UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS

PROXENSE, LLC, *Plaintiffs*, v. MICROSOFT CORPORATION, *Defendants*.	Civil Action No. 6:23-cv-00319 **JURY TRIAL REQUESTED**

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Proxense, LLC ("Proxense" or "Plaintiff") hereby sets forth its Complaint for patent infringement against Defendant Microsoft Corporation ("Microsoft" 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 Microsoft 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, Microsoft is a corporation organized and existing under the laws of the State of Washington, with a principal place of business in this district located at 10900 Stonelake Boulevard, Suite 225, Austin, Texas 78759.

JURISDICTION AND VENUE

4. 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.

5. This Court has personal jurisdiction over Microsoft because it has conducted and continues to regularly conduct business within the State of Texas and this District. Microsoft has purposefully and voluntarily availed itself of the privileges of conducting business in the United States, the State of Texas, and this District by continuously and systematically placing goods into the stream of commerce through an established distribution channel with the expectation that they will be purchased by consumers in this District. Microsoft directly and/or through intermediaries (including distributors, sales agents, and others), ships, distributes, sells, offers to sell, imports, advertises, makes, and/or uses its products (including but not limited to the products accused of infringement herein) in the United States, the State of Texas, and this District.

6. Upon information and belief, Microsoft conducts business within the State of Texas and in this district, and has designated Corporation Service Company d/b/a CSC − Lawyers Incorporating Service Company, 211 E. 7th Street, STE 620, Austin, Texas 78701-3218, as its agent for service of process in this district.

7. On information and belief, Microsoft has been registered to do business within the State of Texas under Texas Secretary of State File Number 0010404606 since about March 1987.

8. On information and belief, Microsoft employs one or more of its data centers in this district in furtherance of infringing acts in this district since at least 2008. For example, Microsoft maintains data centers in this district, located at: 5150 Rogers Road, San Antonio,

Texas 78251, 5200 Rogers Road, San Antonio, Texas 78251,[1] and 3823 Wiseman Boulevard, San Antonio, Texas 78251.[2]

9. On information and belief, Microsoft has operated data centers supporting Microsoft products and services within the State of Texas, and within this district, since at least 2008. Microsoft is building at least three additional data centers in this district, including two data centers located at: 3545 Wiseman Boulevard, San Antonio, Texas 78251, and another data center located at 15000 Block Lambda Drive, San Antonio, Texas 78245. Upon information and belief, Microsoft's data centers, including those in this district, include computer hardware (*e.g.*, memory and processors) that store and execute software that performs some of the actions that infringe on the patents in the lawsuit. On information and belief, Microsoft has employed, is employing, and is offering to employ individuals in this district in furtherance of infringing acts in this district. On information and belief, these employees have direct personal knowledge about the accused products and Microsoft's infringing activities. For example, Justin Santos, Senior Cloud Security Architect, purporting according to his LinkedIn profile to be working at Microsoft in Austin, Texas, describes that he "[d]rive[s] the use and consumption of Microsoft's … Identity products and services in highly regulated industries – Financial, Healthcare, Federal, and State and Local Government across the US as a part of Microsoft's new Security Solution Area within Customer Success United (CSU)." These data centers which Microsoft operates constitute a regular and established physical presence in the district, including, but not limited to, ownership of or control over property, inventory, or infrastructure.

---

[1] *See* https://www.datacenterhawk.com/providers/microsoft-azure (last accessed April 28, 2023).
[2] *See* https://www.virtualbx.com/industry-news/san-antonio-microsoft-reaches-mid-point-on-86m-expansion-in-westover-hills/ (last accessed April 28, 2023).

10. On information and belief, Microsoft has operated permanent office facilities within the State of Texas, and within this district, since at least 2000. The offices Microsoft maintains in this district include locations at 10900 Stonelake Boulevard, Suite 225, Austin, Texas 78759[3] and Concord Park II, 401 East Sonterra Boulevard, Suite 300, San Antonio, Texas 78258.

11. Microsoft operates offices in Austin, Texas for the purpose of selling, promoting, maintaining, and providing support for a suite of products, including the accused products.

12. On information and belief, Microsoft maintains a "Corporate Sales Offices" in Austin, Texas at the following address: 10900 Stonelake Boulevard, Suite 225 Austin, TX, 78759; and Microsoft maintains a "Corporate Sales Office" in San Antonio, Texas at the following address: Concord Park II 401 East Sonterra Boulevard, Suite 300, San Antonio, TX, 78258.

13. On information and belief, one or more of the Accused Products are used, offered for sale, and sold in this district, including by Microsoft and by "Microsoft-certified resellers" (e.g., Heart of Texas Network Consultants, located at 703 Willow Grove Rd., Waco, Texas 76712).

14. On information and belief, Microsoft operated at least ten physical stores throughout Texas, some of which were in this district, from 2009 until they were all closed in 2020. During that time period, Microsoft had physical stores that sold Microsoft's products at least at the following addresses: (a) 3309 Esperanza Crossing, Austin, TX 78758 and (b) 15900 La Cantera Parkway The Shops at La Cantera San Antonio, TX 78256.

---

[3] *See* https://news.microsoft.com/2000/01/05/microsoft-opens-austin-texas-facility/ (last accessed April 28, 2023).

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

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

17. In other recent actions, Microsoft has either admitted or not contested that this federal judicial district is a proper venue for patent infringement actions against it. *See*, *e.g.*, *Thompson v. Microsoft Corp.*, No. 1:19-cv-00680-RP, Dkt. No. 6; *Panther Innovations v. Microsoft Corp.*, No. 6-20-cv-01071, Dkt. No. 14; *Exafer Ltd. v. Microsoft Corp.*, No 1-20-cv-00131, Dkt. No. 15; *WSOU Investments, LLC v. Microsoft Corp.*, No. 20-cv-00464, Dkt. No. 20; *Zeroclick, LLC v. Microsoft Corp*., No. 20-cv-00272, Dkt. No. 14; and *California Institute of Technology v. Microsoft Corp.*, No. 21-cv-00276, Dkt. No. 22.

18. Venue is proper in the Western District of Texas pursuant to 28 U.S.C. §§ 1391 and 1400(b) because Microsoft maintains regular and established places of business in this district and has committed acts of infringement within this district giving rise to this action.

19. Microsoft has committed acts of infringement in this District and does business in this District, including making sales and/or providing service and support for customers and/or end-users in this District. Microsoft purposefully and voluntarily sold one or more infringing products with the expectation they would be purchased in this District. These infringing products have been and continue to be purchased in this District. Thus, Microsoft has committed acts of infringement within the United States, the State of Texas, and this District.

20. Furthermore, Microsoft maintains corporate sales offices in this district, which, on information and belief, provide sales and support for the infringing products.

PATENTS-IN-SUIT

21. 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**.

22. 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**.

23. 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**.

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

25. 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**.

26. 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**.

27. 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.

28. The technologies of the Patents-in-Suit were invented by John Giobbi and, for some of the patents, David Brown. The 730, 954, 905, and 989 Patents generally cover systems, devices, 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 cover 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

29. 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 growing number of devices and services that people use on a regular basis and that require regular authentication, and the rise of biometric readers and high-speed networks, there was a need to implement improved

authentication architectures. The Username and password method is not an ideal authentication mechanisms because people tend to pick low security passwords that are easy to remember, they tend to reuse the same password across multiple devices and services, they tend not the change the password, and they sometimes write the password down on paper or type out the password in public where their device screen can be seen by those around them. This method is also time consuming and inconvenient from a user experience perspective, especially when a user needs to try several passwords or needs to type out a high security (long and complex) password on the small keyboard of a portable device. For all of these reasons and many more, a better architecture is one that relies less on the users' memory.

30. 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. This method relies less on user memory because they need to use their username and password significantly less often.

31. 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.

32. 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.

33. Attempting to eliminate the use of passwords, Microsoft has developed a universal platform "password-less" architecture. The architecture is universal in that it works across platforms, such as iOS, Android, Windows, and Xbox. It is password-less in that passwords have been replaced with the use of authenticators approved or provided by Microsoft. Incorporating OpenID Connect, Microsoft's architecture relies on the issuance of security tokens. The hub of Microsoft's universal platform password-less architecture is Microsoft Identity Platform, which is the evolution of Azure Active Directory. The Microsoft Identity

Platform receives authentication requests from external systems, coordinates the action of authenticators, and issues security tokens.

II. PROXENSE AND ITS INNOVATIVE TECHNOLOGIES

34. Proxense was founded in 2001.[4] 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.

35. 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.

36. 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

[4] 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.

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

37. 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 7**. 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), **Exhibit 8**.

38. 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), **Exhibit 9**. 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…"

39. It would be years until products like Azure Active Directory and Microsoft Identity launched, and accordingly Proxense's technology was years ahead of the industry.

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

III. INFRINGEMENT ALLEGATIONS

1. Proxense's Interactions with Microsoft

41. In 2010, Proxense engaged in discussions with Microsoft for the purpose of potentially integrating Proxense's proprietary secure authentication technology utilizing biometric authentication into Microsoft products.

42. On July 29, 2016, counsel for Proxense sent a letter to Microsoft, advising it as to Proxense's "over 30 patents" included as an attachment, including the Patents-in-Suit, and further advising of "another 20+ US patent applications pending." A copy of the letter and list of patents attached thereto is attached as **Exhibit 10**.

43. Since at least that time, *i.e.* on or about July 29, 2016, Microsoft has had actual notice of the Patents-in-Suit and the scope of their claims as of at least their dates of issue.

44. Microsoft has also had knowledge of the infringing nature of its activities, or at least a willful blindness regarding the infringing nature of its activities, since at least Proxense's making Microsoft aware of the Patents-in-Suit as early as 2016, but at least as of the public filing of this Complaint. This follows where Proxense included with its July 29, 2016 correspondence a written comparison between Proxense's claimed inventions and Microsoft's products, including details of Microsoft's infringing activity. Microsoft was also aware of Proxense's proprietary

technology at least as early as 2010, when Proxense disclosed details of these technologies during discussions with Microsoft at that time.

45. Despite Microsoft's knowledge of the Patents-in-Suit, detailed knowledge of Proxense's proprietary technology, and Microsoft's knowledge that it infringed the Patents-in-Suit, or, at the very least a willful blindness to the fact that Microsoft infringes Proxense's patents, Microsoft continued to infringe the claims of the Patents-in-Suit. Microsoft's infringement has been and continues to be willful since at least 2016. Microsoft released the Azure Active Directory platform and its evolution into the Microsoft Identity platform with the intent they would be used to infringe the Patents-in-Suit.

2. The Accused Products

46. Through its own actions, and the actions of its customers and user, which Microsoft directs and controls, Microsoft 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 (literally or via the doctrine of equivalents) the Patents-in-Suit.

47. Three primary components make up the infringing architecture. The first is the Microsoft Identity Platform (also known as Azure Active Directory), which coordinates the actions of the other components by authenticating users and issuing various bearer tokens. The second is an authenticator permitting user verification by Microsoft Identity Platform. During verification, Microsoft Identity Platform issues commands to the authenticator (called "requests"). Operation of the authenticator, accordingly, is controlled by Microsoft Identity Platform. 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

Microsoft and in a form dictated by Microsoft. The resource then listens for a reply at a callback URL the resource registered with Microsoft. As with the requests, the callbacks are in the form dictated by Microsoft. The resources are hosted by devices and/or servers separate from Microsoft Identity Platform. Accordingly, the resource and Microsoft Identity Platform are separate and distinct entities and the resource, which is not Microsoft Identity Platform, is the system being accessed.

48. Microsoft utilizes the Microsoft Identity Platform to sign users into Windows 10/11, Xbox consoles, Xbox Game Pass, Office 365, Microsoft Family, and other services and products offered by Microsoft. Microsoft sells access to Microsoft Identity Platform to developers, websites, and corporate clients through various subscriptions. Thus, for a fee, Microsoft Identity Platform 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, Microsoft directly and/or indirectly infringes (and literally or under the doctrine of equivalents) the Patents-in-Suit. Furthermore, by utilizing Microsoft Identity Platform for its own product and services, Microsoft directly and/or indirectly infringes (and literally or via the doctrine of equivalents) the Patents-in-Suit.

49. As noted above, Microsoft's infringing universal platform password-less architecture includes an authenticator. One authenticator distributed by Microsoft is Windows Hello, a native component of Windows 10 and 11. Microsoft has distributed variants of Windows Hello that have included functionality to verify a user during authentication of a Windows-compatible device. Since at least November 20, 2018, if not earlier, Windows Hello has enabled password-less sign-in to services and subscriptions offered by Microsoft. *See* **Exhibit 11**. Another authenticator distributed by Microsoft is the Microsoft Authenticator App. Microsoft has

distributed variants of the Microsoft Authenticator App that have included functionality to verify a user during authentication of an Android or iOS compatible mobile device. The current and previous versions of Windows Hello and Microsoft Authenticator, along with devices with them, alone and together, are non-limiting instances of authenticators integrated into the Accused Products.

50. In addition to authenticators developed by Microsoft, authenticators developed by Microsoft's partners may also be incorporated into the Accused Products. To receive the benefit of having their authenticator work on Windows, Microsoft Edge browser, and online Microsoft Accounts, the authenticator must meet the requirements set by Microsoft, including review by Microsoft's engineering team. *See* **Exhibit 12**. By submitting the request for approval, an authenticator developer also enters into a contract with Microsoft in the form of its "Terms of Use," which give Microsoft control over both the terms (which can be changed by Microsoft at any time with no prior notice) and gives Microsoft control over the means by which any such authenticator operates. By setting such requirements, Microsoft directs and controls the actions of its partners.

51. The Microsoft Identity Platform, which directs and controls the actions of the authenticator, is also an element of Accused Product. Microsoft operates and maintains the Microsoft Identity platform, an evolution of the Azure Active Directory platform. When combined with authenticators, such as Windows Hello, Microsoft Authenticator App, and others developed and sold by Microsoft's partners, consumers of Microsoft's products and services receive the benefit of password-less biometric authentication across platforms. For instance, a user may log into and utilize Microsoft Office or their Xbox subscription on their Android phone. Third party developers can purchase identity and access management services from Microsoft to integrate their

applications with Microsoft 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 Microsoft Identity Platform must register their application or website with Microsoft, request authentication by sending a request to a URL provided by Microsoft and in a format dictated by Microsoft, and listen for a callback provided by Microsoft that contains a message generated by Microsoft in a format controlled by Microsoft. To facilitate such controlled interactions across platforms, Microsoft has developed Microsoft Authentication Library (MSAL). *See* **Exhibit 13**.

52. The Accused Products also include a resource accessed following successful authentication of the authenticator by Microsoft Identity Platform. As noted above, the resource may be an application, website, and/or a subscription offered by Microsoft, a subscribing business, or a subscribing developer. When a user has been authenticated via an authenticator offered by Microsoft or one of its partners, various bearer tokens are returned by Microsoft Identity Platform to the callback URL registered with Microsoft Identity Platform. 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 Microsoft or a subscribing developer or business, the resource is on a system separate and distinct from Microsoft Identity Platform. Distribution of bearer tokens and other such access messages by Microsoft Identity Platform is thus necessary for Microsoft Identity Platform to inform the resource that the user has been successfully authenticated via an authenticator distributed by Microsoft or one of its partners. As the bearer tokens are generated and distributed by Microsoft Identity Platform, Microsoft controls their form and how they are distributed.

53. Through the operation of the foregoing as directed and controlled by Microsoft, the Accused Products practice the claims of the Patents-in-Suit to improve the user experience of Microsoft's customers and those of subscribing developers, and to improve Microsoft's position in the market with respect to Identity and Access Management, operating as an identity provider, along with providing access to other products, services, and subscriptions.

3. Microsoft's Direct Infringement of the Patents-in-Suit

54. Microsoft directly infringes the Patents-in-Suit by creating and utilizing a universal platform password-less architecture incorporating authenticators, including Windows Hello on Windows 10 and 11, Microsoft Authenticator App on Android and iOS devices, and controlling the action of its partners to create authenticators that work on Windows, the Microsoft Edge browser, and online Microsoft accounts, incorporating the Microsoft Identity platform which controls the actions of authenticators and the dissemination of bear 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 Microsoft Identity Platform, 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 Microsoft, including the use of Microsoft Authentication Library.

55. Microsoft's password-less architecture verifies a user during authentication of an integrated device. This device may be a computer running Windows 10/11, which includes Windows Hello as a native component. It may also be an Android or iOS device with the Microsoft Authenticator App. Alternatively, it may be an authenticator developed by a partner in accordance with requirements set by Microsoft. Each uses biometrics, such as facial recognition or fingerprints, to verify the user. *See, e.g.*, **Exhibit 14**

56. Following user verification via biometrics, the Microsoft Identity platform utilizes FIDO2 and analogous protocols to authenticate the device and OpenID Connect to permit access to various resources, including Microsoft applications, subscriptions, and services, such as Outlook, Office, Skype, Xbox Live, etc. and those developed by businesses and developers subscribing to Microsoft's Identity and Access Management services. *See* **Exhibit 11.**

57. Windows devices, such as the Microsoft Surface and biometrically enabled laptops produced by Microsoft's partners, persistently store biometric user data. Whether manufactured by Microsoft or its partners, such as Dell, Windows 10 and 11 devices must meet minimum hardware requirements set by Microsoft. *See* **Exhibit 15**. The minimum hardware requirements ensure Windows Hello can utilize specialized hardware and software, such as Virtualization Based Security (VBS) and Trusted Platform Module 2.0 to isolate, protect, and secure the channel by which a user's biometric data is communicated. *See* **Exhibit 16.**

58. For example, when a user elects to use facial recognition for authentication, a face template is generated and encrypted using keys only accessible to the VBS, and then stored on disk. *See id.* Accordingly, facial templates are persistently stored on a Windows device's storage. Likewise, these devices persistently store fingerprint data, but do so instead in the sensor's dedicated memory. *See id.* In all cases, the biometric data is stored so as to prevent unauthorized alterations. This meets the first patented claim limitation of the 730 patent literally or, at least, by the doctrine of equivalents.

59. Android devices with the Microsoft Authenticator App also 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 17.** 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 29**. 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. Furthermore, access to the biometric hardware on Android devices is controlled by Fingerprint HIDL. *Id.*. The methods enabled by the Fingerprint HIDL do not permit altering biometric data. *See id.*

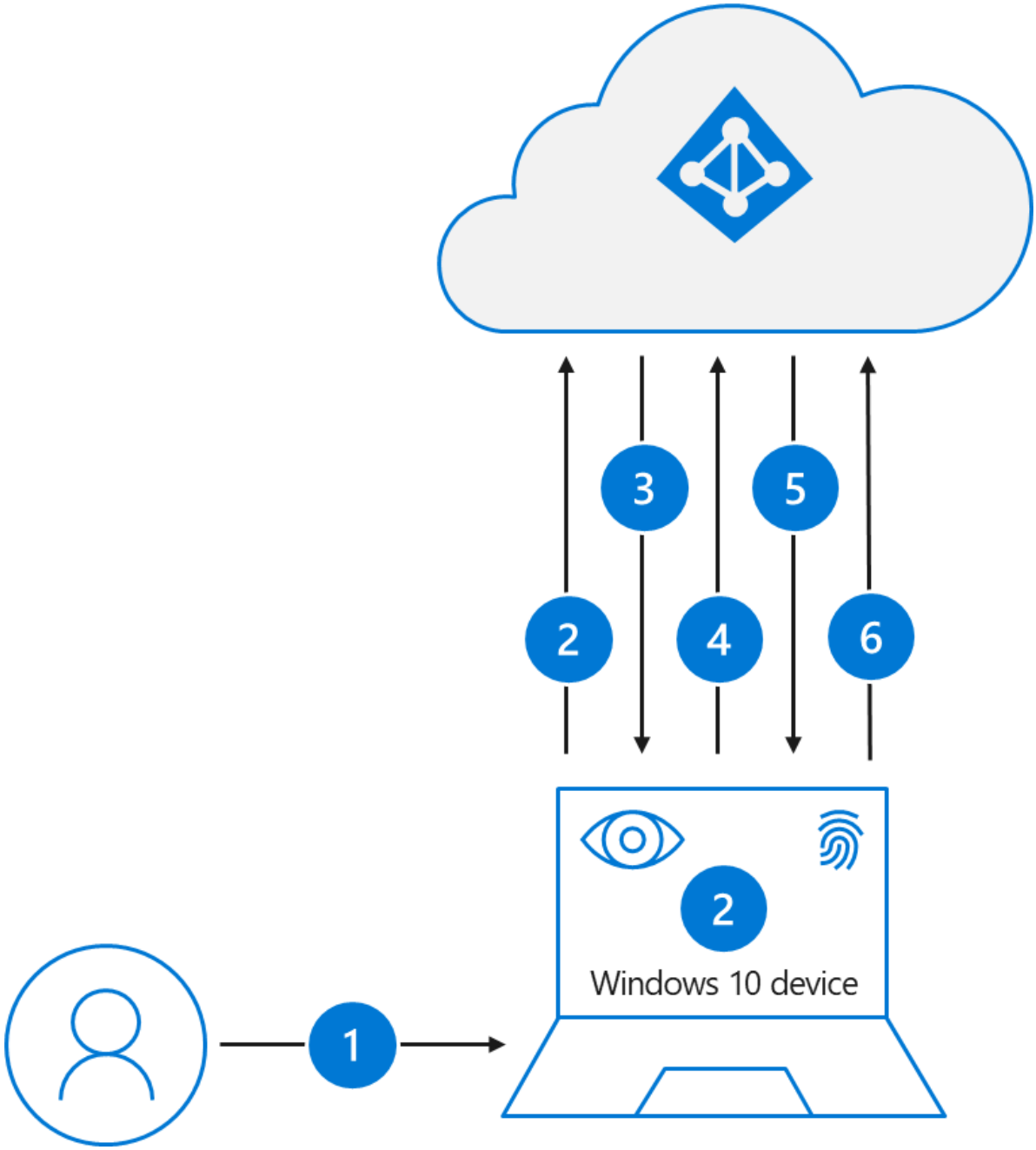
60. iOS devices with the Microsoft Authenticator App also persistently store biometric data in a tamper proof format. Apple's "Secure Enclave is a dedicated secure subsystem … isolated from the main processor to provide an extra layer of security designed to keep sensitive data secure even when the Application Processor kernel becomes compromised." *See* **Exhibit 18**. "During enrollment, the Secure Enclave processes, encrypts, and stores the corresponding Touch ID and Face ID template data." *Id.*. The Secure Enclave thus provides a tamper proof format for sensitive data, such as biometric data, even when a hack or other malicious software compromises the Application Processor.

61. Authenticators developed by Microsoft's partners also store biometric data in a tamper proof format. To receive the benefit of having their authenticator work on Windows, Microsoft Edge browser, and online Microsoft Accounts, a partner's authenticator must meet the requirements set by Microsoft, including being FIDO2 certified. *See* **Exhibit 19**. FIDO compliant authenticators store biometric data of user in tamper proof format unable to be subsequently

altered. Microsoft, accordingly, directs partner authenticators to store biometric data in a tamper proof format. Authenticators developed by Microsoft and its partners persistently store a device ID code uniquely identifying each integrated device. Microsoft's universal platform password-less architecture is based on FIDO2. *See* **Exhibit 16**. Instead of passwords, FIDO2 uses public/private key encryption. *See* **Exhibit 11**. The private key is generated and stored on the authenticator, while the public key is sent to the Microsoft Identity platform. *See id.* When a user attempts to authenticate, Microsoft Identity platform sends a nonce to the authenticator, which is signed with the private key. *See id.* The signed nonce is then returned to the Microsoft Identity Platform and the signature verified with the corresponding public key. For this to work, Microsoft Identity Platform must select the correct public key for the particular authenticator being used. This is accomplished by sending a credential ID indicating which public key to use along with the signed nonce. Accordingly, each public key within Microsoft Identity includes a reference to a credential ID uniquely identifying the device from which it was created. *See* **Exhibit 21**. The credential ID is thus a device ID code that is part of a pair. Being one part of a pair places the Device ID in a tamper proof format on the integrated device.

62. In order to perform biometric verification of the user, authenticators utilized within Microsoft's universal platform password-less architecture causes the device to prompt a user for biometric verification and receive scan data from a biometric scan, which varies depending on how and what the user is logging into.

63. When Windows Hello is used as the authenticator, the prompt for biometric verification occurs after the user dismisses the lock screen, as shown below.

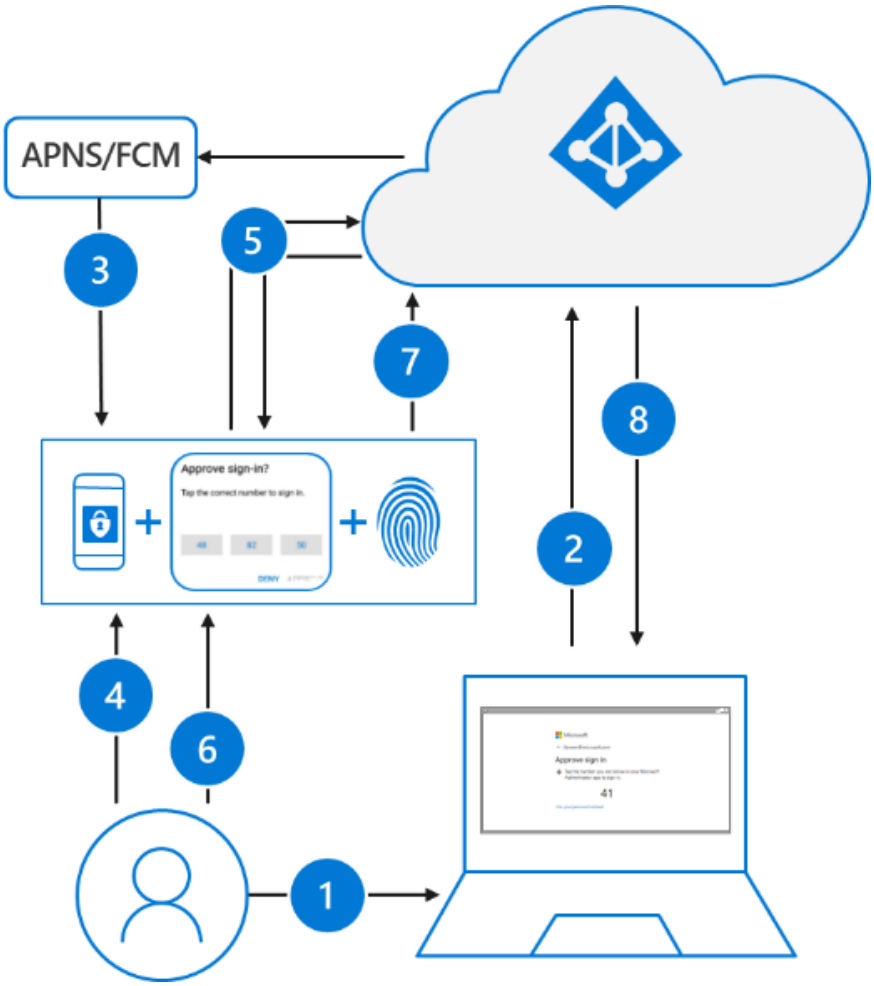


1. A user signs into Windows using biometric or PIN gesture. The gesture unlocks the Windows Hello for Business private key and is sent to the Cloud Authentication security support provider, referred to as the *Cloud AP provider*.
2. The Cloud AP provider requests a nonce (a random arbitrary number that can be used just once) from Azure AD.
3. Azure AD returns a nonce that's valid for 5 minutes.
4. The Cloud AP provider signs the nonce using the user's private key and returns the signed nonce to the Azure AD.

5. Azure AD validates the signed nonce using the user's securely registered public key against the nonce signature. After validating the signature, Azure AD then validates the returned signed nonce. When the nonce is validated, Azure AD creates a primary refresh token (PRT) with session key that is encrypted to the device's transport key and returns it to the Cloud AP provider.
6. The Cloud AP provider receives the encrypted PRT with session key. Using the device's private transport key, the Cloud AP provider decrypts the session key and protects the session key using the device's Trusted Platform Module (TPM).
7. The Cloud AP provider returns a successful authentication response to Windows. The user is then able to access Windows as well as cloud and on-premises applications without the need to authenticate again (SSO).

See **Exhibit 22**.

64. When Microsoft Authenticator App is used as the authenticator, the biometric verification prompt occurs later in the process, as shown below.



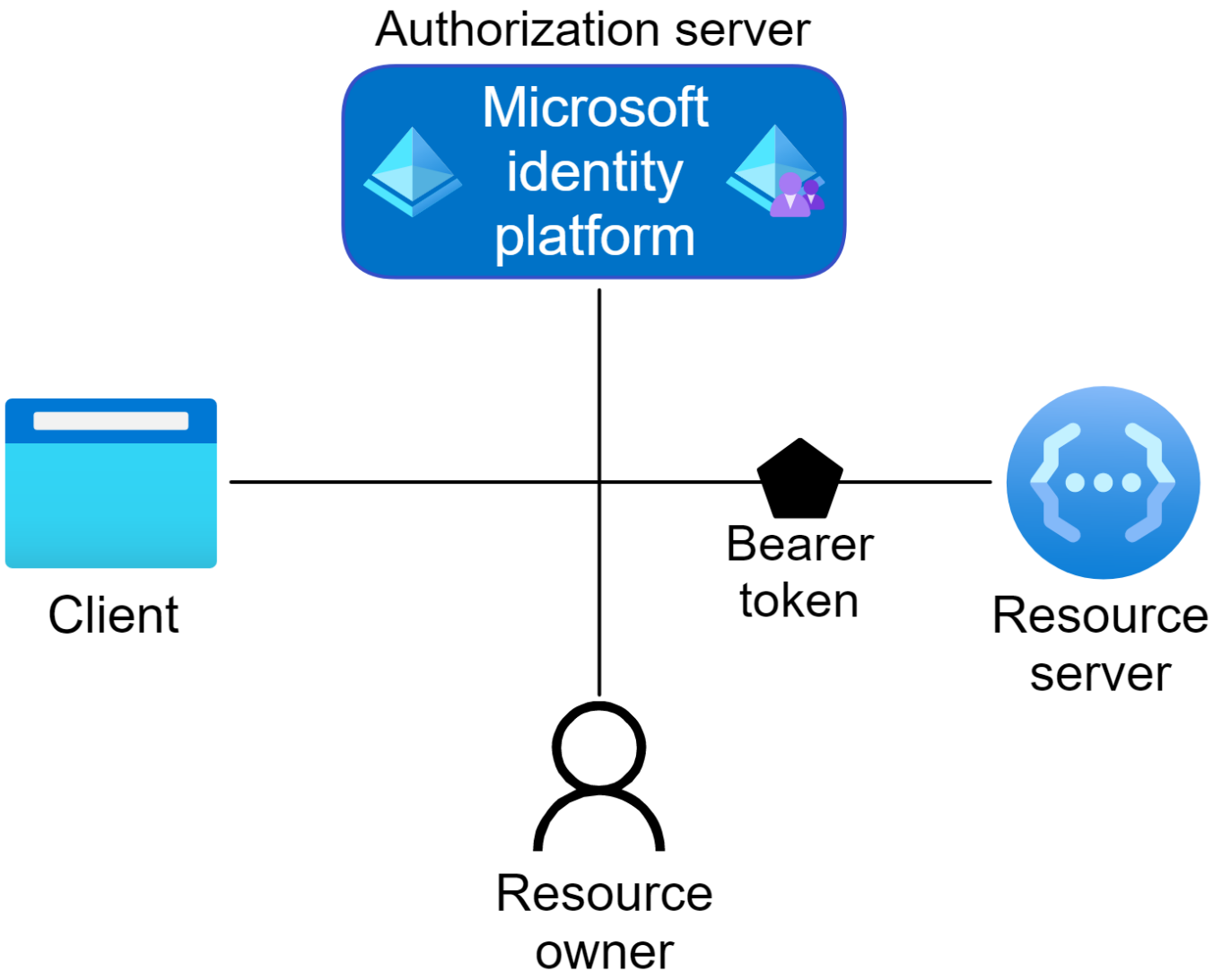
1. The user enters their username.
2. Azure AD detects that the user has a strong credential and starts the Strong Credential flow.
3. A notification is sent to the app via Apple Push Notification Service (APNS) on iOS devices, or via Firebase Cloud Messaging (FCM) on Android devices.
4. The user receives the push notification and opens the app.
5. The app calls Azure AD and receives a proof-of-presence challenge and nonce.

See id. Specifically, at Step 3 "a notification is sent to the app via Apple Push Notification Service (APNS) on iOS devices, or via Firebase Cloud Messaging (FCM) on Android devices." At Step 4, "the user receives the push notification and opens the app." Microsoft prevents the use of the Microsoft Authenticator App on devices without an active security lock, which may be released with biometrics. *See* **Exhibit 23**. Consequently, before opening the App to complete Step 4, the user unlocks the device with biometrics. If the user leaves their device unlocked, App Lock will prompt for a biometric gesture. *See* **Exhibit 24.** Accordingly, regardless of whether the device is locked or unlocked, responding to the notification requires that the device receives data from a biometric scan.

65. The use of Windows Hello, the Microsoft Authenticator App, and partner authenticators within the Microsoft's universal platform password-less architecture is not limited to logging onto a Windows 10/11 computer. For instance, resources, such as applications, services, subscriptions, and websites utilizing the Microsoft Identity Platform for Identity and Access Management natively receive the benefit of utilizing authenticators for password-less user authentication. *See* **Exhibit 25**. As such, the Microsoft Identity Platform permits the use of Microsoft approved authenticators to log into any platform or browser, or confirm any login, with the use of biometric authentication. *See* **Exhibit 18**. Furthermore, developers may utilize Windows Hello on Windows 10/11 computers to protect their Universal Windows Platform (UWP) applications and backend services. *See* **Exhibit 26**. Accordingly, the

request for biometric authentication and the corresponding receipt of biometric scan data occurs when the user logs into any application, device, or service utilizing Microsoft Identity platform for Identity and Access Management.

66. Microsoft Identity platform operates as a third party trusted authority possessing a list of device ID codes uniquely identifying legitimate integrated devices (i.e., Microsoft approved authenticators). Microsoft depicts the relationship between the parties involved in authentication and authorization below.



See **Exhibit 27.**

67. Microsoft Identity takes on the role of the authorization server responsible for authenticating the user. *See id.* Instead of using passwords, it utilizes FIDO2 and analogous protocols to authenticate users. *See* **Exhibit 23.**

68. The protocols employed use public/private key encryption. *See* **Exhibit 11**. Each public key is identified by a unique credential ID. As discussed above, the private key is generated and stored on the authenticator and the public key is sent to Microsoft Identity platform. When a user attempts to authenticate, Microsoft Identity platform sends a nonce to the authenticator, which is signed with the authenticator's private key, returned to Microsoft Identity with a credential ID, and verified with the public key held by Microsoft Identity Platform corresponding to the sent credential ID. *See id.* Sending the credential ID with the signed nonce enables Microsoft Identity Platform to select the correct public key for the particular authenticator being used. A user account, therefore, contains a list of "PassportDevices," (i.e., authenticators) which each entry in the list including a "DeviceId" (i.e., credential ID) and a "PublicKey." *See* **Exhibit 42**. Accordingly, each public key within Microsoft Identity includes a reference to the specific device from which it was created and should be used. *See* **Exhibit 21**.

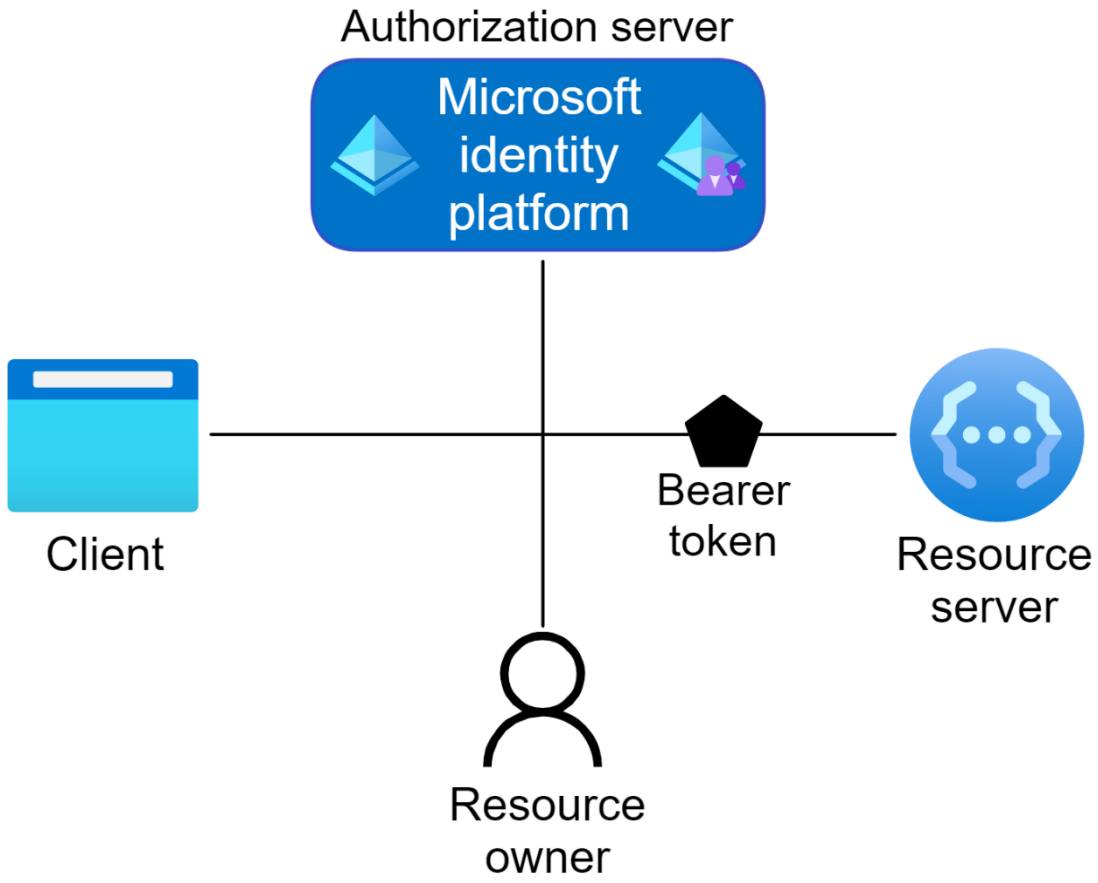
69. Authenticators within Microsoft's universal platform password-less architecture sign the nonce and return it with the credential ID after a determining that the scan data matches the biometric data. As the Microsoft Authenticator App is downloaded to mobile devices, the nonce request (including the device ID codes) is necessarily sent wirelessly. Likewise, computers running Windows 10/11 can wirelessly send the device ID code and nonce request via a wireless connection to a local router or mobile phone.

70. Utilizing OAuth 2.0 and OpenID Connect, Microsoft Identity platform issues access messages in the form of Bearer Tokens to various resources. *See* **Exhibit 36**. Acting as an

authorization server, Microsoft Identity handles the trust relationship between the parties, including issuing security tokens (i.e., Bearer Tokens) for granting access (authorization) after the user has signed in (authenticated). *See* **Exhibit 21**. The Bearer Token is passed between the parties to assure authentication and grant access. There are four types of tokens issued by Microsoft Identity: Access Tokens, ID Tokens, Refresh Tokens, and Primary Refresh Tokens. *See id.* The type of token issued depends on the resource being accessed. Regardless of the resource, the Bearer Tokens received from Microsoft Identity are used to get access to the resource, which may be an application, website, service, subscription, etc. offered by Microsoft or businesses and developers subscribing to Microsoft's Identity and Access Management service and are thus "access messages."

71. The issuance of tokens serving as access messages within Microsoft's universal platform password-less architecture is shown in the figure below. As shown, the resource server providing the application, subscription, service, etc. to be accessed is a separate entity from Microsoft Identity Platform. Thus, regardless of whether the resource is one provided by Microsoft or its customers, the resource server is a separate entity. As such, communication between the resource server and Microsoft Identity Platform is required. This communication is accomplished by the resource server sending a request for authentication using URLs provided by Microsoft and in a form dictated by Microsoft. In many instances, the communication will be mediated via a client, such Microsoft Edge or another web browser, Windows Cloud AP, etc. Regardless of the client, upon receipt of the request form the resource server for user authentication, Microsoft Identity Platform sends requests to a Microsoft approved authenticator to verify the user and sign a nonce. Often the client will facilitate this communication between the Microsoft Identity Platform and the authenticator. After receiving and verifying the signature

generated by the authenticator, Microsoft Identity Platform issues a bearer token. Receipt of the bearer token by the resource server indicates user authentication by Microsoft Identity Platform acting as a third-party with respect to the resource server.



72. As the foregoing shows, Microsoft Identity Platform is at the center of Microsoft's universal platform password-less architecture. It receives messages sent by resource servers in a manner prescribed by Microsoft. It directs the action of Microsoft approved authenticators, and it issues bearer tokens in a manner Microsoft chooses. Accordingly, Microsoft directs and controls the actions of developers and partners wishing to receive the benefit of utilizing or being part of the Accused Product.

73. 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.

4. Microsoft's Indirect Infringement of the Patents-in-Suit

74. Microsoft 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, Microsoft 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. And Microsoft directly profits from its indirect infringement.

75. The knowledge base includes advertising for the infringing features of its password-less architecture. On its website, Microsoft touts that its customers can "securely sign in with your Microsoft account" with "no username or password required" and that authentication occurs through comparing biometric data such as a fingerprint or facial scan. *See, e.g.*, **Exhibit 11**. In making these statements, Microsoft is advertising the functionality claimed by the Patents-in-Suit. These are just a few of examples of an extensive advertising campaign to make the consumers knowledgeable of the infringing components, functionality, and uses of Microsoft's universal platform password-less architecture.

76. In addition to educating consumers on how to actively engage in infringing uses, Microsoft is actively creating a knowledge base among developers on how to integrate their resources to exploit Microsoft's universal platform password-less architecture. For instance, Microsoft teaches developers how to utilize applications that take advantage of authentication

methods such as the use of biometrics and the use of ID and access tokens. *See, e.g.*, **Exhibit 27**. By teaching developers how to integrate their resources into its password-less architecture, Microsoft is directing third parties to perform the infringing uses of Microsoft Identity.

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

78. Microsoft also actively contributes to infringement of the Patents-in-Suit by providing the authenticators Windows Hello and Microsoft Authenticator App to allow users to incorporate their devices into Microsoft's universal platform password-less architecture, as these have no substantial non-infringing use, and are especially made for such infringement.

79. As noted *supra* with regards to direct infringement, Windows Hello and Microsoft Authenticator App are components of the Accused Products. Accordingly, they are especially made for infringement of the Patents-in-Suit. Furthermore, Windows Hello and Microsoft Authenticator App are software, *i.e.*, a mobile application installed on a smartphone and/or computer. Software, when installed, becomes a component of the device it is installed upon. Accordingly, Windows Hello and Microsoft Authenticator App (when installed on a smartphone and/or device) integrates a user's device into the Accused Product. These software applications are thus especially made for making devices and smartphones into components of the Accused Product, and to practice the methods of the Patents-in-Suit. Moreover, Microsoft includes Windows Hello as a native component of Windows 10/11 installed on Windows compatible devices by Microsoft's OEM partners.

80. At all relevant times, since at least July 2016 (when Microsoft was given notice of all of the limitations of the claims of the Patents-in-Suit that are at issue) to the present, Microsoft

has had full knowledge of the specifications upon which Microsoft relies to develop and maintain its universal platform password-less architecture, the underlying infrastructure (including Microsoft Identity, Microsoft approved authenticators, and resources provided by Microsoft or subscribing developers and business), and all third parties who are in privity with Microsoft. Microsoft did review the relevant specifications and their internal documents in light of, and considering, the claims of the Patents-in-Suit from Proxense's letter; alternatively, Microsoft remained willfully blind to its own infringement by failing to review the relevant specifications and its own internal documents in light of, and considering, the claims of the Patents-in-Suit from Proxense's letter. By publicly releasing instructions, guides, and advertising with knowledge of, or willful blindness to, the fact that those acts will lead to direct infringement by their users and third-party developers, Microsoft has induced and is inducing those parties to participate in Microsoft's infringement.

CLAIM 1
(Infringement of the 730 Patent)

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

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

83. Microsoft 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.

84. For example, Microsoft 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 Microsoft's universal platform password-less architecture

incorporating the Microsoft Authenticator App and Windows Hello as authenticators. Under the coordination of Microsoft Identity Platform, the authenticators perform/execute and provide, a method for verifying a user during authentication of the device. Microsoft 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 Microsoft's universal platform password-less architecture incorporating Microsoft Identity platform. The coordination and control provided by Microsoft Identity Platform of the other components within the architecture provides a system for verifying a user during authentication of a device.

85. As described *supra*, the Microsoft Authenticator App "turns any iOS or Android phone into a strong, passwordless credential." Utilizing the Android operating system, Android Phones having the Microsoft Authenticator App 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. Starting with Android 10, stored biometric data on Android phones includes a fingerprint and facial recognition. *See* **Exhibit 22**.

86. As to protect biometric data, Android's implementation guidelines require tamper-proof "raw fingerprint data or derivatives (for example, templates) 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 17**. Requiring acquisition and recognition to occur inside the TEE, fingerprint data never leaves the TEE. Android's TEE, called Trusty, "uses ARM's Trustzone™ to virtualize the main processor and create a secure trusted execution environment" isolated from the rest of the system. 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.

87. On Android phones, access to the biometric hardware is controlled by Fingerprint HIDL. The methods enabled by the Fingerprint HIDL do not permit altering 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, means that Android phones 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.

88. When the Authenticator App is installed on an iOS device, biometric data would be stored within Apple's Secure Enclave. *See* **Exhibit 30**. Designed to keep sensitive data secure even when compromised, the Secure Enclave provides a tamper proof format for sensitive data.

89. Additionally, both iOS and Android phones require user consent to enroll a fingerprint. *See* **Exhibits 31** and **32.** As enrolling fingerprints on both iPhones and Android phones require entering PIN / Passcode / Password to evidence user consent, iPhones and Android phones store biometric data of user in a tamper proof format unable to be subsequently altered.

90. Windows compatible systems meeting Microsoft's minimum hardware requirements also store data of a user in a tamper proof format unable to be subsequently altered.

91. Microsoft details that the Authenticator App uses key based authentication technology similar to Windows Hello. ("Microsoft Authenticator uses key-based authentication to enable a user credential that is tied to a device, where the device uses a PIN or biometric. Windows Hello for Business uses a similar technology." *See* **Exhibit 43**. Windows Hello is FIDO certified. The FIDO specification incorporates the WebAuthn Specification. *See* **Exhibit 33**. Windows Hello, therefore, uses a key-based authentication technology compliant with the

WebAuthn and FIDO specifications. Utilizing a similar technology, MS Authenticator most use a key-based authentication analogous to the WebAuthn and FIDO protocols.

92. Under the WebAuthn specification, "compliant authenticators protect public key credentials." *See* **Exhibit 34**. 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 (i.e., Microsoft Identity Platform). *Id.* Accordingly, a credential ID uniquely identifies a private/public key pair.

93. The credential ID is used to retrieve the correct public key during authentication ceremonies. *See* **Exhibit 34.** The public key retrieved is used to verify the signature generated with its matching private key held by phone running the Microsoft Authenticator App or Windows 10/11 device natively having Window Hello. *See* **Exhibit 34**. Being part of an asymmetric key pair, the pubic key corresponds to a private key.

94. Altering a Windows Hello or Microsoft Authenticator App credential ID causes authentication to fail. Being FIDO certified, Windows Hello utilizes the WebAuthn protocol. Credential IDs within the WebAuthn protocol are in a tamper proof format unable to be subsequently altered because "all that would happen if an authenticator returns the wrong credential ID, or if an attacker intercepts and manipulates the credential ID, is that the WebAuthn Relying Party would not look up the correct credential public key with which to verify the returned signed authenticator data (a.k.a., assertion), and thus the interaction would end in an error." *See* **Exhibit 34**. As altering the device ID (i.e., Credential ID) causes authentication to fail, the credential ID is necessarily in a tamper proof format unable to be altered. "Passwordless

authentication using the Authenticator app follows the same basic pattern as Windows Hello for Business." *See* **Exhibit 22**. Following the same pattern, altering the device ID stored on the phone by the Authenticator App should produce the same result of causing authentication to fail. As Microsoft controls the protocol utilized by its Identity Platform, including the inability to sync passkeys across devices, users wanting the benefit of biometric authentication via the Microsoft Authenticator App or Windows Hello have no choice but to store on their phone or Windows device a credential ID uniquely identifying the integrated device in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered.

95. As with the FIDO standards implemented by Windows Hello, the Microsoft Authenticator app utilizes a private key to sign a challenge. *See* **Exhibit 22**. Private keys are notoriously known secret decryption values. If the Authenticator App and Windows Hello have the private keys, the public keys must be held by Microsoft's Identity Platform. By being part of a pair split between the authenticator and Microsoft's Identity Platform, the private key secret decryption value is in a tamper proof format unable to be subsequently altered. As Microsoft controls the protocol utilized by its Identity Platform, including the use of public/private key validation, users wanting the benefit of biometric authentication via the Microsoft Authenticator App or Windows Hello have no choice but to store on their phone or Windows device a secret decryption value in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered.

96. Microsoft describes user authentication via the Authenticator App and Windows Hello as comprising receipt of a proof-of-presence challenge that is completed when the user enters their biometrics. *See* **Exhibit 22**. Entering their biometric necessarily requires receiving scan data from a biometric scan. When the user completes the challenge, the private key held by the phone

or Windows Device is unlocked. Determining if the user successfully completed the challenge necessarily requires comparing the scan data to the biometric data to determine whether the scan data matches the biometric data.

97. As detailed by Microsoft, after the user has completed the biometric challenge, a signature is generated with the authenticator's private key and sent to Microsoft Identity Platform for verification. *See* **Exhibit 22**. The Microsoft Authenticator App operates either on iOS or Android phones. As iOS and Android phones are cellular phones, often possessing Wi-Fi and Bluetooth capability, anything sent to Microsoft Identity Platform will be sent wirelessly – either via Wi-Fi, Bluetooth, or the cellular network. Laptops with Windows have biometric capabilities and are often connected to the internet via Wi-Fi or using a mobile phone as a hotspot. Accordingly, anything sent to Microsoft Identity Platform will be sent wirelessly via the laptop's Wi-Fi connection or connected phones cellular connection.

98. After the user has completed the biometric challenge, a signature is generated with the authenticator's private key and sent to Microsoft Identity Platform for verification. *See* **Exhibit 22**. As previously noted, password-less authentication using the Microsoft Authenticator App follows the same basic pattern as FIDO certified Windows Hello. FIDO incorporates the WebAuthn Protocol. *See* **Exhibit 35**. Accordingly, the challenge sent to the Authenticator App is the equivalent of a FIDO server challenge sent to the Windows Hello authenticator with a WebAuthn authenticatorGetAssertion request. *Id.* The response to the request, therefore, will contain a signature signed by the authenticator's private key and a credential ID (i.e., device ID) identifying which public key to use to verify the signature. *See* **Exhibit 34**. Upon receiving the response, the WebAuthn/FIDO server will use the device ID (i.e., credential ID) to locate the appropriate public key to verify the signature generated with the private key. *See* **Exhibit 34**.

Accordingly, a FIDO/WebAuthn within the Microsoft Identity Platform will authenticate a device ID (i.e., credential ID) sent in response to successful completion of a biometric challenge.

99. As detailed *supra*, upon successful completion of the biometric challenge, a device ID and signature is sent from the authenticator to Microsoft Identity Platform for authentication. The Microsoft Identity Platform utilizes OpenID Connect and Microsoft encourages developers to gain an understanding of the protocol and concepts to add authentication to applications. *See* **Exhibit 36**. Within the protocol, Microsoft identifies its identity platform as an "authorization server" managing "trust relationships" and issuing security tokens applications and APIs use for granting access (i.e. authorization) and authentication. *See* **Exhibit 27**. Accordingly, Microsoft describes its Identity Platform as a third party that operates as a trusted authority for authentication.

100. As noted *supra*, the FIDO server within the Microsoft Identity Platform uses a credential ID operating as a device ID to verify a signature generated with a private key bound to the identified device (i.e., authenticator). Such action is made possible by registering the credentials with an account and associating the account with the credential ID and credential public key. *See* **Exhibit 34**. Accordingly, the Microsoft Identity Platform maintains a listing of legitimate device IDs.

101. Authentication is provided by a FIDO server incorporated into the Microsoft Identity Platform in conjunction with OpenID Connect Authentication Server that is also incorporated into the Microsoft Identity Platform. While OpenID Connect is a federation protocol, it is compatible and complementary with FIDO. "The value of a FIDO authentication capability is amplified by a federated system, where the federation system extends the benefits of a FIDO authentication to applications and services without requiring FIDO to be directly integrated with those applications." *See* **Exhibit 37**. When a federated system, such as OpenID Connect, is

combined with FIDO, the OpenID Provider (OP) sends a FIDO server challenge to an authenticator which is returned as the FIDO Authentication response. *See* **Exhibit 37**.

102. FIDO incorporates the WebAuthn Protocol. Accordingly, the FIDO server challenge sent will a be WebAuthn authenticatorGetAssertion request. The response to the request received will contain a signature generated by the authenticator's private key and a credential ID. *See* **Exhibit 35**. Upon receiving the response, the WebAuthn/FIDO server will use the credential ID to locate the appropriate public key to verify the signature generated with the private key. *See* **Exhibit 34**. Accordingly, a FIDO/WebAuthn server incorporated into Microsoft Identity Platform will authenticate a device ID (i.e., credential ID). After validating the device ID, the FIDO/WebAuthn portion of the Microsoft Identity Platform "redirects the user agent back to the Application Provider with an authentication assertion". *See* **Exhibit 37**.

103. As the Microsoft Identity Platform utilizes OpenID Connect, the authentication assertion will be an ID Token. The ID Token received from Microsoft's Identity platform enables access to an application. The ID Token is thus an access message provided by Microsoft operating as a third party trusted authority allowing access to an application. Should the user also require an access token to access further features of the application, such as files containing stored data, the ID token can be exchanged for an access token. *See* **Exhibit 27**.

104. Further, Microsoft Family enables parents to "designate the age limit for content [a family member] will have permission to access", such as apps and games. *See* **Exhibit 38**.

105. Microsoft 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 as authenticators Windows Hello preinstalled on Windows 10/11 computers and the Microsoft Authenticator App available for

download and Microsoft Identity Platform for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Microsoft also induces infringement of claims 15, 16, and 17 by making Microsoft Identity available for integration with developer applications, providing the Microsoft Authentication Library, and substantial knowledge base teaching developers and business subscribing to Microsoft'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.

106. Microsoft 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 as authenticators Windows Hello preinstalled on Windows 10/11 computers and the Microsoft Authenticator App available for download and Microsoft Identity Platform for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Microsoft also induces infringement of claims 15, 16, and 17 by making Microsoft Identity available for integration with developer applications, providing the Microsoft Authentication Library, and a substantial knowledge base teaching developers and business subscribing to Microsoft'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.

107. Microsoft received actual notice of the 730 Patent at least as early as July 29, 2016 when Proxense sent Microsoft correspondence attaching a copy of the 730 Patent. Microsoft performed and continues to perform the acts that constitute direct and/or indirect infringement,

with knowledge or willful blindness to the acts that constitute direct and/or indirect infringement of the 730 Patent.

108. Since at least July 29, 2016, through its actions and continued actions, Microsoft has indirectly infringed and continues to indirectly infringe the 730 Patent in violation of 35 U.S.C. § 271(b). Microsoft has actively induced product makers, distributors, retailers, 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 Microsoft promoting the use of the Accused Product are packaging Windows Hello with Windows 10/11 and public documents, which serve no function other than to direct users of the Accused Products toward infringing the 730 Patent.

109. Microsoft does so knowingly and intending that its customers and end users will commit these infringing acts. Microsoft 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.

110. In addition, Microsoft 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.

111. For example, Microsoft 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 730 Patent, including claim 1. Microsoft continues to sell and offer to sell these products in the United States after receiving notice of the 730 Patent and how its products infringe that patent.

112. 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 non-infringing use.

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

114. Upon information and belief, Microsoft 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 Microsoft.

CLAIM 2
(Infringement of 954 Patent)

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

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

117. Microsoft 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.

118. For example, Microsoft 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 the Microsoft Authenticator App and Windows Hello as authenticators. Under the coordination of Microsoft Identity Platform, the authenticators perform/execute and provide a method for verifying a user during authentication of the device. Microsoft 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 the Microsoft's universal platform password-less architecture incorporating Microsoft Identity platform. The coordination and control provided by Microsoft Identity Platform of the other components within the architecture provides a system for verifying a user during authentication of a device.

119. As described *supra*, the Microsoft Authenticator App "turns any iOS or Android phone into a strong, passwordless credential." Utilizing the Android operating system, Android Phones having the Microsoft Authenticator App 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. Starting with Android 10, stored biometric data on Android phones includes a fingerprint and facial recognition. *See* **Exhibit 22**.

120. As to protect biometric data, Android's implementation guidelines require tamper-proof "raw fingerprint data or derivatives (for example, templates) 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 17**. Requiring acquisition and recognition to occur inside the TEE, fingerprint data never leaves the TEE. Android's TEE, called Trusty, "uses ARM's Trustzone™ to virtualize the main processor and create a secure trusted execution environment" isolated from the rest of the system. 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.

121. On Android phones, access to the biometric hardware is controlled by Fingerprint HIDL. The methods enabled by the Fingerprint HIDL do not permit altering 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, means that Android phones 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.

122. When the Authenticator App is installed on an iOS device, biometric data would be stored within Apple's Secure Enclave. *See* **Exhibit 30**. Designed to keep sensitive data secure even when compromised, the Secure Enclave provides a tamper proof format for sensitive data.

123. Additionally, both iOS and Android phones require user consent to enroll a fingerprint. *See* **Exhibits 31** and **32**. As enrolling fingerprints on both iPhones and Android phones require entering PIN / Passcode / Password to evidence user consent, iPhones and Android phones store biometric data of user in a tamper proof format unable to be subsequently altered.

124. Windows compatible systems meeting Microsoft's minimum hardware requirements also store data of a user in a tamper proof format unable to be subsequently altered.

125. Microsoft details that the Authenticator App uses key based authentication technology similar to Windows Hello. ("Microsoft Authenticator uses key-based authentication to enable a user credential that is tied to a device, where the device uses a PIN or biometric. Windows Hello for Business uses a similar technology." *See* **Exhibit 43**. Windows Hello is FIDO certified. The FIDO specification incorporates the WebAuthn Specification. *See* **Exhibit 33**. Windows Hello, therefore, uses a key-based authentication technology compliant with the WebAuthn and FIDO specifications. Utilizing a similar technology, MS Authenticator most use a key-based authentication analogous to the WebAuthn and FIDO protocols.

126. Under the WebAuthn specification, "compliant authenticators protect public key credentials." *See* **Exhibit 34**. 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 (i.e., Microsoft Identity Platform). *Id.* Accordingly, a credential ID uniquely identifies a private/public key pair.

127. The credential ID is used to retrieve the correct public key during authentication ceremonies. *See* **Exhibit 34**. The public retrieved is used to verify the signature generated with its matching private key held by phone running the Microsoft Authenticator App or a Windows 10/11 device natively having Windows Hello. *Id.* Being part of an asymmetric key pair, the public key corresponds to a private key.

128. Altering a Windows Hello or Microsoft Authenticator App credential ID causes authentication to fail. Being FIDO certified, Windows Hello utilizes the WebAuthn protocol.

Credential IDs within the WebAuthn protocol are in a tamper proof format unable to be subsequently altered because "all that would happen if an authenticator returns the wrong credential ID, or if an attacker intercepts and manipulates the credential ID, is that the WebAuthn Relying Party would not look up the correct credential public key with which to verify the returned signed authenticator data (a.k.a., assertion), and thus the interaction would end in an error." *See* ***Id.*** As altering the device ID (i.e., Credential ID) causes authentication to fail, the credential ID is necessarily in a tamper proof format unable to be altered. "Passwordless authentication using the Authenticator app follows the same basic pattern as Windows Hello for Business." *See* **Exhibit 22**. Following the same pattern, altering the device ID stored on the phone by the Authenticator App should produce the same result of causing authentication to fail. As Microsoft controls the protocol utilized by its Identity Platform, including the inability to sync passkeys across devices, users wanting the benefit of biometric authentication via the Microsoft Authenticator App or Windows Hello have no choice but to store on their phone or Windows device a credential ID uniquely identifying the integrated device in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered.

129. As with the FIDO standards implemented by Windows Hello, the Microsoft Authenticator app utilizes a private key to sign a challenge. *See* **Exhibit 22**. Private keys are notoriously known secret decryption values. If the Authenticator App and Windows Hello have the private key, the public key must be held by Microsoft's Identity Platform. By being part of a pair split between the authenticators and Microsoft's Identity Platform, the private key secret decryption value is in a tamper proof format unable to be subsequently altered. As Microsoft controls the protocol utilized by its Identity Platform, including the use of public/private key validation, users wanting the benefit of biometric authentication via the Microsoft Authenticator

App or Windows Hello have no choice but to store on their phone or Windows device a secret decryption value in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered.

130. Microsoft describes user authentication via the Authenticator App and Windows Hello as comprising receipt of a proof-of-presence challenge completed when the user enters their biometrics. *See* **Exhibit 22**. Entering their biometric necessarily requires receiving scan data from a biometric scan. When the user completes the challenge, the private key held by the phone or Windows device is unlocked. Determining if the user successfully completed the challenge necessarily requires comparing the scan data to the biometric data to determine whether the scan data matches the biometric data.

131. As detailed by Microsoft, after the user has completed the biometric challenge, a signature is generated with the authenticator's private key and sent to Microsoft Identity Platform for verification. *See* **Exhibit 22**. The Microsoft Authenticator App operates either on iOS or Android phones. As iOS and Android phones are cellular phones, often possessing Wi-Fi and Bluetooth capability, anything sent to Microsoft Identity Platform will be sent wirelessly – either via Wi-Fi, Bluetooth, or the cellular network. Laptops are common Windows devices having biometric capabilities and are often connected to the internet via Wi-Fi or using a mobile phone as a hotspot. Accordingly, anything sent to Microsoft Identity Platform will be sent wirelessly via the laptop's Wi-Fi connection or connected phones cellular connection.

132. After the user has completed the biometric challenge, a signature is generated with the authenticator's private key and sent to Microsoft Identity Platform for verification. *See* **Exhibit 22**. As previously noted, password-less authentication using the Microsoft Authenticator App follows the same basic pattern as FIDO certified Windows Hello. FIDO incorporates the

WebAuthn Protocol. *See* **Exhibit 35**. Accordingly, the challenge sent to the Authenticator App is the equivalent of a FIDO server challenge sent to the Windows Hello authenticator with a WebAuthn authenticatorGetAssertion request. The response to the request, therefore, will contain a signature signed by the authenticator's private key and a credential ID (i.e., device ID) identifying which public key to use to verify the signature. *See* **Exhibit 34**. Upon receiving the response, the WebAuthn/FIDO server will use the device ID (i.e., credential ID) to locate the appropriate public key to verify the signature generated with the private key. *Id.* Accordingly, a FIDO/WebAuthn server within the Microsoft Identity Platform will authenticate a device ID (i.e., credential ID) sent in response to successful completion of a biometric challenge.

133. As detailed *supra*, upon successful completion of the biometric challenge, a device ID and signature is sent from the authenticator to Microsoft Identity Platform for authentication. The Microsoft Identity Platform utilizes OpenID Connect and Microsoft encourages developers to gain an understanding of the protocol and concepts to add authentication to applications. *See* **Exhibit 36**. Within the protocol, Microsoft identifies its identity platform as an "authorization server" managing "trust relationships" and issuing security tokens applications and APIs use for granting access (i.e. authorization) and authentication. *See* **Exhibit 27**. Accordingly, Microsoft describes its Identity Platform as a third party that operates as a trusted authority for authentication.

134. As noted supra, the FIDO server within the Microsoft Identity Platform uses a credential ID operating as a device ID to verify a signature generated with a private key bound to the identified device (i.e., authenticator). Such action is made possible by registering the credentials with an account and associating the account with the credential ID and credential public key. *See* **Exhibit 34**. Accordingly, the Microsoft Identity Platform maintains a listing of legitimate device IDs.

135. Authentication is provided by a FIDO server incorporated into the Microsoft Identity Platform in conjunction with an OpenID Connect Authentication Server that is also incorporated into the Microsoft Identity Platform. While OpenID Connect is a federation protocol, it is compatible and complementary with FIDO. "The value of a FIDO authentication capability is amplified by a federated system, where the federation system extends the benefits of a FIDO authentication to applications and services without requiring FIDO to be directly integrated with those applications." *See* **Exhibit 37**. When a federated system, such as OpenID Connect, is combined with FIDO, the OpenID Provider (OP) sends a FIDO server challenge to an authenticator which is returned as the FIDO Authentication response. *Id*.

136. FIDO incorporates the WebAuthn Protocol. Accordingly, the FIDO server challenge sent will be a WebAuthn authenticatorGetAssertion request. The response to the request received will contain a signature generated by the authenticator's private key and a credential ID. *See* **Exhibit 35**. Upon receiving the response, the WebAuthn/FIDO server will use the credential ID to locate the appropriate public key to verify the signature generated with the private key. *See* **Exhibit 34**. Accordingly, a FIDO/WebAuthn server incorporated into Microsoft Identity Platform will authenticate a device ID (i.e., credential ID). After validating the device ID, the FIDO/WebAuthn portion of Microsoft's Identity Platform "redirects the user agent back to the Application Provider with an authentication assertion". *See* **Exhibit 37**.

137. As the Microsoft Identity Platform utilizes OpenID Connect, the authentication assertion will be an ID Token. The ID Token received from Microsoft's Identity platform enables access to an application. The ID Token is thus an access message allowing access to an application that is provided by Microsoft operating as a third party trusted authority. Should the user also

require an access token to access further features of the application, such as files containing stored data, the ID token can be exchanged for an access token. *See* **Exhibit 27**.

138. Further, Microsoft Family enables parents to "designate the age limit for content [a family member] will have permission to access", such as apps and games. *See* **Exhibit 38**.

139. Microsoft 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 as authenticators Windows Hello preinstalled on Windows 10/11 computers and the Microsoft Authenticator App available for download and Microsoft Identity Platform for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Microsoft also induces infringement of claims 22, 23, 24, 25, 26, and 27 by making Microsoft Identity available for integration with developer applications, providing the Microsoft Authentication Library, and a substantial knowledge base teaching developers and business subscribing to Microsoft'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. Microsoft 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 as authenticators Windows Hello preinstalled on Windows 10/11 computers and the Microsoft Authenticator App available for download and Microsoft Identity Platform for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Microsoft also induces infringement of claims 22, 23, 24, 25, 26, and 27 by making Microsoft Identity available for integration with developer

applications, providing the Microsoft Authentication Library, and a substantial knowledge base teaching developers and business subscribing to Microsoft'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.

141. Microsoft received actual notice of the 954 Patent at least as early as July 29, 2016 when Proxense sent Microsoft correspondence attaching a copy of the 954 Patent. Microsoft performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness to the acts that constitute direct and/or indirect infringement of the 954 Patent.

142. Since at least July 29, 2016, through its actions and continued actions, Microsoft has indirectly infringed and continues to indirectly infringe the 954 Patent in violation of 35 U.S.C. § 271(b). Microsoft has actively induced product makers, distributors, retailers, 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 Microsoft promoting the use of the accused products are packaging Windows Hello with Windows 10/11 and public documents, which serve no function other than to direct users of the Accused Products toward infringing the 954 Patent.

143. Microsoft does so knowingly and intending that its customers and end users will commit these infringing acts. Microsoft 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.

144. In addition, Microsoft 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.

145. For example, Microsoft 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 954 Patent, including claim 1. Microsoft continues to sell and offer to sell these products in the United States after receiving notice of the 954 Patent and how its products infringe that patent.

146. 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 non-infringing use.

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

148. Upon information and belief, Microsoft will continue to infringe (both directly and indirectly) the 954 Patent unless permanently enjoined by this Court. Pursuant to 35 U.S.C. § 283,

Proxense is entitled to a permanent injunction against further infringement of the 954 Patent by Microsoft.

CLAIM 3
(Infringement of 905 Patent)

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

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

151. Microsoft 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.

152. For example, Microsoft 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 Microsoft Authenticator App and Windows Hello as authenticators. Under the coordination of Microsoft Identity Platform, the authenticators perform/execute and provide a method for verifying a user during authentication of the device. Microsoft 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 Microsoft's universal platform password-less architecture incorporating Microsoft Identity platform. The coordination and control provide by Microsoft Identity Platform of the other components within the architecture provides a system for verifying a user during authentication of the device.

153. As described *supra*, the Microsoft Authenticator App "turns any iOS or Android phone into a strong, passwordless credential." Utilizing the Android operating system, Android

Phones having the Microsoft Authenticator App 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. Starting with Android 10, stored biometric data on Android phones includes a fingerprint and facial recognition. *See* **Exhibit 22**.

154. As to protect biometric data, Android's implementation guidelines require tamper-proof "raw fingerprint data or derivatives (for example, templates) 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 17**. Requiring acquisition and recognition to occur inside the TEE, fingerprint data never leaves the TEE. Android's TEE, called Trusty, "uses ARM's Trustzone™ to virtualize the main processor and create a secure trusted execution environment" isolated from the rest of the system. 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.

155. On Android phones, access to the biometric hardware is controlled by Fingerprint HIDL. The methods enabled by the Fingerprint HIDL do not permit altering 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, means that Android phones 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.

156. When the Authenticator App is installed on an iOS device, biometric data would be stored within Apple's Secure Enclave. *See* **Exhibit 30**. Designed to keep sensitive data secure even when compromised, the Secure Enclave provides a tamper proof format for sensitive data.

157. Additionally, both iOS and Android phones require user consent to enroll a fingerprint. *See* **Exhibits 31** and **32**. As enrolling fingerprints on both iPhones and Android phones require entering PIN / Passcode / Password to evidence user consent, iPhones and Android phones store biometric data of user in a tamper proof format unable to be subsequently altered.

158. Windows compatible systems meeting Microsoft's minimum hardware requirements also store data of a user in a tamper proof format unable to be subsequently altered.

159. Microsoft details that the Authenticator App uses key based authentication technology similar to Windows Hello. ("Microsoft Authenticator uses key-based authentication to enable a user credential that is tied to a device, where the device uses a PIN or biometric. Windows Hello for Business uses a similar technology." *See* **Exhibit 43**. Windows Hello is FIDO certified. The FIDO specification incorporates the WebAuthn Specification. *See* **Exhibit 33**. Windows Hello, therefore, uses a key-based authentication technology compliant with the WebAuthn and FIDO specifications. Utilizing a similar technology, the Microsoft Authenticator App most use a key-based authentication analogous to the WebAuthn and FIDO protocols.

160. Under the WebAuthn specification, "compliant authenticators protect public key credentials." *See* **Exhibit 34**. 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 is part of an asymmetric key pair containing a public key returned to a relying party (i.e., Microsoft Identity Platform). *Id*. Accordingly, a credential ID uniquely identifies a private/public key pair.

161. The credential ID is used to retrieve the correct public key during authentication ceremonies. *Id*. The public key retrieved is used to verify the signature generated with its

matching private key held by phone running the Microsoft Authenticator App or a Windows 10/11 device natively having Windows Hello. *Id*. Being part of an asymmetric key pair, the pubic key corresponds to a private key.

162. Altering a Windows Hello credential ID causes authentication to fail. Being FIDO certified, Windows Hello utilizes the WebAuthn protocol. Credential IDs within the WebAuthn protocol are in a tamper proof format unable to be subsequently altered because "all that would happen if an authenticator returns the wrong credential ID, or if an attacker intercepts and manipulates the credential ID, is that the WebAuthn Relying Party would not look up the correct credential public key with which to verify the returned signed authenticator data (a.k.a., assertion), and thus the interaction would end in an error." *Id*. As altering the device ID (i.e., Credential ID) causes authentication to fail, the credential ID is necessarily in a tamper proof format unable to be altered. "Passwordless authentication using the Authenticator app follows the same basic pattern as Windows Hello for Business." *See* **Exhibit 22**. Following the same pattern, altering the device ID caused to be stored on the phone by the Authenticator App should produce the same result of causing authentication to fail. As Microsoft controls the protocol utilized by its Identity Platform, including the inability to sync passkeys across devices, users wanting the benefit of biometric authentication via the Microsoft Authenticator App or Windows Hello have no choice but to store on their phone or Windows device a credential ID uniquely identifying the integrated device in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered.

163. As with the FIDO standards implemented by Windows Hello, the Microsoft Authenticator app utilizes a private key to sign a challenge. *Id*. Private keys are notoriously known secret decryption values. If the Authenticator App and Windows Hello have the private key, the

public key must be held by Microsoft's Identity Platform. By being part of a pair split between the authenticator and Microsoft's Identity Platform, the private key is a secret decryption value is in a tamper proof format unable to be subsequently altered. As Microsoft controls the protocol utilized by its Identity Platform, including the use of public/private key validation, users wanting the benefit of biometric authentication via the Microsoft Authenticator App or Windows Hello have no choice but to store on their phone or Windows device a secret decryption value in a tamper proof format written to a storage element on the integrated device that is unable to be subsequently altered.

164. Microsoft describes user authentication via the Authenticator App and Windows Hello as comprising receipt of proof-of-presence challenge completed when the user enters their biometrics. *Id*. Entering their biometric necessarily requires receiving scan data from a biometric scan. When the user completes the challenge the private key held by the phone or Windows device is unlocked. Determining if the user successfully completed challenge necessarily requires comparing the scan data to the biometric data to determine whether the scan data matches the biometric data.

165. As detailed by Microsoft, after the user has completed the biometric challenge, a signature is generated with the authenticator's private key and sent to Microsoft Identity Platform for verification. *Id*. The Microsoft Authenticator App operates either on iOS or Android phones. As iOS and Android phones are cellular phones, often possessing Wi-Fi and Bluetooth capability, anything sent to Microsoft Identity Platform will be sent wirelessly – either via Wi-Fi, Bluetooth, or the cellular network. Laptops are common Windows devices having biometric capabilities and are often connected to the internet via Wi-Fi or using a mobile phone as a hotspot. Accordingly,

anything sent to Microsoft Identity Platform will be sent wirelessly via the laptop's Wi-Fi connection or connected phones cellular connection.

166. After the user has completed the biometric challenge, a signature is generated with the authenticator's private key and sent to Microsoft Identity Platform for verification. *Id.* As previously noted, password-less authentication using the Authenticator App follows the same basic pattern as FIDO certified Windows Hello. FIDO incorporates the WebAuthn Protocol. *See* **Exhibit 35**. Accordingly, the challenge sent to the Authenticator App is the equivalent of a FIDO server challenge sent to the Windows Hello authenticator with a WebAuthn authenticatorGetAssertion request. The response to the request, therefore, will contain a signature signed by the authenticator's private key and a credential ID (i.e., device ID) identifying which public key to use to verify the signature. *See* **Exhibit 34**. Upon receiving the response, the WebAuthn/FIDO server will use the device ID (i.e., credential ID) to locate the appropriate public key to verify the signature generated with the private key. *Id.* Accordingly, a FIDO/WebAuthn server within the Microsoft Identity Platform will authenticate a device ID (i.e., credential ID) sent in response to successful completion of a biometric challenge.

167. As detailed *supra*, upon successful completion of the biometric challenge, a device ID and signature is sent from the authenticator to Microsoft Identity Platform for authentication. The Microsoft Identity Platform utilizes OpenID Connect and Microsoft encourages developers to gain an understanding of the protocol and concepts to add authentication to applications. *See* **Exhibit 36**. Within the protocol, Microsoft identifies its identity platform as an "authorization server" managing "trust relationships" and issuing security tokens applications and APIs use for granting access (i.e. authorization) and authentication. *See* **Exhibit 27**. Accordingly, Microsoft describes its Identity Platform as a third party that operates as a trusted authority for authentication.

168. As noted supra, the FIDO server within the Microsoft Identity Platform uses a credential ID operating as a device ID to verify a signature generated with a private key bound to the identified device (i.e., authenticator). Such action is made possible by registering the credentials with an account and associating the account with the credential ID and credential public key. *See* **Exhibit 34**. Accordingly, the Microsoft Identity Platform maintains a listing of legitimate device IDs.

169. Authentication is provided by a FIDO sever incorporated into the Microsoft Identity Platform in conjunction with OpenID Connect Authentication Server that is also incorporated into the Microsoft Identity Platform. While OpenID Connect is a federation protocol, it is compatible and complementary with FIDO. "The value of a FIDO authentication capability is amplified by a federated system, where the federation system extends the benefits of a FIDO authentication to applications and services without requiring FIDO to be directly integrated with those applications." *See* **Exhibit 37**. When a federated system, such as OpenID Connect, is combined with FIDO, the OpenID Provider (OP) sends a FIDO server challenge to an authenticator which is returned as the FIDO Authentication response. *Id.*

170. FIDO incorporates the WebAuthn Protocol. Accordingly, the FIDO server challenge sent will a be WebAuthn authenticatorGetAssertion request. The response to the request received will contain a signature generated by the authenticator's private key and a credential ID. *See* **Exhibit 35**. Upon receiving the response, the WebAuthn/FIDO server will use the credential ID to locate the appropriate public key to verify the signature generated with the private key. *See* **Exhibit 34**. Accordingly, a FIDO/WebAuthn server incorporated into Microsoft Identity Platform will authenticate a device ID (i.e., credential ID). After validating the device ID, the

FIDO/WebAuthn portion of Identity Platform "redirects the user agent back to the Application Provider with an authentication assertion". *See* **Exhibit 37**.

171. As the Microsoft Identity Platform utilizes OpenID Connect, the authentication assertion will be an ID Token. The ID Token received from Microsoft's Identity platform enables access to an application. The ID Token is thus an access message provided by Microsoft operating as a third party trusted authority allowing access to an application. Should the user also require an access token to access further features of the application, such as files containing stored data, the ID token can be exchanged for an access token. *See* **Exhibit 27**.

172. Further, Microsoft Family enables parents to "designate the age limit for content [a family member] will have permission to access", such as apps and games. *See* **Exhibit 38**.

173. Further, Microsoft actively markets to and does business with clients in the financial services space and its Identity Services, including the use of its authenticators to complete a financial transaction.

174. Microsoft 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 as authenticators Windows Hello preinstalled on Windows 10/11 computers and the Microsoft Authenticator App available for download and Microsoft Identity Platform for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Microsoft also induces infringement of claim 15 by making Microsoft Identity available for integration with developer applications, providing the Microsoft Authentication Library, and a substantial knowledge base teaching developers and business subscribing to Microsoft'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.

175. Microsoft contributes to direct 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 as authenticators Windows Hello preinstalled on Windows 10/11 computers and the Microsoft Authenticator App available for download and Microsoft Identity Platform for use by users to access resources offered by Microsoft, including applications, services, and subscriptions. Microsoft also induces infringement of claim 15 by making Microsoft Identity available for integration with developer applications, providing the Microsoft Authentication Library, and a substantial knowledge base teaching developers and business subscribing to Microsoft'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.

176. Microsoft received actual notice of the 905 Patent at least as early as July 29, 2016 when Proxense sent Microsoft correspondence attaching a copy of the 905 Patent. Microsoft performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness to the acts that constitute direct and/or indirect infringement of the 905 Patent.

177. Since at least July 29, 2016, through its actions and continued actions, Microsoft has indirectly infringed and continues to indirectly infringe the 905 Patent in violation of 35 U.S.C. § 271(b). Microsoft has actively induced product makers, distributors, retailers, 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 Microsoft promoting the use of the accused products are packaging Windows Hello with Windows 10/11 and public documents, which serve no function other than to direct users of the Accused Products toward infringing the 954 Patent.

178. Microsoft does so knowingly and intending that its customers and end users will commit these infringing acts. Microsoft 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.

179. In addition, Microsoft 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 905 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.

180. For example, Microsoft 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 905 Patent, including claim 1. Microsoft continues to sell and offer to sell these products in the United States after receiving notice of the 905 Patent and how its products infringe that patent.

181. 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 non-infringing use.

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

183. Upon information and belief, Microsoft 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 Microsoft.

CLAIM 4
(Infringement of 042 Patent)

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

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

186. 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.

187. For example, Microsoft 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 Microsoft's universal platform password-less architecture incorporating the Microsoft Identity Platform and Microsoft approved authenticators developed by itself and its

OEM partners. Under the coordination of Microsoft Identity Platform, the authenticators perform/execute and provide, a method for verifying a user during authentication of the device.

188. To be used with Microsoft's Identity Platform, an authenticator needs to have FIDO2 certification. *See* **Exhibit 19**. Accordingly, all authenticators used to practice the method have to meet the standards set by Microsoft and FIDO.

189. The claims include a controller placing within memory information that can only be accessed by a corresponding access key provided by an external application. Such memory is present in FIDO compliant authenticators. The FIDO CTAP specification incorporates the WebAuthn Specification. Under the WebAuthn specification, "compliant authenticators protect public key credentials." *See* **Exhibit 34**. 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.* Every FIDO compliant authenticator, 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.

190. The credentials stored within a compliant authenticator can only be accessed with the appropriate access key. "A public key credential can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." *Id.* When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external FIDO server. *Id.* Therefore, the RP ID is an access key.

191. During a WebAuthn authentication ceremony, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. *See* **Exhibit 35**. The authenticatorGetAssertion request contains a relying party identifier (RP ID). *Id.* The authenticatorGetAssertion is called in response to a get request issued by the relying party attempting authentication. *Id.* The RP ID is provided by the relying party attempting authentication from an external authenticator. As an authenticator will only return credentials corresponding to the RP ID access key provided by the external relying party, the authenticator has the controller and memory necessary for minimal embodiment of a personal digital key ("PDK").

192. The FIDO standard requires all communications with BLE authenticators be encrypted. *Id.* FIDO compliant BLE capable authenticators, accordingly, include a receiver-decoder circuit ("RDC") enabling encrypted communications.

193. As noted above, FIDO compliant authenticators include the elements of minimal embodiment of a PDK. Windows Hello is a FIDO compliant authenticator and therefore a PC running Windows is an external PDK. Further, Bluetooth or other wireless pairing between a BLE capable Microsoft-compatible FIDO2 security key (i.e., authenticator) and a Windows PC necessarily requires a first wireless link between the devices to establish the pairing.

194. During an authentication ceremony, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. *See* **Exhibit 35**. Browsers, such as Microsoft Edge and Google Chrome, operating on Windows 11 forward the authenticatorGetAssertion to the external authenticator. *See* **Exhibit 39**. Windows supports use of BLE and NFC roaming authenticators with Edge and Chrome web browsers. Regardless of the web browser, Windows provides WebAuthn APIs enabling interactions with

authenticators to take place. *See* **Exhibit 35**. As such, Windows will connect with a roaming authenticator over BLE or NFC to forward to the authenticator the authenticatorGetAssertion request received via either the Edge or Chrome web browser.

195. Authentication is a service provided by a FIDO server incorporated into the Microsoft Identity Platform, and the credential ID is necessary for Microsoft's FIDO server to perform the authentication function. Upon receiving the response (i.e., enablement signal), the WebAuthn/FIDO 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. *See* **Exhibit 34**. As the proper credential ID is needed for Microsoft'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 Microsoft's FIDO server, the response to the authenticatorGetAssertion request generated by the authenticator is an enablement signal enabling authentication by Microsoft'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.

196. The responsibility for sending responses received from an authenticator via BLE or NFC falls upon the WebAuthn Client. *Id.* Microsoft identifies its Edge Browser as the WebAuthn client. *See* **Exhibit 40**. Similarly, 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. *See* **Exhibit 41**. As such, Edge and Chrome will forward the enablement signal received from a roaming authenticator to Microsoft's FIDO server.

197. Microsoft 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 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.

198. Microsoft contributes to direct infringement of at least claim 1 of the 042 Patent in violation of 35 U.S.C. § 271(c) by making 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.

199. Microsoft received actual notice of the 042 Patent at least as early as July 29, 2016 when Proxense sent Microsoft correspondence attaching a copy of the 042 Patent. Microsoft performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness to the acts that constitute direct and/or indirect infringement of the 042 Patent.

200. Since at least July 29, 2016, through its actions and continued actions, Microsoft has indirectly infringed and continues to indirectly infringe the 042 Patent in violation of 35 U.S.C. § 271(b). Microsoft has actively induced product makers, distributors, retailers, 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 Microsoft promoting the use of the accused products are packaging Windows Hello with Windows 10/11 and public documents,

which serve no function other than to direct users of the Accused Products toward infringing the 042 Patent.

201. Microsoft does so knowingly and intending that its customers and end users will commit these infringing acts. Microsoft 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.

202. In addition, Microsoft has indirectly infringed and continues to indirectly infringe the 042 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.

203. For example, Microsoft 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 042 Patent, including claim 1. Microsoft continues to sell and offer to sell these products in the United States after receiving notice of the 042 Patent and how its products infringe that patent.

204. 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 non-infringing use.

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

206. Upon information and belief, Microsoft 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 Microsoft.

CLAIM 5
(Infringement of 289 Patent)

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

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

209. Microsoft 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.

210. For example, Microsoft 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 Microsoft's universal platform password-less architecture incorporating the Microsoft Identity Platform and Microsoft approved authenticators developed by itself and its OEM partners. Under the coordination of Microsoft Identity Platform, the authenticators perform/execute and provide, a method for verifying a user during authentication of the device.

211. To be used with Microsoft's Identity Platform, an authenticator needs to have FIDO2 certification. *See* **Exhibit 19**. Accordingly, all authenticators used to practice the method have to meet the standards set by Microsoft and FIDO.

212. The claims include a controller placing within memory information that can only be accessed by a corresponding access key provided by an external application. Such memory is present in FIDO compliant authenticators. The FIDO CTAP specification incorporates the WebAuthn Specification. Under the WebAuthn specification, "compliant authenticators protect public key credentials." *See* **Exhibit 34**. 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.* Every FIDO compliant authenticator, 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.

213. The credentials stored within a compliant authenticator can only be accessed with the appropriate access key. "A public key credential can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." *Id.* When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external FIDO server. *Id*. Therefore, the RP ID is an access key.

214. During a WebAuthn authentication ceremony, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. *See* **Exhibit 35**. The authenticatorGetAssertion request contains a relying party identifier (RP ID). *Id.*

The authenticatorGetAssertion is called in response to a get request issued by the relying party attempting authentication. *Id.* The RP ID is provided by the relying party attempting authentication from an external authenticator. As an authenticator will only return credentials corresponding to the RP ID access key provided by the external relying party, the authenticator has the controller and memory necessary for minimal embodiment of a personal digital key ("PDK").

215. The FIDO standard requires all communications with BLE authenticators be encrypted. *Id.* FIDO compliant BLE capable authenticators, accordingly, include a receiver-decoder circuit ("RDC") enabling encrypted communications.

216. As noted above, FIDO compliant authenticators include the elements of minimal embodiment of a PDK. Windows Hello is a FIDO compliant authenticator and therefore a PC running Windows is an external PDK. Further, Bluetooth or other wireless pairing between a BLE capable Microsoft-compatible FIDO2 security key (i.e., authenticator) and a Windows PC necessarily requires a first wireless link between the devices to establish the pairing.

217. During an authentication ceremony, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. *See* **Exhibit 35**. Browsers, such as Microsoft Edge and Google Chrome, operating on Windows 11 forward the authenticatorGetAssertion to the external authenticator. *See* **Exhibit 39**. Windows supports use of BLE and NFC roaming authenticators with Edge and Chrome web browsers. Regardless of the web browser, Windows provides WebAuthn APIs enabling interactions with authenticators to take place. *See* **Exhibit 35**. As such, Windows will connect with a roaming authenticator over BLE or NFC to forward to the authenticator the authenticatorGetAssertion request received via either the Edge or Chrome web browser.

218. Authentication is a service provided by a FIDO server incorporated into the Microsoft Identity Platform, and the credential ID is necessary for Microsoft's FIDO server to perform the authentication function. Upon receiving the response (i.e., enablement singal), the WebAuthn/FIDO 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. *See* **Exhibit 34**. As the proper credential ID is needed for Microsoft's FIDO server to authenticate a user, and the credential ID is70ncludeed within a response to a get request having the appropriate relying party ID received from the Microsoft's FIDO server, the response to the authenticatorGetAssertion request generated by the authenticator is an enablement signal enabling authentication by Microsoft'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.

219. The responsibility for sending responses received from an authenticator via BLE or NFC falls upon the WebAuthn Client. *Id.* Microsoft identifies its Edge Browser as the WebAuthn client. *See* **Exhibit 40**. Similarly, 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. *See* **Exhibit 41**. As such, Edge and Chrome will forward the enablement signal received from a roaming authenticator to Microsoft's FIDO server.

220. Microsoft 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 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.

221. Microsoft contributes to direct infringement of at least claims 14 and 16 of the 289 Patent in violation of 35 U.S.C. § 271© by making 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.

222. Microsoft received actual notice of the 289 Patent at least as early as July 29, 2016 when Proxense sent Microsoft correspondence attaching a copy of the 289 Patent. Microsoft performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness to the acts that constitute direct and/or indirect infringement of the 289 Patent.

223. Since at least July 29, 2016, through its actions and continued actions, Microsoft has indirectly infringed and continues to indirectly infringe the 289 Patent in violation of 35 U.S.C. § 271(b). Microsoft 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 Microsoft promoting the use of the accused products are packaging Windows Hello with Windows 10/11 and public documents, which serve no function other than to direct users of the Accused Products toward infringing the 289 Patent.

224. Microsoft does so knowingly and intending that its customers and end users will commit these infringing acts. Microsoft 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.

225. In addition, Microsoft 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.

226. For example, Microsoft 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 claim 14. Microsoft 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.

227. 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 non-infringing use.

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

229. Upon information and belief, Microsoft 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 Microsoft.

CLAIM 6
(Infringement of 960 Patent)

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

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

232. Microsoft 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.

233. For example, Microsoft directly infringes at least claims 14 and 16 selling access to, and/or offering to sell access to within the United States Microsoft's universal platform password-less architecture incorporating the Microsoft Identity Platform and Microsoft approved authenticators developed by itself and its OEM partners. Under the coordination of Microsoft Identity Platform, the authenticators perform/execute and provide, a method for verifying a user during authentication of the device.

234. To be used with Microsoft's Identity Platform, an authenticator needs to have FIDO2 certification. *See* **Exhibit 19**. Accordingly, all authenticators used to practice the method have to meet the standards set by Microsoft and FIDO.

235. The claims include a controller placing within memory information that can only be accessed by a corresponding access key provided by an external application. Such memory is

present in FIDO compliant authenticators. The FIDO CTAP specification incorporates the WebAuthn Specification. Under the WebAuthn specification, "compliant authenticators protect public key credentials." *See* **Exhibit 34**. 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*. Every FIDO compliant authenticator, 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.

236. The credentials stored within a compliant authenticator can only be accessed with the appropriate access key. "A public key credential can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." *Id.* When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external FIDO server. *Id.* Therefore, the RP ID is an access key.

237. During a WebAuthn authentication ceremony, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. *See* **Exhibit 35**. The authenticatorGetAssertion request contains a relying party identifier (RP ID). *Id.* The authenticatorGetAssertion is called in response to a get request issued by the relying party attempting authentication. *Id.* The RP ID is provided by the relying party attempting authentication from an external authenticator. As an authenticator will only return credentials corresponding to the RP ID access key provided by the external relying party, the authenticator

has the controller and memory necessary for minimal embodiment of a personal digital key ("PDK").

238. The FIDO standard requires all communications with BLE authenticators be encrypted. *Id.* FIDO compliant BLE capable authenticators, accordingly, include a receiver-decoder circuit ("RDC") enabling encrypted communications.

239. As noted above, FIDO compliant authenticators include the elements of minimal embodiment of a PDK. Windows Hello is a FIDO compliant authenticator and therefore a PC running Windows is an external PDK. Further, Bluetooth or other wireless pairing between a BLE capable Microsoft-compatible FIDO2 security key (i.e., authenticator) and a Windows PC necessarily requires a first wireless link between the devices to establish the pairing.

240. During an authentication ceremony, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. *See* **Exhibit 35**. Browsers, such as Microsoft Edge and Google Chrome, operating on Windows 11 forward the authenticatorGetAssertion to the external authenticator. *See* **Exhibit 39**. Windows supports use of BLE and NFC roaming authenticators with Edge and Chrome web browsers. Regardless of the web browser, Windows provides WebAuthn APIs enabling interactions with authenticators to take place. *See* **Exhibit 35**. As such, Windows will connect with a roaming authenticator over BLE or NFC to forward to the authenticator the authenticatorGetAssertion request received via either the Edge or Chrome web browser.

241. Authentication is a service provided by a FIDO server incorporated into the Microsoft Identity Platform, and the credential ID is necessary for Microsoft's FIDO server to perform the authentication function. Upon receiving the response (i.e., enablement singal), the WebAuthn/FIDO 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. *See* **Exhibit 34**. As the proper credential ID is needed for Microsoft'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 Microsoft's FIDO server, the response to the authenticatorGetAssertion request generated by the authenticator is an enablement signal enabling authentication by Microsoft'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.

242. The responsibility for sending responses received from an authenticator via BLE or NFC falls upon the WebAuthn Client. *Id*. Microsoft identifies its Edge Browser as the WebAuthn client. *See* **Exhibit 40**. Similarly, 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. *See* **Exhibit 41**. As such, Edge and Chrome will forward the enablement signal received from a roaming authenticator to Microsoft's FIDO server.

243. Microsoft has induced infringement, and continues to induce infringement, of at least claims 14 and 16 of the 960 Patent in violation of 35 U.S.C. § 271 by making 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.

244. Microsoft contributes to direct infringement of at least claims 14 and 16 of the 960 Patent in violation of 35 U.S.C. § 271(c) by making 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.

245. Microsoft received actual notice of the 960 Patent at least as early as July 29, 2016 when Proxense sent Microsoft correspondence attaching a copy of the 960 Patent. Microsoft performed and continues to perform the acts that constitute direct and/or indirect infringement, with knowledge or willful blindness to the acts that constitute direct and/or indirect infringement of the 960 Patent.

246. Since at least July 29, 2016, through its actions and continued actions, Microsoft has indirectly infringed and continues to indirectly infringe the 960 Patent in violation of 35 U.S.C. § 271(b). Microsoft 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 Microsoft promoting the use of the accused products are packaging Windows Hello with Windows 10/11 and public documents, which serve no function other than to direct users of the Accused Products toward infringing the 960 Patent.

247. Microsoft does so knowingly and intending that its customers and end users will commit these infringing acts. Microsoft 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.

248. In addition, Microsoft 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.

249. For example, Microsoft 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 claim 14. Microsoft 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.

250. 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 non-infringing use.

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

252. Upon information and belief, Microsoft 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 Microsoft.

DEMAND FOR JURY TRIAL

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

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for relief against Defendants as follows:

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

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

c. An order awarding damages sufficient to compensate Plaintiff for Defendant's 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: April 28, 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 *(pro hac vice forthcoming)*
ljiang@susmangodfrey.com
SUSMAN GODFREY L.L.P.
1900 Avenue of the Stars, Suite 1400
Los Angeles, CA 90067-6029
Telephone: (310) 789-3100
Facsimile: (3103) 789-3150

Counsel for Plaintiff Proxense, LLC