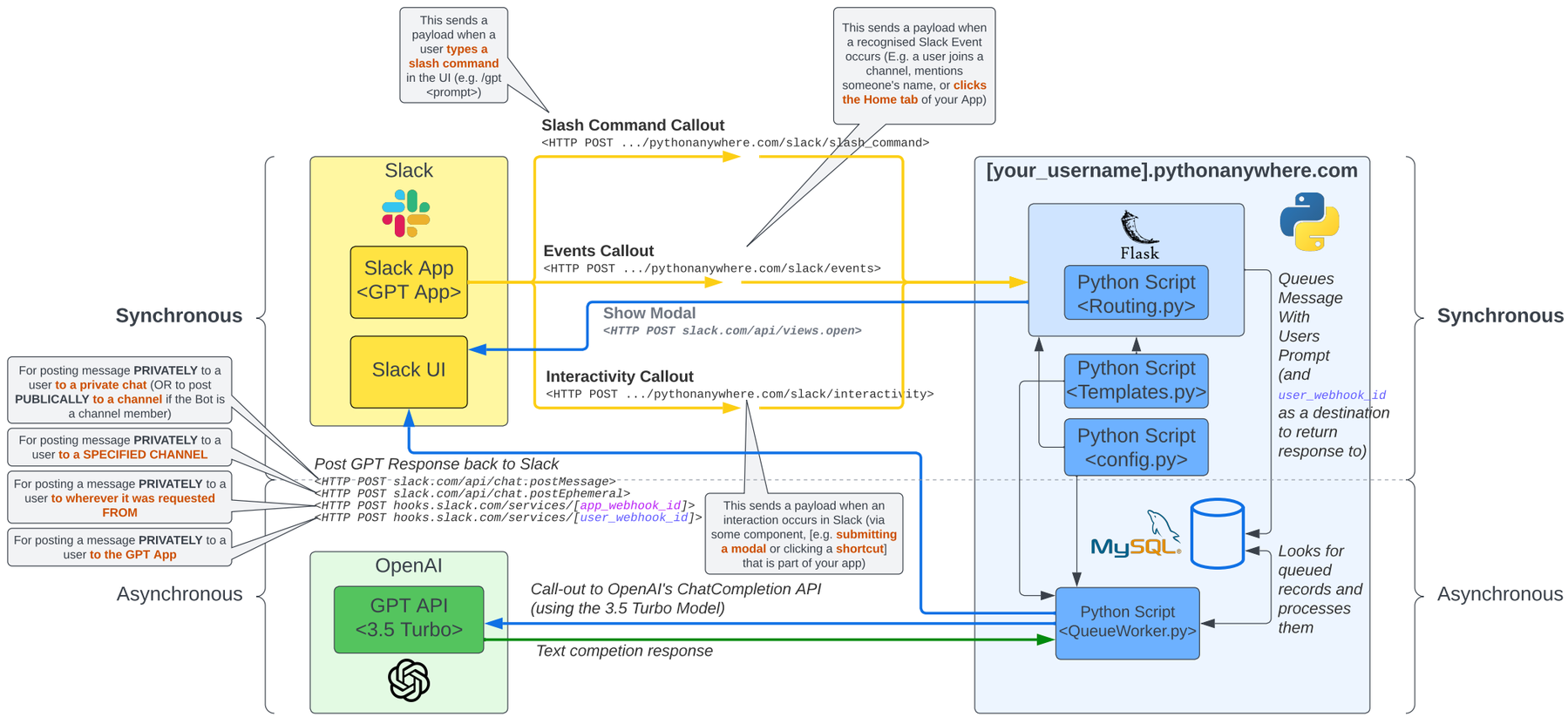
This sends a payload when a user types a slash command in the UI (e.g. /gpt <prompt>)
This sends a payload when a recognised Slack Event occurs (E.g. a user joins a channel, mentions someone's name, or clicks the Home tab of your App)
Slash Command Callout
<HTTP POST .../pythonanywhere.com/slack/slash_command>
Slack
Slack App <GPT App>
Slack UI
Events Callout
<HTTP POST .../pythonanywhere.com/slack/events>
Show Modal
<HTTP POST slack.com/api/views.open>
Interactivity Callout
<HTTP POST .../pythonanywhere.com/slack/interactivity>
[your_username].pythonanywhere.com
Flask
Python Script <Routing.py>
Python Script <Templates.py>
Python Script <config.py>
Queues Message With Users Prompt (and user_webhook_id as a destination to return response to)
Synchronous
Synchronous
For posting message PRIVATELY to a user to a private chat (OR to post PUBLICALLY to a channel if the Bot is a channel member)
For posting message PRIVATELY to a user to a SPECIFIED CHANNEL
For posting a message PRIVATELY to a user to wherever it was requested FROM
For posting a message PRIVATELY to a user to the GPT App
Post GPT Response back to Slack
<HTTP POST slack.com/api/chat.postMessage>
<HTTP POST slack.com/api/chat.postEphemeral>
<HTTP POST hooks.slack.com/services/[app_webhook_id]>
<HTTP POST hooks.slack.com/services/[user_webhook_id]>
This sends a payload when an interaction occurs in Slack (via some component, [e.g. submitting a modal or clicking a shortcut] that is part of your app)
MySQL
Looks for queued records and processes them
Asynchronous
Asynchronous
OpenAI
GPT API <3.5 Turbo>
Call-out to OpenAI's ChatCompletion API (using the 3.5 Turbo Model)
Text competion response
Python Script <QueueWorker.py>