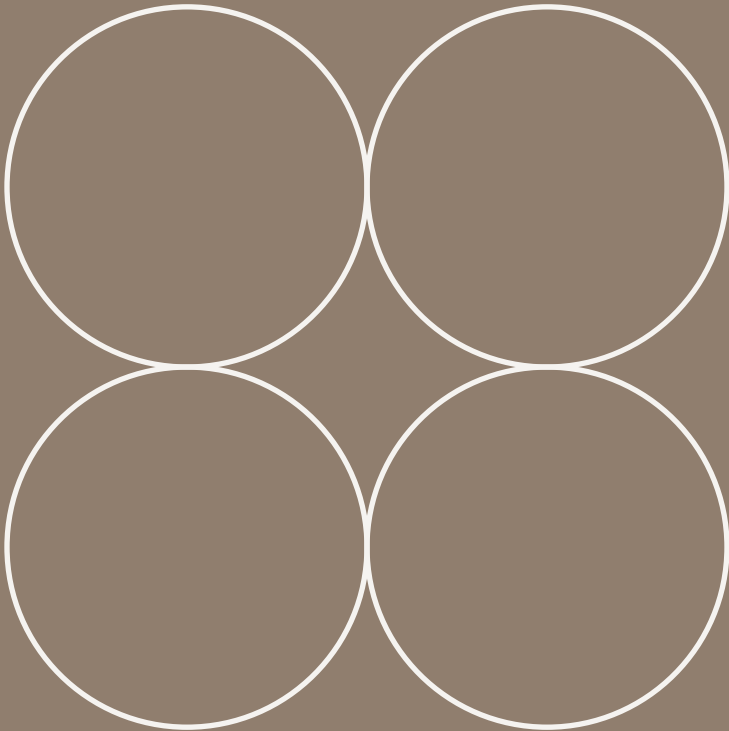


# Interactions API buyer's guide: chat, voice, and video



# Table of contents

- 01** Review : what is an interactions API?
- 02** The benefit of an interactions API : chat, voice, and video
- 03** Why an interactions API?
- 04** Project scope : implementing chat or voice and video an API
- 05** Interactions API features
- 06** How does the interactions API fit with your technology?
- 07** Services for implementing chat, voice, and video
- 08** How is chat, voice, and video priced?
- 09** Executing a successful project rollout for chat or voice, and video
- 10** FAQs

# Review: what is an interactions API?

Twenty-first century digital interactions include large- or small-scale group chat, 1:1 messaging, voice and video calls, all sent using internet protocol (IP). An **interactions API** is a suite of API products that give your business a simple way to customize fully featured **chat or voice and video calls** into your brand's application, and delegate the management of server traffic generated by this service to a third party.

The interactions API provides a RESTful program interface to allow your product and development teams access to every functionality required to build chat, voice, and video, allowing them to customize a communication experience based on simple HTTP requests and responses. Software development kits (SDKs) package the API to help you implement the client application quickly without having to start from scratch. In the end, **it enables your end users to chat and make 1:1 voice and video calls from your application.**

# The benefits of an interactions API : chat, voice, and video

## **Forward-thinking businesses of all types benefit from an interactions API:**

marketplaces, on-demand services, digital health, live-streaming apps, social media, online communities, and gaming.

## **Depending on how a business deploys chat, voice, and video, it can benefit them in many ways:**

### **Benefit | Increase**

- Gross merchandise value (GMV)
- Gross transaction value (GTV)
- Sales conversion
- User engagement in app
- User retention in app
- Resolution of customer inquiries
- Immersive user experience

### **Benefit | Decrease**

- Booking cancellations
- Abandoned carts
- Time to resolve customer inquiries

# Why an interactions API?

Digital interactions have only become more crucial in 2020, when this mode of communication has proven that social distancing and quarantine strengthened the importance for people to connect online. Over 5 billion people chat worldwide through a messaging app. Over 90% of users agree that video calls improve connectedness.

Consumers are already adapted to modern chat, voice, and video experiences and take them for granted as the standard experience. They not only want to send and receive messages, but they also want to send previewable images, emojis, GIFs; they want crisp voice, clear video, and no interruptions; different kinds of moderation features; and the list goes on. Without an interactions API, it would be a serious challenge for a business not focused on chat, voice, or video to create a product that could compete with interaction experiences that 80% of adults use everyday.

## Why a chat API is necessary to deliver high-quality, customizable chat

Take a chat's read receipts as an example. It's a must-have feature for chat. This seemingly simple feature requires that a server keep every user's read status for every conversation forever. And this read status must synchronize to other users' devices when they're online. And it must be updated for every message from each member of the conversation.

All told, the chat system needs to be tested to handle 10-100x more event traffic for every message sent because of this one feature.

Or consider building a chat system that reliably delivers messages during poor connectivity or when some recipients' devices are temporarily offline. Or during spikes in server traffic, when your users jump from 500 to 5,000 or more sending messages, images, and reactions in multiple channels at once. Compounded with the other features layered over a simple message, like the read receipt, delivering a smooth end-user experience quickly becomes untenable.

## **Why a voice and video API delivers an immersive call experience so quickly**

In-app voice and video calling also presents a number of challenges for developers. It requires a specialized networked infrastructure not only to connect calls, but also to process and relay the audio and video streams between thousands of concurrent call participants. Quality poses a challenge too. This interaction must approximate the standard mobile phone calling experience as much as possible: incoming call notifications (even when the app is not open), caller ID, mute, and the ability to select a microphone, speaker, and camera.

Outside of the call experience, the developer faces other challenges like managing a user's call history, enabling redial, and providing a layer of analytics based on individual and aggregate usage. Delivering this level of functionality requires an enormous attention to detail, extensive testing, and continual operational monitoring, all of which consume valuable

human resources from a team for whom in-app voice and video calling is a valuable — but ancillary — feature.

## **Interactions API emerges as the solution for quality, scale, speed, and flexibility**

Beyond high standards for chat, voice, and video, it is challenging to build server architecture that scales with each new feature and your user growth, or keeps up with client version changes or security vulnerabilities — not to mention compliance, performance optimization, maintenance, synchronizing data across iOS, Android, and JavaScript platforms.

With an interactions API, businesses can now create the highest quality chat, voice, and video experience and scale quickly without having to worry about server maintenance, performance, or security. The flexibility of an API allows development teams to customize the service's functions and features to meet their needs or even use the interactions API as a platform to build the entire communication journey across a user's lifecycle. SDKs help with client implementation so flexibility is balanced with a speedy time to market — for example, you can make your first in-app voice call in as little as 15 minutes.

As a result of all these technical challenges, the interactions API emerged as the solution to reliably create the highest quality chat, voice, and video experiences at scale with minimum time to market and maximum flexibility.

# Project scope: implementing chat or voice and video using an API

The project scope for implementing chat, voice, and video will vary according to your definition of a minimum viable product (MVP). To implement chat or voice and video with all the necessary features for basic functionality, expect a timeline as fast as a few hours or a few days depending on your developer's experience and bandwidth.

## Project scope: implementing chat using an API

For most customers, an MVP equates to implementing chat in their application with necessary features like creating users and channels, displaying channels, and sending and receiving messages. This could take anywhere from a few hours to a few days, depending on developer experience and bandwidth, especially given good documentation and fully functional sample apps for each platform. For other customers, an MVP might include every function required to ship.

Whatever goals you set for an MVP, the beauty of an API is that it can get you to market quickly so you can provide a proof of concept or demonstrate initial business outcomes to justify a bigger investment in your roadmap.

The following gives you a range of timelines for different project scopes.

Few hours to a few days	→ 1-3 weeks	→ 1+ month
<b>MVP: Just show us it works</b>	<b>MVP: Fully functioning chat</b>	<b>MVP: Ship it</b>
<b>Last effort basic function</b> <ul style="list-style-type: none"> <li>• Create user</li> <li>• Create channel</li> <li>• Display channels</li> <li>• Send messages</li> <li>• Receive messages</li> </ul>	<b>Medium effort also includes</b> <ul style="list-style-type: none"> <li>• Read receipts</li> <li>• Typing indicators</li> <li>• Offline caching</li> <li>• Push notifications</li> <li>• Thumbnails-images, video, GIFs</li> <li>• More typical chat features</li> </ul>	<b>Most effort also includes</b> <ul style="list-style-type: none"> <li>• Full backend functionality</li> <li>• Customization with Webhooks</li> <li>• Moderation customized, etc.</li> </ul>
Basic UI, using sample app	Custom UI	Everything to ship it

## Project scope: implementing voice and video using an API

For customers using a voice and video API, an MVP tends to mean basic functionality like making a call, receiving a call, ending a call, caller ID and logs, and using contact lists — all across multiple platforms. While Sendbird Calls allows you to make your first test call in 15 minutes, you can develop a high-quality and fully functional MVP for in-app voice and video in at least half the time it takes to build from scratch. Since the goal of voice and video is simpler than chat, you can integrate it into your app’s UX with few modifications.



Without an API and managed infrastructure, however, the MVP is the smaller effort. The greatest effort will be dedicated to load testing for a large scale of concurrent callers. With a managed infrastructure, you can avoid this effort altogether.

A voice and video API and managed infrastructure, not only accelerates MVP development, but it also scales to your needs right away without additional effort.

# Interactions API features

Beyond the MVP, a chat or voice and video API provides other features that can complement your application's business logic or help you reach specific business outcomes, improve user experience, or develop more insight into your users' experience.

## Chat API features

### Messaging features

- Typing indicators
- Read receipts
- Invitations
- Chat history
- Video and image thumbnails
- Send and receive structured media
- Emojis and reactions
- GIFs
- URL previews
- Rich text editing
- Push notifications

### Moderation features

- User-to-user blocking
- Smart throttling
- Auto-moderation

- Image moderation
- Regex and profanity filtering

### **Data and security**

- Advanced analytics
- Encryption in transit and at rest

### **Integration**

- Link API services — maps or payments APIs directly in the message

## **Voice and video API features**

### **Call features**

- Make a call
- Receive a call
- End a call
- Caller ID
- Contact list
- Display a user's call log in the application
- Push notifications for incoming calls
- Multi-device support
- High quality and performance across platforms: iOS, Android, React Native, JavaScript Link voice and video calls to a specific Sendbird chat channel

## **Data and security**

- Call metadata for deeper insights: topic, time, context
- Manage and view call logs in a detailed view, including metadata
- Enable test-only token-less user authentication
- Encrypt call metadata transfer
- Implement standard user login credential policies

# How does the interactions API fit with your technology?

The simplicity of an interactions API allows you to use it flexibly with your code. An interactions API can be boiled down to four different HTTP methods — POST, PUT, DELETE, GET — working in tandem with a server infrastructure. An interactions API also defines the structure of the responses in JSON or XML returned from these HTTP methods. These responses generalize data sent from the servers so they can be as flexible as possible. Using these responses, you can customize how you make use of the interactions API.

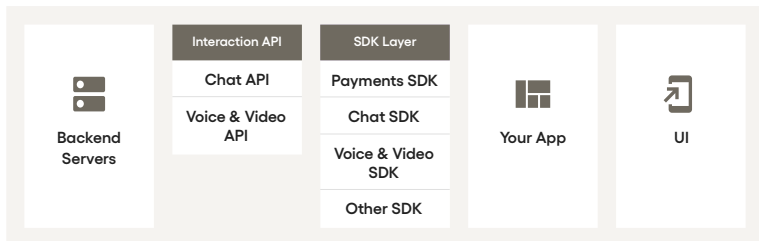
Although each feature uses these simple HTTP methods, the functional implementation of chat, voice, and video is significantly more complex. For example, if you create channel handlers that notify your servers when a user edits a message or if you retry sending messages or making a call during poor connectivity, each has a complex implementation built on the backbone of those four HTTP methods and their responses.

A chat or voice, and video SDK manages the complex implementations for you, while the API creates flexible chat functionality based on HTTP methods. An API communicates with the backend servers, telling them to create new users, generate thumbnails for images sent in a message, or handle reconnection when connection is lost. The SDK, in turn, “wraps” every API call so you don’t have to worry about making them and, then, it returns a response. The chat and voice and videos SDKs sit in parallel with

all your SDKs, so you can use many at once.

For example, instead of sending a message and continually polling our server to get the message's status or handling when a user sends a new message while the other is sending, the SDK will automatically handle each case and notify your app when the event occurs.

When the SDK notifies your app of an event, you simply decide how to display it to your users. SDKs remove the difficulty of chat, voice, and video, allowing you to decide how your users will experience each interaction.



This figure summarizes how an interactions API and its backend servers fit into your stack. At the outermost layer are the backend servers. This is accessible via the chat API and voice and video API. At the API layer, you can do everything necessary for chat, voice, and video like create users with access tokens and channels, register push tokens, and make or receive calls, etc. Next, the API layer is conveniently wrapped by the SDK layer, which allows you to easily and smoothly implement chat, voice, and video in your app. The SDK sits in parallel with all your other application's SDKs in the SDK layer. From there, all you need to do is call the requisite methods and display them in your UI.

# Services for implementing chat, voice and video

There are a range of services and tools that allow you to implement chat, voice, and video in your product and they vary according to how much you need to build to achieve the interactions experience you want. These range from app development and data streaming platforms to SaaS (Software as a Service) tools without any customization. The former are designed for general mobile application development. They tend to offer the most basic functionality and require significant feature building and optimization. The latter, on the other hand, offer little to no flexibility.

In the middle, the API occupies the sweet spot for flexibility and fast development because it provides all the features required for interactions in a RESTful API and client-side SDK, so you can quickly launch and customize your interaction experience. By specializing in chat, voice, and video the API service gives you the technology to customize chat, voice, and video calls in your app according to your specifications.

## **These are the three main categories of tools on the market for interactions platforms:**

- App development and data streaming platforms
  - Typically a managed service that provides a broad platform for building mobile applications and syncing data in real time.

- APIs
  - An API and managed infrastructure to customize chat, voice, and video in your app, and typically including an SDK to launch quickly.
- Software as a Service (SaaS)
  - An out-of-the-box solution suited to departments like marketing and sales that require little to no technical or UI customization.

BUILD   BUY		
App Development & Data Streaming Platforms	Interactions API	SaaS
<ul style="list-style-type: none"> <li>▪ Flexible mobile app development platform</li> <li>▪ Some out-of-box messaging features and basic voice and video</li> <li>▪ Significant effort to develop many individual features</li> <li>▪ Challenge to scale built features</li> <li>▪ Charges for API calls or function usage in addition to other dimensions</li> </ul>	<ul style="list-style-type: none"> <li>▪ Flexible API to integrate chat, voice, and video into your product</li> <li>▪ Fully featured chat, and 1:1 voice and video</li> <li>▪ Customization, webhooks, link multiple APIs</li> <li>▪ Managed infrastructure scales for all features</li> <li>▪ Monthly active users (MAU) or minutes are primary driver for price — premium features, support, and storage added</li> </ul>	<ul style="list-style-type: none"> <li>▪ Out-of-the-box product with minimal integration required</li> <li>▪ Integrations focus on business-to-consumer, marketing, or pre-sales interactions</li> <li>▪ Little to no customization</li> <li>▪ Tends toward low scale use cases</li> <li>▪ Price varies per/seat, per/month, and price adds per feature</li> </ul>

## Spotlight on voice and video APIs

APIs have emerged as the preferred service for in-app voice and video integration. It's helpful to organize the landscape according to two variables: price and level of effort.

### **Lowest level of effort and medium price**

The Sendbird Calls API minimizes the level of effort to implement in-app voice and video. Invested in a streamlined developer experience and thorough documentation, Sendbird Calls API allows you to develop your app with the least amount of effort. Its pricing sits squarely in the middle and appeals to businesses that want to integrate high-quality voice and video with minimal effort or modification to your UX.

### **High level of effort and high price**

These services are the long-established telephony companies or communications APIs, like Vonage or Twilio. They are priced at a premium and require a high level of effort, but offer their longer-established brands and high customization.

### **Medium effort and low price**

Agora.io, Sinch, and Plivo sit in this tier, offering a low price and moderate effort to implement. Plivo only offers voice over telephony networks. Sinch offers a number of communication APIs. Agora.io's original product is voice, video, and other related products like video or audio broadcasting.

# How is chat, voice and video priced?

## Pricing for chat

The primary driver for price in chat is the number of monthly active users (MAU). This variable changes the most over time as you grow in-app chat.

### **Other factors include:**

- Premium features
- Data storage
- Different support levels

SaaS vendors typically charge per seat per month, add to price for additional features, and set limits on API calls. This software is typically used for marketing and sales and follows well-worn pricing models for SaaS.

## Pricing for voice and video

**Companies price voice and video on a per minute basis. The following variations on price also exist:**

- Voice is cheaper than video by nearly half or more
- Companies also occasionally vary the price of voice calls, depending on whether you're receiving a call or making a call (the more expensive of the two)
- The price of video sometimes depends on the quality of video, if options are available
- HIPAA compliance often increases the price of both voice and video
- As volume increases, companies offer cheaper rates per minute for both voice and video

# Executing a successful rollout for chat, or voice and video

## Planning your implementation, benchmarking, and analytics

The most fundamental goal is to understand how chat or voice and video integrates into your main product. In other words, how do you link chat or voice and video to your business, product, and UX goals? You may have a clear idea of the greatest areas of friction in your user experience or you may only have a hunch. Either way, thinking through your implementation goes a long way. We always recommend complementing your own planning by consulting with your interactions API provider about industry and use case best practices.

Every customer of a chat or voice and video API wants their users to talk and, to some extent, greater user engagement is the most fundamental result of implementing messaging. At what points in your user lifecycle or user journey will more engagement either counteract churn or drive the user to continue the journey? Here's a basic but proven example.

An on-demand ride-hailing app sees a lot of cancelled transactions two minutes after a user books a transaction. By giving your users the opportunity to communicate about the transaction before cancelling, you accomplish two things:

1. Create insight into what users talk about before cancelling — e.g., is the driver unable to find the rider or does the rider need another two

minutes?

2. Dramatically decrease booking cancellations by enabling users to communicate about and solve their own issues.

Whatever your goals, set a benchmark and KPI for the issue before implementing chat and voice and video in your app. For example, how many users cancel:

- 0 - 60 seconds after transaction
- 60+ seconds after transaction
- After an SMS
- For specific reasons like the driver cannot reach the rider or other reasons

Once you implement chat, the first goal is to achieve user engagement. This can be measured by growth in messages sent, MAU, messages per user, or messages per transaction, and others. Then, measure against your benchmarks using different types of analytics for different use cases.

For voice and video, the experience is more personal and immersive, but the goals are similar: achieve high engagement and Net promoter score (NPS) along your user's communication journey, leading to increased conversion on your specific business, product, or UX goal. You can measure growth in connected calls, minutes per call, time to resolution, and NPS, among other KPIs.

### **Chat moderation**

Every chat will need some form of moderation to protect its users. It's important to plan ahead: what kind of moderation is appropriate for your

user community?

Do your users share a lot of content? You may want to implement community moderators, user-to-user blocks, and other user-driven moderation.

Do your users mainly chat during transactions? You may need automatic profanity filters, regular expression filters, SPAM filters, or automatic image moderation.

### **Other technical considerations**

- When your app connects to a chat API the backend is extremely important because it relates to your MAU. Does the chat API provider have a connection manager in its SDK? If not, you'll need to build one and determine optimal connection logic.
- What type of authentication fits your security style? Typically, access tokens and session tokens are available for authentication. Use whichever fits your security style.
- When to implement basic functions like starting or archiving a channel. Be sure that your chat API provider can consult you on crucial implementation steps like this.

# FAQs

## **What is an interactions API?**

An interactions API is a simple way for you to customize a fully featured and branded chat, voice, and video into your application or product, and delegate the management of server traffic generated by this service to a third party. Using simple HTTP requests and responses, which contain data generalized for developer use, and an SDK, a Chat or Voice & Video API is a tool to implement an impressive interactions system quickly so you can enable users to chat or make voice and video calls without leaving your app.

## **What are the benefits of an interactions API: chat, voice and video?**

The benefits of a chat and voice and video API will depend on your industry and business model, but some of the many benefits include:

- Increased gross merchandise value (GMV) or gross transaction value (GTV)
- Decreased booking cancellations or abandoned carts
- Increased sales conversion
- Dramatically increased user engagement and retention in your application
- Faster time to resolution for customer inquiries
- Improved user experience

## **Why an interactions API?**

Over 5 billion users and 80% of adults chat every day, so they have high expectations for chat services. Over 90% of voice and video call users believe it improves connectedness. Building a chat or voice and video component for a business's application to meet consumer expectations is prohibitively difficult — not least because adding a simple feature like read receipts to a message can increase your event table 10 - 100x and voice and video requires specialized networked server architecture to connect, process, and relay calls.

APIs have emerged as “Goldilocks” technology that provides every feature you need for chat and voice and video with the right amount of flexibility to build it the way you want.

## **How long does it take to develop an MVP?**

Most of our customers develop MVPs to demonstrate a proof of concept. Depending on developer experience and bandwidth, it can take as little as a few hours to a few days to develop an MVP that demonstrates that chat works. If the specifications of your MVP require more features — from fully functioning chat to a chat product that you can ship — it can take a week to a month. For voice and video, you can make your test call in as little as 15 minutes. An API will cut in half the time it takes to develop an MVP, but scaling your calls will be the main effort. With an API and managed infrastructure, you needn't worry about scaling calls after your MVP. Of course, these times vary from project to project.

## **What features should I expect from an interactions API?**

You should expect features that provide a chat experience as captivating as any messenger app, as well as features for moderation, data, security, and integration. For voice and video, expect crisp voice and clear video with features to track custom call data, configure hardware sources, high standards for security, and every feature to match a mobile phone call experience. The following list is representative of features you should expect, but is not exhaustive.

### **Chat API features**

#### **Messaging features**

- Typing indicators
- Read receipts
- Invitations
- Chat history
- Video and image thumbnails
- Send and receive structured media
- Emojis and Reactions
- GIFs
- URL previews
- Rich text editing
- Push notifications

#### **Moderation features**

- User-to-user blocking
- Smart throttling

- Auto-moderation
- Image moderation
- Regex and profanity filtering

### **Data and Security**

- Advanced analytics
- Encryption in transit and at rest

### **Integration**

- Ability to link API services like Maps or Payments APIs directly in the message interface

### **Voice and video API features**

#### **Call features**

- Make a call
- Receive a call
- End a call
- Caller ID
- Contact list
- Display a user's call log in the application
- Push notifications for incoming calls
- Multi-device support
- High quality and performance across platforms: iOS, Android, React Native, JavaScript
- Link voice and video calls to a specific Sendbird chat channel

## **Data and security**

- Call metadata for deeper insights: topic, time, context
- Manage and view call logs in a detailed view, including metadata
- Enable test-only token-less user authentication
- Encrypt call metadata transfer
- Implement standard user login credential policies

## **How does an interactions API fit into my technology?**

An interactions API is a RESTful API built on simple HTTP requests like POST, PUT, DELETE, GET and responses structured in JSON or XML. These requests and responses are generalized so you can build anything you'd like using them. Chat and Voice & Video APIs will also typically come with an SDK for various platforms like iOS, Android, or JavaScript to ensure a simple implementation of every feature. This SDK will sit in parallel with your products other SDKs. All that's left is to display the chat, voice, and video features to the user in the UI. Often, there is a sample app that will help you create a UI.

## **What's the landscape of chat, voice, and video solutions?**

APIs are the “Goldilocks” technology for implementing chat, voice, and video, combining flexibility with full-featured chat and immersive voice and video. But other tools exist to implement chat, voice, and video like application development platforms or SaaS solutions. Application development platforms tend to fit businesses seeking a more general platform for building an application or syncing data between devices

(rather than a more specialized focus on digital interactions like chat or voice and video). Because these platforms do not specialize in digital interactions, you still need to build a lot of features from scratch and worry about scaling them. SaaS solutions, on the other hand, offer out-of-the-box functionality without any customization and limited white labeling. SaaS solutions are more appropriate for simple sales, marketing, or low volume consumer contexts.

### **How is an interactions API priced?**

Most services price voice and video on a per minute basis and offer different packages according to volume and other specifications.

For application development platforms, you'll also see a limitation to the number of API or function calls you can make.

SaaS follows the well-worn SaaS pricing model: cost per seat per month, in addition to added features and limits to API calls.

### **What should I keep in mind when rolling out chat, voice, and video?**

We always recommend consulting with your interactions API provider for this question, but here are some common considerations to plan before implementing chat:

1. Given that chat, voice, and video's most fundamental result is higher user engagement, how do you want to link these interactions to your business goals, depending on where engagement can impact results?
2. Chat, voice, and video bring greater insight to user experiences. At what

stage in your user journey or communication lifecycle do you want to learn more about what users talk about?

3. Determine your KPIs and set benchmarks. Typically, a customer's first goal is to drive customer engagement. For chat, you can measure this with total message growth, chat MAU growth, messages per user, messages per transaction, and others. For voice and video, you can measure growth in the number of connected calls, minutes per call, time to resolution, and NPS.
4. Determine what kind of chat moderation your application will need. If your users share content, you may need more user-driven moderation. If your users primarily message with strangers during transactions, then you may want more automatic moderation.
5. Make sure that your chat API provider has a connection manager in its SDK or can help you implement this. When your users connect directly correlates to MAU. Also ensure that your provider can consult on fundamental issues like when to create or archive a channel, and other considerations.
6. Think about your authentication process. Which token is appropriate for your product's security style: access or session tokens?

## Add in-app interactions to your application

Get started

 **sendbird**