Rob: Architecture Overview

1 Architecture and Bindings Used

1.1 Durable Objects for Agents

Each user session is represented by Durable Objects, each DO maintains the agents specific to that session, allowing different sessions to have agents with different configurations.

1.2 AI Workers for Communication with LLMs

To interact with language models, AI Workers are used to send requests and receiving responses from the LLMs remotely.

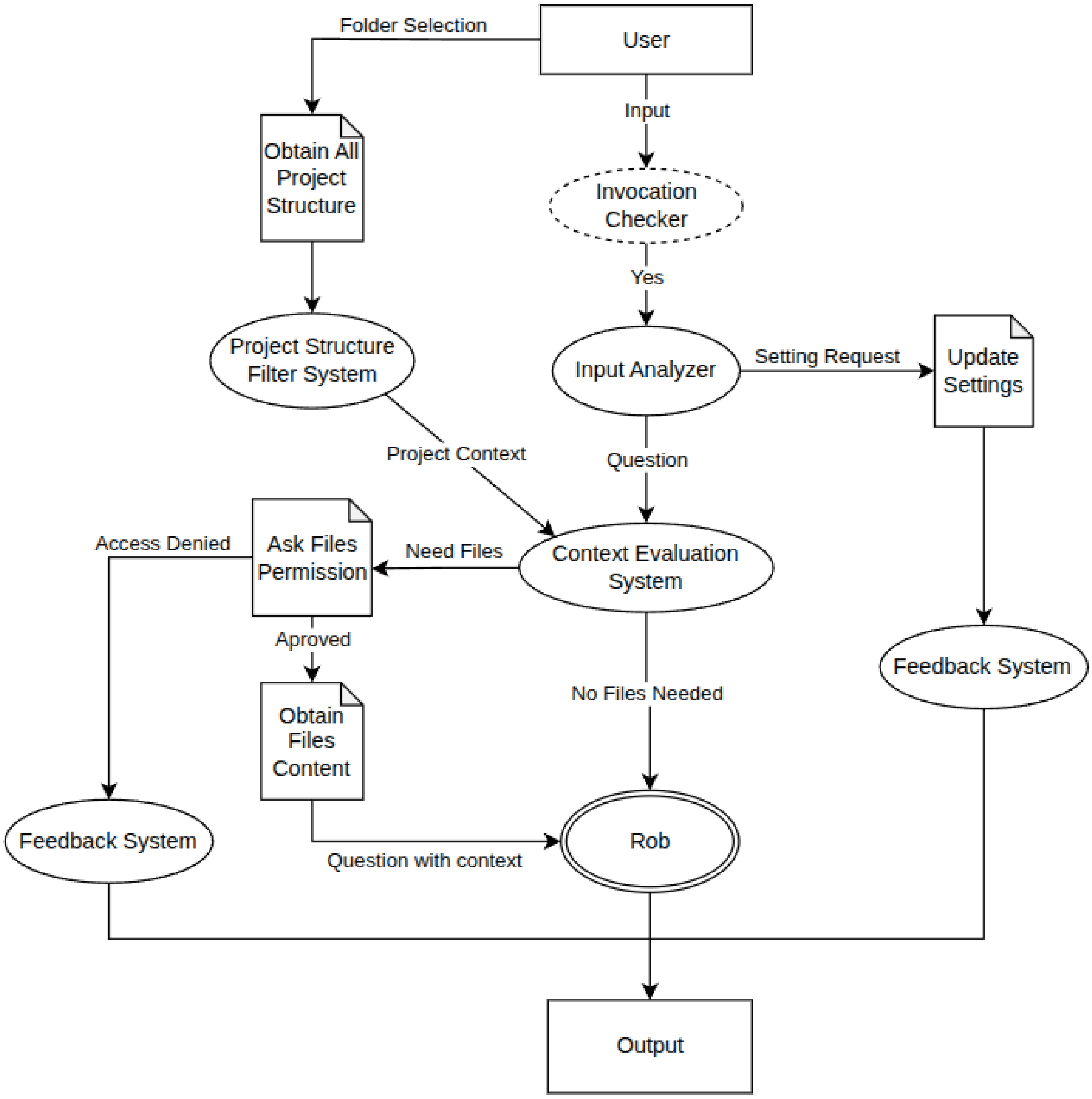
1.3 Data Persistence with D1 Database

All relevant information regarding sessions and chats is stored in a D1 database to ensure no information is lost over time and allowing conversations or agent configurations to be resumed in future sessions.

1.4 Assets Binding for Front-End

On the front-end, the assets binding is used to provide an interactive and responsive interface to the user.

2 Architecture Design



- **Single-line circles**: Specialized LLM-based systems Model: `@cf/meta/llama-3-8b-instruct`
- **Double-line circle**: Main assistant (**Rob**) Model: `@cf/meta/llama-3-8b-instruct`
- **Rectangles**: System components or actions
- **Arrows**: Control or data flow between components

3 User Interaction and Invocation

The interaction begins with the **User**, who can communicate with Rob using voice or text.

- The user provides input.
- An Invocation Checker verifies whether Rob has been explicitly invoked.
- If Rob is not invoked, the system remains idle.
- If invoked, the input proceeds through the processing pipeline.

4 Project Context Initialization

Before asking questions, the user may select a project folder.

- The system obtains the complete project structure (folders and file paths only).
- A Project Structure Filter System removes irrelevant data such as dependencies or build artifacts.
- The resulting project context represents the real structure of the project without exposing file contents.

5 Input Analysis and Settings Management

When a question is submitted:

- The Input Analyzer inspects the request.
- If the input is a settings command (e.g., changing Rob's name or default programming language), it is routed to the Update Settings system.
- The Feedback System confirms the applied changes to the user.
- If the input is a programming question, it is forwarded for contextual evaluation.

6 Context Evaluation and Human-in-the-Loop

The Context Evaluation System determines whether Rob can answer the question using the available information.

It evaluates:

- The current user question
- Recent conversation history
- The project structure (if available)

Two outcomes are possible:

6.1 No Files Required

- The question can be answered without inspecting any files.
- The request is sent directly to Rob.
- Rob generates a response and returns it to the user.

6.2 Files Required

- The system determines that specific files are required.
- Rob asks the user for explicit permission, stating:
  - Which files are needed
  - Why access is required
- If access is denied, the system provides feedback to the user.
- If access is approved:
  - The content of the approved files is obtained
  - The question is enriched with this context
  - The contextualized request is sent to Rob

7 Response Generation

Once Rob receives the final request (with or without file context):

- Rob generates a concise, developer-oriented response.
- The response is streamed back to the user.
- The interaction is stored for future context.