

# Initiate sequential workflow

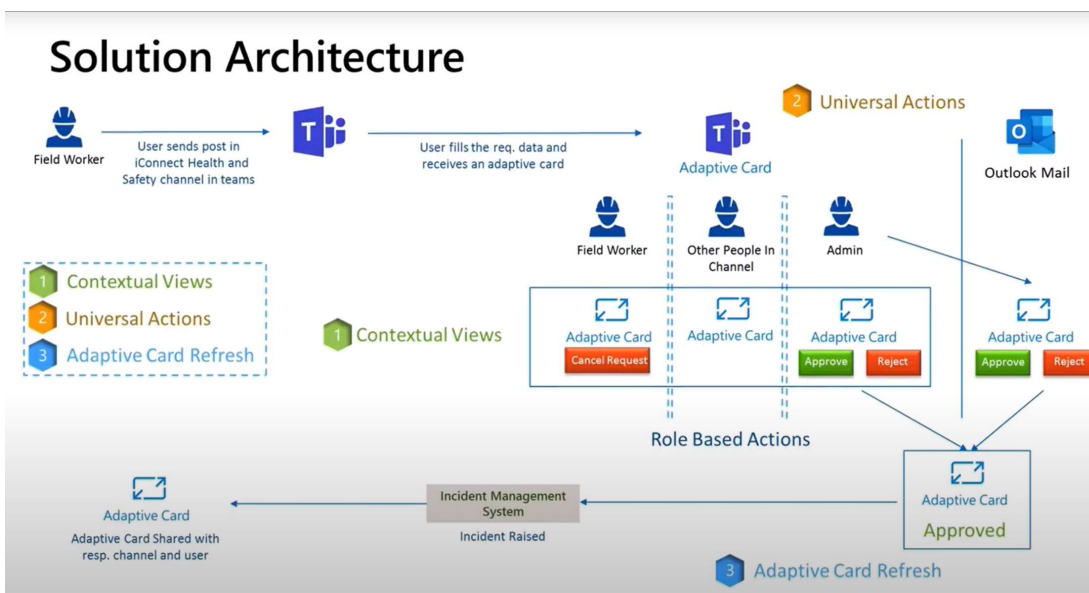
## Overview

Today corporate employees interact with multiple systems to complete day to day tasks. For example, employees are using a vacation system to submit time-off request for manager approval; using a expense report system to create a reimbursement ticket; using a company travel portal to book a flight for a business trip, etc.

In those scenarios, we assume there's already such a system (web services) up and running and our customers wish to integrate that system with Teams.

## Typical Example

A LOB Example: A **Frontline Incident Reporting** that allows front line worker to log real time safety incidents with rich information like locations and photos, and allow safety expert and admins to allow or reject an incident. The solution can be illustrated as below:



Another example from Bosch to allow frontline worker to submit a repair request for a broken machine from Teams: frontline worker to take a picture of broken machine parts, from Teams app, then it looks up the part information and frontline worker submit it

for procurement. Procurement team received request from Teams app notification to either approve it or reject it for more comments.

There are also many examples built by ISVs such as Polly that allows user to start a pull inside conversations.

## **Target Customer**

We target both enterprise LOB developers who are modernizing the digital experience for employees and ISV developers who are customizing their products for Teams.

## **Potential Solutions**

There are multiple approaches for developers to deliver such scenario to the end-user:

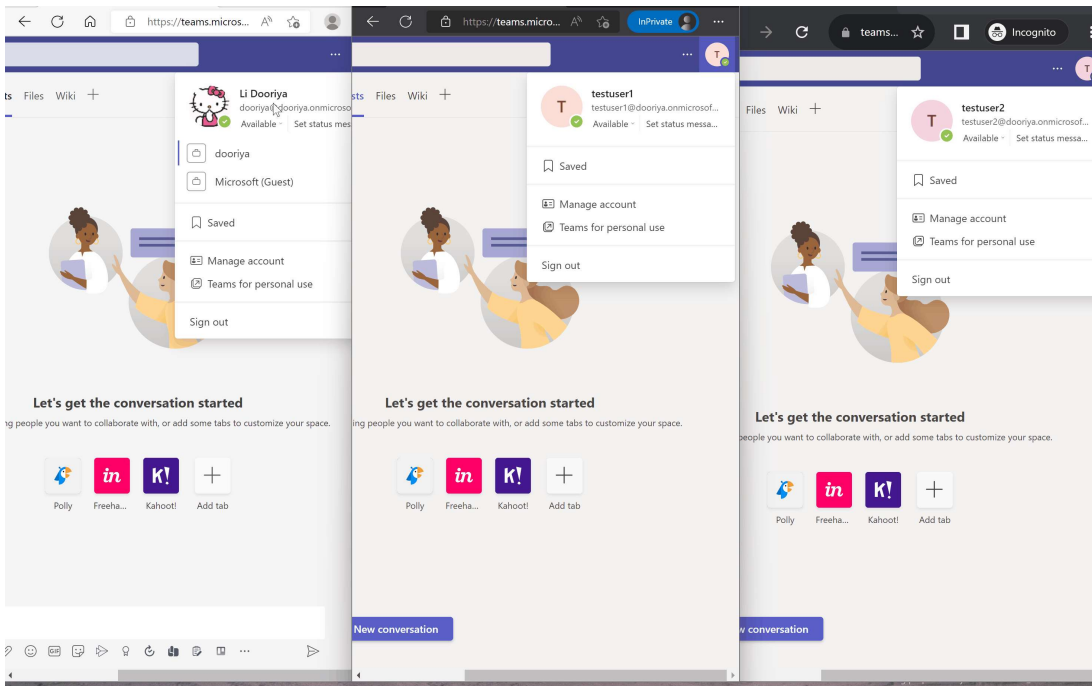
### **Option 1: A UI-based tab experience**

We do not prioritize this because a tab experience will be similar to a web service experience and doesn't showcase the platform differentiation except a tab app can leverage Teams SS0 plus less context switch.

### **Option 2: A command bot and adaptive card with universal actions**

Developer can also build a command bot to initiate the workflow inside a conversation, the follow up actions can be taken from the adaptive card response.

Example:



### Option 3: Action-based message extension and adaptive card in task module

This brings the benefit of triggering the action from multiple locations and task module permits multiple type of content such as `<iframe>` and custom HTML/JavaScript code, as well as adaptive card. Also it doesn't always require the workflow to be initiated from a conversation.

Solution comparison and our recommendation: WIP

### Key Definitions

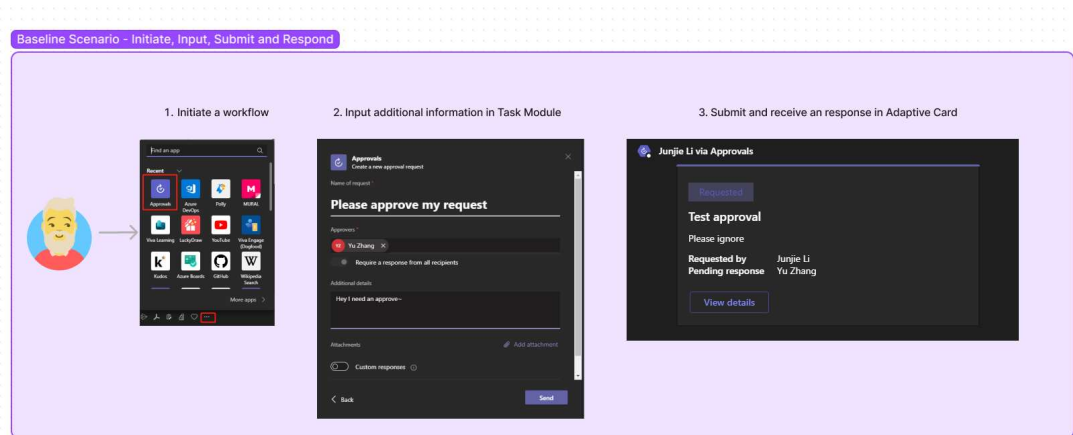
Term	Definition
Sequential Workflow	Refers to a scenario. A complex task users initiate from a bot or a message extension. A modal popup will be presented to users to collect additional information and will be submitted into an external system via api call. And users can also take subsequent actions after initiation e.g in adaptive card.
Action-based message extension	Refers to a solution. A "bot" style Teams application that allows user to trigger an interactive command and takes the request to the external system.

Term	Definition
Task Module	It's essentially a "tab" in a pop up form. It allows several options to render the layout, 1) a static list of parameters defined in manifest which Teams client will render; 2) custom HTML / JavaScript code 3) <code>&lt;iframe&gt;</code> for video etc, 4) Adaptive Card.
Adaptive card with universal actions	Adaptive card with universal action enhances user experience with user specific view, sequential workflow support etc. Reference <a href="#">here</a>

// Below is based on solution #3, maybe changed if we decide to go with solution #2

## E2E Developers Job To Be Done

### Baseline Scenario: Initiate, Input, Submit and Respond



In most typical sequential workflow scenarios, multiple steps can be chained to achieve a complex workflow. The baseline starts with a user initiation, a modal popup to collect additional information with a form style and an action to submit information to an external system, and send the result into conversation.

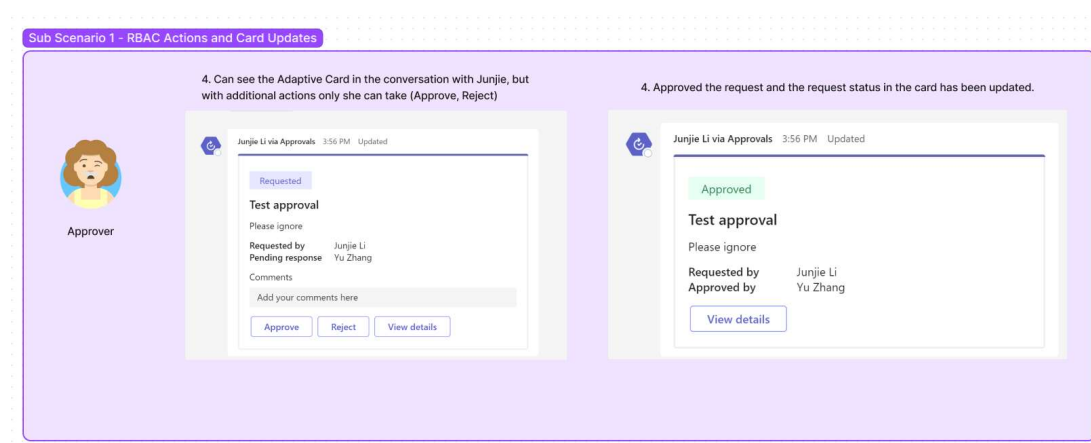
As a developer, following tasks are the JTBDs to achieve this scenario:

1. Define how the users can initiate the workflow by **defining an invoke location**. The workflow can be triggered by a command or from a chat message, or from a compose message area.
2. Obtain user identity in the context of Teams (e.g., single sign on) to log user into external system or consent permissions for the

Teams application perform actions on behalf of user.

3. Define the adaptive card in the task module. Here we opinionated to use adaptive card to render the task module.
4. Respond to the actions (universal actions) triggered in the adaptive card.
5. Define how the final message will be sent: 1) The message extension send an adaptive card to the conversation as a summary after user submit an action. 2) The message extension can insert an adaptive card in compose message box and let user to send it.

## Sub Scenario 1: User Specific View Actions and Card Updates

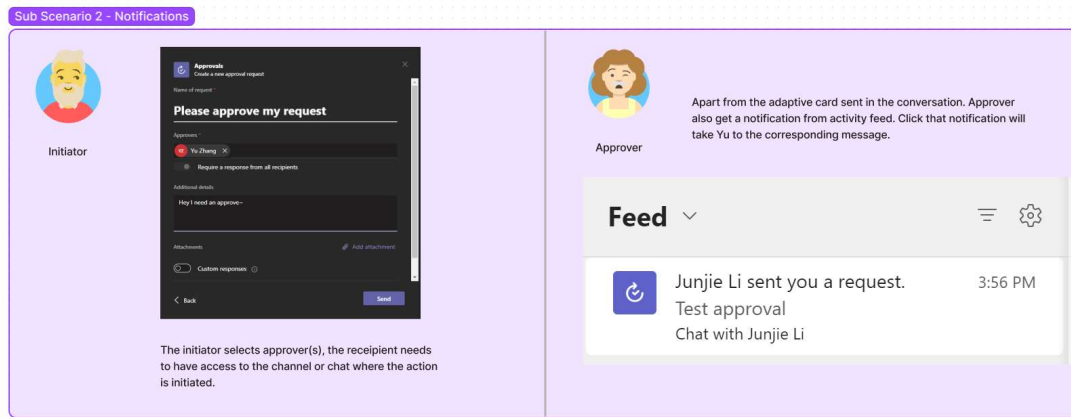


Most workflows involve multiple users and several follow-ups to complete. Taking the approval workflow as an example for follow-up actions: the initiator - can cancel the approval request and the approver - can approve or reject the request. Those actions are presented to different users with users with **user specific view** in adaptive cards.

As a developer, following tasks are the JTBDs to achieve this scenario:

1. Define actions in the Adaptive Card.
2. Define different views/actions for different users.
3. Respond to the actions in adaptive card with a card content refresh (Reference: Adaptive Card universal actions **sequential workflow support**)

## Sub Scenario 2: Notifications in Workflow



When a workflow is initiated, there are many scenarios when notification is involved. For example, creating an approval request and notify the approver; creating a bug in issue tracking system and notify the assignee. There are several ways to achieve this scenario:

1. The workflow needs to be triggered where all participants are present and @mention the user.
2. Include a notification bot and from the action submit to trigger a notification to the user specified during the workflow.
3. Send a notification in the activity feed and include a link to the message.

### Additional Scenarios

1. From my action-based message extension, send an actionable message (adaptive card with universal actions) in email.
2. Copy and paste an adaptive card response and send it in an email (Adaptive Card based loop component).

Additional scenarios are for our team to explore and do experiment on.

## Scope

Scenario Building Blocks:

- (Template) An action-based message extension template for baseline scenario
- (SDK) Programming model simplification with bot framework sdk
- (SDK) SSO authentication

- (SDK) Universal action in adaptive card
- (UI) Adaptive card in task module
- (Docs) Tutorials / Samples: in tutorials & samples, we need to cover:
  - How do I choose between command bot and action-based message extension?
  - How do I chain multiple forms (multi-step) in a Task Module?
  - How do I embed / custom HTML / JavaScript code in Task Module?
  - How do I add more actions and response in adaptive card?
  - How do I extend to notification bot: from my action bot, I can send a notification to a specific user.
  - How do I choose between command bot & action-based message extension? (command bot example: `@polly question? option 1, option 2`)

#### Teams Toolkit Feature Scope

- Scaffold a JavaScript / TypeScript scenario template in Teams Toolkit for Visual Studio Code & CLI.
- Scaffold a C# scenario template in Teams Toolkit for Visual Studio.
- Applicable Teams Toolkit lifecycle support: Create, Debug, Provision, Deploy, Validate, Publish.
- Applicable Teams Toolkit additional features: Add SSO, Add API connection, Add CICD workflow, Add Tab(s).

## Developer Experiences in Teams Toolkit

Developer experiences in Teams Toolkit for this scenarios are mainly delivered through scaffolding templates, SDKs and samples. Here is an example of scaffolding flow in Teams Toolkit CLI to formulate discussion. This also demonstrates how several building blocks can be stitched together to form a complete scenario.

```
teamsfx new
? Create a new teams app from
[x] Create a new teams app
[ ] Start from a sample

? Select an app capability
[ ] Search-based message extension
[x] Action-based message extension
[ ] Notification bot
```

```
? Select where your action command must be invoked?
```

```
[x] From a command box
```

```
[ ] From a compose box
```

```
[ ] From a message
```

```
? Select a programming language
```

```
[ ] TypeScript
```

```
[ ] JavaScript
```

```
[x] C#
```

```
// Add SSO support for action-based ME
```

```
teamsfx add sso
```

```
// Call an API from action-based ME
```

```
teamsfx add api-connection
```

## Features

- Epic 14812636: Enable commands to send notifications to other users
  - Feature 14813311: [App Scenario] Support notify others for example via Graph API (incident management)
- Epic 14812651: Create a sequential approval workflow between users
  - Feature 14677809: [App Scenario] Add SSO to a command
  - Feature 14729797: [App Scenario] Respond to action command with task module using SDK simplification
  - Feature 14729813: [App Scenario] Respond to submit actions in adaptive card (Adaptive Card Universal Actions) / Task Module