UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

PROXENSE, LLC,

Plaintiff,

v.

APPLE, INC.

Defendant.

Civil Action No. 6:24-cv-00143

JURY TRIAL REQUESTED

FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Proxense, LLC ("Proxense" or "Plaintiff") hereby sets forth its First Amended Complaint for patent infringement against Defendant Apple, Inc. ("Apple" or "Defendant"), for patent infringement under 35 U.S.C. § 271 and alleges as follows:

NATURE OF THE CASE

1. This action is for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§ 1, et seq. As further stated herein, Proxense alleges that Apple willfully infringes one or more claims of patents owned by Proxense. Accordingly, Proxense seeks monetary damages and injunctive relief in this action.

THE PARTIES

2. Plaintiff Proxense, LLC is a Delaware company with its principal place of business at 689 NW Stonepine Drive, Bend, Oregon 97703. Defendant Apple, Inc. is a corporation organized and existing under the laws of California, with one or more regular and established places of business in this District at least at 12545 Riata Vista Circle, Austin, Texas 78727; 12801

Delcour Drive, Austin, Texas 78727; 6800 W Parmer Lane, Austin, Texas 78729, and 3121 Palm Way, Austin, Texas 78758. Apple may be served with process through its registered agent, the CT Corp System, at 1999 Bryan St., Ste. 900 Dallas, Texas 75201-3136. In November 2019, Apple stated that it had approximately 7,000 employees in Austin and that it expected to open, in 2022, a $1 billion, 3 million-square foot campus with capacity for 15,000 employees. See **Exhibit 7**. Apple is registered to do business in the State of Texas and has been since at least May 16, 1980.

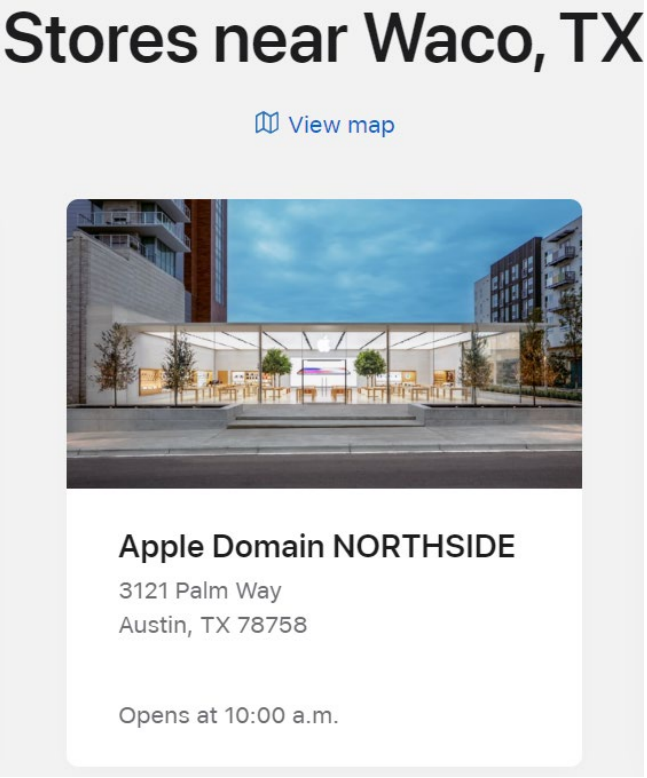
JURISDICTION AND VENUE

3. This Court has exclusive subject matter jurisdiction over this case pursuant to 28 U.S.C. §§ 1331, 1332, 1338, and 1367 on the grounds that this action arises under the Patent Laws of the United States, 35 U.S.C. § 1 et seq., including, without limitation, 35 U.S.C. §§ 271, 281, 284, and 285.

4. The amount in controversy exceeds $75,000.

5. This Court has specific personal jurisdiction over Apple because Apple is a multinational technology company that has a significant presence in the District through the products and services Apple provides residents of this District. Defendant regularly conducts business and have committed acts of patent infringement within this Judicial District that give rise to this action and have established minimum contacts within this forum such that the exercise of jurisdiction over Apple would not offend traditional notions of fair play and substantial justice. Apple has committed and continues to commit acts of infringement in this Judicial District, by, among other things, offering to sell, selling, using, importing, and/or making products and services that infringe the asserted patents. Apple has further induced acts of patent infringement by others in this Judicial District and/or has contributed to patent infringement by others in this Judicial District, the State of Texas, and elsewhere in the United States.

6. Apple operates retail stores located in this Judicial District that offer and sell products on its behalf in this District, including products accused of infringement herein. For example, Apple operates the Apple Domain NORTHSIDE store located at 3121 Palm Way, Austin, Texas 78758 and the Apple Barton Creek store located at 2901 S Capital of Texas Hwy, Austin, Texas 78746. A screenshot of a search for Waco, Texas in the "Find a Store" section of Apple's website, taken on February 10, 2024, reveals a picture of its Apple Domain NORTHSIDE location:



7. Apple's website further indicates that its Apple Domain NORTHSIDE location offers shopping, order pickup, and support/instruction (through the Genius Bar and scheduled workshops).

8. Apple also partners with third parties in this District acting as Apple Authorized Retailers that can sell Apple products via their physical store(s). These include retailers Target

(e.g., the retail location at 5401 Bosque Blvd., Waco, TX), Walmart (e.g., the retail locations at 4320 Franklin Ave and 600 Hewitt Dr located in Waco, TX), Best Buy (e.g. the retail location at 4627 S Jack Kultgen Expy, Waco, TX), Office Depot (e.g., the retail locations at 5524 Bosque Blvd and 4627 Jack Kultgen Fwy in Waco, TX), T-Mobile (e.g., the retail locations at 2448 W Loop 340 Suite 24a and 100 N New Rd Ste 110 in Waco, TX), and Verizon (e.g., the retail locations at 2812 W Loop 340 Ste H12 and 5301 Bosque Blvd Ste 120 in Waco, TX). On information and belief, Apple Authorized Resellers are important to Apple's retail strategy and represent a large portion of sales of Apple products worldwide.

9. Proxense's causes of action arise directly from Apple's business contacts and other activities in the State of Texas and this District.

10. Apple has derived substantial revenues from its infringing acts within the State of Texas and this District.

11. Venue is proper in this Judicial District pursuant to 28 U.S.C. §§ 1400(b) and 1391(b) and (c) because Defendant is subject to personal jurisdiction in this Judicial District, has committed acts of patent infringement in this Judicial District, and has a regular and established place of business in this Judicial District. Defendant, through its own acts, makes, uses, sells, and/or offers to sell infringing products within this Judicial District, regularly does and solicits business in this Judicial District, and has the requisite minimum contacts with the Judicial District such that this venue is a fair and reasonable one.

12. Apple also maintains a significant physical presence in this Judicial District and employs many people in this Judicial District. On November 20, 2019, Apple announced that it was breaking ground on its $1 billion, 3-million-square-foot campus "as part of its broad expansion in the city." See Apple Newsroom, Press Release: Apple expands in Austin, **Exhibit 7**. Apple

also described in that 2019 press release that it was "steadily growing in Austin with approximately 7,000 employees in the city." *Id*. Not only did Apple build a new campus in Austin, "Apple and its manufacturing partners invested over $200 million in the Mac Pro facility in Austin, building out the complex assembly line where the Mac Pro is produced." *Id*.

13. The Mac Pro is an accused product in this case capable of using Apple Pay. Apple's press release also revealed a photo of its Mac Pro facility in Austin, which it describes as employing more than 500 people:

14. In 2023, Apple "filed projects totaling $240 million for an expansion of its north Austin campus…" **Exhibit 8** "Capstone Phase Two AC09 and Capstone Phase Two AC07, as the projects are called, are a four-story and a five-story building, respectively, set for construction at 6900 Parmer Lane. Between the two buildings, Apple is adding 419,441 square-feet of office space. Construction begins for both buildings on September 30 of [2023] and have an estimated completion date of March 30, 2025." *Id*.

15. The "Careers at Apple" section of Apple's website highlights Austin as having a "range of teams and specialties." **Exhibit 9**. Among the categories of roles shown are Hardware, Operations and Supply Chain, Corporate Functions, Support and Service, Software and Services, and Sales and Business Development. So important to Apple is the Austin campus that Apple recently threatened that one of its teams, located in California, to relocate to Austin or be terminated. **Exhibit 10**.

16. Among the teams located in Austin is the Apple Pay team. Apple Pay is pervasive and ubiquitous, included in nearly all of Apple's offerings, from the Mac Pro to the iPhone to the recently introduced Vision Pro. Several key members of the Apple Pay team are located in Texas, including:

- Alexander Antunovic is the Servicing and Readiness Lead for Apple Pay Business Operations, located in Austin, Texas. Mr. Antunovic describes on his LinkedIn profile that he has "[o]ver 15 years of contactless EMV payment experience that has involved physical card issuance, digital (device & e-commerce) tokenization, merchant acceptance, implementation, remediation, business development, commercialization, product roadmap enhancement and drafting of business rules." Mr. Antunovic has worked at Apple on its Apple Pay team since January 2019. Prior to his role at Apple, Mr. Antunovic worked for Mastercard in New York. He studied Computer Science at Concordia University. **Exhibit 11**.
- Alex Coulter is the Global Operations Project Manager for the Apple Pay group, located in Austin, Texas.

- Geoff Johnson is a "proven business leader" working for Apple Pay Business Development group, located in Austin, Texas. According to his LinkedIn profile, he "helps bring the future of commerce to life with Apple Pay." **Exhibit 12.**
- Jennifer Cervantes is in Partner Operations for the Apple Pay group, located in Austin, Texas. She has worked for the Apple Pay group since May 2020. Prior to her role at Apple, Ms. Cervantes lived in California where she worked at TrustCommerce as a Vice President of Account Management. **Exhibit 13.**

17. Apple has also partnered with third parties, such as Austin Telco Federal Credit Union, to utilize credit and debit cards issued by those third parties, with Apple Pay as "an easy, secure, and private way to make purchases." **Exhibit 14**. Apple Pay can also be used with shopping apps such as those for Target.

18. On information and belief, Apple has committed acts of willful direct and indirect infringement in the Western District of Texas. For example, Apple sells computers (e.g., Mac Pros, iMacs), laptops (e.g., MacBooks and MacBook Pros), iPads, iPhones, Apple Watches, and the new Vision Pro to individuals in this Judicial District, which ship with various versions of MacOS, iOS, iPadOS, watchOS, and visionOS. Apple also distributes Apple Pay, in this Judicial District, available across its device offerings.

PATENTS-IN-SUIT

19. Intentionally left blank.

20. On January 8, 2013, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 8,352,730 (the "730 Patent") entitled "Biometric Personal Data Key (PDK) Authentication." A true and correct copy of the 730 Patent is attached hereto as **Exhibit 2**.

21. On November 11, 2014, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 8,886,954 (the "954 Patent") entitled "Biometric Personal Data Key (PDK) Authentication." A true and correct copy of the 954 Patent is attached hereto as **Exhibit 3**.

22. On June 30, 2020, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 10,698,989 (the "989 Patent") entitled "Biometric personal data key (PDK) Authentication." A true and correct copy of the 989 Patent is attached hereto as **Exhibit 4**.

23. On June 13, 2017, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 9,679,289 (the "289 Patent") entitled "Hybrid Device Having a Personal Digital Key and Receiver-Decoder circuit and Methods of Use." A true and correct copy of the 289 Patent is attached hereto as **Exhibit 5**.

24. On February 4, 2014, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 8,646,042 (the "042 Patent") entitled "Hybrid Device Having a Personal Digital Key and Receiver-Decoder Circuit and Methods of Use." A true and correct copy of the 042 Patent is attached hereto as **Exhibit 6**.

25. Proxense is the sole and exclusive owner of all right, title, and interest to and in, or is the exclusive licensee with the right to sue for, the 730, 954, 989, 289, and 042 Patents (together, the "Patents-in-Suit"), and holds the exclusive right to take all actions necessary to enforce its rights to the Patents-in-Suit, including the filing of this patent infringement lawsuit. Proxense also has the right to recover all damages for past, present, and future infringement of the Patents-in-Suit and to seek injunctive relief as appropriate under the law.

26. The technologies of the Patents-in-Suit were invented by John Giobbi and David L. Brown. The 730 and 954 Patents generally cover systems and methods for an integrated device that persistently stores biometric data for a user in a tamper-resistant format. Subsequently, scan

data collected from a user (e.g., a fingerprint) can be compared against the stored biometric data. Once the user has been biometrically verified by the integrated device, a code can be wirelessly transmitted for authentication. The 989 Patent generally cover systems and methods of verifying a user during authentication of an integrated device.

27. The 289 and 042 Patents generally cover a hybrid device including a personal digital key ("PDK") and receiver-decoder circuit ("RDC"), wherein the PDK and RDC are coupled for communication with each other. The 289 and 042 Patents also includes a number of system configurations for use of the hybrid device including: use of the hybrid device in a cell phone; simultaneous use of the PDK and the RDC functionality of hybrid device; and use of multiple links of hybrid device to generate an enablement signal, use of multiple PDK links to the hybrid device to generate an enablement signal.

FACTUAL ALLEGATIONS

I. TECHNOLOGY BACKGROUND

28. Authentication is the process by which the identity of a user is confirmed on a device, including computers, tablets, and phones. When a person is authenticated, the goal is to verify that the credentials presented are authentic. For years, users were authenticated with usernames and passwords. However, with the amount of sensitive personal and financial information currently stored on personal devices, and the rise of biometric readers and high-speed networks, there was a need to implement improved authentication architectures. For over a decade, Apple has integrated biometric reader hardware into its devices, beginning with fingerprint scanners (i.e., Touch ID) and now spanning to facial (i.e., Face ID) and retina scanners (Optic ID), the latter being introduced in the 2024-released Vision Pro augmented reality headset.

29. In 2012, Apple acquired AuthenTec, a company focused on fingerprint-reading and identification management software, for $356 million. By September 10, 2013, Apple would integrate fingerprint reader technology in the iPhone 5S. The feature was marketed as "Touch ID." With the unveiling of the iPhone 6 and 6 Plus at a keynote event on September 9, 2014, Touch ID was expanded from being used to unlock the device and authenticating App Store purchases to also authenticating Apple Pay. The first generation of Touch ID was used in iPhone and iPad products from 2013-2021, including the iPhone 5S, SE (1st generation), 6, and 6 Plus and the iPad Pro 12.9 inch (2015) and 9.7 inch (2016), Air 2, Mini 3, Mini 4, and 5th to 9th generations (2015-2021). The second generation of Touch ID was used in iPhone, iPad, MacBook, iMac, and the Magic Keyboard from 2015 to products that Apple continues to sell today, including the iPhone 6S, 6S Plus, 7, 7 Plus, 8, 8 Plus, SE (2nd and 3rd generations); iPad (10th generation), iPad Pro (2nd generation), Air (3rd to 5th generations), Mini (5th and 6th generations), MacBook Pro (from 2018-2023), MacBook Air

30. The successor to Touch ID was announced by Apple during the unveiling of the iPhone X on September 12, 2017. It was marketed as Face ID. Like Touch ID, Face ID allows biometric authentication for unlocking a device and making payments through Apple Pay. Face ID has been available on newer iPhone models beginning with the iPhone X and all iPad Pro models, including 2018 iPad Pros and later.

31. In 2024, Apple introduced Retina ID with the Apple Vision Pro. Apple describes that: "In the same way that Touch ID revolutionized authentication using a fingerprint and Face ID revolutionized authentication using facial recognition, Optic ID revolutionizes authentication using iris recognition." **Exhibit 15** (last visited February 11, 2024). "With a look, Optic ID securely unlocks your Apple Vision Pro. You can use it to authorize purchases from the App Store

and Book Store, payments using Apple Pay, and more." *Id*. "Developers can also allow you to use Optic ID to sign into their apps." *Id*. "Apps that support Touch ID or Face ID automatically support Optic ID." *Id*.

32. Apple leverages the biometric readers integrated on its devices (and Touch ID / Face ID / Optic ID) for authenticating financial transactions, through Apple Pay. as well as for passwordless logins. For example, Apple uses "federated authentication" (also known as "federated identity"), which relies on an external trusted system to provide the service and function of authenticating users. See e.g. **Exhibit 16**. In a federated authentication solution, the system being accessed must request authentication of the user from the external system that is providing the user authentication functions and services. The external system providing the authentication service will then communicate successful authentication back to the system being accessed. Successful authentication is communicated between the two systems with the issuance security tokens containing claims about user authentication. Upon successful authentication of a user, the external system issues a security token which can be exchanged for access to the other system. One such federated architecture is OpenID Connect, which Apple has adopted.

33. While OpenID and OAuth 2 based identity services limit the use of passwords, they do not eliminate them. Authentication services geared towards eliminating passwords include those utilizing the WebAuthn protocol and its derivative, FIDO2, an open authentication standard developed by the FIDO Alliance, of which Apple is a member. Identity services utilizing the WebAuthn protocol, and the derivative protocol FIDO2, utilize an asymmetric key pair to authenticate a device. Possession and control of the device verifies the identity of the user. The device, referred to as an authenticator, generates a private/public key pair and a credential ID uniquely identifying the key pair. The public key and credential ID are sent to the authentication

service – called in the protocol the "relying party". The private key is held by the authenticator. During authentication, the authenticator sends a signature generated with its private key and the credential ID identifying the private key used to generate the signature. The relying party (i.e., authentication service) uses the credential ID to retrieve the matching public key. The signature is then verified with the public key. Upon successful verification of the signature, the relying party authentication service issues an authentication response. Apple has also adopted FIDO2. **Exhibit 17**.

34. WebAuthn and federated protocols can be combined by an identity service provider. When combined, the system to be accessed by the user requests authentication by a WebAuthn / FIDO2 server of the identity service provider. The server issues an authentication request to the user's authenticator. The authenticator responds by providing a signature and credential ID to the WebAuthn/FIDO2 server. If the signature is verified, the WebAuthn/FIDO2 server informs the OpenID / OAuth 2 server of successful authentication. The OpenID / OAuth 2 server of the identity service provider then sends a security token to be used to access the system requesting authentication.

35. Attempting to eliminate the use of passwords, Apple has developed a universal platform "passwordless" architecture (e.g. Apple ID and "Sign in with Apple"). See **Exhibit 18.** The architecture is universal in that it works across ecosystems, such as iOS (and the related iPadOS, watchOS, and visionOS) and MacOS, as well as on PCs and even Android. It is password-less in that passwords have been replaced with the use of passkeys. **Exhibit 19**. At Apple's Worldwide Developer Conference in 2022, Apple announced: "In macOS Monterey and iOS 15, we announced a developer preview of the solution -- passkeys -- and got so much great feedback.

In macOS Ventura and iOS 16, we're excited to make passkeys available to everyone. Now is the time to adopt them." **Exhibit 20**.

36. Apple's architecture relies on the issuance of security tokens, adopting standards OAuth 2.0 and OpenID Connect. The hub of Apple's universal platform password-less architecture is Apple ID, which receives authentication requests from external systems, coordinates the action of trusted devices, and issues security tokens.

37. Apple encourages its users to register an Apple ID to use various functionalities across its operating systems, including iOS to MacOS.

38. Apple leverages its vast userbase for "Sign in with Apple," which allows users to utilize their Apple ID to sign into a website. It starts with a button in the app or website labeled "Sign In with Apple". **Exhibit 21**.

39. Apple uses OAuth 2.0 and OpenID Connect terminology in its documentation and API calls. Applications providing "Sign in with Apple" must be registered in the Apple Developer Portal. Apple uses a public/private key client authentication methodology.

40. Apple users are typically locked into the Apple ecosystem. For example, Apple users with iPhones typically buy compatible Apple Watches, which are tightly integrated in the Apple ecosystem (other devices, like the newer models of Samsung's Galaxy Watch, do not even work with Apple devices). Apple utilizes "trusted devices" that a user owns to enhance two-factor authentication. "A trusted device is an iPhone, iPad, iPod touch, Apple Watch, or Mac that you've already signed in to using two-factor authentication. It's a device that we know is yours and that can be used to verify your identity by displaying a verification code from Apple when you sign in

on a different device or browser. **Exhibit 22**. "When you sign in with your Apple ID user name and password for the first time on a new device or the web, you'll receive a notification on your trusted devices that someone is trying to sign in with your Apple ID." *Id*.

41. In Apple's ecosystem, a user's "Apple ID is the account that you use to access all Apple services and make all of your devices work together seamlessly." **Exhibit 23**. "After you sign in, you can use the App Store, iCloud, iMessage, Apple Music, Apple TV+, and more."

42. Apple uses Apple ID (in conjunction with biometrics sensors available on various devices) across its devices, including iPhone, iPad, and Apple Vision Pro. Apple instructs its users to (1) Open the Settings app; (2) Tap Sign in to your [device]; and (3) Choose a way to sign in (see **Exhibit 23**):

43. For the "Use Another Apple Device Option," Apple instructs that "If you have another iPad or iPhone that's signed in with your Apple ID, bring that device nearby and follow

the instructions on both devices to complete sign-in." **Exhibit 23**. Further, "[i]f you're already signed in on the iPhone paired with your Apple Watch, you will automatically be signed in to that Apple ID on your watch." *Id*. Apple ID is also used for Mac computers, Apple TV, and signing into the web as well as other apps (and other devices), as described above. *Id*. Apple's users are directed to sign in using their Apple ID during the setup process (i.e., after purchase, upon first turning on) of its devices.

44. Federated authentication is not the only authentication architecture Apple has incorporated into its operating systems; Apple Pay, which incorporates EMV's payment tokenization architecture utilizes token issued by a third-party and stored on the phone. The tokens are a surrogate for a credit card, debit card, or other Primary Account Number (PAN), and can be used anywhere the underlying account is accepted as payment. As with OpenID / OAuth 2 tokens, EMV payment tokens indicate a previous authentication of the user. The tokens differ with respect to authorization claims. OpenID / OAuth 2 authenticates the user and obtains the user's consent before issuing a token. As such, the token represents authentication of the user and what the user has been authorized. As authorization is obtained before token issuance, a token can never be used for more than authorization claim it contains. EMV tokens, on the other hand, contain no claims with respect to authorization. Authorization, rather, is indicated by release of an EMV token from a device. Unlike OpenID / OAuth 2 tokens, payment tokens are locked in a device and can only be released following verification of the user by the device. As only the legitimate use can be verified, release of an EMV token is indicative of user consent. Presentation of an EMV payment token, accordingly, is indicative of user consent to the accompany charge to their account. Regardless of the differences in their manner of representing authorization and the timing of

obtaining authorization, both OpenID / OAuth 2 tokens and EMV tokens are indicative of user authentication and authorization.

45. To facilitate the use of EMV's payment tokenization architecture, Apple has deployed technology to secure payment tokens in connection with its operating systems and Apple Pay. Using biometric security features integrated into its devices (i.e., the ability to unlock Apple devices through a fingerprint, facial recognition, or an iris scan), Apple offered biometric authentication for Apple Pay beginning in 2014. *See* **Exhibit 24**.

46. As the popularity of mobile payment solutions increased, Apple Pay surfaced as a key player in the market. On information and belief, Apple Pay has over 45 million users in the United States. On further information and belief, Apple Pay's convenience and security features, attributable to its use of (biometric) authentication, are the major driving forces of its adoption among users.

47. As for Sign in with Apple, Built With indicates that 15,258 sites utilize Apple's sign-in solution. This authentication mechanism competes with the likes of "Sign in with Google," "Sing in with Facebook," and Microsoft Authentication.

II. PROXENSE AND ITS INNOVATIVE TECHNOLOGIES

48. Proxense was founded in 2001.[1] From approximately 2004-2012, Proxense developed, *inter alia*, mobile payment technologies and commercial products, employing over thirty engineers, and investing many millions of dollars in product development and other research and development efforts. Foundational capabilities of Proxense's technologies included a secure

[1] The company was formally incorporated as an LLC in 2001 under the name Margent Development LLC; in 2005, the business was renamed to Proxense LLC.

element, biometrics captured and stored thereon, retrieval of biometrics and token passing to a trusted third party, and completion of a mobile payment transaction.

49. Proxense also developed sophisticated, proprietary, proximity-based detection, authentication, and automation technology, built on the concept of wirelessly detecting, authenticating, and communicating with personal digital keys ("PDKs"). Proxense's technology enabled PDKs to run for as long as two years on tiny batteries. "ProxPay" technology also included biometrically-based user and device authentication options, the ability to conduct biometric-verified transactions without sending or exposing the underlying biometric data or storing it anywhere except the PDK, and the incorporation of a registration for maintaining or verifying the PDK. Significant financial and engineering resources were deployed to make this possible. The resulting developments became primary differentiators of Proxense's product line, and significant elements on which its business was built.

50. John Giobbi is the founder and CEO of Proxense. He is an experienced product designer and prolific inventor (a named inventor on approximately 200 patents, including the asserted patents), with over 35 years of experience as an entrepreneur and product development executive. For example, Mr. Giobbi was a Senior Vice President at WMS Gaming, and managed over 200 staff; in his six-year tenure at that company, its market capitalization soared from approximately $80 million to about $1 billion. Mr. Giobbi was also the founder and President of Prelude Technology Corp. and InPen.

51. The innovative, visionary nature of Proxense's technology was recognized in the media, beginning in mid-2008, when, The Bulletin featured a story on Proxense's mobile payment technology, titled "A pint-sized virtual wallet." Andrew Moore, The Bulletin (May 7, 2008), **Exhibit 25**. The story describes a future that greatly resembles the present-day, including a

"wireless wallet" and "fingerprint" verification, including the use of such technology to pay for goods using such wireless methods protected by biometric measures like a fingerprint. In 2009, Trend Hunter ran a similar story titled "Virtual Biometric Wallets," featuring Proxense and Mr. Giobbi. Michael Plishka, Trend Hunter (January 4, 2009), *See* **Exhibit 26**.

52. Another 2009 article, ran in DARKReading, a publication in InformationWeek's IT Network, also featured the company and Mr. Giobbi in an article titled "Startup May Just Digitize Your Wallet." George V. Hulme, DARKReading (February 8, 2009), *See* **Exhibit 27**. The DARKReading article described that Proxense was "in the process of bringing to market a proximity-based communications device that aims to provide a way to securely share information and conduct payments." Proxense's Personal Digital Keys (PDKs) were described as "carried by users, perhaps even within a cell phone, and can security hold data and manage authentication." Mr. Giobbi explained that "the data within the PDK also can be protected by additional layers of authentication, such as biometric…"

53. It would be years until products like Google Wallet (2011), Apple Pay (2014), and Samsung Pay (2015) were launched and became mainstream; Apple's Touch ID, which involves fingerprint recognition technology, for example, was introduced in 2013. Accordingly, Proxense's technology was years ahead of the industry.

54. After the launch of services like Apple Pay (as well as Apple ID, Touch ID, Face ID, and Optic ID), and their inextricable link to the some of the most popular smartphone hardware devices in the United States, and the world, Proxense would find itself unable to compete, even though Proxense invented the technology utilized in these solutions.

55. Today, Proxense holds 80 patents on related technology, including digital content distribution, digital rights management, personal authentication, biometric data management and mobile payments. Proxense continues to prosecute new patents on its proprietary technology.

III. INFRINGEMENT ALLEGATIONS

1. Proxense's Interactions with Apple

56. In or around July 25, 2016, Proxense sent a formal letter to Apple regarding its patents, including a list of granted patents, which at the time included the 730, 954, and 042 Patents. Proxense's letter also included a family tree which set forth the applications that would lead to the 289 and 989 Patents.

57. Since at least that time, Apple has had actual notice of the Patents-in-Suit and the scope of their claims. Apple has also had knowledge of the infringing nature of its activities, or at least a willful blindness regarding the infringing nature of its activities, since at least Proxense's making Apple aware of the Patents-in-Suit, if not as of the public filing of this Complaint. Despite Apple's knowledge of the Patents-in-Suit, and its constructive knowledge of its infringing actions, Apple continued to infringe the claims of the Patents-in-Suit. Apple's infringement has been and continues to be willful since at least the date of the public filing of this Complaint.

58. Apple is the assignee of U.S. Patent No. 10,171,458 ("the 458 Patent"). On May 25, 2016, during prosecution of the 458 Patent, Apple listed the 9,298,905 Patent (of which the asserted 730 and 954 are members) in an information disclosure statement to the United States Patent Office. Apple therefore had knowledge of the 9,298,905 Patent family since at least September 23, 2016. On information and belief, Apple's investigation that discovered the 9,298,905 Patent family provided it with knowledge of the other Patents-in-Suit at least as of the same time.

59. The first Complaint in this matter, filed on 03/18/2024, also provided Apple with knowledge of the Patents-in-Suit, and the way that its products infringe those patents.

60. Apple's infringement of the Patents-in-Suit is willful. Apple continues to commit acts of infringement despite a high likelihood that its actions constitute infringement, and Apple knew or should have known that its actions constituted an unjustifiably high risk of infringement.

2. The Accused Products

61. Through its own actions, and the actions of its customers and users, which Apple directs and controls, Apple has manufactured, used, marketed, sold, offered for sale, and exported from and imported into the United States devices and software that directly and/or indirectly infringe (literally or via the doctrine of equivalents) the Patents-in-Suit.

62. Apple has distributed variants of Apple Pay that have included functionality to verify a user during authentication of a smartphone (iPhone), tablet (iPad), computer (Mac, iMac, and MacBook), and augmented reality goggles (Vision Pro). Apple Pay is operable on a range of Apple devices, including at least all iPhones beginning with the iPhone 6, iPhone 6 Plus, and above, including, at least all variants of the following Apple devices: iPhone 6, iPhone 6 Plus, iPhone SE, iPhone 7, iPhone 7Plus, iPhone 8, iPhone 8 Plus, iPhone X, iPhone 11, iPhone 12, iPhone 13, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 14, iPhone 14 Pro, iPhone 14 Pro Max; and iPad models including all iPad Pro, iPad Air, iPad, and iPad mini models with Touch ID or Face ID, Apple Watch Series 1 and later, Mac models with Touch ID, Mac computers with Apple Silicon that are paired with a Magic Keyboard with Touch ID, and all Apple devices released since October 2014.[2] The current and previous versions of Apple Pay and devices running Apple Pay, alone and together, are non-limiting instances of the Accused Products. The Accused Products

[2] **Exhibit 29**.

include, for example, the representative iPhone X running Apple Pay. The current and previous versions of Apple Pay and devices with Apple Pay, alone and together, are non-limiting instances of the Accused Products. The Accused Products practice the claims of the Patents-in-Suit to improve the shopping experience of their users, and to improve Apple's position in the mobile payment market.

63. These Accused Products also include devices shipped with at least iOS 16, iPadOS 16, macOS 13, and visionOS 1.0 (i.e., all Vision Pro products), or devices on which Apple makes these versions available, which are capable of using passkeys. These Accused Products include an integrated personal digital key and an integrated receiver decoder circuit as set forth herein. These Accused Products practice the claims of the Patents-in-Suit to eliminate the creation, management, or use of passwords. Apple touts the use of passkeys as "profoundly improv[ing] security." **Exhibit 28**.

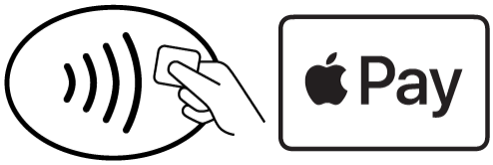
3. Apple's Infringement of the Patents-in-Suit

a. Apple Pay

64. Apple directly infringes the Patents-in-Suit by, for example, selling devices preloaded with Apple Pay. Apple Pay is "built into iPhone, Apple Watch, Mac, and iPad." **Exhibit 30**. Apple also actively induces infringement the Patents-in-Suit by disseminating Apple Pay as the default payment mechanism across its product line. "Start by adding your credit or debit card to the Wallet app on your iPhone, and you'll have the option to add it to your other devices in one easy step." *Id*. For purposes of this complaint and Proxense's infringement reads, Apple Pay is used interchangeably with Apple Wallet and the Wallet app, which provides Apple Pay functionality.

65. When an Apple device equipped with Apple Pay is presented to make a contactless payment transaction, the user infringes the Patents-in-Suit.

66. Apple devices can make contactless payments. **Exhibit 24**. ("With your Apple Cash, Apple Card, and other credit and debit cards stored in the Wallet app on iPhone, you can use Apple Pay for secure, contactless payments in stores, restaurants, and more."). To do so, users can hold their Apple Pay equipped device close to a payment terminal. "You can use Apple Pay wherever you see contactless payment symbols such as the following:"



Id.

67. When using Apple Pay, users will be prompted to authenticate using biometrics. For example, using an iPhone with Face ID or Touch ID. See **Exhibit 24**. Apple's instructions to its users in connection with the use of Apply Pay using Face ID and Touch ID are set forth below:

Pay with your default card on an iPhone with Face ID

1. Double-click the side button.
2. When your default card appears, glance at iPhone to authenticate with Face ID, or enter your passcode.
3. Hold the top of your iPhone near the card reader until you see Done or a checkmark on the screen.

Pay with your default card on an iPhone with Touch ID

1. Rest your finger on Touch ID.
2. Hold the top of your iPhone near the card reader until you see Done or a checkmark on the screen.

68. Unlocking a phone with either Face ID or Touch ID (or Retina ID) requires receiving scan data from a biometric scan and comparing the scan data to biometric data

persistently stored in a tamperproof format. "On iPhone, iPad, Apple Watch, Mac computers with Touch ID, and Mac computers with Apple silicon that use the Magic Keyboard with Touch ID, the Secure Enclave manages the authentication process and allows a payment transaction to proceed." **Exhibit 31** at p. 131 (hereinafter "Apple Platform Security").

69. "[B]efore information is transmitted, the user must authenticate using Face ID, Touch ID, or their passcode." *Id*. at 136. "No payment information is sent without user authentication." *Id*. Biometric authentication on Apple's platform involves the Secure Enclave, which persistently stores biometric data (*i.e*., Touch ID and Face ID template data) in a tamperproof format. "The Secure Enclave is a dedicated secure subsystem integrated into Apple systems on chip (SoCs)." **Exhibit 32**. "The Secure Enclave is isolated from the main processor to provide an extra layer of security and is designed to keep sensitive user data secure even when the Application Processor kernel becomes compromised." *Id*. "It follows the same design principles as the SoC does—a boot ROM to establish a hardware root of trust, an AES engine for efficient and secure cryptographic operations, and protected memory." *Id*. A graphical overview of the secure enclave is set forth below:

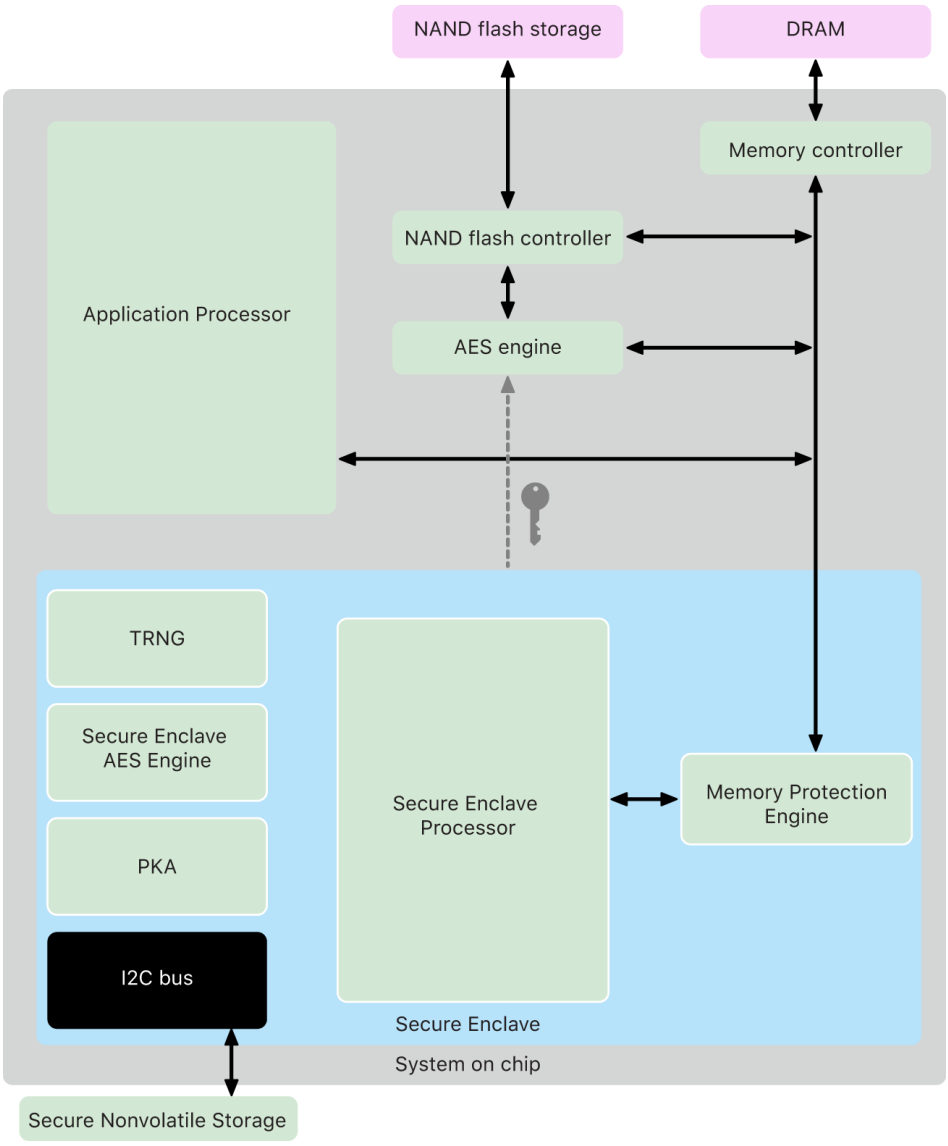


Exhibit 32.

70. In response to a request for biometric authentication, a user places a finger on the fingerprint sensor or looks at their device. "The sensor captures the biometric image and securely transmits it to the Secure Enclave." Apple Platform Security, p. 18. Secure transmission of the biometric image is accomplished when the image is "encrypted and authenticated with a session key that's negotiated using a shared key provisioned for each Touch ID sensor and its corresponding Secure Enclave at the factory." *Id.*, p. 19.

71. During user enrollment for Face ID, Touch ID, or Retina ID, biometric data is vectorized into a map of nodes and securely stored by the Secure Element. "Although the Secure Element doesn't include storage, it has a mechanism to store information securely on attached storage separate from the NAND flash storage that's used by the Application Processor and operating system." Apple Platform Security, page 9. Consequently, "the resulting map of nodes is stored in an encrypted format that can be read only the Secure Enclave as a template to compare against future matches…" Apple Platform Security, page 19. The Secure Enclave, therefore, retrieves and decrypts the stored template and compares the scan data received from the sensor against the stored template to determine if they match.

72. If the scan matches the template, "the Secure Enclave then sends signed data about the type of authentication and details about the transaction to the (contactless or within apps) to the Secure Element." Apple Platform Security, page 138. "It's securely delivered to the Secure Element by leveraging the pairing key." *Id*.

73. "After a credit, debit, or prepaid card (including store cards) payment is authorized by the cardholder using Face ID, Touch ID, or a passcode, or on an unlocked Apple Watch by double-clicking the side button, contactless responses prepared by the payment applets within the Secure Element are exclusively routed by the Controller to the NFC field." Apple Platform Security, page 134. A Device Account Number (*i.e.*, payment token) is only released from the Secure Element and transmitted to the payment terminal after biometric verification of the user. Accordingly, "[a]fter the user authenticates, the Device Account Number and a transaction-specific dynamic security code are used when processing payments." Apple Platform Security, page 138. "Neither Apple nor a user's device sends the full credit or debit card numbers to merchants." *Id*.

74. "When you make a purchase, Apple Pay uses a device-specific number and unique transaction code." *Id*. "So your card number is never stored on your device or on Apple servers." *Id*. "And when you pay, your card numbers are never shared by Apple with merchants." *Id*.

75. Apple Pay's process follows the EMV contactless specification. See e.g. **Exhibit 33**. ("The Secure Element IC and the Java Card platform are certified in accordance with the EMVCo Security Evaluation process. After the successful completion of the security evaluation, EMVCo issues unique IC and platform certificates."). EMV, which originally stood for "Europay, Mastercard, and Visa," the three companies which created the standard, is a payment method based upon a technical standard for smart payment cards and for payment terminals and automated teller machines which can accept them. Notably, the relevant EMV specifications were first published in 2007, years after the priority dates of all the Patents-in-Suit.

76. EMVCo, LLC ("EMVCo") facilitates worldwide interoperability and acceptance of secure payment transactions. EMVCo is supported by dozens of banks, merchants, processors, vendors and other industry stakeholders, including Apple. EMVCo manages and evolves the EMV Specifications and related testing processes. This includes, but is not limited to, card and terminal evaluation, security evaluation, and management of interoperability issues.

77. One means of promoting payment security is "tokenization," an approach adopted by Apple for Apple Pay that substitutes sensitive data like account numbers and other personally identifiable information with a non-sensitive equivalent that has no intrinsic or exploitable meaning or value.

78. EMV payment tokens are open-loop tokens provisioned by a token service provider ("TSP"). Like other tokens, EMV payment tokens are used to replace the actual payment credential (*e.g.*, PAN) with another numeric value.

79. The U.S. Payments Forum (formerly the EMV Migration Forum) is a cross industry body focused on supporting the introduction and implementation of EMV chip and other new and emerging technologies that protect the security of and enhance opportunities for payment transactions. According to the U.S. Payments Forum, Apple was "among the first to implement EMV payment tokens in digital wallets that hold credentials for several payments use cases." **Exhibit 34**.

80. On information and belief, EMV payment tokens are issued to Apple Pay equipped devices in exchange for a credit card number by a TSP such as Visa, Mastercard, American Express or Discover. Device-specific payment tokens are stored by Apple Pay-equipped devices.

81. The device-specific EMV payment tokens are stored by Apple's Secure Element. "The Secure Element is an industry-standard certified chip running the Java Card platform, which is compliant with financial industry requirements for electronic payments." Apple Platform Security, p. 131. "The Secure Element IC and the Java Card platform are certified in accordance with the EMVCo Security Evaluation process." *Id*. After the successful completion of the security evaluation, EMVCo issues unique IC and platform certificates. *Id*.

82. After a transaction authorization is initiated, a corresponding message is sent to the merchant acquirer/processor containing the payment token from the cardholder's device, along with a unique cryptogram.

83. The merchant acquirer/processor receives the transaction request, uses the token (which looks like a PAN) to perform a token Bank Identification Number ("BIN") lookup, and determines the networks to which the transaction can be routed. The merchant acquirer/processor routes the transaction to the appropriate payment network (based on the preferred routing choice, least cost, or some other criterion agreed to with the merchant).

84. The payment network then determines that the transaction is based on a token BIN and issues a request to the appropriate token service provider ("TSP") to validate the unique cryptogram and detokenize the token to the PAN. The TSP verifies the cryptogram and returns the clear PAN to the payment network. As the token is the "unique Device Account Number" held by the iPhone's Secure Element, the token is a unique device ID utilized to authenticate the device.

85. The payment network then forwards the transaction with the clear PAN to the appropriate issuer processor. Since the PAN is only unlocked by the validation of the unique device account number token, forwarding of the PAN with the authorization request is an access message.

86. The issuer processor forwards the authorization request, with the clear PAN, to the issuer. The issuer completes final authorization and sends an authorization response to the issuer processor. Accordingly, the access message comprising the PAN authorization request allows the user access to the issuer's computer software necessary to process and the authorize the payment.

87. The issuer processor then sends the authorization response to the payment network. The payment network sends the authorization response to the merchant acquirer/processor, ensuring that the token, not the clear PAN, is included.

88. The merchant acquirer/processor responds to the contactless terminal to complete the transaction. Meanwhile, the issuer processor sends a transaction completion notification, with the token, to the TSP, indicating the outcome of the transaction. The TSP then pushes a notification to the mobile device on which the token was initially provisioned during the enrollment process – which is another variant of an access message within the scope of the claims.

b. Sign in with Apple and Apple ID

89. Apple has introduced Passkeys as the replacement to passwords.

90. Apple devices using iOS 16, iPadOS 16, macOS 13, or visionOS 1 (or later), which are capable of utilizing Passkeys, include an integrated personal digital key ("PDK") and an integrated receiver decoder circuit. These devices utilize the WebAuthn standard and use public-key cryptography (**Exhibit 20**), which in turn requires memory for storing information particular to a user.

91. Under the WebAuthn specification, "compliant authenticators protect public key credentials." W3C Specification, Web Authentication: An API for accessing Public Key Credentials Level 2 (April 8, 2021) ("hereinafter, W3C Specification"), available at **Exhibit 35**, § 1. A public key credential refers to a public key credential source, which includes a credential ID. *Id*., § 4 (defining "public key credential" and "public key credential source"). The credential ID uniquely identifies its public key credential source. *Id*. (defining a credential ID as a "probabilistically-unique byte sequence identifying a public key credential source and its authentication assertions"). In addition to the credential ID, each public key credential source contains a "credential private key." *Id*. (defining "public key credential source" as a data structure including the credential private key and the credential ID). "The credential private key is bound to a particular authenticator" and part of an asymmetric key pair containing a public key returned to a relying party. *Id*. (defining a "credential key pair") Apple devices using iOS 16, iPadOS 16, macOS 13, or visionOS 1 (or later), which are capable of using Passkeys, ("Apple Passkey-capable Devices") must store within memory a credential comprising a private key of an asymmetric key pair and a credential ID uniquely identifying the private/public key pair to which the private key belongs.

92. The credentials stored within Apple Passkey-capable Devices can only be accessed with the appropriate access key. "A public key credential can only be used for authentication with

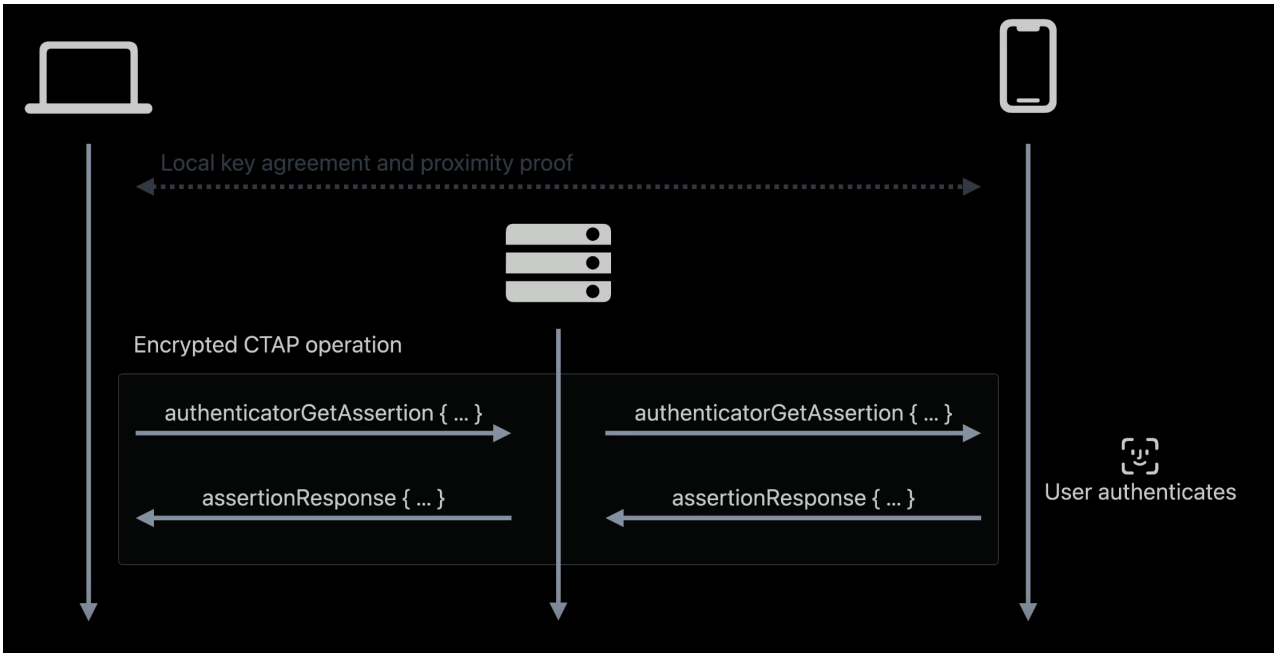
the same entity (as identified by the RP ID) it was registered with." W3C Specification, § 4 ("A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with"). When generating a response, an authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external server. Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2.2, **Exhibit 36**, ("7.1 If the allowList parameter is present and is non-empty, locate all denoted credentials created by this authenticator and bound to the specified rpId. 7.2 If an allowList is not present, locate all discoverable credentials that are created by this authenticator and bound to the specified rpId."). Since only credentials corresponding to the RP ID will be retrieved, the RP ID is an access key. In Apple's words, "the system will take care of only letting me use it in the correct app or website, with strong built-in phishing resistance." **Exhibit 20**.

93. Apple Passkey-capable Devices will only return credentials corresponding to the RP ID access key provided by the external relying party; these devices include a controller and memory.

94. Apple further described in its WWDC conference in 2022 that:

> Passkeys can also be used to sign in across devices in a secure, phishing-resistant manner. Here's how that works. There are two devices here. The client, which is the device or web browser where I'm signing in, and the authenticator, which is the device which has my passkey. First, the client shows a QR code, which the authenticator scans. This QR code contains a URL that encodes a pair of single-use encryption keys. Then, the authenticator produces a Bluetooth advertisement containing routing information for a network relay server. Once the local exchange and key agreement have happened, the two devices connect to a relay server picked by the phone. From there, they perform a standard FIDO CTAP operation, which is encrypted using the keys from earlier, so the relay server can't see anything that's going on. This whole process is performed by the device and the web browser. The website is not involved at any point in the cross-device communication.

Exhibit 20. The process is shown in the below figure:



Id.

95. Establishing a connection over the internet requires utilizing either a device's Wi-Fi or cellular capabilities (i.e., a wireless protocol). Accordingly, Apple Passkey-capable Devices must have the antenna and transceiver necessary to implement the wireless protocol(s) enabling transmission over the internet.

96. Apple Passkey-capable Devices store local, secure biometric information for authenticating users. "Passkeys are a replacement for passwords. They are faster and easier to sign in. Just use Touch ID or Face ID to authenticate and you're done." **Exhibit 37**. "Apple platforms will always require UV [user verification] for passkeys when biometrics are available, so you don't have to worry about that." **Exhibit 20**.

97. As described above with respect to Apple Pay, Apple devices utilize the Secure Enclave to securely store biometric information for authentication. "During enrollment, the Secure Enclave processes, encrypts, and stores the corresponding Touch ID and Face ID template data." Apple Platform Security, page 19.

98. "The Secure Enclave is a dedicated secure subsystem … isolated from the main processor to provide an extra layer of security designed to keep sensitive data secure even when the Application Processor kernel becomes compromised." Apple Platform Security, page 9. Accordingly, the biometric data for Touch ID and Face ID are securely stored.

99. As detailed above, Apple Passkey-capable Devices enable the use of passkeys to sign in across devices in a secure, phishing-resistant manner by utilizing either the device's Wi-Fi or cellular capabilities – both of which are wireless protocols. Consequently, Apple Passkey-capable Devices are capable of communicating wirelessly with an external receiver decoder circuit ("RDC").

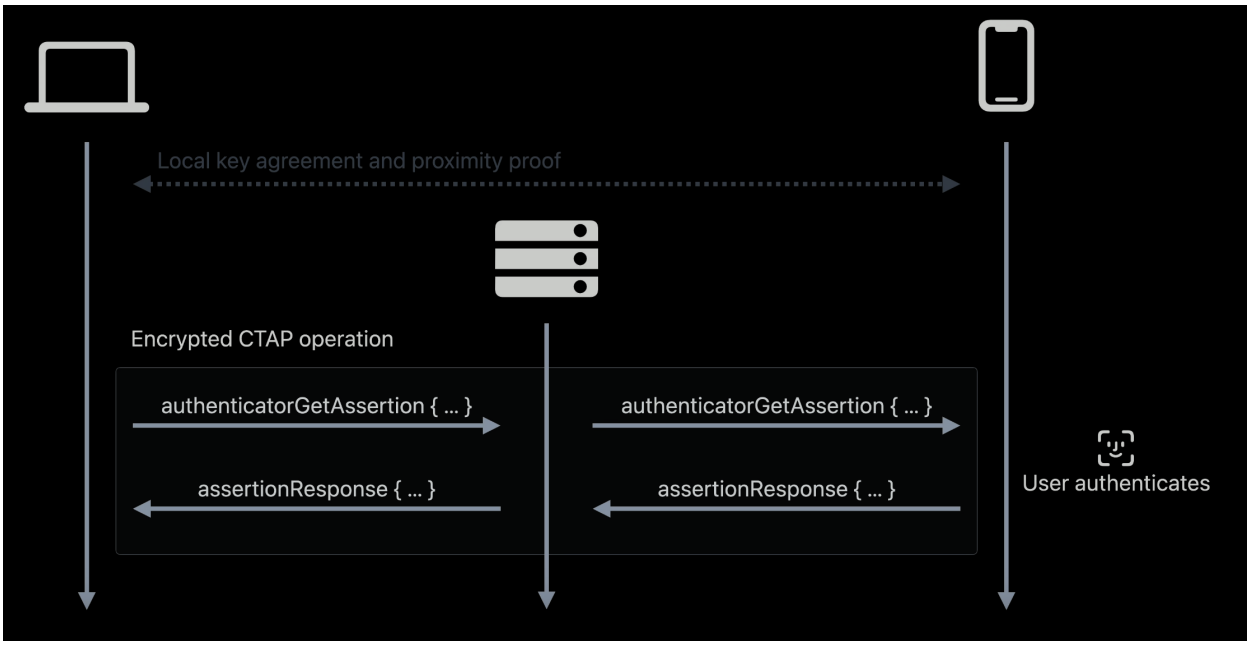
100. Apple Passkey-capable Devices communicate passkey signatures over an encrypted connection, through the internet, via Wi-Fi, and/or cellular protocols. Accordingly, Apple Passkey-capable Devices include an RDC enabling wireless communications with at least one external device, such as another Apple Passkey-capable Device (e.g. a Mac running macOS 13).

101. Apple Passkey-capable Devices also include a signal line for communication that couples the integrated RDC to the integrated PDK. As noted above, using a passkey on Apple iPhone to sign into a website on external device begins by scanning a QR code.

> First, the client shows a QR code, which the authenticator scans. This QR code contains a URL that encodes a pair of single-use encryption keys. Then, the authenticator produces a Bluetooth advertisement containing routing information for a network relay server. Once the local exchange and key agreement have happened, the two devices connect to a relay server picked by the phone. From there, they perform a standard FIDO CTAP operation, which is encrypted using the keys from earlier, so the relay server can't see anything that's

going on. This whole process is performed by the device and the web browser. The website is not involved at any point in the cross-device communication.

Exhibit 20. The process is shown in the below figure:



102. As shown in the figure, an "authenticatorGetAssertion" is forwarded to the phone, which is a request to provide cryptographic proof of user authentication. Client to Authenticator Protocol (CTAP) (fidoalliance.org), **Exhibit 36**, § 6.2 (defining authenticatorGetAssertion as the method "used by a host to request cryptographic proof of user authentication as well as user consent to a given transaction, using a previously generated credential that is bound to the authenticator and relying party identifier"). The authenticatorGetAssertion request contains a relying party identifier (RP ID) access key. *Id*. (defining the input parameters of the authenticatorGetAssertion as including a required relying party identifier.). But a passkey "can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." W3C Specification, § 4 ("A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with."). Accordingly, the PDK on the

phone must receive the RP ID access key, unlock the necessary passkey to generate cryptographic proof, and send the proof back to the external device via the relay server. The PDK within the Apple Passkey-capable Device can only do so if the integrated RDC receiving the authenticatorGetAssertion and returning the cryptographic proof is communicatively coupled to the PDK. Accordingly, Apple Passkey-capable Devices necessarily have an integrated RDC coupled to the integrated PDK by a first signal line for communication.

103. Apple Passkey-capable Devices also necessarily have a second signal line coupling the RDC to at least one other component. To function, the RDC must receive power from a battery and/or be coupled to at least one application processor or similar processing unit and/or one or more antenna.

104. Authentication is a service provided by the relying party, and the credential ID is necessary for the relying party to perform the authentication function. Upon receiving the response, the relying party will use the credential ID to locate the appropriate public key to verify a signature generated with the private key held by the authenticator. W3C Specification, § 7.2 ("7. Using credential.id (or credential.rawId, if base64url encoding is inappropriate for your use case), look up the corresponding credential public key and let credentialPublicKey be that credential public key… 20. Using credentialPublicKey, verify that sig is a valid signature over the binary concatenation of authData and hash… 22. If all the above steps are successful, continue with the authentication ceremony as appropriate. Otherwise, fail the authentication ceremony.") "[I]f an authenticator returns the wrong credential ID, or if an attacker intercepts and manipulates the credential ID, is that the WebAuthn Relying Party would not look up the correct credential public key with which to verify the returned signed authenticator data (a.k.a., assertion), and thus the interaction would end in an error." *Id.* § 13.1. As the proper credential ID is needed for the relying

party to authenticate a user, and the credential ID held within the PDK of the Apple Passkey-capable Device is included within a response to the authenticatorGetAssertion request generated, the PDK of the Apple Passkey-capable Device enables authentication by a relying party.

105. Proxense has at all times complied with the marking provisions of 35 U.S.C. § 287 with respect to the Patents-in-Suit. On information and belief, any prior assignees and licensees have also either complied with the marking provisions of 35 U.S.C. § 287, or else were excused from the obligation to mark because § 287 does not apply.

CLAIM 1
(withdrawn)

106. Intentionally left blank.

107. Intentionally left blank.

108. Intentionally left blank.

109. Intentionally left blank.

110. Intentionally left blank.

111. Intentionally left blank.

112. Intentionally left blank.

113. Intentionally left blank.

114. Intentionally left blank.

115. Intentionally left blank.

116. Intentionally left blank.

117. Intentionally left blank.

118. Intentionally left blank.

119. Intentionally left blank.

CLAIM 2
(Infringement of the 730 Patent)

120. Proxense repeats and realleges all preceding paragraphs, as if fully set forth herein.

121. Proxense has not licensed or otherwise authorized Apple to make, use, offer for sale, sell, or import any products that embody the inventions of the 730 Patent.

122. Apple infringes at least claims 1, 2, 3, 5, 15, 16, and 17 of the 730 Patent in violation of 35 U.S.C. § 271 with respect to the accused products. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

123. For example, Apple directly infringes at least claims 1, 2, 3, and 5 of the 730 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering to sell access within the United States the Apple Pay service, associated Apple Pay software necessary to access the service and products including Apple Pay. That software performs/executes, and those products provide, a method for verifying a user during authentication of the device.

124. As described supra, Apple devices with Apple Pay persistently store biometric user data, e.g., a fingerprint, face scan, and/or iris profile of a user, and codes and other data values comprising a device ID code uniquely identifying the integrated device and a secret decryption value, in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered. "During enrollment, the Secure Enclave processes, encrypts, and stores the corresponding Touch ID and Face ID template data." Apple Platform Security, page 19.

125. Apple devices with Apple Pay safeguard financial information, e.g., credit card information, with EMV payment tokens as described *supra*. Indeed, Apple was among the first to implement EMV payment tokens in digital wallets that hold credentials. On information and belief, the EMV payment tokens utilized by Apple Pay uniquely identify the Apple device. On

further information and belief, the payment tokens are stored in a secure element that is tamper proof, e.g., the Secure Element. Indeed, adding a card to Apple Pay on an iPhone causes "a unique Device Account Number [to be] created, encrypted, and then stored in the Secure Element." Apple Platform Security, page 142.

126. Apple devices with Apple Pay also contain a secret decryption value, also called a "private key," which is used for, inter alia, decrypting. "Communication between the Secure Enclave and the Secure Element takes place over a serial interface…Though not directly connected, the Secure Enclave and Secure Element can communicate securely using a shared pairing key that provisioned during the manufacturing process." Apple Platform Security, page 144. For the shared key to persist, it must be stored by both the Secure Element and the Secure Enclave. "The Secure Enclave is a dedicated secure subsystem … isolated from the main processor to provide an extra layer of security designed to keep sensitive data secure even when the Application Processor kernel becomes compromised." Apple Platform Security, page 9. The secret decryption value, like the biometric data of the user and plurality of codes and other data values comprising a device ID, is tamper-proof as a result of being securely stored on Apple devices.

127. As described supra, Apple devices with Apple Pay may utilize biometric data, such as fingerprint and/or retina scan data from a biometric scan, and verify (e.g., authenticate) biometric data of a user. The location of a purchase may determine how the request for verification is received (e.g., a push notification received from a merchant's website, an API call from a merchant app installed on the device, or direct prompt for Apple Pay when using NFC in stores). When an Apple device utilizes and verifies fingerprint biometric data, for example, scan data is compared with the fingerprint data on the device to determine whether there is a match.

128. After receiving the determination that the scan data matches the biometric data, Apple devices with Apple Pay wirelessly send one or modes codes, including device ID codes, regardless of where the user is shopping. For example, Apple devices wirelessly transmit EMV payment tokens. EMV payment tokens, as detailed *supra*, are values uniquely identifying Apple Pay preloaded smartphones provided during card enrollment, and thus are an identified embodiment of device ID codes. The payment tokens may be authenticated by an agent, e.g., a token service provider.

129. Apple devices with Apple Pay receive an access message from agents that are a third-party trusted authority, e.g., token service providers which are responsible for, inter alia, de-tokenization. These providers also maintain a token vault, which is a "repository that maintains the established Payment Token / Token Expiry Date mapping to the underlying PAN / PAN Expiry Date and includes Payment Token related data." The providers compare the token, which includes one or more codes from a plurality of codes and other data values including a device ID device, wirelessly transmitted by an Apple device to the tokens stored in its repository for authentication.

130. When using Apple Pay to pay in-store, the user "holds the top of [their] iPhone near the contactless read until [they] see the Done and a checkmark on the display." **Exhibit 38**. The push notification from the agent indicates that the user has been allowed access to an application, for example, an ATM machine, computer software, a web site and/or a file.

131. Apple has induced infringement, and continues to induce infringement, of at least claims 1, 2, 5, 6, 8 and 9 of the 730 Patent in violation of 35 U.S.C. § 271 by providing the Apple Pay software, along with a substantial knowledge base teaching about the features, use and integration of Apple Pay, to sellers, resellers, and end-user customers who transact using Apple Pay via the Accused Products. For example, Apple induces infringement of at least claim 1 and 8

of the 730 Patent by making the software application Apple Pay available for use on Apple devices. The software, which is pre-installed by Apple across its ecosystem, including iPhones, iPads, Macs, and even the Vision Pro, creates an integrated device in accordance with claim 8. When executed by a user for its intended and advertised purpose, the software performs/executes a method in accordance with claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

132. Apple contributes to direct infringement of at least claims 1, 2, 5, 6, 8 and 9 of the 730 Patent in violation of 35 U.S.C. § 271(c) by providing Apple Pay pre-installed, along with a publicly-accessible knowledge base which includes the claimed limitations. For example, Apple contributes to infringement of at least claim 1 and 8 of the 730 Patent by making the software application Apple Pay available for use on Apple mobile phones. When downloaded and installed, or pre-installed by Apple, the software creates an integrated device in accordance with claim 8. When executed by a user for its intended and advertised purpose, the software performs/executes a method in accordance with claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

133. Apple received constructive notice of the 730 Patent at least as early as July 2016 and actual notice of the 730 Patent at least as early as the filing of this Complaint. See **Exhibit 39**. Apple performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness that the acts would constitute direct and/or indirect infringement of the 730 Patent.

134. Proxense has been injured and seeks damages to adequately compensate it for Apple's infringement of the 730 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

135. Upon information and belief, Apple will continue to infringe (both directly and indirectly) the 730 Patent unless permanently enjoined by this Court. Pursuant to 35 U.S.C. § 283, Proxense is entitled to a permanent injunction against further infringement of the 730 Patent by Apple.

CLAIM 3
(Infringement of 954 Patent)

136. Proxense repeats and realleges all preceding paragraphs, as if fully set forth herein.

137. Proxense has not licensed or otherwise authorized Apple to make, use, offer for sale, sell, or import any products that embody the inventions of the 954 Patent.

138. Apple infringes at least claims 1, 2, 3, 5, 6, 7, 22, 23, 24, 25, 26, and 27 of the 954 Patent in violation of 35 U.S.C. § 271 with respect to the accused products. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

139. For example, Apple directly infringes at least claims 1, 2, 3, 5, 6, and 7 of the 954 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering to sell access within the United States the Apple Pay software/service and products including Apple Pay. That software performs/executes, and those products provide, a method for verifying a user during authentication of the device.

140. As described supra, Apple devices with Apple Pay persistently store biometric user data, e.g., a fingerprint, face scan, and/or iris profile, of a user, and codes and other data values comprising a device ID code uniquely identifying the integrated device and a secret decryption value, in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered. "During enrollment, the Secure Enclave processes, encrypts, and stores the corresponding Touch ID and Face ID template data." Apple Platform Security, page 19.

141. Apple devices with Apple Pay safeguard financial information, e.g., credit card information, with EMV payment tokens as described *supra*. Indeed, Apple was among the first to implement EMV payment tokens in digital wallets that hold credentials. On information and belief, the EMV payment tokens utilized by Apple Pay uniquely identify the Apple device. On further information and belief, the payment tokens are stored in a secure element that is tamper proof, e.g., the Secure Element. Indeed, adding a card to Apple Pay on an iPhone causes "a unique Device Account Number [to be] created, encrypted, and then stored in the Secure Element." Apple Platform Security, page 142.

142. Apple devices with Apple Pay also contain a secret decryption value, also called a "private key," which is used for, inter alia, decrypting. "Communication between the Secure Enclave and the Secure Element takes place over a serial interface…Though not directly connected, the Secure Enclave and Secure Element can communicate securely using a shared pairing key that provisioned during the manufacturing process." Apple Platform Security, page 144. For the shared key to persist, it must be stored by both the Secure Element and the Secure Enclave. "The Secure Enclave is a dedicated secure subsystem … isolated from the main processor to provide an extra layer of security designed to keep sensitive data secure even when the Application Processor kernel becomes compromised." Apple Platform Security, page 9. The secret decryption value, like the biometric data of the user and plurality of codes and other data values comprising a device ID, is tamper-proof as a result of being securely stored on Apple devices.

143. As described supra, Apple devices with Apple Pay may utilize biometric data, such as fingerprint and/or retina scan data from a biometric scan, and verify (e.g., authenticate) biometric data of a user. The location of a purchase may determine how the request for verification

is received (e.g., a push notification received from a merchant's website, an API call from a merchant app installed on the device, or direct prompt for Apple Pay when using NFC in stores). When an Apple device utilizes and verifies fingerprint biometric data, for example, scan data is compared with the fingerprint data on the device to determine whether there is a match.

144. After receiving the determination that the scan data matches the biometric data, Apple devices with Apple Pay wirelessly send one or modes codes, including device ID codes, regardless of where the user is shopping. For example, Apple devices wirelessly transmit EMV payment tokens. EMV payment tokens, as detailed *supra*, are values uniquely identifying Apple Pay preloaded smartphones provided during card enrollment, and thus are an identified embodiment of device ID codes. The payment tokens may be authenticated by an agent, e.g., a token service provider.

145. Apple devices with Apple Pay receives an access message from agents that are a third-party trusted authority, e.g., token service providers which are responsible for, inter alia, de-tokenization. These providers also maintain a token vault, which is a "repository that maintains the established Payment Token / Token Expiry Date mapping to the underlying PAN / PAN Expiry Date and includes Payment Token related data." The providers compare the token, which includes one or more codes from a plurality of codes and other data values including a device ID device, wirelessly transmitted by an Apple device to the tokens stored in its repository for authentication.

146. When using Apple Pay to pay in-store, the user "holds the top of [their] iPhone near the contactless read until [they] see the Done and a checkmark on the display." **Exhibit 38**. The push notification from the agent indicates that the user has been allowed access to an application.

147. Apple has induced infringement, and continues to induce infringement, of at least claims 1, 2, 3, 5, 6, and 7 of the 954 Patent in violation of 35 U.S.C. § 271 by providing the Apple

Pay software, along with a substantial knowledge base teaching about the features, use and integration of Apple Pay, to sellers, resellers, and end-user customers who transact using Apple Pay via the Accused Products. For example, Apple induces infringement of at least claim 1 of the 954 Patent by making the software application Apple Pay available for use on Apple devices. The software, which is pre-installed by Apple across its ecosystem, including iPhones, iPads, Macs, and even the Vision Pro, creates an integrated device in accordance with claim 12. When executed by a user for its intended and advertised purpose, the software performs/executes a method in accordance with claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

148. Apple contributes to direct infringement of at least claims 1, 2, 3, 5, 6, and 7 of the 954 in violation of 35 U.S.C. § 271(c) by providing Apple Pay pre-installed, along with a publicly-accessible knowledge base which includes the claimed limitations. For example, Apple contributes to infringement of at least claim 1 of the 954 Patent by making the software application Apple Pay available for use on Apple mobile phones. When downloaded and installed, or pre-installed by Apple, the software creates an integrated device in accordance with claim 12. When executed by a user for its intended and advertised purpose, the software performs/executes a method in accordance with claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

149. Apple received constructive notice of the 954 Patent at least as early as July 2016 and actual notice of the 954 Patent at least as early as the filing of this Complaint. See **Exhibit 39**. Apple performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness that the acts would constitute direct and/or indirect infringement of the 954 Patent.

150. Proxense has been injured and seeks damages to adequately compensate it for Apple's infringement of the 954 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

151. Upon information and belief, Apple will continue to infringe (both directly and indirectly) the 954 Patent unless permanently enjoined by this Court. Pursuant to 35 U.S.C. § 283, Proxense is entitled to a permanent injunction against further infringement of the 954 Patent by Apple.

Claim 4
(Infringement of 989 Patent)

152. Proxense repeats and realleges all preceding paragraphs, as if fully set forth herein.

153. Proxense has not licensed or otherwise authorized Apple to make, use, offer for sale, sell, or import any products that embody the inventions of the 989 Patent.

154. Apple infringes at least claims 1-6 of the 989 Patent in violation of 35 U.S.C. § 271 with respect to the accused products. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

155. For example, Apple directly infringes at least claim 1 of the 989 Patent by making, using (e.g., performing/executing), selling access to, and/or offering to sell access within the United States the Apple Pay software/service and products including Apple Pay. That software performs/executes, and those products provide, a method for verifying a user during authentication of the device.

156. As described supra, Apple devices with Apple Pay, including smartphones specified in the claims, persistently store biometric user data, e.g., a fingerprint, face scan, and/or iris profile, of a user, and an ID code uniquely identifying the device. "During enrollment, the

Secure Enclave processes, encrypts, and stores the corresponding Touch ID and Face ID template data." Apple Platform Security, page 19.

157. Apple devices with Apple Pay safeguard financial information, e.g., credit card information, with EMV payment tokens as described supra. Indeed, Apple was among the first to implement EMV payment tokens in digital wallets that hold credentials. On information and belief, the EMV payment tokens utilized by Apple Pay uniquely identify the Apple device. On further information and belief, the payment tokens are stored in a secure element that is tamper proof, e.g., the Secure Element. Indeed, adding a card to Apple Pay on an iPhone causes "a unique Device Account Number [to be] created, encrypted, and then stored in the Secure Element." Apple Platform Security, page 142.

158. As described supra, Apple devices with Apple Pay may utilize biometric data, such as fingerprint and/or retina scan data from a biometric scan, and verify (e.g., authenticate) biometric data of a user. The location of a purchase may determine how the request for verification is received (e.g., a push notification received from a merchant's website, an API call from a merchant app installed on the device, or direct prompt for Apple Pay when using NFC in stores). When an Apple device utilizes and verifies fingerprint biometric data, for example, scan data is compared with the fingerprint data on the device to determine whether there is a match.

159. After receiving the determination that the scan data matches the biometric data, Apple devices with Apple Pay wirelessly send one or modes codes, including a device ID, regardless of where the user is shopping. For example, Apple devices wirelessly transmit EMV payment tokens. EMV payment tokens, as detailed *supra*, are values uniquely identifying Apple Pay preloaded smartphones provided during card enrollment, and thus are an identified

embodiment of device ID codes. The payment tokens may be authenticated by an agent, e.g., a token service provider.

160. When using Apple Pay to pay in-store, the user "holds the top of [their] iPhone near the contactless read until [they] see the Done and a checkmark on the display." **Exhibit 38**. The push notification from the agent indicates that the user has been allowed access to an application.

161. Apple has induced infringement, and continues to induce infringement, of at least claims 1-6 of the 989 Patent in violation of 35 U.S.C. § 271 by providing the Apple Pay software, along with a substantial knowledge base teaching about the features, use and integration of Apple Pay, to sellers, resellers, and end-user customers who transact using Apple Pay via the Accused Products. For example, Apple induces infringement of at least claim 1 of the 989 Patent by making the software application Apple Pay available for use on Apple devices. When executed by a user for its intended and advertised purpose, the software performs/executes a method in accordance with claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

162. Apple contributes to direct infringement of at least claims 1-6 of the 989 in violation of 35 U.S.C. § 271(c) by providing Apple Pay pre-installed, along with a publicly-accessible knowledge base which includes the claimed limitations. For example, Apple contributes to infringement of at least claim 1 of the 989 Patent by making the software application Apple Pay available for use on Apple mobile phones. When executed by a user for its intended and advertised purpose, the software performs/executes a method in accordance with claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

163. Apple received constructive notice of the application that led to the 989 Patent at least as early as July 2016 and actual notice of the 989 Patent at least as early as the filing of this Complaint. Apple performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness that the acts would constitute direct and/or indirect infringement of the 989 Patent.

164. Proxense has been injured and seeks damages to adequately compensate it for Apple's infringement of the 989 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

165. Upon information and belief, Apple will continue to infringe (both directly and indirectly) the 989 Patent unless permanently enjoined by this Court. Pursuant to 35 U.S.C. § 283, Proxense is entitled to a permanent injunction against further infringement of the 989 Patent by Apple.

CLAIM 5
(Infringement of 289 Patent)

166. Proxense repeats and realleges all preceding paragraphs, as if fully set forth herein.

167. Proxense has not licensed or otherwise authorized Apple to make, use, offer for sale, sell, or important any products that embody the inventions of the 289 Patent.

168. Apple infringes at least claims 1, and 11 of the 289 Patent in violation of 35 U.S.C. § 271 with respect to the Accused Products capable of utilizing Passkeys (i.e., the Apple Passkey-compatible Devices). Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

169. For example, Apple directly infringes at least claims 1 and 11 of the 289 Patent by making, using (*e.g.*, performing/executing), selling, and/or offering to sell within the United States the Apple Passkey-compatible Devices.

170. As described *supra*, Apple's Passkey-compatible Devices include an integrated personal digital key and an integrated receiver decoder circuit. ("In macOS Monterey and iOS 15, we announced a developer preview of the solution -- passkeys -- and got so much great feedback. In macOS Ventura and iOS 16, we're excited to make passkeys available to everyone. Now is the time to adopt them." https://developer.apple.com/videos/play/wwdc2022/10092/.

171. The 289 Patent defines a PDK as including "an antenna and a transceiver for communicating with an RDC (not shown) and a controller and memory for storing information particular to a user". 289 Patent, 14: 19-23. A component of the integrated PDK of the hybrid device, accordingly, is "a controller and memory for storing information particular to a user". The 289 Patent defines the operation of the controller and memory for storing information particular to the user as entailing the receipt of an access key from an external application that is used to access a specific service block. 289 Patent, 6:57-61 ("Regardless of how created, once created, external applications (such as applications 120 in FIG. 1) can gain access to specific service block 112 by providing the corresponding access key 118. In FIG. 2., this is shown conceptually by control logic 250.") The integrated PDK of the hybrid device, therefore, includes a controller placing within memory information that can only be accessed by a corresponding access key provided by an external application. Such memory is present in Apple Passkey-compatible Devices because it is required by the standard. "Passkeys are built on the WebAuthentication -- or WebAuthn standard -- and use public-key cryptography." W3C Specification.

172. Under the WebAuthn specification, "compliant authenticators protect public key credentials." W3C Specification, § 1. A public key credential refers to a public key credential source, which includes a credential ID. *Id*., § 4 (defining "public key credential" and "public key credential source"). The credential ID uniquely identifies its public key credential source. *Id*., § 4 (defining a credential ID as a "probabilistically-unique byte sequence identifying a public key credential source and its authentication assertions"). In addition to the credential ID, each public key credential source contains a "credential private key." *Id*., § 4, (defining "public key credential source" as a data structure including the credential private key and the credential ID). "The credential private key is bound to a particular authenticator" and part of an asymmetric key pair containing a public key returned to a relying party. *Id*., § 4 (Defining a "credential key pair".). To comply with the WebAuthn standard, Apple Passkey-compatible Devices must therefore store within memory a credential comprising a private key of an asymmetric key pair and a credential ID uniquely identifying the private/public key pair to which the private key belongs.

173. The credentials stored within Apple Passkey-compatible Devices can only be accessed with the appropriate access key. "A public key credential can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." W3C Specification, § 4 ("A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with."). When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external server. Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2.2, https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-errata-20220621.html ("7.1 If the allowList parameter is present and is non-empty, locate all denoted credentials created by this authenticator and bound to the specified rpId. 7.2 If an allowList

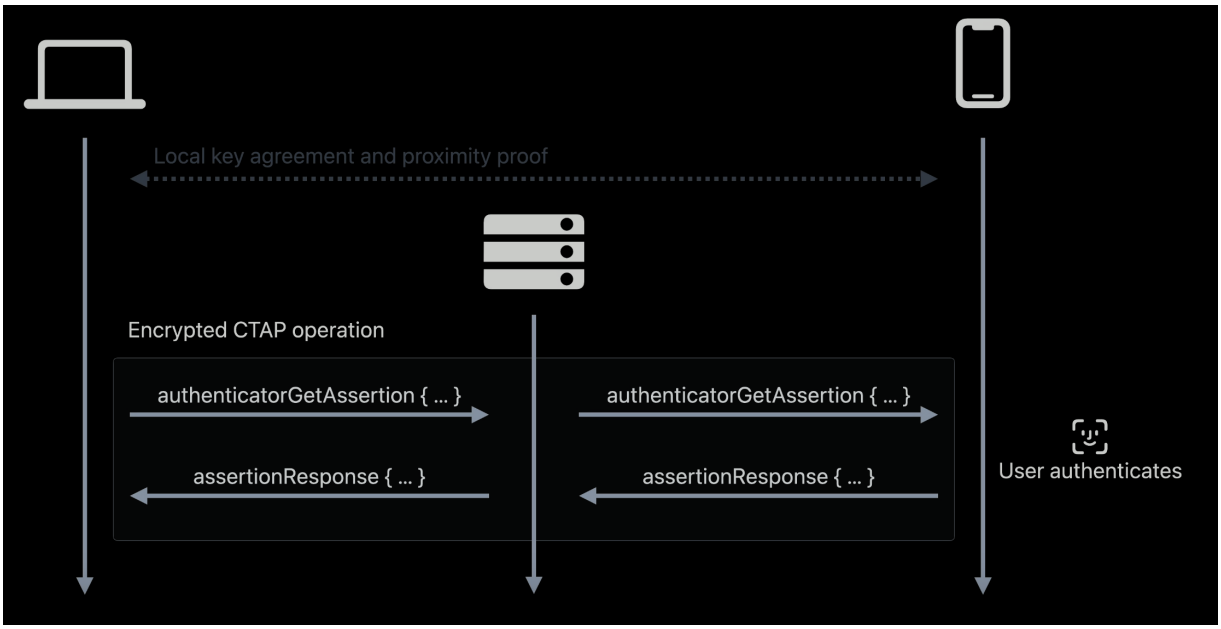
is not present, locate all discoverable credentials that are created by this authenticator and bound to the specified rpId."). As only credentials corresponding to the RP ID will be retrieved, the RP ID is an access key. Apple states: "And the system will take care of only letting me use it in the correct app or website, with strong built-in phishing resistance." https://developer.apple.com/videos/play/wwdc2022/10092/.

174. As Apple Passkey-compatible Devices will only return credentials corresponding to the RP ID access key provided by the external relying party, these devices must have the controller and memory necessary for a minimal embodiment of a PDK.

175. In addition to the controller and memory, a minimal embodiment of PDK is defined by the 289 Patent as including "an antenna and a transceiver for communication with an RDC..." 289 Patent, 14: 19-23. With Apple Passkey-compatible Devices:

> Passkeys can also be used to sign in across devices in a secure, phishing-resistant manner. Here's how that works. There are two devices here. The client, which is the device or web browser where I'm signing in, and the authenticator, which is the device which has my passkey. First, the client shows a QR code, which the authenticator scans. This QR code contains a URL that encodes a pair of single-use encryption keys. Then, the authenticator produces a Bluetooth advertisement containing routing information for a network relay server. Once the local exchange and key agreement have happened, the two devices connect to a relay server picked by the phone. From there, they perform a standard FIDO CTAP operation, which is encrypted using the keys from earlier, so the relay server can't see anything that's going on. This whole process is performed by the device and the web browser. The website is not involved at any point in the cross-device communication."

https://developer.apple.com/videos/play/wwdc2022/10092/. The process is shown in the below figure:



Id.

176. Establishing a connection over the internet requires utilizing either the device's Wi-Fi or cellular capabilities – both of which are wireless protocols. Accordingly, Apple Passkey-compatible Devices must have the antenna and transceiver necessary to implement the wireless protocols enabling transmission over the internet.

177. Apple Passkey-compatible Devices have each of the elements of a minimal embodiment of a PDK; these devices include an integrated PDK.

178. Apple Passkey-compatible Devices also store local, secure biometric information for authenticating a user.

179. "Passkeys are a replacement for passwords. They are faster and easier to sign in. Just use Touch ID or Face ID to authenticate and you're done." https://developer.apple.com/videos/play/wwdc2023/10263/. "Apple platforms will always require UV for passkeys when biometrics are available, so you don't have to worry about that." https://developer.apple.com/videos/play/wwdc2022/10092/. Accordingly, Apple Passkey-compatible Devices locally store biometric information for authenticating user.

180. Apple Passkey-compatible Devices utilize the Secure Enclave to securely store the biometric information for authentication. "During enrollment, the Secure Enclave processes, encrypts, and stores the corresponding Touch ID and Face ID template data." Apple Platform Security, page 19.

181. "The Secure Enclave is a dedicated secure subsystem … isolated from the main processor to provide an extra layer of security designed to keep sensitive data secure even when the Application Processor kernel becomes compromised." Apple Platform Security, page 9. The biometric data for Touch ID and Face ID, which are protected even when a hack or malware comprises the Application Processor, are securely stored.

182. Apple Passkey-compatible Devices therefore locally store secured biometric data for authenticating a user.

183. As detailed above, Apple Passkey-compatible Devices enable the use of passkeys to sign in across devices in a secure, phishing-resistant manner by utilizing either the device's Wi-Fi or cellular capabilities – both of which are wireless protocols. Consequently, Apple Passkey-compatible Devices are capable of communicating wirelessly with an external receiver decoder circuit.

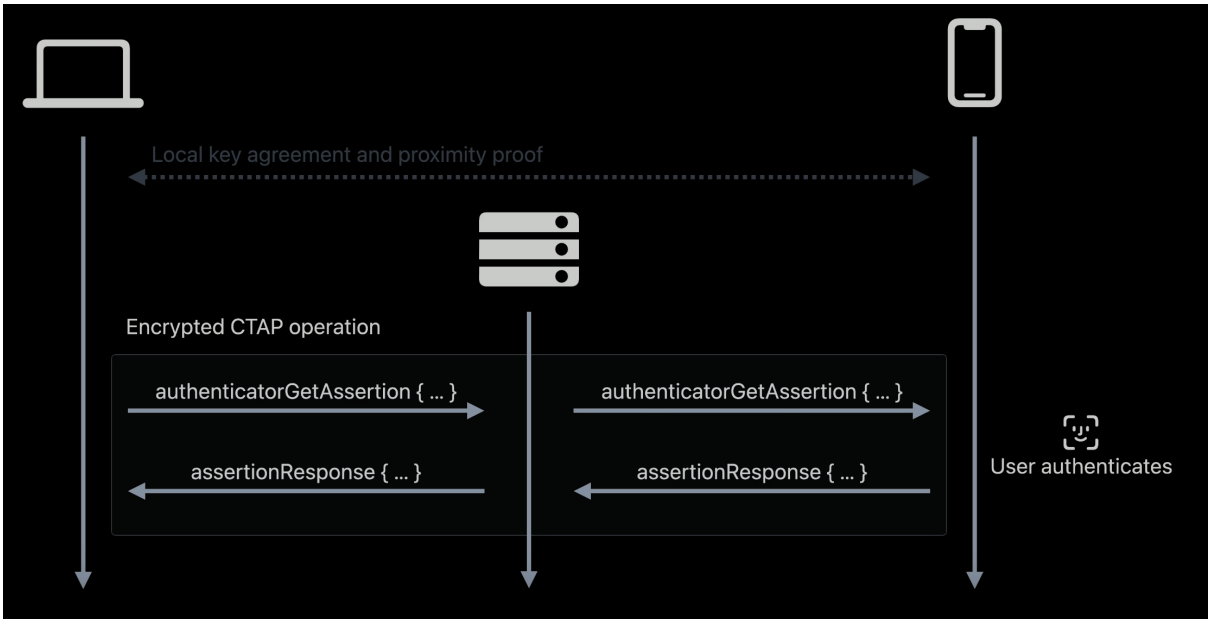
184. As detailed above, Apple Passkey-compatible Devices communicate passkey signatures over an encrypted connection, through the internet, via Wi-Fi, and/or cellular protocols. Accordingly, Apple Passkey-compatible Devices include an RDC enabling wireless communications with at least one external device, such as a Mac computer running macOS 13 (or later).

185. Enabling the use of passkeys across devices with a QR code, Apple Passkey-compatible Devices include a signal line for communication that couples the integrated RDC to

the integrated PDK. As noted above, using a passkey on Apple Passkey-compatible Devices to sign into a website on external device begins by scanning a QR code:

> First, the client shows a QR code, which the authenticator scans. This QR code contains a URL that encodes a pair of single-use encryption keys. Then, the authenticator produces a Bluetooth advertisement containing routing information for a network relay server. Once the local exchange and key agreement have happened, the two devices connect to a relay server picked by the phone. From there, they perform a standard FIDO CTAP operation, which is encrypted using the keys from earlier, so the relay server can't see anything that's going on. This whole process is performed by the device and the web browser. The website is not involved at any point in the cross-device communication.

https://developer.apple.com/videos/play/wwdc2022/10092. The process is shown in the below figure.



186. As shown above, an "authenticatorGetAssertion" is forward to the phone, which is request to provide cryptographic proof of user authentication. Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2, https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-errata-20220621.html (defining authenticatorGetAssertion as the method "used by a host to request cryptographic proof of user authentication as well as user consent to a given transaction, using a previously generated

credential that is bound to the authenticator and relying party identifier"). The authenticatorGetAssertion request contains a relying party identifier (RP ID) access key. *Id*., § 6.2 (defining the input parameters of the authenticatorGetAssertion as including a required relying party identifier). A passkey, however, can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." W3C Specification, § 4 ("A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with."). As such, the PDK on the Apple Passkey-compatible Device must receive the RP ID access key, unlock the necessary passkey to generate cryptographic proof, and send the proof back to the external device via the relay server. The PDK within the Apple Passkey-compatible Device, however, could only do so if the integrated RDC receiving the authenticatorGetAssertion and returning the cryptographic proof was communicatively coupled to the PDK. Apple Passkey-compatible Devices, therefore, necessarily have an integrated RDC coupled to the integrated PDK by a first signal line for communication.

187. Apple Passkey-compatible Devices necessarily have a second signal line coupling the RDC to at least one other component. To function, the RDC must receive power from a battery and/or be coupled to at least one application processor or similar processing unit and/or one or more antenna. The 289 Patent states in Col. 14, ll. 53-55, "the cell phone components and a battery 2004 are coupled to the RDC 304a by signal line 1106."

188. When using passkeys to sign in across devices, the PDK of an Apple Passkey-compatible Devices enables an authentication service.

189. Authentication is a service provided by the relying party, and the credential ID is necessary for the relying party to perform the authentication function. Upon receiving the response, the relying party will use the credential ID to locate the appropriate public key to verify a signature

generated with the private key held by the authenticator. W3C Specification, § 7.2 ("7. Using credential.id (or credential.rawId, if base64url encoding is inappropriate for your use case), look up the corresponding credential public key and let credentialPublicKey be that credential public key… 20. Using credentialPublicKey, verify that sig is a valid signature over the binary concatenation of authData and hash… 22. If all the above steps are successful, continue with the authentication ceremony as appropriate. Otherwise, fail the authentication ceremony.") "[I]f an authenticator returns the wrong credential ID, or if an attacker intercepts and manipulates the credential ID, is that the WebAuthn Relying Party would not look up the correct credential public key with which to verify the returned signed authenticator data (a.k.a., assertion), and thus the interaction would end in an error." W3C Specification, § 13.1. The proper credential ID is needed for the relying party to authenticate a user, and the credential ID held within the PDK of the Apple Passkey-compatible Device is included within a response to the authenticatorGetAssertion request generated by the iPhone, so the PDK of the Apple Passkey-compatible Device is enabling authentication service / function by a relying party. The PDK of the Apple Passkey-compatible Devices, accordingly, enables one or more of an application, a function, and a service.

190. By way of another example, with respect to claim 1 of the 289 Patent, the PDK of an Apple Passkey-compatible Device enables an authentication service by a relying party that is external to the iPhone. The relying party must be communicatively coupled to the RDC of the external device to receive the Apple Passkey-compatible Device response during the authentication ceremony. Accordingly, the application, the function, and the service are enabled at least in part on a device external to the hybrid device and communicatively coupled to the external RDC.

191. Intentionally left blank.

192. Intentionally left blank.

193. Intentionally left blank.

194. Apple received constructive notice of the 289 Patent on or around July 25, 2016 when Proxense sent Apple correspondence attaching a copy of U.S. Patent 9,049,188 Patent, which is in the same patent family. See **Exhibit 39**. The same correspondence also attached examples from which Apple had a basis to be aware of its infringing conduct. Apple performed and continues to perform the acts that constitute willful direct infringement, with knowledge or willful blindness that the acts would constitute direct infringement of the 289 Patent.

195. Proxense has been injured and seeks damages to adequately compensate it for Apple's infringement of the 289 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

196. Upon information and belief, Defendant will continue to infringe the 289 Patent unless permanently enjoined by this Court. Pursuant to 35 U.S.C. § 283, Proxense is entitled to a permanent injunction against further infringement of the 289 Patent by Defendant.

CLAIM 6
(Infringement of 042 Patent)

197. Proxense repeats and realleges all preceding paragraphs, as if fully set forth herein.

198. Proxense has not licensed or otherwise authorized Apple to make, use, offer for sale, sell, or import any products that embody the inventions of the 042 Patent.

199. Apple infringes at least claims 1 and 10 of the 042 Patent in violation of 35 U.S.C. § 271 with respect to the Accused Products capable of utilizing Passkeys (i.e., the Apple Passkey-compatible Devices). Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

200. For example, Apple directly infringes at least claims 1 and 10 of the 042 Patent by making, using (*e.g.*, performing/executing), selling, and/or offering to sell within the United States the Apple Passkey-compatible Devices. Apple's Passkey-compatible Devices also perform/execute the method of claim 10 and those products provide the hybrid device of claim 1.

201. As described *supra*, Apple's Passkey-compatible Devices include an integrated personal digital key and an integrated receiver decoder circuit. ("In macOS Monterey and iOS 15, we announced a developer preview of the solution -- passkeys -- and got so much great feedback. In macOS Ventura and iOS 16, we're excited to make passkeys available to everyone. Now is the time to adopt them." https://developer.apple.com/videos/play/wwdc2022/10092/.

202. The 042 Patent defines a PDK as including "an antenna and a transceiver for communicating with an RDC (not shown) and a controller and memory for storing information particular to a user." 042 Patent, 13: 46-49. A component of the integrated PDK of the hybrid device, accordingly, is "a controller and memory for storing information particular to a user." The 042 defines the operation of the controller and memory for storing information particular to the user as entailing the receipt of an access key from an external application that is used to access a specific service block. 042 Patent, 6:23-27 ("Regardless of how created, once created, external applications (such as applications 120 in FIG. 1) can gain access to specific service block 112 by providing the corresponding access key 118. In FIG. 2, this is shown conceptually by control logic 250.") The integrated PDK of the hybrid device, therefore, includes a controller placing within memory information that can only be accessed by a corresponding access key provided by an external application. Such memory is present in Apple Passkey-compatible Devices because it is required by the standard. "Passkeys are built on the WebAuthentication -- or WebAuthn standard -- and use public-key cryptography." W3C Specification.

203. Under the WebAuthn specification, "compliant authenticators protect public key credentials." W3C Specification, § 1. A public key credential refers to a public key credential source, which includes a credential ID. *Id*., § 4 (defining "public key credential" and "public key credential source"). The credential ID uniquely identifies its public key credential source. *Id*., § 4 (defining a credential ID as a "probabilistically-unique byte sequence identifying a public key credential source and its authentication assertions"). In addition to the credential ID, each public key credential source contains a "credential private key." *Id*., § 4, (defining "public key credential source" as a data structure including the credential private key and the credential ID). "The credential private key is bound to a particular authenticator" and part of an asymmetric key pair containing a public key returned to a relying party. *Id*., § 4 (Defining a "credential key pair".). To comply with the WebAuthn standard, Apple Passkey-compatible Devices must therefore store within memory a credential comprising a private key of an asymmetric key pair and a credential ID uniquely identifying the private/public key pair to which the private key belongs.

204. The credentials stored within Apple Passkey-compatible Devices can only be accessed with the appropriate access key. "A public key credential can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." W3C Specification, § 4 ("A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with."). When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external server. Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2.2, https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-errata-20220621.html ("7.1 If the allowList parameter is present and is non-empty, locate all denoted credentials created by this authenticator and bound to the specified rpId. 7.2 If an allowList

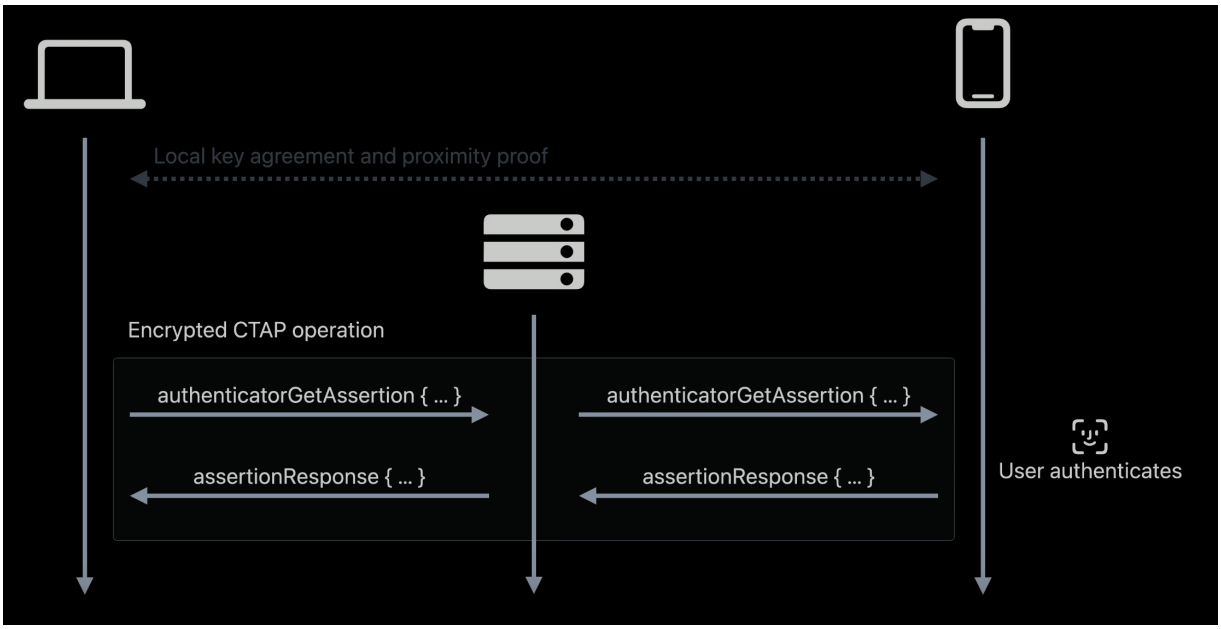
is not present, locate all discoverable credentials that are created by this authenticator and bound to the specified rpId."). As only credentials corresponding to the RP ID will be retrieved, the RP ID is an access key. Apple states: "And the system will take care of only letting me use it in the correct app or website, with strong built-in phishing resistance." https://developer.apple.com/videos/play/wwdc2022/10092/.

205. As Apple Passkey-compatible Devices will only return credentials corresponding to the RP ID access key provided by the external relying party, these devices must have the controller and memory necessary for a minimal embodiment of a PDK.

206. In addition to the controller and memory, a minimal embodiment of PDK is defined by the 042 Patent as including "an antenna and a transceiver for communication with an RDC." 042 Patent, 13:46-48. With Apple Passkey-compatible Devices:

> Passkeys can also be used to sign in across devices in a secure, phishing-resistant manner. Here's how that works. There are two devices here. The client, which is the device or web browser where I'm signing in, and the authenticator, which is the device which has my passkey. First, the client shows a QR code, which the authenticator scans. This QR code contains a URL that encodes a pair of single-use encryption keys. Then, the authenticator produces a Bluetooth advertisement containing routing information for a network relay server. Once the local exchange and key agreement have happened, the two devices connect to a relay server picked by the phone. From there, they perform a standard FIDO CTAP operation, which is encrypted using the keys from earlier, so the relay server can't see anything that's going on. This whole process is performed by the device and the web browser. The website is not involved at any point in the cross-device communication."

https://developer.apple.com/videos/play/wwdc2022/10092/. The process is shown in the below figure:



Id.

207. Establishing a connection over the internet requires utilizing either the device's Wi-Fi or cellular capabilities – both of which are wireless protocols. Accordingly, Apple Passkey-compatible Devices must have the antenna and transceiver necessary to implement the wireless protocols enabling transmission over the internet.

208. Apple Passkey-compatible Devices have each of the elements of a minimal embodiment of a PDK; these devices include an integrated PDK.

209. As detailed above, Apple Passkey-compatible Devices enable the use of passkeys to sign in across devices in a secure, phishing-resistant manner by utilizing either the device's Wi-Fi or cellular capabilities – both of which are wireless protocols. Consequently, Apple Passkey-compatible Devices are capable of communicating wirelessly with an external receiver decoder circuit.

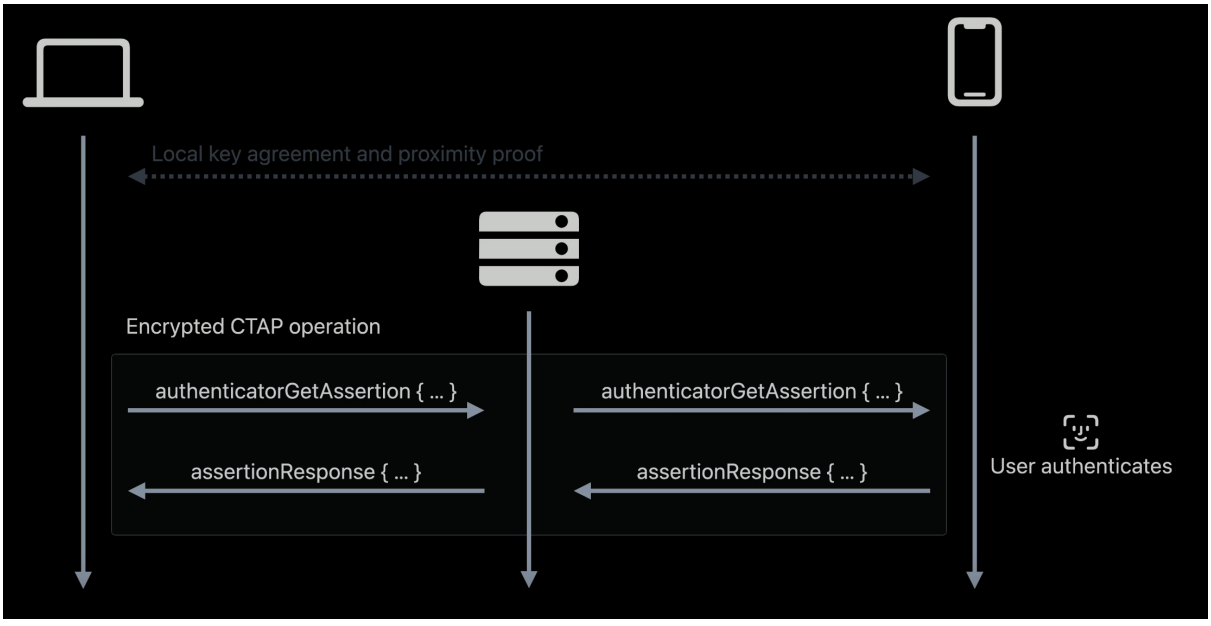
210. As detailed above, Apple Passkey-compatible Devices communicate passkey signatures over an encrypted connection, through the internet, via Wi-Fi, and/or cellular protocols. Accordingly, Apple Passkey-compatible Devices include an RDC enabling wireless

communications with at least one external device, such as a Mac computer running macOS 13 (or later).

211. Enabling the use of passkeys across devices with a QR code, Apple Passkey-compatible Devices include a signal line for communication that couples the integrated RDC to the integrated PDK. As noted above, using a passkey on Apple Passkey-compatible Devices to sign into a website on external device begins by scanning a QR code:

> First, the client shows a QR code, which the authenticator scans. This QR code contains a URL that encodes a pair of single-use encryption keys. Then, the authenticator produces a Bluetooth advertisement containing routing information for a network relay server. Once the local exchange and key agreement have happened, the two devices connect to a relay server picked by the phone. From there, they perform a standard FIDO CTAP operation, which is encrypted using the keys from earlier, so the relay server can't see anything that's going on. This whole process is performed by the device and the web browser. The website is not involved at any point in the cross-device communication.

https://developer.apple.com/videos/play/wwdc2022/10092. The process is shown in the below figure.



212. As indicated above, the process begins with a local key exchange. "This local exchange allows selecting a server and sharing routing information, but also serves two additional

functions. It performs an out-of-band key agreement that the server can't see, so everything going over the network is end-to-end encrypted and the server can't read anything. It also provides a strong claim that these two devices are in physical proximity." https://developer.apple.com/videos/play/wwdc2022/10092/. Accordingly, the integrated RDC of an Apple Passkey compatible device is communicating with an external PDK within a proximity zone.

213. As shown above, an "authenticatorGetAssertion" is forwarded to the phone, which is request to provide cryptographic proof of user authentication. Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2, https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-errata-20220621.html (defining authenticatorGetAssertion as the method "used by a host to request cryptographic proof of user authentication as well as user consent to a given transaction, using a previously generated credential that is bound to the authenticator and relying party identifier"). The authenticatorGetAssertion request contains a relying party identifier (RP ID) access key. *Id*., § 6.2 (defining the input parameters of the authenticatorGetAssertion as including a required relying party identifier). A passkey, however, can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." W3C Specification, § 4 ("A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with."). As such, the PDK on the Apple Passkey-compatible Device must receive the RP ID access key, unlock the necessary passkey to generate cryptographic proof, and send the proof back to the external device via the relay server. The PDK within the Apple Passkey-compatible Device, however, could only do so if the integrated RDC receiving the authenticatorGetAssertion and returning the cryptographic proof was communicatively coupled to the PDK. Apple Passkey-

compatible Devices, therefore, necessarily have an integrated RDC coupled to the integrated PDK by a first signal line for communication.

214. Apple Passkey-compatible Devices necessarily have a second signal line coupling the RDC to at least one other component. To function, the RDC must receive power from a battery and/or be coupled to at least one application processor or similar processing unit and/or one or more antenna. The 042 Patent states in Col. 14, ll. 12-14, "the cell phone components and a battery 2004 are coupled to the RDC 304a by signal line 1106."

215. When using passkeys to sign in across devices, the PDK of an Apple Passkey-compatible Devices enables an authentication service.

216. Authentication is a service provided by the relying party, and the credential ID is necessary for the relying party to perform the authentication function. Upon receiving the response, the relying party will use the credential ID to locate the appropriate public key to verify a signature generated with the private key held by the authenticator. W3C Specification, § 7.2 ("7. Using credential.id (or credential.rawId, if base64url encoding is inappropriate for your use case), look up the corresponding credential public key and let credentialPublicKey be that credential public key… 20. Using credentialPublicKey, verify that sig is a valid signature over the binary concatenation of authData and hash… 22. If all the above steps are successful, continue with the authentication ceremony as appropriate. Otherwise, fail the authentication ceremony.") "[I]f an authenticator returns the wrong credential ID, or if an attacker intercepts and manipulates the credential ID, is that the WebAuthn Relying Party would not look up the correct credential public key with which to verify the returned signed authenticator data (a.k.a., assertion), and thus the interaction would end in an error." W3C Specification, § 13.1. The proper credential ID is needed for the relying party to authenticate a user, and the credential ID held within the PDK of the Apple

Passkey-compatible Device is included within a response to the authenticatorGetAssertion request generated by the iPhone, so the PDK of the Apple Passkey-compatible Device is enabling authentication by relying party. The PDK of the Apple Passkey-compatible Devices, accordingly, enables one or more of an application, a function, and a service.

217. Apple has induced infringement, and continues to induce infringement, of at least claims 1 and 10 of the 042 Patent in violation of 35 U.S.C. § 271(b), by providing Apple Passkeys to Apple Passkey-compatible Devices, along with a substantial knowledge base on the features, use and integration of Passkeys, to sellers, resellers and end-user customers who authenticate via Apple Passkey-compatible Devices. For example, Apple induces infringement of at least claims 1 and 10 of the 042 Patent by making Passkeys available for use on Apple Passkey-compatible Devices. When executed by a user for its intended and advertised purpose, Apple Passkey-compatible Devices performs/executes a method in accordance with claim 10 by using inherent capabilities to produce a hybrid device infringing claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

218. Apple contributes to direct infringement of at least claims 1 and 10 of the 042 Patent by providing Apple Passkeys to Apple Passkey-compatible Devices. For example, Apple contributes to infringement of at least claims 1 and 10 of the 042 Patent by making Passkeys available for use on Apple Passkey-compatible Devices. When executed by a user for its intended and advertised purpose, Apple Passkey-compatible Devices performs/executes a method in accordance with claim 10 by using inherent capabilities to produce a hybrid device infringing claim 1. Proxense contends each limitation is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.

219. Apple received constructive notice of the 042 Patent on or around July 25, 2016 when Proxense sent Apple correspondence. See **Exhibit 39**. The same correspondence also attached examples from which Apple had a basis to be aware of its infringing conduct. Apple performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness that the acts would constitute direct and/or indirect infringement of the 042 Patent.

220. Proxense has been injured and seeks damages to adequately compensate it for Apple's infringement of the 042 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

221. Upon information and belief, Defendant will continue to infringe the 042 Patent unless permanently enjoined by this Court. Pursuant to 35 U.S.C. § 283, Proxense is entitled to a permanent injunction against further infringement of the 042 Patent by Defendant.

DEMAND FOR JURY TRIAL

Plaintiff hereby requests a jury trial of all issues so triable.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for relief against Defendants as follows:

a. Entry of judgment declaring that Defendant infringes one or more claims of each of the Patents-in-Suit;

b. Entry of judgment declaring that Defendant's infringement of the Patents-in-Suit is willful;

c. An order awarding damages sufficient to compensate Plaintiff for Defendants' infringement of the Patents-in-Suit, but in no event less than a reasonable royalty, including supplemental damages post-verdict, together with pre-judgment and post-judgment interest and costs;

d. Enhanced damages pursuant to 35 U.S.C. § 284;

e. Entry of judgment declaring that this case is exceptional and awarding Plaintiff its costs and reasonable attorney fees pursuant to 35 U.S.C. § 285;

f. An accounting for acts of infringement;

g. Such other equitable relief which may be requested and to which the Plaintiff is entitled; and

h. Such other and further relief as the Court deems just and proper.

Dated: October 28, 2024

Respectfully Submitted,

/s/ David L. Hecht
David L. Hecht (Co-Lead Counsel)
dhecht@hechtpartners.com
Maxim Price (*pro hac vice* pending)
mprice@hechtpartners.com
Yi Wen Wu (*pro hac vice* pending)
wwu@hechtpartners.com
HECHT PARTNERS LLP
125 Park Avenue, 25th Floor
New York, New York 10017
Telephone: (212) 851-6821

Brian D. Melton (Co-Lead Counsel)
bmelton@susmangodfrey.com
Geoffrey L. Harrison
gharrison@susmangodfrey.com
Meng Xi
mxi@susmangodfrey.com
SUSMAN GODFREY L.L.P.
1000 Louisiana Street, Suite 5100
Houston, Texas 77002-5096
Telephone: (713) 653-7807
Facsimile: (713) 654-6666

Lear Jiang
ljiang@susmangodfrey.com
SUSMAN GODFREY L.L.P.
1900 Avenue of the Stars, Suite 1400
Los Angeles, CA 90067-6029
Telephone: (310) 789-3100
Facsimile: (3103) 789-3150

Counsel for Plaintiff Proxense, LLC