ArchSense — Project Proposal

1. Project Idea and Motivation

Modern engineering teams rely heavily on Jira to track complex work such as architectural refactors, system extractions, and cross-service changes. While Jira issues are effective at describing tasks and requirements, they often fail to capture the broader architectural context in which the work takes place. In particular, issues frequently omit a clear description of the existing system architecture, the components and dependencies affected by the change, the expected future architecture, and the risks or tradeoffs involved.

At the same time, architectural knowledge is typically distributed across multiple sources such as Confluence pages, Notion documents, historical discussions, and informal team knowledge. As a result, engineers are often required to manually reconstruct a mental model of the system before they can reason effectively about a proposed change.

ArchSense is motivated by this gap. The project aims to transform a Jira issue from a task-oriented description into a structured architectural understanding. By doing so, it enables engineers to reason about system changes earlier in the development process and with greater confidence.

The project explicitly focuses on architectural sense-making and decision support, rather than automation or code generation.

---

2. Presumed Solution (High-Level)

ArchSense is a context-aware analytical assistant that ingests a Jira issue and produces structured technical artifacts describing the current system state, the expected future state, and the architectural impact of the proposed change along with possible resolution strategies.

The system combines three controlled inputs. The first is structured issue metadata and descriptions retrieved from Jira. The second is organizational knowledge retrieved through an enterprise knowledge hub, such as Pipehub. The third is large language model based reasoning that is explicitly constrained to analysis, synthesis, and explanation.

The output of the system is a read-only technical brief. This brief is intended to support architectural discussions, design reviews, and onboarding activities,

rather than to execute changes or modify systems automatically.

3. Problem Being Solved

The core problem addressed by ArchSense is the lack of explicit architectural understanding during the early stages of engineering work. Jira issues often provide insufficient architectural context, while system knowledge is scattered, partially outdated, or difficult to navigate. As a result, engineers frequently reason about architectural impact and dependencies too late in the process.

Existing AI tools tend to focus on task breakdown, text generation, or implementation assistance, rather than on building a coherent structural understanding of the system. This situation leads to misaligned designs, unnecessary rework, inefficient reviews, and increased onboarding costs.

ArchSense reframes the problem by emphasizing the importance of architectural understanding before implementation begins. Instead of immediately asking how to implement a change, teams are encouraged to first understand which system is being changed and how.

4. Expected User Persona

The primary users of ArchSense are software engineers working on medium to large codebases. These engineers are typically assigned Jira issues that involve non-trivial architectural changes and require a solid understanding of existing systems. They participate in design discussions and reviews and benefit from clear architectural context early in the process.

Secondary users include architecture and platform engineers, technical leads, and engineering managers. These users are primarily consumers of the generated artifacts for review and alignment purposes, rather than direct operators of the system.

The system assumes that users are technically proficient, but not necessarily domain experts in every subsystem involved.

5. Assumptions

The design of ArchSense is based on several key assumptions. Jira serves as the system of record for engineering work. Organizational knowledge already exists, even if it is fragmented or inconsistently maintained. The system

operates strictly in read-only mode, and its outputs are intended for human consumption rather than machine execution. Evaluation of the system focuses on clarity, correctness, and usefulness, rather than on automation or operational impact.

These assumptions intentionally constrain the scope of the project in order to ensure feasibility and clear evaluation within a workshop setting.

---

6. Alternative Approaches

Several alternative approaches could be used to address similar problems. Manual documentation practices are common, but they are time-consuming, inconsistent, and difficult to keep up to date. Codebase analysis tools provide insights into implementation details, but they do not capture intent or planned future changes. General-purpose AI assistants can generate text, but they often lack structured architectural grounding. Knowledge search tools retrieve relevant documents, but they do not synthesize architectural understanding.

ArchSense differentiates itself by bridging issue intent with architectural reasoning, rather than attempting to replace documentation systems or implementation tools.

---

7. Expected Components

ArchSense is composed of modular components that can be evaluated independently. These include a Jira issue ingestion component that reads structured metadata and descriptions, an issue analyzer that classifies issue types and extracts technical intent, and a context builder that retrieves and consolidates relevant organizational knowledge through a two-way interaction with the knowledge hub.

Additional components include an architecture synthesizer that constructs representations of the current and future system architecture, a solution generator that produces resolution strategies along with their tradeoffs, and an evaluation layer that supports explainability and structured assessment of the output.

---

8. Main Features and User Flow

The main features of ArchSense include reconstruction of the current system architecture, interpretation of the expected future architecture, identification of

affected components and dependencies, explicit articulation of assumptions and constraints, and the presentation of multiple resolution strategies with associated risks and tradeoffs. All outputs are delivered in the form of a structured technical brief.

A typical user flow begins when an engineer is assigned a Jira issue. ArchSense analyzes the issue and its context, retrieves relevant organizational knowledge, synthesizes an architectural understanding, and generates structured artifacts that can be reviewed and discussed by the team.

9. External Dependencies

ArchSense depends on Jira for issue metadata and descriptions, an enterprise knowledge hub such as Pipehub for documentation retrieval, and a large language model provider such as OpenAI or Claude. No direct integration with code repositories or deployment systems is required.

10. Milestones

The project is planned across several milestones. The first milestone focuses on defining scope, assumptions, and evaluation criteria. The second milestone covers implementation of issue ingestion and analysis. The third milestone addresses context building and knowledge retrieval integration. The fourth milestone focuses on architecture synthesis and structured output generation. The final milestone involves evaluation and refinement based on representative example issues.

11. Timeline (High-Level)

The project timeline is organized across four weeks. The first week is dedicated to problem definition and design finalization. The second week focuses on implementing the core processing pipeline. The third week covers knowledge integration and architectural synthesis. The fourth week is dedicated to evaluation, iteration, and documentation.

12. Expected Outcome

By the end of the workshop, ArchSense is expected to demonstrate how a Jira issue can be transformed into a clear and structured architectural problem

statement. The resulting artifacts should enable more informed reasoning, discussion, and design decisions before implementation begins.

TL;DR

ArchSense is a scoped, read-only analytical system that transforms Jira issues into structured architectural understanding by combining issue metadata, organizational knowledge, and controlled language model reasoning. The project emphasizes clarity, evaluability, and architectural insight rather than automation or code generation.