# Module Specification & Integration Proposal

**Objective:** This document serves as the technical specification for a new, standalone module. Its purpose is to provide our intelligent review system, Guardian, with the necessary details to perform a "friendly forensic" analysis, assessing the module's architecture and determining its potential for safe integration with a System Core.

Please provide clear and concise answers about your module's design. This questionnaire is designed to be completed by a project lead, a developer, or a sufficiently advanced AI agent.

### **Section 1: Module Identity**

1.1 Module Name:

A clear, descriptive name for the proposed module.

(e.g., Real-time Analytics Engine, User Profile Gamification, AI-Powered Content Summarizer)

1.2 Version / Family (Optional):

If applicable, provide a version number or associate this module with a family of tools.

(e.g., v2.1.0, or Pulse::RealtimeHeatmap)

1.3 Primary Goal:

In one or two sentences, what is the core purpose of this module? What problem does it solve or what value does it add?

(e.g., To process and visualize user engagement data to identify trends. To increase user retention by adding a badge and level-up system to profiles.)

### **Section 2: Technical Specification**

2.1 Module Type:

Classify your module by selecting the best fit from the options below. This helps Guardian understand its fundamental role.

(Options: UI Feature, Background Agent, API Wrapper, Logic Block, System Control Point)

2.2 Module's Required Interfaces & Endpoints:

Define the "plugs" on your module. What specific data does it need to receive to function, and what actions or events does it need to trigger in its host environment?

(e.g., **Input:** Requires a user\_id string and a post\_content text block. **Output:** Triggers an onSummaryComplete event with a summary\_text payload.)

2.3 Core Technologies & Dependencies:

What language, framework, and key libraries is your module built with? List all external dependencies required for it to operate.

(e.g., **Language:** TypeScript. **Framework:** None (vanilla). **Dependencies:** d3.js for data visualization, zod for internal schema validation.)

2.4 Data Schema & Output:

Describe the structure and format of the primary data your module produces. This is not about how it's stored, but the shape of the data itself.

(e.g., **Output Format:** JSON. **Schema:** { "summary": "string", "keywords": ["string"], "sentiment\_score": "number" })

2.5 Intended User Roles:

Describe the target user for this module. Who is it designed for?

(e.g., All users, authenticated users only, administrators, content editors.)

### **Section 3: Guardian Metadata (For Internal Review Only)**

*This section is generated by the Guardian AI after parsing the applicant's submission. It is not visible to the applicant.*

**3.1 Internal Risk Level:**

(Generated: Low, Medium, High, Critical)

**3.2 Suggested Isolation Level:**

(Generated: Sandboxed IFrame, Containerized Service, Direct Core Access)

**3.3 Observed Conflicts with Past Integrations:**

(Generated: e.g., "This module's dependency on old-lib-v1.2 conflicts with Module-B which requires old-lib-v2.0. Potential for namespace collision.")

**3.4 Recommendation:**

(Generated: Approve, Quarantine, Analyze Deeper, Reject)