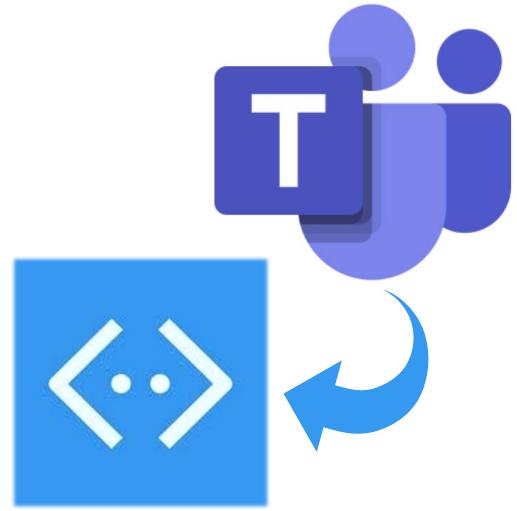
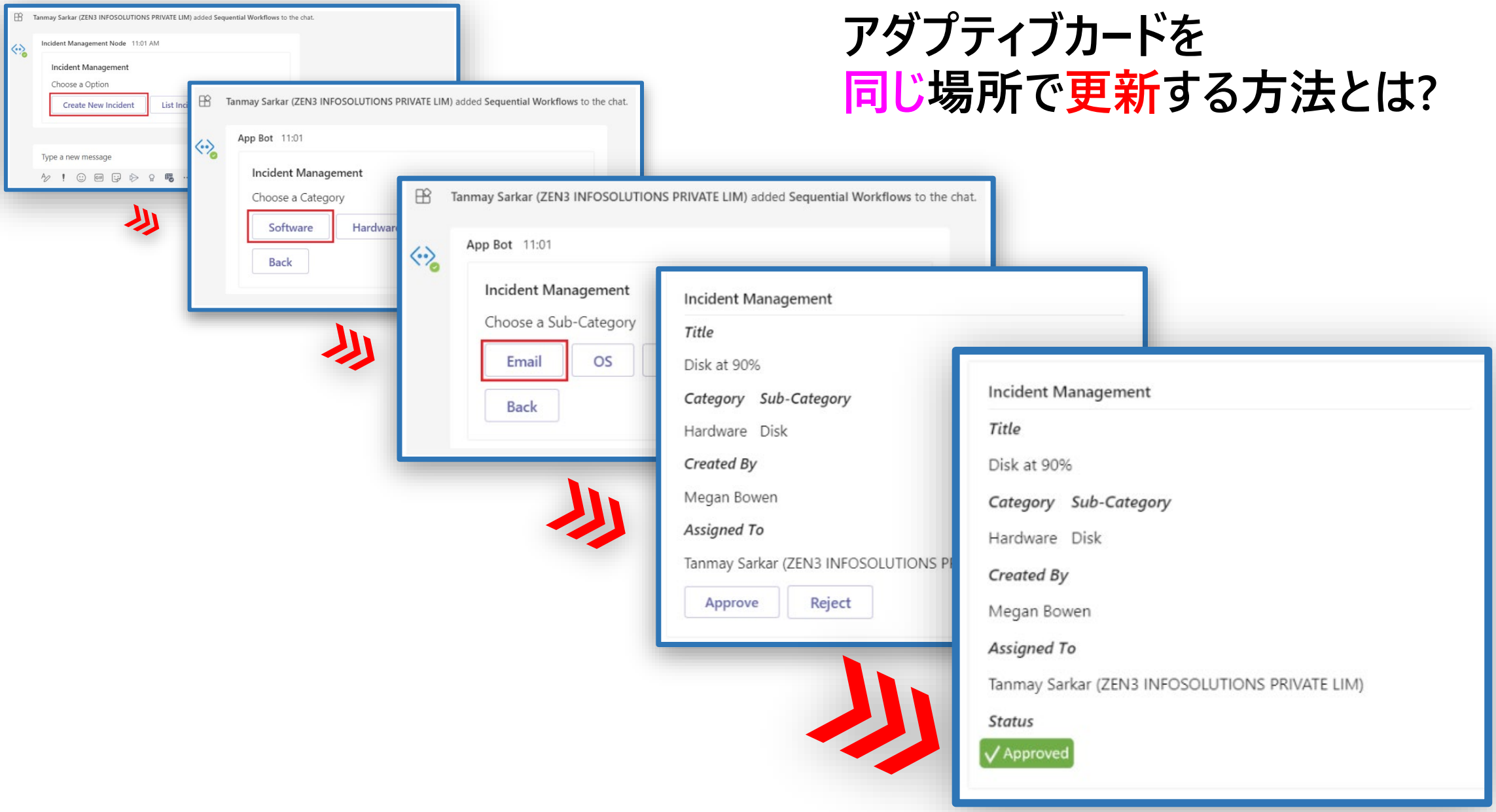


Teamsに表示した
アダプティブカードを更新する

アダプティブカードを
同じ場所で更新する方法とは?



コードサンプル:
Sequential workflow adaptive cards

[GitHub] OfficeDev/Microsoft-Teams-Samples: bot-sequential-flow-adaptive-cards

Sequential workflow adaptive cards

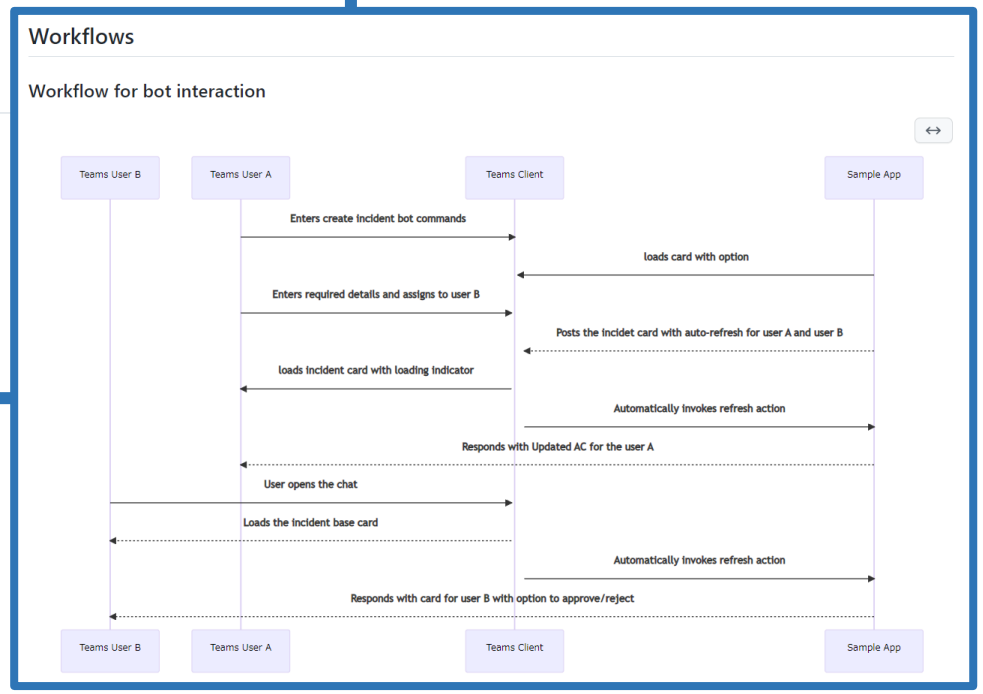
This App talks about the Teams Bot User Specific Views and Sequential Workflows in adaptive card with Node JS

This bot has been created using Bot Framework v4, it shows how to create a simple bot that accepts food order using Adaptive Cards V1.4

This is a sample app that provides an experience of managing incidents. This sample makes use of Teams platform capabilities like Universal Bots with below mentioned capabilities. User Specific Views Sequential Workflows Up to date cards

Key features

- Incident Creation
  - Choose Category
  - Choose Sub Category
  - Create Incident
  - Edit/ Approve/ Reject Incident
- List Incidents

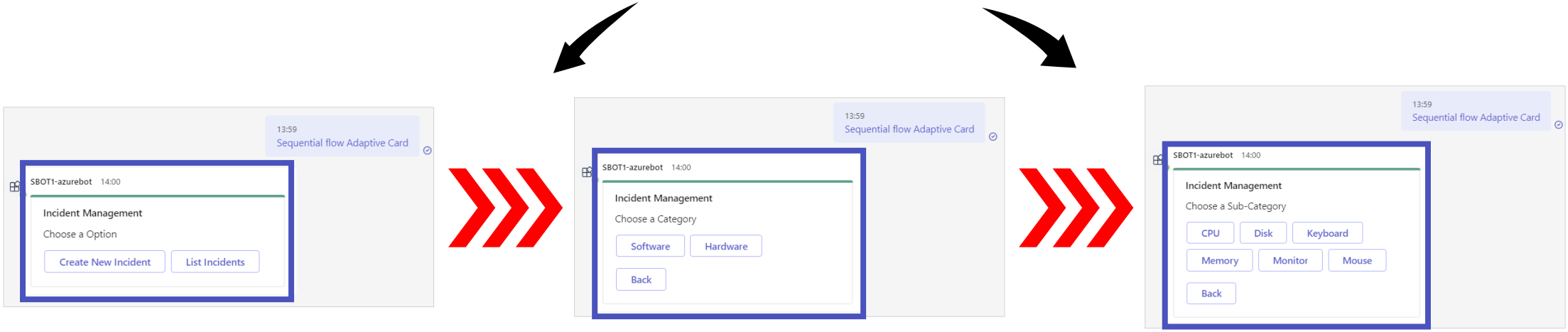


[結論]

同じactivityIdを

updateActivity (conversationId, activityId, activity)に渡すと

同じ場所で更新される



※ アダプティブカードをアクティビティのattachmentとして表示している

会話スレッド上でアクティビティを置き換えるメソッド

connectorClient.conversations.**updateActivity**(conversationId, activityId, activity)

activityId

1652511436743

activity オブジェクト

```
{
    "type": "message", [...]
    "attachments": [
        {
            "contentType": "application/vnd.microsoft.card.adaptive",
            "content": {
                "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
                "refresh": {
                    "action": {
                        "type": "Action.Execute",
                        "title": "Refresh",
                        "verb": "refresh_edit_status",
                        "data": {
                            "incident": {
                                "title": "CPU issue 1 ",
                                "category": "Hardware",
                                "subCategory": "CPU",
                                "assignedTo": {
                                    "id": "29:1aZ-91TBfggzndIVwgUf-zb4xYX9okot0LT3E [...]
                                }}},
                        "userIds": [
                            "29:1aZ-91TBfggzndIVwgUf-zb4xYX9okot0LT3 [...]
                        ]
                    },
                    "body": [ [...]
                    "type": "AdaptiveCard",
                    "version": "1.4"
                }}],
    "inputHint": "acceptingInput",
    "id": "1652511436743"
}
```

activityオブジェクトにも
activityId をコピーしておく

updateCardInTeams

[GitHub] OfficeDev/Microsoft-Teams-Samples: bot-sequential-flow-adaptive-cards

```
const updateCardInTeams = async (context, card) => {
    const serviceUrl = context.activity.serviceUrl;
    const credentials = new MicrosoftAppCredentials(process.env.MicrosoftAppId, process.env.MicrosoftAppPassword);
    const connectorClient = new ConnectorClient(credentials, { baseUri: serviceUrl });
    const conversationId = context.activity.conversation.id;
    const activityId = context.activity.replyToId;
    const replyActivity = MessageFactory.attachment(CardFactory.adaptiveCard(card));
    replyActivity.id = activityId;
    replyActivity.attachments = [CardFactory.adaptiveCard(card)];
    await connectorClient.conversations.updateActivity(conversationId, activityId, replyActivity);
};
```

server¥models¥adaptiveCard.js

selectResponseCard

[GitHub] OfficeDev/Microsoft-Teams-Samples: bot-sequential-flow-adaptive-cards

```
const selectResponseCard = async (context, user, members) => {
    allMembers = members;
    otherMembers = allMembers.filter(tm => tm.aadObjectId !== user.aadObjectId);
    const action = context.activity.value.action;
    const verb = action.verb;
    const prefix = verb.split('_')[0];

    if (verb && prefix === 'choose') {
        return await chooseCategory();
    } else if (verb && prefix === 'category') {
        return await chooseSubCategory(verb);
    } else if (verb && prefix === 'subcategory') {
        return createInc(verb, user);
    } else if (verb && prefix === 'edit') {
        return editInc(action);


    } else if (verb && prefix === 'refresh') {
        return await refreshInc(user, action);
    } else if (verb && prefix === 'list') {
        return await viewAllInc();
    } else {
        return optionInc();
    }
};
```

```
} else if (verb && prefix === 'save') {
    const card = await saveInc(action);
    await updateCardInTeams(context, card);
    return card;
```

```
} else if (verb && prefix === 'update') {
    const card = await updateInc(action);
    await updateCardInTeams(context, card);
    return card;
```

```
} else if (verb && prefix === 'status') {
    const card = await updateStatusInc(action);
    await updateCardInTeams(context, card);
    return card;
```

server¥models¥adaptiveCard.js

async onInvokeActivity(context)

[GitHub] OfficeDev/Microsoft-Teams-Samples: bot-sequential-flow-adaptive-cards

```
async onInvokeActivity(context) {
    console.log('Activity: ', context.activity.name);
    const user = context.activity.from;
    const action = context.activity.value.action;

～中略～

    if (context.activity.name === 'adaptiveCard/action') {
        const action = context.activity.value.action;
        console.log('Verb: ', action.verb);
        const allMembers = await (await TeamsInfo.getMembers(context)).filter(tm => tm.aadObjectId);
        const responseCard = await adaptiveCards.selectResponseCard(context, user, allMembers);
        return adaptiveCards.invokeResponse(responseCard);
    }
```

アクティビティがアダプティブカードの**action**ならば

selectResponseCardを実行する

server¥bot¥**botActivityHandler.js**

selectResponseCard

[GitHub] OfficeDev/Microsoft-Teams-Samples: bot-sequential-flow-adaptive-cards

```
const selectResponseCard = async (context, user, members) => {
    allMembers = members;
    otherMembers = allMembers.filter(tm => tm.aadObjectId !== user.aadOb
    const action = context.activity.value.action;
    const verb = action.verb;
    const prefix = verb.split('_')[0];

    if (verb && prefix === 'choose') {
        return await chooseCategory();
    } else if (verb && prefix === 'category') {
        return await chooseSubCategory(verb);
    } else if (verb && prefix === 'subcategory') {
        return createInc(verb, user);
    } else if (verb && prefix === 'edit') {
        return editInc(action);
```

prefixがsave、update、statusならば

```
} else if (verb && prefix === 'save') {
    const card = await saveInc(action);
    await updateCardInTeams(context, card);
    return card;
```

```
} else if (verb && prefix === 'update') {
    const card = await updateInc(action);
    await updateCardInTeams(context, card);
    return card;
```

```
} else if (verb && prefix === 'status') {
    const card = await updateStatusInc(action);
    await updateCardInTeams(context, card);
    return card;
```

updateCardInTeamsを実行する

```
};
```

server¥models¥adaptiveCard.js

updateCardInTeams

[GitHub] OfficeDev/Microsoft-Teams-Samples: bot-sequential-flow-adaptive-cards

```
const updateCardInTeams = async
    const serviceUrl = context.activity.serv
    const credentials = new MicrosoftAppCredentials
    const connectorClient = new ConnectorClient
    const conversationId = context.activity.conversa
    const activityId = context.activity.replyTo
    const replyActivity = MessageFactory.attachme
    replyActivity.id = activityId;
    replyActivity.attachments = [CardFactory.adaptiveCard(card)];
    await connectorClient.conversations.updateActivity(conversationId, activityId, replyActivity);
};
```

新しいactivityオブジェクトとともに、
前のアクティビティと

同じactivityIdを

server¥models¥adaptiveCard.js

updateActivity (conversationId, activityId, activity)に渡す

updateCardInTeams

[GitHub] OfficeDev/Microsoft-Teams-Samples: bot-sequential-flow-adaptive-cards

```
const updateCardInTeams = async
    const serviceUrl = context.activity.serv
    const credentials = new MicrosoftAppCredentials(process.env.MicrosoftAppId, process.env.MicrosoftAppPassword);
    const connectorClient = new ConnectorClient(credentials, { baseUri: serviceUrl });
    const conversationId = context.activity.conversation.id;
    const activityId = context.activity.replyToId;
    const replyActivity = MessageFactory.attachment(CardFactory.adaptiveCard(card));
    replyActivity.id = activityId;
    replyActivity.attachments = [CardFactory.adaptiveCard(card)];
    await connectorClient.conversations.updateActivity(conversationId, activityId, replyActivity);
};
```

新しいactivityオブジェクトとともに、前のアクティビティと

同じactivityIdを

server¥models¥adaptiveCard.js

updateActivity (conversationId, activityId, activity)に渡す

[結論]

同じactivityIdを

updateActivity (conversationId, activityId, activity)に渡すと

同じ場所で更新される

※ アダプティブカードをアクティビティのattachmentとして表示している

アクティビティ(ボットが受け取るイベント)を
後から操作できる

↓

ボットからの応答として表示された
アダプティブカードも後から変更できる

「Sequencial workflow adaptive cards」で実現されている動作

Up-to-date cards（アップトゥデイト カード）

アダプティブカードをそのままの場所で更新する

Contextual views（コンテクスチュアル ビュー）

参加者の役割に応じてアダプティブカードを切り替えて表示する

Sequential workflow（シーケンシャル ワークフロー）

参加者が並行して、待機・追尾しながらワークフローを進めていく