Communication flow for Microsoft Teams bot applications

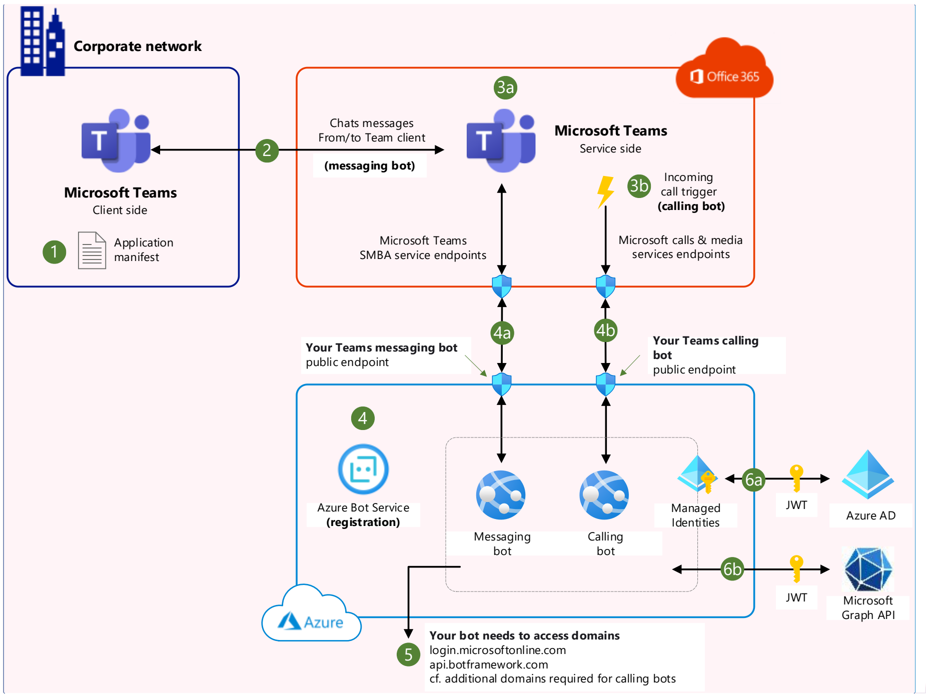
This article describes the communication flow and how the traffic is routed between a user in Microsoft Teams and a bot application running in Azure. This information will help you learn how to inspect the traffic and understand the user data flow in transit and at rest for bots in Teams.

This article covers two use cases that involve messaging bots and calling bots for Microsoft Teams. It answers frequently asked questions and provides best practices for how to integrate and secure the network connectivity for your Teams bots.

Note: This document applies to customers using Microsoft 365 worldwide or United States Government Community Cloud (GCC) – it does not apply to other cloud endpoints.

Architecture and data flow

The following diagram illustrates the communication flow between the Teams client and your bot application running on Azure.



The following sections explain the data flow shown in the diagram.

Commented [LG1]: You will need an H1 title for this document ; something like "Communication flow between Teams and Azure bot applications".

Commented [AK2R1]: Done

Commented [LG3]: Please explain this diagram using text. Images are not localizable and cannot be read by a screen reader. This makes using images to convey important information/concepts an accessibility and globalization issue.

Commented [AK4R3]: Done

1 – Teams app manifest

The Teams app manifest contains the definition of the bot capabilities and application ID registered in Azure AD. Teams users can only send and receive messages to and from your bot if they have the application installed directly (personal scope) in Teams or if the user belongs to a team, group chat, or meeting where the bot is also installed with permissions to read the rosters.

Commented [LG5]: Per branding, we can refer to the app just as Teams after first mention, FYI.

Commented [LG6]: This doesn't read like an actual step but more like it is providing the context. Should it be a paragraph instead of a numbered step? Or should it be rewritten to be an actual step, i.e. "A Teams user who has the application installed… or who belongs to… sends or receives…. ".

Commented [AK7R6]: You're right - We should replace "steps" by "talking points" (or "highlights") = "These are the technical requirements to consider to enable the communication with your Teams bot application"

2 – Teams client connection

The Teams client can run from any location or device (web, desktop, mobile) only if it has access to Microsoft Office 365 endpoints, as defined in Managing Office 365 endpoints. No extra IP, port, protocol, or FQDN is required to use bots in Teams.

3 - Bot messages and calls signaling transit via the Teams service

For messaging bots, chat messages are sent to and received from the Teams service, hosted by Microsoft (3a).

For calling bots, the Teams service sends the notification for incoming calls and provides the endpoints for the media streams, call signaling, and control plane (3b).

4 – Azure bot registration

The Azure Bot Service is required for the registration of your bot, including:

- The bot's name, description, and logo
- The supported authentication type (single-tenant, multi-tenant, or user-managed identities)
- The associated app ID and app registration in Azure AD
- Activated channels and bot endpoints
- Other settings like OAuth provider or public access

The Teams channel should be activated with the appropriate endpoints set for messaging bots and calling bots in your Azure Bot configuration. Note that the endpoints for Teams messaging (4a) and calling (4b) bots are configured independently and do not have the same requirements for network configuration.

Your bot application receives activities coming from the Teams service directly, not from the Teams client. For messaging bots, the Teams service provides a reply to URL in the form https://smba.trafficmanager.net/{region}, where region depends on the location for your Microsoft 365 service (for example, emea, amer, in, apac).

4a – Inbound/ingress rules for messaging bot

Source IP	Destination	Role	Destination port	Protocol
Teams service 52.112.0.0/14	Your messaging bot	Communication channel for activity messages	443	TCP

Commented [AK8]: TBC - IP range for
GCC-H:
52.127.88.0/21
and DoD:
52.181.167.113/32

Commented [AK9R8]: Added comment in the doc: we cover only WW & GCC cloud

4b – Inbound/ingress rules for calling bot

Source	Destination	Role	Destination port	Protocol
Teams service 52.112.0.0/14 and 52.120.0.0/14	Your calling bot	Real-time media port range for video/ audio stream	Configurable via SDK Minimal port range of 1000 ports. For example, 16000 - 17000	UDP
Any – Can be limited to Azure Cloud IP range via Azure Service tag. See Azure IP Ranges.	Your calling bot	Teams call signaling	Configurable via SDK Default: 443	TCP
Any – Can be limited to Azure Cloud IP range via Azure Service tag See Azure IP Ranges.	Your calling bot	Media control plane for Teams	Configurable via SDK Default: 8445	TCP

Commented [ZN10]: Which API/SDK are we refer to here? The Graph API for calling/meeting are exposed through Graph AGS service, PMA and MediaPaaS services. Instead of exposing Microsoft Teams Service(52.112.0.0/14). Are you refer to some other API/SDK, or should we change it to Graph interface?

For Graph interface we don't have public IP range available for whitelisting in customer side yet, the effort is in plan while no committed timeline. FYI, @Stephen Sulzer.

Commented [AK11R10]: Hi @Zheng Ni , @Stephen Sulzer the objective of this doc is to clarify the network connectivity to enable a calling bot for MS Teams. Ideally, we need a network diagram (like the one pasted at the end of this doc) and this table should reflect the IP/ports rules to set (IN/OUT) for each service (Graph AGS, Teams, ... or whatever makes sense to document)

Commented [AK12R10]: Done

5 – Access to domains

Your bot needs access to Microsoft services to do operations like validate the JWT token sent in the HTTP Authorization header or facilitate user single sign-on (SSO). Because the list of IP addresses can vary over time, we recommend that you implement FQDN-based filtering.

Outbound/egress rules for messaging bot

Source	Destination	Role	Dest Port	Protocol
Your messaging bot	login.microsoftonline.com	OAuth login URL	443	TCP
Your messaging bot	api.botframework.com	OAuth scope	443	TCP

Outbound/egress rules for calling bot

Source	Destination	Role	Dest Port	Protocol
Your calling bot	login.microsoftonline.com	OAuth sign in URL	443	TCP
Your calling bot	api.botframework.com api.aps.skype.com pma.teams.microsoft.com pma.cloud.microsoft	OAuth scope	443	TCP
Your calling bot	Skype for Business Online and Microsoft Teams endpoints defined in Office 365 URLs and IP address ranges.	Skype for Business Online and Microsoft Teams	Multiple	TCP & UDP

6 – Bot permissions on Microsoft Graph API

If your bot requires additional permissions to perform operations on your Microsoft 365 environment, you need to trigger an authentication flow to get the appropriate access token from Azure AD. A best practice is to implement user-managed identities; this simplifies and secures the management of application secrets. Messaging bots will generally use a delegated permission (on-behalf-of the connected user), whereas calling bots will require application permission to have control over the call (hang up, redirect, join participants, access the audio stream).

Your bot needs access to the **graph.microsoft.com** domain to query the Microsoft Graph API (required for calling bots; optional for messaging bots, depending on the use case).

Scenario details

Bots allow Teams users to interact with web services through text, interactive cards, and task modules. The Microsoft Bot Framework and Azure Bot Services give you an easy-to-use set of tools for creating and managing these bots.

You can develop bots by using a variety of languages, such as C#, JavaScript, and Python. After you develop your bots, you can deploy them to Azure. A key component of a bot is the web app, which contains the core logic and interface that users communicate with. One of the key requirements for the bot to work is that it must expose a publicly accessible HTTPS endpoint.

InfoSec policy commonly requires that all incoming traffic to web apps go through a corporate firewall. This means that all traffic that goes to a bot, and responses from the bot, must route through a corporate firewall, as with any other web app.

Potential use cases

Teams messaging bots (conversational bots)

Messaging bots are used to implement chat-based interaction between a user in Teams and your bot. This is usually a two-way communication channel (user sends a chat message to the bot and gets an answer) but can be configured for notification only (bot sends message but user can't query or answer). For details about how to create a conversation bot, see Create a Teams conversation bot.

Teams calling bots

Calling bots are used to implement voice-based interaction between a user in Teams and your bot. The bot will also be able to answer an incoming call, join a call, and manage its lifecycle. Calling bots are also used for compliance recording in regulated industries. For details about how to create a calling bot, see Calls and online meetings bots and Compliance recording for calls and meetings.

Organizations can use bots for mobile and desktop users. Some examples include:

- Simple queries. Bots can deliver an exact match to a query or a group of related matches to help with disambiguation.
- Multi-turn interactions. By helping anticipate possible next steps, bots make it much easier for people to a complete task flow.

- Reaching out to users. Bots can send a message (notification) when something has changed in a document, or a work item is closed.
- Bots can be integrated in multiple ways into Microsoft Teams: as a personal application, in a channel or group chat, as a message extension (to easily search and share data), or in a meeting.
- Calling bots are a specific use case enabled for Teams where the bot can respond to incoming calls, manage participants, process audio and video media streams, and more.

Frequently asked questions (FAQ)

Does user data (such as chat messages) transit via the Azure Bot Service with Microsoft Teams channel?

No. No user data transits via the Azure Bot Service for the Teams channel (both for the messaging and calling endpoints). For first-party channels such as Teams, Outlook, Skype, Search (Preview), and Direct Line Speech, user data goes directly to the Microsoft service endpoint and does not transit via the Azure Bot Service.

How does user data transit from the Teams client to the bot application?

For first-party channels such as Teams, user data transits via the Microsoft 365 location that you configured during the provisioning of your services. For details, see Where your Microsoft 365 customer data is stored.

Can we disable public access and use private access for bots in Teams?

No. Teams is SaaS (software as a service) and only provides public endpoints that Teams clients need to join. Disabling public access is only supported in combination with Direct Line App Service extension and is not supported for Teams.

Can I activate Azure AD tenant restrictions with the Azure Bot Service?

Yes. With tenant restrictions, organizations can specify the list of tenants that users on their network can access. Azure AD then only grants access to these permitted tenants - all other tenants are blocked, even the ones that your users may be guest members of. For details, see Restrict access to a tenant.

For your bot application, and bot users, to be able to authenticate on the Azure Bot Service, your proxy server needs to add the following tenants to the allow list:
- botframework.com if the Azure Bot Service is configured for multi-tenant.
- Your own company tenant (for example, contoso.com) if Azure Bot Service is configured for single-tenant.

Can we host a bot for Teams outside of Azure?

It depends on the scenario, as follows:

- **Messaging bots** can be hosted on any infrastructure if all required FQDN, IP addresses and ports (in and out) are on the allow list.

Commented [AK13]: @Laura Graham , I see that you change "whitelist" to "allow list" - "IP whitelisting" is a term commonly used when related to IP filtering. I don't know if "allow list" is more appropriate here.

Commented [LG14R13]: We can't use the term "whitelist" in any Microsoft documentation for reasons of diversity and inclusion: whitelist - Microsoft Style Guide | Microsoft Learn

I'm open to rephrasing if you have a different suggestion, although allow list might be clear enough, in any event.

Commented [AK15]: @Laura Graham, I see that you change "whitelist" to "allow list" - "IP whitelisting" is a term commonly used when related to IP filtering. I don't know if "allow list" is more appropriate here.

Commented [LG16R15]: Yes, we just need to avoid the term "whitelist". There should be a movement across the industry to avoid the terms whitelist and blacklist.

Commented [AK17R15]: OK - Thank you for the explanation - I'm fine with allow list

- **Calling bots** can only be hosted on Microsoft Azure and specific services. For details, see Requirements and considerations for application-hosted media bots.

Next steps

- Build bots for Microsoft Teams
- Connect a bot to Microsoft Teams
- Register a calling bot for Microsoft Teams
- Working with the cloud communications API in Microsoft Graph

Related resources

- Azure Architecture Center
- Azure and Microsoft 365 scenarios
- Office 365 URLs and IP address ranges
- Help secure your Microsoft Teams channel bot and web app behind a firewall