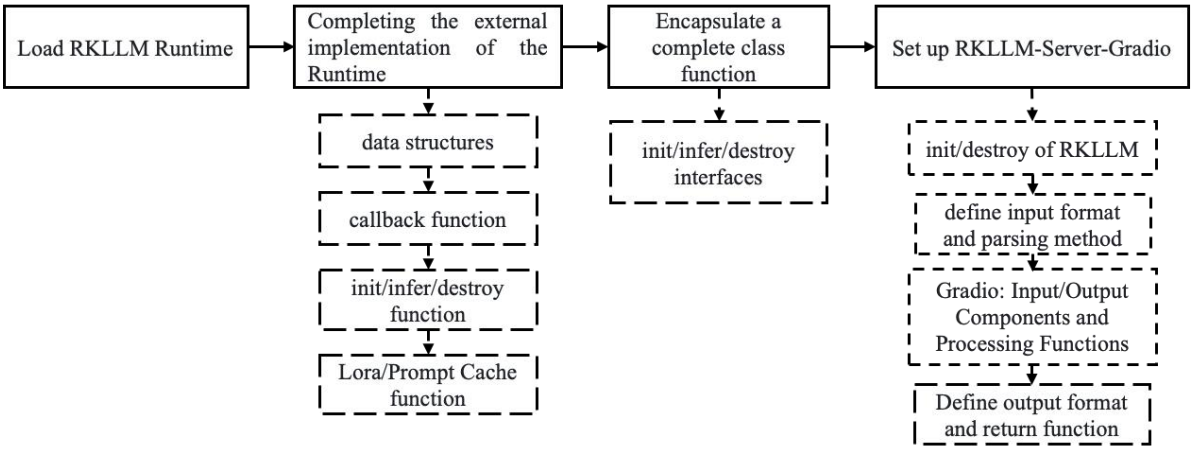

Figure 3-6 Overview of the RKLLM-Server-Flask Deployment Implementation Process

The difference is that RKLLM-Server-Gradio chooses to use the gradio library to implement the server setup and complete communication with the client, providing a simple web-based service. This requires specific handling of the Gradio interface in RKLLM-Server-Gradio as follows:

1) The Gradio function library provides complete communication for input and output data, so there is no need to define complex input-output implementations for the server via Flask.

2) Gradio is a deployment framework based on different control elements. During usage, it is necessary to call various Gradio components to complete the interface design and specify the trigger conditions, function call logic, and the data flow logic between different components for each element.

Users can refer to the main code in rkllm_server/gradio_server.py to understand the specific implementation of RKLLM-Server-Gradio, and by modifying the initialization definitions for the RKLLM model within it, they can implement different custom models. Additionally, users can refer to the deployment example of the RKLLM-Server-Gradio to deploy their own custom server.

3.4.2.3 Client: RKLLM-Server-Gradio Usage Instructions

After successfully deploying the RKLLM-Server-Gradio on a Linux development board, users can access it via two methods: "Interface Access" and "API Access".

1) Interface Access: Upon successfully starting the RKLLM-Server-Gradio with the one-click deployment script, users can directly access the RKLLM model for quick interaction by opening any web