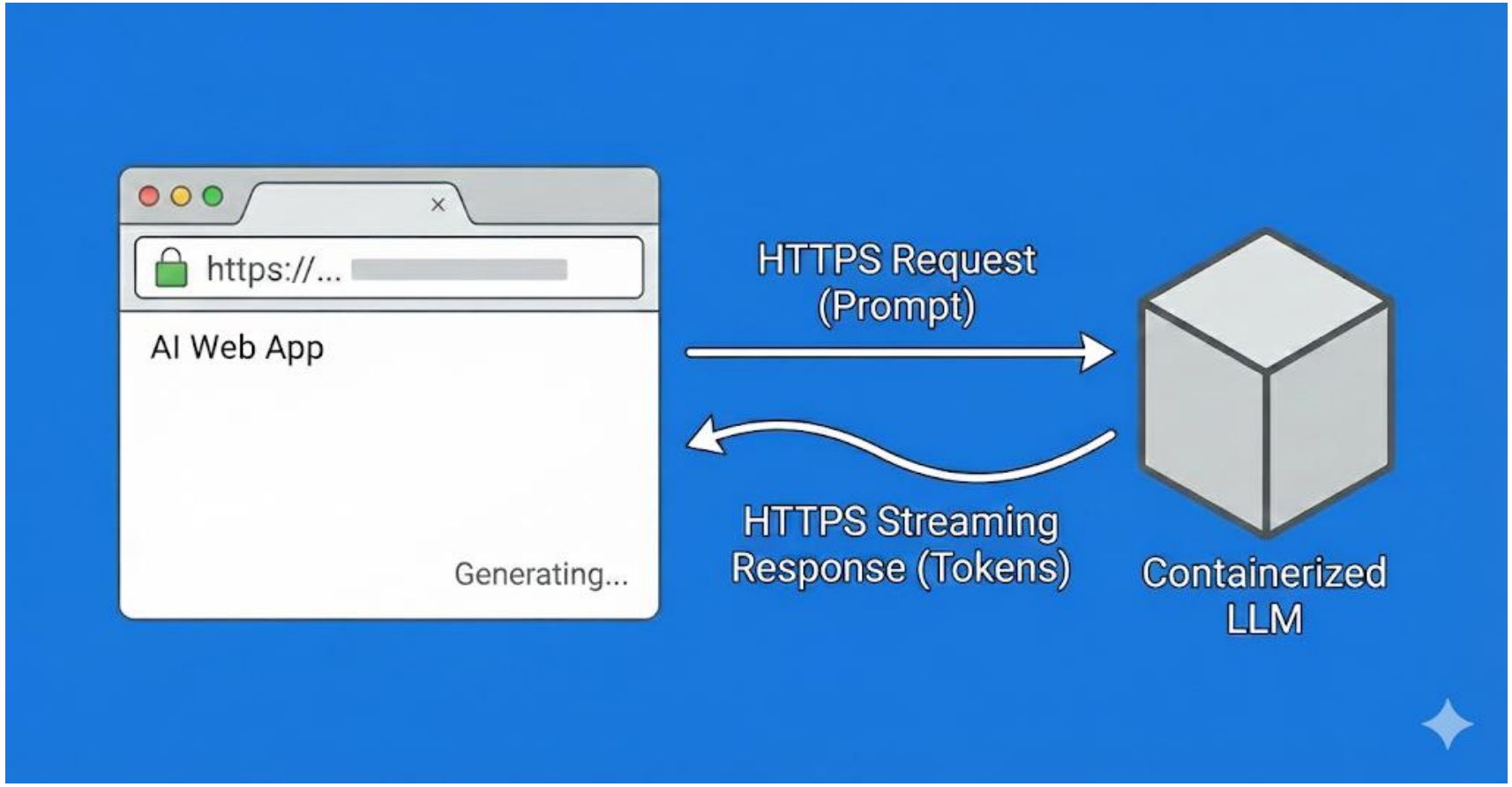
AI Chat Bots

https://jimfawcett.github.io/NewSite/Code/CodeBites_ChatBotAI.html

Introduction

- Chat Bots are web applications consisting of:
  - **Browser-based interface** that accepts prompts and displays results
  - A containerized pre-trained **Large Language Model (LLM)** running in a Linux server in a data center
  - **Https-based communication infrastructure** to ferry prompts to, and results from the LLM

AI Web App Structure



Limitations

- AI Chat Bots have limitations:
  - Cannot read from or write to local repositories because they are browser based
  - Cannot read files from github repository

Work Arounds

- You can paste code text at the end of a prompt
  - Prompt specifies what to do with the following code
- You can create a zip for a local repository or download a github zip
  - Paste zip file at the end of a prompt
- You can request Chat Bot to download its results to your local download directory

Example

– Size and Complexity of code functions

- This prompt:

  “Generate a list of function line counts and complexities for all functions in the zip file pasted at end of this prompt.

  Line count is total number of lines including code, whitepace, comments

  Complexity is the number of open braces in each function”

- Generated a list of all the functions with correct line counts and complexities for a Rust Thread Pool.
  - Note: Rust code is easier to parse than C++ and doesn’t have function overloads.

Conclusion

- You can find all the details here:

  https://jimfawcett.github.io/NewSite/Code/CodeBites_ChatBotAI.html

- The next presentation explores:
  - AI Agents
  - Use local application to communicate with the LLM
  - Uses HTTPS-based API
  - Can read from and write to local file system
- The one after that explores:
  - AI Consoles
  - Have access to local repositories
  - Don’t need to write Agent code