

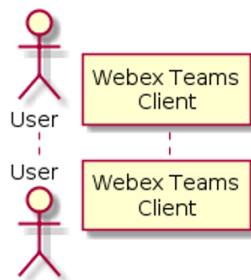
Task1: Identify Actors

Your first task will be to identify the actors and other participants in the process of ChatOps implementation.

- **Procedure1:**

Add a user and a Webex Teams Client.

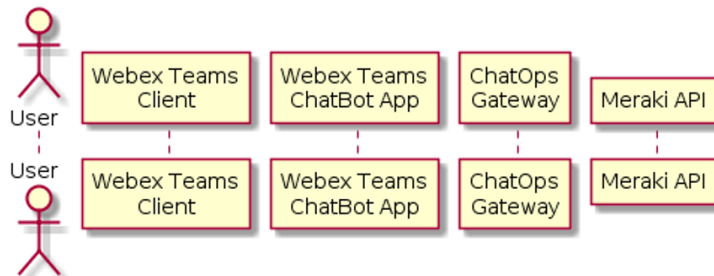
You can start modeling the sequence diagram by identifying the actors. The base for any ChatOps service is the user and the client, through which the user communicates with the system. In this example, the client will be the Cisco Webex Teams client.



- **Procedure2:**

Add a Webex Teams ChatBot App, a ChatOps gateway, and a Meraki API as participants.

Add the other participants. A Webex Teams ChatBot App is your bot application for Webex Teams that resides in the Webex Teams Cloud. A ChatOps gateway is an application server that handles ChatOps webhook events. And finally, the Meraki API is the API endpoint for management of the Cisco Meraki platform.



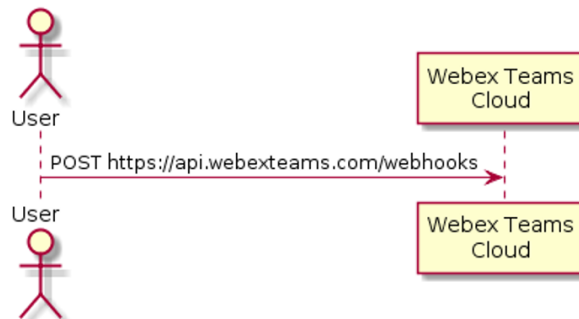
Task2: Model Cisco Webex Teams Callback Setup

First, you need to register a webhook with Webex Teams so that you are notified whenever a new message is posted to your ChatBot application. The activity assumes that the ChatBot application has already been created on Webex Teams Cloud. The sequence diagram will not be a part of the actual ChatOps sequence diagram. It serves as a reference for how to set up the callback in Webex Teams.

- **Procedure1:**

Register a webhook.

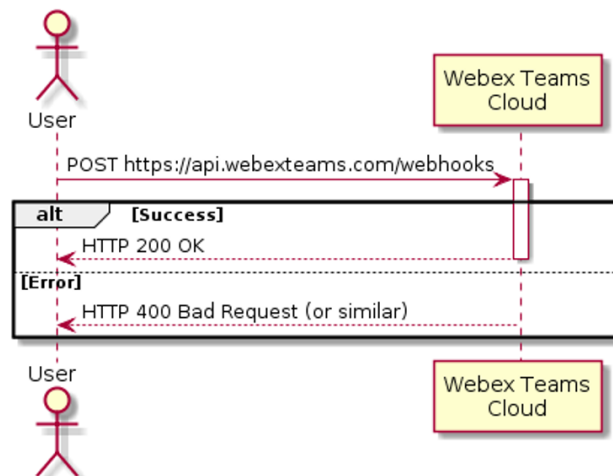
Cisco Webex Teams uses the API endpoint '/webhooks' to register a new webhook. User can use a tool like cURL or Postman to create an HTTP request. The webhook's target URL should be the URL of an event handler on the ChatOps gateway, such as, "https://public.example.com/events".



- **Procedure2:**

Model the Webex Teams response.

Cisco Webex Teams responds with either a 200 OK status, or a 400 or similar status if the webhook has not been registered successfully.



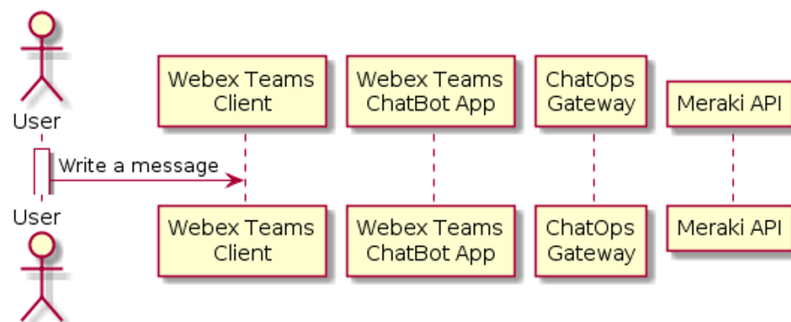
Task3: Model User Interaction

The functionality of the ChatOps system in the example can be broken down into two parts: the part that is triggered by the user and the part that is triggered by the callback. In this task you will model the user-triggered part of the sequence, up until the moment a callback happens.

- Procedure1:**

Model the User submitting a message using the Webex Teams Client.

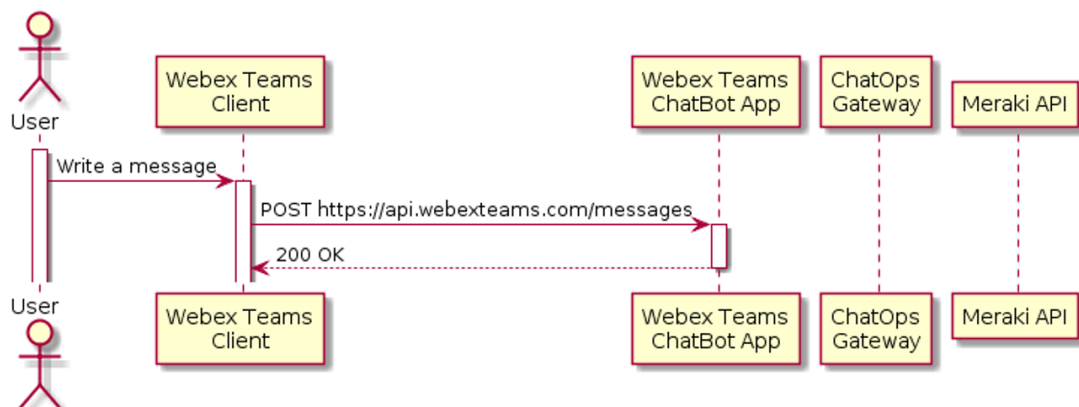
The sequence begins with the user writing a message in the Webex Teams Client (e.g. "Create SSID MyNetwork1"). This is a synchronous call which activates the user's lifeline, since the user expects feedback.



- Procedure2:**

Model the Webex Teams Client sending the message to the Webex Teams ChatBot App.

When the client application detects that a user has input a message, it relays that message as an HTTP request to the relevant room on Webex Teams – in this example, the room where the Webex Teams ChatBot App resides. This briefly activates the ChatBot application's lifeline. The App responds with an HTTP response.



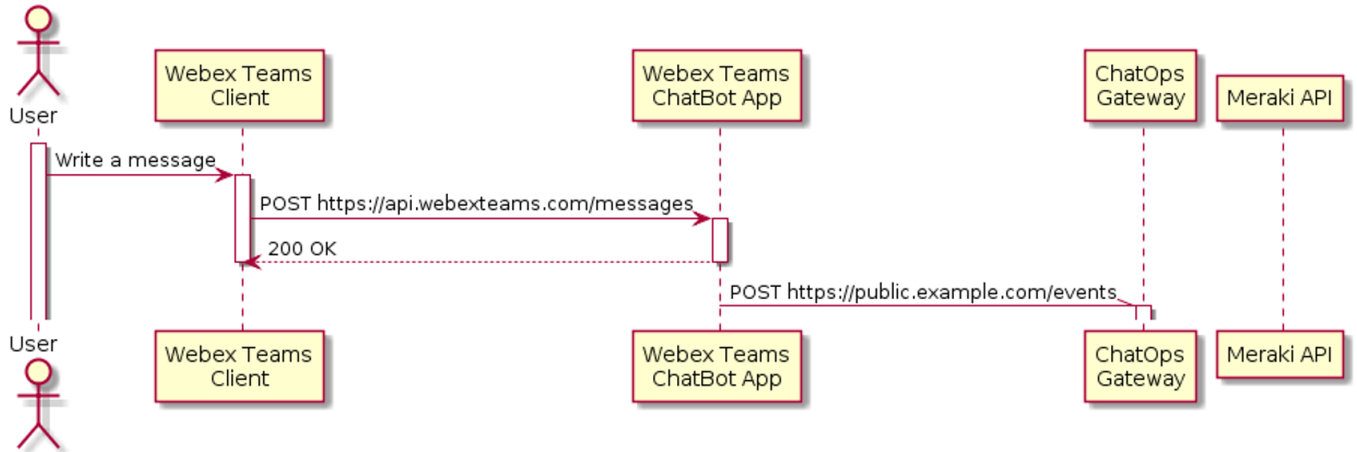
Task4: Model Cisco Webex Teams Callback

In this task, you will create the second part of the ChatOps functionality. You will model the webhook's callback as the message is received.

• Procedure1:

Model the webhook's callback.

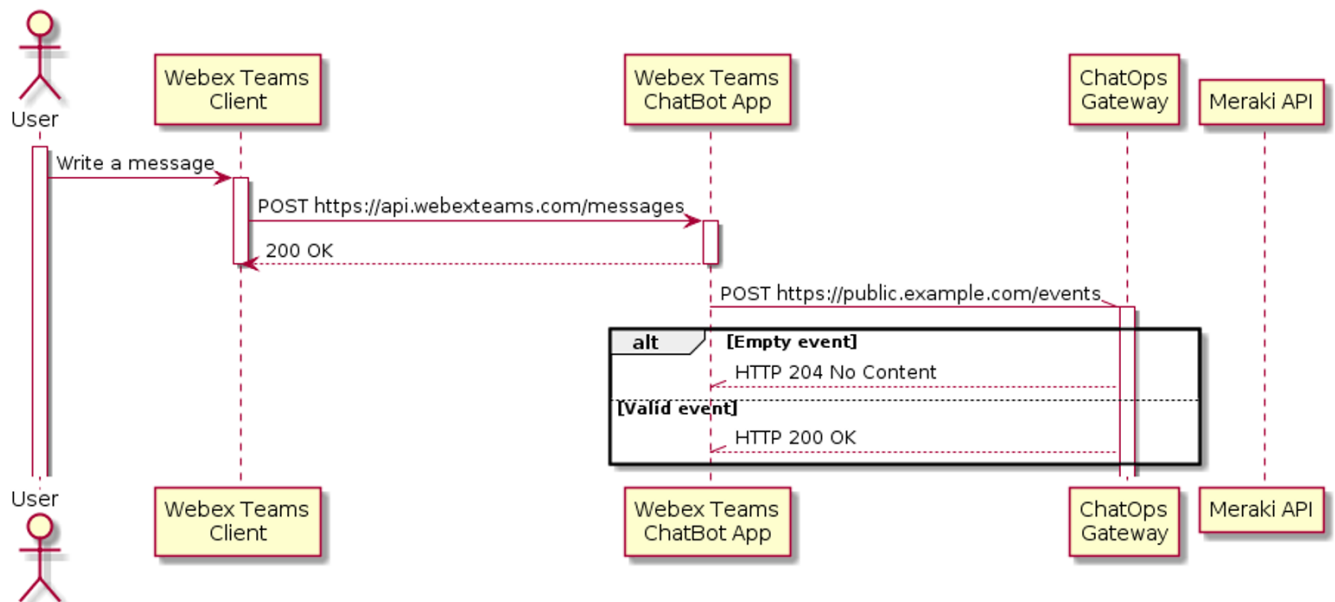
When the Cisco Webex Teams ChatBot App detects a new message, the webhook subscribed to the room will be triggered. The webhook will send a POST request toward the event handler of the ChatOps gateway. The gateway's lifeline will become activated, because the participant is performing an operation.



• Procedure2:

Handle the event.

ChatOps gateways handle many types of events. The first thing you need to do is handle the event sent as an HTTP request toward the ChatOps gateway. Events can be valid (response 200 OK), or empty (response 204 No Content). The choice of how to handle the event is represented with an alternative operator. While the HTTP responses from the Gateway could also be 4xx or 5xx for malformed requests, they are not covered in this example.



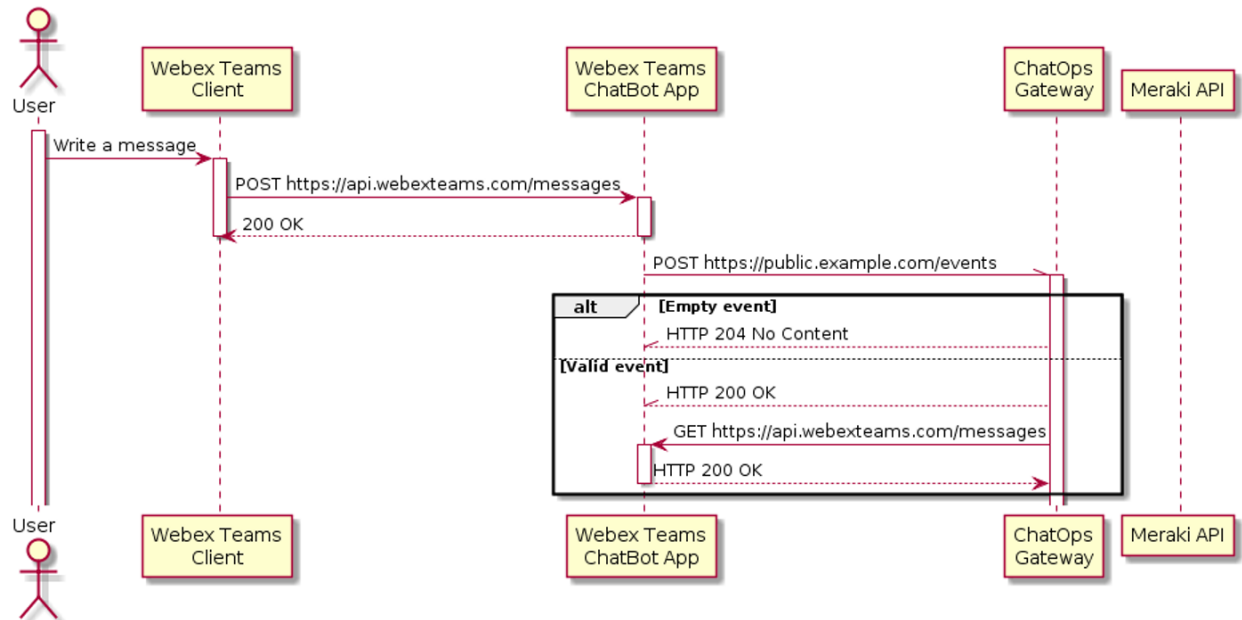
Task5: Model back-end System Call

For the final task, your job is to model what the ChatOps gateway does with the information it receives in the event.

• Procedure1:

Read the extra message data.

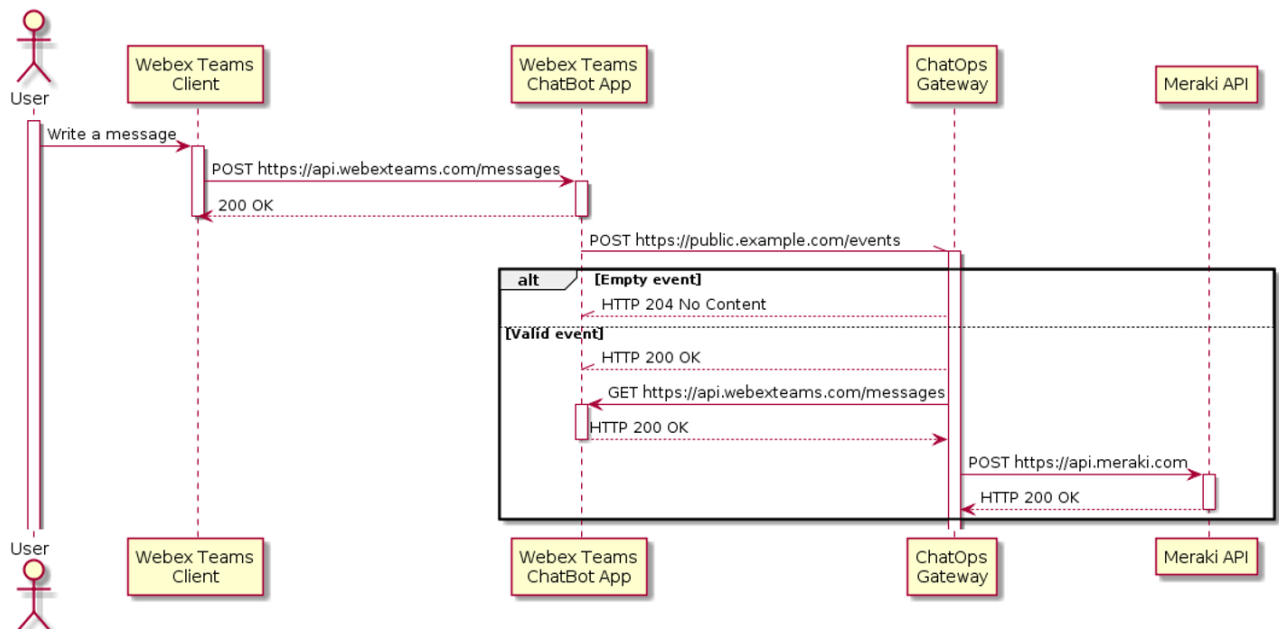
Since Webex Teams webhook events do not contain any actual data about the message, only references and IDs of the users and resources, this data must be read from the ChatBot Application. An extra API call must be made to the Webex Teams cloud for a message and for a user.



• Procedure2:

Create a new SSID via Meraki API.

ChatOps gateway now has all the information required to create a new SSID. Create a POST request toward the Meraki API to create it.



• Procedure3:

Notify the user.

You can now notify the user that a new SSID has been created. Send a POST request to the ChatBot application, which can then post a new message to the Webex Teams room. The client is notified about the new message asynchronously by listening for new events via a web socket. When the client receives the message, it displays it for the user to see.

