(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 December 2024 (26.12.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/263888 A2

(51) International Patent Classification:
G06F 21/31 (2013.01) *H04L 9/40* (2022.01)

(21) International Application Number:
PCT/US2024/034977

(22) International Filing Date:
21 June 2024 (21.06.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/509,978 23 June 2023 (23.06.2023) US

(71) Applicant: **TWOSENSE, INC.** [US/US]; 2108 Fifth Avenue, Suite 5, New York, New York 10035 (US).

(72) Inventors: **GORDON, Dawud**; c/o TwoSense, Inc., 2108 Fifth Avenue, Suite 5, New York, New York 10035 (US). **TANIOS, John**; c/o Twosense, Inc., 2108 Fifth Avenue, Suite 5, New York, New York 10035 (US).

(74) Agent: **KOFFSKY, Mark I.**; Koffsky Schwalb LLC, 500 Seventh Avenue, 8th Floor, New York, New York 10018 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE,

(54) Title: HUMAN-TO-HUMAN, FACE-TO-FACE VERIFICATION AS A FACTOR IN MULTI-FACTOR AUTHENTICATION

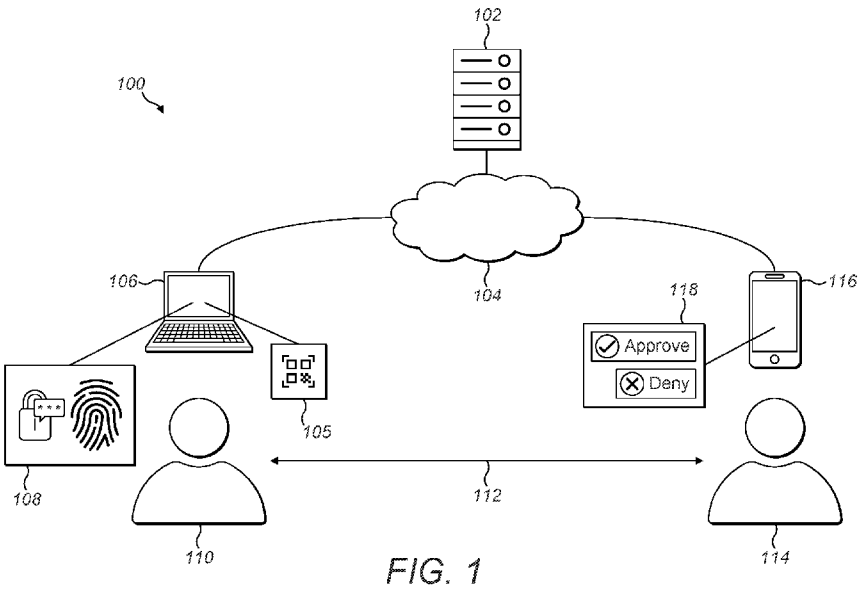

FIG. 1

(57) Abstract: A trusted 3rd party human steps into a single multi-factor authentication (MFA) transaction and act as a stand-in for a temporary failure of a single factor, replacing hardware/software authentication of a human with human-to-human, face-to-face (H2F2) authentication. This procedure includes the process for verifying and digitizing H2F2 authentication, certifying its correctness, and incorporating it into an MFA transaction.

WO 2024/263888 A2

[Continued on next page]

SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

HUMAN-TO-HUMAN, FACE-TO-FACE VERIFICATION AS A FACTOR IN MULTI-FACTOR AUTHENTICATION

RELATED APPLICATIONS

[0001] This application claims the benefit of the following U.S. Provisional Patent Application, which is incorporated by reference in its entirety:

[0002] U.S. Provisional Application Serial No. 63/509,978, filed on June 23, 2024.

BACKGROUND

[0003] Multi-factor authentication (MFA) systems use a combination of more than one of "something you know" (e.g., a password), "something you have" (e.g., a phone or a hardware key), and "something you are" (e.g., fingerprint or retina biometric scan). Typically, MFA consists of 2 examples, one from each of the above categories. Each of these categories may fail at times: a user forgets a password, loses a device, or has a band-aid on their finger. Failure results in an account restore procedure to reset authentication, which, if spoofable, creates a security weakness and a path to compromise for an attacker.

[0004] If all else fails, digital identity relies on physical identity verification as a transaction at the root of trust of digital identity - humans recognizing humans, for example replacing a driver's license at the DMV, or going to the IT helpdesk to unlock an account. These interactions are a single transaction between a human and another human, for the purpose of resetting or unlocking an account. They are expensive, take place in the physical world, and circumvent the entire authentication process, essentially revisiting the entire know your customer (KYC) process, often using documents and other non-factors for identity verification to reset and resign one or all authentication factors for the user. Once the account is reset, the user must progress through the re-enrollment process and complete it before they may continue.

[0005] Account resets, whether automated or human-to-human, represent both the process for which an authorized user may reset their digital account, and also a huge security gap where an attacker may find purchase and an entryway into an otherwise secure system.

[0006] Furthermore, as organizations move away from passwords to a passwordless future, the choice of factors is limited to "something you have" and "something you are," both of which are required. Without passwords, users lose the option to use another factor category

should they lose their key/device or be unable to pass a biometric check. Losing access to either forces the reset and re-enrollment flow for one or both, making the problem larger in scale. Finally, biometric systems are notorious for false rejects, where an authorized user cannot be authenticated, either temporarily or permanently, forcing account reset transactions.

SUMMARY

[0007] This disclosure solves the foregoing problem by using human recognition as a "something you are" factor in a multi-factor authentication transaction, as a fallback for failed biometric authentication. Rather than using human verification as a gatekeeper for account or factor reset, this disclosure leverages the process as a response to a single factor challenge in a multi-factor transaction. Rather than using human-to-human verification at a helpdesk to allow a reset of a forgotten password, or to unenroll a biometric and allow the user to re-enroll themselves, this disclosure allows a user to pass a multi-factor authentication challenge by presenting another factor, such as a password or cryptographic device, then be verified by another human to complete a single multi-factor authentication challenge without needing to reset, re-enroll, or modify any of the factors currently enrolled. This disclosure allows a trusted 3rd party human to step into a single multi-factor authentication transaction and act as a stand-in for a temporary failure of a single factor, replacing hardware/software authentication of a human with human-to-human, face-to-face (H2F2) authentication. The disclosure is the process for verifying and digitizing H2F2 authentication, certifying its correctness, and incorporating it into an MFA transaction.

BRIEF DESCRIPTION OF THE FIGURES

[0008] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views, together with the detailed description below, are incorporated in and form part of the specification, and serve to further illustrate embodiments of concepts that include the claimed disclosure and explain various principles and advantages of those embodiments.

[0009] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been drawn to scale. For example, the dimensions of some of the elements in the figures may be exaggerated relative to other elements to help to improve understanding of embodiments of the present disclosure.

[0010] Figure 1 is a schematic showing components of the disclosure.

[0011] Figure 2 is a schematic showing steps of the disclosure.

[0012] Figure 3 shows an MFA and Request Certification Application.

[0013] Figure 4 shows an Approval Certification Application.

[0014] The apparatus and method components have been represented where appropriate by conventional symbols in the drawings, showing only those specific details that are pertinent to understanding the embodiments of the present disclosure so as not to obscure the disclosure with details that will be readily apparent to those of ordinary skill in the art having the benefit of the description herein.

DETAILED DESCRIPTION

[0015] The foregoing descriptions, formulations, diagrams, and figures are provided merely as illustrative examples, and they are not intended to require or imply that the steps of the various embodiments must be performed in the order presented or that the components of the disclosure be arranged in the same manner as presented. The steps in the foregoing descriptions and illustrations may be performed in any order, and components of the disclosure may be arranged in other ways. Words such as "then," "next," etc., are not intended to limit the order of the steps or the arrangement of components; these words are used merely to guide the reader through the description of the disclosure. Although descriptions and illustrations may describe the operations as a sequential process, one or more of the operations may be performed in parallel or concurrently, or one or more components may be arranged in parallel or sequentially. In addition, the order of the operations may be rearranged. In addition, one or more components and one or more steps may be omitted from the apparatus or process.

[0016] Further, any or all of the apparatuses or methods described herein may be performed on any combination of hardware or software. Hardware implementations include laptops, phones, PCs, tablets, central processing units (CPU), random access memory (RAM), monitors, mouses, keyboards, computer data storages, graphics cards, sound cards, speakers, motherboards, virtual reality devices, augmented reality devices, and the like. Software implementations include software running in the cloud, on devices, in virtual environments, on any hardware device, and the like.

[0017] I. <u>How This Disclosure Differs from Other Solutions</u>

[0018] Humans use H2F2 authentication for every interaction we have with other humans. However, for digital identity, its use is limited as much as possible.

[0019] Known solutions use H2F2 to create a separate transaction, usually as a last line of defense, for reinstating a connection that has been lost between a human and a digital identity. That connection is normally made with authentication or MFA. Examples are common and familiar to most people. If a user forgets a password or loses a phone, there is a need to contact support and be verified by support, often through knowledge-based authentication, but sometimes by walking to the helpdesk and showing your ID. Support personnel will then grant the user the ability to create a new password, re-enroll the new phone, etc.

[0020] The granularity may be at the factor level, allowing the user re-enrollment of a single factor like adding their new phone as an authenticator to an existing account, but there is no action possible beyond resetting all or some of the factors on an account and granting the user to re-enroll themselves. This process relies on humans to execute the process of approving the reset and doing the work of reset and re-enrollment. Support personnel, acting as the verifier, make the best decision on information given to grant an account reset and/or factor re-enrollment. The user must then do the work of re-enrollment, even if factor failure is only temporary. Often times the cause of the temporary factor failure will also cause re-enrollment to fail until the cause is alleviated (e.g., wearing surgical gloves for fingerprint scanners).

[0021] This process is also automated in a way that is now also mundane. A user forgets a password or loses a phone, and the automated system sends a link to an email that to click and re-enroll. This supplants H2F2 with something else, that may grant the user the ability to drive a reset, and then drive through the re-enrollment.

[0022] II. Shortcomings of Other Solutions

[0023] Account restore for users represents one of the greatest weaknesses in authentication systems and is one of the most exploited avenues of entry for attackers.

[0024] Automated account restores flows when implemented well, essentially offering a user a backup multi-factor authentication flow to reset and re-enroll a primary MFA. However, often, there are not enough enrolled factors to implement a distinct MFA using alternative factors for reset (for example a backup hard token, two types of biometrics, a second password, etc.), and the cause of the loss of the primary factors may very well affect the backup factors, making another flow to reset the backup factors necessary. Essentially, no matter how many factors or factor categories a system supports, any pragmatic system will require a reset flow to account for users losing access to, or use of, some or all of the factors. The result is that these flows are done with often only a single factor, and often with factors that are not secure, such as email verification links (verifies "a channel you have access to" in

the hope that it represents "something you have") or knowledge-based-authentication (KBA), which uses semi-public historical information in lieu of a password.

[0025] Making account restoration the only recourse for a temporary loss of a factor (e.g., battery dead, finger injury) creates a high-effort task for the user that, as will be demonstrated, is not necessary.

[0026] Making account restore the fallback transaction allows passwords to be required even in "passwordless" systems because they are needed as a replacement factor for avoiding reset events. For example, if a biometric fingerprint scanner fails, most systems will allow a fall back to a password instead.

[0027] Account restore for temporary factor disability creates more opportunities for human error since the human has authority over modification and re-enrollment of factors. Support personnel need to reset user accounts often, therefore the process must be easy and fast, and is therefore also easier to social engineer. Even a temporary loss of access to any factor (forgot a 2-factor authentication (2FA) phone at home, eye infection prevents a retina scan today) requires factor reset, which creates undue friction for the user, and effort for IT teams. This makes the restore transaction a process that is heavily used, creating a path for attackers to gain entry through a routine process.

[0028] This H2F2 solution, while complex, creates a solution to a problem that is a common occurrence and heavily negative in its impact.

[0029] III. <u>This Disclosure Improves on Existing Solutions</u>

[0030] This disclosure allows organizationally trusted humans to act as "Verifiers" and use physical devices to digitize their verification of a known user's identity. This does not grant account reset but serves as a stand-in for a single factor of authentication in a single, complete, multi-factor authentication transaction: use H2F2 authentication as a temporary biometric.

[0031] This disclosure is more secure since the human may only grant passage for a single factor of a multi-factor challenge: a user must still have access to at least one factor to be able to access the system. It provides better ease of use since every issue affecting only a single factor does not generate an unnecessary reset process. This results in fewer helpdesk tickets and overhead caused by repeated resets. It also allows multi-factor authentication in environments where only 2 out of 3 factor categories are available, for example: (1) true passwordless authentication where "something you know" cannot be used; or (2) clean rooms where mobile devices are not allowed and "something you have" cannot be used.

[0032] Currently, for such scenarios, MFA is difficult to impossible to implement due to the cost of IT support and maintenance of supporting account resets, as well as the downtime accrued from failed authentication.

[0033] This disclosure leverages H2F2 authentication to create a way to make passwordless authentication usable. Without it, account resets become a liability, since a reset is the only recourse for any single factor failing, even if only temporarily. This puts an undue burden on the user, an undue burden on the IT/support desk, and costs significant downtime, all of which make passwordless authentication unaffordable.

[0034] The following describes the Components and Steps that are required for the disclosure.

[0035] Figure 1 shows a schematic 100 of such various components:

- Component 1 (C1) - User Device 106
- Component 2 (C2) - Authentication Orchestrator 102
- Component 3 (C3) - Request Certification Application 105
- Component 4 (C4) - Verifier Device 116
- Component 5 (C5) - Multi-factor Login Application 108
- Component 6 (C6) - Approval Certification Application 118
- Figure 1 also shows: (a) a human user 110 engaging in H2F2 112 with a human verifier 114; (b) a network 104 connected to C2 102, C6 118, and C4 116.

[0036] C1 106 is an interactive User Device, like a laptop, desktop, mobile device, ATM, etc., any device that a user may interact with and has the ability to authenticate a user. It also has the required hardware and software for network connectivity. It allows a user to authenticate, and then do work of some kind using native functions and applications.

[0037] C2 102 - The Authentication Orchestrator is a system or application that implements logic to orchestrate the steps in Figure 2 for H2F2. It also has access to or contains a database containing user ID tuples that map each User ID to Verifier IDs such that for each user, there is an assigned Verifier. C2 102 also contains the logical implementation to take the action required for Verifier Approval and Denial. C2 102 functions to trigger and route verification requests, validate responses, and then trigger the correct actions based on the contents of the response.

[0038] C3 105 - The Request Certification Application is a custom software application with a hardware interface that runs on the User Device (C1 106) for the purpose of informing the user of the state of the Approval Request and providing Certification information to C6 118

on the Verifier Device. It also contains a user interface informing the user of the stage and outcome of the verification process. It also implements a process to validate the certification that the Verifier correctly verified the correct user for the correct transaction.

[0039] C4 116 - Verifier Device is an interactive device similar to C1 106 with the addition of a camera. Its purpose may be varied and include the functions of C1 106, but its purpose for this disclosure is to validate that the verification took place correctly and record the Verifier's response to the verification request.

[0040] C5 108 - Multi-factor Login Application is a software application requiring the user to correctly authenticate using 2 or more factors from at least 2 different factor categories. It implements the hardware/software interfaces required for factor input collection (a keyboard or mouse for a password or pass gesture, a fingerprint sensor for biometrics, a USB port or RFID reader to interface with key material on hardware token, etc.). C5 108 also contains the data and secrets to validate the success or failure of all traditional factors, using software (e.g., comparing the hash of a password typed on a keyboard) and hardware (e.g., the secure enclave containing the biometric values of an authorized and enrolled user) and the hardware/software capabilities to evaluate correctness.

[0041] C5 108 also contains the implemented logic to communicate with C2 102 upon failure of the biometric factor, to trigger an H2F2 flow, launch the Request Certification Application (C3 105) if needed, and the implemented logic to receive the approval or denial signal. It also contains the implemented logic to take the correct action (grant access or deny access) based on that signal. Its function is to ensure that a User has completed an MFA challenge before the user may access the system, send the signals and run the logic necessary for a verification request from the user's side, if necessary, grant access on success, and deny access otherwise.

[0042] C6 118 - The Approval Certification Application is a custom software application with a hardware interface that runs on the Verifier Device for use by the Verifier to run the approval process and collect information for certification. The device has input devices and connectivity for ascertaining the outcome of the verification from the authorized Verifier. It also has the implementation of the logic to gather certification information, record the Verifier's decision, and send that information to C2 102 for validation.

[0043] Figure 2 shows a flowchart 200 of the following Steps:

- Step 1 (S1) - User enters credentials 202
- Step 2 (S2) - User submits biometric 206

- Step 3 (S3) - Send verification request to Verifier 210
- Step 4 (S4) - Verify User ID and validate 212
- Step 5 (S5) - Verification and validation check 218
- Step 6 (S6) - User logs in 214
- Step 7 (S7) - User is locked out 222

[0044] S1 202 - The User initiates a login attempt, by entering the correct credentials, such as username and password.

[0045] If the credentials are incorrect, repeat this step.

[0046] If the credentials are correct 203, proceed to S2 206

[0047] S2 206 - The user submits their biometric information for authentication by, for example, placing or swiping their finger on the fingerprint sensor.

[0048] If the biometric authentication succeeds 204, proceed to S6 214 as multi-factor authentication has been successfully completed.

[0049] If the biometric fails 208, proceed to S3 210 for H2F2 factor authentication.

[0050] S3 210 - The authorized Verifier is identified, and a verification request is sent to the Verifier.

[0051] S4 212 - The Verifier verifies the user H2F2 and collects all information required to certify the verification.

[0052] The verifier inputs their Verification decision into the system.

[0053] Proceed to S5 218

[0054] S5 218 - The certification information from a verification request response is validated and the response itself is checked.

[0055] The certification is information generated by the Multi-factor Login Application (C5 108) indicating the User account, device, and the MFA transaction ID as a random one-time password (OTP), which is provided to the Verifier via the Approval Certification Application (C6 118) and to the Authentication Orchestrator (C2 102).

[0056] Validation is the process, conducted by the Authentication Orchestrator, of checking that the Verification Response matches the Request, i.e. that the Verifier verified the right user account on the right device and for the right MFA transaction.

[0057] If the verification and the validation are successful 216 (the response is "Approve" and the Certification is Approved), proceed to S6 214, otherwise 220, proceed to S7 222.

[0058] S6 214 - The User is logged in and has been multi-factor authenticated.

[0059] S7 222 - The User has been identified as a threat actor and has been locked out as an initial remediation.

[0060] Figure 3 shows a schematic 300 of an MFA and Request Certification Application from the user's point of view. When a user tries to authenticate, a message is displayed informing the user that manager approval is required 302. If the manager authenticates using, for example, H2F2, the user is informed that the request is approved 304 or denied 306. Alternatively, the user is informed that the request has expired 308.

[0061] Figure 4 shows a schematic 400 of an Approval Certification Application from the verifier's point of view. When a user tries to authenticate, a verifier receives a

[0062] In S1 202, the User enters their username and password into the multi-factor login application (C5 108) on the User Device (C1 106). C5 108 takes that information and validates that the password is correct for the user account using the APIs that run the validation against the information stored locally. If that validation does not succeed, C5 108 informs the user of failure and reverts to its original state.

[0063] If the validation succeeds, C5 108 then progresses to S2 206 to request biometric authentication using the APIs associated with the biometric fingerprint reader embedded in C1 106. If biometric authentication succeeds, C5 108 logs the user in, granting access. If biometric authentication fails, C5 108 then generates the Verifier Approval Request to send to the Authentication Orchestrator (C2 102) and launches the Request Certification Application (C3 105) with the requisite information for certification. C2 102 then identifies the correct Verifier for this process by querying its mapping database and sends the Verification request to the Verifier's Approval Certification Application (C6 118) on the Verifier's Verifier Device (C4 116).

[0064] The system then progresses to S4 212 where the Verifier verifies the User's identity. First, the Verifier certifies that the request they are verifying is the intended MFA transaction (the right user account on the right device) through an interaction between C6 118 and C3 105. The Verifier then proceeds to have a human-to-human, face-to-face interaction with the User. Based on prior knowledge of and interactions with the User, a shared history of interaction, the use of their biological senses, assessing shared context and history from those interactions, basically recognizing the User intuitively as humans do, the Verifier makes a judgment call about the voracity of the User's identity matching the identity of the account. That judgment is entered into C6 118 (approve or deny) which is then returned to the Orchestrator C2 102 and the process moves forward to S5 218.

[0065] If the result is 'Approve', the Orchestrator C2 102 informs the MFA Login app (C5 108) of success. C5 108 then terminates C3 105, and grants access to the User.

[0066] If the result is 'Deny', the Orchestrator C2 102 informs C5 108 of failure, which terminates C3 105 but then locks the device until further instructed by C2 102.

[0067] Any of the steps in Figure 2 may revert to any other step on success or failure, rather than simply failing closed to "locked out" or succeeding to open "logged in". For example, if H2F2 times out, fails validation, or is denied by the Verifier, the system may fail back to S1 202, presenting the user with the prompt to enter their credentials and try again. For another example, successful H2F2 Verification could transition to S2 206, requiring another form of biometric or H2F2 verification before proceeding to S6 214. Any of these loops may be bound by minimum or maximum transition counts. For example, a User may be allowed to fail H2F2 Verification a certain number of times before being locked out, or pass a certain number of H2F2 verifications before being signed in.

[0068] If the Verifier approves but Certification fails to validate in S5 218 (e.g., User, Device, or OTP does not match the Request), the system may revert back to the H2F2 step S3 210, or any other step, rather than failing closed (locked).

[0069] IV. Building the Disclosure

[0070] The following is a guide to a simple yet effective way to build this disclosure.

[0071] C1 106:

1. Acquire a commercial, off-the-shelf (COTS) Windows 11 device with a USB biometric fingerprint scanner and authenticator, also COTS available, that leverages the Trusted Platform Module (TPM) for authentication.
2. Create a local account for the User on the device and set a password.
3. Enroll the User's fingerprint in the fingerprint reader.
4. Note the User account's security identifier (SID) for use later

[0072] C2 102:

1. This application will be implemented as a server-based application with an API on the network, or on a local computer.
2. Create a database containing a data table, mapping the User's SID to the Verifier's SID.
3. Create a user interface to allow User and Verifier SID tuples to be entered in the data table of the database.
4. Create a web API to receive a call from an application (C5 108) running on the User Device (C1 106) to request a verification.

5. When a request is received, take the User's SID and use the database mapping to find the Verifier's SID. Create a routine to execute an API call to send a verification request to the registered Verifier application (C6 118) on a Verifier Device (C4 116) for the Verifier specified by their SID.
6. When a response is received from a Verifier, implement the Certification Validation as follows: comparing the information from the Verification Request from C5 108 and the Verification Response from C6 118, check that the User's user ID and device ID are identical, and that the OTP matches. If so, then validation succeeds, otherwise it fails.
7. Implement the logic to return the success or failure of the verification to the application (C5 108) requesting verification. Success is returned when validation is successful (the response is for the correct User ID on the correct User Device, with the correct OTP), and the Verifier's response is "Approve", otherwise failure is returned.

[0073] C3 105:

1. This is a custom application that is implemented as a user view in the H2F2 Custom Credential Provider of C5 108.
2. Create a single window non-interactive application that generates and displays a QR code on the screen of the User Device (C1 106).
3. Receive the OTP, User Device SID, and User SID from the Multi-factor Login (C5 108) application.
4. Append the OTP, User Device SID, and the User's user SID, separated by a delimiting escape character, and encode that string into the QR Code.
5. Wait until terminated by the Multi-Factor Login Application (C5 108) on Verification completion.

[0074] C4 116:

1. Acquire a mobile Windows 11 device, such as a Surface Pro.
2. Connect the device to the network and enable the camera.
3. Set up the Verifier's account, enrolling the Verifier with the desired level of security (e.g., password only, Windows Hello, or other form of MFA).
4. Note the SID for this account and add it to the mapping in the Authentication Orchestrator in C2 102 as the Verifier SID authorized for the User's SID.

[0075] C5 108:

1. This component is a custom application and therefore the MFA authentication flow will need to be created from scratch to implement H2F2 verification as a factor. This

Component is the most difficult to implement due to low-level APIs, hardware interfaces, and lack of documentation.

2. Create a new Custom Credential Provider that has the capability to wrap other credential providers (Custom Credential Provider Wrapper, CCPW) which will implement the logic associated with the different factors and the H2F2 factor.
3. Each factor, including the H2F2 factor, will now require its own Credential Provider that will be wrapped and orchestrated by the CCPW.
4. Implement the CCPW to wrap the native Windows Password Credential Provider for primary authentication using the User's enrolled username and password and display it as the default.
5. Implement another Custom Credential Provider to request a fingerprint from the User using the USB fingerprint reader, and to use the embedded TPM to evaluate the success or failure of the biometric authentication.
6. Implement another Custom Credential Provider for the H2F2 factor to send a verification request containing the User's SID, a unique device identifier (device SID), and a randomly generated One-time-password (OTP) to the Authentication Orchestrator (C2 102). Also, implement this Credential Provider to implement and launch the Request Certification Application (C3 105) as a new user view when the Verification Request is sent, and pass the OTP, User SID, and device SID to the Request Certification (C3 105) Application as an argument. Implement this Credential Provider to wait until either the Verification Request returns success or failure, or the Verification request times out.
7. Now implement the CCPW to orchestrate the following steps.
8. First, the CCPW should by default display the Windows Default Credential Provider. If the Windows Default Credential Provider returns failure (wrong username or password entered), reset to the Windows Default Credential Provider again.
9. If the Windows Default Credential Provider returns success (credentials correct), implement the CCPW to launch the Fingerprint Custom Credential Provider. If the Fingerprint Custom Credential Provider returns success (the fingerprint was a biometric match), implement the CCPW to log the user in.
10. If the Fingerprint Custom Credential Provider returns failure (the User is not biometrically authenticated) Implement the CCPW to launch the H2F2 Credential Provider.

11. If the Verification Request API call response is "Approve" (the Verifier approved and the Certification was Validated) the Request Certification Application (C3 105) is terminated, and the user is logged in.
12. Implement the CCPW such that if the Verification Request API call response is "Deny" (the Verifier denied or the Certification Validation failed) and the H2F2 Credential Provider returns failure, or the request times out, the system is locked, and no further action is possible on the part of the User.

[0076] C6 118:

1. This component is a custom application.
2. Create a native application for the Verifier Device C6 118 using the logged-in Windows user identity (the Verifier's account).
3. Create a sign-in flow for the Verifier to authenticate to the application using their Windows identity.
4. Implement necessary security measures to ensure that only the authorized Verifier may sign into the application under their account.
5. Implement the flow that on sign-in, the application registers itself with the Authentication Orchestrator (C2 102) to receive Verification Requests meant for the Verifier's user ID.
6. Implement the application to alert a Verifier with relevant user information about the User and User Device (C1 106) so they may begin the verification process.
7. Implement the application to use the camera to scan a QR code from the Request Certification Application (C3 105) and extract an OTP, User Device SID, and the User SID of the user from the string in the QR code.
8. Once the scan is complete, implement a user interface (UI) with an "Approve" and a "Deny" button.
9. Once the buttons are pressed, return the Verification Request API call to the Authentication Orchestrator (C2 102) with the OTP, the User Device SID, the User's user SID, the Verifier's user ID, and the result of Approve or Deny based on the Verifier's selection.

[0077] V. Alternatives

[0078] The components that are necessary to implement this are:

1. An Authentication flow with at least one factor on a physical interface protecting access to a physical or digital resource for use by a User.

2. A method of hardware or software, allowing a second, trusted Verifier to intercede on the user's behalf and grant them access even on failure of one or more factors.

[0079] The Request Certification Application may be made optional, integrated directly into C5 108, or entirely eliminated, making approval entirely at the Verifier's discretion.

[0080] The Verifier may be required to perform MFA themselves to sign in to their Verifier Device, or the Approval Certification Application.

[0081] Verifier Device and Approval Certification Application may be implemented as a web app, or not as an application at all, but rather through email or some other form of message-based communication.

[0082] The Approval Certification Application may be implemented as an application within a communication platform such as Slack or MSFT Teams for example, or as a component within another application entirely such as a workforce management application.

[0083] Alternatively, the H2F2 disclosure may also serve its purpose for cloud-based applications, and be partially or entirely implemented as any combination of cloud applications (e.g., C2 102 may be easily implemented as a cloud service), as web applications (e.g., MFA Login C5 108, Request Certification Application C3 105, or Approval Certification Application C6 118 as web pages), or virtual devices (e.g., User Device C1 106 or Verifier Device C4 116).

[0084] H2F2 verification may be used for verification of "something you have" where a Verifier verifies the user has something unique to them, like an ID badge or an item with a symbol on it.

[0085] Alternatively, the Verifier simply verifies the user is the authorized user, but the verification is used as a stand-in for 2FA.

[0086] The H2F2 verification step may also be used to pass a user on the biometric or other factor challenge, and also allow a user to reset/re-enroll one, more, or all factors, or both, where the User is granted access and from there may re-enroll the unusable factor.

[0087] Multiple biometric modalities may be used in sequence or in parallel. One example is using facial recognition first, then defaulting to a thumbprint if facial recognition fails, then only defaulting to H2F2 verification by a human if the fingerprint authenticator fails.

[0088] H2F2 verification may be used as an additional factor rather than a replacement, based on some other risk factor or criteria, or as the normal mode of operation.

[0089] H2F2 verification may also be considered its own type outside of know/have/are - a whole new factor category "something you are recognized to be." This factor could be a factor in a multi-factor authentication in combination with any one of the traditional factor

categories, creating an MFA transaction in combination with another biometric factor, a Password, or a "Something you have" factor like a hard token.

[0090] H2F2 verification may also be used as the only factor for authentication, as a single-factor authentication flow.

[0091] A system may automatically fall back to H2F2 if one or any factor fails, automatically requesting the H2F2 factor. Alternatively, the User, Verifier, or admin may also be allowed to request the use of the H2F2 factor.

[0092] The system could be implemented such that each User is uniquely mapped to 1 Verifier (1-to-1 mapping).

[0093] Alternatively, each user could be mapped to multiple Verifiers, all of whom would be required to verify the User for a single MFA transaction (1 to N).

[0094] Many Users may also be mapped to a single Verifier (N to 1),

[0095] Alternatively, a User may be mapped to multiple Verifiers but the system may only require verification from any one Verifier, or a subset of them, or all of them.

[0096] Alternatively, one User could be mapped to a Pool of Verifiers (1 to N) or a pool of many Users (e.g., a team) could be mapped to a pool of many Verifiers (N to N).

[0097] Verifiers may be selected at random, using a "round robin" approach, or requests may be sent to some or all Verifiers, and any Verifier available could respond.

[0098] Any combination of specific Verifiers (1 to 1) and pools of Verifiers (N to N) are possible, which each using any combination of AND and OR logic of required Verifiers and number of Verifications before MFA is considered complete.

[0099] Verifiers could be a group with a specific organizational or cultural role, for example, managers of a work crew or team may be the Verifiers, Or Verifiers could be peers where teammates may act as Verifiers for each other.

[00100] Alternatively, Users may be able to elect their own Verifier or Verifiers, for example in a scenario where this disclosure is implemented and used for the consumer userbase of a B2C application.

[00101] Alternatively, H2F2 Verification may be used to grant access to someone other than the authorized User, for example, when granting a family member access to the social media account of a deceased relative.

[00102] There may be several forms or mediums of User-Verifier interaction. The example given in this disclosure is H2F2 communication. Alternatively, communication and verification may occur over a video call using C1 106 or another device, over an audio call

through the phone, VOIP, through a text-based conversation, in virtual reality, or any other form of physical or digital communication using C1 106, C4 116, or other devices.

- Alternatively, user verification may occur without interaction with the Verifier. For example, it may use video monitoring feeds of the User, C1 106, and its environment, or video monitoring feeds generated by a camera on C1 106 or C4 116.
- There may be alternative ways of limiting verifier trust and error with certification using systems to ensure verification happened for the correct transaction, device, and user. The example in this disclosure was given using QR code scans but any other auto ID methodology may be used. Alternatively, it could use:
- Another form of stenography in C3 105;
- An additional scan of a User's ID badge;
- An OTP sent to the Verifier for entry by the User or Verifier into C5 108 or C3 105,
- An OTP displayed on C3 105 needed by the Verifier to initiate approval;
- An NFC or RFID chip containing key material unique to the User, Verifier, C1 106 or C4 116 that is scanned/read by the C6 118 or C3 105;
- A number-matching solution where C3 105 displays a number, letter, word, or image, and C6 118 displays several options including the one shown on C3 105 for the Verifier to select correctly;
- Number-matching with C6 118 and C3 105 roles reversed for the User to select correctly;
- Environmental audio fingerprinting on C3 105 and C6 118;
- Environmental wireless fingerprinting and matching on C3 105 and C6 118;
- Bluetooth communication, Wi-Fi communication, ZigBee or other peer-to-peer communication protocols;
- Values encoded in tones played by and recorded on C6 118 and C3 105 for decoding;
- Verification documentation, such as a badge, ID cards, photos, audio/video recordings or conversation logs; or
- Any other method that allows User and Verifier to be attributed to specific MFA transactions at Certification Validation time.

[00103] C6 118 may also incorporate the ability to match the User with their physical documentation, such as biometrically matching a real-time picture of the user with their photo ID badge, or their passport photo, or a photo on record stored on a component such as C2 102, another component, or a remote resource.

[00104] Certification may also require H2F2 authentication of some kind in which the User and Verifier must authenticate themselves to each other in an H2F2 way, e.g.:

- A shared secret between User and Verifier;
- Secret handshake equivalent or passphrase of the day; or
- The User or Verifier proving "something they have" that need not be digital or have computing capabilities, such as an object, ID badge, etc.;
- Using augmented reality with facial recognition - the Verifier could authenticate the User with a body-worn or head-up camera, or Verifier Device, or the other way around with the User's system authentication the Verifier; or
- Any other method that allows one person to authenticate another.

[00105] Request Certification could also be implemented as a Verifier authentication flow that could be single factor or multifactor and be implemented in the Request Certification Application on the User Device, or on the Verifier Device. For example, the Verifier could enter a password, enter their biometric information, or present a cryptographic key to the User Device.

[00106] This could also be implemented using hardware/certificate-based authentication. For example, this includes where Verifiers have a hard token - certificate-based 2FA authenticator that the Verifier may use in C5 108 on the User Device to certify their Verification.

[00107] Approval Certification Application may be omitted and implicit, e.g., simply certifying the request by scanning the code, or Verifier authentication, implicitly grants approval, and the lack thereof is denial (i.e. timeout and approval are the only possible outcomes).

[00108] The certification process may be omitted completely, allowing a Verifier to verify an MFA transaction solely by approving, without an implemented Certification Application on either the User or Verifier Device. One such example would be Verifications sent to the Verifier via text message or email with one-time links for Approve and Deny.

[00109] A system may use human verification at the user's request, or force the approval flow:

- Always;
- Randomly;
- After a factor fails once;
- After a factor attempt fails a set number of times;

- After multiple factors fail once or a number of times; or
- After an MFA transaction fails once or a set number of times.

[00110] H2F2 may also be used as a stand-in for a factor that has been un-enrolled, or has not yet been enrolled. For example, it could be used as a factor protecting the enrollment of a biometric factor for a new user that only has a password or temporary password, or other factor or factors.

[00111] For multiple authorized verifiers, the system may:

- pick a specific verifier at random or based on criteria;
- have a pattern or sequence in which verifiers are selected;
- also time out if a verifier does not respond;
- cycle to another verifier, at random or based on criteria, if verification fails or times out; or
- also be configured to grant user access if a verification request times out or expires.

[00112] This disclosure may be configured to grant access and send the Verification Request simultaneously, allowing the User to perform normal duties and actions by default but terminating the session if the Verifier clicks Deny.

[00113] This disclosure may also be configured to grant access even on Verifier denial. For example, it would monitor tactics, techniques, and procedures of an attacker in a honeypot scenario.

[00114] The User and Verifier Devices could be the same device, e.g., with a different flow for the Verifier to authenticate themselves, possibly with MFA, and approve user access in situ. This may occur with no need for Verifier Device or approval application. In fact, the entirety of this disclosure and all components thereof could be implemented entirely on the User Device (C1 106) as a single component that is completely offline, e.g., an application with User MFA and Verifier MFA for a single User and Verifier. Such use would not even require network connectivity.

[00115] Multiple methods and forms of H2F2 may be used in parallel. For example, the form of H2F2 implemented entirely on the User Device (C1 106) could be used as a backup offline, e.g., during a network outage, where the default mode uses a Verifier Device and Orchestrator.

[00116] This disclosure may serve its intended purpose for a User Device, but also for a virtual device, web application, virtual application, application, or system in virtual or augmented reality, as well as physical access control scenario. This disclosure may be used

for device access, on-device application access, cloud application access, SSO access, or anywhere else that authentication or MFA is required.

[00117] All or some devices may be virtual, e.g., the User and Verifier applications running entirely virtually. Some or all Factors in the MFA transaction may be virtual as well (e.g., behavioral biometrics on a virtual machine). Physical devices or thin clients may implement all or part of user and verifier applications

[00118] H2F2 verification may be used in lieu of multiple factors and multiple categories. For example, in an MFA flow require something you know, you have, and you are. Verifier approval may be deemed sufficient to grant access to a user with an authorized hard token, who can't remember the password or pass the biometric check.

[00119] Depending on the level of trust an organization puts on the Verifiers, the systems for validating approval certifications may be weakened or omitted.

[00120] The H2F2 factor may be reused mid-session. At any point in time during use, e.g., randomly, based on a tip of bad actions, if suspicious behavior is detected, or if a continuous biometric flags unauthorized use, a User Verification Request may be sent to a Verifier mid-session as a method of investigation. This Verification Request may be silent (invisible to the User) and undetectable unless the Verifier initiates the Deny sequence. The User may also be notified and the Request Certification Application re-initiated (e.g., in the corner of the screen while normal usage and work continues), Or the system may be locked and returned to Step 3 in Figure 2, only to be unlocked by approval from a Verifier.

[00121] The Verifier actions of Approve and Deny may be connected to various other systems and processes such as employee time cards, risk, and fraud detection engines, physical security forces and alerting systems, event logging systems, endpoint security, network security, cloud security systems, productivity software, etc., which may take actions or make use of that information.

[00122] A User in one transaction that requires a Verifier, may be used to serve as a Verifier in another, and a Verifier may themselves require a Verifier in another transaction, becoming a User. User and Verifier may be ephemeral roles at a point in time assigned dynamically based on organizational role, availability, or any other logic. Alternatively, the role of Verifier may only apply to certain user accounts as an authorization or entitlement.

[00123] This disclosure may be constructed without an Authentication Orchestrator (C2 102) as a separate unit. It may be implemented in a single component, or entirely within C4 116 or C5 108.

[00124] This disclosure may also be implemented in tandem with further identity security systems such as Single Sign-On, 3rd party multi-factor authentication providers, passwordless authentication solutions, reverse proxies, etc.

[00125] This disclosure works with varying authentication systems relying on any number of factors, from any number of factor categories. The logic of the order of factor use, the number of factor and factor categories and retries a User is entitled to before Verification, the number of factor successes required for successful authentication, the number of factors within an MFA transaction that requires H2F2 verification, the number of independent Verifiers required per factor, all of these parameters are dependent on the configuration of the system but do not impede functionality or utility of this disclosure.

[00126] A Verifier may require H2F2 verification from someone else before being entitled to act as a Verifier, either temporarily, or permanently.

[00127] H2F2 may also be used for authorization. For example, to become an authorized Verifier, a single H2F2 verification may be required in addition to a successful MFA, to change the status of a User to a Verifier and authorize their device as a Verifier Device.

[00128] H2F2 Verification may be used to grant resource access authorization, e.g., a User may be required to pass H2F2 Verification before getting access to a sensitive resource (either temporarily for a period of time or permanently). This may involve an MFA transaction, single factor step-up, with an additional H2F2 verification, or it may simply be an H2F2 verification on an authenticated session.

[00129] This disclosure could also serve as an aid in for KYC where an existing trusted user or customer could serve as a Verifier for a new user signing up.

[00130] This disclosure may also be used to Notarize an authentication or transaction, with the Verifier role being filled by a licensed Notary Public. This would also apply to other types of digital transactions that require the participation of an official role, such as a virtual wedding with an officiant acting as a Verifier, or other situations where the prestige, title, or authority of the Verifier adds value or import to the record of the transaction.

[00131] Two Users' authentication simultaneously could also serve as Verifiers for each other, requiring pairwise Verification as a security precaution, for example, where some critical action should be protected by the "two-key principle", akin to the physical dual authentication with physical keys required for a missile launch.

[00132] Alternatively, this disclosure could be used in a way where a User may elect to enable and enforce H2F2 Verification as an additional factor for every authentication, for example as extra security for a high-value financial account.

[00133] Any component in the H2F2 system may fail or be unreachable, for example, due to a network failure, software error, device failure, or a dead battery. The system behavior for a Deny, Certification Validation failure, Verification Request Timeout, and any other type of failure may each be configured to have a different outcome, for example, granting access if the Verifier Device is unreachable, but locking the User Device on Deny.

[00134] Each factor in the MFA challenge, including H2F2, as well as the entire MFA challenge, may be configured with different timeout periods or expiration times independently, and each of those timeouts may be configured to have a different outcome arbitrarily, e.g., reverting to the Password screen when the fingerprint factor times out, but reverting to the fingerprint factor retry and when H2F2 times out.

[00135] VI. Usage of the Disclosure

[00136] To use this disclosure as it is intended, deploy it as described across the employee workstations of a company or team, and add the authorized employee accounts to the User group. Then designate a trusted set of those users as Verifiers, probably either shift supervisors or the IT helpdesk employees and add them to the Verifiers group. The Verifiers now have the authority and ability to keep business processes and User work moving even through temporary limitations of access to single factor, issues that would normally disrupt an entire work day for a User or group of Users. Depending on the factors being used, adverse impacts would be caused by routine events such as forgetting a phone at home (disrupts mobile MFA push challenges), wet hands, injured finger or wearing surgical gloves (disrupts fingerprint scanner usage) or wearing a facemask (disrupts facial recognition).

[00137] By being able to withstand the temporary failure of any single factor without requiring User and IT time and effort to reset all or some of the factors, a multi-factor authentication system that only relies on 2 out of 3 factor categories becomes feasible, for example, passwordless authentication where "something you know" is eliminated and every MFA must rely on something you "have" and "are".

[00138] A. Configuration.

- To begin, deploy the two end-user devices C1 106 and C4 116, create the User and Verifier accounts respectively, and record the security identifiers (SIDs) for each.
- Deploy the Authentication Orchestrator C2 102.
- Add the User's SID to the User-Verifier mapping with the Verifier's SID.
- Enroll the User on the User Device with their password and fingerprint and install C5 108/C3 105 on the User Device.

- Enroll the Verifier on the Verifier Device and install C6 118, and have the Verifier sign in to C6 118.

[00139] Usage

[00140] When the User fails biometric authentication (false negative), C5 108 launches C3 105 and alerts C2 102 of the Verification Request. That alert contains the User SID, the Device SID, and an OTP designating this MFA transaction. The Orchestrator queries the User-Verifier mapping in the database, finding the Verifier's SID. The Orchestrator then sends the request to the Verifier Device, which launches C6 118, informing the Verifier of the request, which user account the request is for, and which User Device generated the request. Using that information, the Verifier proceeds to locomote to the User with their Verifier Device in hand, and using C6 118, scans the code presented by C3 105, thereby C6 118 now has the User SID, User Device SID, and the MFA transaction OTP. Verifying that this is the correct person for the User ID, the Verifier then clicks the Approve button. C6 118 then sends the notice of Approval to the Orchestrator C2 102, containing the User ID, the Device ID, and the OTP and the Approval to the Orchestrator C2 102.

[00141] The Orchestrator C2 102 may now validate the Approval Certification, validating that the User ID and Device ID match the request (i.e. the Verifier is validating the correct User on the correct device), and that the OTP matches the request (the Verifier verified and approved the correct MFA transaction). If validation succeeds, C2 102 sends the Access Granted signal to C5 108 which kills the C3 105 application and unlocks the endpoint. If validation fails, the orchestrator does not send the Access Granted signal since the Verifier has attempted to approve a Verification Request for the wrong User, User Device, or the wrong transaction.

[00142] If the Verifier clicks Deny and the Orchestrator C2 102 validates the Certification, C2 102 sends a Lock signal to User Device. C2 102 also notifies security personnel and infrastructure that a breach attempt has been detected and remediated, but that an investigation is warranted, and immediate further steps may be needed.

[00143] VII. Other Applications

[00144] This disclosure also solves the problem of deploying multi-factor authentication in a clean room or Sensitive Compartmented Information Facility (SCIF), where technology cannot be brought in or out, and "something you have" is removed from the equation. MFA, therefore, relies on a password and a biometric. However, biometrics often produce false negatives, where an authorized user is rejected. False negative rates vary from user to user,

and often reflect the biases of the manufacturer with respect to diversity in the training data. Even if rare, the non-determinism in the authentication system may be prohibitive to the deployment of strong MFA and good security practices. This disclosure solves that problem by providing a secure and reliable workaround for biometric false rejects in a clean room environment.

[00145] VIII. Other Benefits

[00146] One of the incidental benefits of this disclosure is the unequivocal and immediate detection of an attacker. Traditionally, MFA challenges fail all the time and there is little to no way for IT and InfoSec teams to know the difference between an abandoned MFA challenge and an attempted intrusion. Verifier Approval immediately detects unauthorized access attempts: a 'Deny' result indicates that a breach attempt was detected and remediated and may be escalated within the organization for investigation.

[00147] Another incidental benefit is that traditionally, biometrics have an enrollment and authentication phase, and biometric data is immutable after enrollment. This is a security precaution to prevent an unauthorized attacker from inserting their biometric data into the data store. Verifier Approval provides a label, or ground truth that a biometric rejection was in error and actually a false rejection. This Labeling may provide training data for continuous performance improvement of biometric authenticators that now get better over time.

[00148] Verifier approval may also provide significant cost savings, where certain factors are expensive for environmental or business process reasons. With Verifier approval it becomes viable to implement MFA with only 2 factors, eliminating the cost of a third factor category that would be otherwise needed as a backup.

[00149] There may also be benefits with respect to accessibility as certain types of disabilities make it difficult or impossible to use certain types of authentication factors reliably. At the same time, cost reasons may make it prohibitive to make authentication policy exceptions. Allowing humans to provide aid may give certain secure systems wider accessibility, especially with compliance mandates that are unbending.

[00150] This disclosure enables multi-factor authentication enrollment for a new user, requiring only a single factor (e.g., initial password) and Verifier approval. This solves security issues with enrollment and re-enrollment, for example, if a factor, or entire account, reset is required, H2F2 may be used to make an otherwise dangerous process much more secure and may protect MFA enrollment with MFA - e.g., with a temporary password and Verifier Approval.

[00151] Further incidental benefits:

- Reduction in helpdesk tickets from reducing account resets;
- Reduced resets mean less User confusion, e.g., needing to remember a new password after a reset, leading to a further reduction of resets and helpdesk tickets; and
- Productivity improvements by reducing resets, but also removing downtime from locked-out users.

[00152] IX. Conclusion

[00153] While the foregoing descriptions disclose specific values of voltage, capacitance and current, any other specific values may be used to achieve similar results. Further, the various features of the foregoing embodiments may be selected and combined to produce numerous variations of improved haptic systems.

[00154] In the foregoing specification, specific embodiments have been described. However, one of ordinary skill in the art appreciates that various modifications and changes may be made without departing from the scope of the disclosure as set forth in the claims below. Accordingly, the specification and figures are to be regarded in an illustrative rather than a restrictive sense, and all such modifications are intended to be included within the scope of present teachings.

[00155] Moreover, in this document, relational terms such as first and second, top and bottom, and the like may be used solely to distinguish one entity or action from another entity or action without necessarily requiring or implying any actual such relationship or order between such entities or actions. The terms "comprises," "comprising," "has", "having," "includes", "including," "contains", "containing" or any other variation thereof, are intended to cover a non-exclusive inclusion, such that a process, method, article, or apparatus that comprises, has, includes, contains a list of elements does not include only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus. An element proceeded by "comprises ... a", "has ... a", "includes ... a", "contains ... a" does not, without more constraints, preclude the existence of additional identical elements in the process, method, article, or apparatus that comprises, has, includes, contains the element. The terms "a" and "an" are defined as one or more unless explicitly stated otherwise herein. The terms "substantially", "essentially", "approximately", "about" or any other version thereof, are defined as being close to as understood by one of ordinary skill in the art. The term "coupled" as used herein is defined as connected, although not necessarily directly and not necessarily mechanically. A device or structure that is

"configured" in a certain way is configured in at least that way but may also be configured in ways that are not listed.

[00156] The Abstract of the Disclosure is provided to allow the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, it may be seen that various features are grouped together in various embodiments for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus, the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separately claimed subject matter.

CLAIMS

1. A system comprising:

a User Device component;

an Authentication Orchestrator component;

a Request Certification Application component;

a Verifier Device component;

a Multi-factor Login Application component; and

an Approval Certification Application component.

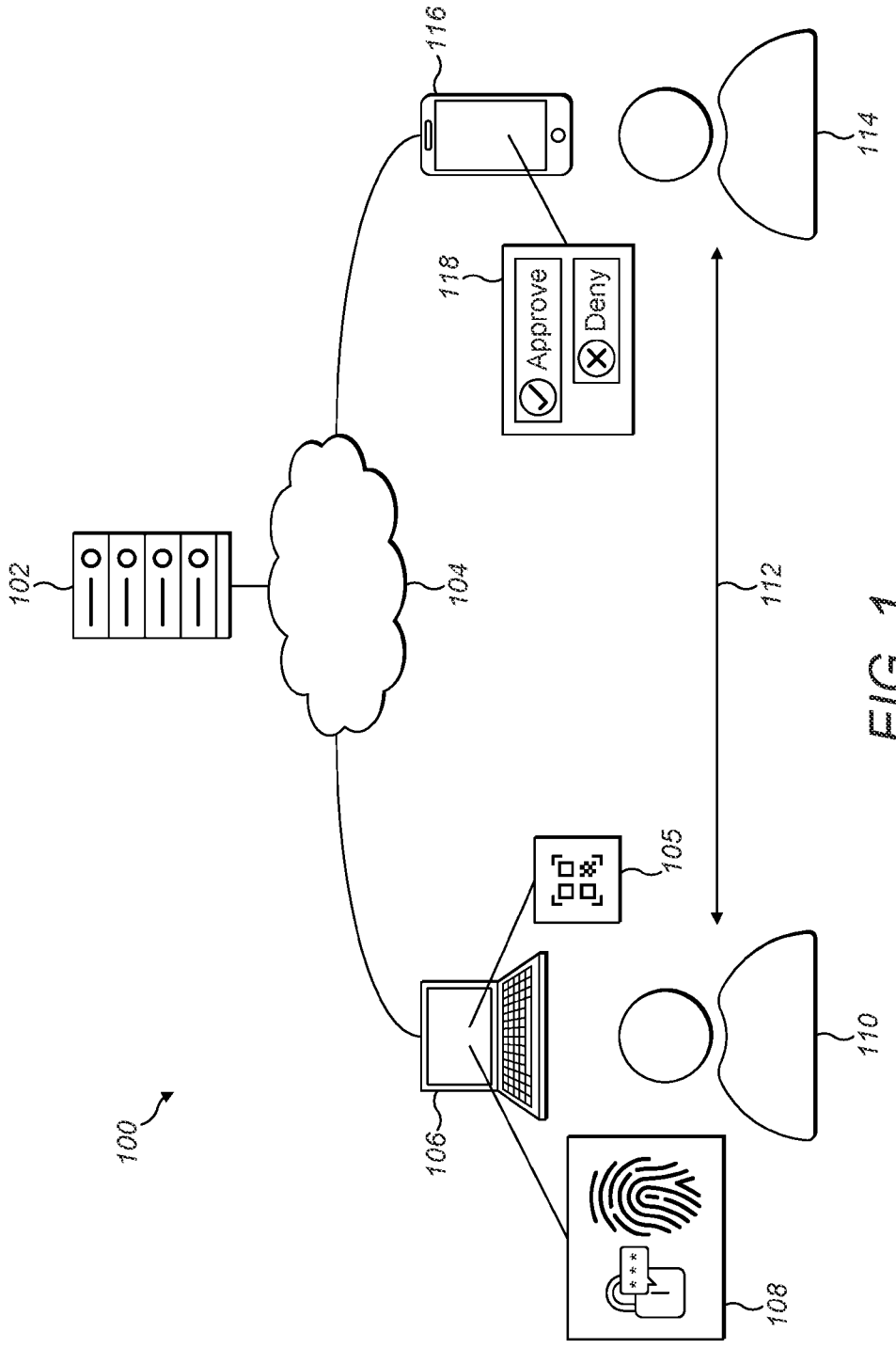

FIG. 1

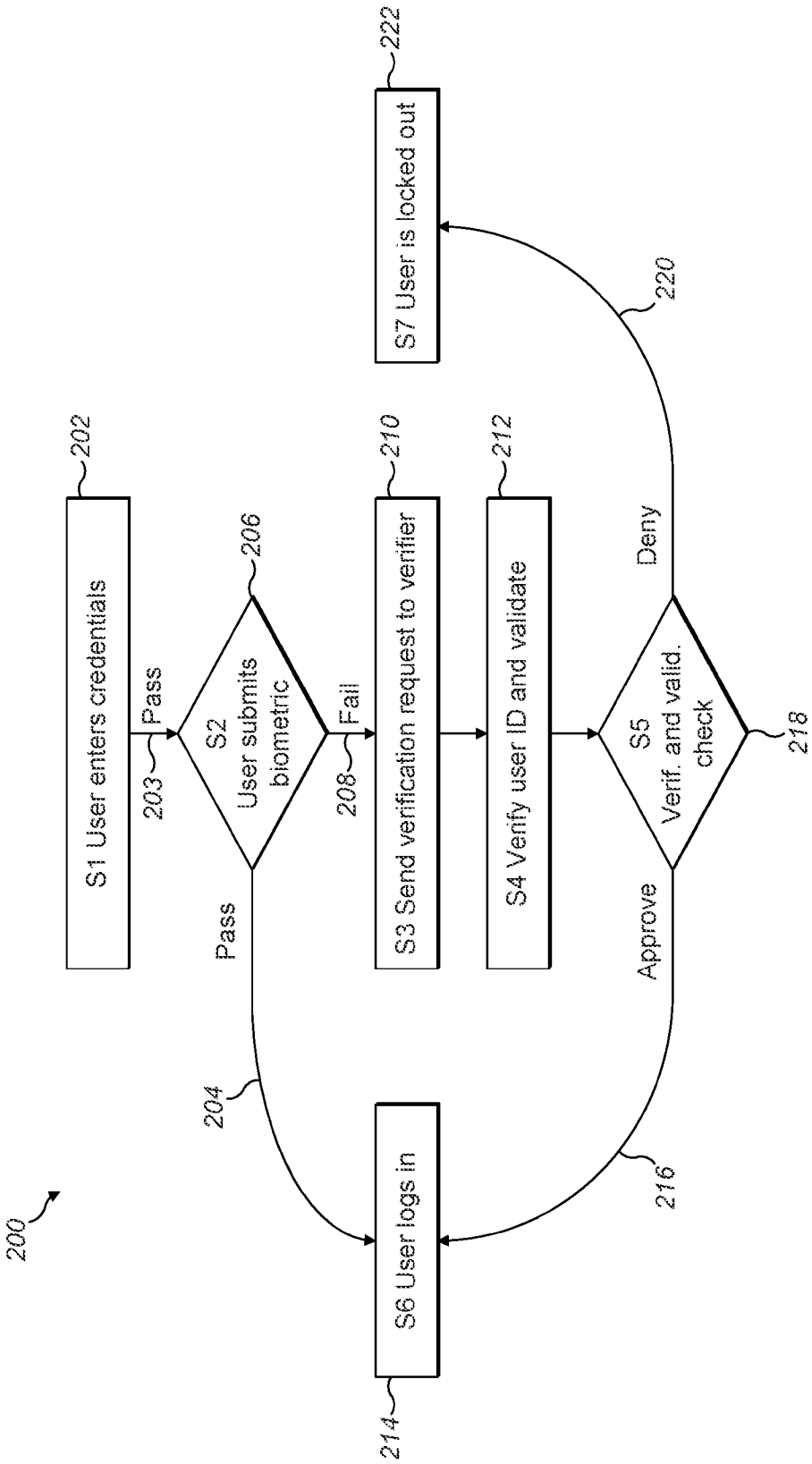
200
S1 User enters credentials
202
Pass
203
S2
User submits
biometric
206
Pass
204
Fail
208
S3 Send verification request to verifier
210
S4 Verify user ID and validate
212
S5
Verif. and valid.
check
218
Deny
220
S7 User is locked out
222
Approve
216
S6 User logs in
214

FIG. 2

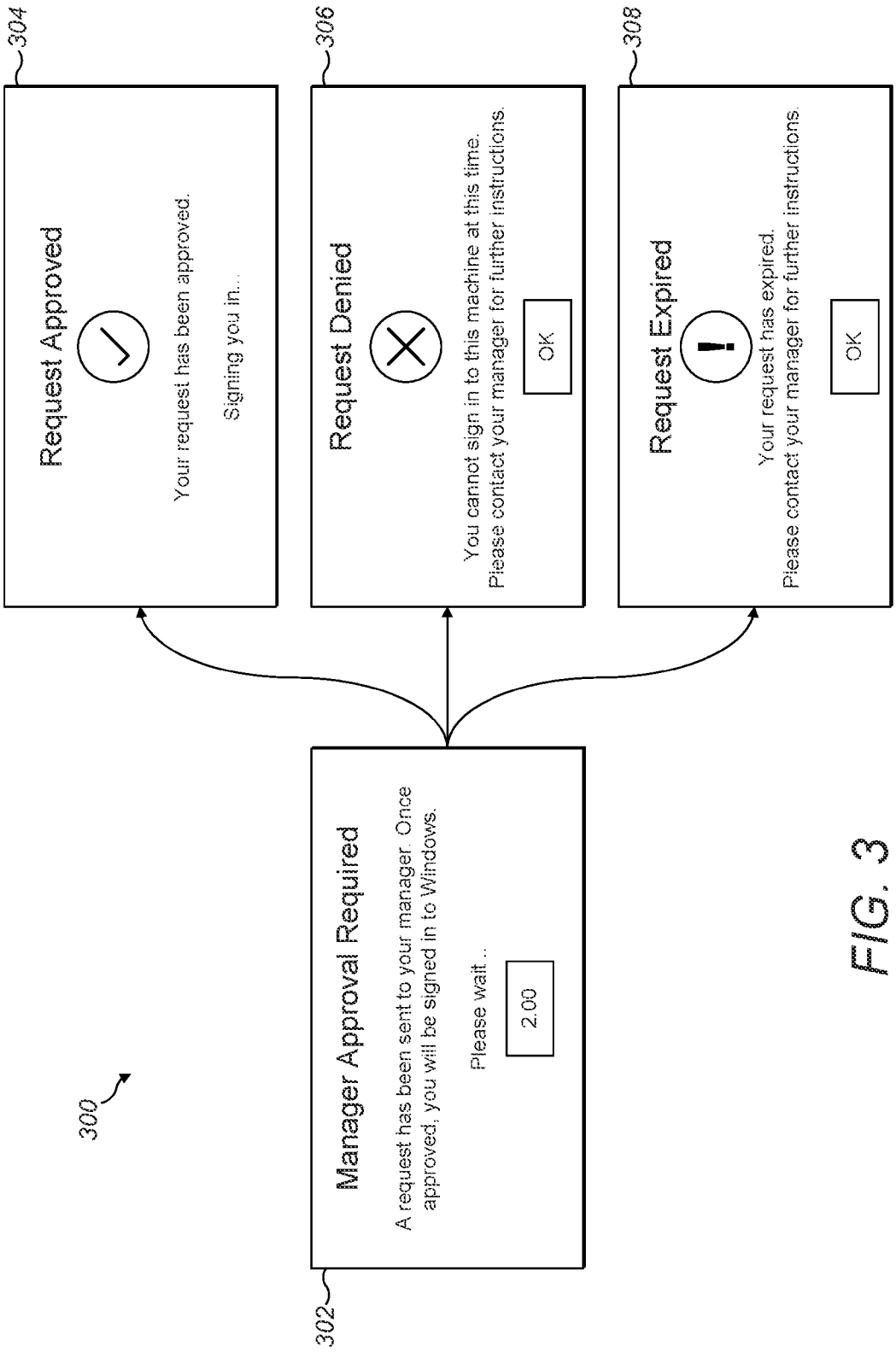
300
302
Manager Approval Required
A request has been sent to your manager. Once approved, you will be signed in to Windows.
Please wait ...
2.00
304
Request Approved
Your request has been approved.
Signing you in...
306
Request Denied
You cannot sign in to this machine at this time.
Please contact your manager for further instructions.
OK
308
Request Expired
Your request has expired.
Please contact your manager for further instructions.
OK

FIG. 3

400
(1) QR Code Scanner for Certification
403
(2) Verification Request
Windows Sign-in Request
Matt H
2:06:14 PM
Approval needed to sign in to Windows.
Username: matth
Device: -12345
Approve
Deny
404
(3A) Approval
Windows Sign-in Request
Matt H
2:06:14 PM
Approval needed to sign in to Windows.
Username: matth
Device: -12345
Approved by Jane S at 2:07:05
406
(3B) Denial
Windows Sign-in Request
Matt H
2:06:14 PM
Approval needed to sign in to Windows.
Username: matth
Device: -12345
Denieded by Jane S at 2:07:05
408

FIG. 4