

Serverless Holiday Hackathon 2023





Our plans for today

About Serverless Guru



Bedrock Code Samples

Building Our Own AI Chatbot



Q & A



About Serverless Guru

Your Guide to Serverless
Adopt. Migrate. Build.

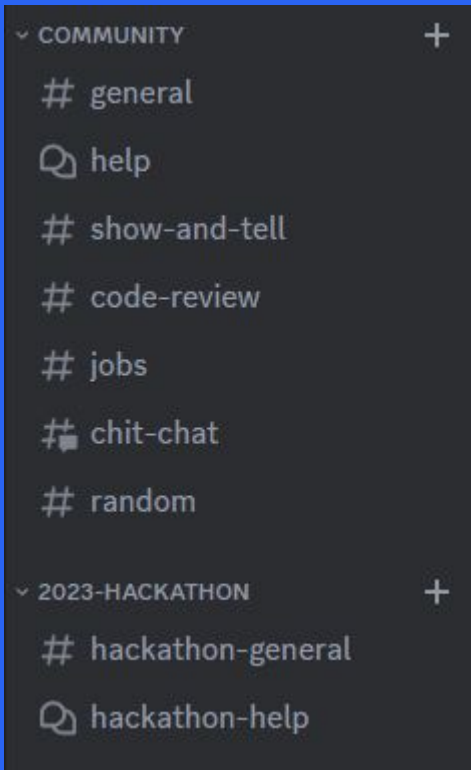


Our Company

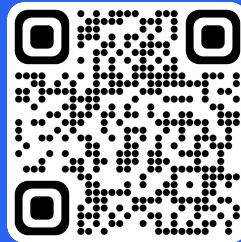
We love to take on challenges, face the unknown, and be on the forefront of serverless innovation.

Helping companies and people realize their full potential gives us energy.

Speak to a Guru today!



Join our discord Server!



Submission Requirements

Submit your project

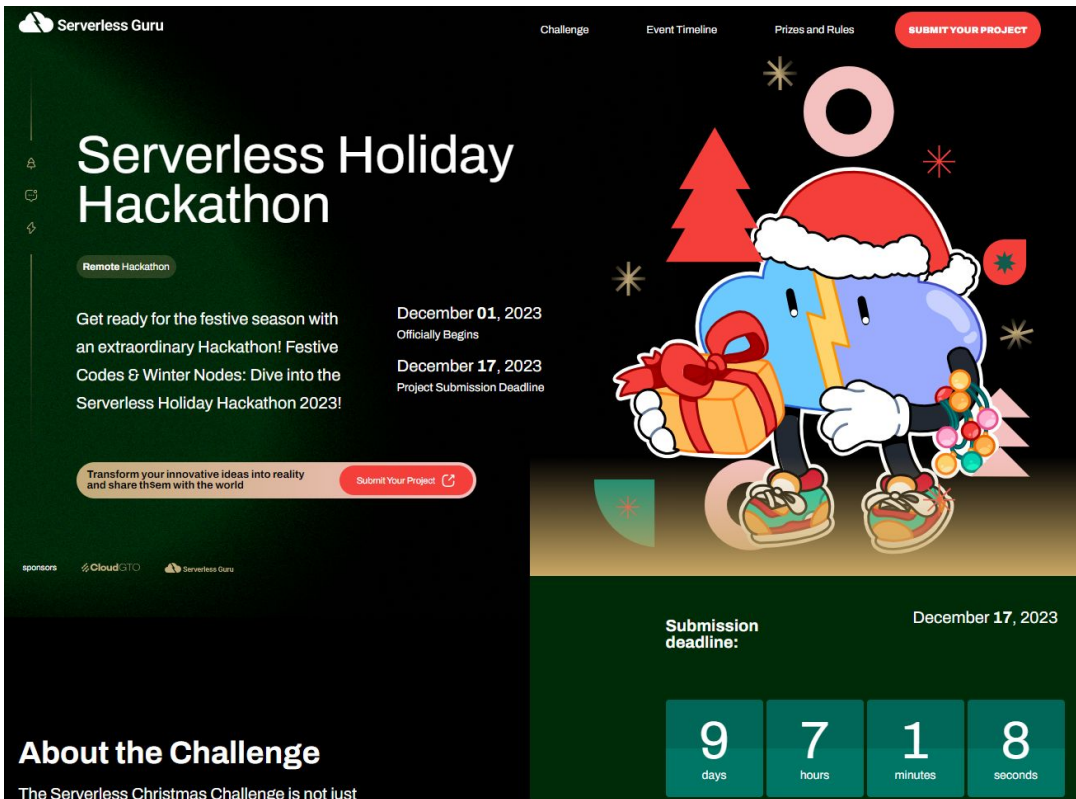
What is required?

- Link to **Article**: Published Article link (Hashnode, Medium, dev.to, ...)
- Link to **Demo Video**: Has to be accessible via public link (Youtube, Loom, ...)
- Link to **Public Repository**: Repository has to be public and visible by anyone
- Any additional comments, f.e.: the link to hosted app, any required credentials, ...

Deadline December 17, 2023, 11:59 PM PST

Any change made after the deadline won't be reviewed

Submissions Form



The screenshot shows the 'Serverless Holiday Hackathon' submission form. The header includes the 'Serverless Guru' logo, navigation links for 'Challenge', 'Event Timeline', and 'Prizes and Rules', and a red 'SUBMIT YOUR PROJECT' button. The main content area features the event title, a 'Remote Hackathon' tag, and a description: 'Get ready for the festive season with an extraordinary Hackathon! Festive Codes & Winter Nodes: Dive into the Serverless Holiday Hackathon 2023!'. It lists key dates: 'December 01, 2023 Officially Begins' and 'December 17, 2023 Project Submission Deadline'. A central illustration depicts a blue cloud character wearing a Santa hat, holding a gift and a string of lights, with a red Christmas tree in the background. At the bottom, a submission deadline counter shows '9 days', '7 hours', '1 minutes', and '8 seconds' remaining, with the date 'December 17, 2023'. The footer includes the 'About the Challenge' section, which begins with 'The Serverless Christmas Challenge is not just'.

Serverless Guru

Challenge Event Timeline Prizes and Rules [SUBMIT YOUR PROJECT](#)

Serverless Holiday Hackathon



Remote Hackathon

Get ready for the festive season with an extraordinary Hackathon! Festive Codes & Winter Nodes: Dive into the Serverless Holiday Hackathon 2023!

December 01, 2023
Officially Begins

December 17, 2023
Project Submission Deadline

Transform your innovative ideas into reality and share them with the world [Submit Your Project](#)

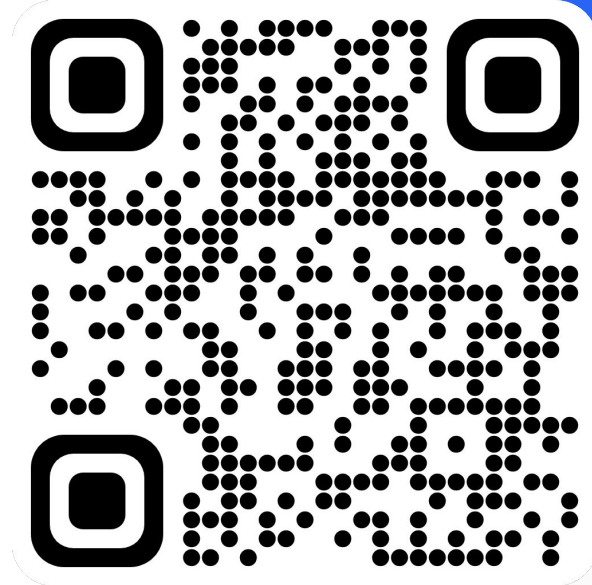
sponsors  

About the Challenge

The Serverless Christmas Challenge is not just

Submission deadline: December 17, 2023

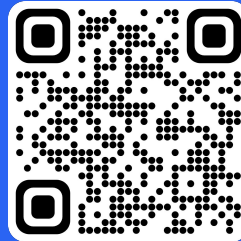
9	7	1	8
days	hours	minutes	seconds





Bedrock Code Samples

Examples to get started using AWS Bedrock with or without Langchain



Bedrock Code Samples

Samples

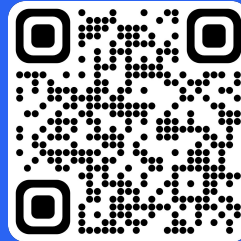
Following is a list of the currently available provided samples.

AWS SDK v3

- [Basic Example](#): Sample of executing a single synchronous call to AWS Bedrock.
- [Streamed Response](#): Improvement of the first example to allow for receiving streamed responses from Bedrock.
- [Bedrock Agent](#): Sample code on how to trigger the execution of an existing Bedrock Agent.

Langchain

- [Basic Example](#): Sample of executing a single synchronous call to AWS Bedrock using the Langchain framework.
- [Streamed Response](#): Example code to get a streamed response from Bedrock when executing it with Langchain.
- [Conversation Chain + Memory](#): Sample of how to configure Langchain to use a Conversation Chain and store the Conversation History on a DynamoDB table
- [Conversation Chain + Memory + Streamed Response](#): Improved the previous sample to also return the Bedrock response as a stream.



Bedrock Code Samples

Preserve Conversation?

```
const client = new BedrockRuntimeClient({ region });

const mockRequest = async (userInput) => {
  const textInput = `\n\nHuman: ${userInput}\n\nAssistant:`;
  const command = new InvokeModelCommand({
    ...
  })
},
});
```

```
const client = new BedrockAgentRuntimeClient({ region });

const mockRequest = async (sessionId, userInput) => {
  const input = {
    ...
  };
  const command = new InvokeAgentCommand(input);
```



Bedrock Code Samples

Streamed Response?

```
const client = new BedrockRuntimeClient({ region });

const mockRequest = async (userInput) => {
  const textInput = `\n\nHuman: ${userInput}\n\nAssistant:`;
  const command = new InvokeModelCommand({
    ...
  })
},
});
```

```
const client = new BedrockRuntimeClient({ region });

const mockRequest = async (userInput) => {
  const textInput = `\n\nHuman: ${userInput}\n\nAssistant:`;
  const command = new InvokeModelWithResponseStreamCommand({
    ...
  })
},
});
```



Bedrock Code Samples

Lang* vs AWS Ecosystem?

Fix streaming bug in Bedrock LLM family #3113

 Merged

jacoblee93 merged 4 commits into

main

from [brace/bedrock-streaming](#)

on Nov 1

```
const mockRequest =  
const mockResponse =
```

```
... model, new ConversationChain({  
  memory,  
  prompt: chatPrompt,  
});  
  
const res = await chain.call({ input: userInput });  
console.log(res);  
return res.response;  
};
```



Demo!

Check out our very own Santa Chatbot



Building Our Own AI ChatBot

Check out ho ho how we built our very own Santa Chatbot



Challenges Faced

Overview

- Implementing LLMs
 - Conversation Memory
 - Streamed Responses
- WebSocket API
 - Decoupling long-lasting execution
 - Authentication and State Management



Challenges Faced - LLM

Conversation Memory

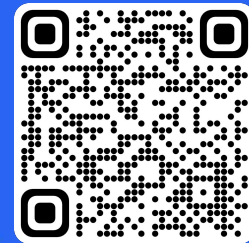
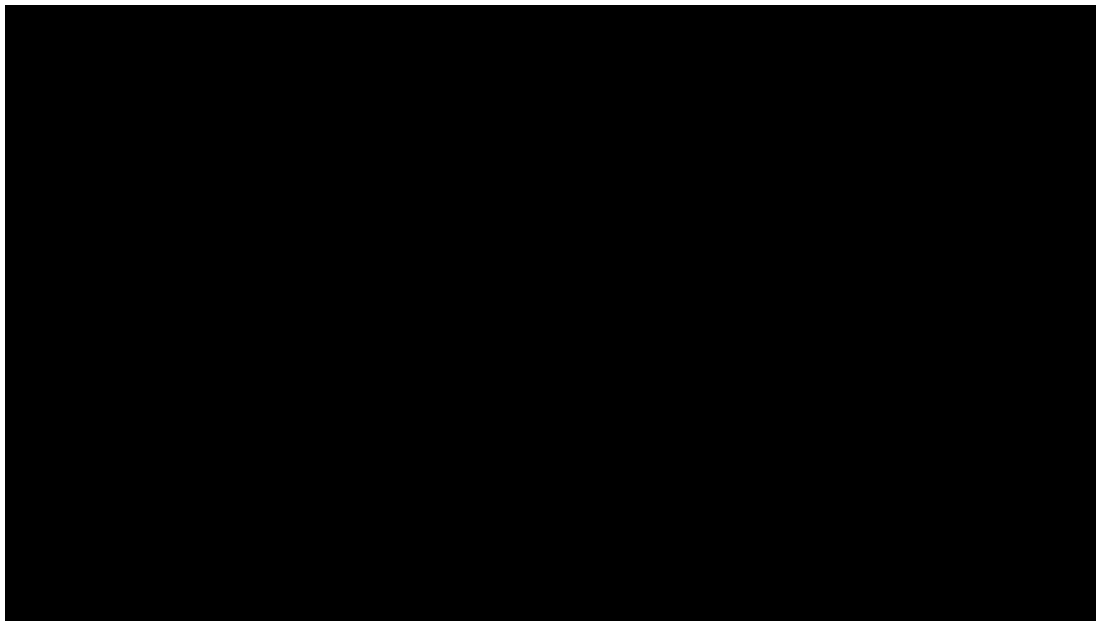
```
const memory = new BufferMemory({
  chatHistory: new DynamoDBChatMessageHistory({
    ...
  }),
  ...
});

const chain = new ConversationChain({
  llm: model,
  memory,
  prompt: chatPrompt,
});

const res = await chain.call({ input: userInput });
```


Challenges Faced - LLM

Streamed Responses

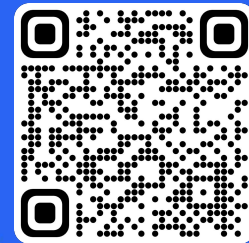




Challenges Faced - LLM

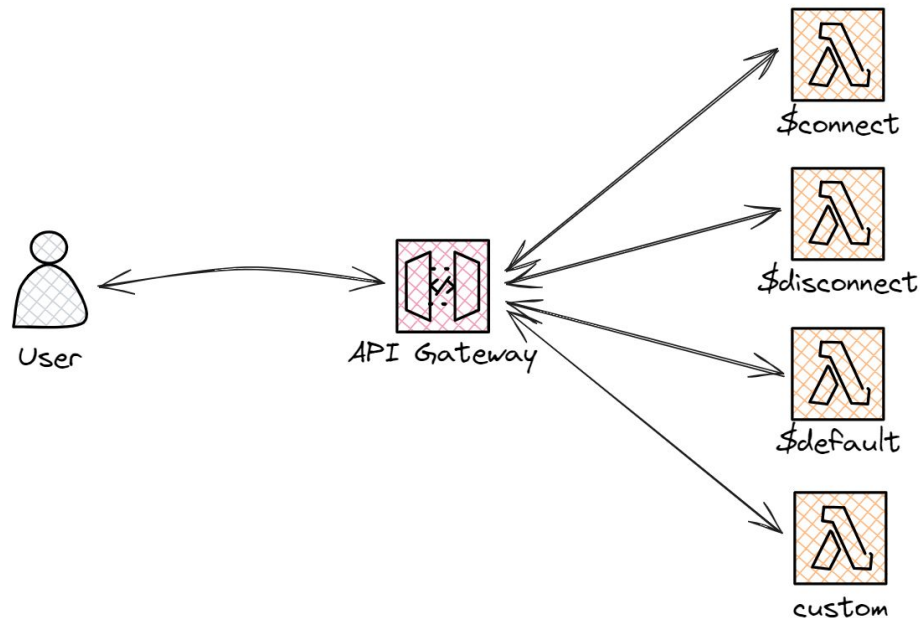
Streamed Responses

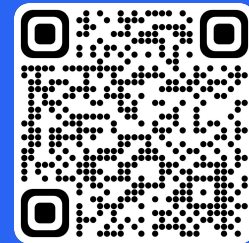
```
await chain.call({ input: userInput }, [
  {
    async handleLLMNewToken(token) {
      streamedResponse += token;
      console.log(streamedResponse);
    },
  },
]);
```



Challenges Faced - WebSocket

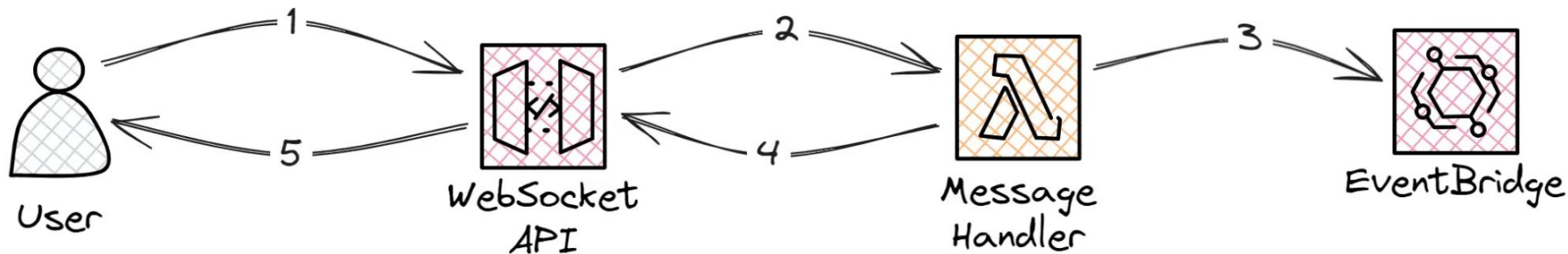
Decoupling long-lasting execution





Challenges Faced - WebSocket

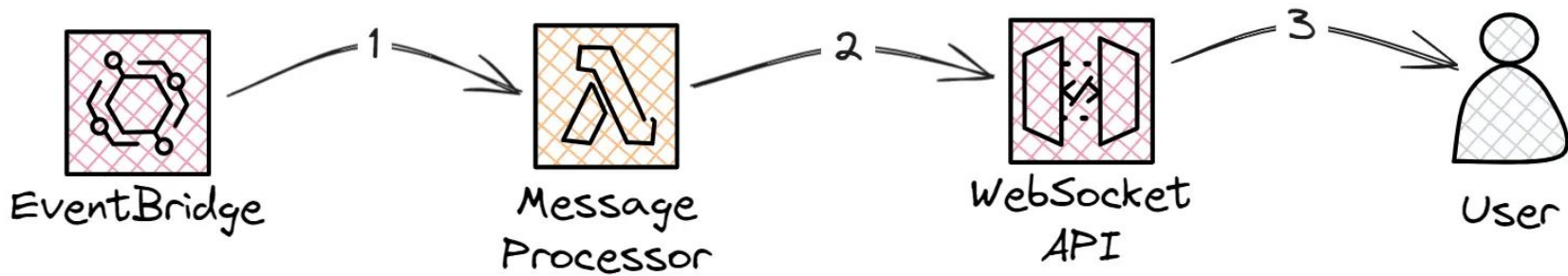
Decoupling long-lasting execution

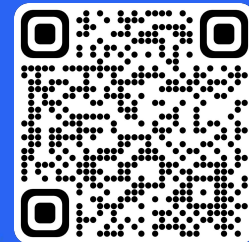




Challenges Faced - WebSocket

Decoupling long-lasting execution





Challenges Faced - WebSocket

Decoupling long-lasting execution

```
const {
  ApiGatewayManagementApiClient,
  PostToConnectionCommand,
} = require("@aws-sdk/client-apigatewaymanagementapi");

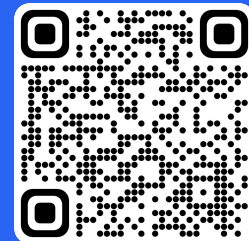
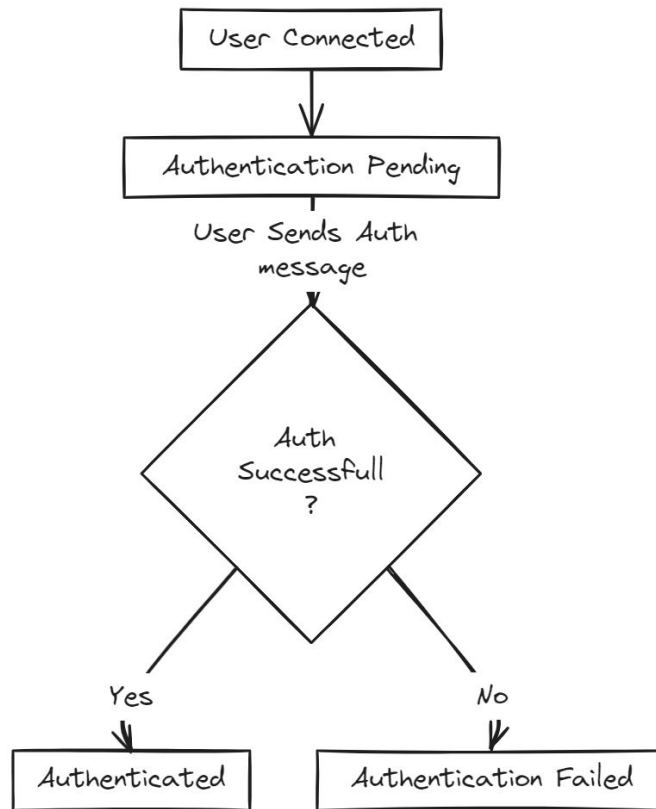
const client = new ApiGatewayManagementApiClient({
  endpoint: endpoint,
});

const command = new PostToConnectionCommand({
  ConnectionId: connectionId,
  Data: Buffer.from(JSON.stringify(data)),
});

await client.send(command);
```

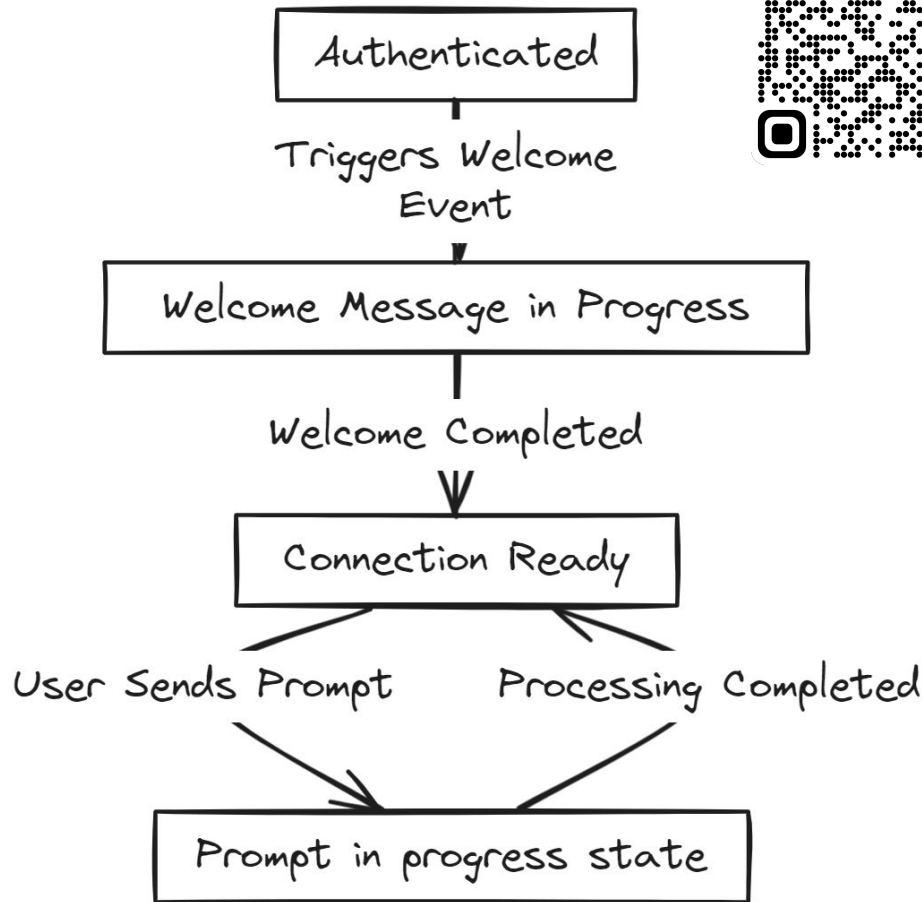
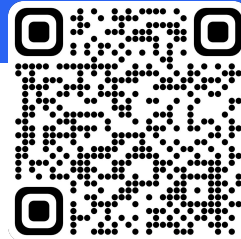
Challenges Faced - WebSocket

Authentication and State Management



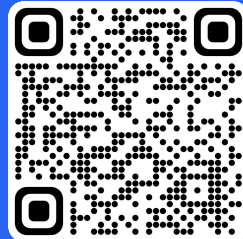
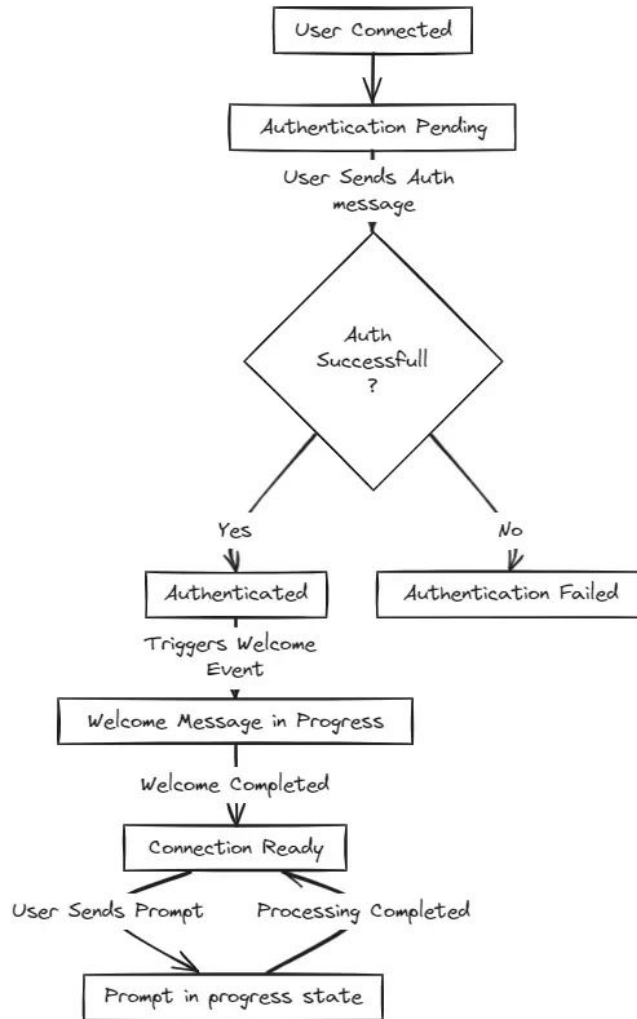
Challenges Faced - WebSocket

Authentication and State Management



Challenges Faced - WebSocket

Authentication and State Management

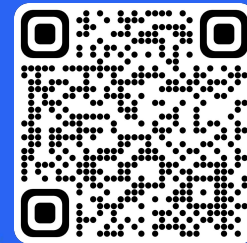




Architecture Overview

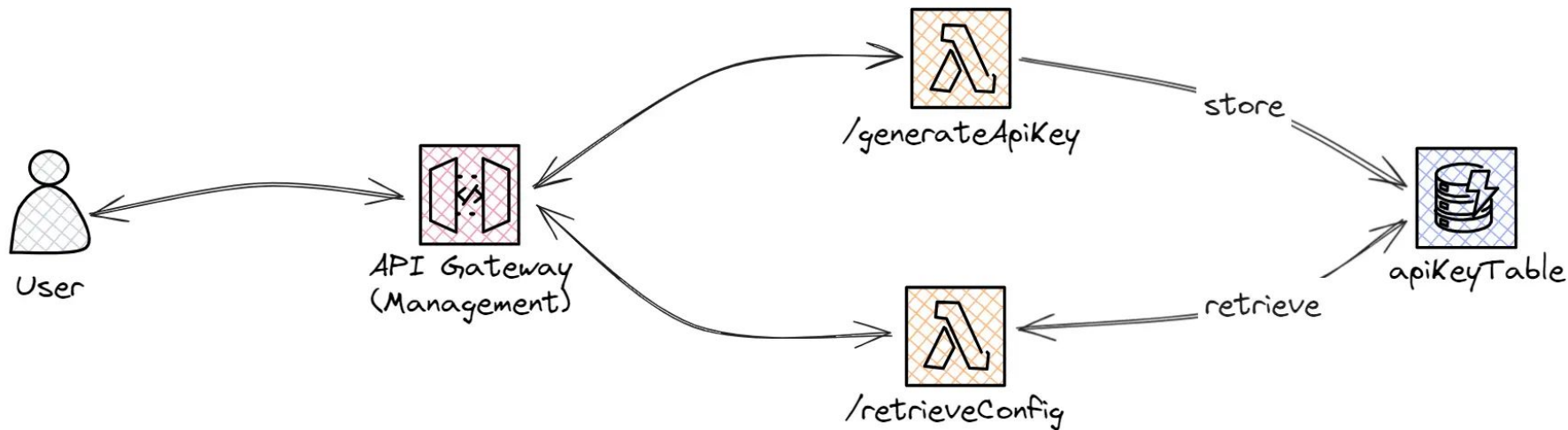
Overview

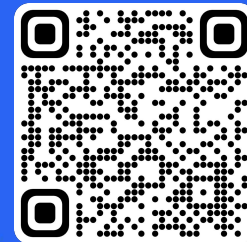
- Management API
- WebSocket API
 - Default Actions
 - Custom Actions
 - Decoupled Events



Architecture Overview

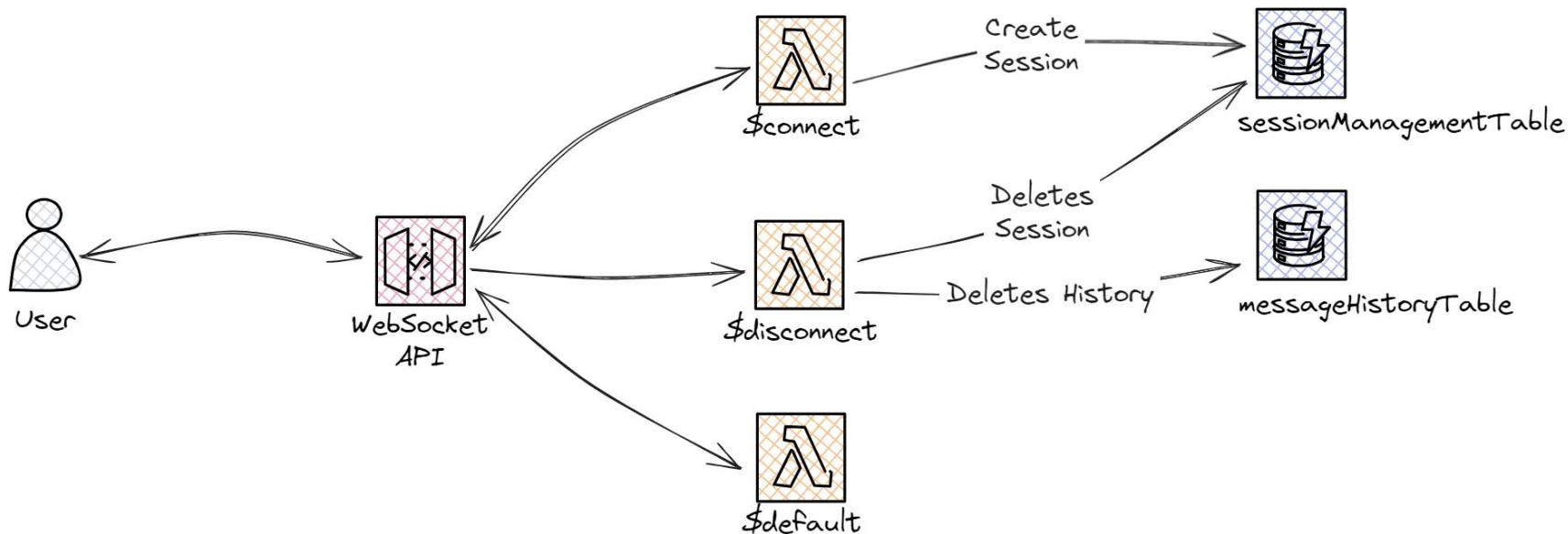
Management API

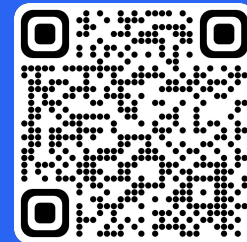




Architecture Overview

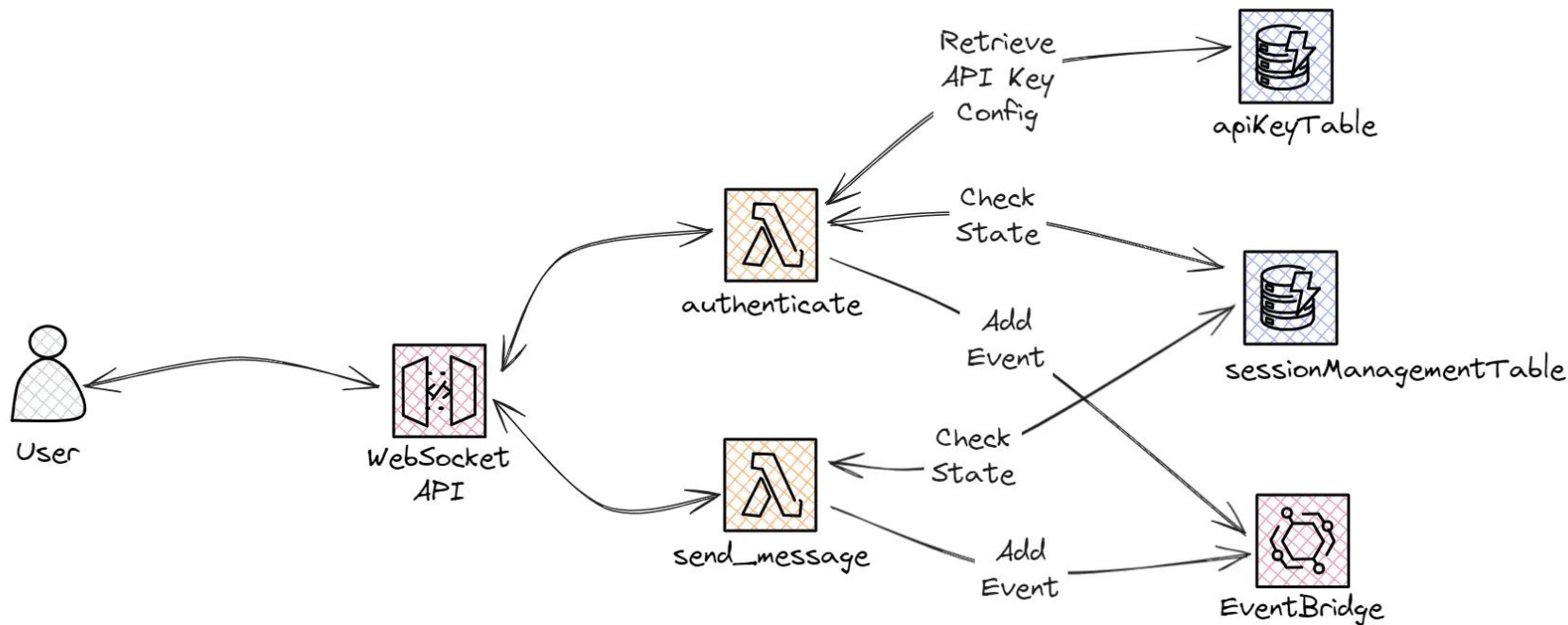
WebSocket API - Default Actions

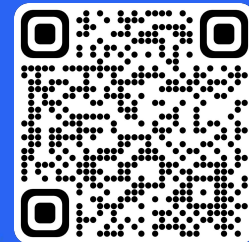




Architecture Overview

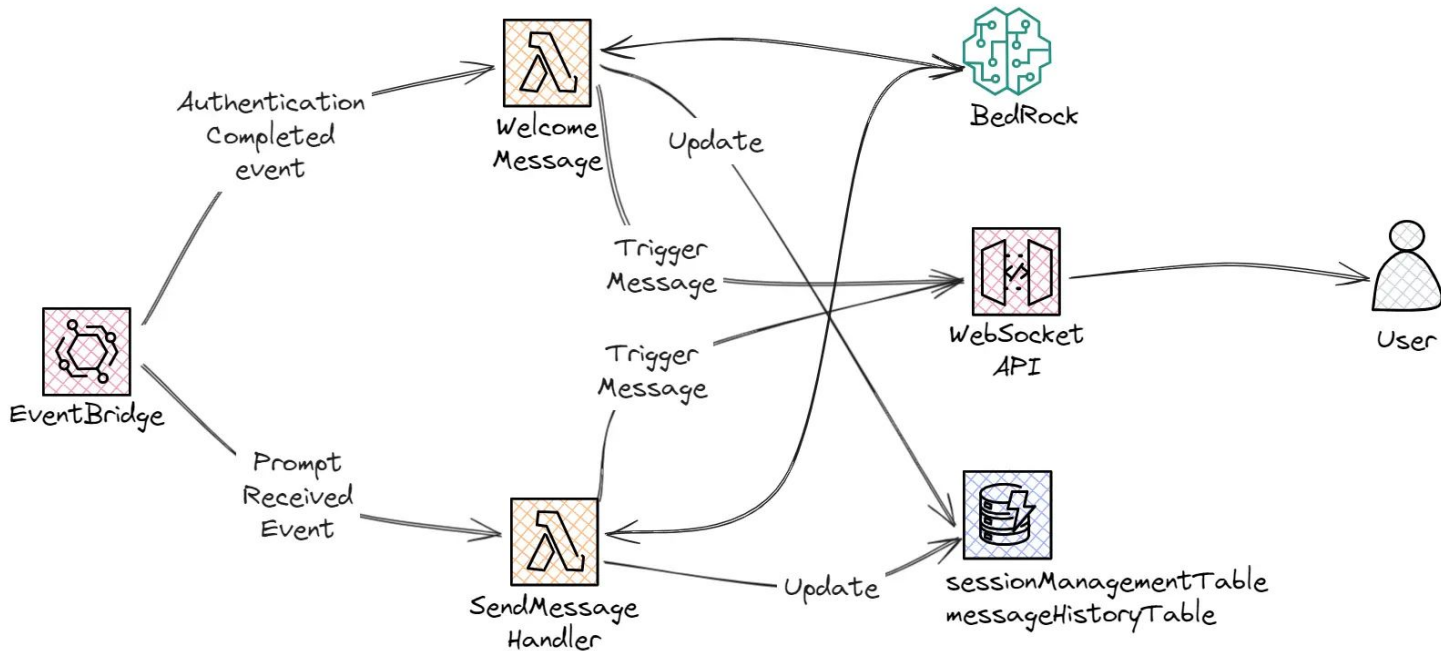
WebSocket API - Custom Actions

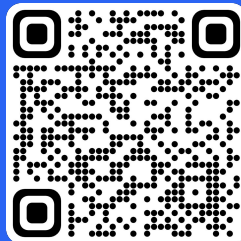




Architecture Overview

WebSocket API - Decoupled Events





Future Work

Possible Improvements

- Enable **voice-to-text** and **text-to-voice** transcription to take the application a step further and make it feel more like a real conversation and not just a chat.
- Improve **Authentication** and allow for **persistent message history**.
- Add any **additional features**, such as **implementing an Agent** to allow the LLM to query another API to f.e.: List possible gift ideas or build a custom wishlist to send to Santa.



Questions & Answers

We are here to help!

Raise any questions or concerns you might have

Thank you!



Join our discord Server!



Submit your Project!