interlock - Project Proposal

> Interlock is a Jira-native AI companion designed to govern and improve agent-assisted ticket resolution. Its scope is limited to supporting reasoning, documentation, and coordination within Jira tickets. It does not automate implementation or enforce workflows.

Problem and Motivation

Engineering teams increasingly rely on Jira as the system of record for tracking work, yet much of the reasoning involved in resolving complex tickets happens outside the ticket itself. Engineers often consult documentation, search for similar tickets, review code, or use AI assistants privately to reason about solutions. These activities are rarely captured in a structured or persistent way inside the ticket.

A growing part of this problem is ungoverned AI usage. When each engineer interacts with a private AI agent, the reasoning, insights, and decisions produced during those sessions are siloed, ephemeral, and lost after the session ends. Other engineers cannot learn from them, reviewers lack visibility, and future agent interactions are unaware of past resolutions.

As a result, Jira tickets frequently record outcomes without context. Decisions are difficult to audit, similar problems are repeatedly analyzed from scratch, and tickets provide limited long-term value as learning artifacts.

Interlock addresses this gap by focusing on agent session governing and shared reasoning at the ticket level.

Proposed Solution and Core Idea

Interlock is a Jira extension that embeds an AI companion directly into the lifecycle of a Jira ticket. Instead of interacting with an agent in private sessions, engineers interact with the agent through Jira ticket comments.

The agent acts as a step-by-step companion throughout ticket resolution. Its role is to gather relevant context, summarize information, structure discussions, and help translate decisions into actionable steps. The agent does not design

solutions, make architectural decisions, or implement changes. Human engineers remain fully responsible for judgment and execution.

All agent interaction is visible, persistent, and scoped to the ticket. Each ticket becomes a governed agent session that accumulates reasoning and context over time.

Users and Usage Flow

The primary users of Interlock are software engineers who are assigned Jira tickets that involve analysis, coordination, or architectural impact. Secondary users include technical leads, reviewers, and engineering managers who benefit from improved visibility into how tickets are resolved.

A typical usage flow includes:

- An automated overview scan of the ticket, retrieving similar past tickets, documentation, and related context.
- Generation of a concise problem summary and multiple possible resolution strategies.
- A focused deep dive into the strategy selected by the assignee, including dependencies, risks, and potential cross-team impact.
- Breakdown of the chosen strategy into tasks, subtasks, and acceptance criteria within the ticket.
- Continuous accompaniment during execution, with progress and updates documented through comments.

All interaction occurs through Jira comments, either internal or public, according to the assignee’s choice.

Alternatives and Design Rationale

Alternative approaches include manual documentation, private use of general-purpose AI assistants, or AI systems that attempt to generate solutions or code directly. Manual approaches are inconsistent and time-consuming, while private AI usage leads to siloed and non-reusable reasoning. Solution-authoring AI systems assume a level of design reliability that is often unrealistic.

Interlock takes a different approach. It deliberately avoids acting as a designer or authority and instead focuses on governing how assistance is applied. By

embedding the agent inside the ticket and keeping all interaction visible, Interlock improves transparency, learning, and trust without replacing human decision making.

Dependencies and Implementation Scope

Interlock depends on Jira as the execution environment and system of record. It integrates with an organizational knowledge hub, such as Pipehub, to retrieve documentation, historical tickets, and related context. It also relies on a language model provider for constrained summarization and reasoning.

The system does not require integration with code repositories, deployment pipelines, or production systems. Its scope is limited to analysis, documentation, and guidance within Jira.

Expected Impact and Future Direction

Interlock improves the quality and transparency of ticket resolution by turning Jira tickets into durable records of reasoning and decisions. It reduces duplicated analysis, supports cross-team collaboration, and enables both engineers and future agent executions to learn from past work.

Future extensions could include improved analysis of resolved tickets, refinement of strategy suggestions based on historical outcomes, and deeper integration with organizational knowledge systems. These extensions build on the same core principle of governed, ticket-level agent interaction.