

March 28, 2024

Ms. Marlene H. Dortch  
Secretary  
Federal Communications Commission  
45 L St. NE  
Washington, DC 20554

re: Regulatory Status of Wireless Messaging Service, WT Docket No. 08-7; IP-Enabled Services, WC Docket No. 04-36

Dear Ms. Dortch:

On Tuesday, March 26, Eric Migicovsky, CEO of Beeper; Kevin Joseph, advisor to Beeper; John Bergmayer, Legal Director of Public Knowledge; Harold Feld, Senior Vice President of Public Knowledge; and Thomas Jones, outside counsel to Reset.tech, met with Hannah Lepow, Legal Advisor for Media and Consumer Protection, and Shiva Goel, Legal Advisor, Wireless, Space, & International, from Commissioner Starks office.

Beeper is a software company that develops interoperable messaging apps, including Beeper Mini, an app that brought end-to-end encrypted iMessage communication between iPhone and Android users. Apple blocked Beeper from providing this functionality, as highlighted in the recent antitrust lawsuit brought by the Department of Justice, and a coalition of states and the District of Columbia.<sup>1</sup> Reset.tech is a global not-for-profit dedicated to realigning digital media markets with democratic values. Public Knowledge is a consumer rights group that promotes freedom of expression, an open internet, and access to affordable communications tools and creative works.

The purpose of this meeting was to inform the Commission about how Beeper Mini works, and to discuss the legal implications of Apple's blocking it that are relevant to the Commission's authority—in particular, but not limited to, interconnection requirements under Title II of the Communications Act and possible implications with regard to the 21st Century Communications and Video Accessibility Act.

Eric Migicovsky explained that Beeper Mini is an Android application that implements the iMessage protocol, allowing iPhone and Android users to send and receive encrypted messages with each other. At present, Apple bundles all its text messaging capability into its Messages App. When Apple users communicate with other Apple users, they have access to

---

<sup>1</sup> Complaint, United States v. Apple Inc., No. 2:24-cv-04055 (D.N.J. filed Mar. 21, 2024), <https://www.justice.gov/opa/media/1344546/dl?inline>.

features such as end-to-end encryption, high-resolution images, delivery receipts, the ability to seamlessly participate in group texts, and more. These texts appear in a “blue bubble” which highlights these improved functionalities. But when Apple users attempt to text Android users (or when Android users text Apple users), Apple restricts these texts to SMS — which lacks these enhanced features. To highlight the limitations of SMS as compared to Apple’s Message app, texts between Apple users and Android users are displayed in a green bubble. A technical explanation of how Beeper functions is attached to this filing, as well as relevant passages from the DOJ’s complaint.

Public Knowledge reiterated its support for Title II classification of interconnected VOIP,<sup>2</sup> and of SMS text messaging (SMS). These services are “telecommunications” in that they provide “the transmission, between or among points specified by the user, of information of the user’s choosing, without change in the form or content of the information as sent and received.”<sup>3</sup>

Public Knowledge, joined by other public interest groups, first petitioned the FCC to classify SMS as “telecommunications” in 2007,<sup>4</sup> after Verizon blocked access to SMS short codes to NARAL (now known as Reproductive Freedom for All). The Commission put this petition out for public comment in January 2008,<sup>5</sup> but took no subsequent action. In 2015, Twilio filed a petition for an expedited declaratory ruling, asking the Commission “to declare that messaging services are governed by Title II” of the Communications Act. The Commission put this petition out for public comment as well,<sup>6</sup> and again took no immediate action.

In December 2018, under new leadership, the Commission issued a Declaratory Ruling that SMS was a Title I information service.<sup>7</sup> Public Knowledge, joined by other public interest groups, filed a petition for reconsideration, arguing among other things that the Declaratory Ruling’s reasoning was flawed.<sup>8</sup> The DC Circuit subsequently validated this position by

---

<sup>2</sup> Petition of Public Knowledge et al. for Declaratory Ruling That Facilities-Based Interconnected VoIP & Nomadic Interconnected VoIP Are Title II Services, RM-\_\_\_\_\_ (filed Mar. 2, 2022), [https://publicknowledge.org/wp-content/uploads/2022/03/VOIP-Declaratory-Ruling-Petition\\_03-02-22.pdf](https://publicknowledge.org/wp-content/uploads/2022/03/VOIP-Declaratory-Ruling-Petition_03-02-22.pdf).

<sup>3</sup> 47 U.S.C. § 153(50).

<sup>4</sup> Petition of Public Knowledge et al. for Declaratory Ruling that Text Messaging and Short Codes are Title II Services or are Title I Services Subject to Section 202 Nondiscrimination Rules, WT Docket No. 08-7, (filed Dec. 11, 2007), <https://publicknowledge.org/policy/fcc-petition-for-text-messaging-and-short-codes>.

<sup>5</sup> Comment Sought on Petition for Declaratory Ruling That Text Messages and Short Codes Are Title II Services or Are Title I Services Subject to Section 202 Non-Discrimination Rules, WT Docket No 08-7, 73 FR 4866 (2008), <https://docs.fcc.gov/public/attachments/DA-08-282A1.pdf>

<sup>6</sup> WTB Seeks Comment on a Petition for Declaratory Ruling Clarifying the Regulatory Status of Mobile Messaging Services, WT Docket No. 08-7, 80 FR 69630 (2015), <https://docs.fcc.gov/public/attachments/DA-15-1169A1.pdf>.

<sup>7</sup> Petitions for Declaratory Ruling on Regulatory Status of Wireless Messaging Serv., Declaratory Ruling, WT Docket No. 08-7, 33 FCC Rcd 12075 (2018), <https://docs.fcc.gov/public/attachments/FCC-18-178A1.pdf>.

<sup>8</sup> Petition for Reconsideration of Public Knowledge et al., WT Docket No. 08-7, (filed Jan. 28, 2019), [https://publicknowledge.org/wp-content/uploads/2021/11/PK\\_et\\_al\\_SMS\\_Order\\_recon.pdf](https://publicknowledge.org/wp-content/uploads/2021/11/PK_et_al_SMS_Order_recon.pdf). *See also* Reply to Oppositions of Public Knowledge, WT Docket No. 08-7 (April 2, 2019), <https://publicknowledge.org/policy/fcc-sms-ruling-reply-to-opposition>.

rejecting the same reasoning in the broadband context in *Mozilla*.<sup>9</sup> The Commission put this petition for reconsideration out for public comment in March 2019,<sup>10</sup> but took no subsequent action.

During this time, SMS has become more important than ever. Smartphones have achieved almost universal adoption. Schools and governments use SMS to communicate with parents and the public. One-time security codes for accessing banking services are sent via SMS, and SMS, as a carrier-based service, remains the only way senders can be sure to reach nearly every member of the public, relative to non-carrier services like WhatsApp that require users to install and set up a third-party app. The problems caused by the Commission’s inaction, and flawed actions on this subject, have not gone away. If anything they have gotten worse, and though the FCC and FTC have taken actions directed at the scourge of robotexts and scams,<sup>11</sup> stronger legal authority would allow the Commission to take swifter and firmer action.

Non-carrier services like iMessage may also fit the definition of “telecommunications.” Apple, via the built-in and uninstalleable Messages app on the iPhone, combines SMS and iMessage into one seamless user experience—which users experience as “green bubbles” (SMS) and “blue bubbles” (iMessage).<sup>12</sup> Like with interconnected VOIP services, though with text and media messages instead of calls, Apple allows users to receive communications “that originate on the public switched telephone network” and to send messages that “terminate... [on] the public switched telephone network.”<sup>13</sup> As the DOJ notes in its recent lawsuit against Apple, Apple has prevented third-party apps like WhatsApp and Beeper from integrating with SMS as it does with its bundled app. Further, it has been slow to adopt industry standards like RCS that would improve cross-platform messaging, and has taken affirmative steps to block Beeper’s attempt to interoperate with iMessage.

---

<sup>9</sup> *Mozilla Corp. v. FCC*, 940 F.3d 1 (D.C. Cir. 2019).

<sup>10</sup> Petition for Reconsideration of a Declaratory Ruling on Regulatory Status of Wireless Messaging Service, 84 FR 8497 (2019).

<sup>11</sup> E.g., Targeting and Eliminating Unlawful Text Messages, CG Docket No. 21-402, Report and Order and Notice of Proposed Rulemaking, 38 FCC Rcd 2744 (2023); FTC, IYKYK: The Top Text Scams of 2022, <https://www.ftc.gov/news-events/data-visualizations/data-spotlight/2023/06/iykyk-top-text-scams-2022>.

<sup>12</sup> In the *Pulver.com* Petition, the Commission distinguished between services that did not provide transport and relied on a database of user names which the service provider independently stores, and services which provided transport and/or relied on North American Numbering Plan (NANP) phone numbers. In the 20 years since then, the Commission has not considered how technological and market changes may have changed the underlying reasoning in *Pulver.com*. But the Commission need not go so far as to reconsider *Pulver* to reach the conclusion that iMessage and SMS are properly classified as Title II.

<sup>13</sup> 47 CFR § 9.3.

Beeper also highlighted a recent letter from the American Economic Liberties Project et al.,<sup>14</sup> consistent with arguments put forward by Commissioner Carr,<sup>15</sup> that Apple’s blocking of Beeper Mini may violate the 21st Century Communications and Video Accessibility Act.<sup>16</sup> The letter argues that Apple’s actions to block Beeper Mini’s functionality with iMessage may have violated the FCC’s Part 14 rules, which are based on the CVAA. These rules require “covered providers,” including Apple, to ensure accessibility and usability of advanced communications services and equipment for people with disabilities. By impeding Beeper Mini’s interoperability with iMessage, and given the accessibility shortcomings of SMS relative to iMessage, Apple has potentially violated the rule stating that covered providers “shall not install network features, functions, or capabilities that impede accessibility and usability.”<sup>17</sup>

Interoperability and interconnection are core Title II requirements with a long pedigree. In *Carterfone*, the Commission required AT&T to allow third-party equipment to be used with the telephone system,<sup>18</sup> and followed this up with standard-setting activity designed to facilitate interconnection and industry standardization.<sup>19</sup> The Commission has used its Title II authority to promote competition and protect consumers in a variety of contexts, such as requiring interoperability measures like number portability. The Commission has also used its Title II authority to promote accessibility, through the establishment of TTY rules and other measures. The Title II toolbox is available to be used for pro-competition, public interest obligations, and is relevant to the current situation with SMS and iMessage. Just as the Commission used its Title II authority to promote competition and protect consumers in the telephone context, it can use that same authority to promote competition and protect consumers in the context of SMS and iMessage. By classifying these services as Title II telecommunications services, the Commission could ensure that they are subject to the same pro-competition, pro-consumer rules that have long applied to other important telecommunications services.

Respectfully submitted,

/s/ John Bergmayer  
Legal Director  
Public Knowledge

cc: Hannah Lepow, Shiva Goel

---

<sup>14</sup> Letter from the American Economic Liberties Project et al. to The Honorable Jessica Rosenworcel, Chairwoman, FCC (Feb. 27, 2024),

<https://www.economicliberties.us/wp-content/uploads/2024/02/2024-02-27-FCC-Apple-Beeper-Mini-Letter.pdf>.

<sup>15</sup> Emma Roth, *FCC Commissioner Wants to Investigate Apple over Beeper Mini Shutdown*, The Verge (Feb. 12, 2024), <https://www.theverge.com/2024/2/12/24071226/fcc-commissioner-brendan-carr-apple-beeper-mini>.

<sup>16</sup> Pub. Law No. 111-260, codified at 47 U.S. Code § 617 et seq.

<sup>17</sup> 47 CFR § 14.20.

<sup>18</sup> Use of the Carterfone Device in Message Toll Telephone Service, 13 FCC 2d 420 (1968).

<sup>19</sup> The FCC’s rules regarding interconnection between telephone networks and “terminal equipment,” which reference standards such as the RJ11 phone jack, are codified in its Part 68 rules. See 47 C.F.R. § 68.1-68.614; FCC, <https://www.fcc.gov/part-68>.

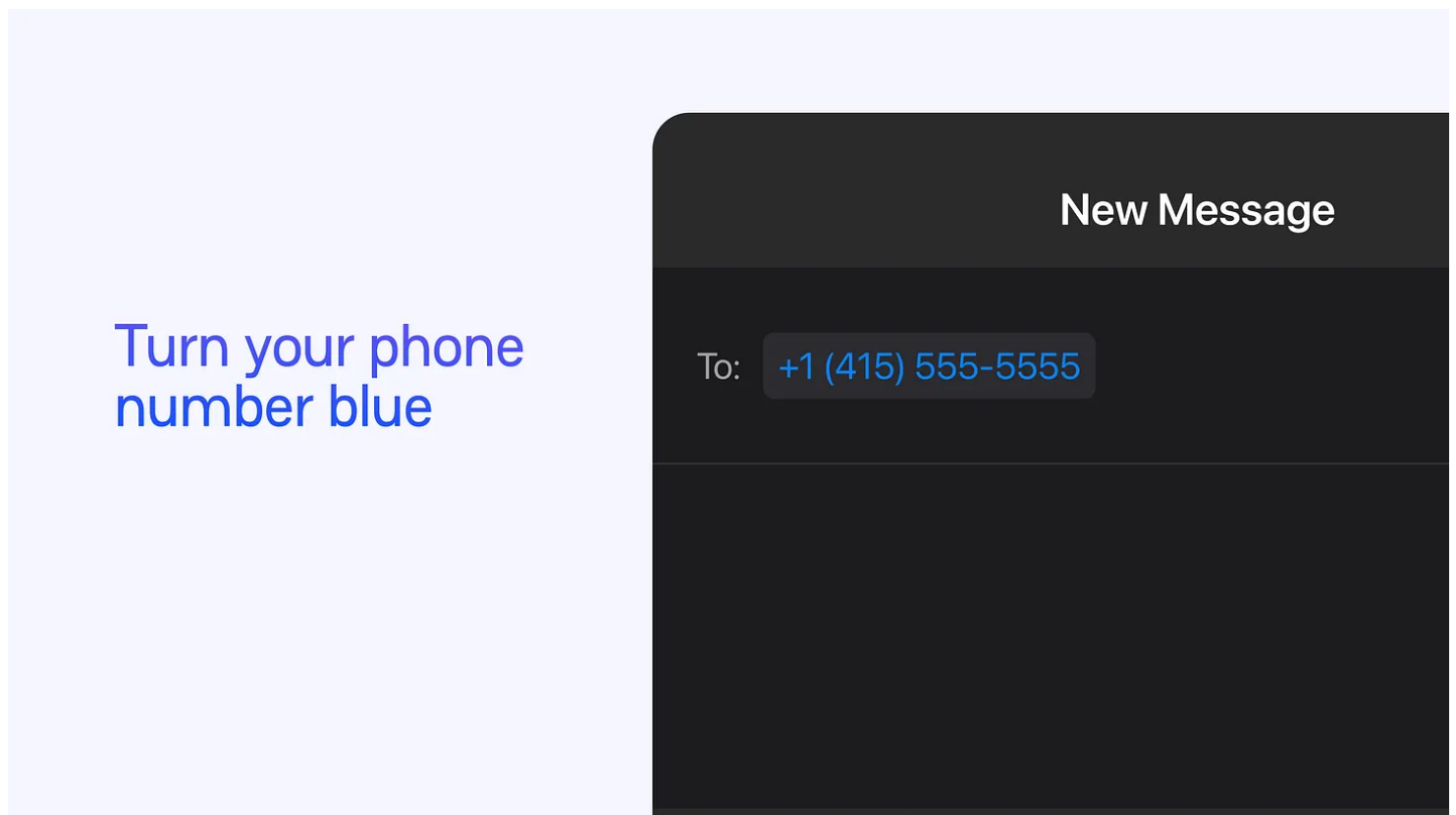
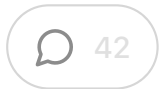
# How Beeper Mini Works

It's a technical deep dive, so buckle up!



BEEPER

DEC 05, 2023



We've written this blog post to help you understand how [Beeper Mini](#) works. At Beeper, we believe that it is critical for you to be able to trust the software that you use, especially something as important and sensitive as your chat app. We work to earn and keep your trust in three ways:

1. Transparency - since we started Beeper 3 years ago, we've been taking opportunities like this to explain how Beeper works. We have a proud history of building products, like Pebble, and [stand publicly](#) behind our work.
2. Open source - each major piece of software that we've built to interact with other chat

networks is open source at [github.com/beeper](https://github.com/beeper).

3. Privacy and security-aligned business model - we make great software and charge a small subscription fee. Simple as that. No ads. Your data stays private.

## Security and privacy

Read the entire post for the full story. TLDR: the following features of Beeper Mini ensure that all communication is encrypted and secure.

- All messages are end-to-end encrypted before being sent. Beeper (and Apple) cannot see your messages.
- Encryption keys never leave your device.
- Beeper Mini connects directly to Apple servers. There is no Mac server relay, like other apps.
- No Apple ID is required. Beeper does not have access to your Apple account.
- Your contact list never leaves your device.

Don't believe this is possible? Try the [open-source Python proof of concept](#) on your own computer to see for yourself. Security researchers are invited to verify all claims that we make, see appendix below.

## How it works

Beeper Mini works differently than Beeper Cloud in important ways that increase your privacy and security. Beeper Mini is a standalone Android app. It does not require a cloud server to send and receive messages. It also implements [Apple's end-to-end encryption protocol](#) natively within the Android app itself. All messages are end-to-end encrypted before they are transmitted directly from your device to Apple servers. Learn more about iMessage encryption on [Apple Platform Security](#) page.

This is now possible because the iMessage protocol and encryption have been reverse engineered by jjtech, a security researcher. Leveraging this research, Beeper Mini implements the iMessage protocol locally within the app. All messages are sent and

received by Beeper Mini Android app directly to Apple's servers. The encryption keys needed to encrypt these messages never leave your phone. Neither Beeper, Apple, nor anyone except the intended recipients can read your messages or attachments. Beeper does not have access to your Apple credentials.

We built Beeper Mini by analyzing the traffic sent between the native iMessage app and Apple's servers, and rebuilding our own app that sends the same requests and understands the same responses. Learn more by reading jjtech's blog post, [iMessage Explained](#), and his proof-of-concept [Python implementation on Github](#). Anyone can download this code, run it on any computer that supports Python, login to their iMessage account, and send and receive iMessage protocol messages. No Apple hardware required.

Another change is that Beeper Mini does not use the [Matrix](#) protocol, encryption or code like Beeper Cloud. It is a completely new codebase, versus our first Android app, which was a fork of [Element](#). In the future, we are planning to add Matrix network support back in, along with support for the 15 other chat networks in Beeper Cloud. Read more about [our roadmap](#).

## Inside the Beeper Mini Android app

### 1. Sign in

When you first start the Beeper Mini app and sign in with Google, a registration request is sent to our Beeper API Server. This service only exists to verify your subscription status, as well as give our support team the information they need to debug any issues that you may be running into (including your name and email address). No iMessage credentials or messages are transmitted through these servers, which are for Beeper Mini account management only.

### 2. Permissions and registration

After that, you are prompted to allow notifications, which sends a push token to Beeper Push Notification service, which enables our servers to send push notifications to your Android device. These push notifications do not contain the contents of messages.

Next, you are prompted to grant contact list and SMS permission access.

- Contact list access is used to match phone numbers to contact names, and display profile pictures. Your contact list is never sent to Beeper servers.
- SMS access is used to send an SMS text message from your number to Apple's "Gateway" service. The gateway sends a response via SMS, and the contents from that SMS response are sent to Apple to register your phone number as a blue bubble. Your SMS chat history is also used to determine if any of your recent SMS chats were with people who have iPhones. If so, these chats are shown in the inbox.

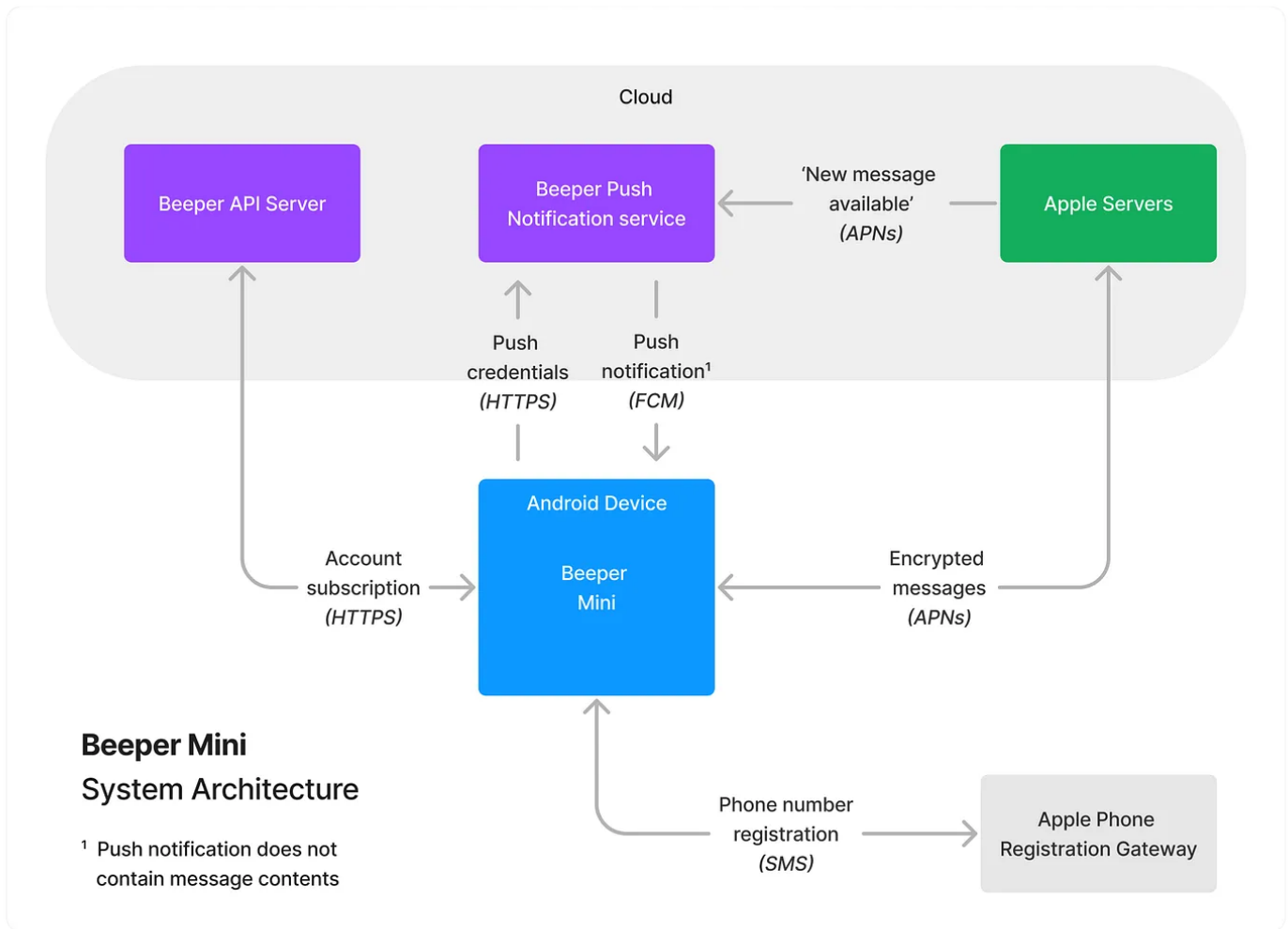
It's at this point that the app generates encryption keys that are used for end-to-end encrypted messaging. The public key is sent to Apple servers, and the private keys are stored in the Android device local filesystem. Beeper Mini is now signed in.

### **3. Optional Apple ID sign in**

Optionally, you may also sign in to your Apple ID to enable sending/receiving from your email address. This will also enable you to send and receive messages from other Apple devices like iPad or Macs. The Apple ID login sends your username, password and a 2-factor code using encrypted HTTPS requests directly to Apple servers.

### **4. Sending and receiving messages**





Apple's iMessage protocol works over [Apple Push Notification service](#), which most developers would be familiar as the service that allows them to send push notifications to their iOS applications. For iMessage protocol, all messaging traffic flows over this service in both directions, encrypted with keys generated locally on each device. Beeper Mini connects to APNs over TCP, using the credentials generated during the login process.

A persistent connection to APNs is needed to be notified of new incoming messages in real-time. On an iPhone, an APNs connection is maintained by the operating system, and connected at all times. In Beeper Mini, the connection can only be maintained when the app is running, since Android does not support APNs natively.

To work around this limitation, we built Beeper Push Notification service (BPNs). BPNs connects to Apple's servers on your behalf when Beeper Mini Android app isn't running. We can do this while preserving user privacy thanks to Apple separating the credentials

needed to connect to APNs to send and receive content (the “push” credentials) and the keys needed to encrypt and decrypt messages (the “identity” keys). Push credentials can be shared securely with the Beeper Push Notification service, and BPNs can connect to APNs on your behalf. Whenever BPNs receives an encrypted message that it won't be able to decrypt, it simply disconnects from APNs and sends an FCM push notification to wake up the Android app, which then connects to APNs, downloads, decrypts and processes the incoming message. BPNs can only tell when a new message is waiting for you - it does not have credentials to see or do anything else.

BPNs will be notified when you receive a message, but without the encryption keys it can't decrypt anything BPNs receives. Also, without the identity credentials, BPNs can't send messages on your behalf. If you don't mind not receiving real-time push notifications for new messages, your BPNs can be disabled entirely by going to Settings → Manage Connection → Enable Push.

When you create a new chat, the phone number or email address of your intended recipient is transmitted to Apple servers. If the contact is on iMessage, a public key is returned.

Sending messages is even simpler. When you hit send, the message is encrypted with the public keys of the intended recipients and sent directly to Apple servers via an SSL encrypted TCP connection over APNs.

## **5. Analytics and other services**

Beeper Mini connects to a few other services as part of its operation. We use a self-hosted installation of Rudderstack (<https://rudderstack.beeper-tools.com>) for analytics and diagnostic events, which we use for improving the app but can be disabled in Settings → Preferences → Share Diagnostics. We use OneSignal to send education and account related push notifications, and RevenueCat to help integrate Google Play subscriptions.

Other than that, that's it! No other servers or services are used. Beeper Mini keeps your messaging secure by keeping all messaging credentials, keys, messages and media local to your phone, and only sends them directly to Apple's servers after encrypting them with iMessage's end-to-end encryption algorithm.

We value, actually, we treasure feedback. If you run into a bug or have a feature request, there's a button in-app to report a problem. We read every single report.







**Brad Murray and Eric Migicovsky**

Beeper cofounders







## Appendix

To write this blog post, we performed a red team analysis on our own app. We made extensive use of the excellent [mitmproxy](#) project to capture the network traffic coming from a real phone running a modified version of the Beeper Mini client. A modified version was needed for this analysis in order to disable certificate pinning, so that the Beeper Mini Android app would accept being connected to mitmproxy instead of only accepting Apple's certificates for that connection. If researchers would like a copy of this version of Beeper Mini (with cert pinning disabled) to perform a similar analysis, please contact us at [security@beeper.com](mailto:security@beeper.com).

Below is a capture of the requests that we make with Apple's servers over HTTPS when logging into iMessage with your phone number. We first register with a service named `albert.apple.com`, which sets up our "push" credentials and allows us to connect to APNS. We then make two requests to get the number we need to send an SMS to register our phone number which is different for each carrier (This capture was taken with a device registered with Rogers, a Canadian cell phone carrier 🇨🇦). Finally, we take the contents of the response SMS (not shown here) and send it to `identity.ess.apple.com`, registering our account with iMessage and generating the "identity" credentials we'll use to send and receive.

Path	Method	Status	Size	Time
 https://albert.apple.com/WebObjects/ALUnbrick.woa/wa/deviceActivation?device=Windows	POST	200	13.8kb	109ms
 https://itunes.apple.com/WebObjects/MZStore.woa/wa/com.apple.jingle.appserver.client.MZITunesClientCheck/version?languageCode=en	GET	200	5.1mb	5s
 https://updates.cdn-apple.com/20230602/carrierbundles/032-23859/3D89D248-1960-4935-8780-07F0BF259703/Rogers_ca_iPhone.ipcc	GET	200	111.6kb	123ms
 https://identity.ess.apple.com/WebObjects/TDIdentityService.woa/wa/authenticatePhoneNumber	POST	200	3.3kb	121ms
 https://profile.ess.apple.com/WebObjects/VCProfileService.woa/wa/idsGetHandles	GET	200	891b	164ms
 https://identity.ess.apple.com/WebObjects/TDIdentityService.woa/wa/register	POST	200	8.4kb	368ms

Optionally, you can also register your Apple ID with Beeper Mini as well, as shown in this capture. You first provide your username and password over encrypted HTTPS directly to Apple's servers, followed by a second request to provide your 2FA code. We can then register for iMessage again, this time providing the certificates from both the earlier phone number registration and our new Apple ID registration. Registering these together in the same call links them together, allowing any other device that you're logged in with your Apple ID to send and receive with both your Apple ID emails and your phone number.

	https://profile.ess.apple.com/WebObjects/VCProfileService.woa/wa/authenticateUser	POST	200	1.1kb	219ms
	https://profile.ess.apple.com/WebObjects/VCProfileService.woa/wa/authenticateUser	POST	200	1.8kb	203ms
	https://profile.ess.apple.com/WebObjects/VCProfileService.woa/wa/authenticateDS	POST	200	5.6kb	214ms
	https://profile.ess.apple.com/WebObjects/VCProfileService.woa/wa/idsGetHandles	GET	200	891b	92ms
	https://profile.ess.apple.com/WebObjects/VCProfileService.woa/wa/idsGetHandles	GET	200	1.4kb	162ms
	https://identity.ess.apple.com/WebObjects/TDIdentityService.woa/wa/register	POST	200	14.1kb	251ms

Next, a capture of the keys shared with the Beeper Push Notification service (hostname `imux.beeper.com`). Note, the RSA private key in this request is your “push” credentials that allow you to connect to APNs, not your “identity” credentials that allow you to encrypt and decrypt iMessages. Push credentials cannot be used to escalate permissions or access anything other than the presence of a new APNs push notification. Check out [apns.py](#) in [pypush PoC](#) to learn more about push credentials.

PUT https://imux.beeper.com/v1/apns/device HTTP/2.0

**user-agent:** imessagego

**authorization:** Bearer

imat\_51efd3n9enyukwr97khfrtnrddrkvmkm16n7n1ufm9ddw6y7e140

**content-length:** 3254

**accept-encoding:** identity

JSON

Edit

Replace

View: json ▾

```
{
  "apns": {
    "cert": "-----BEGIN CERTIFICATE-----\nMIIDdzCCAUCgAwIBAgIKAYdc
    "key": "-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEA/ReLX
    "token": "YQ2kV8xIAACxiLRHZhtrxQXLl8d9jTzRsy/n+ZSD8hg="
  },
  "fcm": {
    "token": "dtis9GENRzqd9dTY2mQMMS:APA91bFrS CrRM6qIH2st4sAWMEAHc
  }
}
```

Sending and receiving is not shown here, as they are not done over HTTP but instead through an SSL encrypted TCP connection to APNs. The APNs servers are hosted at `*-courier.push.apple.com`, where the asterisk is replaced by a number between 1 and 30. All message contents and media are encrypted with your “identity” keys, which never leave your Android phone.

There is a `/login` endpoint on Beeper servers, but as mentioned previous, this is only for subscription management purposes. The client submits the token received from the Google login process to our servers, and the response contains their subscription status. No iMessage credentials are ever sent to Beeper servers.

POST https://api.beeper.com/ima/login HTTP/2.0

user-agent: Beeper/1.0.0 (Google Pixel 3 XL; Android 12; SP1A.210812.016.C2)

accept: application/json

accept-charset: UTF-8

content-type: application/json

content-length: 1129

accept-encoding: identity

JSON

Edit

Replace

View: auto

```
{
  "token": "eyJhbGciOiJIJSUzI1NiIsImtpZCI6ImU0YWVmYjQzNmI5ZTE5N2UyZTE5MDZhZjJjODQyMjg0ZTQ5ODZhZmYiLCJ0eXAiOiJK"
}
```

JSON

Edit

Replace

View: auto

```
{
  "access_token": "imat_51efd3n9enyukwr97khfrtnrddrkvmkm16n7n1ufm9ddw6y7e140",
  "analytics_id": "cl_xnz67krf70",
  "ima_user_token": "imau_f878geu2ar",
  "subscription": {
    "active": true,
    "expires_at": "2223-10-08T00:39:43Z",
    "product_identifier": "rc_promo_imessage-on-android_lifetime"
  }
}
```

**Note:** Beeper and Beeper Mini are entirely independent software products, with no relationship to, or endorsement by, Apple, Google, or any other supported chat networks.

iMessage, Apple, Mac and iPhone are trademarks of Apple, Inc.

Android is a trademark of Google, LLC.



19 Likes · 4 Restacks

**B. Apple uses APIs and other critical access points in the smartphone ecosystem to control the behavior and innovation of third parties in order to insulate itself from competition**

**i. Messaging: Apple protects its smartphone monopoly by degrading and undermining cross-platform messaging apps and rival smartphones**

80. Apple undermines cross-platform messaging to reinforce “obstacle[s] to iPhone families giving their kids Android phones.” Apple could have made a better cross-platform messaging experience itself by creating iMessage for Android but concluded that doing so “will hurt us more than help us.” Apple therefore continues to impede innovation in smartphone messaging, even though doing so sacrifices the profits Apple would earn from increasing the value of the iPhone to users, because it helps build and maintain its monopoly power.

81. Messaging apps allow smartphone users to communicate with friends, family, and other contacts and are often the primary way users interact with their smartphones. In Apple’s own words, messaging apps are “a central artery through which the full range of customer experience flows.”

82. Smartphone messaging apps operate using “protocols,” which are the systems that enable communication and determine the features available when users interact with each other via messaging apps.

83. One important protocol used by messaging apps is SMS.<sup>1</sup> SMS offers a broad user network, but limited functionality. For example, all mobile phones can receive SMS messages, but SMS does not support modern messaging features, such as large files, edited messages, or reactions like a “thumbs up” or a heart.

---

<sup>1</sup> Following industry practice, throughout this complaint, “SMS” refers to both SMS and MMS (“multimedia messaging service”). MMS is a companion protocol to SMS that allows for group messages and messages with basic multimedia content, such as small file sharing.

84. Many messaging apps—such as WhatsApp, Facebook Messenger, and Signal—use proprietary, internet-based protocols, which are sometimes referred to as OTT (“over the top”) protocols. OTT messaging typically involves more secure and advanced features, such as encryption, typing indicators, read receipts, the ability to share rich media, and disappearing or ephemeral messages. While all mobile phones can send and receive SMS messages, OTT only works between users who sign up for and communicate through the same messaging app. As a result, a user cannot send an OTT message to a friend unless the friend also uses the same messaging app.

85. Apple makes third-party messaging apps on the iPhone worse generally and relative to Apple Messages, Apple’s own messaging app. By doing so, Apple is knowingly and deliberately degrading quality, privacy, and security for its users. For example, Apple designates the APIs needed to implement SMS as “private,” meaning third-party developers have no technical means of accessing them and are prohibited from doing so under Apple’s contractual agreements with developers. As a result, third-party messaging apps cannot combine the “text to anyone” functionality of SMS with the advanced features of OTT messaging. Instead, if a user wants to send somebody a message in a third-party messaging app, they must first confirm whether the person they want to talk to has the same messaging app and, if not, convince that person to download and use a new messaging app. By contrast, if an Apple Messages user wants to send somebody a message, they just type their phone number into the “To:” field and send the message because Apple Messages incorporates SMS and OTT messaging.

86. Apple prohibits third-party developers from incorporating other important features into their messaging apps as well. For example, third-party messaging apps cannot continue operating in the background when the app is closed, which impairs functionality like



message delivery confirmation. And when users receive video calls, third-party messaging apps cannot access the iPhone camera to allow users to preview their appearance on video before answering a call. Apple Messages incorporates these features.

87. If third-party messaging apps could incorporate these features, they would be more valuable and attractive to users, and the iPhone would be more valuable to Apple in the short term. For example, by incorporating SMS, users would avoid the hassle of convincing someone to download a separate app before sending them a message. Third-party messaging apps could also offer the ability to schedule SMS messages to be sent in the future, suggest replies, and support robust multi-device use on smartphones, tablets, and computers—as they have already done on Android.

88. Moreover, messaging apps benefit from significant network effects—as more people use the app, there are more people to communicate with through the app, which makes the app more valuable and in turn attracts even more users. Incorporating SMS would help third-party messaging apps grow their network and attract more users. Instead, Apple limits the reach of third-party messaging apps and reinforces network effects that benefit Apple.

89. Recently, Apple has stated that it plans to incorporate more advanced features for cross-platform messaging in Apple Messages by adopting a 2019 version of the RCS protocol (which combines aspects of SMS and OTT). Apple has not done so yet, and regardless it would not cure Apple's efforts to undermine third-party messaging apps because third-party messaging apps will still be prohibited from incorporating RCS just as they are prohibited from incorporating SMS. Moreover, the RCS standard will continue to improve over time, and if Apple does not support later versions of RCS, cross-platform messaging using RCS could soon be broken on iPhones anyway.

90. In addition to degrading the quality of third-party messaging apps, Apple affirmatively undermines the quality of rival smartphones. For example, if an iPhone user messages a non-iPhone user in Apple Messages—the default messaging app on an iPhone—then the text appears to the iPhone user as a green bubble and incorporates limited functionality: the conversation is not encrypted, videos are pixelated and grainy, and users cannot edit messages or see typing indicators. This signals to users that rival smartphones are lower quality because the experience of messaging friends and family who do not own iPhones is worse—even though Apple, not the rival smartphone, is the cause of that degraded user experience. Many non-iPhone users also experience social stigma, exclusion, and blame for “breaking” chats where other participants own iPhones. This effect is particularly powerful for certain demographics, like teenagers—where the iPhone’s share is 85 percent, according to one survey. This social pressure reinforces switching costs and drives users to continue buying iPhones—solidifying Apple’s smartphone dominance not because Apple has made its smartphone better, but because it has made communicating with other smartphones worse.

91. Apple recognizes that its conduct harms users and makes it more difficult to switch smartphones. For example, in 2013, Apple’s Senior Vice President of Software Engineering explained that supporting cross-platform OTT messaging in Apple Messages “would simply serve to remove [an] obstacle to iPhone families giving their kids Android phones.” In March 2016, Apple’s Senior Vice President of Worldwide Marketing forwarded an email to CEO Tim Cook making the same point: “moving iMessage to Android will hurt us more than help us.”

92. In 2022, Apple's CEO Tim Cook was asked whether Apple would fix iPhone-to-Android messaging. "It's tough," the questioner implored Mr. Cook, "not to make it personal but I can't send my mom certain videos." Mr. Cook's response? "Buy your mom an iPhone."

93. Recently, Apple blocked a third-party developer from fixing the broken cross-platform messaging experience in Apple Messages and providing end-to-end encryption for messages between Apple Messages and Android users. By rejecting solutions that would allow for cross-platform encryption, Apple continues to make iPhone users' less secure than they could otherwise be.