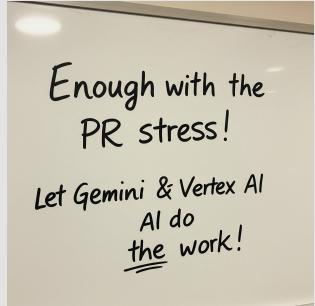


EVIDEN



Gemini

Dość stresu z PR-ami –
niech Gemini & Vertex AI zrobią robotę!

EVIDEN



Piotr Hyży
Lead Technical Architect
Eviden

EVIDEN

- 01 Intro & Agenda
- 02 Context & Key Concepts
- 03 Simple PR Summary
- 04 PR Summary + Review (detailed feedback)
- 05 Review + Publish (GitHub-ready response)
- 06 Repo Guidelines + Full Flow (compliance check)
- 07 What More? (more advanced expansions)
- 08 GitHub Actions Gemini PR (ready to use solution)
- 09 Summary & Q&A

Agenda

EVIDEN

Context and Key Concepts

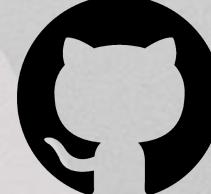
EVIDEN

Gemini

Gemini
Code Assist



Gemini Code
Assist
For GitHub


GitHub

Context and Key Concepts

EVIDEN



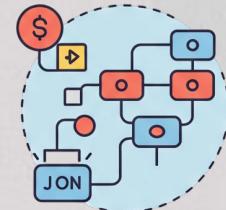
Prompt Engineering

Context and Key Concepts

EVIDEN

Structured Output

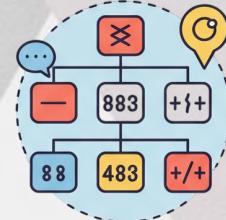
Context and Key Concepts



JSON



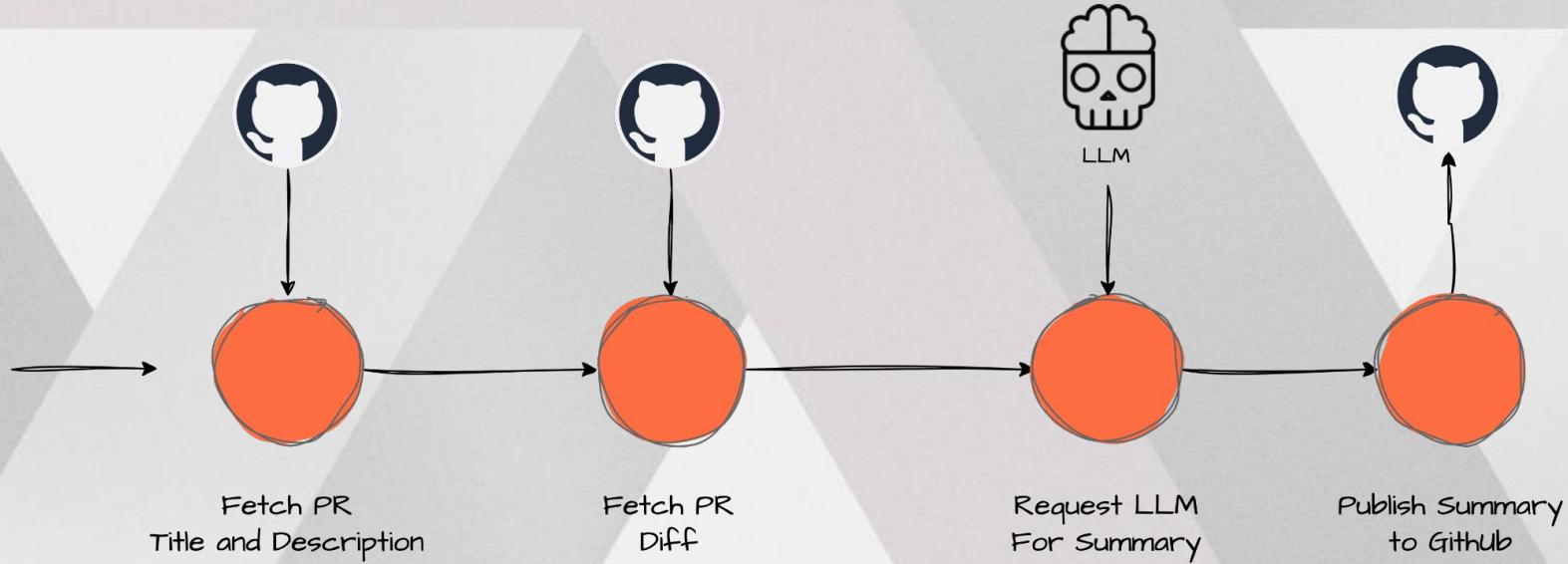
Markdown



EVIDEN

Simple PR Summary

EVIDEN



Workflow

Simple PR Summary

EVIDEN

Issue Comment

- Appears in the **Pull Request's** main “Conversation” thread, rather than on specific lines of code.
- Serves as **general feedback** or discussion, not tied to a particular commit or code snippet.
- Typically created with methods like `create_comment` or `create_issue_comment` in **PyGitHub**.

Review Comment

- Specifically attached to a file and line in the **pull request diff** (“Files changed”).
- Allows in-context discussion of code changes, making it easier to **pinpoint** the exact lines being referenced.
- Typically created with **create_review_comment** or included in a larger **create_review**.

GitHub Comment Type

Simple PR Summary

EVIDEN

- Fetch Data From Github
- Request LLM For Summary
- Publish Response on GitHub PR
- **Outcome:** AI provides a concise summary of changes, helping reviewers understand the scope and intent of the PR in seconds.

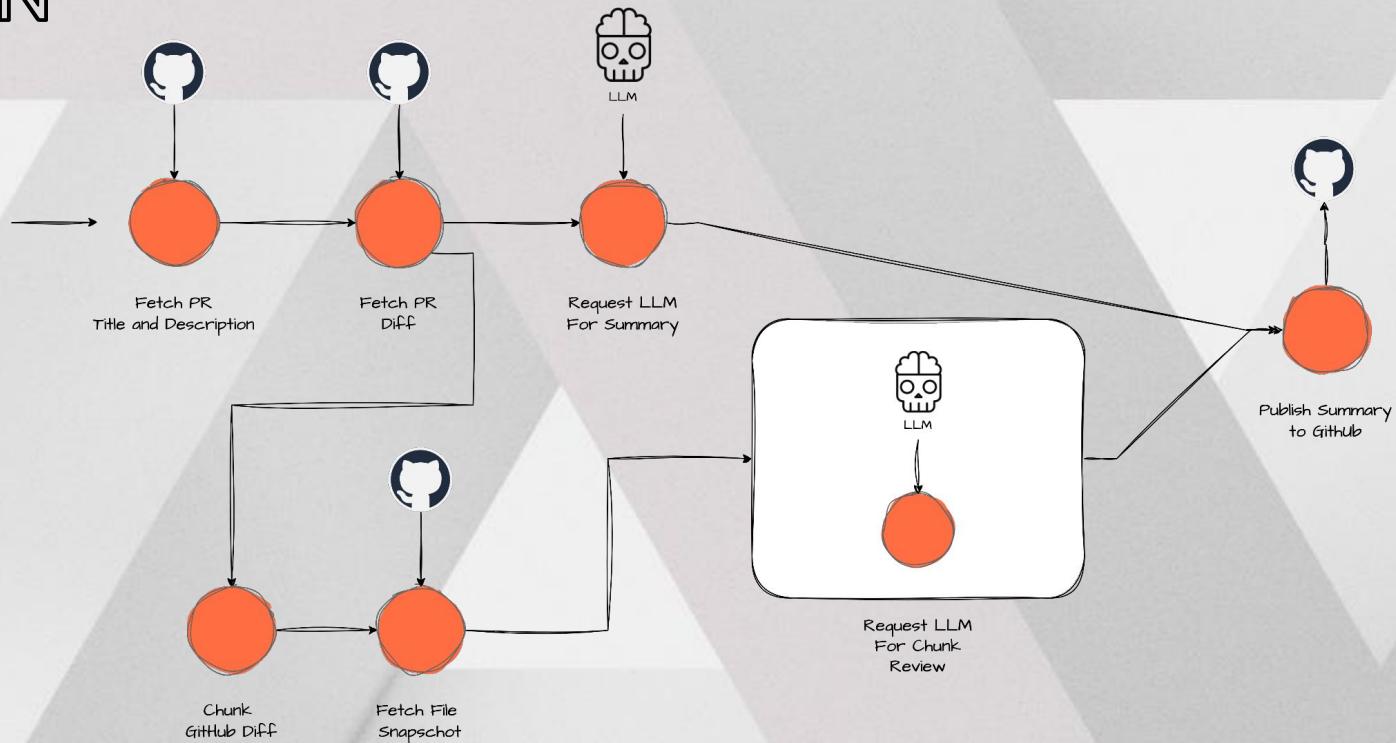
Key Elements

Simple PR Summary

EVIDEN

PR Summary + Review

EVIDEN



Workflow

PR Summary + Review

EVIDEN

Step-by-Step Prompting

Instead of asking the AI to "review the code," ask it to first list the changes, understand the changes, then check for style issues, and finally suggest improvements.

```
You are an expert software engineer tasked with reviewing a pull request. Your role is to identify potential issues, suggest improvements, and provide overall feedback. You must follow a step-by-step process: 1. Understand the Changes: Carefully analyze the proposed changes. 2. Consider the Context: Use the original file as a reference. 3. Focus on Code Quality: Evaluate the changes based on correctness and readability. * **Correctness:** ... * **Readability:** ...
```

Context Enrichment

When summarizing the PR, if review comments are available, they are appended to the prompt to give the model additional context for its summary.

```
if review_comments:
    prompt_text += """
Review Comments:
Here are some review comments and the proposed changes to the PR:
"""
for i, comment in enumerate(review_comments):
    prompt_text += f"""
Comment {i+1}:
File: {comment.path}
Lines: {comment.start_line}-{comment.line}
Comment: {comment.body}
"""


```

Prompt Techniques

PR Summary + Review

EVIDEN

Provide only the output no other comments or introductions

Pull Request Summary:

...

Review Summary:

...

Overall Quality Assessment

...

Merge Assessment

...

Structured Output with Sectioned Formatting

The final merged review prompt explicitly instructs the model to output sections (e.g., Pull Request Summary, Review Summary, Overall Quality Assessment, Merge Assessment) to ensure clarity and organization.

Prompt Techniques

PR Summary + Review

EVIDEN

- Chunk and Structure PR Diff For Better Processing
- Enrich Chunk with Original File Snapshot
- Request LLM For File Level Review
- **Outcome:** AI-generated insights now highlight specific coding issues, propose fixes, and ensure alignment with project guidelines.

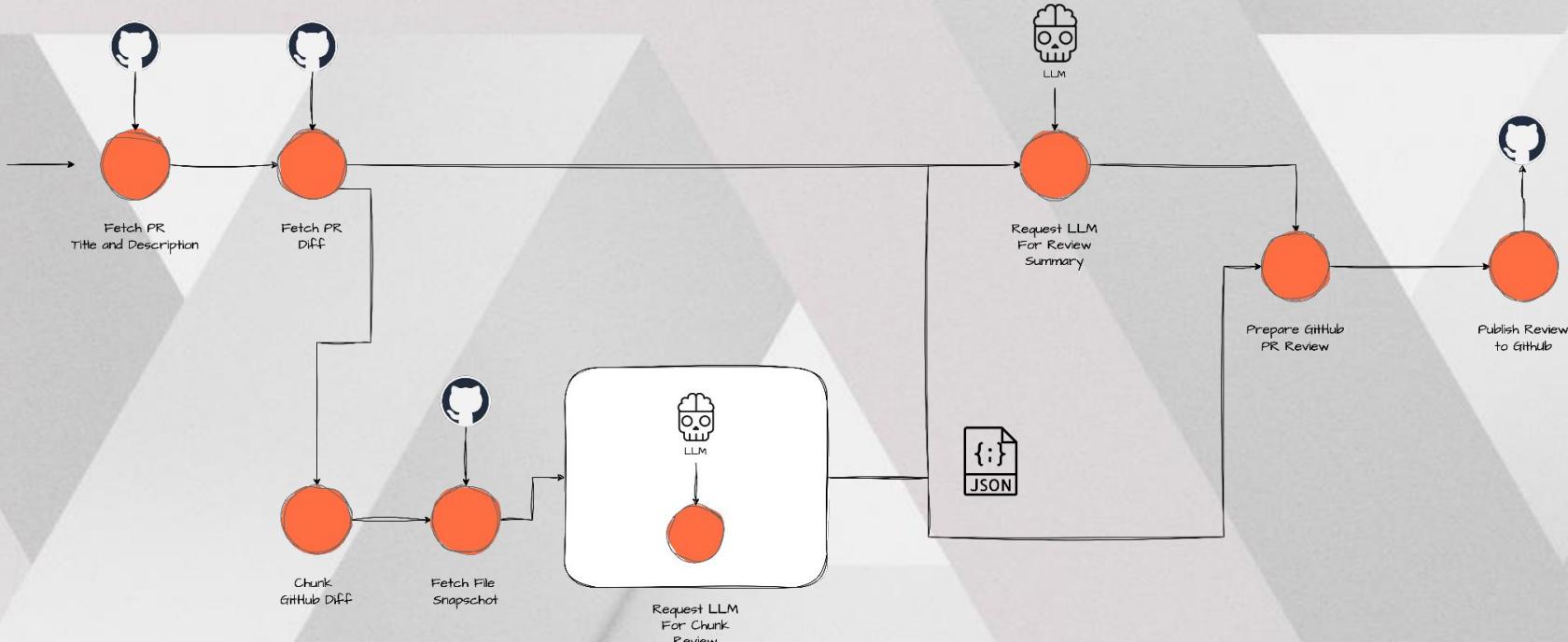
Key Elements

PR Summary + Review

EVIDEN

Review + Publish

EVIDEN



Workflow

Review + Publish

EVIDEN

Structured, Schema-Enforced Output (JSON Schema)

The prompt embeds a JSON schema to force the model's output to conform to a pre-defined structure. This is used for generating file reviews that are machine-readable and consistent.

```
**JSON Output Schema:**  
```json  
{json.dumps(json_output_schema)}```
```

## Prompt Techniques

## Code Suggestion Format

The prompt instructs the model to include code suggestions using GitHub's suggestion format. This ensures that when improvements are needed, the model returns suggestions in a format directly usable by developers.

```
Include code suggestions in the review body when it can be helpful for the developer using the GitHub suggestion format:
```suggestion  
# Your improved code here
```

Review + Publish

EVIDEN

- Request Structured Review Comments in JSON
- Combine PR Summary and Chunk Reviews, into Holistic Summary Review
- Convert AI Feedback into GitHub API-Ready Format
- Display Fully Automated PR Review within GitHub UI, mark AI BOT as Reviewer
- **Outcome:** AI-generated feedback is automatically posted to GitHub as actionable comments, enabling quick review and fixes directly within the PR.

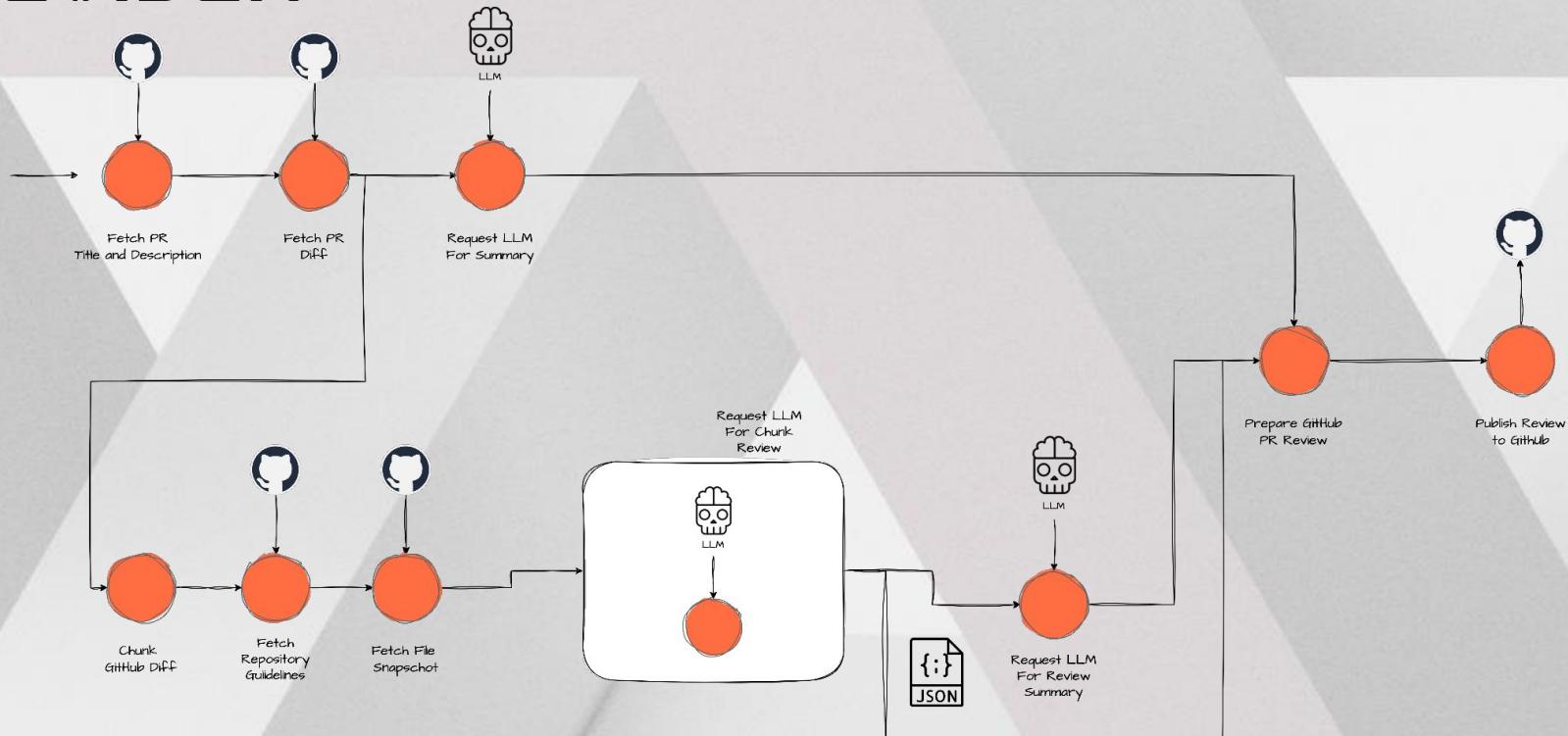
Key Elements

Review + Publish

EVIDEN

Repository Guideline

EVIDEN



Workflow

Repo Guidelines

EVIDEN

- Bot Detects and Reads Repo-Specific Guidelines
- AI Adjusts Feedback Based on Repo-Specific Rules.
- Elastic way of Providing Repository Level Policy
- **Outcome:** AI reviews code against repository-specific guidelines, ensuring consistency and enforcing coding standards automatically.

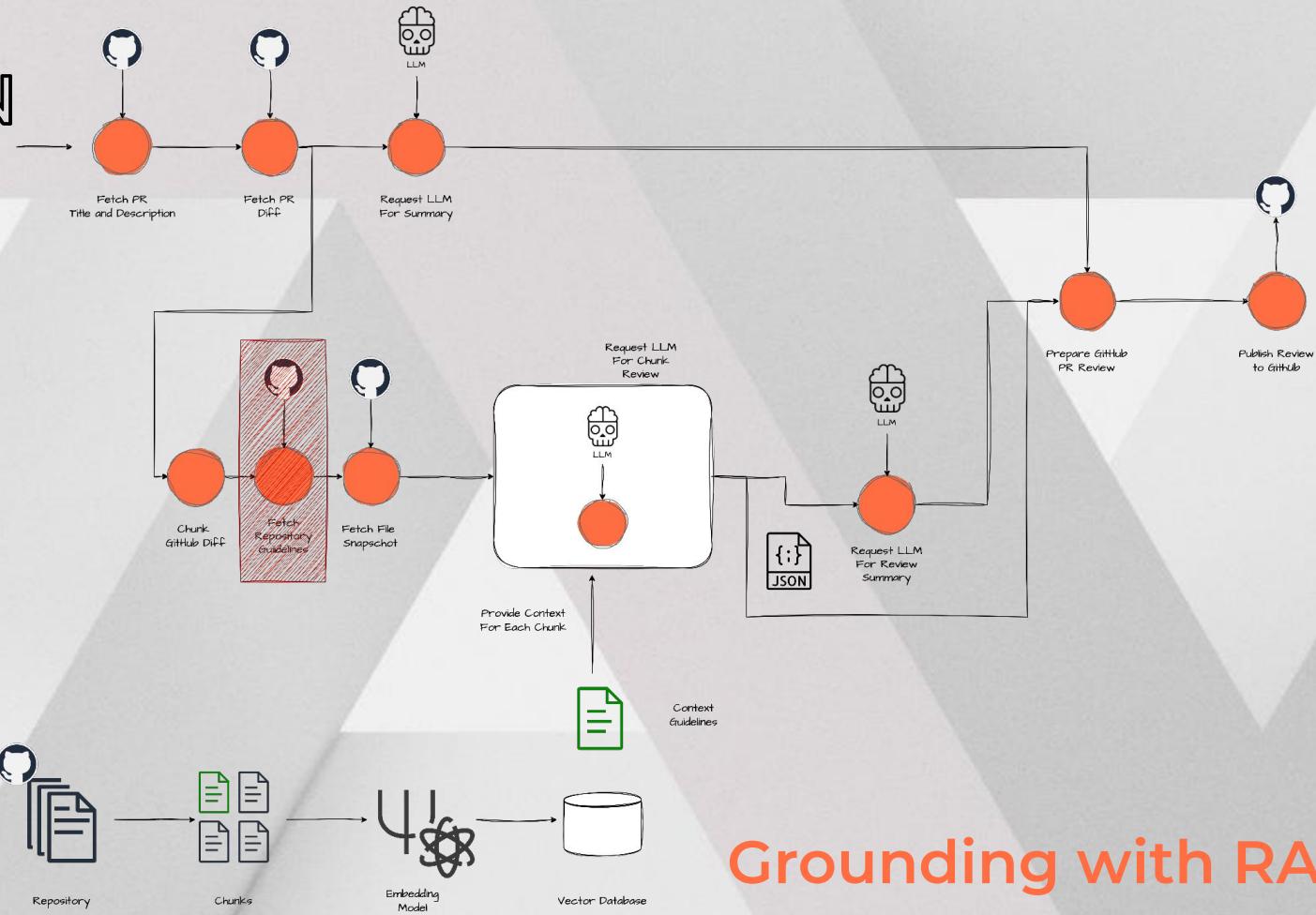
Key Elements

Repo Guidelines

EVIDEN

Grounding with RAG

EVIDEN



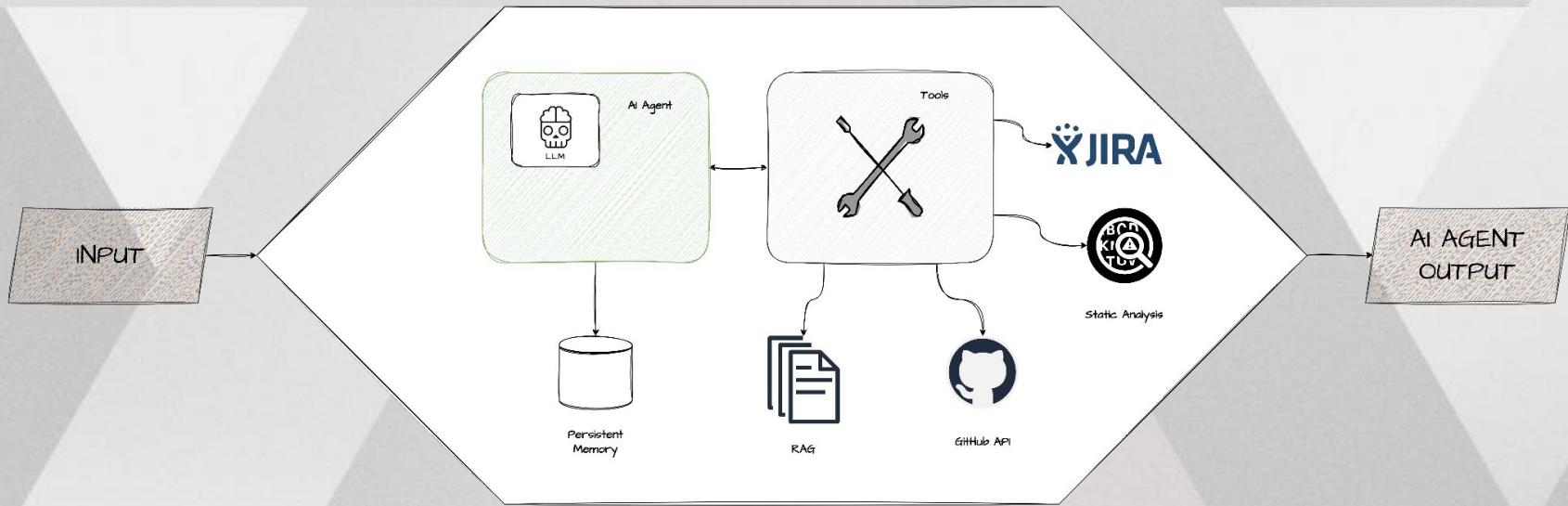
Workflow

Grounding with RAG

EVIDEN

Agentic Approach

EVIDEN

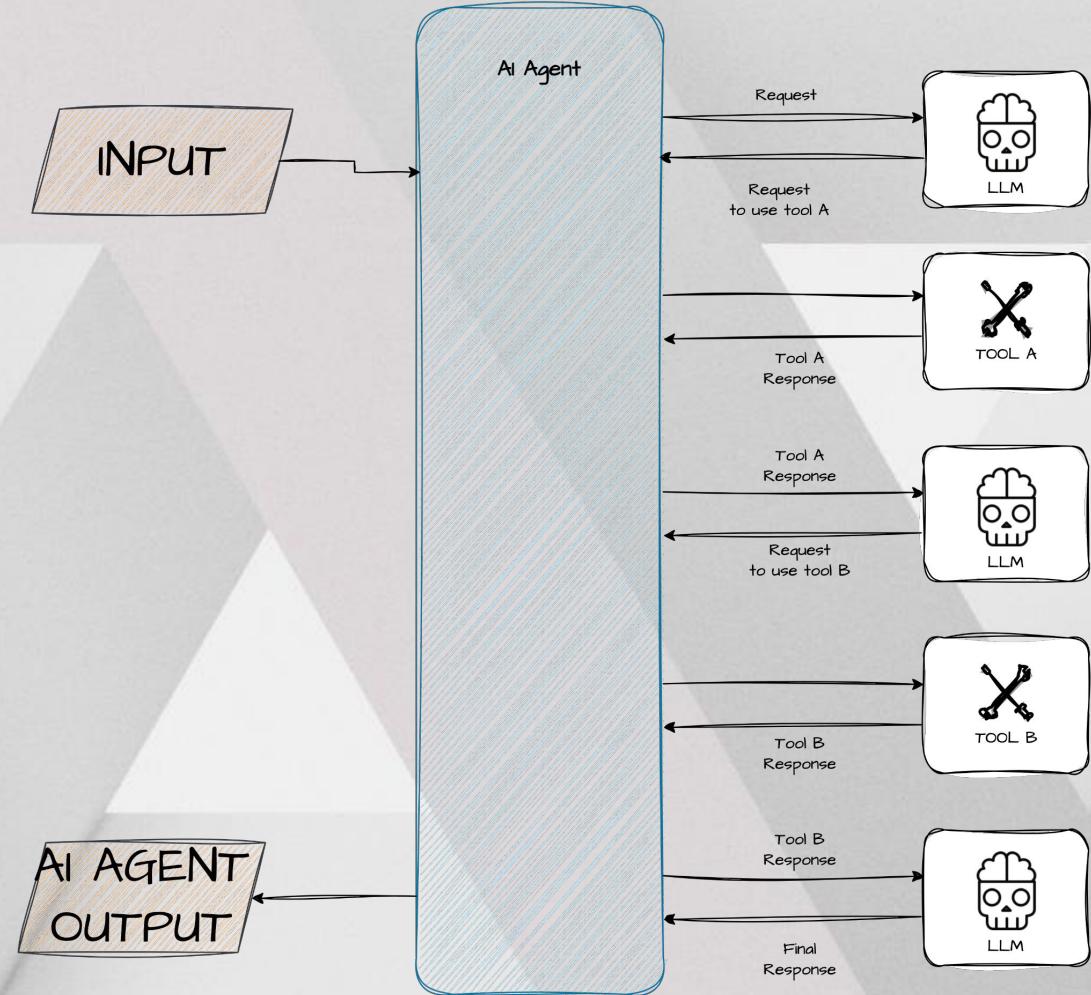


Workflow

Agentic Approach

EVIDEN

Agentic Approach



EVIDEN

<https://github.com/GoogleCloudPlatform/agent-starter-pack>

The screenshot shows the GitHub repository page for 'Agent Starter Pack'. At the top, there are links for 'README', 'Code of conduct', and 'Apache-2.0 license'. Below the header, there's a section titled 'Agent Starter Pack' with a rocket icon. It includes links for 'pypi v0.1.4', '1-Minute Overview', and 'Documentation'. The main content area describes the repository as a collection of production-ready Generative AI Agent templates built for Google Cloud, accelerating development by providing a holistic solution for challenges like Deployment & Operations, Evaluation, Customization, and Observability. A table highlights four key features: 'Launch' (Pre-built agent templates), 'Experiment' (Vertex AI evaluation), 'Deploy' (Production-ready infra with monitoring, observability, and CI/CD), and 'Customize' (Extend and customize templates). Below this, a section titled 'Get Started in 1 Minute' provides a command to build an AI agent:

```
# Create and activate a Python virtual environment
python -m venv venv && source venv/bin/activate

# Install the agent starter pack
pip install agent-starter-pack

# Create a new agent project
agent-starter-pack create my-awesome-agent
```

Agentic Approach

EVIDEN

Gemini Code Assist For GitHub

EVIDEN

Gemini Code Assist

Search

Home Guides Resources

Filter

Discover

Overview

Privacy notices

Supported languages, IDEs, and interfaces

How Gemini Code Assist works

How Gemini Code Assist Standard and Enterprise use your data

Responsible AI

Get started

Set up Gemini Code Assist for individuals

Standard and Enterprise

Code in IDEs

Code with Gemini Code Assist for individuals

Standard and Enterprise

Review code in GitHub

Review GitHub code with Gemini Code Assist

Customize Gemini Code Assist behavior in GitHub

Configure

Standard and Enterprise

Troubleshoot

Troubleshoot Gemini Code Assist

Troubleshoot access to Gemini Code Assist

Was this helpful?

Review GitHub code using Gemini Code Assist

Preview

This product or feature is in preview. Products and features that are in preview are available "as is".

Gemini Code Assist for GitHub brings the power of Gemini to the pull request process by acting as a code reviewer. Gemini Code Assist speeds up and increases the quality of code using a Gemini-powered agent that automatically summarizes pull requests and provides in-depth code reviews. You can invoke Gemini Code Assist at any stage of the pull request to review the code.

You can interact with Gemini in the pull request comments directly by:

- Asking clarifying questions on the review that Gemini creates.
- Prompting Gemini by adding the `/gemini` tag to your comments to ask questions in the context of the pull request.

Gemini will automatically retrieve helpful information from the repository and pull request to perform its tasks.

This document is intended for developers of all skill levels. It assumes that you have a working knowledge of GitHub.

Before you begin

To test the steps in this document, make sure you do one of the following:

- Use your own GitHub repository
- Create a fork of [our sample repository](#)

Install Gemini Code Assist for GitHub

Gemini Code Assist for GitHub

EVIDEN

The screenshot shows the GitHub Marketplace page for the "gemini-code-assist" app. At the top, there's a navigation bar with icons for Marketplace, search, and user profile. Below the header, the app's logo (a blue square with a white arrow) and name "gemini-code-assist" are displayed, along with an "App" badge.

The main content area has tabs for "README" and "Transparency". The "README" tab is active, showing the following sections:

- Accelerate code reviews and improve code quality**: A brief description of the app's purpose, mentioning AI-powered code reviews, automatic pull request summaries, and code suggestions.
- Pull request summary and feedback**: Details how the app provides an initial review within five minutes and adds a reviewer to the pull request.
- Ready-to-commit code suggestions**: Information about how the app provides code changes to implement suggestions directly from the Pull Request.
- Invoking Gemini Code Assist**: Instructions on how to request assistance from Gemini via issue comments or commands.

On the right side of the page, there's an "About" section with the app's name, developer information (Google), and install count (7,506). It also includes a "Verified" badge, which indicates GitHub has verified the publisher's identity, ownership of their domain, and compliance with other requirements. Below this are sections for "Tags" (ai-assisted), "Pricing" (Free plan available), and "Resources" (Support, Documentation, Privacy Policy, Terms of Service, Report abuse).

Feature	Command	Description
Code Review	/gemini review	Performs a code review for the current pull request in its current state.
PR Summary	/gemini summary	Provides a summary of the current pull request in its current state.

Gemini Code Assist for GitHub

EVIDEN

Summary

EVIDEN

Value of Automated PR Review

- **Increased Efficiency and Speed:** Faster code scanning and quicker review turnaround.
- **Consistent Code Quality:** Enforces coding standards for maintainable systems.
- **Reduced Reviewer Fatigue:** Less mental load for reviewers.
- **Early Detection of Bugs and Vulnerabilities:** Identifies issues earlier for faster fix.

Pros and Cons

Summary

EVIDEN

Value of Automated PR Review

- **Increased Efficiency and Speed:** Faster code scanning and quicker review turnaround.
- **Consistent Code Quality:** Enforces coding standards for maintainable systems.
- **Reduced Reviewer Fatigue:** Less mental load for reviewers.
- **Early Detection of Bugs and Vulnerabilities:** Identifies issues earlier for faster fix.

Risks of Dehumanizing PR Review

- **Loss of Context and Nuance:** Misses subtle issues requiring human understanding.
- **Discouraged Collaboration and Learning:** Less developer interaction and knowledge sharing.
- **Erosion of Team Cohesion:** Impersonal reviews damage team morale.
- **False Sense of Security:** Complacency leads to missed critical issues.

Pros and Cons

Summary

EVIDEN

Key Takeaways - Revolutionizing Code Review with AI

- **The Importance of Automated Code Review:** Faster feedback, improved code quality, developer productivity boost.
- **Leveraging Vertex AI and Gemini:** Powerful AI engine enables advanced code understanding and generation.
- **Progression from Summaries to AI Code Review:** Evolving from basic summarization to complete AI review.
- **Structured Output & Pipeline Integration:** Automated reports, seamless integration into existing workflows possible.
- **GitHub Action for Real-World Implementation:** Easily automate code reviews in your workflow in GitHub.
- **Significant Time Savings Possible:** AI can handle 50% review comments = 50% time saved!

Key Takeaways

Summary

EVIDEN

Conclusion and Next Steps

- **AI Augments, Doesn't Replace, Human Reviewers:** Collaboration key; AI handles grunt work, humans provide nuance.
- **AI Integration is Achievable:** Gemini Code Assist lowers the barrier to adoption of AI.
- **Vertex AI Offers Customization and Scalability:** Allows you to tailor AI solutions to specific project needs.
- **AI Boosts Productivity & Code Quality:** Thoughtful integration significantly improves developer efficiency.
- **Try Vertex AI SDK and Agent Starter Pack:** Experiment with code diffs and AI suggestions directly.
- **Pilot AI-Powered PR Reviews:** Start with summaries, documentation changes, or non-critical tasks.
- **Explore Resources & Further Reading:** Links to GitHub Action and Vertex AI documentation are available.

Next Steps

Summary

EVIDEN

Questions & Answers