# Developing Customer Experience Integrations

Rue Green
Sr. Solutions Engineer, Full Stack Software Developer

Session ID: BRKCCT-2556

# Cisco Webex App

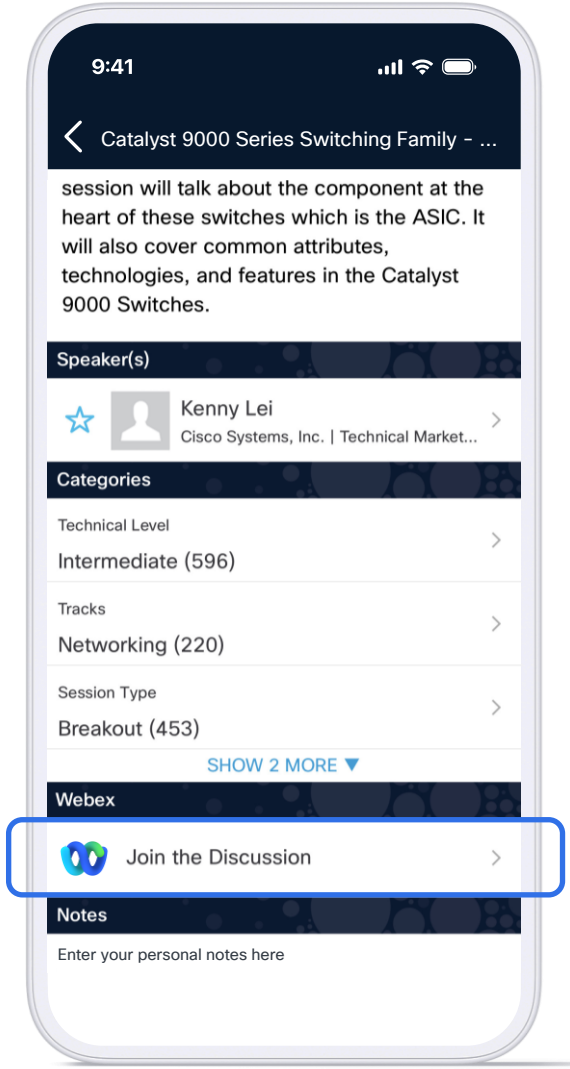
## Questions?

Use Cisco Webex App to chat
with the speaker after the session

## How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

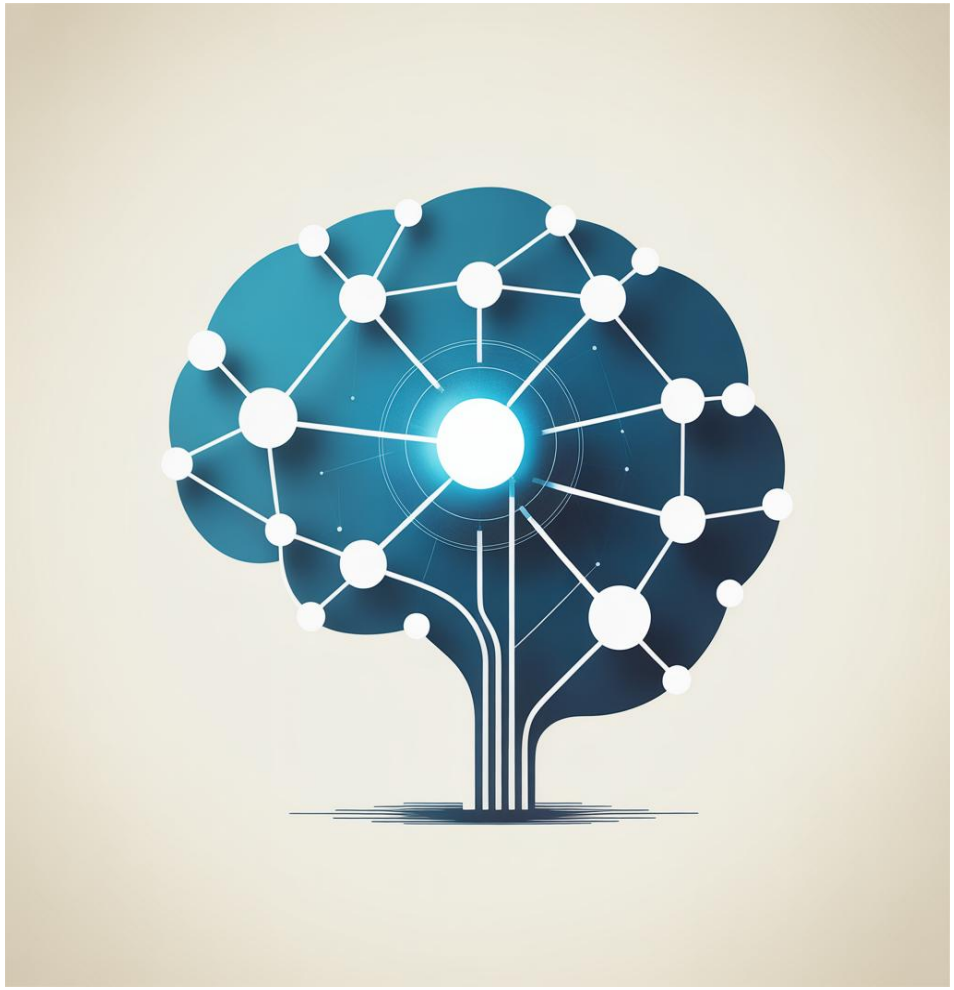
**Webex spaces will be moderated by the speaker until 14 November 2025.**




https://ciscolive.ciscoevents.com/
ciscolivebot/**#BRKCCT-2556**

© 2025 Cisco and/or its affiliates. All rights reserved.

# Agenda



01 REST / OAuth Primer

02 Custom Nodes in WxConnect

03 Custom Node Use Cases

04 A2A / MCP Primer

05 Roadmap for A2A / MCP

© 2025 Cisco and/or its affiliates. All rights reserved.

# About the Presenter – Rue Green

- Sr. Solutions Engineer, Full Stack Software Developer
- Quad CCIE #9269
  - (Enterprise, Collaboration, Voice, Automation)
  - Author of Unified CVP Cisco Press book (2011)
- 20+ years CC Experience
  - WxCC, WxCCE, CPaaS
- 30+ years in software development
  - Certified Full Stack Developer (MERN, MEAN)
- 20 years at Cisco
- Professional Beach Volleyball Player and Coach

© 2025 Cisco and/or its affiliates. All rights reserved.

# REST and OAuth Primer

A Quick Introduction to Modern API Architecture and Security

# REST Fundamentals

**REST (Representational State Transfer)** is an architectural style for building APIs that leverages standard HTTP methods to create predictable, scalable web services. It's the foundation of most modern web APIs, enabling clean separation between client and server.

### GET

Retrieve data from the server without modifying anything

```
GET /api/users/123
```

### POST

Create new resources on the server

```
POST /api/users
```

### PUT/PATCH

Update existing resources with new data

```
PUT /api/users/123
```

### DELETE

Remove resources from the server

```
DELETE /api/users/123
```

### Core Principles

- **Stateless:** Each request contains all necessary information; the server doesn't maintain session state
- **Standard Status Codes:** Uses HTTP codes like 200 OK, 404 Not Found, 401 Unauthorized for clear communication
- **Resource-Based URLs:** Endpoints use nouns representing resources, not verbs describing actions
- **Client-Server Separation:** Clear boundaries enable independent evolution of frontend and backend

© 2025 Cisco and/or its affiliates. All rights reserved.

# What is OAuth?

**OAuth 2.0** is an industry-standard authorization framework that allows applications to securely access user resources without requiring users to share their passwords. It's the technology behind "Sign in with Google," "Connect with Facebook," and similar delegated authorization patterns.

### The Problem OAuth Solves

Imagine you want to let a photo printing app access your Google Photos. Without OAuth, you'd have to give them your Google password — risky and uncontrollable. OAuth provides a secure alternative: grant limited, revocable access without exposing credentials.

## Real-World Scenario

01

**You want to print photos from Google Photos using PrintMaster app**

02

**Instead of sharing your Google password, OAuth enables secure delegation**

03

**You control exactly what PrintMaster can access and can revoke access anytime**

04

**PrintMaster never sees your password and Google maintains security**

© 2025 Cisco and/or its affiliates. All rights reserved.

# The OAuth Authorization Flow

The OAuth authorization code flow is the most secure method for web applications. It ensures that sensitive credentials never pass through the browser and provides a robust mechanism for delegated authorization.

1. **User Initiates Login**

   Application redirects user to the authorization server's login page with client ID, requested scopes, and a redirect URI.

2. **Authentication & Consent**

   User authenticates with their credentials and reviews the permissions the application is requesting. User decides whether to grant access.

3. **Authorization Code Issued**

   After user approval, authorization server redirects back to the application with a short-lived authorization code in the URL.

4. **Token Exchange**

   Application's backend exchanges the authorization code for an access token and refresh token by authenticating with client secret.

5. **Access Granted**

   Application uses the access token to make API calls on behalf of the user, accessing only the resources within the granted scopes.

© 2025 Cisco and/or its affiliates. All rights reserved.

# Critical OAuth Values and Tokens

OAuth relies on several key values that flow through the authorization process. Each serves a specific security purpose and has different lifespans and usage patterns.

**Authorization Code**

Short-lived, single-use code exchanged for tokens. Expires in minutes.

**Refresh Token**

Long-lived credential to obtain new access tokens without re-authentication.

**Access Token**

Credential for API access. Typically valid for minutes to hours.

## Understanding Each Component

### Access Token

The credential sent with every API request to prove authorization. Included in the Authorization header as a Bearer token:

```
Authorization: Bearer eyJhbGc...
```

Short lifespan limits damage if compromised.

### Refresh Token

Allows obtaining new access tokens without requiring the user to log in again. Must be stored securely on the backend, never in browser storage. Treat refresh tokens like passwords.

### Client ID & Secret

**Client ID:** Public identifier for your application, safe to expose.

**Client Secret:** Sensitive password proving your app's identity. Must remain on backend only, never in frontend code or version control.

## Scopes: Fine-Grained Permissions

Scopes define the specific permissions your application requests. Examples include `read:user`, `write:files`, or `admin:org`. Always follow the principle of least privilege — request only the minimum scopes necessary for your application's functionality.

© 2025 Cisco and/or its affiliates. All rights reserved.



cisco

# REST + OAuth in Action

When REST and OAuth work together, you get secure, scalable API access. REST provides the structure for API calls, while OAuth ensures only authorized users can access protected resources. Here's how they integrate in practice.

**Your Application**

Client application needs to access user data via REST API

**Authenticated Request**

Makes REST API call with Bearer token in Authorization header

**API Server Validation**

Validates token signature, expiration, and checks user permissions

**JSON Response**

Returns requested data in standard REST format with appropriate status code

**Example API Call**

```
GET /api/user/profile HTTP/1.1Host: api.example.comAuthorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...Accept: application/json
```

**The Complete Picture**

**REST defines HOW** to structure your API calls using standard HTTP methods and resource-based URLs.

**OAuth defines HOW** to secure access to those APIs using token-based authorization instead of sharing passwords.

Together, they enable the secure, scalable, modern applications that power today's web.

© 2025 Cisco and/or its affiliates. All rights reserved.



cisco

# Key Takeaways

### REST Provides Structure

Architectural style using standard HTTP methods, stateless communication, and resource-based URLs. Foundation for building scalable, maintainable APIs.

### OAuth Provides Security

Authorization framework enabling delegated access through tokens instead of passwords. Flow: Code → Access Token + Refresh Token, controlled by scopes.

### Together They Power Modern APIs

REST and OAuth combine to create the secure, scalable architecture behind every major platform: Google, Facebook, GitHub, Stripe, and thousands more.

---

**Every modern API you interact with relies on these two technologies working in harmony to deliver functionality and security.**

Understanding REST and OAuth isn't just theoretical knowledge — it's essential for building any application that integrates with external services or exposes its own API to partners and developers.

© 2025 Cisco and/or its affiliates. All rights reserved.

# Custom Node Creation

Building Reusable Integrations in Webex Connect

# What Are Custom Nodes?

### Definition

Custom Nodes allow you to create reusable integrations with third-party REST or SOAP APIs that can be used across multiple flows in Webex Connect.

### Key Benefits

- Reusability across flows
- Centralized configuration
- Advanced authentication support
- Rate limiting controls

### When to Use

- Need reusable integration nodes
- Require OAuth 2.0 authentication
- Complex API requirements
- Multiple flows use same API

### Availability

Custom Node integration is an add-on feature available in select Webex Connect tiers. Contact your account manager to enable it.

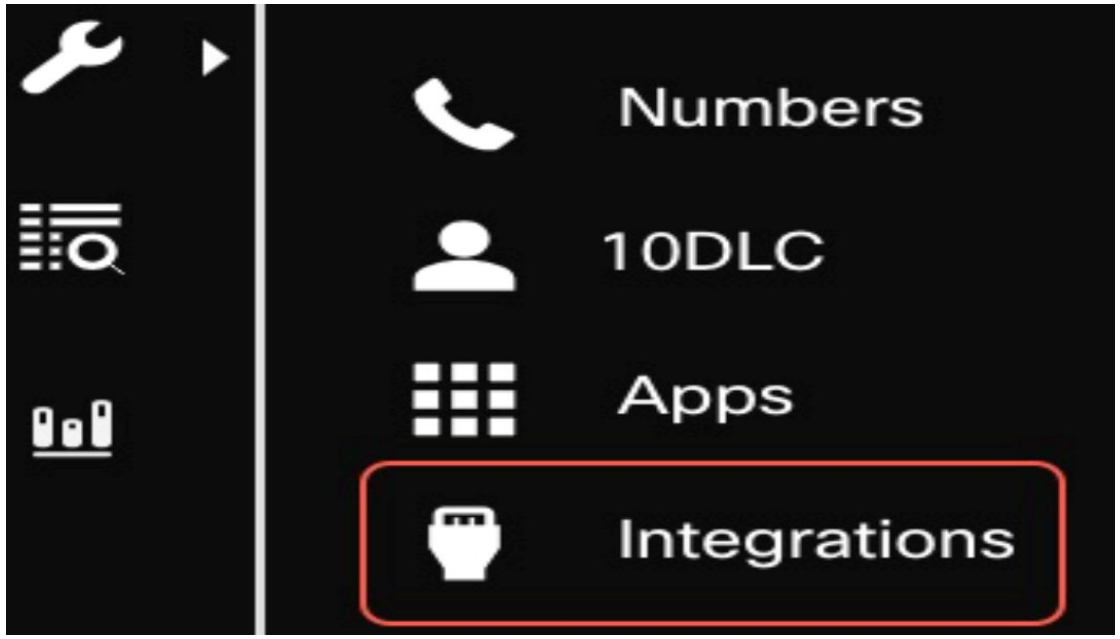
© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 1: Navigate to Integrations

**Navigation Path**

**Assets → Integrations**

This will take you to the Integrations screen where you can view all existing inbound/outbound integrations and create new custom nodes.




© 2025 Cisco and/or its affiliates. All rights reserved.

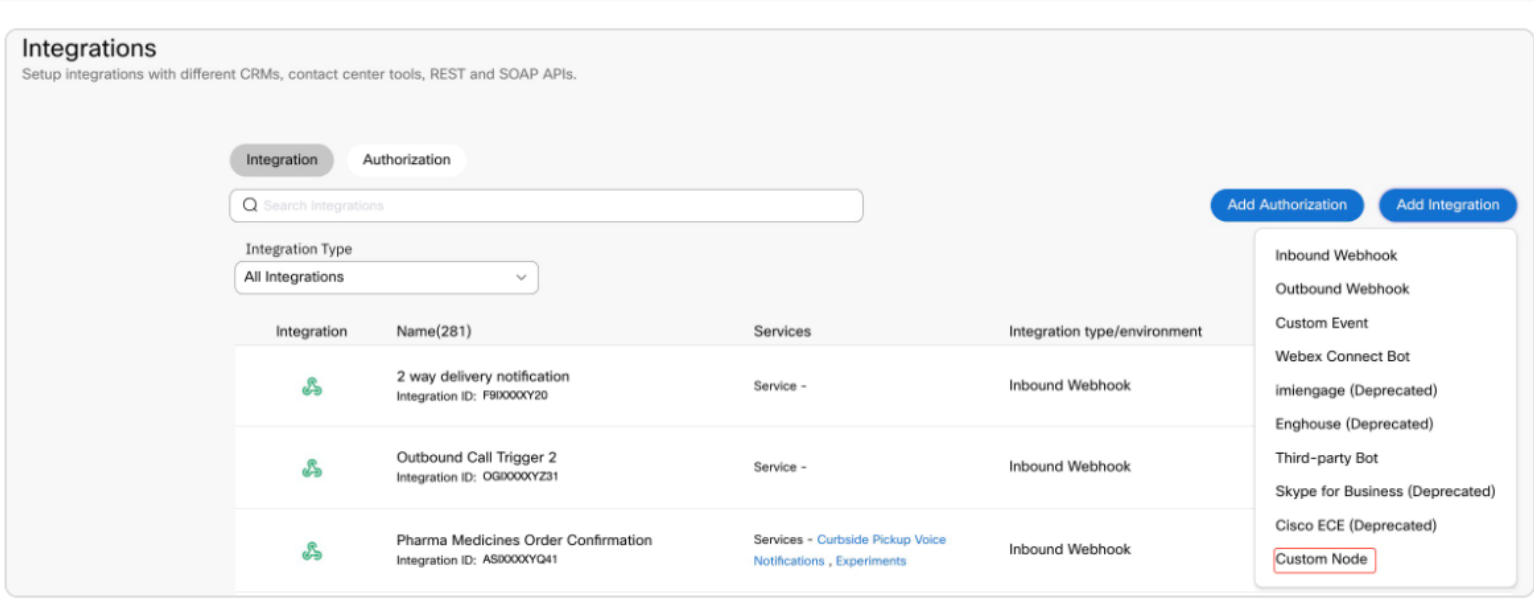
# Step 2: Add a Custom Node

Click ADD INTEGRATION and select CUSTOM NODE

**What Happens Next?**

A "Create new custom node" pop-up window will appear where you can choose between REST API and SOAP API integration types.

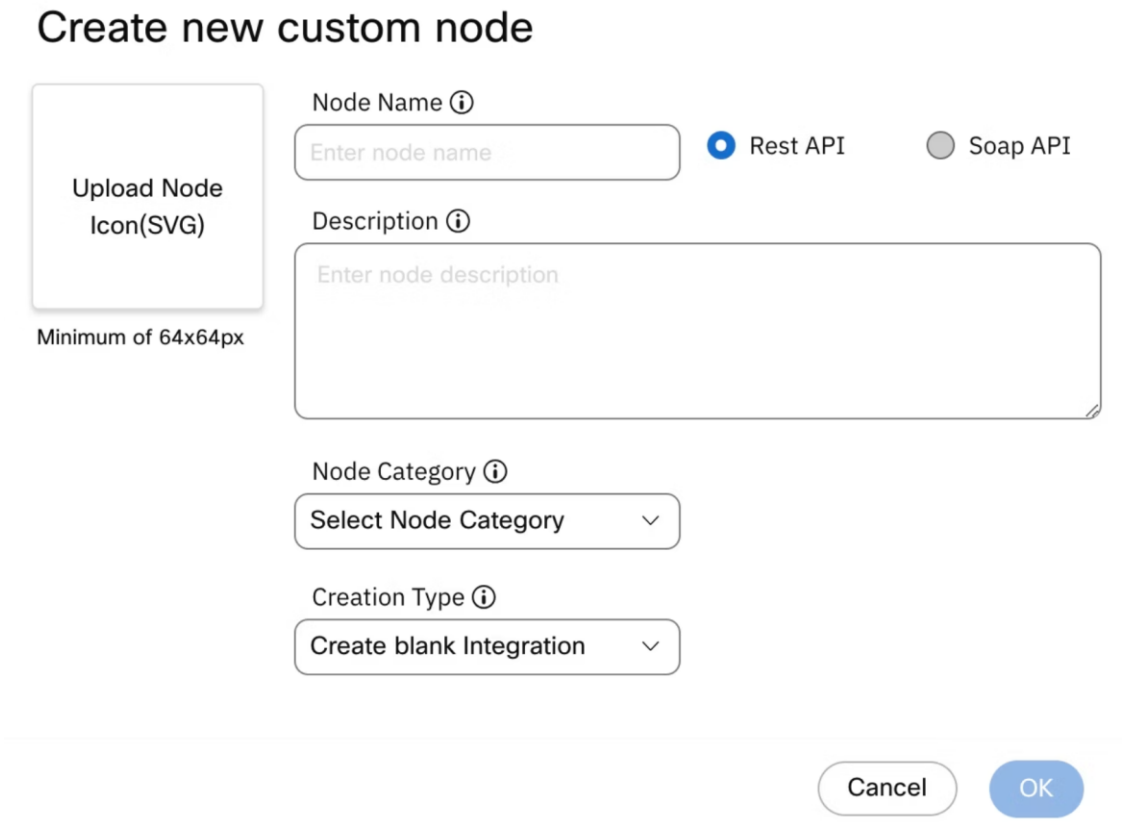
Click ADD INTEGRATION and select CUSTOM NODE




*Select Custom Node from the Add Integration dropdown*

© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 3: Configure Basic Node Settings




*Create new custom node configuration dialog*

## Required Fields

01

### Node Name

Provide a user-friendly name for your integration

02

### Description

Brief description of what the node does

03

### SVG Icon

Upload a relevant SVG image for visual identification

04

### API Type

Choose REST API or SOAP API

© 2025 Cisco and/or its affiliates. All rights reserved.

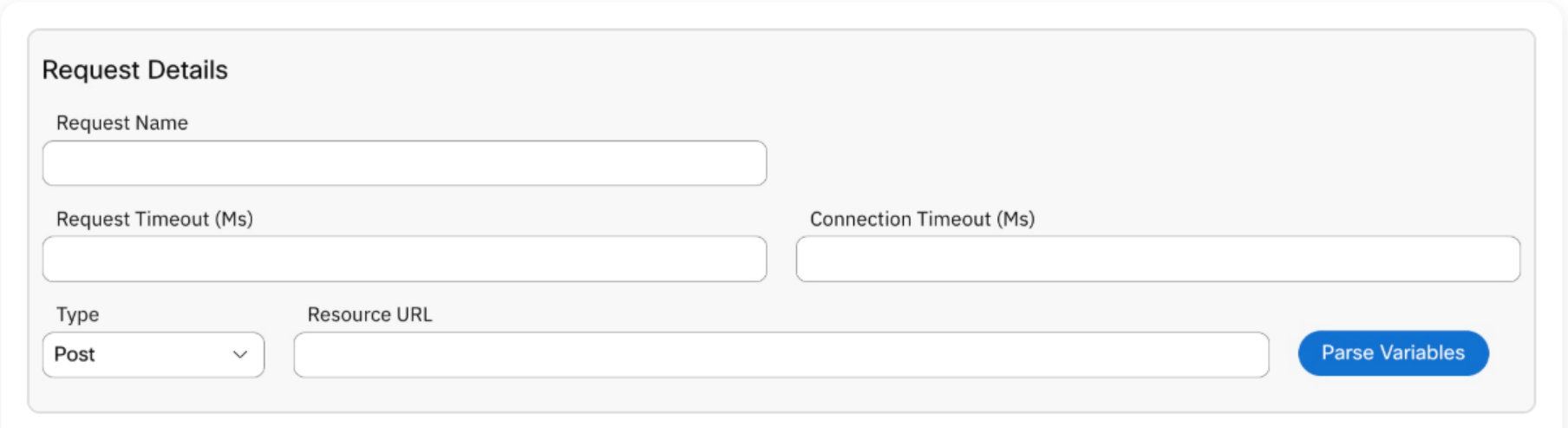
# Step 4: Configure Request Details

## Basic Settings

- **Request Name:** Descriptive method name
- **HTTP Type:** GET, POST, PUT, DELETE
- **Resource URL:** API endpoint URL

## Parameters

- **URL Parameters:** Query string values
- **Parameter Type:** Static or Dynamic
- **Field Values:** Variable references




© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 5: Configure Authentication

*Example: Basic Authentication configuration*

### 🔑 Basic Auth

Configure username and password fields with static or dynamic values

### 🎫 API Key

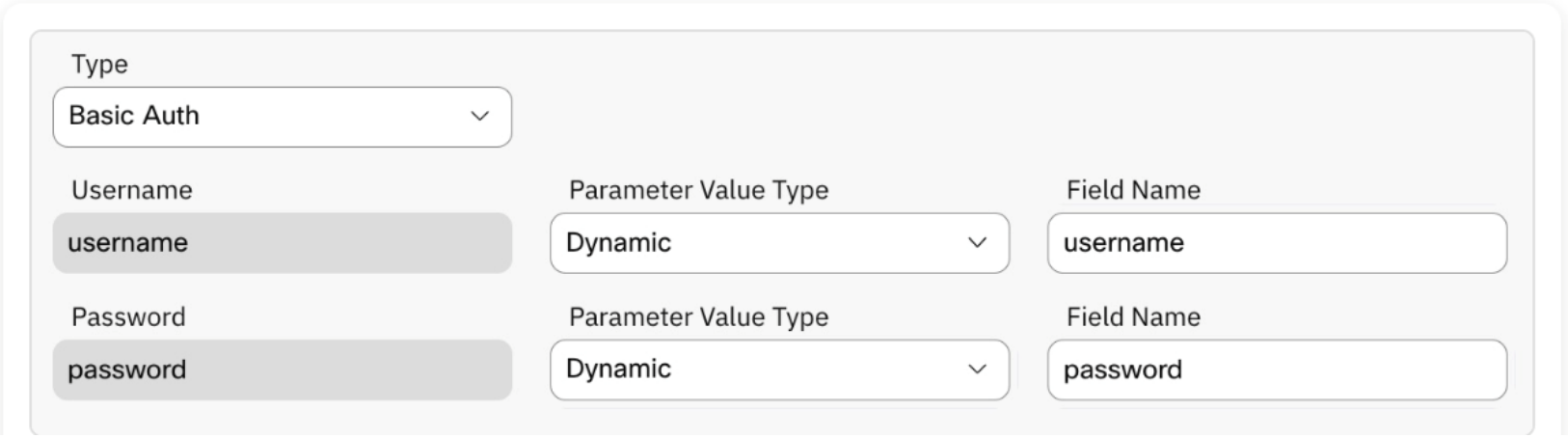
Simple token-based authentication in headers

### 🔐 OAuth 2.0

Authorization Code or Client Credentials grant types

### ☁️ AWS Signature

HMAC-based authentication for AWS services

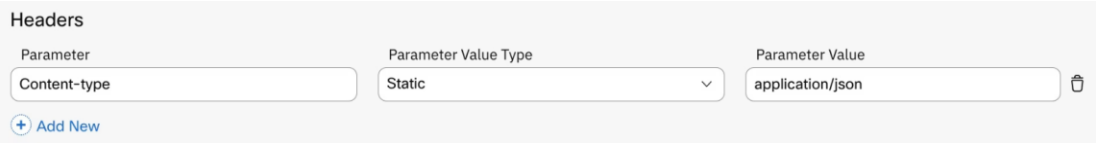



© 2025 Cisco and/or its affiliates. All rights reserved.



cisco

# Step 6: Configure Headers




*Setting up request headers*

## Common Headers

- **Content-Type:** application/json, text/xml
- **Authorization:** Bearer tokens, API keys
- **Accept:** Expected response format
- **Custom Headers:** API-specific requirements

**💡 Pro Tip**

Use **Static** for fixed values and **Dynamic** for runtime values

© 2025 Cisco and/or its affiliates. All rights reserved.

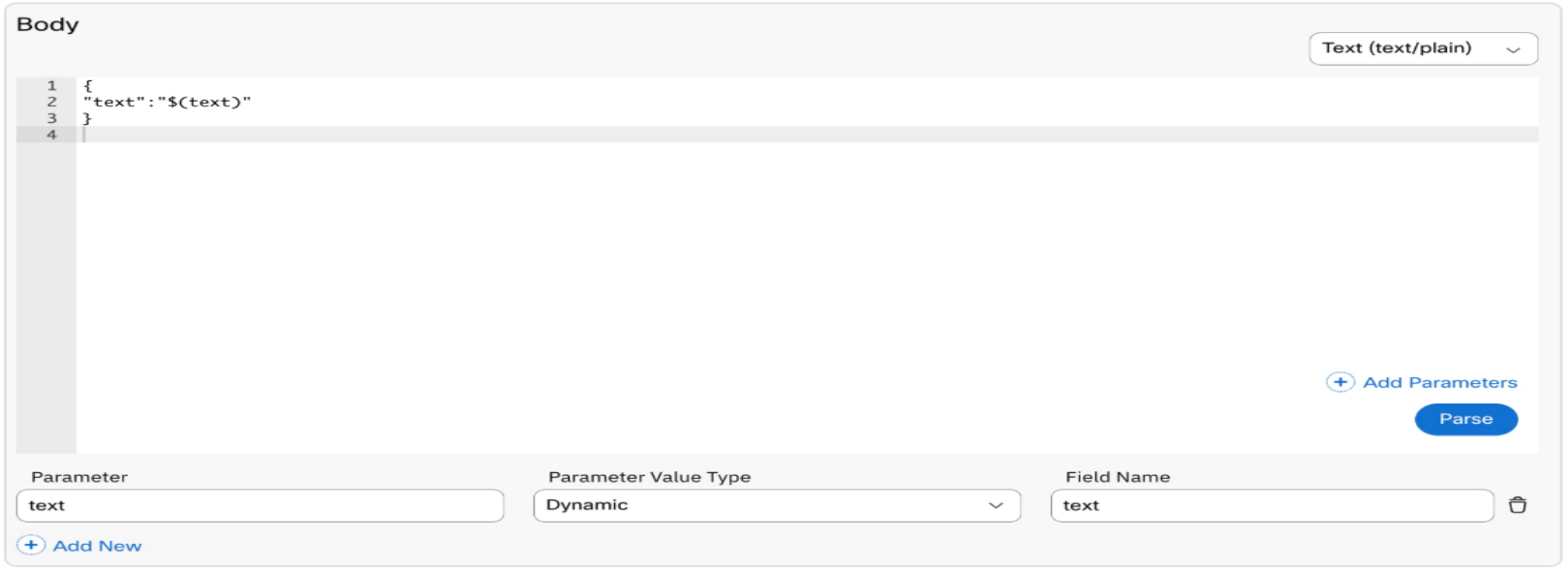
# Step 7: Configure Request Body

*Setting up JSON request body with dynamic parameters*

## Body Configuration Options

```
{  "text": "$(text)",  "version": "$(version)"}
```

Use **$(variableName)** syntax to reference dynamic parameters provided at runtime. Mark variables as mandatory or optional in Node UI.




*Setting up JSON request body with dynamic parameters*

© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 8: Configure Response Handling

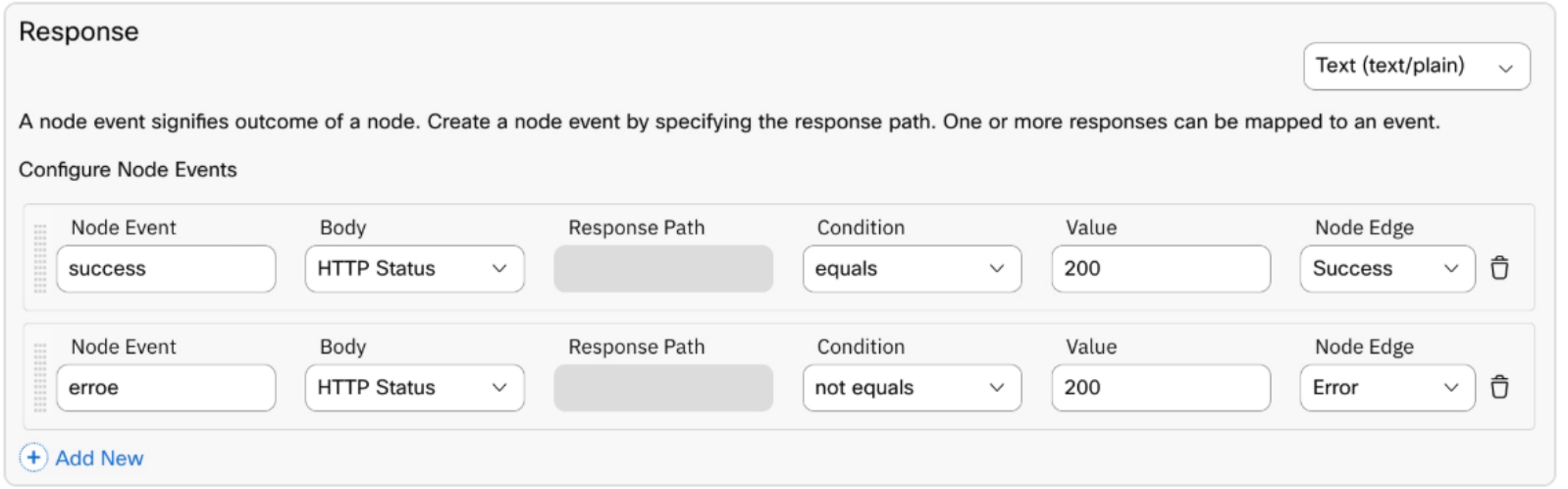
*Configuring node events and success/error conditions*

## Node Events

- **Success Event:** HTTP Status = 200
- **Error Event:** HTTP Status ≠ 200
- Map to Success/Error node edges

## Response Parameters

- Extract values using JSON paths
- Example: $.data.user.email
- Available as output variables




*Configuring node events and success/error conditions*

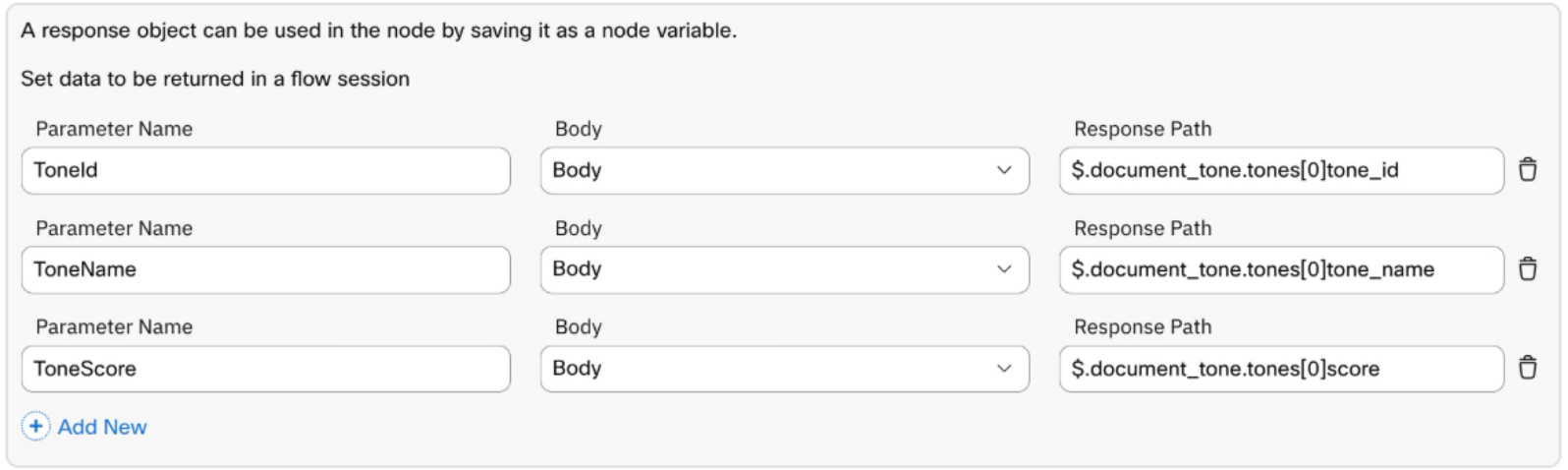
© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 9: Map Response Parameters

*Extracting specific values from API response using JSON paths*

## JSON Path Examples

```
Parameter: ToneIdSource: BodyResponse Path: $.document_tone.tones[0].tone_idParameter:
ToneNameSource: BodyResponse Path: $.document_tone.tones[0].tone_nameParameter: ToneScoreSource:
BodyResponse Path: $.document_tone.tones[0].score
```




*Extracting specific values from API response using JSON paths*

© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 10: Configure Node UI

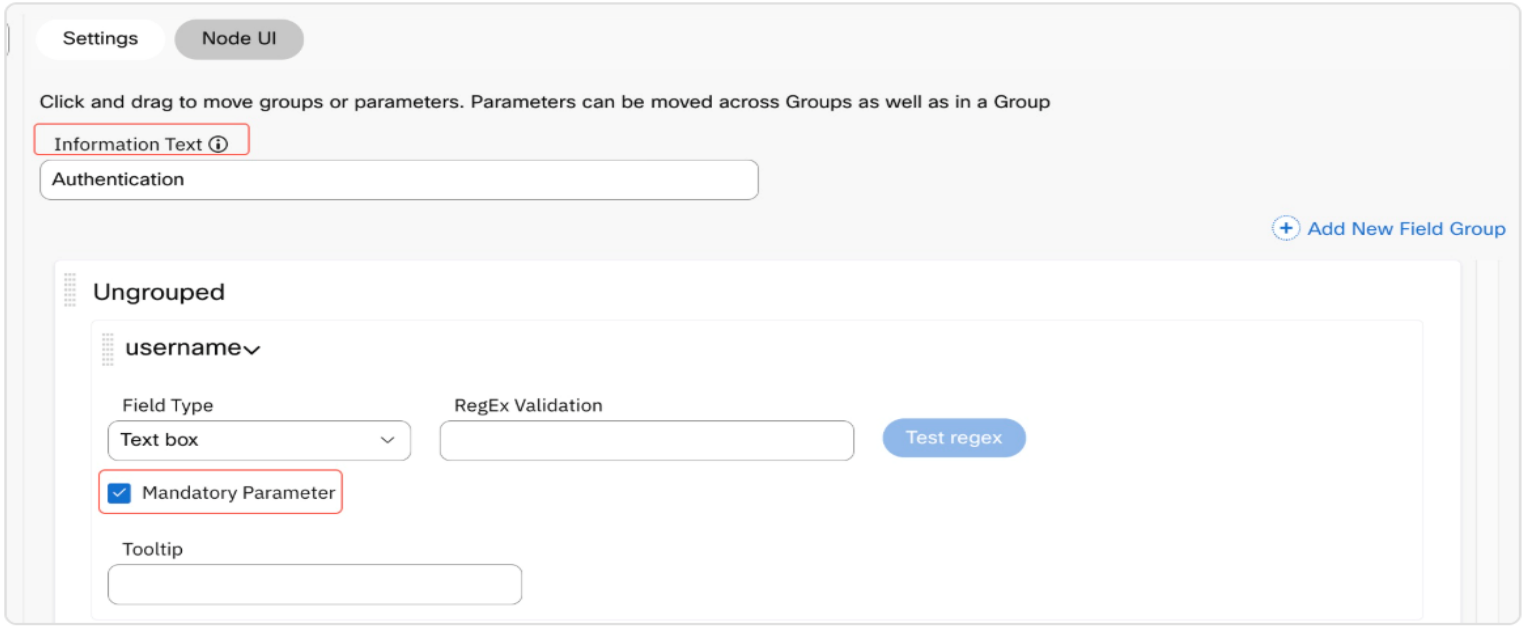
*Designing the user interface for the custom node*

## Field Types

- Text box – Single line input
- Text area – Multi-line input
- Dropdown – Selection list
- Checkbox – Boolean values

## Field Configuration

- Information Text – Label/section header
- Mandatory – Required fields
- Default Values – Pre-populated data
- Help Text – User guidance




*Designing the user interface for the custom node*

© 2025 Cisco and/or its affiliates. All rights reserved.

# OAuth 2.0 Authentication Setup

## Configuration Steps

01

### Select OAuth 2.0

In the Custom Node configuration, choose **OAuth 2.0** as the authentication method

02

### Choose Grant Type

**Authorization Code:** User-context operations (requires user login)**Client Credentials:** Server-to-server operations (machine-to-machine)

03

### Configure OAuth Settings

Enter Client ID, Client Secret, Authorization URL, Token URL, and required scopes from your API provider

04

### Set Callback URL

Use the Webex Connect provided callback URL in your third-party application's OAuth configuration

05

### Authorize the Integration

Navigate to **Assets → Integrations → Manage**, find your custom node, and complete authorization

© 2025 Cisco and/or its affiliates. All rights reserved.



cisco

# Managing Node Authorizations

## Authorization Lifecycle

- **Pending:** Authorization not yet completed
- **Authorized:** Successfully authenticated and ready
- **Expired:** Token needs refresh or re-authorization

## Available Actions

- **View:** Review authorization details
- **Update:** Modify configuration
- **Make Default:** Set as default
- **Remove:** Delete authorization

## Important Considerations

### ⚠️ Token Expiration

If credentials are updated in the integrating system, you must re-authorize the node

### ✓ Callback URLs

Updated to Webex Connect branded URLs. Ensure firewalls don't restrict access

### 🔄 Token Refresh

OAuth 2.0 tokens auto-refresh, but may need re-auth if refresh token expires

© 2025 Cisco and/or its affiliates. All rights reserved.

# Testing Your Custom Node

## Test Configuration

Before saving, use the built-in test feature to verify your custom node works correctly.

- Provide sample input values
- Check for HTTP 200 response
- Verify response parameters
- Review any error messages

## Success Indicators

**✓ Successful Test**

Response status 200, all expected parameters returned, no errors in logs

**✗ Common Issues**

- Incorrect endpoint URL
- Missing auth credentials
- Invalid JSON paths
- Wrong parameter types

© 2025 Cisco and/or its affiliates. All rights reserved.

# Best Practices & Tips

### 🎯 Design

- Use descriptive names for nodes and parameters
- Create reusable nodes for common integrations
- Document your custom nodes thoroughly
- Upload clear, recognizable SVG icons

### 🔒 Security

- Use OAuth 2.0 when available
- Never hardcode credentials in flows
- Configure security certificates when required
- Monitor authorization expiration dates

### ⚡ Performance

- Enable rate limiting to prevent API throttling
- Use appropriate timeout values
- Test thoroughly before making live
- Monitor transaction logs for errors

### 📊 Maintenance

- Keep track of certificate expiry dates
- Update callback URLs when needed
- Review and update scopes regularly
- Test after any API changes

© 2025 Cisco and/or its affiliates. All rights reserved.

# Key Takeaways

**1 Reusability**

Custom nodes enable you to build once and use across multiple flows, saving time and ensuring consistency

**2 Flexibility**

Support for REST and SOAP APIs with multiple authentication methods including OAuth 2.0

**3 Testing**

Always test your custom nodes before deployment to ensure proper configuration and response handling

**4 Management**

Monitor authorizations, maintain certificates, and keep your integrations up-to-date

© 2025 Cisco and/or its affiliates. All rights reserved.

# Use Cases

Advanced REST and Custom Node Use Cases

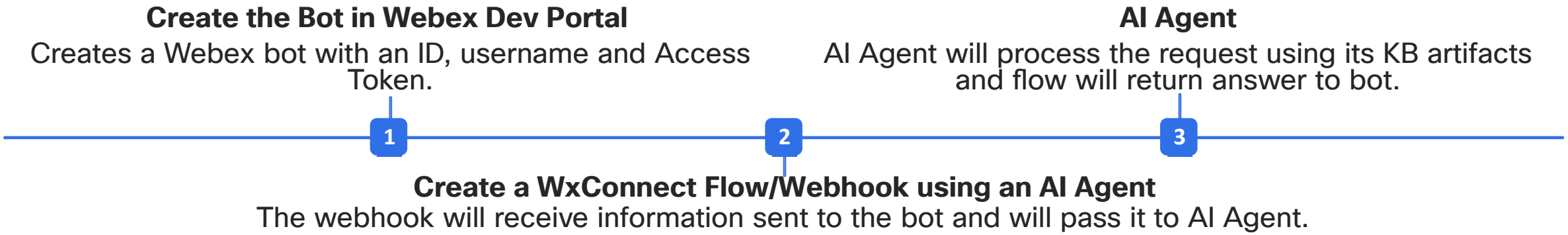
# Webex RFP Bot

## Integrating Webex with AI Agent

© 2025 Cisco and/or its affiliates. All rights reserved.

# Creating a Webex RFP Bot

The following are the steps required to build a Webex Bot and integrate it with an AI Agent in WxConnect.




## Understanding Each Component

### Creating the Webex Bot

First step to create the Webex bot integration that we will register the webhook with for processing questions to the Webex bot.

The registration is done via POSTMAN, there is no UI current to do this!

### Create the WxConnect Flow / Webhook

This webhook will be provided a payload that contains the BOT ID, with other information so the message can be extracted in the flow. The flow will also make additional API calls to extract the message that was sent to the Webex BOT.

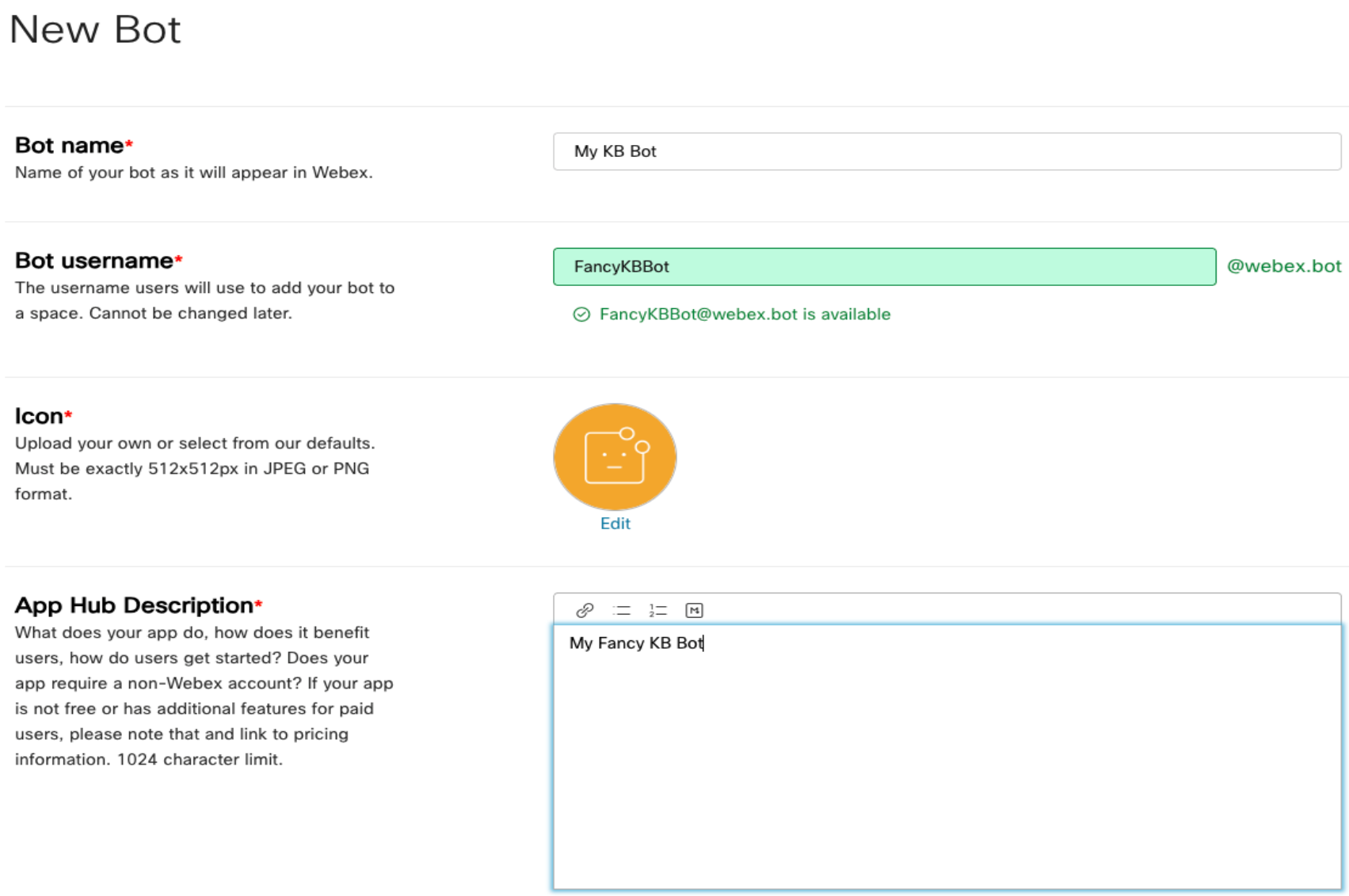
This message is then passed into the AI Agent node in the flow for processing against its KB artifacts.

### AI Agent

Processes the message extracted earlier using its KB artifacts that have been provided when the AI Agent was created. The answer is then then sent back via a Webex API call to the Webex Bot for deliver to the user.

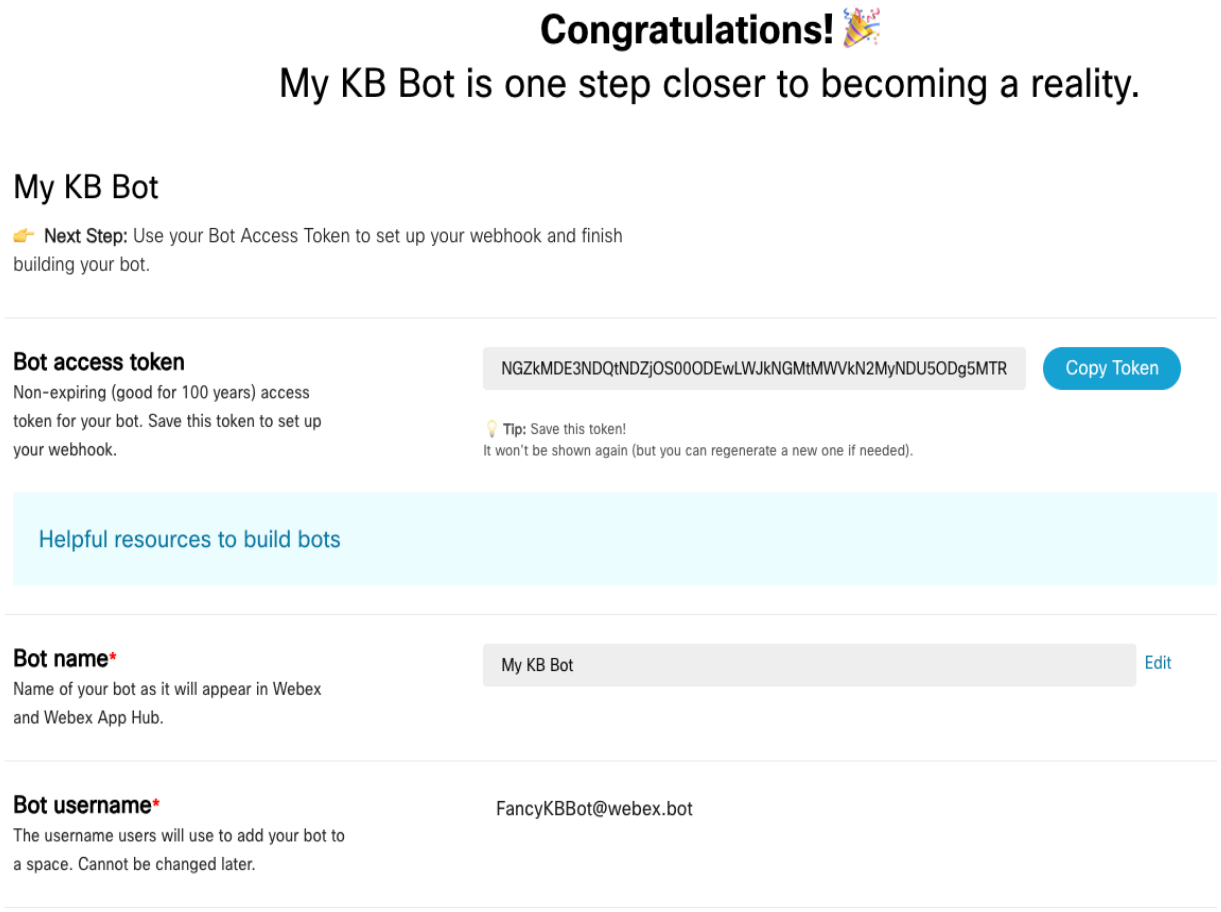
© 2025 Cisco and/or its affiliates. All rights reserved.

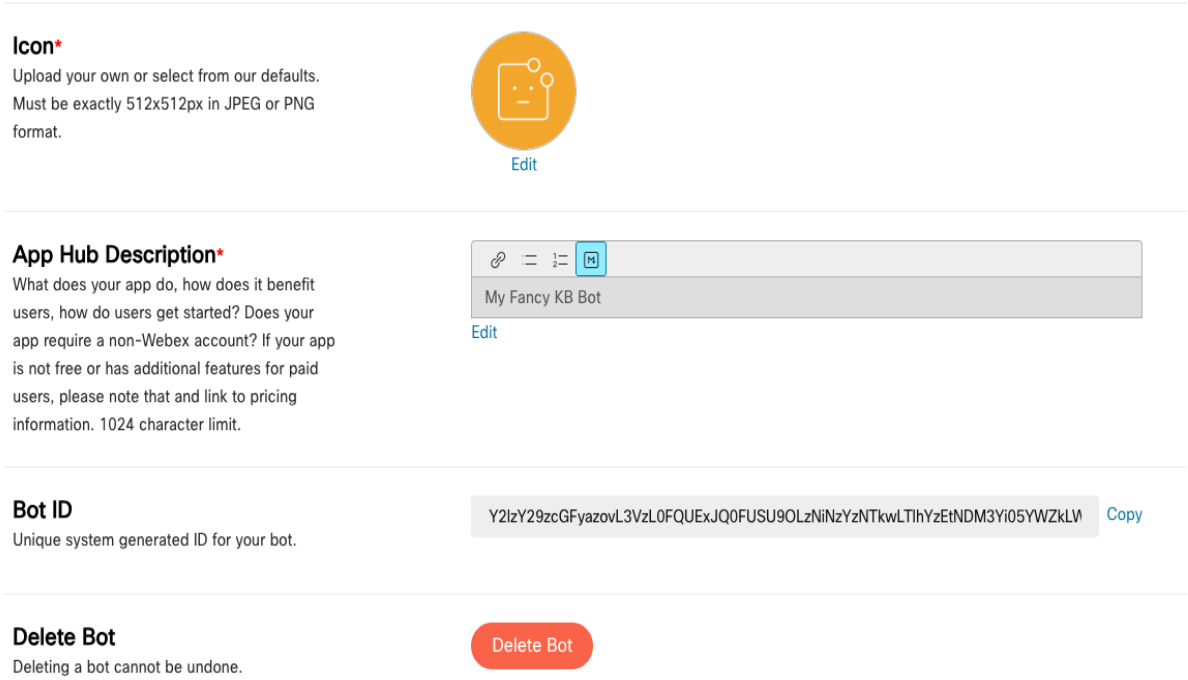
# Step 1: Create a Webex Bot




© 2025 Cisco and/or its affiliates. All rights reserved.

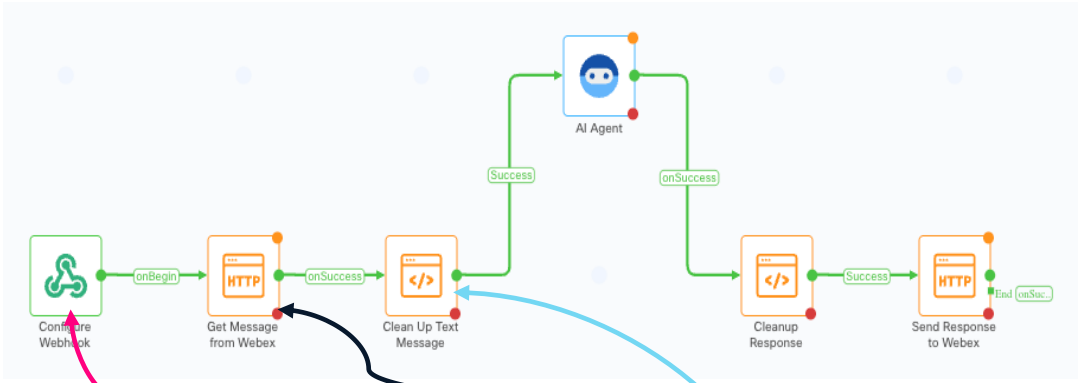
# Step 1: Create a Webex Bot (Continued)




© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 2: Create a WxConnect Flow/Webhook




Example Data

Paste Json | Paste XML | Send Via Hook

PROVIDE SAMPLE INPUT

```
{
    "id": "Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3Qt
    "name": "Healthcare Communications vertical bot
    "resource": "Resource",
    "event": "created",
    "orgId": "asdsada",
    "appId": "sdfsf",
    "ownedBy": "creator",
    "status": "active",
    "actorId": "sdflsjdflfj",
    "data": {
        "id": "sdfs",
        "type": "sdfs",
        "messageId": "sdfsf",
        "roomId": "sdfsf"
```

SELECT OUTPUT VARIABLES

PARSED VARIABLES(14)

- data.created:sdfsf
- resource:Resource
- orgId:asdsada
- data.roomId:sdfsf
- data.messageId:sdfsf
- actorId:sdflsjdflfj

**Get Message from Webex** — Help

Configuration | Transition Actions (Optional)

Make GET, POST, PUT, PATCH and DELETE requests to your application

Method: GET

Endpoint URL: https://webexapis.com/v1/messages/ $(n2.inboundWebhook.data.id)

| Header | Value |
|---|---|
| Authorization | Bearer $(Bot_Token) |
| Content-Type | application/json |

+ Add Another Header

Connection Timeout: e.g., 1000 ms

Request Timeout: e.g., 1000 ms

Proxy Address (Optional): e.g., 192.128.19.243:80

Output Variables

JSON | XML — Import From Sample

| Output Variable Name | Response Entity | Response Path |
|---|---|---|
| messageText | Body | text |

+ Add Variable

**Clean Up Text Message** — Help

Configuration | Transition Actions (Optional)

Configure node on-enter / on-leave operations

Action 1

Time: On-enter

Action: Set variable

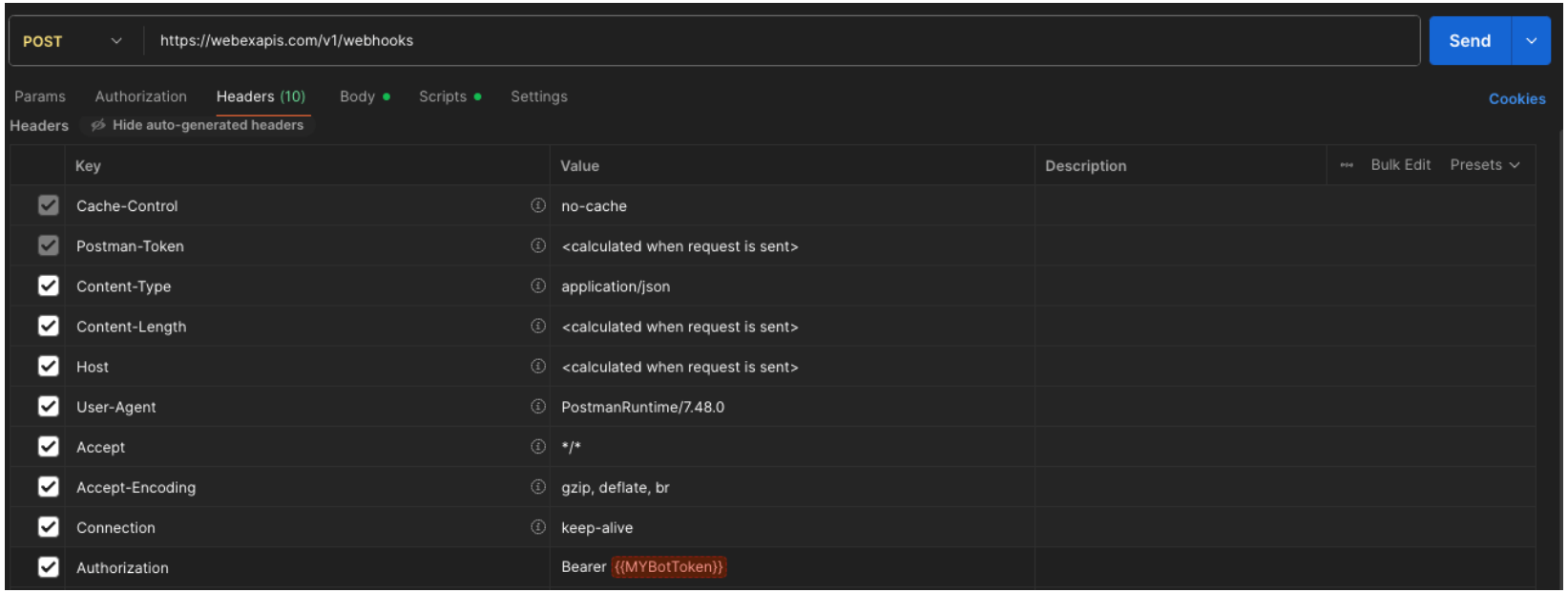
| Variable | Value |
|---|---|
| inboundMessage | $(n10.messageText) |

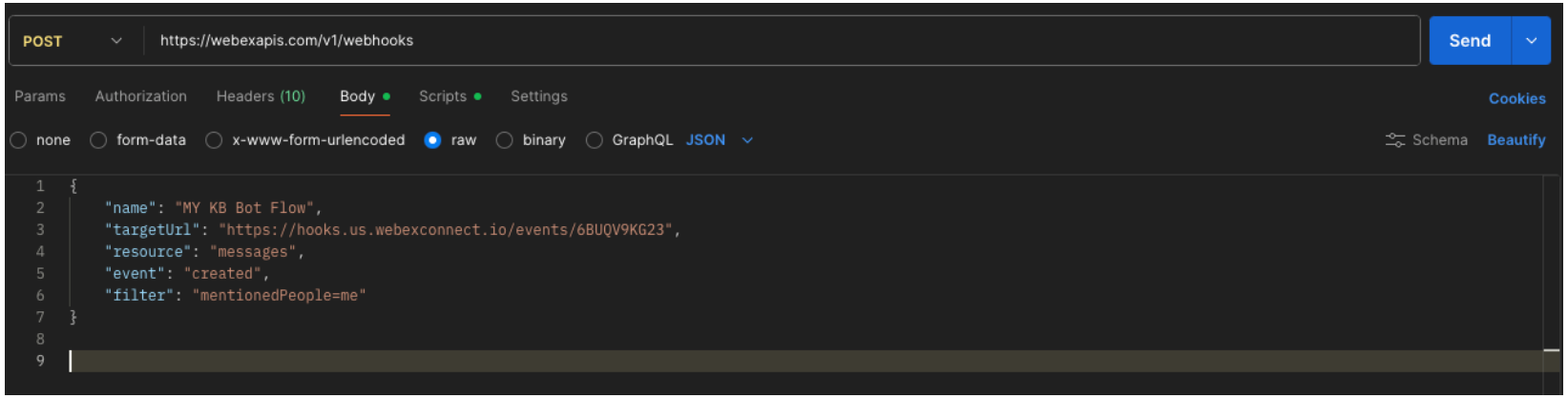
+ Add

+ Add Action

© 2025 Cisco and/or its affiliates. All rights reserved.

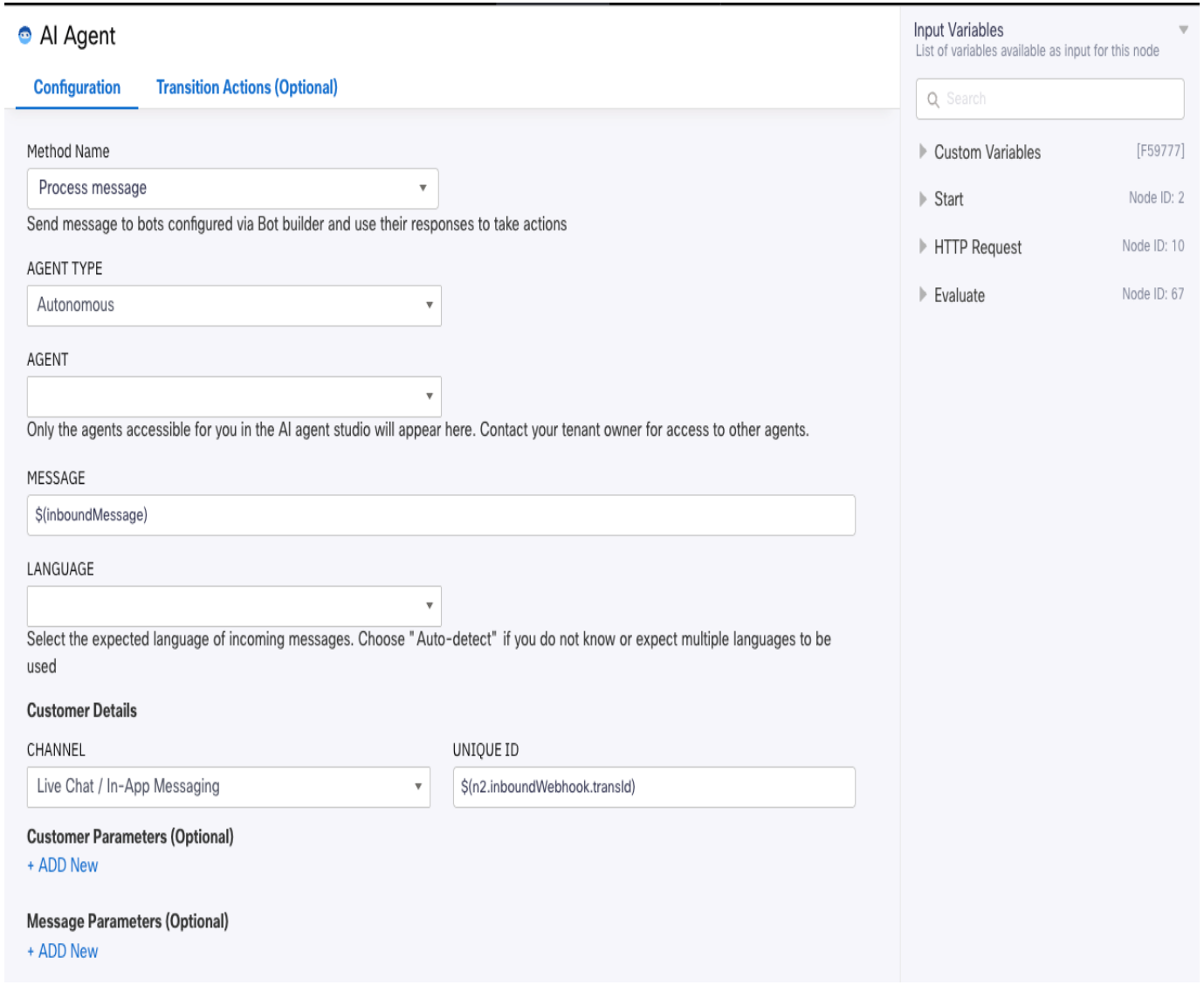
# Step 2: Register the Webhook with Webex Bot (Continue)

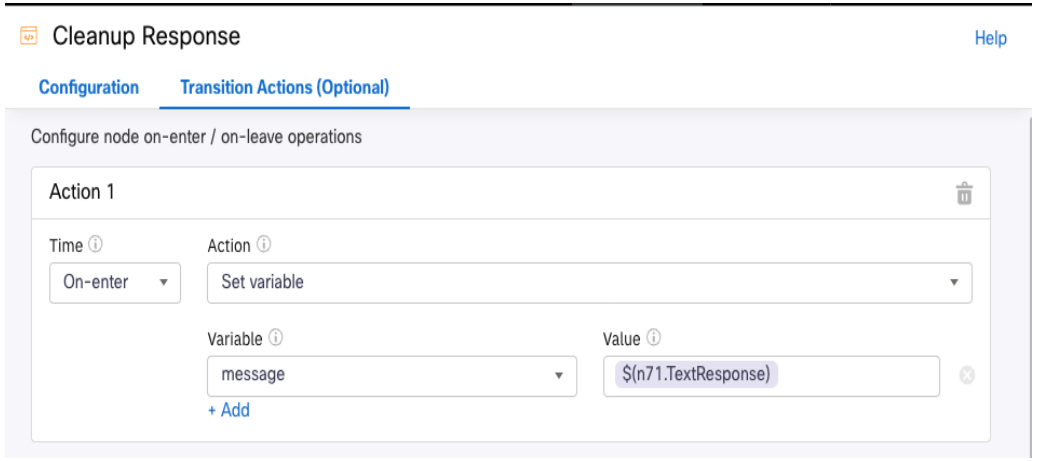


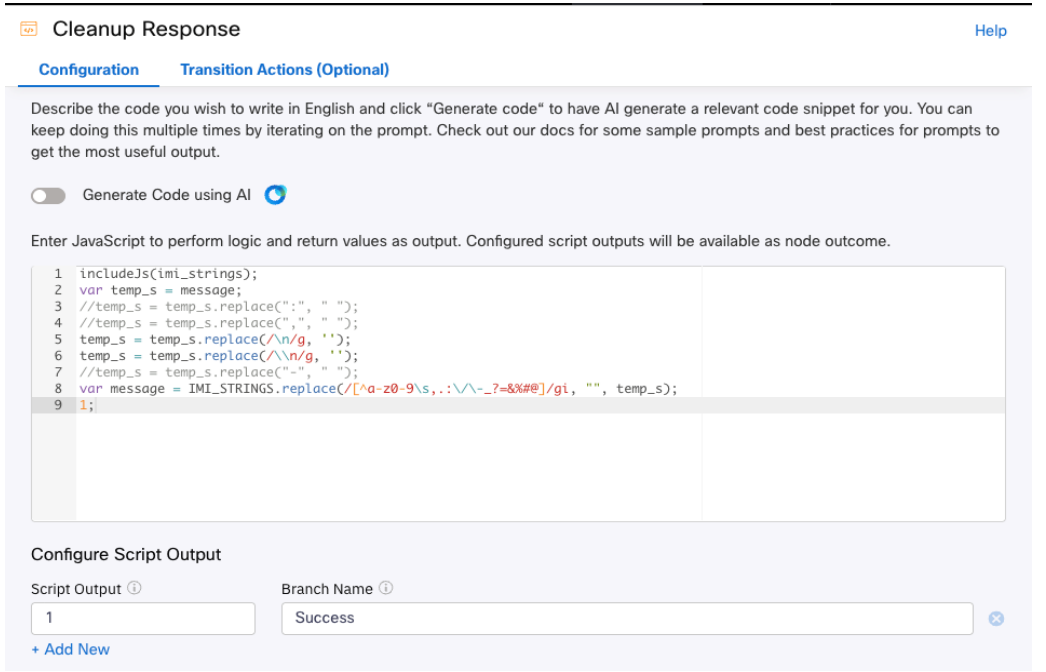




© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 3 : Process Webex Bot message with AI Agent

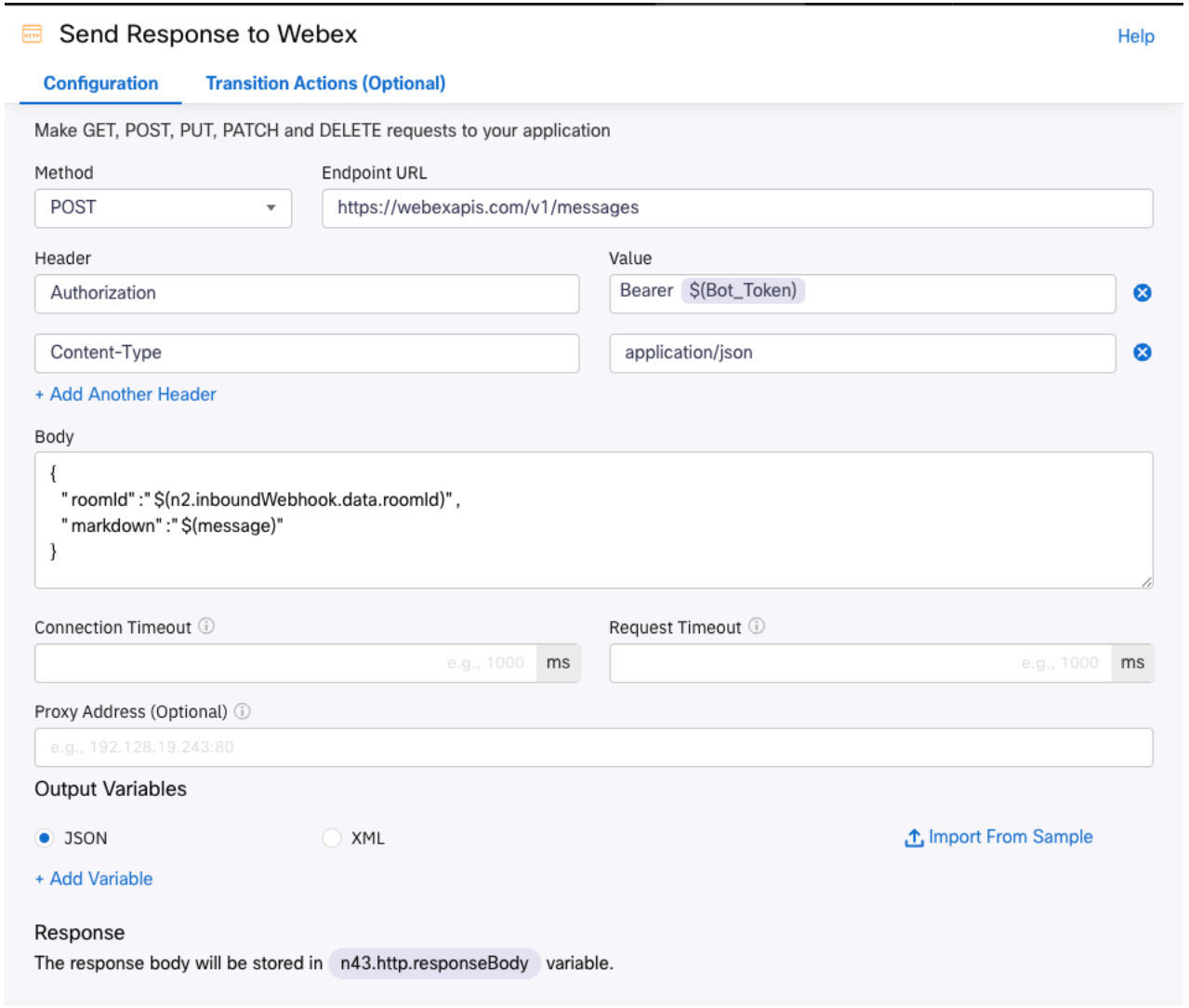



© 2025 Cisco and/or its affiliates. All rights reserved.

# Step 3 : Return message to Webex Bot (Continued)



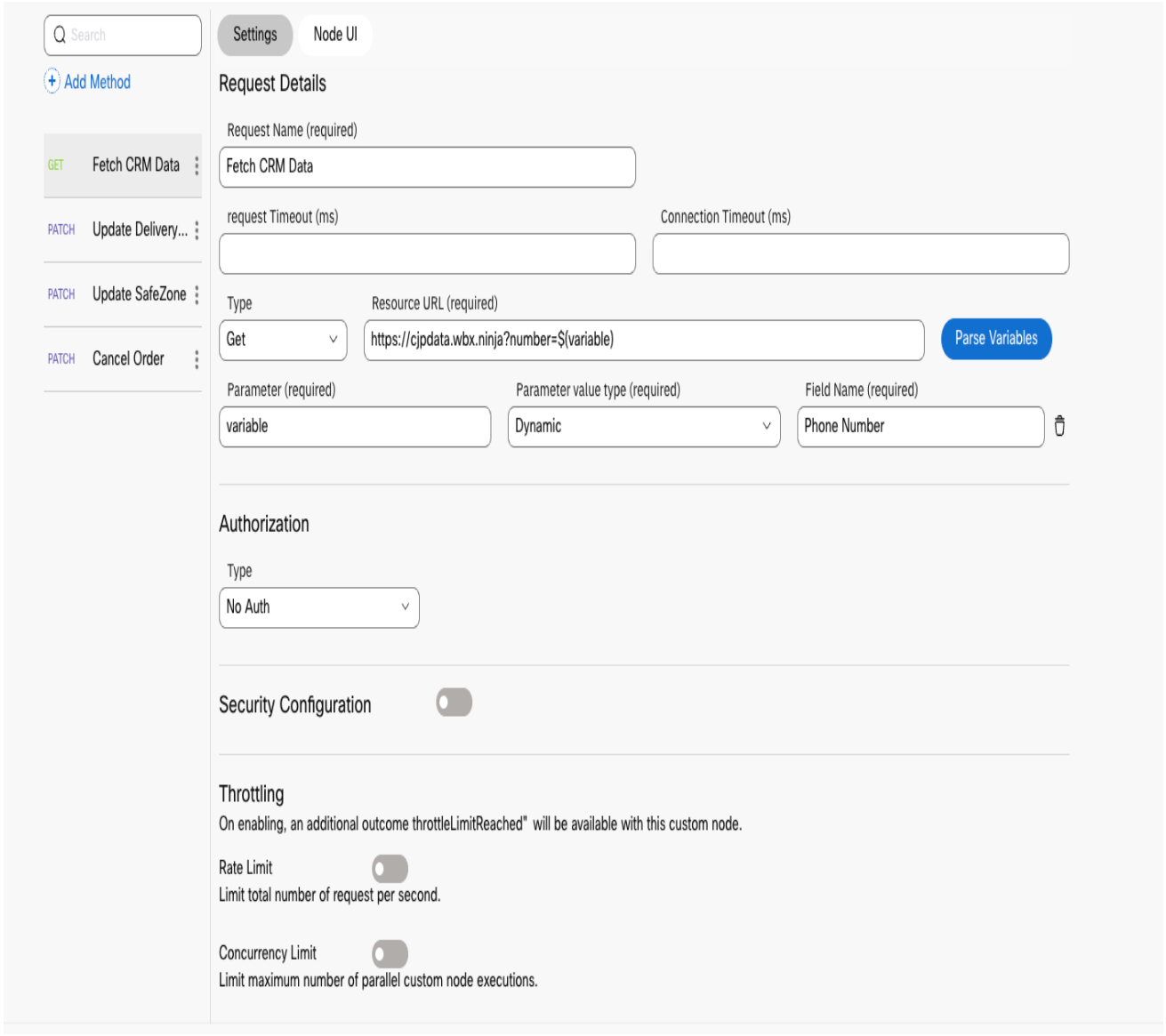

© 2025 Cisco and/or its affiliates. All rights reserved.

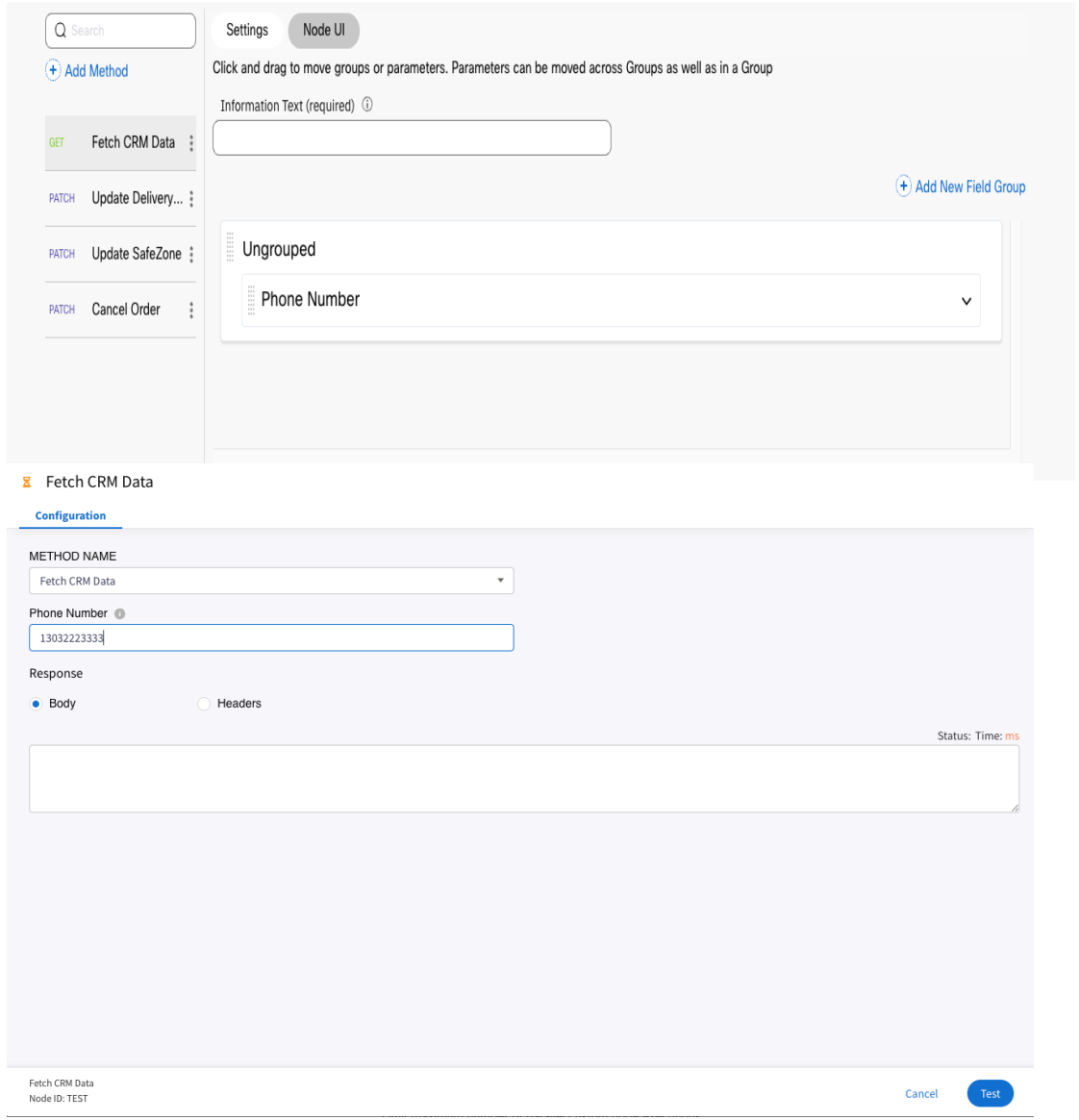
# Custom CRM Node

Example of a Custom CRM node

© 2025 Cisco and/or its affiliates. All rights reserved.

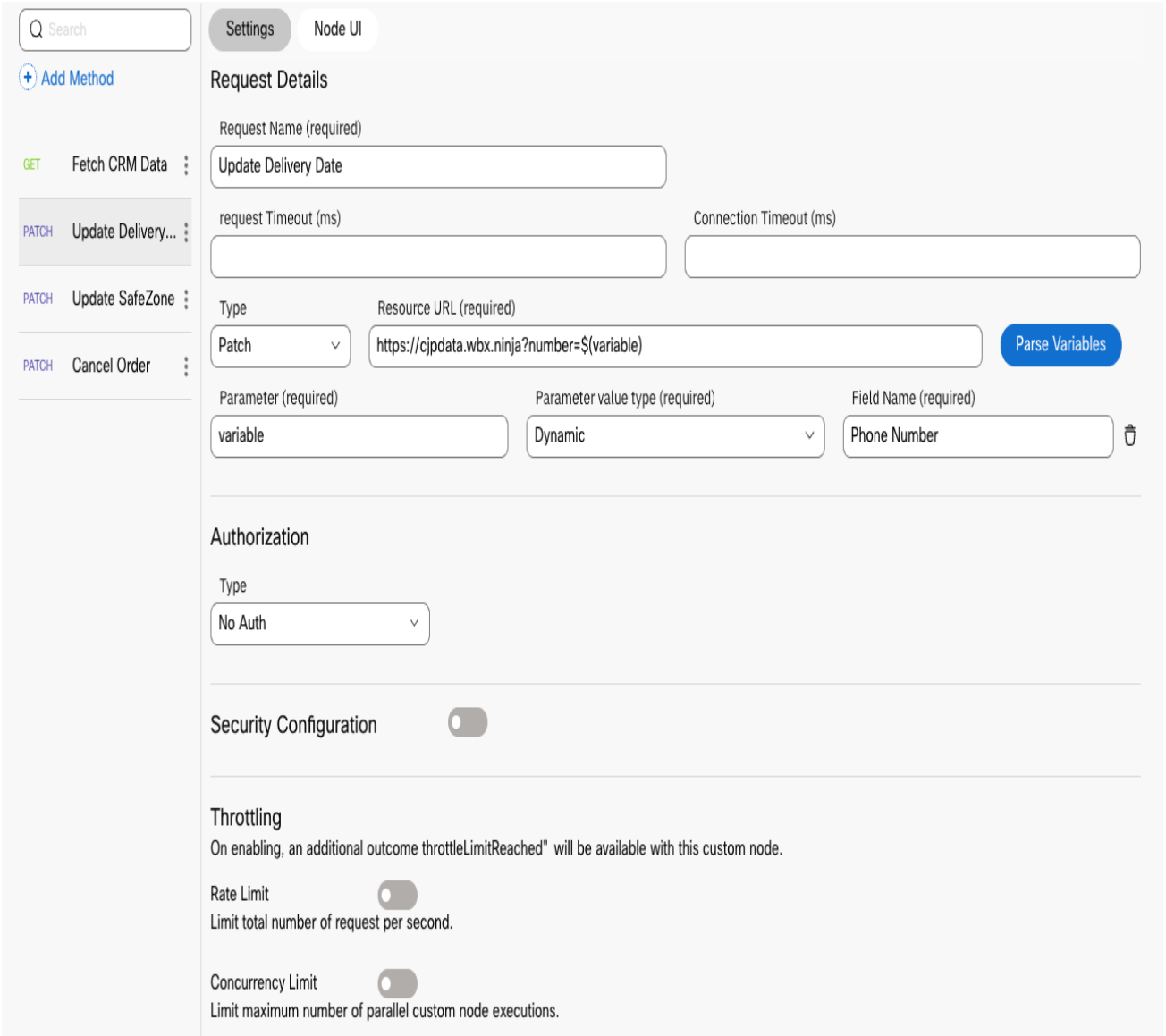
# Custom CRM Node Review - GET

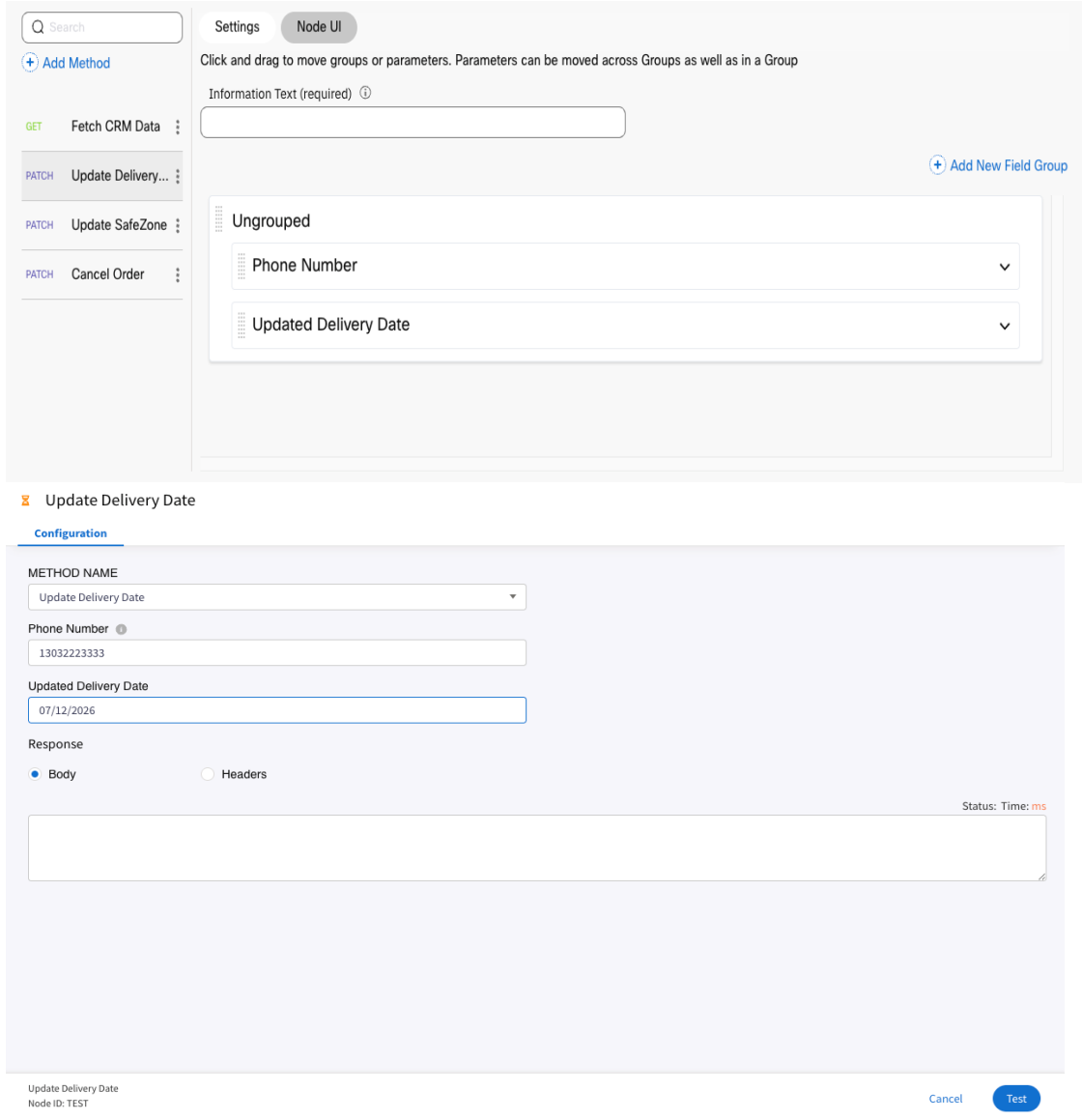



© 2025 Cisco and/or its affiliates. All rights reserved.

# Custom CRM Node Review – Patch



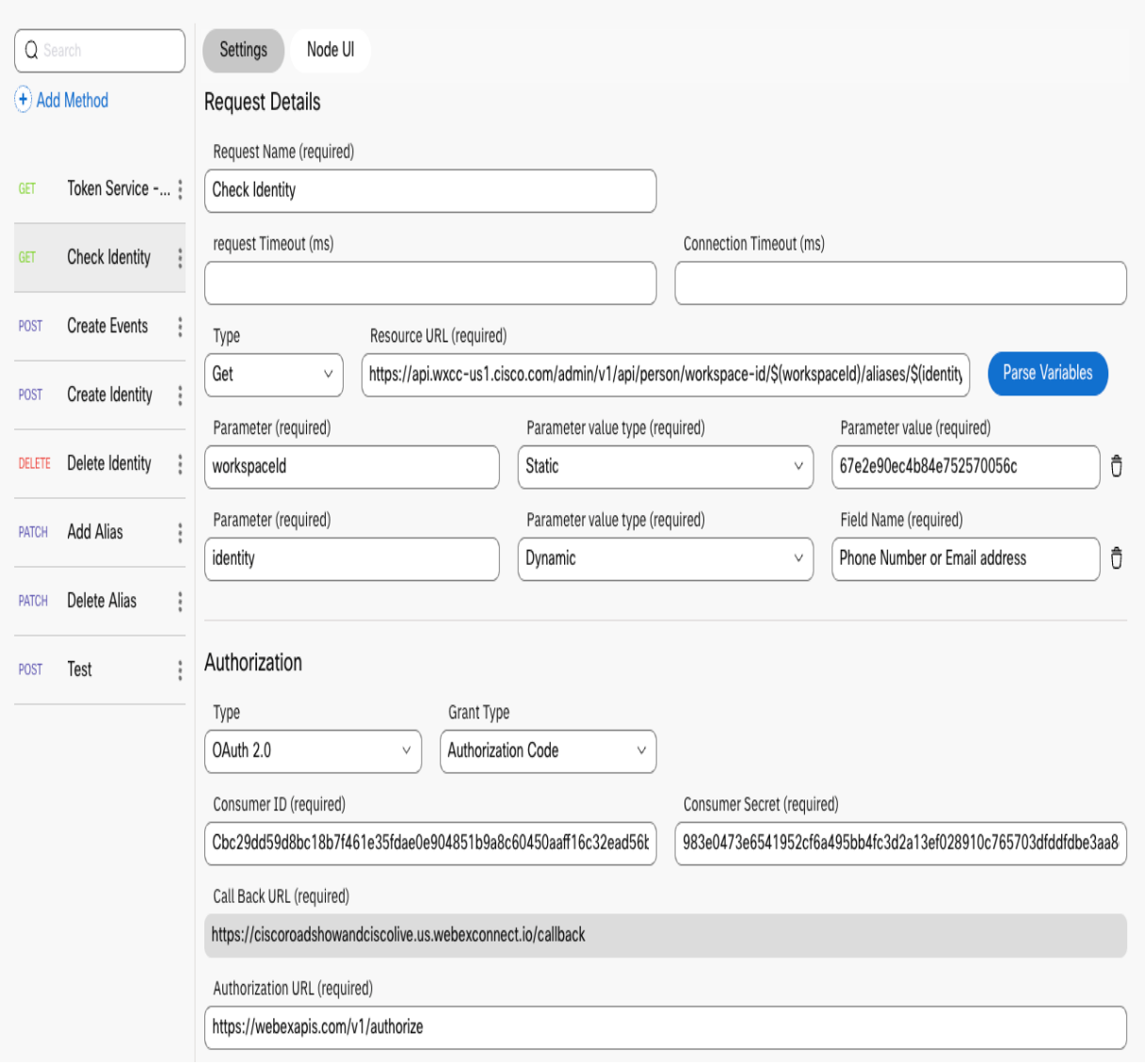

© 2025 Cisco and/or its affiliates. All rights reserved.

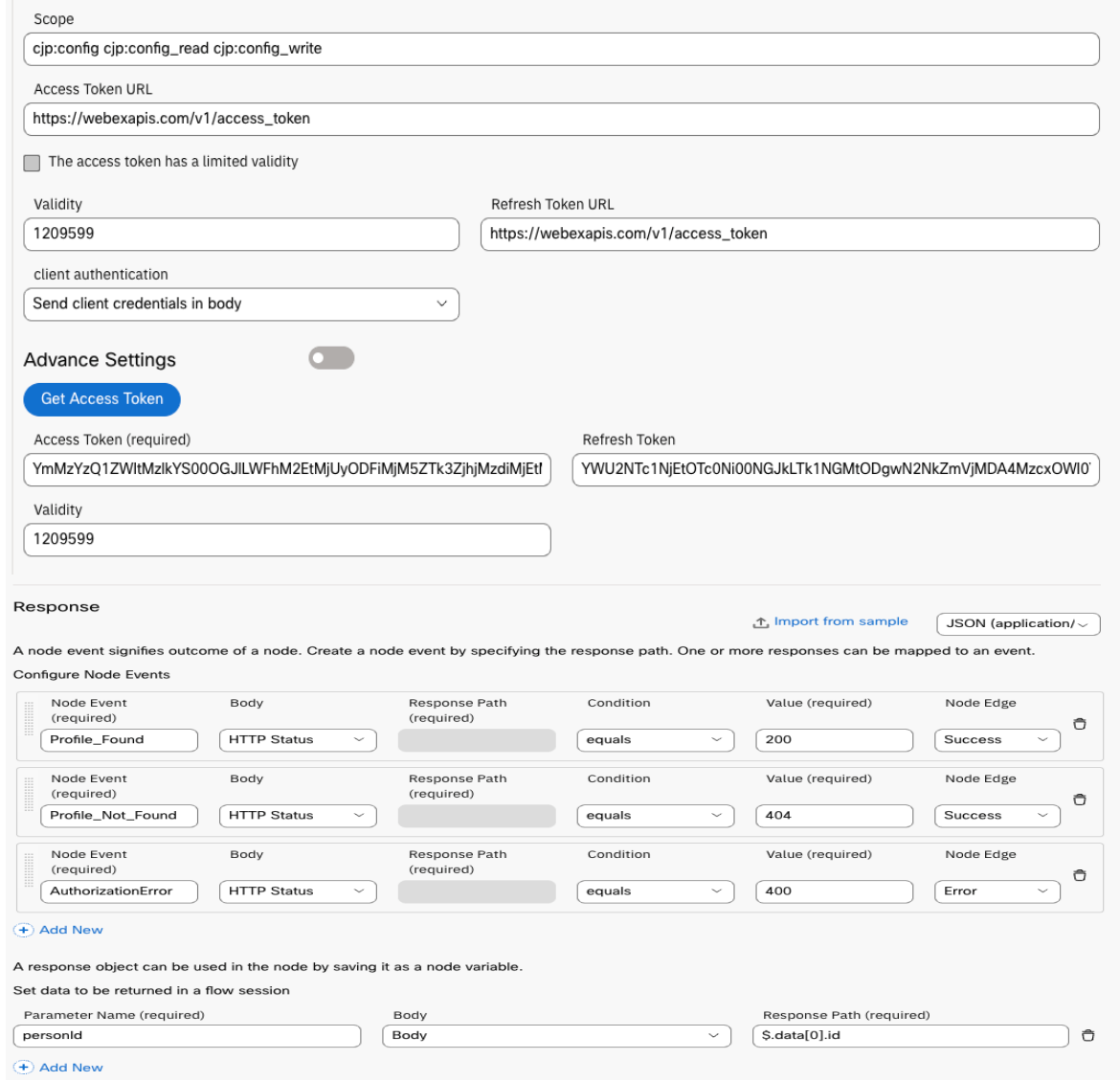
# Advanced Custom Node Review

Old JDS custom node example

© 2025 Cisco and/or its affiliates. All rights reserved.

# JDS Node Review – GET (Check Identity)

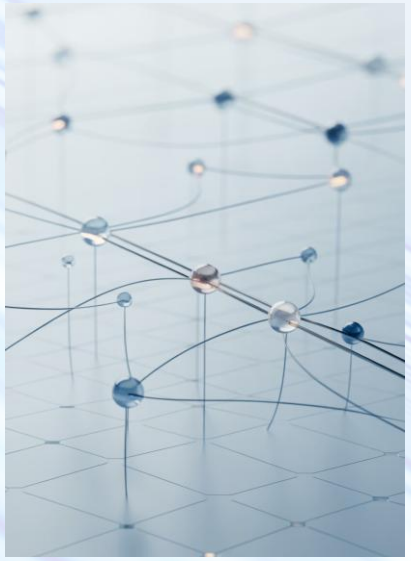



© 2025 Cisco and/or its affiliates. All rights reserved.

# A2A / MCP Primer

Understanding Agent-to-Agent Communication and Model Context Protocol



# What is Agent-to-Agent (A2A)?

Agent-to-Agent communication represents a fundamental shift in how AI systems operate. Rather than functioning in isolation, AI agents can now interact, collaborate, and coordinate with each other autonomously to accomplish complex tasks that would be impossible for a single agent.

### Autonomous Communication

AI systems engage in direct dialogue with each other without requiring human intermediation, enabling real-time decision making and coordination

### Structured Protocols

Standardized methods for exchanging information, delegating tasks, and maintaining consistent communication patterns across diverse AI systems

### Cross-Platform Coordination

Different AI models from various vendors working together seamlessly, sharing context and collaborating on unified objectives

### Complex Workflows

Multi-step problem solving where agents divide work, execute specialized tasks, and synthesize results into comprehensive solutions

**Real-world analogy:** Like microservices in software architecture, A2A enables specialized AI agents to work together as a cohesive system, each contributing their unique capabilities to accomplish complex goals.

© 2025 Cisco and/or its affiliates. All rights reserved.

# Model Context Protocol (MCP)

The Model Context Protocol is an open standard developed by Anthropic that fundamentally changes how AI applications connect to data sources and tools. MCP eliminates the need for custom integrations by providing a universal interface that any AI model can use to access resources consistently and securely.

### Universal Connector

A standardized protocol that allows AI models to connect to any data source using the same interface, regardless of the underlying technology

### No Custom Integrations

Eliminates the need for building one-off connections between each AI model and data source, dramatically reducing development time

### Tools & Data Access

Models can access resources, execute functions, and utilize templates consistently across different platforms and environments

### Open Standard

An open-source protocol developed by Anthropic for universal adoption, fostering innovation and preventing vendor lock-in

**Think of MCP as USB for AI** – just as USB allows any device to connect to any computer, MCP allows any AI model to connect to any data source or tool through a single, standardized interface.

© 2025 Cisco and/or its affiliates. All rights reserved.

# How MCP Works: Architecture Overview

MCP operates through a three-tier architecture that separates concerns and enables flexible, secure connections between AI applications and their data sources. This design ensures scalability while maintaining simplicity.

1

**MCP Host (Application)**

The client application that wants AI capabilities – such as Claude Desktop, IDEs, or custom business applications

2

**MCP Client (Connector)**

The connector layer that maintains protocol connections and manages communication between the host and servers

3

**MCP Server**

Lightweight programs that expose specific capabilities to AI models through the standardized MCP protocol

4

**Resources, Tools & Prompts**

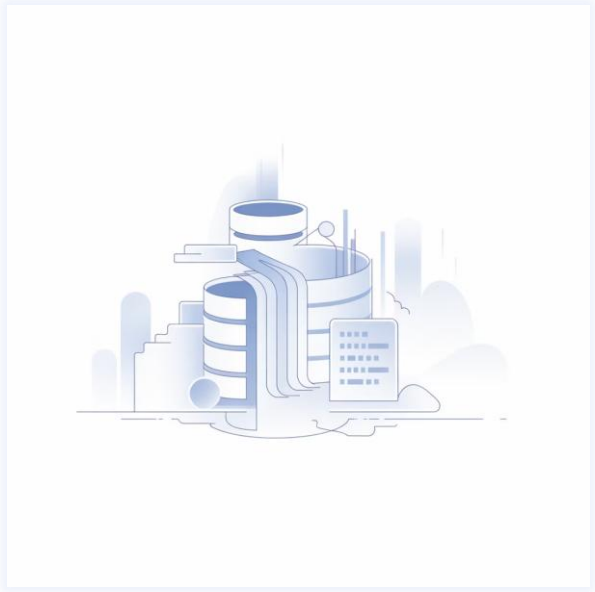
The actual data, functions, and templates that the AI model can access and utilize through the server

© 2025 Cisco and/or its affiliates. All rights reserved.

# MCP Components Explained

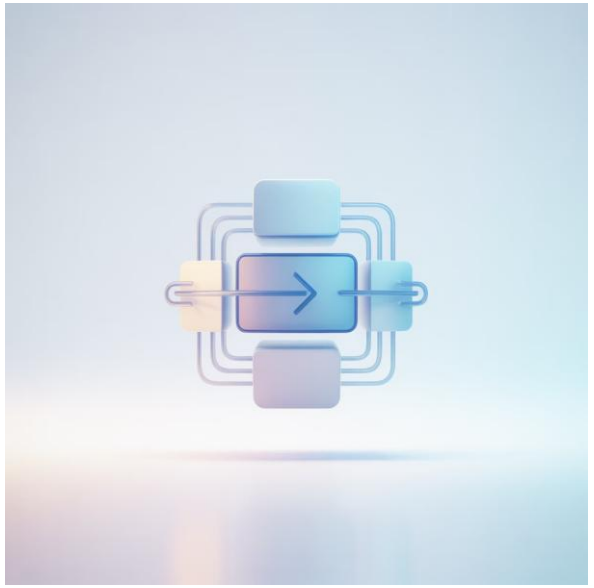
MCP servers expose three primary types of capabilities to AI models, each serving a distinct purpose in enabling intelligent, contextual interactions with external systems and data.

## Resources



Data sources that models can read and analyze, including files, database contents, API responses, and live system data. Resources provide the context that models need to make informed decisions.

## Tools



Executable functions that models can invoke to perform actions, such as creating records, sending notifications, running queries, or triggering workflows. Tools extend the model's capabilities beyond text generation.

## Prompts

Reusable templates that standardize common tasks and queries. Prompts ensure consistency and best practices when interacting with specific data sources or performing routine operations.

- **Communication Protocol:** MCP uses JSON-RPC 2.0 for message exchange, supporting both stdio (standard input /output) and HTTP with Server-Sent Events (SSE) for real-time updates.

© 2025 Cisco and/or its affiliates. All rights reserved.

# Benefits of A2A with MCP

The combination of Agent-to-Agent communication and the Model Context Protocol creates a powerful ecosystem that transforms how organizations deploy and manage AI capabilities at scale.

### For Organizations

- Dramatically reduced integration complexity across diverse AI systems and platforms
- Accelerated deployment timelines for new AI capabilities and features
- Enhanced data security with centralized access control and governance
- Vendor-agnostic architecture that prevents lock-in and enables best-of-breed solutions
- Lower total cost of ownership through reusable integrations

### For Developers

- Write integration code once, connect to any compatible AI model
- Build libraries of reusable components and integrations
- Clear separation of concerns between data sources and AI logic
- Thriving open source ecosystem with community-contributed servers
- Simplified testing and debugging with standardized protocols

**Key Advantage:** Organizations can build AI integrations that work seamlessly across multiple platforms and models without vendor lock-in, future-proofing their AI investments.

© 2025 Cisco and/or its affiliates. All rights reserved.

# Key Takeaways

The combination of Agent-to-Agent communication and Model Context Protocol creates a powerful, flexible ecosystem that is transforming how we build and deploy AI systems. Here's what you need to remember:

### A2A Enables Autonomy

Agent-to-Agent communication allows AI systems to work together autonomously, coordinating complex workflows without human intervention and achieving outcomes impossible for single agents.

### MCP Provides the Standard

Model Context Protocol delivers the universal language for AI-to-data connections, eliminating custom integrations and enabling any model to access any resource through a consistent interface.

### Together They Scale

Combined, A2A and MCP create a flexible, scalable AI ecosystem that adapts to changing needs, supports diverse use cases, and grows with organizational requirements.

### Open Standards Win

Open standards drive innovation, reduce vendor lock-in, and foster thriving ecosystems where developers can build, share, and benefit from community contributions.

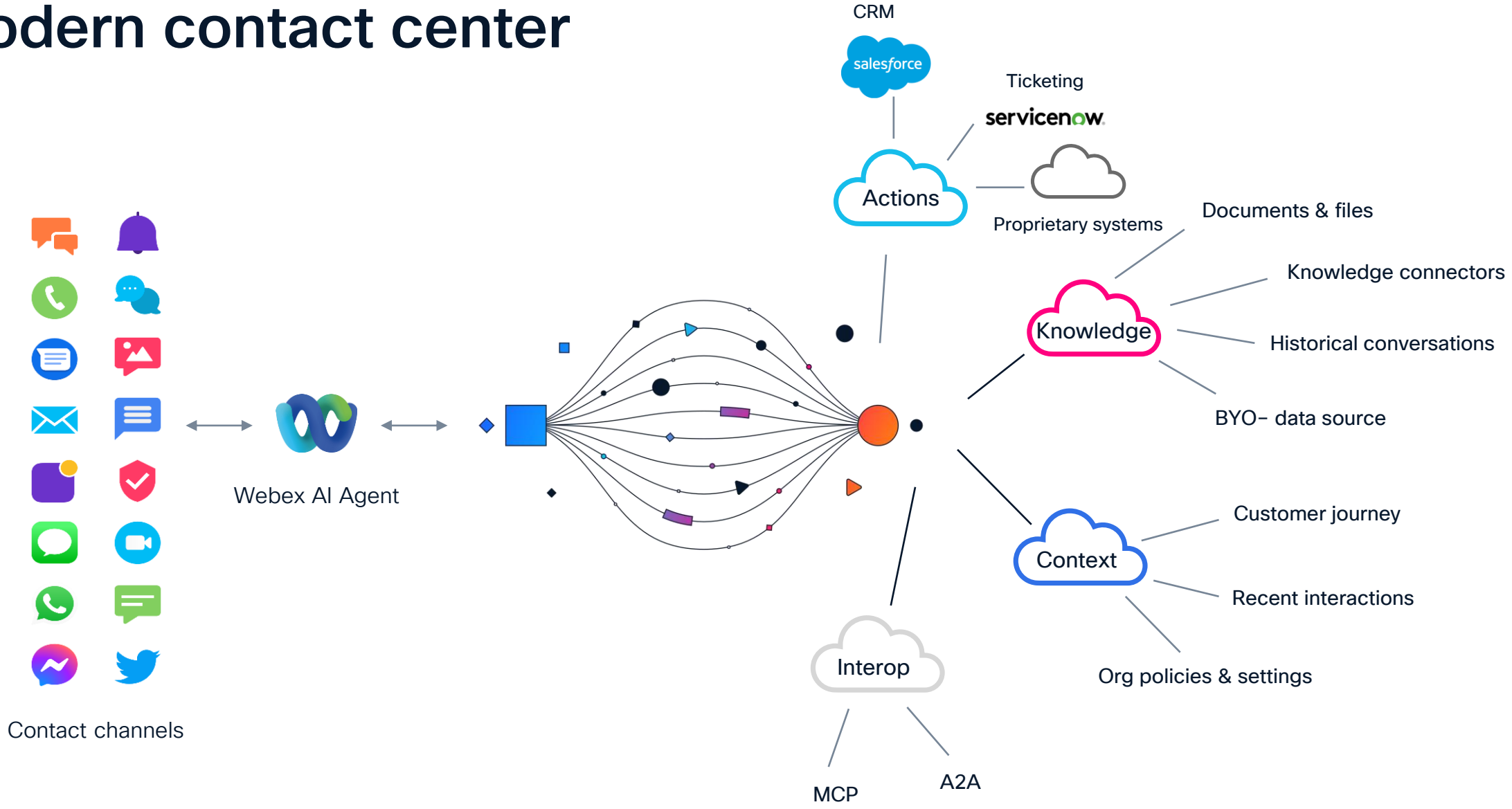
**The Big Picture:** The combination of A2A and MCP represents a fundamental shift in how we build and deploy AI systems – moving from isolated, custom-built solutions to an interconnected ecosystem of specialized agents working together through open standards.

© 2025 Cisco and/or its affiliates. All rights reserved.

# Roadmap

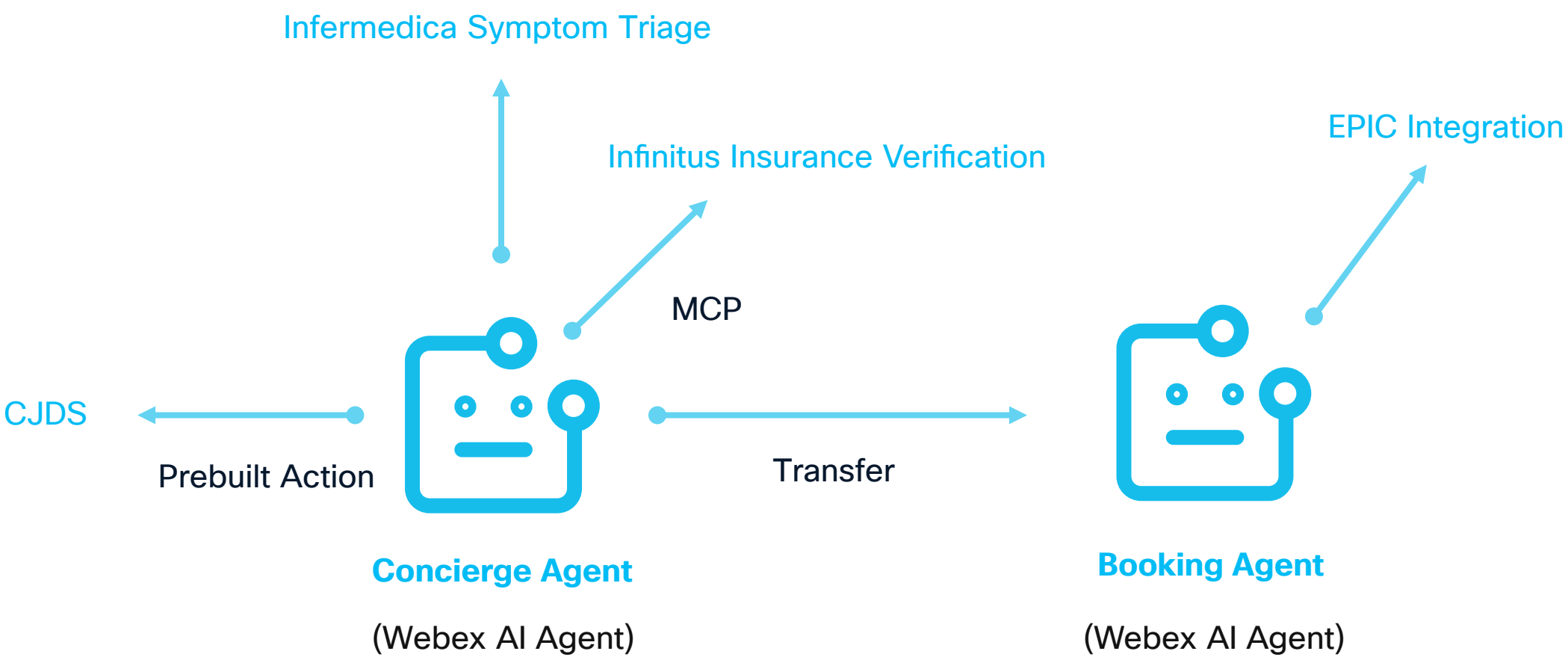
Cisco Roadmap for A2A and MCP Support

# An agentic and intelligent front door for the modern contact center



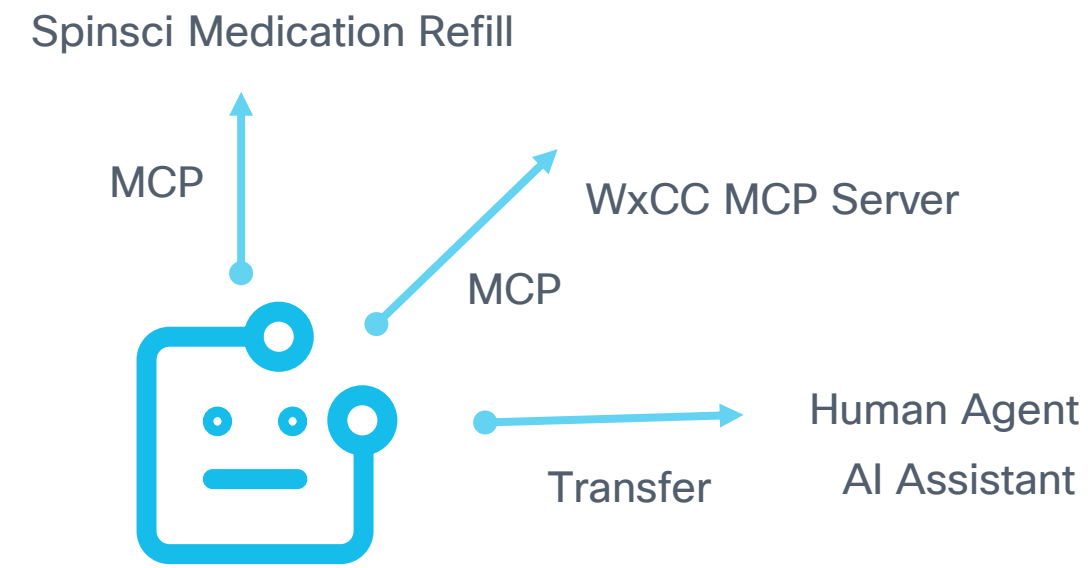

© 2025 Cisco and/or its affiliates. All rights reserved.

# Engage via Contact Center



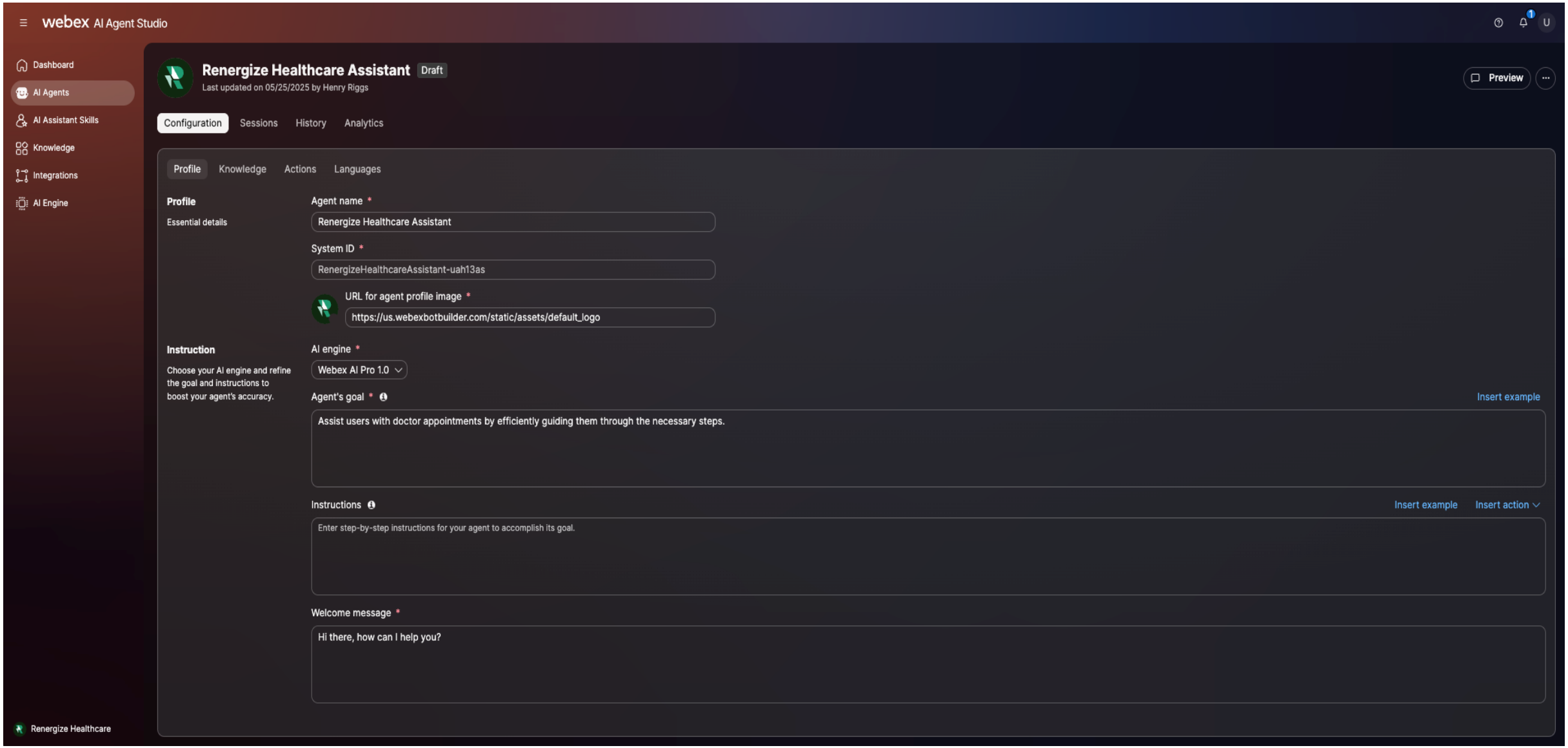

© 2025 Cisco and/or its affiliates. All rights reserved.

# Proactive Engagement



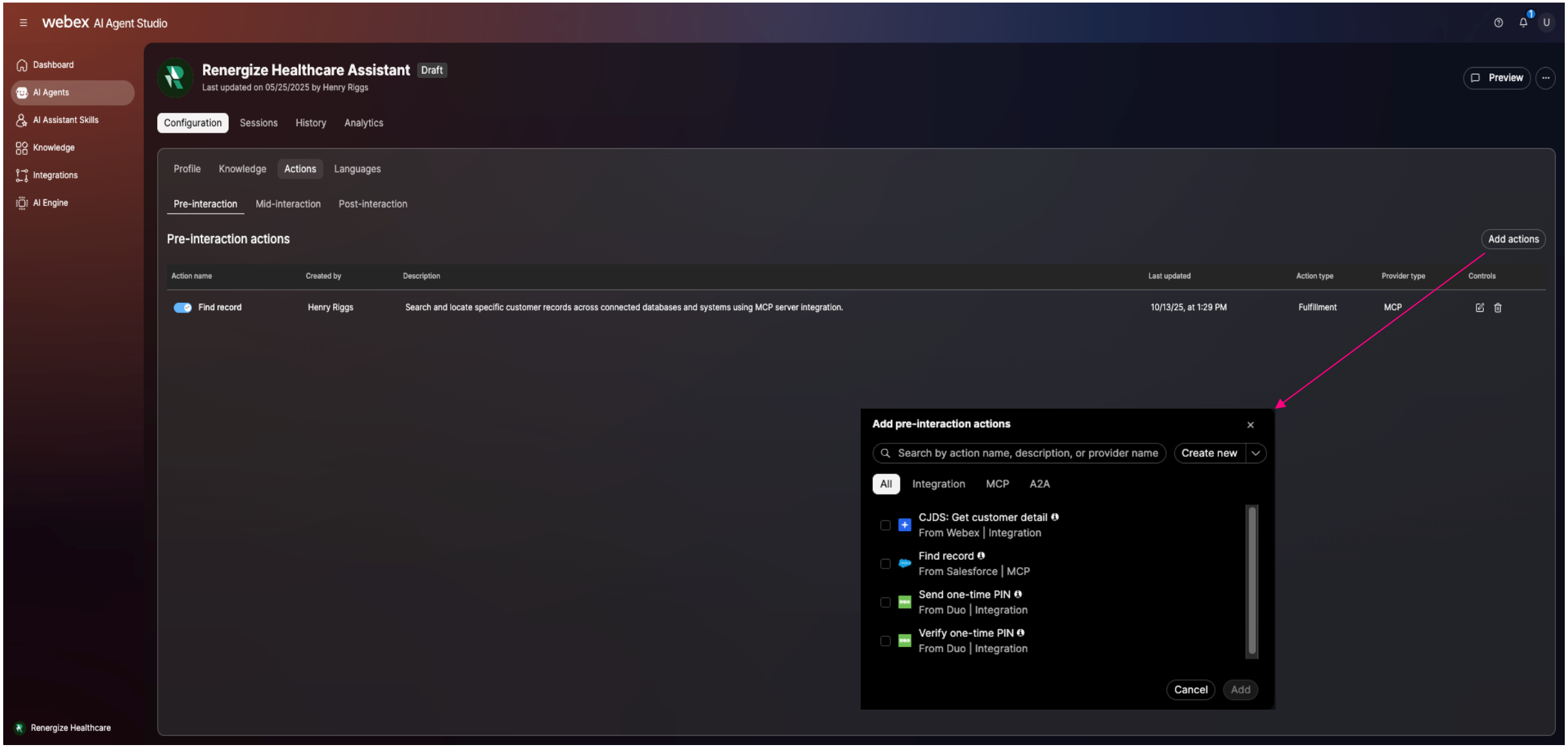

© 2025 Cisco and/or its affiliates. All rights reserved.

# New Webex AI Agent Studio: AI Agent Configuration



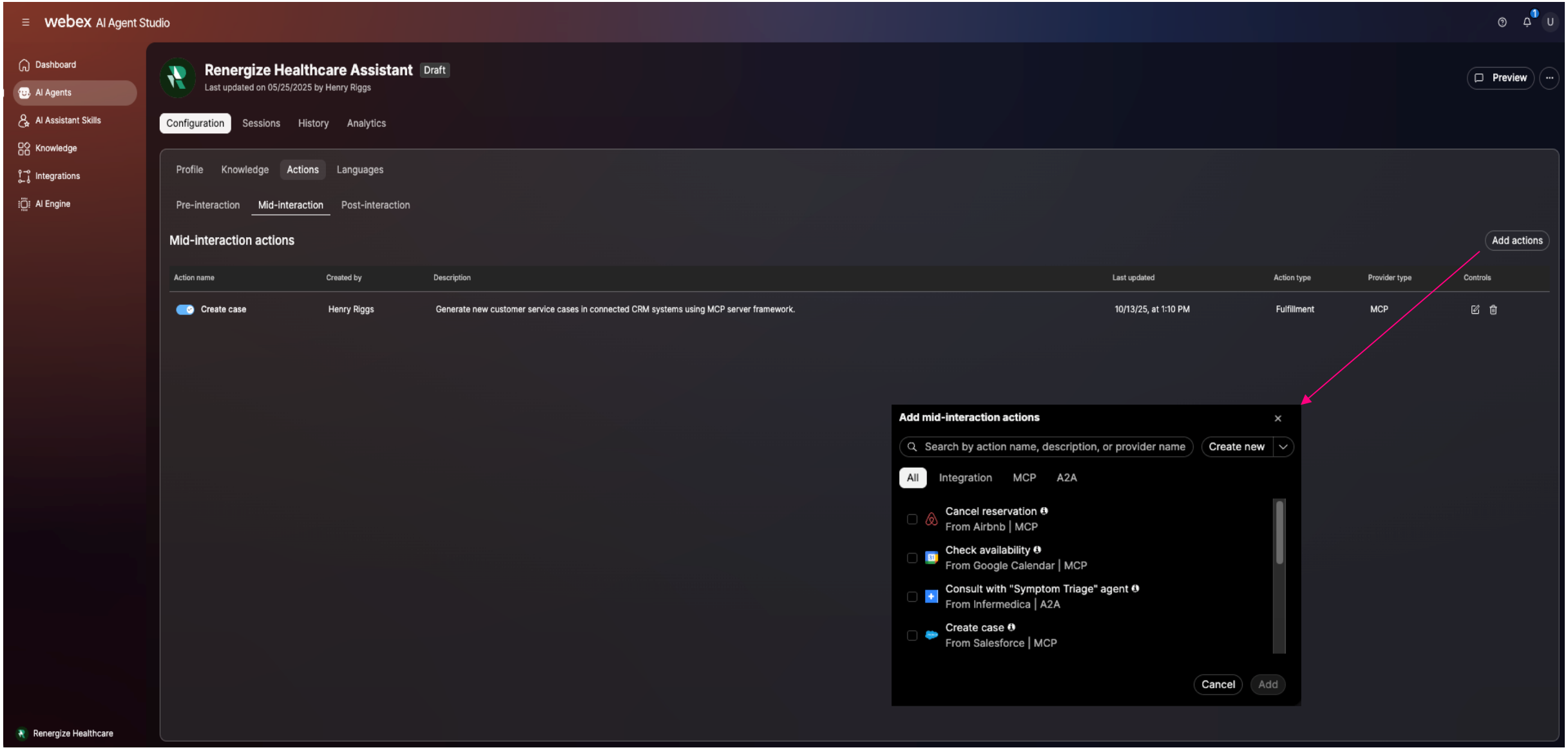

© 2025 Cisco and/or its affiliates. All rights reserved.

# New Webex AI Agent Studio: Pre-Interaction



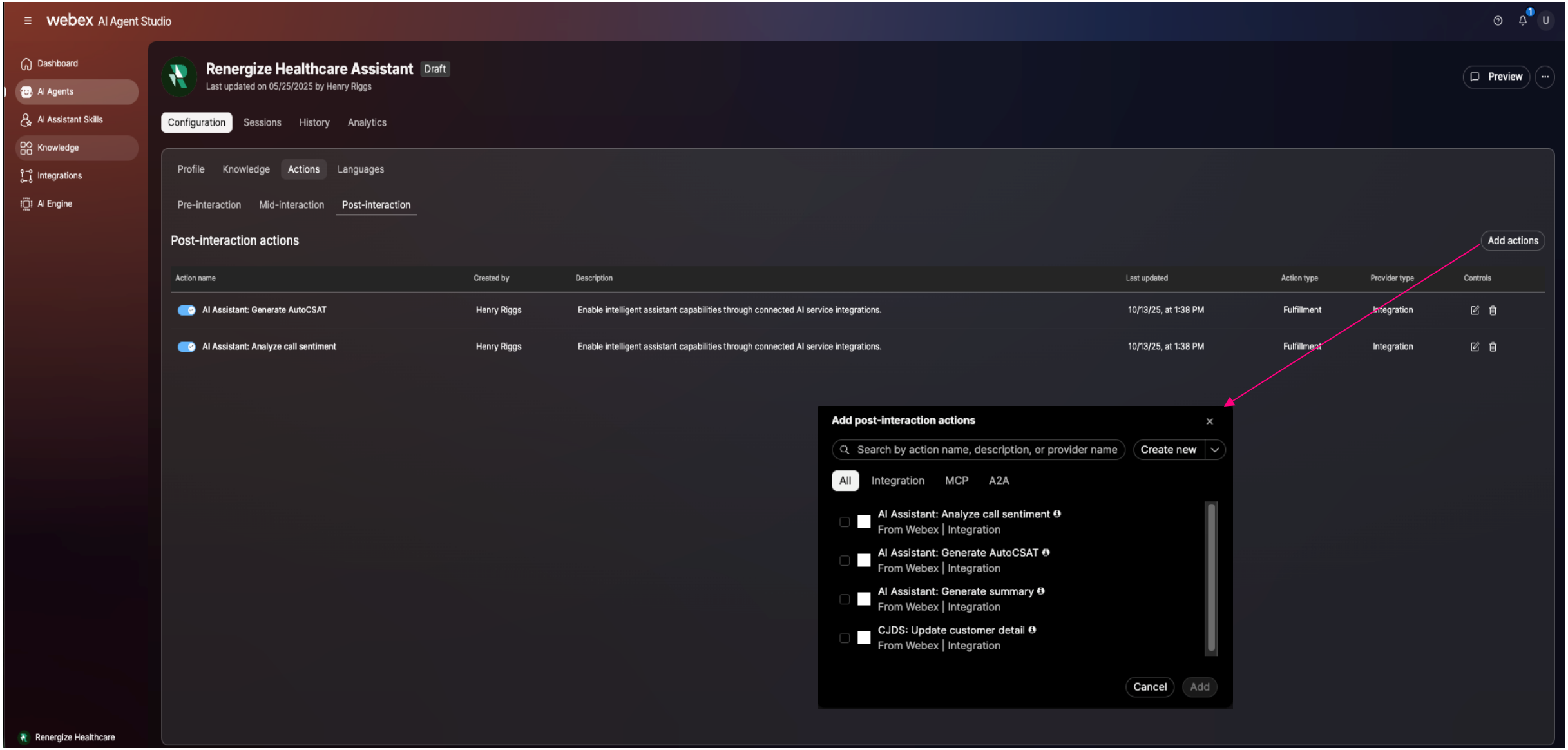

© 2025 Cisco and/or its affiliates. All rights reserved.

# New Webex AI Agent Studio: Mid-Interaction



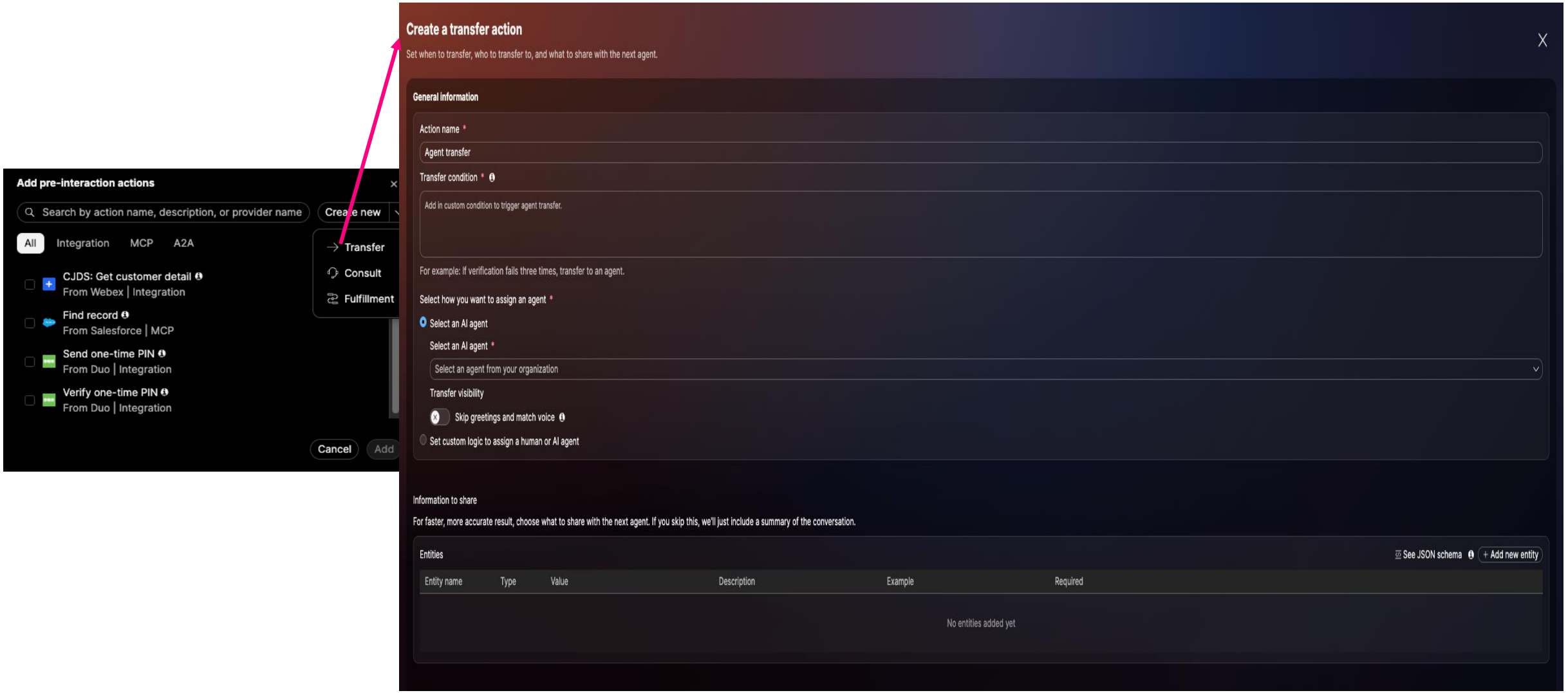

© 2025 Cisco and/or its affiliates. All rights reserved.

# New Webex AI Agent Studio: Post-Interaction



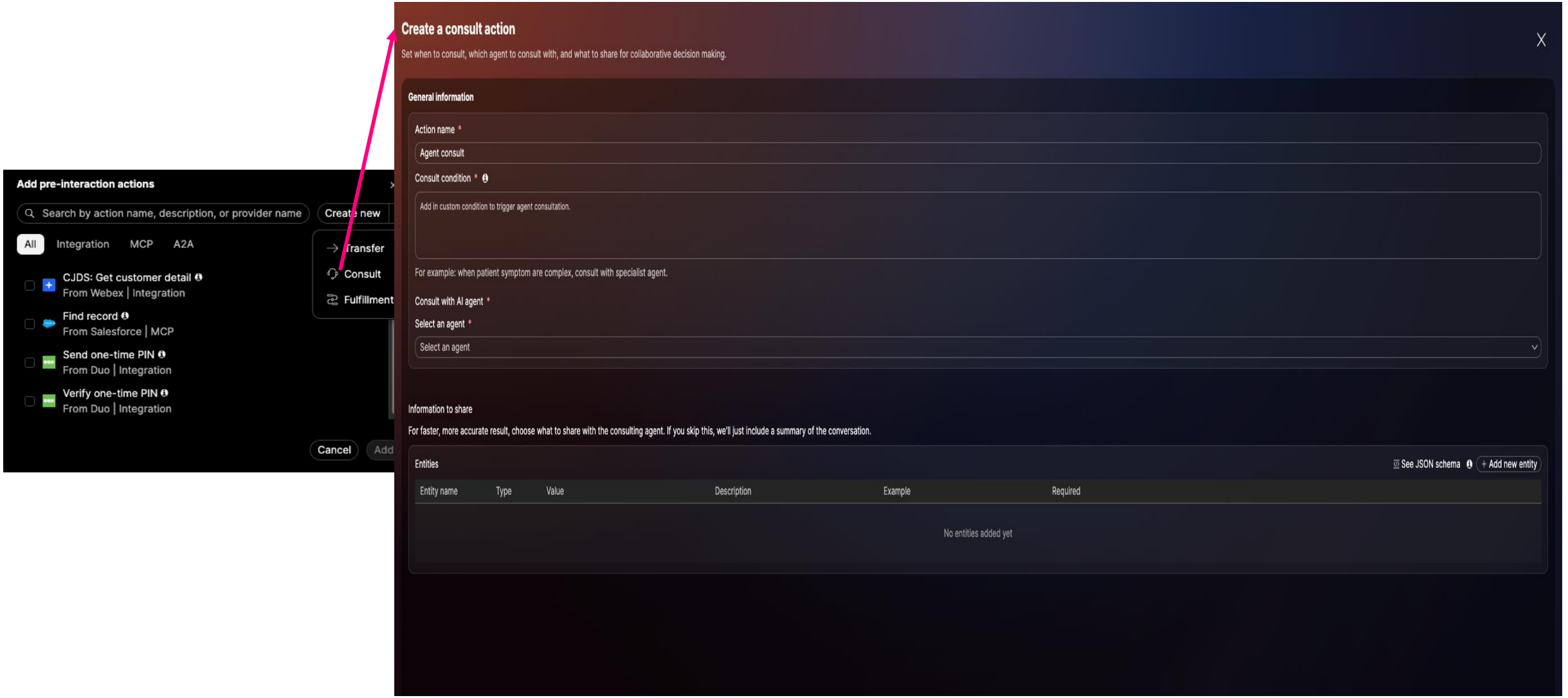

© 2025 Cisco and/or its affiliates. All rights reserved.

# New Webex AI Agent Studio: Create a new transfer




© 2025 Cisco and/or its affiliates. All rights reserved.

# New Webex AI Agent Studio: Create a new consult

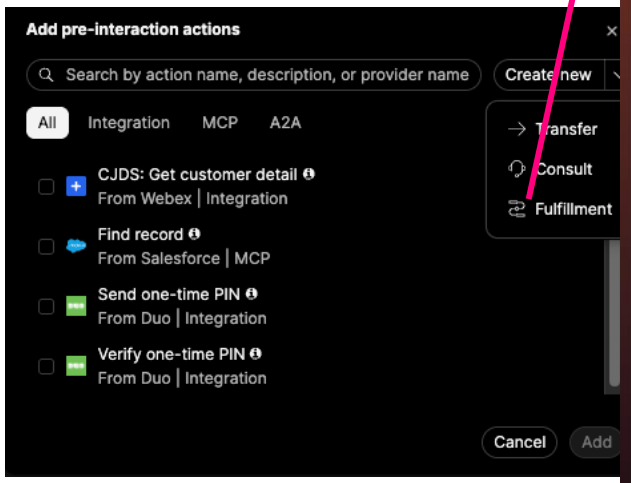


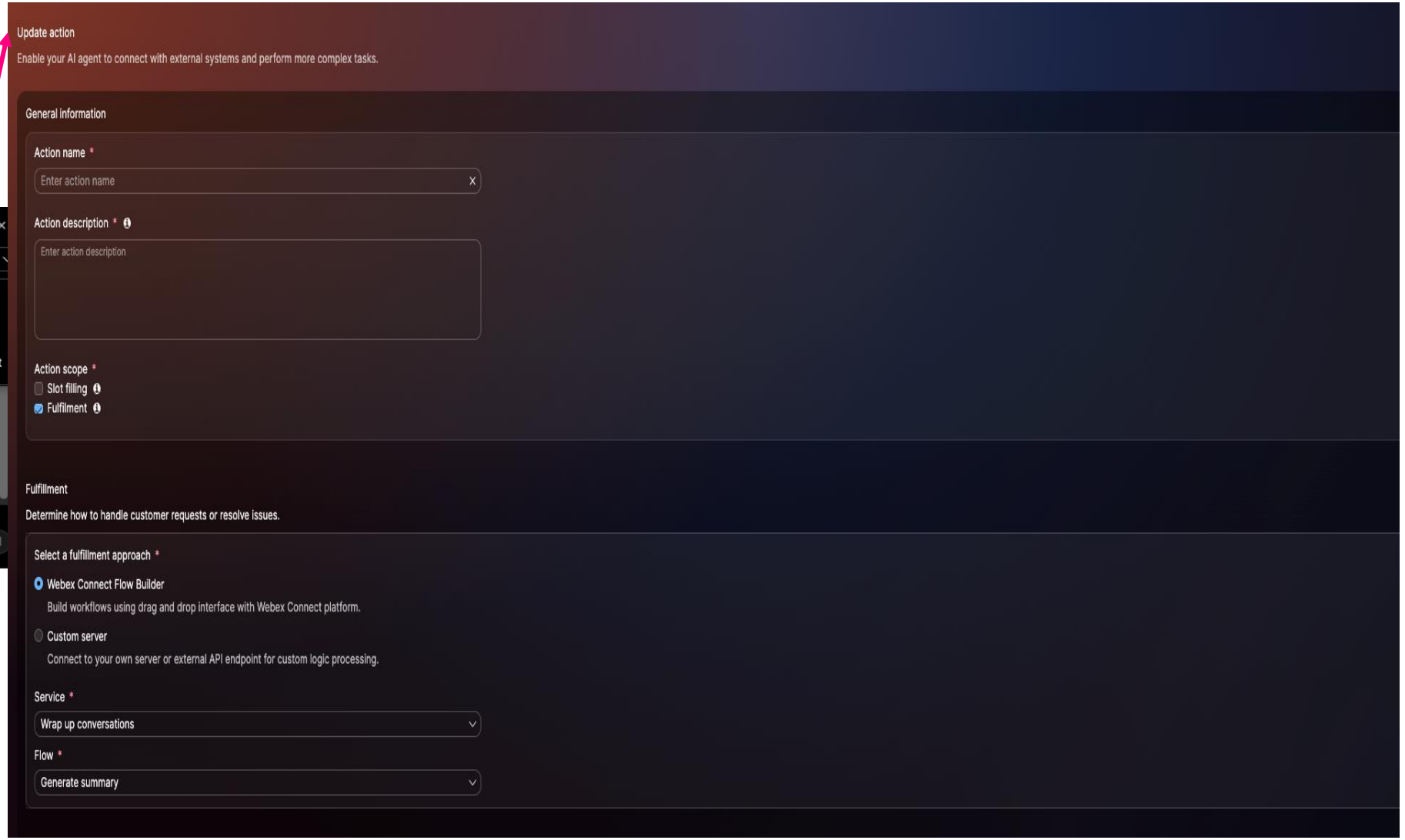

© 2025 Cisco and/or its affiliates. All rights reserved.



cisco

# New Webex AI Agent Studio: Create a new fulfillment




© 2025 Cisco and/or its affiliates. All rights reserved.

# Complete your session evaluations

**Complete** a minimum of 4 session surveys and the Overall Event Survey to claim a Cisco Live T-Shirt.

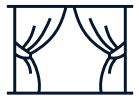
**Earn** up to 800 points by completing all surveys and climb the Cisco Live Challenge leaderboard.

**Level up** and earn exclusive prizes!

**Complete your surveys** in the Cisco Live Events app.

© 2025 Cisco and/or its affiliates. All rights reserved.

# Continue your education



**Visit** the Cisco Stand for related demos

**Book** your one-on-one Meet the Expert meeting

**Attend** the interactive education with Capture the Flag, and Walk-in Labs

**Visit** the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

© 2025 Cisco and/or its affiliates. All rights reserved.

# Thank you

© 2025 Cisco and/or its affiliates. All rights reserved.

CISCO