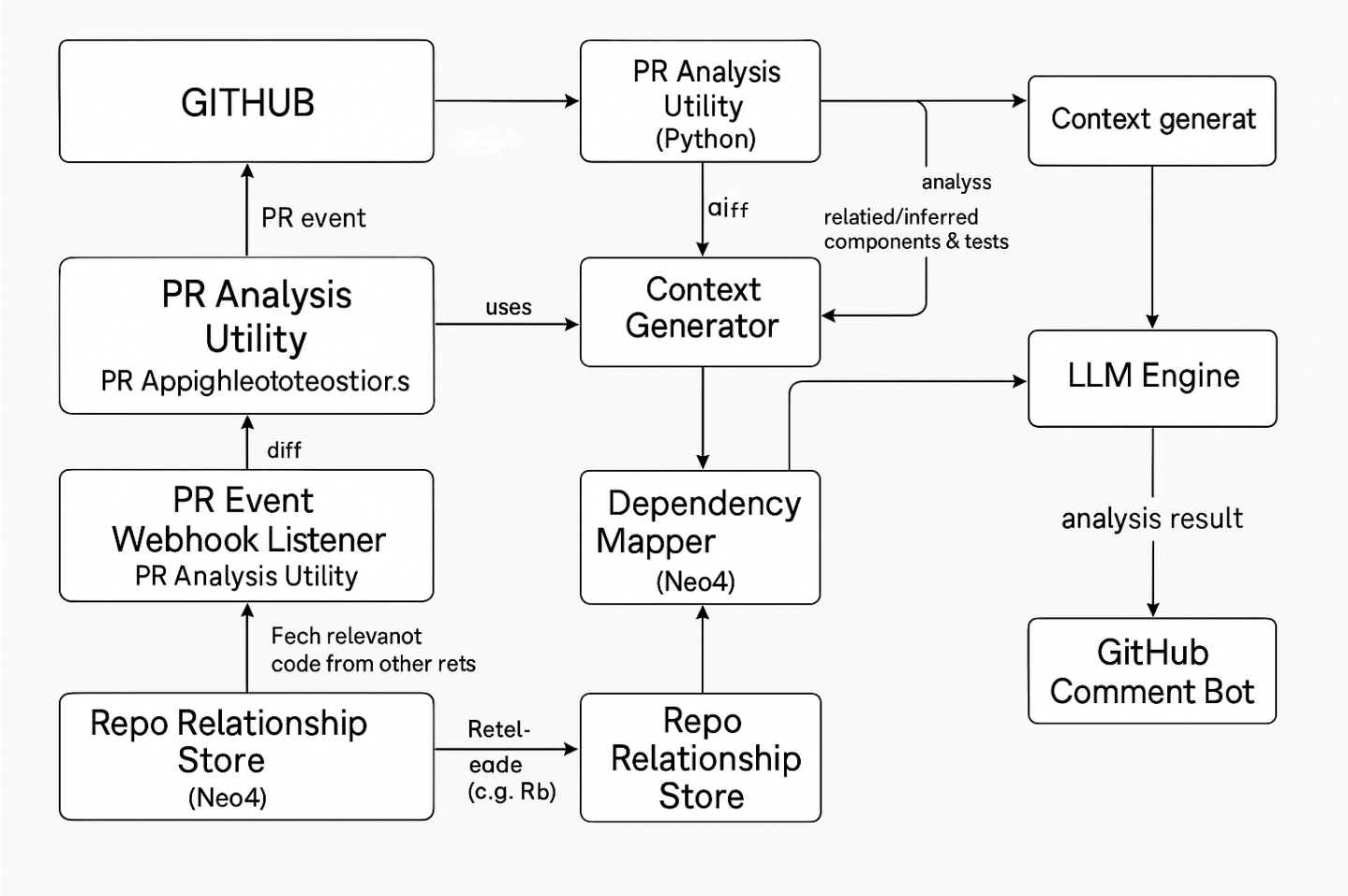
Automated PR Analysis Utility for Java Microservices

This document explains the architecture and each component of a Python-based automated utility that analyzes Pull Requests (PRs) in Java Spring Boot microservice repositories stored on GitHub. The tool supports single and cross-repository analysis and provides AI-powered suggestions using LLMs.



GitHub

Source control system triggering the flow.

Input: Pull Request (PR) creation/update.

Output: Webhook event sent to the automation utility.

PR Event Webhook Listener

Receives GitHub webhook and extracts PR metadata.

Input: PR webhook payload.

Output: Triggers PR analysis with metadata and diffs.

Repo Relationship Store

Stores cross-repo dependencies.

Input: Service definitions, config files, custom admin input.

Output: List of related repositories and components.

PR Analysis Utility

Central orchestrator for processing the PR and determining impacts.

Input: PR metadata, diffs, related repo info.

Output: Change summary for further analysis.

Dependency Mapper

Maps component interactions within and across services.

Input: Code diffs, static analysis results.

Output: Impacted modules and test files.

Context Generator

Builds structured prompt for LLM.

Input: Diffs, test names, cross-repo impacts.

Output: Prompt text for LLM.

LLM Engine

Performs reasoning and generates analysis.

Input: Context prompt.

Output: Summary and recommendations.

GitHub Comment Bot

Posts the LLM analysis to GitHub PR.

Input: LLM output.

Output: GitHub comment with detailed suggestions.