UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS

PROXENSE, LLC, *Plaintiffs*, v. GOOGLE LLC and GOOGLE PAYMENT CORP. *Defendants*.	Civil Action No. 6:23-cv-320 **JURY TRIAL REQUESTED**

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Proxense, LLC ("Proxense" or "Plaintiff") hereby sets forth its Complaint for patent infringement against Defendants Google LLC and Google Payment Corp. ("Google" or "Defendant"), and states 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 Google 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.

3. On information and belief, Google LLC is a Delaware corporation with a physical address of 500 West Second Street, Austin, Texas, 78601. On information and belief, Google LLC

also maintains a regular and established place of business at 110 East Houston Street, #300, San Antonio, Texas 78205. Google LLC is registered to do business in the State of Texas and has been registered since 2006. Google LLC may be served through its registered agent, the Corporation Service Company d/b/a CSC – Lawyers Incorporating Service Company, at 211 East Seventh Street, Suite 620, Austin, Texas, 78701 - 3218.

4. Google Payment Corp. is a Delaware corporation and maintains its principal place of business at 1600 Amphitheatre Parkway, Mountain View, California 94043. Google Payment Corp. is registered to do business in the State of Texas and has been registered since 2005. Google Payment Corp. may be served with process through its registered agent, Corporation Service Company d/b/a CSC – Lawyers Incorporating Service Company, at 211 East Seventh Street, Suite 620, Austin, Texas, 78701 - 3218.

JURISDICTION AND VENUE

5. This Court has exclusive subject matter jurisdiction over this case pursuant to 28 U.S.C. §§ 1331 and 1338(a) 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.

6. This Court has personal jurisdiction over Google LLC and Google Payment Corp. (collectively, "Google") because Google is a multinational technology company that has a significant presence in the District through the products and services Google provides residents of this District. Defendants regularly conduct 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 Google would not offend traditional notions of fair play and substantial justice. Google 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. Google 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.

7. On information and belief, Google has authorized retailers in this Judicial District that offer and sell products on its behalf in this District, including products accused of infringement herein. On information and belief, these include Best Buy, *e.g.*, at 4627 S. Jack Kultgen Expy., Waco, TX 76706; Target, e.g., at 5401 Bosque Blvd., Waco, TX 76710; T-Mobile, *e.g.*, 100 N New Rd Ste 110, Waco, TX 76710 and 2448 West Loop 340 Suite 24a, TX 76711; and Verizon, *e.g.*, 1820 S Valley Mills Dr., Waco, TX 76711 and 2812 W Loop 340 Waco, TX, 76711.

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

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

10. Venue is proper in this Judicial District pursuant to 28 U.S.C. § 1400(b). Google is registered to do business in Texas and, upon information and belief, Google has transacted business in the Western District of Texas and has a regular and established places of business in this Judicial District.

11. Joinder of Google LLC and Google Payment Corp. is proper because they are related entities that are either jointly and severally liable for infringement, or that make, use, sell, offer to sell, and/or import the same or similar products accused of infringement herein. Further, upon information and belief, Google LLC and Google Payment Corp. act in concert to provide the

software, hardware, infrastructure, and payment networks used by the accused products. Therefore, the factual question of infringement will substantially overlap between Google LLC and Google Payment Corp. Additionally, Proxense anticipates there will be substantial overlap with respect to discovery.

12. Google also maintains a significant physical presence in this Judicial District and employs many people in this Judicial District. According to CultureMap Austin, Google already leases more than 550,000 square feet at three locations in and around downtown Austin: 100 Congress Ave., 901 E. Fifth St., and 500 W. Second St. *See* John Egan, "Austin's next iconic high-rise sails towards opening date with Google as anchor tenant," CultureMap Austin (March 19, 2021) *See* **Exhibit 7** In addition to its current offices, including its Austin headquarters at 500 W 2nd Street in Austin, Google has leased an entire building down the street at 601 West 2nd Street that is being built out. Last year, Google announced that it plans to occupy the 37-story building, which will be Austin's tallest office tower and contains 814,081 square feet. *Id*.

13. Google currently employs approximately 1,100 people in Austin. Among these employees are people specifically responsible for the infringing technology in this lawsuit. For example, Nik Bhattacharya, a Senior Software Engineering Manager, who works for Google in Austin, Texas, describes on his LinkedIn profile that he is "**[l]eading** the FIDO passkeys effort at Google." *See* **Exhibit 8**. Aspects of Google's particular implementation of FIDO are among the grounds for the claims of patent infringement asserted here.

14. On Google's Careers website, careers.google.com, as of December 15, 2022, Google lists 209 open jobs for its Austin, Texas location. Many of its jobs list multiple potential locations for the job applicant to work from, because Google employees are able to work collaboratively from essentially anywhere, on the "cloud," because all relevant information,

including on information and belief documents relevant to this litigation, including documents and code are decentralized and exist on Google's intranet that is available from any of its physical office locations, or even remotely. By way of a non-limiting example, as of December 15, 2022, Google's Careers website had a listing for a Senior Product Manager, Core Privacy, Safety, and Security. The position listed potential in-office locations as Seattle, Washington; Austin, Texas; and New York City, New York. The position's responsibilities included: "[l]aunch[ing] new products and features…"; "[w]ork[ing] collaboratively with engineering, marketing, legal, UX, and other teams on cutting edge technologies"; and "[d]evelop[ing] solutions to problems by collaborating as needed across regions, product areas, and functions."

15. On information and belief, Google has also committed acts of direct and indirect infringement in the Western District of Texas. For example, Google sells its Pixel line of phones to individuals in this Judicial District, which ship with various versions of the Android operating system ("Android OS" or "Android"). Google also distributes Google Pay, also known as "G Pay," "Pay with Google" and "Android Pay" which it describes as a "simple, secure way to pay and save," in this Judicial District, both on its Pixel phones and otherwise. *See* **Exhibit 9**.

PATENTS-IN-SUIT

16. 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 1**.

17. 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 2**.

18. On March 26, 2016, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 9,298,905 (the "905 Patent") entitled "Biometric Personal Data Key (PDK) Authentication." A true and correct copy of the 905 Patent is attached hereto as **Exhibit 3**.

19. 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 4**.

20. 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**.

21. On September 11, 2018, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 10,073,960 (the "960 Patent") entitled "Hybrid Device Having a Personal Digital Key and Receiver-Decoder Circuit and Methods of Use." A true and correct copy of the 960 Patent is attached hereto as **Exhibit 6**.

22. 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, 905, 042, 289, and 960 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.

23. The technologies of the Patents-in-Suit were invented by John Giobbi. The 730 and 905 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 042, 289, and 960 Patents generally covers systems, devices, and methods of utilizing personal digital keys for verifying a user in order to enable applications, functions, or services.

FACTUAL ALLEGATIONS

I. TECHNOLOGY BACKGROUND

24. 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.

25. One such architecture is "federated authentication" (also known as "federated identity"), which relies on an external trusted system to authenticate users. In a federated authentication solution, the system being accessed must request authentication of the user from the external system that is used to authenticate users. The external system authenticating the user 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.

26. While OpenID connect limited the use of passwords, it did not eliminate them. Authentication protocols geared towards eliminating passwords include WebAuthn and its derivative, FIDO2, an open authentication standard developed by the FIDO alliance. WebAuthn, 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 server – 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 server) 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 issues an authentication response.

27. WebAuthn and federated protocols can be combined. When combined, the system to be accessed by the user requests authentication by a WebAuthn / FIDO 2 server. 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 connect of successful authentication. The OpenID connect server then sends a security token to be used to access the system requesting authentication.

28. Attempting to eliminate the user of passwords, Google has developed a universal platform "passwordless" architecture. The architecture is universal in that it works across platforms, such as iOS, Android, and Windows. It is password-less in that passwords have been

replaced with the use of authenticators incorporated into Android OS 9 and higher. Incorporating OpenID Connect, Google's architecture relies on the issuance of security tokens. The hub of Google's universal platform password-less architecture is Google Identity, which receives authentication requests from external systems, coordinates the action of authenticators, and issues security tokens.

29. Federated authentication is not the only authentication architecture Google has incorporated into the Android OS. Google has incorporated EMV's payment tokenization architecture. As with OpenID Connect, EMV's payment tokenization 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 Connect tokens, EMV payment tokens indicate a previous authentication of the user. The tokens differ with respect to authorization claims. OpenID 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 payment from a device. Unlike OpenID Connect 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 connect tokens and EMV tokens are indicative of user authentication and authorization.

30. As to facilitate the use of EMV's payment tokenization architecture, Google has deployed technology to secure payment tokens in connection with its Android OS, Google Pay (initially and subsequently called Google Wallet, collectively "Google Pay" unless otherwise specified). Using biometric security features (i.e., the ability to unlock mobile devices through a fingerprint or facial recognition) available in Android, Google offered biometric authentication of the Google Pay app, in 2019. *See* **Exhibit 10**. Previously, the only Google Pay authentication option was entering a PIN number, which is guessable and "crackable," as opposed to more secure and user-friendly biometrics.

31. Incorporating authenticators (including Google's Authenticator) into the Android OS is, on information and belief, necessary for Google Cloud Services to remain competitive with Microsoft's cloud services, which supports the Windows 10/11 integrated authenticator, Windows Hello, and the iOS/Android authenticator, Microsoft Authenticator. The worldwide public cloud services market had revenues in 2021 totaling $408.6 billion. Likewise, the ability to secure Google Pay transactions with biometrics was, on information and belief, important for Google to remain competitive with Apple Pay and other payment platforms that had implemented the feature. Global contactless transaction values were estimated at $2 trillion in 2020. By 2024, they may reach $6 trillion according to forecasts. Mobile payments, or transactions initiated on mobile devices such as cell phones or tablet computers, have become increasingly popular with the increasing use of applications like Google Pay (also known as Google Wallet), launched in 2011.

II. PROXENSE AND ITS INNOVATIVE TECHNOLOGIES

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

33. 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.

34. 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

[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.

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.

35. 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 11**. 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 12**.

36. 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 13**. 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…"

37. It would be years until products like Google Wallet (2011), Apple Pay (2014), and Samsung Pay (2015) were launched and became mainstream; Apple's TouchID, which involves fingerprint recognition technology, for example, was introduced in 2013. It would take Google

until 2019 to enable biometric authentication for Android 10 phones and Google Pay. Accordingly, Proxense's technology was years ahead of the industry.

38. After the launch of services like Google Wallet/Pay, and its 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 with companies like Google, even though Proxense invented the technology utilized in these solutions.

39. 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. The Accused Products

40. Through its own actions, and the actions of its customers and users, which Google directs and controls, Google has manufactured, used, marketed, sold, offered for sale, and exported from and imported into the United States a universal platform password-less architecture that directly and/or indirectly infringes (and literally or via the doctrine of equivalents) the Patents-in-Suit. Accordingly, the Accused Products include the servers, cloud resources, and software comprising Google's password-less architecture and devices including, supporting and integrating into the architecture, including devices running the Android OS, Chrome OS, and/or Chrome browser.

41. Three primary components make up the infringing identity architecture. The first is Google Identity (also known as Google Identity Services) which coordinates the actions of the other components by authenticating users and issuing various bearer tokens. See e.g., **Exhibit 14**. The second is an authenticator, like Google Authenticator, permitting user verification by Google

Identity. During verification, Google Identity issues commands to the authenticators (called "requests"). The operation of the authenticators, accordingly, is controlled by Google. The third component of the system is a resource, such as an application, website, or subscription, requesting authentication of the user. During user login, the resource sends requests to a URL provided by Google and in a form dictated by Google. The resource then listens for a reply at a callback URL the resource registered with Google. As with the requests, the callbacks are in the form dictated by Google. The resources are hosted by device and/or server separate from Google Identity. Accordingly, the resource and Google Identity are separate and distinct entities and the resource (not Google Identity) is the system being accessed.

42. Google utilizes Google Identity to sign users into Gmail, Google Drive, Google Pay and other services and products offered by Google. Google sells access to Google Identity to developers, websites, and corporate clients through various subscriptions. Thus, for a fee, Google Identity becomes an identity provider for applications, businesses, and websites. Selling such identity and access management (IAM) services and controlling the actions of subscribers and users, Google directly and/or indirectly infringes (and literally or under the doctrine of equivalents) the Patents-in-Suit. Furthermore, by utilizing Google Platform for its own product and services, Google directly and/or indirectly infringes (and literally or via the doctrine of equivalents) the Patents-in-Suit.

43. As noted above, Google's infringing universal platform password-less architecture includes an authenticator. One authenticator distributed by Google is a native component of Android OS 9 and higher. Accordingly, since at least August 6, 2018, if not earlier, the Android OS has enabled password-less sign-in to services and subscriptions offered by Google. *See* **Exhibit 15**. Another authenticator developed by Google is the Titan Security Key that includes

the functionality to wirelessly verify a user during authentication of an iOS compatible devices. **Exhibit 16**. The current and previous versions of the Android OS and Titan Security Key are non-limiting instances of authenticators integrated into the Accused Products.

44. Google Identity, which directs and controls the actions of the authenticator, is also an element of the Accused Products. Google Identity includes a series of APIs launched on August 3, 2021. *See* Google Developers Blog: Launching our new Google Identity Services APIs (googleblog.com), August 3, 2021, **Exhibit 17** ("Today we are launching our new family of Identity APIs called Google Identity Services."). Google operates and maintains Google Identity. When combined with authenticators, such the Android OS and Titan Security Key, consumers of Google's products and services receive the benefit of password-less biometric authentication across platforms. For instance, a user may log into and utilize Gmail or Google Pay on their Windows PC or iOS device. Third party developers can purchase identity and access management services from Google to integrate their applications with Google Identity in order to offer cross platform password-less authentication via biometrics and the use of ID and access tokens to their customers and subscribers. Developers subscribing to Google Identity must register their application or website with Google, request authentication by sending a request to a URL provided by Google and in a format dictated by Google, and listen for a callback provided by Google that contains a message generated by Google in a format controlled by Google.

45. The Accused Products also include a resource accessed following successful authentication of the authenticator by Google Identity. As noted above, the resource may be an application, website, and/or a subscription offered by Google, a subscribing business, or a subscribing developer. When a user has been authenticated via an authenticator offered by Google, various bearer tokens are returned by Google to the callback URL registered with Google Identity.

The bearer token allows access to the application, and as such is an access message. Regardless of whether the resource is an application, a subscription, or a service offered by Google or a subscribing developer or business, the resource is on a system separate and distinct from Google Identity. Distribution of bearer tokens and other such access messages by Google Identity is thus necessary for Google Identity to inform the resource that the user has been successfully authenticated via an authenticator distributed by Google. As the bearer tokens are generated and distributed by Google Identity, Google controls their form and how they are distributed.

46. Current and previous versions of Google Identity are non-limiting instances of the Accused Products.

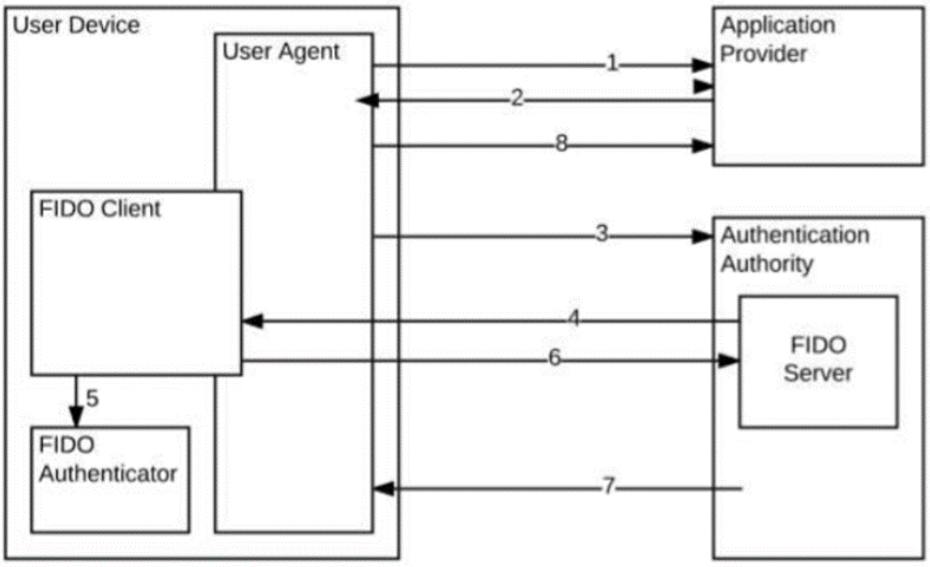
47. Another instance of the accused product is Google Wallet (also known as Google Pay), which includes functionality to biometrically verify a user during authentication of a smartphone. Google Pay is operable on a range of devices, including, at least all recent smartphones that run the Android OS, such the Google Pixel line of smartphones. Current and previous versions of Google Pay, and Google devices such as the Pixel that use Google Pay, alone and together, are further non-limiting instances of the Accused Products.

48. The Accused Products practice the claims of the Patents-in-Suit to improve security and/or the shopping experience of Google's users, and to improve Google's position in the market.

49. Google directly infringes the Patents-in-Suit through the operation of the foregoing as directed and control by Google. The Accused Products practice the claims of the Patents-in-Suit to improve the user experience of Google's customers and those of subscribing developers, and to improve Google's position in the market with respect to identity and access management, operating as an identity provider, and other products, services, and subscriptions.

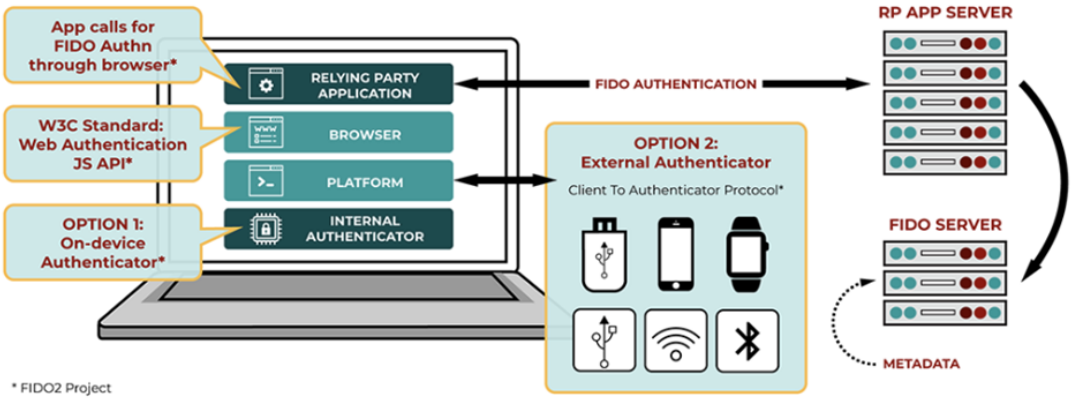
50. Google directly infringes the Patents-in-Suit by creating and utilizing a universal platform password-less architecture incorporating authenticators, including the Android OS and Titan Security Key, the Chrome browser, online Google accounts, and Google Identity, which controls the actions of authenticators and the dissemination of tokens and other such access messages, offering services, applications subscriptions and other such resources hosted by separate systems which are accessed via tokens provided by Google Identity, and offering businesses and developers identity and access management services which entail requesting user authentication and receiving tokens in a manner directed and controlled by Google.

51. Google's password-less architecture verifies a user during authentication of an integrated device – *i.e.*, a phone with Android OS 9 or higher or Titan Security Key - when a user wants to sign into an application or service developed by Google or a business / developer subscribing to Google Identity. **Exhibit 18** and **Exhibit 19**. Google's universal platform password-less architecture is built upon the FIDO and OpenID connect protocols. OpenID Connect and FIDO are two separate standards that can be combined. *See* Enterprise Adoption Best Practices Integrating FIDO & Federation Protocols, December 2017, **Exhibit 20**. When combined, the login procedure resembles that shown below.



Id., page 10

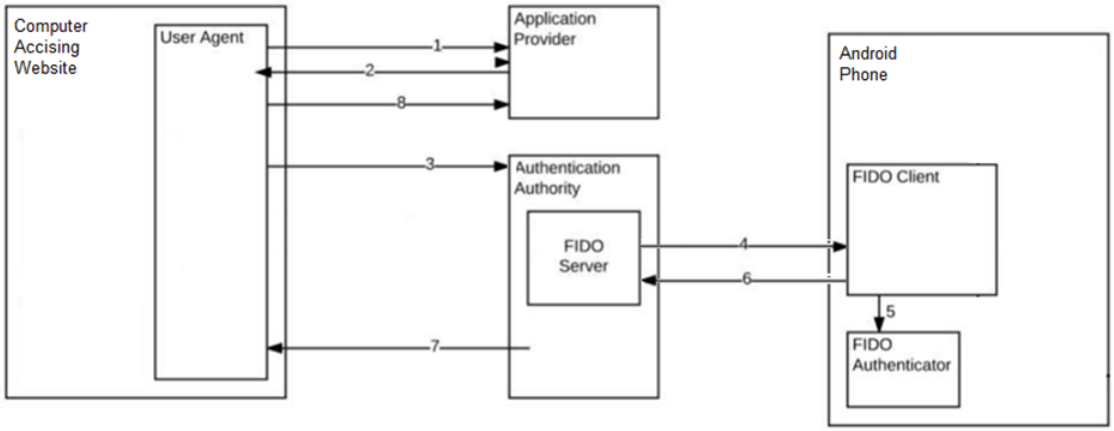
52. The figure above presents a scenario where the user is authenticated using the same device they are using to access a website or other application. This is not always the case, as the below figure shows.



See **Exhibit 21**

53. As shown above, an "External Authenticator" can be used, such as a user's Android phone. Google's "User Your Phone to Sign In" and passkey allows Google account holders to sign into application, websites and services by tapping on their phone. *See* Sign in with Google prompts - Android - Google Account Help., **Exhibit 22** ("When you sign in to your

Google Account, you can tap a notification on your phone to confirm it's you."); and **Exhibit 23** ("Let's say a user has an Android device and created a passkey on a website via Chrome. The passkey is saved and synced among Android devices, but not other ecosystems. When the user tries to sign in to the same website on macOS 13 Safari, there are no passkeys saved on the Mac. The user can still use the Android device to sign in by selecting to use a passkey from a second device. Safari shows a QR code that the user can scan using the Android phone, select the passkey and verify with their screen lock."). Accordingly, a process that may begin on a traditional computer may continue on a user's Android phone. The following diagram modifies the first figure above to reflect the second figure:



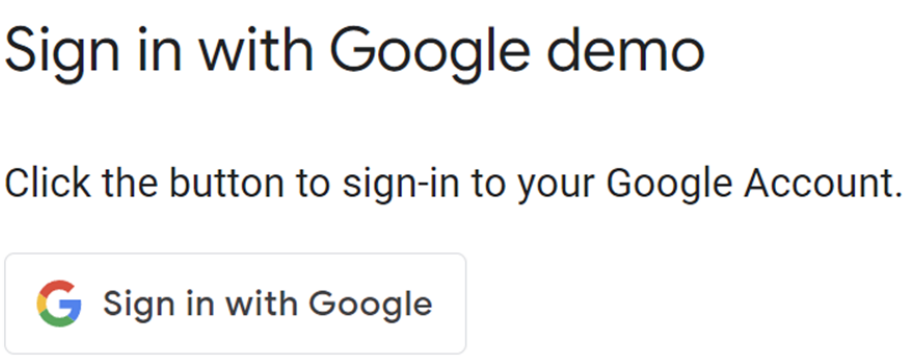
54. The steps shown in the figure above are implemented by Google Identity, the user's Android phone, the user's browser to relay messages from Google Identity to the phone, and the website, application, or service being accessed.

55. Step 1 begins when the user attempts to access a website, application, or service using either "Sign-in with Google" or a passkey. *See* **Exhibit 24** ("Sign in with Google helps [developers] to quickly and easily manage user authentication on [their] website. Users sign into a Google Account, provide their consent, and securely share their profile with [the developer's]

platform."); and **Exhibit 18** ("Passkeys are a safer and easier replacement for passwords. With passkeys, users can sign in to apps and websites with a biometric sensor (such as a fingerprint or facial recognition), PIN, or pattern, freeing them from having to remember and manage passwords.").

56. To receive the benefit of quickly and easily managing user authentication utilizing "Sign-in with Google", the website must display a "Sign in with Google" button, which is generated by the Google Identity Services JavaScript library, or provide the user the option to sign in with a passkey. **Exhibit 25** ("Put it another way, the Sign in with Google button must be generated by the Google Identity Services JavaScript library now. The button rendering API allows you to customize the color, shape, text, and size to meet the branding requirements of your website, whereas still stick to Google's guidelines."); and **Exhibit 23.**

57. A functional demonstration of the "Sign in with Google" button is provided in Google Identity documentation, as shown below.

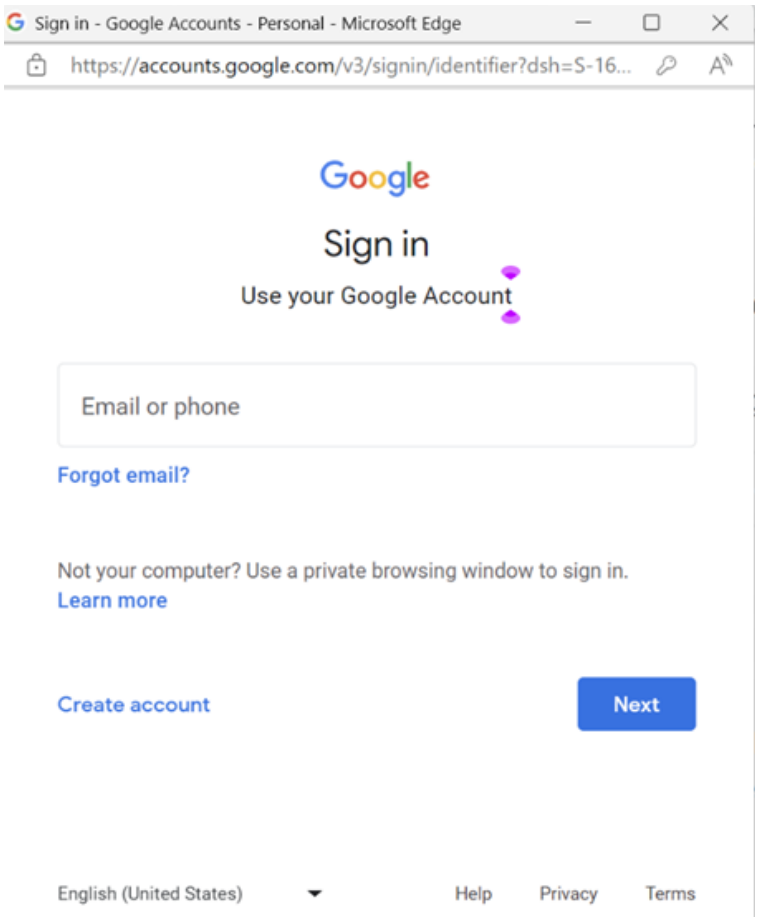


See **Exhibit 24**.

58. Alternatively, a user may be provided the convenience of biometric login without passkeys through of use of Passkeys. **Exhibit 18**. ("Passkeys are a safer and easier replacement for passwords. With passkeys, users can sign in to apps and websites with a biometric sensor

(such as a fingerprint or facial recognition), PIN, or pattern, freeing them from having to remember and manage passwords.") **Exhibit 23.**

59. Step 2 in the figure above involves the presentation of this "Sign in with Google" button to the user. When this button is clicked, the dialog box pictured below is presented to the user.



60. Accordingly, the process has moved from step 2 to step 3, where the website has redirected the browser to Google Identity's FIDO enabled servers and the user has accessed them. The user enters an email and then clicks "next". If the user has activated "Use your phone to sign in" in their Google Account, the following dialog box is presented directing the user to complete the sign on with their phone, thereby moving process along to step 4, where Google Identity servers send a challenge to the FIDO client on the user's Android phone.

61. Google Identity servers then "challenges the user to login with a previously registered device." *See* **Exhibit 26.** The previously registered device may be the user's Android smartphone, such as a Google Pixel device. Since the process is based on the presence of the Android OS, it is the same for any Android phone, including any Pixel model phones or Android phones from other Original Equipment Manufacturers ("OEM"). To register a device as a legitimate, verified device, the user simply signs into a Google account with the Android phone. *See* **Exhibit 22**, ("You'll get Google prompts on any Android phone signed in to your Google Account."). *See also* **Exhibit 27**.

62. Alternatively, if the user opted to sign in passkey, a QR is presented to the user to scan with their Android phone. *See* **Exhibit 23**. Scanning the codes causes the devices to connect. *Id.*

63. Google maintains a list of legitimate devices for each user. The FIDO standard utilized by Google Identity Service incorporates the WebAuthn standard. *See* **Exhibit 28** ("Today, WebAuthn is part of the FIDO Alliance's FIDO2 specifications and the FIDO Alliance runs certificate programs to ensure compliance."). One of the use cases for WebAuthn is decommissioning lost, stolen, and disgarded devices such that "assertions signed by this credential are rejected." *See* Web Authentication: An API for accessing Public Key Credentials - Level 3, Sec. 1.3.5 Decommissioning, **Exhibit 29**. A user may decommission devices they no longer use by signing out of them via their Google Account. *See* **Exhibit 30** (Instructing users how to secure lost, disgareded, or stolen devices by signing out of them.). To prevent account access from a device, and mark it as no longer legitimate, a user must sign out of the device and all session associated with the device. *Id* ("If you want to make sure there's no account access from a device, sign out of all the session with this device name."); and **Exhibit 22** ("To not get

Google Prompts on a device, sign out of your Google Account on that device."). Accordingly, by maintaining a list of devices the user has selected to remain signed in on, Google Identity maintains a list of legitimate integrated devices.

64. Next at step 4, the Chrome browser receives the "authenticatorGetAssertion" from Google Identity and forwards the request to the Titan Security Key or Android phone utilizing the Android OS authenticator. *See* **Exhibit 15** ("Chrome on all desktop platforms supports using passkeys from mobile devices."). Alternatively, if Sign in with Google is being utilized, a analogous is sent directly to the user's Android phone.

65. Android phones include the Android SafetyNet Authenticator, which is FIDO certified. *See* FIDO® Certified - FIDO Alliance, **Exhibit 31**. SafetyNet provides a set of services and APIs to protect against fake users and is present on Google-approved Android devices. *See* **Exhibit 32** ("SafetyNet provides a set of services and APIs that help protect your app against security threats, including device tampering, bad URLs, potentially harmful apps, and fake users."); **Exhibit 33** ("Android devices offer a "SafetyNet API," which is part of the Google Play Services layer installed on Google-approved Android devices. ").

66. The SafetyNet Authenticator is accessed via a FIDO client that uses the FIDO2 API for Android. *See* **Exhibit 19** ("The FIDO2 API allows Android applications to create and use strong, attested public key- based credentials for the purpose of authenticating users. The API provides a WebAuthn Client implementation, which supports the use of BLE, NFC, and USB roaming authenticators (security keys) as well as a platform authenticator, which allows the user to authenticate using their fingerprint or screenlock.") Using these APIs, the SafetyNet Authenticator can be located. In operation, the SafetyNet Authenticator utilizes the native capacities of the Android OS to provide a secure authenticator. *See* **Exhibit 34** ((("The typical

way to develop secure authenticator on Android mobile devices (smartphones and tablets) is to use a secured hardware-backed operating environment... This includes technology such as TEE ("Trusted Execution Environment") that performs cryptographic and other sensitive operations including digital signing and biometric data processing used to support FIDO functionality... Android Keystore allows app developers to store cryptographic keys in a container and use them in cryptographic operations via APIs. Android also offers protection of the fingerprint sensor data via a TEE, which allows for encryption and cryptographic authentication... Adding a small application to complement some function that are necessary to perform as UAF 1.1 authenticator enables Android Keystore with hardware-backed key attestations and fingerprint sensor to become a secure FIDO UAF 1.1 authenticator."). The Android OS runs on all Android phones (such as the Pixel), and thus provides such devices with the necessary capabilities for passwordless FIDO Passkey authentication via Google Identity Service.

67. This brings the process to step 5, wherein the user is authenticated. When "Sign in with Google" is being utilized the user is prompted to complete a biometric "challenge," which requires receiving scan data from a biometric scan and comparing the scan data to biometric data persistently stored in a tamperproof format discussed above. A similar challenge is presented in when signing on with passkey. *See* **Exhibit 23.**

68. Devices utilizing the Android OS persistently store biometric data in a tamper proof format. Android's implementation guidelines require tamper-proof "raw fingerprint data or derivatives (for example, templates) [that] must never be accessible from outside the sensor driver or TEE" (trusted execution environment) and "fingerprint acquisition, enrollment, and recognition must occur inside the TEE." *See* **Exhibit 35**. When following these guidelines, requiring acquisition and recognition to occur within the TEE means that the biometric data never leaves the

TEE. Android's TEE, called Trusty, "uses ARM's TrustzoneTM to virtualize the main processor and create a secure trusted execution environment" isolated from the rest of the system. *See* **Exhibit 36**. Accordingly, the biometric data, which never leaves the TEE, also never leaves the Trustzone housing, Trusty. Keeping biometric data within the Trustzone, Android phones persistently store biometric data in a tamper proof format.

69. Additionally, Android phones require user consent to enroll a fingerprint. *See* **Exhibit 37**. As enrolling fingerprints on Android phones require entering PIN / Passcode / Password to evidence user consent, Android phones store biometric data of user in a tamper proof format unable to be subsequently altered.

70. Furthermore, access to the biometric hardware on Android devices is controlled by Fingerprint HIDL. *See* **Exhibit 35**. The methods enabled by the Fingerprint HIDL do not permit altering biometric data. *See id.*

71. In order to perform biometric verification of the user, Android OS authenticators utilized within Google's universal platform password-less architecture causes the device to prompt a user for biometric verification and receive scan data from a biometric scan.

72. When the user is biometrically verified using the fingerprint sensor, the process proceeds to step 6 where a FIDO passkey is used to sign the challenge. Signing the is accomplished utilzing a information within memory that can only ba accessed by a correspnding access key provided by an extenral application. The FIDO CTAP specification incorporates the WebAuthn Specification. Under the WebAuthn specification, "compliant authenticators protect public key credentials." *See* **Exhibit 38**. A public key credential refers to a public key credential source, which includes a credential ID. *Id.* The credential ID uniquely identifies its public key credential source. *Id.* In addition to the credential ID, each public key credential source contains a "credential

private key". *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.* Authenticators within the Android OS, therefore, will 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.

73. The credentials stored by the Android OS authenticator can only be accessed with the appropriate access key. During a WebAuthn authentication ceremony, the Android authenticator receives an "authenticatorGetAssertion", issued by Google Identity, containing the RP ID. *Id.* "A public key credential can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." *Id.* When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by Google Identity in the authenticatorGetAssertion request. *Id.* Therefore, the RP ID is an access key.

74. The credential (i.e. Passkey) used to generate the signature within the request is synced with all of a user's Android phones. *See* **Exhibit 15** ("Passkeys are stored when the user creates a passkey on an Android device, and synced; their passkeys are synch'd wth user's other Android devices...") The passkey, accordingly, is not unique to any particular device, but rather backed up and shared across devices.

75. Passkeys in the Google Password Manager are always end-to-end encrypted. *See* **Exhibit 39.** When a passkey is backed up, its private key is uploaded only in its encrypted form using an encryption key that is only accessible on the user's own devices. *Id*. Without access to the private key, such an attacker cannot use the passkey to sign into its corresponding online account. *Id*. Accordingly, Android phones must contain a secrete decryption value (i.e., the encryption key only accessible on the user's own device) to decrypt and use synced and

backed up passkeys. If this decryption value was altered, it would longer be able to decrypt the passkey for use. As such, Android phones contain a secret decryption value in a tamper proof format unable to subsequently altered.

76. The passkey is held securely within the keystore until it unlocked following biometric verification of the user and used to "sign the service's challenge". See **Exhibit 26**; *see also* **Exhibit 40 (**"When generating or importing a key into the Android KeyStore, you can specify that the key is only authorized to be used if the user has been authenticated. The user is authenticated using a subset of their secure lock screen credentials (pattern/PIN/password, biometric credentials)."). Accordingly, by utilizing Fingerprint HIDL in combination with the KeyStore, phone utilzing the Android OS securely sign challenges received from Google Identity only after successful biometric verification.

77. When using a Titan Passkey or the passkey functionality of the Android OS, step 6 concludes with transferring the signed challenge back the user computer over bluetooth. **Exhibit 23**. The FIDO standard requires all communications with BLE authenticators be encrypted. *See* **Exhibit 41**. FIDO compliant BLE capable authenticators, accordingly, include a receiver-decoder circuit ("RDC") enabling encrypted communications.

78. At step 7 the signed challenge is then sent to Google Identity Servers and verified. Authentication is a service provided by the FIDO server incorporated into Google Identity, and the credential ID is necessary for Google's FIDO server to perform the authentication function. Upon receiving the response (i.e., enablement signal), the server will use the credential ID to locate the appropriate public key to verify a signature generated with the private key held by the authenticator. *Id*. As the proper credential ID is needed for Google's FIDO server to authenticate a user, and the credential ID is included within a response to a get request having the appropriate

relying party ID received from the Google's FIDO server, the response to the authenticatorGetAssertion request generated by the authenticator is an enablement signal enabling authentication by Google's FIDO server. The authenticator, accordingly, generates an enablement signal enabling one or more of an application, a function and a service on a device associated with an external RDC Of course, a critical component of verifying the challenge is verifying that it came from a legitimate device, i.e., one in which the user is currently signed on with their Google Account.

79. The responsibility for sending responses received from an authenticator via BLE or NFC falls upon the WebAuthn Client. *See* **Exhibit 15**. Google identifies its Chrome browser as a WebAuthn client by noting it supports the use of passkeys from mobile devices, including permitting the use of an Android phone as a roaming authenticator on Windows. *Id.* As such, Chrome will forward the enablement signal received from a roaming authenticator to Google's FIDO server.

80. Identifying a device as legitimate or not necessitates some form of device ID uniquely identifying the device. Phones utilizing the Android OS also persistently store a device ID code uniquely identifying the phone in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered. The authenticator integrated into the Android OS supports the Device-bound Public Key Webauthn extension (*devicePubKey*). **Exhibit 39.** Device bound public keys are part of a hardware bound key pair that is used to identify devices. Each time a user authenticates with their Android phone, Google Identity servers will receive a public key uniquely identifying the Android phone. With this unique device ID, Google Identity Services can determine if the Android phone is a legitimate device. Rather, the device-bound key pair enables a device ID, in the form of the device-bound public

key stored on the user's device, to returned with the signed challenge to ensure it originated from a legitimate device. As device-bound public keys are not synced, "observing two passkey signatures with the same device-bound public key is a strong signal that the signatures are generated by the same device." *Id.* On the other hand, if a relying party observes a device-bound public key it has not seen before, this may indicate that the passkeys has been synced to a new device. *Id* ("This *device-bound* private key is unique to the passkey in question, and each response includes a copy of the corresponding device-bound public key.").

81. Device Bound Public Keys are stored in a format that when altered causes verification to fail. When a Device Bound Public Key is present, the integrated authenticator within the Android OS causes the phone to store a private key bound to the device. **Exhibit 29** at § 10.2.2.2 (Defining making a credential with a device bound public key as including "5. Let dpk be the newly created or existing device public key, in COSE_Key format in the same fashion as for the user credential's credentialPublicKey when the latter is conveyed in attested credential data. 6. Let devicePrivateKey be the newly created or existing device private key.")) The public key corresponding to the private key is returned to the relying party and saved. **Exhibit 29** at § 10.2.2.3.1 (Stating registration at the relying party includes step "6. Create a new device-bound key record … add this device-bound key record to the devicePubKeys member of the new credential record.").) When processing an authentication request received from Google Identity, the Android OS authenticator creates a signature with the private key associated with the device-bound public key and then sends the signature and the device-bound public key back to Google Identity for verification. *Id*. at § 10.2.2.2 (Defining authentication as including "9. Let dpkSig be the result of signing the assertion signature input with devicePrivateKey. 10. Output dpkSig as the extension's unsigned extension output.").) Upon receipt of the authentication response, the

Google Identity extracts the device bound public key from the response and uses it to verify the signature. *Id.* at § 10.2.2.3.2 (Defining Relying Party verification of the authentication assertion as including "3. Extract the contained fields from attObjForDevicePublicKey: aaguid, dpk, scope, nonce, fmt, attStmt. 4. Verify that signature is a valid signature over the assertion signature input (i.e. authData and hash) by the device public key dpk.")) If the device bound public key were changed at any time, Google Identity would not be able to verify the signature as it would no longer correspond to the authenticator's device bound private key. Accordingly, by being part of a key pair, the device bound public key is in a format unable to be subsequently altered.

82. Of course, if both the private key and public key were altered, then the signature could be verified. In such a situation, however, the altered device bound public key would not match any of the device bound public keys stored by the Relying Party (i.e., Microsoft and Google). *See Id.* at § 10.2.2.3.2 (Steps 5 and 6 detailing, during an authentication ceremony, comparing the received device bound public key to a list of stored device bound public keys, where the lack of a match indicates a new device.). The Relying Party, accordingly, would treat the altered device bound public key as representing a new device and not allow the request absent further user verification. Accordingly, by being recorded by the relying party, device bound public keys are further placed in format unable to be subsequently altered.

~~83.~~ Device-bound public keys are further placed in tamper proof format unable to be subsequently altered by requiring user consent to be created. A device-bound public keys are created when either first using an Android OS authenticate or when authenticating with a new device. **Exhibit 39.** When an authenticator is first used to authenticate an new credential is created, which requires user verification. **Exhibit 41**, at § 6.1 ("For backwards compatibility,

platforms must be aware that FIDO_2_0 (aka CTAP2.0) authenticators always require some form of user verification for authenticatorMakeCredential operations. If a platform attempts to create a non-discoverable credential on a CTAP2.0 authenticator without including the "uv" option key or the pinUvAuthToken parameter that authenticator will return an error. In contrast, a FIDO_2_1 (aka CTAP2.1) authenticator with the makeCredUvNotRqd option ID (set to true) in the authenticatorGetInfo response structure, will allow creation of non-discoverable credentials without requiring some form of user verification.").) Device-bound public keys are also created when user a authenticates. **Exhibit 39.** "The device-bound key pair is created and stored on-demand. That means relying parties can request the devicePubKey extension when getting a signature from an existing passkey, even if devicePubKey was not requested when the passkey was created."). Authentication is the process of verifying the user, and thus requires user consent to complete.

~~84.~~ Google Identity also includes a third party trusted authority possessing a list of device ID codes uniquely identifying legitimate integrated devices. At the end of step 7, if Google Identity Services verify the signed challenge and that the public key corresponds to a legitimate device, an access message is returned in the form of an authorization code.

85. Finally at step 8, the user agent utilizes the authentication code to log the user into the website by exchanging the authorization code for the appropriate token. If resources in addition to the website are to be accessed, the Authorization Code may be exchanged for an ID Token or an Access Token. To make the exchange, Google requires the client agent (i.e., website loaded in the computer's browser) send a POST request containing the authorization code received from Google, as well as a client ID and client secret obtained when the developer registered the User Agent with Google Identity Services. *See* **Exhibit 42**. The information

contained in the ID Token is then used to querry the developer's database to determine if the user exists, and if so, to start sesssion for the user given access to the website or application. *Id.* ("After obtaining user information from the ID Token, you should querry your app's database. If the user already exsits in your database, you should start an appication session for that user if all login requirements are met by the Google API response."). The Authorization Code received from Google Identity Services is a message permitting access to the application or website by being exchangable for an ID Token used to allow access.

86. If, however, the User Agent needs to access a resource such file in Google Drive or other resource, the authentication code may be exhanged for an Access Token. *Id.* ("One of the advantages of using OAuth 2.0 for authentication is that your application can get permission to use other Google APIs on hehalf of the user (such as YouTube, Google Drive, Calendar, or Contacts) at the same time as you authenticate the user."). To exchange the Authorization Code for an Access Token, Google requires the client agent send a POST request containing the authorization code received from Google, as well as a client ID and client secret obtained when the developer registered the User Agent with Google Identity Services. *Id.* The Authorization Code received from Google Identity Service, thus, is a message permitting access to files and applications associated with Google APIs.

87. As the foregoing shows, Google Identity is at the center of Google's universal platform password-less architecture. It receives messages sent by applications, services, and websites in a manner prescribed by Google. It directs the action of Android OS authenticators, and it issues bear tokens in the manner that Google chooses. Accordingly, Google directs and controls the actions of developers and partners wishing to receive the benefit of utilizing or being part of the Accused Product.

88. 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 for the reason that § 287 does not apply.

a) Google Pay / Google Wallet and Android Phones Pre-Loaded Therewith

89. Google directly infringes the Patents-in-Suit by, for example, selling Pixel phones preloaded with Google Pay. Google also actively induces infringement the Patents-in-Suit by disseminating Google Wallet and Google Pay as the default payment app of Android OS. When an Android phone equipped with Google Pay in the lock state is presented to make a contactless payment transaction, the user infringes the Patents-in-Suit.

90. Android phones can make contactless payments. *See* **Exhibit 43** and **Exhibit 44** ("Google Pay can be used to make NFC transactions in stores…"). To do so, users can hold their Android phone close to a payment terminal. If the device is locked, the user will be prompted to unlock the phone, which can be done with a fingerprint (or other biometric authentication mechanism available and supported). *See* **Exhibit 45**.

91. Unlocking a phone with a fingerprint, for example, requires receiving scan data from a biometric scan and comparing the scan data to biometric data persistently stored in a tamperproof format. On Android devices, access to the biometric hardware is controlled by Fingerprint HIDL. "Android uses Fingerprint Hardware Interface Definition Language (HIDL) to connect to a vendor-specific library and fingerprint hardware (for example, a fingerprint sensor)." *See* **Exhibit 35**.

92. The methods enabled by the Fingerprint HIDL do not permit altering biometric data. *Id*. (providing a listing of methods, none of which allowing for the alternation of biometric

data.). Keeping fingerprint and other biometric data within a portion of the Trustzone only accessible by Fingerprint HIDL, which lacks a method for altering biometric data, Android devices persistently store biometric data of the user in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered.

93. Android's implementation guidelines require tamper-proof "raw fingerprint data or derivatives (for example, templates) [that] must never be accessible from outside the sensor driver or TEE" (trusted execution environment) and "fingerprint acquisition, enrollment, and recognition must occur inside the TEE." Android Open-Source Project: Fingerprint HIDL, *Id.* Android's TEE, called Trusty, "uses ARM's TrustzoneTM to virtualize the main processor and create a secure trusted exaction environment" isolated from the rest of the system. *See* **Exhibit 36**. Accordingly, fingerprint data, which never leaves the TEE, also never leaves the Trustzone housing, Trusty. Keeping biometric data within the Trustzone, Android phones persistently store biometric data in a tamper proof format.

94. When the user is biometrically verified with the fingerprint sensor and the phone unlocked, an EMV Payment Token stored on the phone is sent via NFC to the payment terminal. "When a user successfully adds their card to Google Pay, Google Pay stores a uniquely generated token on the device that has its own value." *See* **Exhibit 46**. A user with multiple device or multiple Google account on the same device can add the same card to each of their accounts and devices. *Id*. Though each user tokenizes a card using the same PAN, each wallet receives a unique number. *Id*.

95. This new number, called a dynamic primary account number (DPAN) or device token, is similar to a credit card number. *Id.* As each wallet receives a unique DPAN, i.e., Token, a token is a device ID uniquely identifying an Android device.

96. Provisioning is the process "whereby a Payment Token and related data are delivered to the Token Location." EMV Payment Tokenisation Specification, Technical Framework v2.2, page 11. Android phones support Host Card Emulation (HCE) to enable contactless payment transactions. "When NFC card emulation is provided using a secure element, the card to be emulated is provisioned into the secure element on the device through an Android application." *See* **Exhibit 43.**

97. Since 2018, all Google's Pixel phones have included a "Titan M" chip, a tamper-resistant hardware enclave, also known as a Secure Element ("SE"). See **Exhibit 47**. An SE, according to the U.S. Payments Forum, is "a dynamic environment to store data securely, process data securely and perform communication with external entities securely," that "will not allow unauthorized access." Accordingly, the payment token (DPAN) uniquely identifying an Android phone is stored in a tamperproof format within the Android phone's secure element. After unlocking an Android phone held close to the payment terminal with a fingerprint (or other biometric verification), Google Pay passes the token to the payment terminal instead of the actual card number. *See* **Exhibit 46** ("A DPAN improves account security because Google Pay passes it to a terminal during payment instead of the actual card number.")

98. After being received by the payment terminal, the token is forwarded to Token Service Provider. First the token is packaged into a Token Payment Request by the payment terminal and passed to the payment network. *See* **Exhibit 48,** Figure 10.1 and Table 10-1. During transaction routing within the payment network, the Token Payment Request is transformed to a Token Authorization Request, which still contains the token uniquely identifying the Android device. *Id.,* Figure 10.1 and Table 10-3. Accordingly, the payment token sent in the Token Payment Request from the merchant continues to persist during the

token authorization request process. "The Token Authorisation request process continues until De-Tokenisation has been completed." *Id.*, page 86. De-Tokenisation is performed by the token service provider.

99. "Token Service Providers are responsible for a number of discrete functions which may include, but are not limited to: Maintenance and operation of a Token Vault . . . [and] De-Tokenisation". *Id.* A Token Vault is repository that maintains a mapping of the token to the underlying credit card number. *Id.,* page 12 (Defining Token Vault as "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."). By maintaining the token vault, token service providers keep a list of device ID codes uniquely identifying legitimate integrated devices. Accordingly, token service providers are one example of a third-party trusted authority possessing a list of device ID codes uniquely identifying legitimate Android phones.

100. Detokenisation is "the process of converting a Payment Token and Token Expiry Date to its underlying PAN and PAN Expiry Date based on the Payment Token / Token Expiry Date mapping to the underlying PAN / PAN Expiry Date stored in the Token Vault." By converting the payment token to its underlying account number based on the Payment Token / Token Expiry Date mapping to the underlying PAN / PAN Expiry Date stored in the token vault, the token service provider necessarily authenticates the payment token received from the Samsung Pay preloaded smartphone.

101. After the token has been de-tokenized, the actual credit card number received from the token vault is placed within a PAN Authorization Request which is sent to the bank that issued the credit card. *Id*. Upon receipt of the request, the issuer completes final authorization and sends an authorization response. Accordingly, the authorization request containing the PAN

in place of the payment token allows the user access to the issuing bank's computer software necessary to process and authorize the payment.

The issuing banks then sends a PAN Authorization Response back to the Token Service Provider. A push notification is then sent from the TSP to the Android Phone. *See* **Exhibit 49**, Figures 6, 7, and 11. As the push notification is received in response to processing of the transaction request by the bank, the push notification received indicates allowed access to the banks software necessary to process the transaction.

b. Google's Indirect Infringement of the Patents-in-Suit

102. Google actively induces infringement of the Patents-in-Suit by taking active steps to encourage direct infringement, despite having actual and constructive knowledge about the Patents-in-Suit, as alleged above, and that the induced acts would amount to infringement of the Patents-in-Suit. Specifically, Google actively engaged in encouraging infringement of the Patents-in-Suits by creating, providing and maintaining a substantial knowledge base online, teaching how to use the features and how to integrate its universal platform password-less architecture into various applications, websites, and processes.

103. The knowledge base includes advertising for the infringing features of its password-less architecture.

104. In addition to educating consumers on how to actively engage in infringing uses, Google is actively creating a knowledge base among developers on how to integrate their resources to exploit Google's universal platform password-less architecture. By teaching developers how to integrate their resources into its password-less architecture, Google is assisting in performing the infringing uses of Microsoft Identity.

105. Through integration and developer guides, instructions on Google's websites, and advertising, Google has created and is actively providing and maintaining a knowledge base encouraging infringement of the Patents-in-Suit.

106. Google also actively contributes to infringement of the Patents-in-Suit by providing a native authenticator within the Android OS to allow users to incorporate their devices into Google's universal platform password-less architecture, as these have no substantial non-infringing use, and are especially made for such infringement.

107. As noted *supra* with regards to direct infringement, Titan Security Keys and native authenticator within the Android OS are components of the Accused Products. Accordingly, they are especially made for infringement of the Patents-in-Suit.

CLAIM 1
(Infringement of the 730 Patent)

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

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

110. Google 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.

111. For example, Google 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 to within the United States Google's universal platform password-less architecture incorporating the Android OS integrated authenticator. Under the coordination of Google Identity, the integrated Android OS authenticator performs/executes and provides, a method for verifying a

user during authentication of the device. Google also infringes at least claims 15, 16, and 17 of the 730 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering to sell access to within the United States Google's universal platform password-less architecture incorporating Google Identity. The coordination and control provided by Google Identity of the other components within the architecture provides a system for verifying a user during authentication of a device.

112. Google has induced infringement, and continues to induce infringement, of at least claims 1, 2, 3, and 5 of the 730 Patent in violation of 35 U.S.C. § 271 by providing use of its universal platform password-less architecture incorporating integrated Android OS authenticator, Chrome browser, and Google Identity for use by users to access resources offered by Google, including applications, services, and subscriptions. Google also induces infringement of claims 15, 16, and 17 by making Google Identity available for integration with developer applications, and substantial knowledge base teaching developers and business subscribing to Google's Identity and Access Management services about the features, use and integration of their resources into the password-less architecture. 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.

113. Google contributes to direct infringement of at least claims 1, 2, 3, and 5 of the 730 Patent in violation of 35 U.S.C. § 271(c) by providing use of its universal platform password-less architecture incorporating integrated Android OS authenticator, Chrome browser, and Google Identity for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Google also induces infringement of claims 15, 16, and 17 by making Google Identity available for integration with developer applications, providing the Microsoft Authentication Library, and substantial knowledge base teaching developers and business

subscribing to Google's Identity and Access Management services about the features, use and integration of their resources into the password-less architecture. 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.

114. Google received actual notice of the 730 Patent at least as early as the filing of this Complaint. Google 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.

115. Since at least the date of service of this Complaint, through its actions and continued actions, Google has indirectly infringed and continues to indirectly infringe the 730 Patent in violation of 35 U.S.C. § 271(b). Google has actively induced product makers and/or end users of the Accused Products to directly infringe the 730 Patent throughout the United States, including within this Judicial District, by, among other things, advertising and promoting the use of the Accused Products on various websites and in marketing material, including providing and disseminating product descriptions, operating manuals, and other instructions on how to implement and configure the Accused Products. Some examples of Google promoting the use of the Accused Product are the public documents discussed supra, which serve no function other than to direct users of the Accused Products toward infringing the 730 Patent.

116. Google does so knowingly and intending that its customers and end users will commit these infringing acts. Google also continues to make, use, offer for sale, sell, and/or import the Accused Products, despite its knowledge of the 730 Patent, thereby specifically intending for and inducing its customers to infringe the 730 Patent through the customers' normal and customary use of the Accused Products.

117. In addition, Google has indirectly infringed and continues to indirectly infringe the 730 Patent in violation of 35 U.S.C. § 271(c) by selling or offering to sell in the United States, or importing into the United States, the Accused Products with knowledge that they are especially designed or adapted to operate in a manner that infringes the 730 Patent and despite the fact that the infringing technology or aspects of the products are not a staple article of commerce suitable for substantial non-infringing use.

118. The infringing aspects of the Accused Products can be used only in a manner that infringes the 730 Patent and thus have no substantial non-infringing uses. The infringing aspects of those instrumentalities otherwise have no meaningful use, let alone any meaningful noninfringing use.

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

120. Upon information and belief, Google 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 Google.

CLAIM 2
(Infringement of 954 Patent)

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

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

123. Google 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.

124. For example, Google 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 to within the United States Microsoft's universal platform password-less architecture incorporating integrated Android OS authenticator. Under the coordination of Google Identity, the authenticator performs/executes and provides a method for verifying a user during authentication of the device. Google also infringes at least claims 22, 23, 24, 25, 26, and 27 of the 954 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering to sell access to within the United States access to Google's universal platform password-less architecture incorporating Google Identity. The coordination and control provided by Google Identity Platform of the other components within the architecture provides a system for verifying a user during authentication of a device.

125. Google 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 use of its universal platform password-less architecture incorporating integrated Android OS authenticator, Chrome browser, and Google Identity for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Google also induces infringement of claims 22, 23, 24, 25, 26, and 27 by making Google Identity available for integration with developer applications and substantial knowledge base teaching developers and business subscribing to Google's Identity and Access Management services about the features, use and integration of their

resources into the password-less architecture. 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.

126. Google contributes to direct infringement of at least claims 1, 2, 3, 5, 6, and 7 of the 954 Patent in violation of 35 U.S.C. § 271(c) by providing use of its universal platform password-less architecture incorporating integrated Android OS authenticator, Chrome browser, and Google Identity for use by users to access resources offered by Google, including applications, services, and subscriptions. Google also induces infringement of claims 22, 23, 24, 25, 26, and 27 by making Google Identity available for integration with developer applications and substantial knowledge base teaching developers and business subscribing to Google's Identity and Access Management services about the features, use and integration of their resources into the password-less architecture. 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.

127. Google received actual notice of the 954 Patent at least as early as the filing of this Complaint. Google 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.

128. Since at least the date of service of this Complaint, through its actions and continued actions, Google has indirectly infringed and continues to indirectly infringe the 954 Patent in violation of 35 U.S.C. § 271(b). Google has actively induced product makers and/or end users of the Accused Products to directly infringe the 954 Patent throughout the United States, including within this Judicial District, by, among other things, advertising and promoting the use of the Accused Products on various websites and in marketing material, including providing and disseminating product descriptions, operating manuals, and other instructions on

how to implement and configure the Accused Products. Some examples of Google promoting the use of the Accused Product are the public documents discussed supra, which serve no function other than to direct users of the Accused Products toward infringing the 954 Patent.

129. Google does so knowingly and intending that its customers and end users will commit these infringing acts. Google also continues to make, use, offer for sale, sell, and/or import the Accused Products, despite its knowledge of the 954 Patent, thereby specifically intending for and inducing its customers to infringe the 954 Patent through the customers' normal and customary use of the Accused Products.

130. In addition, Google has indirectly infringed and continues to indirectly infringe the 954 Patent in violation of 35 U.S.C. § 271(c) by selling or offering to sell in the United States, or importing into the United States, the Accused Products with knowledge that they are especially designed or adapted to operate in a manner that infringes the 954 Patent and despite the fact that the infringing technology or aspects of the products are not a staple article of commerce suitable for substantial non-infringing use.

131. The infringing aspects of the Accused Products can be used only in a manner that infringes the 954 Patent and thus have no substantial non-infringing uses. The infringing aspects of those instrumentalities otherwise have no meaningful use, let alone any meaningful noninfringing use.

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

133. Upon information and belief, Google 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 730 Patent by Google.

CLAIM 3
(Infringement of 905 Patent)

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

135. Proxense has not licensed or otherwise authorized Microsoft to make, use, offer for sale, sell, or import any products that embody the inventions of the 905 Patent.

136. Google infringes at least claims 1, 2, and 15 of the 905 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.

137. For example, Google directly infringes at least claims 1 and 2 of the 905 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering to sell access to within the United States Microsoft universal password-less architecture incorporating the integrated Android OS authenticator. Under the coordination of Google identity, the authenticators perform/execute and provide a method for verifying a user during authentication of the device. Google also infringes at least claim 15 of the 905 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering access to sell within the United States Google's universal platform password-less architecture incorporating Google Identity. The coordination and control provide by Google Identity of the other components within the architecture provides a system for verifying a user during authentication of the device.

138. Google has induced infringement, and continues to induce infringement, of at least claims 1 and 2 of the 905 Patent in violation of 35 U.S.C. § 271 by providing use of its universal

platform password-less architecture incorporating the integrated Android OS authenticator, Chrome browser, and Google Idenity for use by users to access resources offered by Google, including applications, services, and subscriptions. Google also induces infringement of claim 15 by making Google Identity available for integration with developer applications and substantial knowledge base teaching developers and business subscribing to Google's Identity and Access Management services about the features, use and integration of their resources into the password-less architecture. 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. Google contributes to direct infringement of at least claims 1 and 2 of the 905 Patent in violation of 35 U.S.C. § 271(c) by providing use of its universal platform password-less architecture incorporating the integrated Android OS authenticator, Chrome browser, and Google Identity for use by users to access resources offered by Google, including applications, services, and subscriptions. Google also induces infringement of claim 15 by making Google Identity available for integration with developer applications and substantial knowledge base teaching developers and business subscribing to Google's Identity and Access Management services about the features, use and integration of their resources into the password-less architecture. 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.

140. Google received actual notice of the 905 Patent at least as early as the filing of this Complaint. Google 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 905 Patent.

141. Since at least the date of service of this Complaint, through its actions and continued actions, Google has indirectly infringed and continues to indirectly infringe the 954 Patent in violation of 35 U.S.C. § 271(b). Google has actively induced product makers and/or end users of the Accused Products to directly infringe the 905 Patent throughout the United States, including within this Judicial District, by, among other things, advertising and promoting the use of the Accused Products on various websites and in marketing material, including providing and disseminating product descriptions, operating manuals, and other instructions on how to implement and configure the Accused Products. Some examples of Google promoting the use of the Accused Product are the public documents discussed supra, which serve no function other than to direct users of the Accused Products toward infringing the 905 Patent.

142. Google does so knowingly and intending that its customers and end users will commit these infringing acts. Google also continues to make, use, offer for sale, sell, and/or import the Accused Products, despite its knowledge of the 905 Patent, thereby specifically intending for and inducing its customers to infringe the 905 Patent through the customers' normal and customary use of the Accused Products.

143. In addition, Google has indirectly infringed and continues to indirectly infringe the 905 Patent in violation of 35 U.S.C. § 271(c) by selling or offering to sell in the United States, or importing into the United States, the Accused Products with knowledge that they are especially designed or adapted to operate in a manner that infringes the 954 Patent and despite the fact that the infringing technology or aspects of the products are not a staple article of commerce suitable for substantial non-infringing use.

144. The infringing aspects of the Accused Products can be used only in a manner that infringes the 905 Patent and thus have no substantial non-infringing uses. The infringing aspects

of those instrumentalities otherwise have no meaningful use, let alone any meaningful noninfringing use.

145. Proxense has been injured and seeks damages to adequately compensate it for Google's infringement of the 905 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

146. Upon information and belief, Google will continue to infringe (both directly and indirectly) the 905 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 905 Patent by Google.

CLAIM 3
(Infringement of 402 Patent)

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

148. Proxense has not licensed or otherwise authorized Google to make, use, offer for sale, sell, or import any products that embody the inventions of the 402 Patent.

149. Microsoft infringes at least claim 1 of the 042 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.

150. For example, Google directly infringes at least claim 1 of the 042 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering to sell access to within the United States Google's universal platform password-less architecture incorporating the integrated Android OS authenticator, Titan Security Key authenticator, and Google Identity. Under the coordination of Google Identity, the authenticators perform/execute and provide, a method for verifying a user during authentication of the device.

151. Google has induced infringement, and continues to induce infringement, of at least claim 1 of the 042 Patent in violation of 35 U.S.C. § 271 by making Google Identity available for integration with developer applications and creating a knowledge base on how to do so. 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.

152. Google contributes to direct infringement of at least claim 1 of the 042 Patent in violation of 35 U.S.C. § 271(c) by Google Microsoft Identity available for integration with developer applications and creating a knowledge base on how to do so. 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.

153. Google received actual notice of the 042 Patent at least as early as the filing of this Complaint. Google 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.

154. Since at least the date of service of this Complaint, through its actions and continued actions, Google has indirectly infringed and continues to indirectly infringe the 042 Patent in violation of 35 U.S.C. § 271(b). Google has actively induced product makers and/or end users of the Accused Products to directly infringe the 042 Patent throughout the United States, including within this Judicial District, by, among other things, advertising and promoting the use of the Accused Products on various websites and in marketing material, including providing and disseminating product descriptions, operating manuals, and other instructions on how to implement and configure the Accused Products. Some examples of Google promoting the

use of the Accused Product are the public documents discussed supra, which serve no function other than to direct users of the Accused Products toward infringing the 042 Patent.

155. Google does so knowingly and intending that its customers and end users will commit these infringing acts. Google also continues to make, use, offer for sale, sell, and/or import the Accused Products, despite its knowledge of the 042 Patent, thereby specifically intending for and inducing its customers to infringe the 042 Patent through the customers' normal and customary use of the Accused Products.

156. In addition, Google has indirectly infringed and continues to indirectly infringe the 905 Patent in violation of 35 U.S.C. § 271(c) by selling or offering to sell in the United States, or importing into the United States, the Accused Products with knowledge that they are especially designed or adapted to operate in a manner that infringes the 042 Patent and despite the fact that the infringing technology or aspects of the products are not a staple article of commerce suitable for substantial non-infringing use.

157. The infringing aspects of the Accused Products can be used only in a manner that infringes the 042 Patent and thus have no substantial non-infringing uses. The infringing aspects of those instrumentalities otherwise have no meaningful use, let alone any meaningful noninfringing use.

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

159. Upon information and belief, Google will continue to infringe (both directly and indirectly) 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 Google.

CLAIM 5
(Infringement of 289 Patent)

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

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

162. Google infringes at least claims 14 and 16 of the 289 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.

163. For example, Google directly infringes at least claims 14 and 16 of the 289 Patent by making, using (*e.g.*, performing/executing), selling access to, and/or offering to sell access to within the United States Google's universal platform password-less architecture incorporating the integrated Android OS authenticator, Titan Security Key authenticator, and Google Identity. Under the coordination of Google Identity, the authenticators perform/execute and provide, a method for verifying a user during authentication of the device.

164. Google has induced infringement, and continues to induce infringement, of at least claims 14 and 16 of the 289 Patent in violation of 35 U.S.C. § 271 by making Google Identity available for integration with developer applications and creating a knowledge base on how to do so. 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.

165. Google contributes to direct infringement of at least claims 14 and 16 of the 289 Patent in violation of 35 U.S.C. § 271(c) by making Google Identity available for integration with

developer applications and creating a knowledge base on how to do so. 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.

166. Google received actual notice of the 289 Patent at least as early as the filing of this Complaint. Google 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 289 Patent.

167. Through its actions and continued actions, Google has indirectly infringed and continues to indirectly infringe the 289 Patent in violation of 35 U.S.C. § 271(b). Google has actively induced product makers, distributors, retailers, and/or end users of the accused products to directly infringe the 289 Patent throughout the United States, including within this Judicial District, by, among other things, advertising and promoting the use of the accused products on various websites and in marketing material, including providing and disseminating product descriptions, operating manuals, and other instructions on how to implement and configure the accused products. Some examples of Google promoting the use of the accused products are packaging Google Identity with Android phones and public documents, which serve no function other than to direct users of the Accused Products toward infringing the 289 Patent.

168. Google does so knowingly and intending that its customers and end users will commit these infringing acts. Google also continues to make, use, offer for sale, sell, and/or import the accused products, despite its knowledge of the 289 Patent, thereby specifically intending for and inducing its customers to infringe the 289 Patent through the customers' normal and customary use of the Accused Products.

169. In addition, Google has indirectly infringed and continues to indirectly infringe the 289 Patent in violation of 35 U.S.C. § 271(c) by selling or offering to sell in the United States, or importing into the United States, the accused products with knowledge that they are especially designed or adapted to operate in a manner that infringes the 289 Patent and despite the fact that the infringing technology or aspects of the products are not a staple article of commerce suitable for substantial non-infringing use.

170. For example, Google is aware that the technology described above included in the accused products enables the product to operate as described above and that such functionality infringes the 289 Patent, including claims 14 and 16. Google continues to sell and offer to sell these products in the United States after receiving notice of the 289 Patent and how its products infringe that patent.

171. The infringing aspects of the Accused Products can be used only in a manner that infringes the 289 Patent and thus have no substantial non-infringing uses. The infringing aspects of those instrumentalities otherwise have no meaningful use, let alone any meaningful noninfringing use.

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

173. Upon information and belief, Google will continue to infringe (both directly and indirectly) 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 Google.

CLAIM 6
(Infringement of the 960 Patent)

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

175. Proxense has not licensed or otherwise authorized Google to make, use, offer for sale, sell, or import any products that embody the inventions of the 960 Patent.

176. Google infringes at least claims 14 and 16 of the 960 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.

177. For example, Google directly infringes at least claims 14 and 16 selling access to, and/or offering to sell access to within the United States Google's universal platform password-less architecture incorporating the integrated Android OS authenticator, Titan Security Key authenticator, and Google Identity. Under the coordination of Google Identity, the authenticators perform/execute and provide, a method for verifying a user during authentication of the device.

178. Google received actual notice of the 960 Patent at least as early as the filing of this Complaint. Google 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 960 Patent.

179. Through its actions and continued actions, Google has indirectly infringed and continues to indirectly infringe the 960 Patent in violation of 35 U.S.C. § 271(b). Google has actively induced product makers, distributors, retailers, and/or end users of the accused products to directly infringe the 960 Patent throughout the United States, including within this Judicial District, by, among other things, advertising and promoting the use of the accused products on various websites and in marketing material, including providing and disseminating product

descriptions, operating manuals, and other instructions on how to implement and configure the accused products. Some examples of Google promoting the use of the accused products are packaging Google Identity with Android phones and public documents, which serve no function other than to direct users of the Accused Products toward infringing the 960 Patent.

180. Google does so knowingly and intending that its customers and end users will commit these infringing acts. Google also continues to make, use, offer for sale, sell, and/or import the accused products, despite its knowledge of the 960 Patent, thereby specifically intending for and inducing its customers to infringe the 960 Patent through the customers' normal and customary use of the Accused Products.

181. In addition, Google has indirectly infringed and continues to indirectly infringe the 960 Patent in violation of 35 U.S.C. § 271(c) by selling or offering to sell in the United States, or importing into the United States, the accused products with knowledge that they are especially designed or adapted to operate in a manner that infringes the 960 Patent and despite the fact that the infringing technology or aspects of the products are not a staple article of commerce suitable for substantial non-infringing use.

182. For example, Google is aware that the technology described above included in the accused products enables the product to operate as described above and that such functionality infringes the 960 Patent, including claims 14 and 16. Google continues to sell and offer to sell these products in the United States after receiving notice of the 960 Patent and how its products infringe that patent.

183. The infringing aspects of the Accused Products can be used only in a manner that infringes the 960 Patent and thus have no substantial non-infringing uses. The infringing aspects

of those instrumentalities otherwise have no meaningful use, let alone any meaningful noninfringing use.

184. Proxense has been injured and seeks damages to adequately compensate it for Google's infringement of the 960 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

185. Upon information and belief, Google will continue to infringe (both directly and indirectly) the 960 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 960 Patent by Google.

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 Defendants infringe one or more claims of each of the Patents-in-Suit;

b. Entry of judgment declaring that Defendants' 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: May 2, 2023

Respectfully submitted,

/s/ David L. Hecht
David L. Hecht (**Co-Lead Counsel**)
dhecht@hechtpartners.com
Maxim Price (*pro hac vice* forthcoming)
mprice@hechtpartners.com
Conor B. McDonough (*pro hac vice* forthcoming)
cmcdonough@hechtpartners.com
Yi Wen Wu (*pro hac vice* forthcoming)
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
Bryce T. Barcelo
bbarcelo@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, California 90067-6029
Telephone: (310) 789-3100
Facsimile: (310) 789-3150

Counsel for Plaintiff Proxense, LLC