4400 NE 77th Ave, Suite 275,
Vancouver, WA, 98662
United States

Security Overview for Collaborate

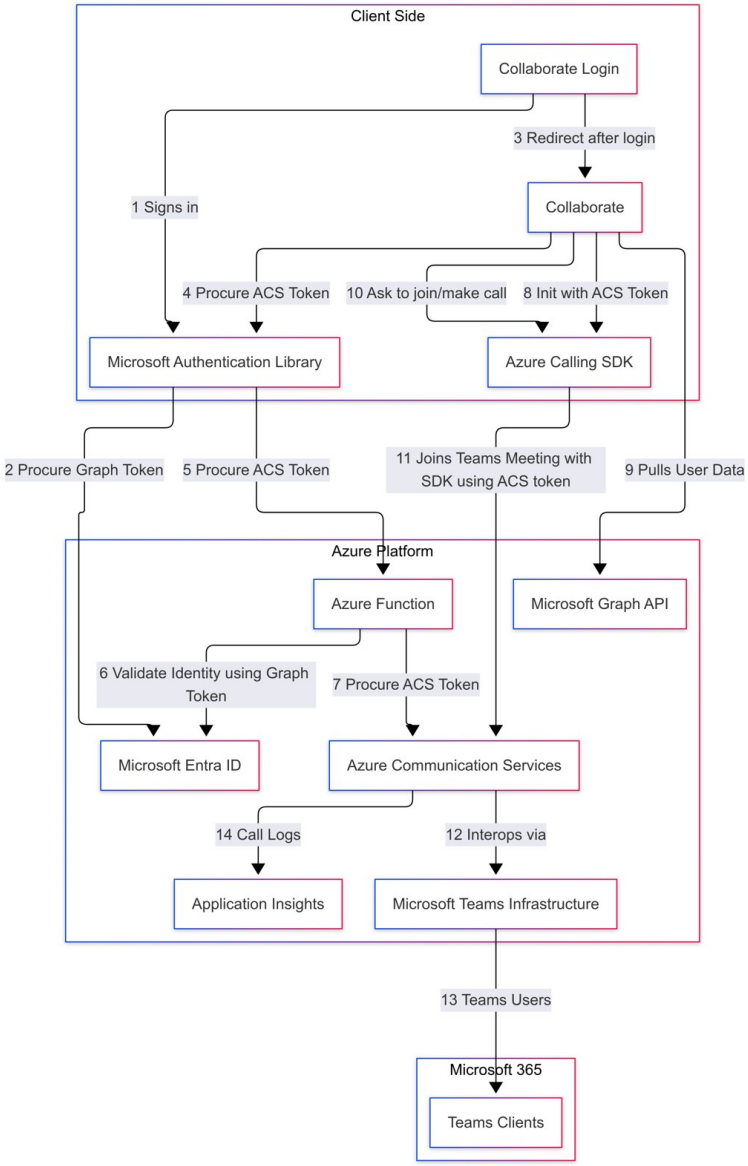
Executive Summary

Collaborate is a voice-controlled Microsoft Teams client designed exclusively for RealWear headsets. Leveraging Azure Communication Services (ACS) in the backend, this app enables seamless hands-free Teams communication for frontline workers. With its RealWear-centric interface, Collaborate ensures effortless access to meetings and calls using just your voice, maximizing efficiency and safety in industrial environments. Whether on-site or in the field, stay connected and collaborate with your team, all while keeping your hands and vision focused on the task at hand.

This document aims to describe what measures were & are taken in regards to ensuring a fully secure end user experience.

4400 NE 77th Ave, Suite 275,
Vancouver, WA, 98662
United States

Architecture Overview



Client Side Components

1. Signs in — The user initiates the login process via the client interface.

2. Procure Graph Token — The client uses MSAL to get a token for Microsoft Graph API access, which allows it to query user information and permissions.

3. Redirect after login — After authentication, the user is redirected to the appropriate app view (Collaborate main screen).

4. & 5. Procure ACS Token — The client requests a communication token for ACS via a MS secured backend (Azure Function).

8. Init with ACS Token — The Azure Calling SDK is initialized with the ACS token to prepare for a call or meeting.

9. Pulls User Data — Using the Graph token, the app queries user data from Microsoft Graph API (e.g. calendar, Teams meetings) being limited by the user-granted granular permissions.

10. Ask to Join/Make Call — The user action triggers the client to join or initiate a call using the ACS infrastructure.

11. Joins Teams Meeting with SDK using ACS token — The Azure Calling SDK joins the Teams meeting, authenticated via ACS.

Components:

- **Collaborate Login / Collaborate:** Custom UI components for login and post-login collaboration interface.
- **Microsoft Authentication Library:** Manages the authentication process and token acquisition.
- **Azure Calling SDK:** Handles VoIP/video call setup via ACS.

Azure Platform Components

6. Validate Identity using Graph Token — Azure Function uses the Graph token to validate the user against Microsoft Entra ID (Azure AD).

7. Procure ACS Token — MS secured backend issues an ACS access token after validating the identity.

12. Interops via — ACS interops with Microsoft Teams infrastructure to enable calling into Teams meetings or users.

14. Call Logs — Anonymous call data (e.g. call duration, QoS) is logged to Application Insights for monitoring and troubleshooting.

Components:

- **Azure Function:** Microsoft intermediary backend for token generation and identity validation.
- **Microsoft Entra ID:** The new name for Azure Active Directory; used for identity services.
- **Azure Communication Services:** Provides calling/chat features and interop with Teams.
- **Microsoft Teams Infrastructure:** Native Microsoft Teams backend services for call routing and collaboration.
- **Application Insights:** Microsoft Azure service used for telemetry and diagnostics.

Microsoft 365 Ecosystem

13. Teams Users — The end recipients of the call/chat session — native Microsoft Teams users.

Components:

- **Microsoft 365:** The broader ecosystem encompassing Teams, Exchange, SharePoint, etc.
- **Teams Clients:** Native desktop, mobile, or web-based Microsoft Teams clients.

Authentication & Authorization

As can be observed from the architectural diagram, all authentication & authorization is handled directly or indirectly via a Microsoft provided service. This way we ensure adherence to all of the standards of security Microsoft adheres to.

Data Handling & Privacy

Thanks to the tight integration with Microsoft services and SDKs, Collaborate's data flow benefits from enterprise grade encryption in transit.

Privacy being a fore-most concern, we have made sure the Collaborate client app is limited, by design, in the data it has access to on the user's Microsoft Azure tenant. Read below regarding granular permissions.

As for local storage policies, the Collaborate app caches contact information on-device for low-latency look-ups and ease of use. However, the stored contact list is deleted upon uninstalling the app.

Security in Software Development Lifecycle (SDL)

The only data that is impacted by RealWear practices is on-device data. Our SDL takes such measures as:

- security-conscious feature requests
- security code reviews
- integrating security best practices into QA test cases (e.g. validating cache removal upon uninstalling the app)
- organizational pipelines ensuring QA approval before changes take effect
- use of trusted, Microsoft-hosted services for internal communication & organization

Deployment & Infrastructure Security

Permissions

All permissions requested are granular and Delegated Style permissions: Microsoft identity platform delegated access scenario - Microsoft identity platform

Granularity ensures that the end user may pick and choose which permissions to enable on their Microsoft Azure tenant such that their specific security concerns

4400 NE 77th Ave, Suite 275,
Vancouver, WA, 98662
United States

are satisfied. Below is a table describing the rationale behind each and every permission and how these map to Collaborate's features:

Permission	Reason Why
https://graph.microsoft.com/Calendars.Read	Viewing Calendar events to show in the Calendar options
https://graph.microsoft.com/User.ReadBasic.All	Read user contacts, loading user profile images
https://graph.microsoft.com/Group.Read.All	Read user groups and calling groups
https://graph.microsoft.com/Presence.ReadWrite	Set user status to Online / Away / Busy, depending on their status
https://auth.msft.communication.azure.com/Teams.ManageCalls	Obtaining Azure Communication Token, allowing the signed in user to initiate, accept and join calls.
https://auth.msft.communication.azure.com/Teams.ManageChats	Used exclusively for call functionality, the ManageChats permission is unused but a prerequisite for call functionality. Information from Microsoft can be found here
api://app.realwearteams.com/32dfa5db-1bf5-4c70-bb8b-ab2fe987bf2b/access_as_user	The 'user_impersonation' permission enables access to a trusted authentication service, ensuring secure communication token exchange. Learn more from Microsoft here
https://login.microsoftonline.com/organizations/v2.0/adminconsent?client_id=32dfa5db-1bf5-4c70-bb8b-ab2fe987bf2b&scope=https://graph.microsoft.com/Chat.ReadWrite	Required for 'Take Picture' to work.

Firewall setup

Following is a comprehensive external dependency list meant to offer a security overview of and help with setting up Collaborate:

- RealWear Meeting Demo - for the "Connect me" code and participating in meetings without the "Teams.ManageCalls" permission
- https://login.microsoftonline.com/ - for user authentication (MSAL)
- Microsoft Graph Dev Center | APIs and app development - for syncing data of the authenticated user
- https://realwear-teams-token.azurewebsites.net/api/Function - for issuing a communication token for authenticated user

- Range of Azure public cloud IP addresses 20.202.0.0/16. The range provided above is the range of IP addresses on either Media processor or Azure Communication Services TURN service; Ports UDP 3478 through 3481, TCP ports 443 - for MediaTraffic in Azure Communication Services
- *.skype.com, *.microsoft.com, *.azure.net, *.azure.com, *.office.com; Ports TCP 443, 80 - for signaling, telemetry, registration in Azure Communication Services
- https://api.lokalise.com/ and https://ota.lokalise.com/ - for Lokalise translations (however there is no dedicated firewall setup page in their documentation)
- Firebase Analytics app-measurement.com - Firebase general app analytics (anonymized)
- Sentry - Sentry general app crash reporting (anonymized)
  - o4504485976014848.ingest.us.sentry.io — Application Performance Monitoring & Error Tracking Software
- Data dog - Temporary log collection tool (anonymized)

Compliance & Standards Alignment

Microsoft Teams is designed and developed in compliance with the Microsoft Trustworthy Computing Security Development Lifecycle (SDL), which is described at Microsoft Security Development Lifecycle (SDL). This, of course, includes the Microsoft-backed infrastructure we make use of in our custom client. Thus, Collaborate's backend architecture inherits compliance posture from Microsoft Azure and ACS (ISO 27001, SOC 2, GDPR, HIPAA).

Known Limitations / Risk Acknowledgment

Given the security overview provided above, the following should be taken into consideration as limitations and risks that Collaborate is open to:

- **Physical Access to the Device —** Collaborate runs on proprietary hardware, which includes OS-level protections such as secure boot and encrypted storage. However, if an attacker gains physical access to the device, risks such as tampering, credential harvesting, or side-channel attacks could emerge. Devices should be deployed in controlled physical environments and managed with appropriate asset protection policies.
- **Social Engineering Attacks —** While Collaborate enforces secure authentication and authorization via Microsoft Entra ID and ACS, it cannot prevent users from being tricked into sharing credentials, tokens, or access through phishing, impersonation, or other social engineering tactics. Users should be educated and trained on secure practices.
- **Shared Credential Misuse —** Although authentication tokens are securely stored in-memory, if users share accounts or login credentials (against policy), this undermines individual accountability and may allow unauthorized access.

- **Local Network Exposure —** Collaborate assumes that the device has access to the internet or a corporate network. If deployed on unsecured or open networks, there is a potential risk of man-in-the-middle (MitM) attempts, although TLS encryption mitigates most of this risk. Device network configurations should restrict traffic to approved endpoints only.
- **Third-Party Dependencies —** The solution depends on Microsoft services (e.g., ACS, Graph API, Entra ID). While these are highly secure and compliant, any vulnerabilities or outages in these services could affect application behavior or data confidentiality beyond our direct control.
- **User Misconfiguration —** Incorrect configuration of Azure AD roles, permissions, or ACS usage scopes may inadvertently expose sensitive functionality. Customers are encouraged to follow Microsoft's best practices for tenant-level security hardening and RBAC.

Appendices

Read more about permissions and how an organization admin can set these up here: Collaborate Technical Info

Support Contact

support@realwear.com

Support: support.realwear.com/knowledge/collaborate-technical-info

Terms of use: realwear.com/legal/collaborate-terms

Privacy Policy: realwear.com/legal/collaborate-privacy-policy