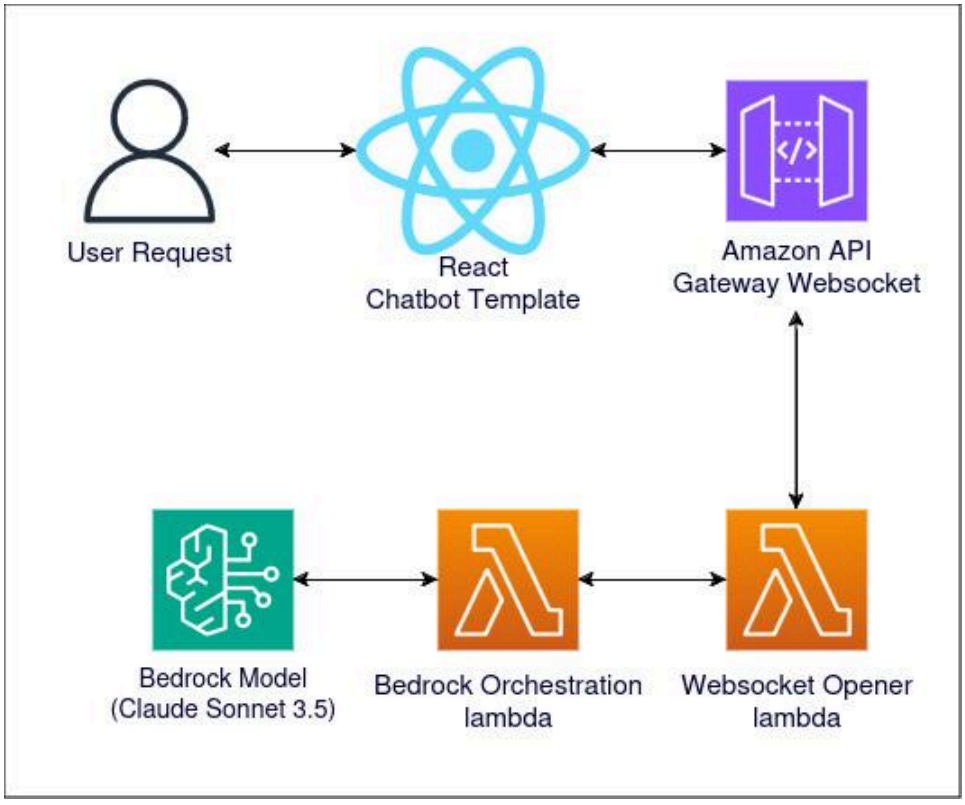
Bedrock Streaming Response

What it does

The Response Streaming setup for bedrock is the same pipeline as the standard Bedrock invocation process, with one major difference. Rather than returning the full text in a single request, it returns each token as they're outputted. This allows for quicker response times, and more interesting frontend UIs.

The specific demo uses a React hosted frontend utilizing the Chatbot Template (https://github.com/ASUCICREPO/ChatbotTemplate), this invokes the backend through an Api Gateway Websocket, rather than the standard REST api gateway. Through this the request is routed to a lambda function, the "connection opener" lambda. The purpose of this lambda is to allow the initial opening of the websocket connection, and to call the main function. (TODO: I believe this can be done with step functions to simplify to one lambda) Next the main lambda function will call bedrock through the converse stream API, for each token returned, it will be sent back to the api gateway to the frontend.



How we can use it at the CIC

The main way that we can use it at the CIC is to improve the Time to First Token. For most LLM pipelines the process looks like:

Request->Processing->Generation->Response in Full

By streaming the response we can fully get rid of the Generation step, since the first response is received as soon as the first token is generated.

Request->Processing->Response of First Token

This is especially useful for projects that require more complex thought (Longer responses). The longer the average response length, the more time implementing streaming can save. This saved time can be used to do additional pre-processing (RAG, more complex thought time, etc), or use a slower but more intelligent model, such as Haiku -> Sonnet/Opus. The other major use for streaming the response is to create more interesting frontend UI's, since you receive each token, you can create a cursor effect that shows the output as its generated, boosting the appeal of the design.

In summary, it is best used for projects with complex thinking processes, and long responses to decrease the time to the first token, and for better looking UIs.