METHODS FOR MANAGING IDENTIFICATION, VERIFICATION, AND VALIDATION OF A PLURALITY OF COMPUTING DEVICES

Technical Field

[0001] The present disclosure relates generally to identifying computing devices and/or users, and more specifically, to methods, systems, and devices for managing identification, verification, and/or validation of computing devices and/or users.

Background

[0002] There continues to be an ever-growing number and variety of online services available for users to access. Such online services may include and/or be provided via websites, software applications (e.g., applications installed on a laptop or desktop; mobile applications installed on mobile devices; etc.; referred to herein as an "**app**", "**application**", "**software application**", "**mobile application**", or the like), widgets, request forms, search functions, access to and/or participation in systems and/or platforms, etc. Computing devices used by users to access such online services may include, but are not limited to, mobile phones, wearable devices, tablets, laptops, desktops, etc. (referred to herein as "**computing devices**" or "**mobile devices**").

[0003] Service providers of such online services continue to seek improvements to their online service offerings so as to, among other things, improve users' online experiences, obtain better results from their online services (e.g., more targeted services and/or information, etc.), gain more accurate and/or meaningful statistics of their online services, and/or meet user demands. For example, service providers continue to offer better search results for user-initiated searches, content that is more relevant to certain users, advertisements that are more interesting to certain users, improved visual and/or audio quality, etc.

Brief Summary

[0004] As service providers continue to strive for improvements to their online service offerings, users also continue to demand more from online services. In this regard, service providers who are unable to meaningfully improve their online services and/or meet user demands will run the risk of losing out to other service providers who offer better, more interesting, and/or more relevant online services.

[0005] For many service providers, improvements to their online service offerings oftentimes center around providing more customized, tailored, targeted, focused, and/or relevant services, functions, actions, and/or information to the computing devices of users. Presently, conventional approaches to providing such customized, tailored, targeted, focused, and/or relevant services, functions, actions, and/or information involve, among other things, service

providers requesting, receiving, obtaining, and/or using personal information and/or personally identifiable information (PII) of users. Such personal information and/or personally identifiable information (PII) of users can be obtained in a plurality of ways, including, but not limited to, requiring users to register/create an account; requesting users to complete a questionnaire, request form, poll, or the like; requesting, obtaining, and/or pulling information from user computing devices; etc.

[0006] For example, it may be advantageous or useful to service providers to be able to identify and/or determine whether a user (using a computing device) who is submitting a request form had previously submitted the same or similar request forms (e.g., avoid wasting resources to process the request form again, etc.). As another example, being able to identify and/or determine whether a user (using a computing device) who is visiting a website (or using a mobile application) had previously been served a particular advertisement (e.g., a banner ad on a website or mobile application) may enable the service provider to make a better decision on whether to display a different (or the same) advertisement during the current visit to the website (or current use of the mobile application).

[0007] While such approaches of using personal information and/or personally identifiable information (PII) of users can be effective in providing customized, tailored, targeted, focused, and/or relevant services, functions, actions, and/or information to users, it is recognized in the present disclosure that service providers are becoming increasingly restricted from and/or being increasingly held accountable for receiving, storing, and/or using personal information and/or personally identifiable information (PII) of users without consent.

[0008] Present example embodiments relate generally to and/or include systems, subsystems, processors, devices, logic, methods, and processes for addressing conventional problems, including those described above and in the present disclosure, and more specifically, example embodiments relate to systems, subsystems, processors, devices, logic, methods, and processes for managing identification, verification, and/or validation of computing devices and/or users.

[0009] In an exemplary embodiment, a method of managing computing device identification, verification, and/or validation is disclosed. The method includes establishing a first communication channel between a processor and a first computing device. The method also includes selecting, by the processor, a first group of applications installed on the first computing device. The first group of applications is selected based on an application selection criteria. The method also includes generating, by the processor, a ranked group of applications. The ranked group of applications is generated by ranking two or more applications in the first

group of applications based on a predetermined metadata of the two or more applications in the first group of applications. The method also includes selecting, by the processor for the ranked group of applications, at least a first metadata type and second metadata type from among a plurality of metadata types. The first metadata type is different from the second metadata type. The method also includes obtaining, by the processor, a first metadata and second metadata of each application in the ranked group of applications. The first metadata of each application in the ranked group of applications includes metadata corresponding to the first metadata type. The second metadata of each application in the ranked group of applications includes metadata corresponding to the second metadata type. The method also includes generating, by the processor, an information payload. The information payload includes at least the first metadata and second metadata of each application in the ranked group of applications. The method also includes generating, by the processor, a unique device identifier for the first computing device. The unique device identifier is generated based on at least the information payload.

[0010] In another exemplary embodiment, a method of managing computing device identification, verification, and/or validation is disclosed. The method includes establishing a first communication channel between a processor and a first computing device. The method may also include identifying, by the processor, an initial set of applications. The initial set of applications may include applications that were installed on the first computing device within a first period of time. The method may also include selecting, by the processor, a first group of applications from among the initial set of applications. The first group of applications may be selected based on an application selection criteria. The method may also include selecting, by the processor for the first group of applications, a first metadata type from among a plurality of metadata types. The first metadata type may be selected based on a first metadata selection criteria. The method may also include obtaining, by the processor, a first metadata of each application in the first group of applications. The first metadata of each application in the first group of applications may include metadata corresponding to the first metadata type. The method may also include generating, by the processor, a ranked group of applications. The ranked group of applications may be generated by ranking two or more applications in the first group of applications based on the first metadata of the two or more applications in the first group of applications. The method may also include selecting, by the processor for the ranked group of applications, at least a second metadata type and third metadata type from among the plurality of metadata types. The second metadata type may be selected based on a second

metadata selection criteria. The third metadata type may be selected based on a third metadata selection criteria. The second metadata selection criteria may be different from the third metadata selection criteria. The method may also include obtaining, by the processor, a second metadata and third metadata of each application in the ranked group of applications. The second metadata of each application in the ranked group of applications may be metadata corresponding to the second metadata type. The third metadata of each application in the ranked group of applications may be metadata corresponding to the third metadata type. The method may also include generating, by the processor, a unique device identifier for the first computing device. The unique device identifier may be generated based on at least the second and third metadata of each application in the ranked group of applications.

[0011] In another exemplary embodiment, a method of managing computing device identification, verification, and/or validation is disclosed. The method includes establishing a first communication channel between a processor and a first computing device. The method may also include selecting, by the processor, a first group of applications installed on the first computing device. The first group of applications may be selected based on an application selection criteria. The method may also include generating, by the processor, a ranked group of applications. The ranked group of applications may be generated by ranking two or more applications in the first group of applications based on a predetermined metadata of the two or more applications in the first group of applications. The method may also include selecting, by the processor for the ranked group of applications, at least a first metadata type from among a plurality of metadata types. The method may also include selecting, by the processor for the ranked group of applications, a second metadata type from among a plurality of metadata types. The first metadata type is different from the second metadata type. The method may also include obtaining, by the processor, a first metadata and second metadata of each application in the ranked group of applications. The first metadata of each application in the ranked group of applications may be metadata corresponding to the first metadata type. The second metadata of each application in the ranked group of applications may be metadata corresponding to the second metadata type. The method may also include generating an information payload. The information payload may include the first metadata of each application in the ranked group of applications. The information payload may also include the second metadata of each application in the ranked group of applications. The method may also include generating, by the processor, a unique device identifier for the first computing device. The unique device identifier may be generated based on at least the information payload.

[0012] In another exemplary embodiment, a method of managing computing device identification, verification, and/or validation is disclosed. The method includes establishing a first communication channel between a processor and a first computing device. The method may also include selecting, by the processor, a first group of applications. The first group of applications may include applications installed on the first computing device. The method may also include generating, by the processor, a ranked group of applications. The ranked group of applications may be generated by ranking two or more applications in the first group of applications. The method may also include obtaining, by the processor, a first metadata of each application in the ranked group of applications. The first metadata of each application in the ranked group of applications may include metadata corresponding to a first metadata type. The method may also include generating, by the processor, a unique device identifier for the first computing device. The unique device identifier may be generated based on at least the first metadata of each application in the ranked group of applications.

Brief Description of the Figures

[0013] For a more complete understanding of the present disclosure, example embodiments, and their advantages, reference is now made to the following description taken in conjunction with the accompanying figures, in which like reference numbers indicate like features, and:

[0014] Figure 1 is an illustration of an example embodiment of a system for managing computing device identification, verification, and validation;

[0015] Figure 2 is an illustration of an example embodiment of a main processor for managing computing device identification, verification, and validation;

[0016] Figure 3 is an illustration of an example embodiment of a unique device identifier processor for generating unique device identifiers;

[0017] Figure 4A is an illustration of an example of an initial list of applications;

[0018] Figure 4B is an illustration of an example of a first group of applications;

[0019] Figure 4C is an illustration of an example of a ranked group of applications;

[0020] Figure 4D is an illustration of an example of a final group of applications;

[0021] Figure 4E is an illustration of a concatenation of information pertaining to a computing device and metadata obtained from each application in the final group of applications;

[0022] Figure 4F is an illustration of an example of a unique device identifier generated by hashing a concatenation of information pertaining to a computing device and metadata obtained from each application in the final group of applications;

[0023] Figure 5 is an illustration of an example embodiment of a method of managing computing device identification, verification, and validation;

[0024] Figure 6 is an illustration of another example embodiment of a method of managing computing device identification, verification, and validation; and

[0025] Figure 7 is an illustration of another example embodiment of a method of managing computing device identification, verification, and validation.

[0026] Although similar reference numbers may be used to refer to similar elements in the figures for convenience, it can be appreciated that each of the various example embodiments may be considered to be distinct variations.

[0027] Example embodiments will now be described with reference to the accompanying figures, which form a part of the present disclosure and which illustrate example embodiments which may be practiced. As used in the present disclosure and the appended claims, the terms "embodiment", "example embodiment", "exemplary embodiment", and "present embodiment" do not necessarily refer to a single embodiment, although they may, and various example embodiments may be readily combined and/or interchanged without departing from the scope or spirit of example embodiments. Furthermore, the terminology as used in the present disclosure and the appended claims is for the purpose of describing example embodiments only and is not intended to be limitations. In this respect, as used in the present disclosure and the appended claims, the term "in" may include "in" and "on", and the terms "a", "an", and "the" may include singular and plural references. Furthermore, as used in the present disclosure and the appended claims, the term "by" may also mean "from," depending on the context. Furthermore, as used in the present disclosure and the appended claims, the term "if" may also mean "when" or "upon", depending on the context. Furthermore, as used in the present disclosure and the appended claims, the words "and/or" may refer to and encompass any and all possible combinations of one or more of the associated listed items.

Detailed Description

[0028] Present example embodiments relate generally to and/or include systems, subsystems, processors, devices, logic, methods, and processes for addressing conventional problems, including those described above and in the present disclosure, and more specifically, example embodiments relate to systems, subsystems, processors, devices, logic, methods, and processes for managing computing device identification, verification, and validation. It is to be understood in the present disclosure that one or more elements, actions, and/or aspects of example embodiments may include and/or implement, in part or in whole, solely and/or in

cooperation with other elements, using, for example, networking technologies, cloud computing, distributed ledger technology (DLT) (e.g., blockchain), artificial intelligence (AI), machine learning, deep learning, etc. Furthermore, although example embodiments described in the present disclosure may be directed to use of information (e.g., metadata) pertaining to applications installed on a user's computing device and information pertaining to the user's computing device, it is to be understood in the present disclosure that example embodiments may also be directed to use of other non-personal information and/or other non-personally identifiable information (PII) without departing from the teachings of the present disclosure.

[0029] Example embodiments will now be described below with reference to the accompanying figures, which form a part of the present disclosure.

[0030] **Example embodiments of a system for managing computing device identification, verification, and/or validation (e.g., system 100).**

[0031] **FIGURE 1** illustrates an example embodiment of a system (e.g., system 100) for managing computing device identification, verification, and/or validation. Such managing of computing device identification may include, for example, generating of unique device identifiers (or the like) for a plurality of computing devices 10. As further described in the present disclosure, a unique device identifier generated by the system 100 for a first computing device 10 will be different from a unique device identifier generated by the system 100 for other computing devices 10 (i.e., unique device identifiers are unique to the computing device 10 that it is generated for).

[0032] The system 100 is also configurable or configured to verify and/or validate computing devices 10 based on unique device identifiers generated for the computing device 10. For example, the system 100 may determine whether or not a particular computing device 10 has previously visited and/or interacted with a particular website, submitted a particular request form, received a particular advertisement, received a particular message, received a particular notification, is an approved, recognized, and/or known computing device, etc.

[0033] As illustrated in Figure 1, the system 100 includes one or more main processors (e.g., main processor 200). Each main processor 200 is configurable or configured to connect to, communicate with, establish a communication channel with, and/or receive communications from a variety of computing devices 10 (e.g., mobile devices 10, user devices 10, laptops 10, wearable devices 10, etc.), websites 20, applications (or apps) 20, widgets, platforms 20, systems 20, communication channels 30, networks 30, databases 40, and/or other main processors 200. The system 100 also includes and/or communicates with one or more

networks, communication channels, or the like (e.g., communication channels 30), which enable communications between elements of the system 100. The system 100 also includes and/or communicates with one or more databases, distributed ledgers (DLT), or the like (e.g., database 40). The system 100 also includes and/or communicates with one or more users and/or user devices (e.g., user 10, user device 10, computing device 10, mobile device 10, or the like). Although the figures may illustrate the system 100 as having one main processor 200, one communication channel 30, one database 40, four user devices 10, one website 20, one app 20, one platform 20, and one system 20, it is to be understood that the system 100 may include more than one main processor 200, more than one communication channel 30, more than one database 40, more or less than four user devices 10, more or less than one website 20, more or less than one app 20, more or less than one platform 20, and/or more or less than one system 20 without departing from the teachings of the present disclosure.

[0034] As used in the present disclosure, when applicable, a reference to a system (e.g., system 100) or processor (e.g., main processor 200) may also refer to, apply to, and/or include one or more computing devices, processors, servers, systems, cloud-based computing, or the like, and/or functionality of one or more processors, computing devices, servers, systems, cloud-based computing, or the like. The system 100 and/or processor 200 (and/or its elements, as described in the present disclosure) may be any processor, server, system, device, computing device, controller, microprocessor, microcontroller, microchip, semiconductor device, or the like, configurable or configured to perform, among other things, a processing and/or managing of information, searching for information, identifying of information, data communications, and/or any one or more other actions described above and in the present disclosure. Alternatively, or in addition, the system 100 and/or processor 200 (and/or its elements, as described in the present disclosure) may include and/or be a part of a virtual machine, processor, computer, node, instance, host, or machine, including those in a networked computing environment.

[0035] As used in the present disclosure, a communication channel 30, network 30, cloud 30, or the like, may be or include a collection of devices and/or virtual machines connected by communication channels that facilitate communications between devices and allow for devices to share resources. Such resources may encompass any types of resources for running instances including hardware (such as servers, clients, mainframe computers, networks, network storage, data sources, memory, central processing unit time, scientific instruments, and other computing devices), as well as software, software licenses, available network services, and other non-

hardware resources, or a combination thereof. A communication channel 30, network 30, cloud 30, or the like, may include, but is not limited to, computing grid systems, peer to peer systems, mesh-type systems, distributed computing environments, cloud computing environment, telephony systems, voice over IP (VoIP) systems, voice communication channels, voice broadcast channels, text-based communication channels, video communication channels, etc. Such communication channels 30, networks 30, clouds 30, or the like, may include hardware and software infrastructures configured to form a virtual organization comprised of multiple resources which may be in geographically disperse locations. Communication channel 30, network 30, cloud 30, or the like, may also refer to a communication medium between processes on the same device. Also as referred to herein, a network element, node, or server may be a device deployed to execute a program operating as a socket listener and may include software instances.

[0036] These and other elements of the system 100 will now be further described with reference to the accompanying figures.

[0037] **The main processor (e.g., main processor 200).**

[0038] As illustrated in at least Figure 1, the system 100 for managing computing device identification, verification, and validation includes one or more main processors (e.g., main processor 200). The main processor 200 is configurable or configured to communicate with one or more elements of the system 100, including one or more computing devices 10, one or more websites 20, one or more applications 20, one or more platforms 20, one or more systems 20, one or more networks 30 (or communication channels 30), one or more databases 40, and/or one or more other main processors (not shown). The main processor 200 is also configurable or configured to perform a variety of other actions, functions, processes, and/or methods.

[0039] In an example embodiment, the main processor 200 is configurable or configured to generate unique device identifiers for a plurality of computing devices 10. Such unique device identifiers may be generated by the main processor 200 with or without personal information and/or personally identifiable information (PII) of the user of the computing device 10. In preferred embodiments, the main processor 200 is configured to generate unique device identifiers for computing devices 10 without using any personal information and/or personally identifiable information (PII) of the user of the computing device 10.

[0040] The main processor 200 is also configurable or configured to verify and/or validate computing devices 10 and/or unique device identifiers. For example, the main processor 200 is configured to verify and/or validate a computing device 10 by generating a unique device

identifier for the computing device 10, comparing such generated unique device identifier with a database 40 of known or previously generated unique device identifiers, and performing the verification and/or validation based on such comparison. As another example, the main processor 200 is configured to verify and/or validate a computing device 10 for a particular purpose 20, website 20, application 20, platform 20, system 20, and/or the like, by generating a unique device identifier for the computing device 10, comparing such generated unique device identifier with known or previously generated unique device identifiers for that particular purpose 20, website 20, application 20, platform 20, system 20, and/or the like stored in a database 40, and performing the verification and/or validation based on such comparison. As a more specific example, an application 20 may want to identify users/computing devices 10 that have previously submitted a particular request form (e.g., chance to win a prize, etc.). In such an example, the main processor 200 will generate a unique device identifier for each user/computing device 10 that submits the request form and compare such generated unique device identifier with unique device identifiers previously generated and stored in a database 40 dedicated/assigned to such request form. When a match is found, the main processor 200 is configured to determine that the user/computing device 10 had previously submitted the request form and take an appropriate and/or desired action.

[0041] To perform the actions, functions, processes, and/or methods described above and in the present disclosure, example embodiments of the main processor 200 include one or more elements. For example, as illustrated in **FIGURE 2**, the main processor 200 includes one or more main interfaces (e.g., main interface 210). The main processor 200 also includes one or more unique device identifier processors (e.g., unique device identifier processor 300). The main processor 200 also includes one or more unique device identifier validators (e.g., unique device identifier validator 400). The main processor 200 also includes one or more results processors (e.g., results processor 220). Although the figures may illustrate one main interface 210, one unique device identifier processor 300, one unique device identifier validator 400, and one results processor 220, it is to be understood that the main processor 200 may include more or less than one main interface 210, more or less than one unique device identifier processor 300, more or less than one unique device identifier validator 400, and/or more or less than one results processor 220 without departing from the teachings of the present disclosure. It is also to be understood in the present disclosure that, although the actions, functions, processes, and/or methods performed by the main processor 200 may be described in the present disclosure as being performed by one or more particular elements of the main processor

200, the actions, functions, processes, and/or methods performed by a particular element of the main processor 200 may also be performed by one or more other elements and/or cooperatively performed by more than one element of the main processor 200 (and/or other elements of the system 100) without departing from the teachings of the present disclosure. It is also to be understood in the present disclosure that, although the actions, functions, processes, and/or methods performed by the main processor 200 are described in the present disclosure as being performed by particular elements of the main processor 200, the actions, functions, processes, and/or methods performed by two or more particular elements of the main processor 200 may be combined and performed by one element of the main processor 200 without departing from the teachings of the present disclosure.

[0042] These elements of the main processor 200 will now be further described with reference to the accompanying figures.

[0043] The main interface (e.g., main interface 210).

[0044] In an example embodiment, the main processor 300 includes one or more main interfaces (e.g., main interface 210). The main interface 210 may be configurable or configured to interface with and/or communicate with one or more elements of the system 100.

[0045] For example, the main interface 210 may be configurable or configured to interface with and/or communicate with one or more computing devices 10, such as mobile devices 10, tablets 10, laptops 10, and/or other computing devices 10. As will be further described in the present disclosure, such communications may include, but are not limited to, communications to establish communication channels 30 with computing devices 10; communications with computing devices 10 via communication channels 30; receiving and/or requesting information pertaining to initial sets of applications (as described in the present disclosure) from computing devices 10; receiving and/or requesting information pertaining to one or more metadata types and metadata of applications previously and/or currently installed on the computing device 10; etc. In an example embodiment, the website 20, application 20, platform 20, system 20, etc. (also referred to herein as a "**service provider**") identifies the computing devices 10 that interfaces and/or communicates with the main interface 210.

[0046] As another example, the main interface 210 may be configurable or configured to interface with and/or communicate with one or more websites 20, applications 20, platforms 20, systems 20, etc. Such communications may include, but are not limited to, communications to establish communication channels 30 with such websites 20, applications 20, platforms 20, systems 20, etc.; communications with such websites 20, applications 20, platforms 20, systems

20, etc. via communication channels 30; receiving of a request, or the like, from such websites 20, applications 20, platforms 20, systems 20, etc. to determine whether a computing device 10 has previously, recently, historically, or the like, visited, interacted with, submitted a form using, etc. such websites 20, applications 20, platforms 20, systems 20, etc.; receiving of a request, or the like, from such websites 20, applications 20, platforms 20, systems 20, etc. to generate a unique device identifier for a computing device 10; receiving of a request, or the like, from such websites 20, applications 20, platforms 20, systems 20, etc. to validate and/or verify a unique device identifier of a computing device 10; etc.

[0047] The main interface 210 may also be configurable or configured to interface with and/or communicate with one or more networks 30, communication channels 30, or the like; one or more databases 40, data storage units 40, cloud computing 40, blockchains 40, distributed ledgers (DLT), or the like; and/or one or more other main processors 200, or the like.

[0048] After the main interface 210 receives information from one or more elements of the system 100, as described above and in the present disclosure, the main interface is configurable or configured to communicate such received information to one or more elements of the main processor 200, including the unique device identifier processor 300, the unique device identifier validator 400, and the results processor 220.

[0049] The unique device identifier processors (e.g., unique device identifier processor 300).

[0050] As illustrated in at least Figure 2, the main processor 200 for managing computing device identification, verification, and validation includes one or more unique device identifier processors (e.g., unique device identifier processor 300). The unique device identifier processor 300 is configurable or configured to communicate with other elements of the main processor 200, including the main interface 210, the unique device identifier validator 400, and the results processor 210. As further described in the present disclosure, the unique device identifier processor 300 is then configurable or configured to generate unique device identifiers for a plurality of computing devices 10.

[0051] In an example embodiment, the unique device identifier processor 300 generates a unique device identifier for a particular computing device 10 (e.g., a mobile device 10) by first establishing a communication channel 30 with the computing device 10 (e.g., via the main interface 210). The unique device identifier processor 300 is then configurable or configured to select a first group of applications. The first group of applications may include applications currently (and/or previously) installed on the computing device 10. The unique device identifier processor 300 is then configurable or configured to select a first metadata type for

the first group of applications. The unique device identifier processor 300 is then configurable or configured to obtain a first metadata of each application in the first group of applications. The first metadata of each application in the first group of applications is a metadata corresponding to the first metadata type. The unique device identifier processor 300 is then configurable or configured to generate a ranked group of applications. The ranked group of applications may be generated by ranking two or more applications (or all of the applications) in the first group of applications. Such ranking may be based on the first metadata (and/or first metadata type). The unique device identifier processor 300 is then configurable or configured to generate the unique device identifier for the computing device 10. The unique device identifier may be generated based on the first metadata of each application in the ranked group of applications (or of two or more of the top ranked applications in the ranked group of applications). For example, the unique device identifier may be a hash of a concatenation of the first metadata of two or more (or all) of the applications in the ranked group of applications.

[0052] Alternatively, the unique device identifier processor 300 may not generate the unique device identifier based on the first metadata of each application in the ranked group of applications, but instead select a second metadata type for the ranked group of applications, followed by obtain a second metadata for each application in a final group of applications (the final group of applications being a group of applications formed by selecting two or more of the top ranked applications in the ranked group of applications (or selecting all of the applications in the ranked group of applications)). The second metadata of each application in the ranked group of applications is a metadata corresponding to the second metadata type. The unique device identifier processor 300 is then configurable or configured to generate the unique device identifier for the computing device 10. The unique device identifier may be generated based on the second metadata of each application in the final group of applications. For example, the unique device identifier may be a hash of a concatenation of the second metadata of all applications in the final group of applications.

[0053] As used in the present disclosure, a "**metadata**" for an application (or app) may be any metadata for the application including, but not limited to, a date/timestamp when the application was installed, date/timestamp when the application was uninstalled, date/timestamp when the application was updated, date/timestamp when the application was first used by the user on the computing device, date/timestamp indicating that the application was installed by the user (i.e., not factory-installed by the manufacturer; which can be determined in one or more ways, such as identifying whether the date/timestamp for the installation of the

application includes milliseconds), date/timestamp when the application transitioned from a free trial to a full or paid version, name of the application, name of the developer of the application, version number of the application, number of times each application has been used/launched/selected by the user within a predetermined time period, duration of use of each application within a predetermined time period, location of the application (e.g., position on page, page number, etc.), etc.

[0054] In another example embodiment, the unique device identifier processor 300 generates a unique device identifier for a particular computing device 10 (e.g., a mobile device 10) by first establishing a communication channel 30 with the computing device 10 (e.g., via the main interface 210). The unique device identifier processor 300 is then configurable or configured to select a first group of applications. The first group of applications may include applications currently (and/or previously) installed on the computing device 10. The unique device identifier processor 300 is then configurable or configured to generate a ranked group of applications. The ranked group of applications is generated by ranking two or more (or all) applications in the first group of applications based on a predetermined metadata of the two or more (or all) applications in the first group of applications. The unique device identifier processor 300 is then configurable or configured to select, for two or more applications in the ranked group of applications, a first metadata type from among a plurality of metadata types. The unique device identifier processor 300 may also be configurable or configured to select, for two or more applications in the ranked group of applications, a second metadata type from among a plurality of metadata types. The first metadata type is a different metadata type than the second metadata type. The unique device identifier processor 300 is then configurable or configured to obtain a first metadata of each application in the ranked group of applications (or of two or more applications in the ranked group of applications). In example embodiments where a second metadata type is selected, the unique device identifier processor 300 is also configurable or configured to obtain a second metadata of each application in the ranked group of applications (or of two or more applications in the ranked group of applications). The first metadata of each application in the ranked group of applications (or of two or more applications in the ranked group of applications) is a metadata corresponding to the first metadata type. The second metadata of each application in the ranked group of applications (or of two or more applications in the ranked group of applications) is a metadata corresponding to the second metadata type. The unique device identifier processor 300 is then configurable or configured to generate an information payload. The information payload includes the first metadata of

each application in the ranked group of applications (or of two or more applications in the ranked group of applications). The information payload may also include the second metadata of each application in the ranked group of applications (or of two or more applications in the ranked group of applications). The unique device identifier processor 300 is then configurable or configured to generate the unique device identifier for the computing device 10. The unique device identifier is generated based on at least the information payload. For example, the unique device identifier may be a hash of information payload.

[0055] In another example embodiment, the unique device identifier processor 300 generates a unique device identifier for a particular computing device 10 (e.g., a mobile device 10) by first establishing a communication channel 30 with the computing device 10 (e.g., via the main interface 210). The unique device identifier processor 300 is then configurable or configured to identify an initial set of applications for the computing device 10. The unique device identifier processor 300 is then configurable or configured to select a first group of applications from among the initial set of applications. The first group of applications may include applications currently (and/or previously) installed on the computing device 10. The unique device identifier processor 300 is then configurable or configured to select a first metadata type for the first group of applications. The unique device identifier processor 300 is then configurable or configured to obtain, for each application in the first group of applications, a first metadata of the application. The unique device identifier processor 300 is then configurable or configured to generate a ranked group of applications. The ranked group of applications is generated by ranking two or more applications in the first group of applications based on the obtained first metadata of the two or more applications in the first group of applications. The unique device identifier processor 300 is then configurable or configured to select, for the ranked group of applications, a second metadata type from among a plurality of metadata types. The unique device identifier processor 300 may also be configurable or configured to select, for the ranked group of applications, a third metadata type from among a plurality of metadata types. The second metadata type is a different metadata type than the third metadata type. The unique device identifier processor 300 is then configurable or configured to obtain a second metadata of each application in the ranked group of applications (or of two or more of the top ranked applications in the ranked group of applications). The unique device identifier processor 300 may also be configurable or configured to obtain a third metadata of each application in the ranked group of applications (or of two or more of the top ranked applications in the ranked group of applications). The second metadata of each

application in the ranked group of applications (or of two or more of the top ranked applications in the ranked group of applications) is a metadata corresponding to the second metadata type. The third metadata of each application in the ranked group of applications (or of two or more of the top ranked applications in the ranked group of applications) is a metadata corresponding to the third metadata type. The unique device identifier processor 300 is then configurable or configured to generate the unique device identifier for the computing device 10. The unique device identifier is generated based on the second metadata of each application in the ranked group of applications (or of two or more of the top ranked applications in the ranked group of applications). The unique device identifier may also be generated based on the third metadata of each application in the ranked group of applications (or of two or more of the top ranked applications in the ranked group of applications). For example, the unique device identifier may be a hash of a concatenation of the second metadata (and/or third metadata) of two or more (or all) of the applications in the ranked group of applications.

[0056] To perform the actions, functions, processes, and/or methods described above and in the present disclosure, example embodiments of the unique device identifier processor 300 include one or more elements. For example, as illustrated in **FIGURE 3**, the unique device identifier processor 300 includes one or more initial set of applications processors (e.g., initial set of applications processor 301). The unique device identifier processor 300 also includes one or more application selection criteria generators (e.g., application selection criteria generator 302). The unique device identifier processor 300 also includes one or more first group of applications selectors (e.g., first group of applications selector 310). The unique device identifier processor 300 also includes one or more first stage metadata processors (e.g., first stage metadata processor 312). The unique device identifier processor 300 also includes one or more ranked group of applications generators (e.g., ranked group of applications generator 320). The unique device identifier processor 300 also includes one or more second stage metadata processors (e.g., second stage metadata processor 322). The unique device identifier processor 300 also includes one or more final group of applications selectors (e.g., final group of applications selector 330). The unique device identifier processor 300 also includes one or more unique device identifier generators (e.g., unique device identifier generator 340). Although the figures may illustrate one initial set of applications processor 301, one application selection criteria generator 302, one first group of applications selector 310, one first stage metadata processor 312, one ranked group of applications generator 320, one second stage metadata processor 322, one final group of applications selector 330, and one

unique device identifier generator 340, it is to be understood that the unique device identifier processor 300 may include more or less than one initial set of applications processor 301, more or less than one application selection criteria generator 302, more or less than one first group of applications selector 310, more or less than one first stage metadata processor 312, more or less than one ranked group of applications generator 320, more or less than one second stage metadata processor 322, more or less than one final group of applications selector 330, and/or more or less than one unique device identifier generator 340 without departing from the teachings of the present disclosure. It is also to be understood in the present disclosure that, although the actions, functions, processes, and/or methods performed by the unique device identifier processor 300 may be described in the present disclosure as being performed by one or more particular elements of the unique device identifier processor 300, the actions, functions, processes, and/or methods performed by a particular element of the unique device identifier processor 300 may also be performed by one or more other elements and/or cooperatively performed by more than one element of the unique device identifier processor 300 (and/or other elements of the main processor 200) without departing from the teachings of the present disclosure. It is also to be understood in the present disclosure that, although the actions, functions, processes, and/or methods performed by the unique device identifier processor 300 are described in the present disclosure as being performed by particular elements of the unique device identifier processor 300, the actions, functions, processes, and/or methods performed by two or more particular elements of the unique device identifier processor 300 may be combined and performed by one element of the unique device identifier processor 300 without departing from the teachings of the present disclosure.

[0057] These elements of the unique device identifier processor 300 will now be further described with reference to the accompanying figures.

[0058] *The initial set of applications processor (e.g., initial set of applications processor 301).*

[0059] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more initial set of applications processors (e.g., initial set of applications processor 301). The initial set of applications processor 301 is configurable or configured to communicate with other elements of the main processor 200, including the main interface 210, the first group of applications selector 310, the application selection criteria generator 302, the first stage metadata processor 312, the ranked group of applications generator 320, the second stage metadata processor 322, the final group of applications selector 330, the unique device

identifier generator 340, the unique device identifier validator 400, and/or the results processor 220.

[0060] After the main interface 210 establishes a communication channel with a computing device 10, the initial set of applications processor 301 is configurable or configured to receive information pertaining to an initial set of applications of the computing device 10. For example, information pertaining to the initial set of applications of the computing device 10 may include a list, table, or the like, of some or all of the applications currently installed on the computing device 10. Such information may include a variety of metadata types and metadata of the application including, but not limited to, the name of each application, date/time of installation of each application, name of the developer of each application, version number of each application, whether or not the application was installed by a user of the computing device 10 (i.e., not factory installed), date/time of updating of each application, etc. The initial set of applications processor 301 may also receive information pertaining to the computing device 10 (e.g., make, model, version, operating system, etc. of the computing device 10).

[0061] **FIGURE 4A** illustrates an example of a portion of a table of an initial set of applications of a computing device 10, which includes the name of each application 402, date/time of installation of each application 404, and whether or not each application was installed by the user of the computing device 10 406. In an example embodiment, the initial set of applications processor 301 (and/or one or more other elements of the main processor 200) may be configurable or configured to determine whether or not an application in the initial set of applications (and/or other groupings, such as the first group of applications selected by the first group of applications selector 310, ranked group of applications generated by the ranked group of applications generator 320, and/or final group of applications selected by the final group of applications selector 330) was installed by the user of the computing device 10. Such determination may be performed in one or more ways.

[0062] In an example embodiment, the initial set of applications processor 310 (and/or one or more other elements of the main processor 200) may also be configurable or configured to determine whether or not an application was installed by the user of the computing device 10 based on the date/time of installation 404 of the application. For example, the initial set of applications processor 310 (and/or one or more other elements of the main processor 200) may determine that an application was installed by the user of the computing device 10 (i.e., not factory installed, or not installed before the user activated or first used the computing device 10) when the date/time of installation of the application includes a "non-zero" milliseconds

portion in the date/time of installation. Examples of "non-zero" milliseconds portion in the date/time of installation are illustrated in Figure 4A for applications "com.skype.raider", "com.fecredit.RoboLending", and "com.zing.zalo". In such an example, applications "com.skype.raider", com.fecredit.RoboLending", and "com.zing.zalo" (see Figure 4A) would be determined the initial set of applications processor 310 (and/or one or more other elements of the main processor 200) to be applications that were installed by the user of the computing device 10 (and not factory-installed).

[0063] In an example embodiment, an application is determined to not be installed by the user of the computing device 10 (i.e., an application that was factory installed, or installed before the user activated or first used the computing device 10) when the initial set of applications processor 310 (and/or one or more other elements of the main processor 200) determines that the date/time of installation of the application includes "zero-only" milliseconds portion in the date/time of installation. Examples of "zero-only" milliseconds portion in the date/time of installation are illustrated in Figure 4A for applications "com.miui.screenrecorder", "com.android.cts.priv.ctsshim", and "com.qualcomm.qti.qms.service.telemetry". In such an example, applications "com.miui.screenrecorder", "com.android.cts.priv.ctsshim", and "com.qualcomm.qti.qms.service.telemetry" (see Figure 4A) would be determined to be applications that were factory installed on the computing device 10 (and not installed by the user of the computing device 10).

[0064] In example embodiments, information pertaining to the initial set of applications of the computing device 10 may also include a list, table, or the like, of: all applications that have ever been installed on the computing device 10; applications that were installed on the computing device 10 within a particular period of time; applications that were installed on but not uninstalled from the computing device 10 within a particular period of time; applications that were installed and updated on the computing device 10 within a particular period of time; etc. Other examples of initial sets of applications are also contemplated without departing from the teachings of the present disclosure.

[0065] The initial set of applications processor 301 is then configurable or configured to provide the information pertaining to the initial set of applications of the computing device 10 to the first group of applications selector 310 and/or one or more other elements of the main processor 200.

[0066] *The application selection criteria generator (e.g., application selection criteria generator 302).*

[0067] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more application selection criteria generators (e.g., application selection criteria generators 302). The application selection criteria generator 302 is configurable or configured to communicate with other elements of the unique device identifier processor 300, including the first group of applications selector 310, the initial set of applications processor 301, the first stage metadata processor 312, the ranked group of applications generator 320, the second stage metadata processor 322, the final group of applications selector 330, the unique device identifier generator 340, the unique device identifier validator 400, and/or the results processor 220.

[0068] In an example embodiment, the application selection criteria generator 302 is configurable or configured to select a predetermined application selection criteria/criterion to be used by the first group of applications selector 310. Alternatively, the application selection criteria generator 302 is configurable or configured to generate a dynamically generated application selection criteria/criterion to be used by the first group of applications selector 310.

[0069] For example, the application selection criteria may be a criteria that limits the first group of applications to only applications that were installed on the computing device 10 by the user within a particular period of time (e.g., within 1 week of activating or first use of the computing device 10). As another example, the application selection criteria may be a criteria that limits the first group of applications to only the first 10 (or more or less) applications that were installed on the computing device 10 by the user. As another example, the application selection criteria may be a criteria that limits the first group of applications to only applications that were installed on the computing device 10 by the user (e.g., only those installed by the user, and excluding factory-installed applications). Other examples of application selection criteria/criterion are also contemplated without departing from the teachings of the present disclosure.

[0070] The application selection criteria generator 302 is then configurable or configured to provide the application selection criteria to the first group of applications selector 310 and/or one or more other elements of the main processor 200.

[0071] *The first group of applications selector (e.g., first group of applications selector 310).*

[0072] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more first group of applications selectors (e.g., first group of applications selector 310). The first group of applications selector 310 is configurable or configured to communicate with other elements of the unique device identifier processor 300. For example, the first group of

applications selector 310 may receive information pertaining to the initial sets of applications from the initial set of applications processor 301. The first group of applications selector 310 also receives the application selection criteria/criterion from the application selection criteria generator 302. The first group of applications selector 310 also provides results (i.e., selected first group of applications) to the first stage metadata processor 312 and/or ranked group of applications generator 320.

[0073] In an example embodiment, the first group of applications selector 310 is configurable or configured to select a first group of applications (or select information pertaining to a first group of applications). In embodiments where information pertaining to an initial set of applications is received by the initial set of applications processor 301, the first group of applications selector 310 is configurable or configured to select a first group of applications from the initial set of applications (or select information pertaining to a first group of applications from the information pertaining to the initial set of applications). The first group of applications selector 310 selects the first group of applications (or information pertaining to the first group of applications) based on the application selection criteria/criterion selected or generated by the application selection criteria generator 302.

[0074] FIGURE 4B illustrates an example of a table of a first group of applications 412 selected based on the application selection criteria being the selecting of only applications that were installed by the user (and not factory-installed). As illustrated in Figure 4B, the table of the first group of applications of the computing device 10 includes the name of each application 412, date/time of installation of each application 414, and indication that each application was installed by the user of the computing device 10 416.

[0075] The first group of applications selector 310 then provides the selected first group of applications to the first stage metadata processor 312, the ranked group of applications generator 320, and/or one or more other elements of the main processor 200.

[0076] *The first stage metadata processor (e.g., first stage metadata processor 312).*

[0077] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more first stage metadata processors (e.g., first stage metadata processor 312). The first stage metadata processor 312 is configurable or configured to communicate with other elements of the unique device identifier processor 300, including the first group of applications selector 310 and/or the ranked group of applications generator 320.

[0078] In an example embodiment, the first stage metadata processor 312 is configurable or configured to select and/or generate a first metadata type for the ranked group of applications

processor 320. More specifically, the first stage metadata processor 312 selects a first metadata type for each first group of applications selected by the first group of applications selector 310 (which is provided to the ranked group of applications processor 320). Such selection of the first metadata type may be based on a variety of considerations including, but not limited to: information pertaining to the first group of applications selected by the first group of applications selector 310 (e.g., an assessment of one or more metadata of the applications; as a more specific example, if the first stage metadata processor 312 determines that the date/time of installation of most or all applications in the first group of applications is the same, then the first stage metadata processor 312 may not select the date/time of installation as the first metadata type); the application selection criteria generated by the application selection criteria generator 302; a predetermined, dynamically determined, and/or randomly generated criteria/criterion selected by and/or for the website 20, application 20, platform 20, system 20, etc.; and/or a common first metadata type for all first groups of applications selected by the first group of applications selector 310.

[0079] After selecting a first metadata type for each first group of applications selected by the first group of applications selector 310, an example embodiment of the first stage metadata processor 312 is then configurable or configured to obtain (for each application in the first group of applications, for each application in the ranked group of applications, and/or any other grouping of applications) a first metadata of each application in the first group of applications. Such first metadata corresponds to the selected first metadata type. The first stage metadata processor 312 is then configurable or configured to provide the obtained first metadata of each application in the first group of applications to the ranked group of applications generator 320 and/or one or more other elements of the unique device identifier processor 300. Although the present disclosure may describe the first stage metadata processor 312 as selecting a first metadata type for each first group of applications and obtaining a first metadata for each application in each first group of applications, it is to be understood that the first stage metadata processor 312 may select two or more metadata types (e.g., a first metadata type, second metadata type, etc.) for each first group of applications and obtain two or more metadata (each corresponding to the selected two or more metadata types) for each application in each first group of applications without departing from the teachings of the present disclosure.

[0080] The first metadata type may be any metadata of each application. For example, the first metadata type may be a date/timestamp when the application was installed on the computing device 10. The first metadata type may also be a date/timestamp when the application was

uninstalled from the computing device 10. The first metadata type may also be a date/timestamp when the application was last updated on the computing device 10. The first metadata type may also be a date/timestamp when the application was first used by the user on the computing device 10. The first metadata type may also be a date/timestamp when the application transitioned from a free trial to a full or paid version on the computing device 10. The first metadata type may also be a name of the application. The first metadata type may also be a name of the developer of the application. The first metadata type may also be a version number of the application. The first metadata type may also be a number of times the application has been used, launched, and/or selected by the user within a predetermined time period. The first metadata type may also be a duration of use of the application within a predetermined time period. The first metadata type may also be a location of the application (e.g., position on page, page number, etc.), etc.

[0081] The first stage metadata processor 312 is then configurable or configured to provide the selected first metadata type and/or the obtained first metadata of each application in each first group of applications to the ranked group of applications generator 320 and/or one or more other elements of the main processor 200.

[0082] *The ranked group of applications processor (e.g., ranked group of applications processor 320).*

[0083] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more ranked group of applications processors (e.g., ranked group of applications processor 320). The ranked group of applications processor 320 is configurable or configured to communicate with other elements of the unique device identifier processor 300. For example, the ranked group of applications processor 320 receives first groups of applications selected by the first group of applications selector 310. The ranked group of applications processor 320 also receives selected first metadata types for first groups of applications from the first stage metadata processor 312. The ranked group of applications processor 320 also receives obtained first metadata for first groups of applications from the first stage metadata processor 312. The ranked group of applications processor 320 also provides ranked groups of applications to the final group of applications selector 330 and/or second stage metadata processor 322.

[0084] In an example embodiment, the ranked group of applications processor 320 is configurable or configured to generate a ranked group of applications by ranking two or more applications (or all of the applications) in the first group of applications. The ranking of the

two or more applications (or all of the applications) in the first group of applications is performed based on the first metadata type and/or first metadata selected/obtained by the first stage metadata processor 312. For example, the ranked group of applications processor 320 may rank all of the applications in the first group of applications (as received from the first group of applications selector 310) by obtaining and/or identifying the first metadata of each of the applications and ranking the applications based on such first metadata. As another example, the ranked group of applications processor 320 may first select two or more applications from the first group of applications received from the first group of applications selector 310. The ranked group of applications processor 320 then obtains and/or identifies the first metadata of each of the two or more applications and ranks the applications based on such first metadata.

[0085] For example, if the first metadata type selected by the first stage metadata processor 312 is a date/timestamp when an application was installed on a computing device 10, then the ranked group of applications will be two or more applications in the first group of applications that are ranked based on the obtained first metadata of each application (i.e., ranking based on the date/timestamp when the applications were installed on the computing device 10).

[0086] FIGURE 4C illustrates an example of a ranked group of applications 422 (for all applications in the first group of applications) generated based on the first metadata type being the date/timestamp of installation of the application and the first metadata being details of the date/timestamp of installation 424. The table of the ranked group of applications in Figure 4C also indicates that all applications in the ranked group of applications was installed by the user 426 (and not factory installed), which corresponds to the applications selection criteria selected by the application selection criteria generator 302 (see also Figure 4B).

[0087] FIGURE 4D illustrates an example of a ranked group of applications 432 (for only a predetermined number of 5 applications in the first group of applications) generated based on the first metadata type being the date/timestamp of installation of the application and the first metadata being details of the date/timestamp of installation 434. The table of the ranked group of applications in Figure 4D also indicates that all applications in the ranked group of applications was installed by the user 426 (and not factory installed), which corresponds to the applications selection criteria selected by the application selection criteria generator 302 (see also Figure 4B).

[0088] The ranked group of applications processor 320 is then configurable or configured to provide the ranked group of applications to the final group of applications selector 330, the second stage metadata processor 322, and/or other elements of the main processor 200.

[0089] *The second stage metadata processor (e.g., second stage metadata processor 322).*

[0090] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more second stage metadata processors (e.g., second stage metadata processor 322). The second stage metadata processor 322 is configurable or configured to communicate with other elements of the unique device identifier processor 300, including the ranked group of applications generator 320, the final group of applications selector 330, and/or the unique device identifier generator 340.

[0091] In an example embodiment, the second stage metadata processor 322 is configurable or configured to select a second metadata type for the final group of applications processor 330. More specifically, the second stage metadata processor 322 selects a second metadata type for each ranked group of applications (which is provided to the final group of applications processor 330). Such selection of the second metadata type may be based on a variety of considerations including, but not limited to: information pertaining to the ranked group of applications selected by the ranked group of applications selector 320 (e.g., an assessment of one or more metadata of the applications); the first metadata type selected by the first stage metadata processor 312; the application selection criteria generated by the application selection criteria generator 302; a predetermined, dynamically determined, and/or randomly generated criteria/criterion selected by and/or for the website 20, application 20, platform 20, system 20, etc.. Alternatively, the second metadata type may be a default and/or common metadata type used for some or all ranked groups of applications selected by the ranked group of applications selector 320.

[0092] The second metadata type may be any metadata type. For example, the second metadata type may be a date/timestamp when the application was installed on the computing device 10. The second metadata type may also be a date/timestamp when the application was uninstalled from the computing device 10. The second metadata type may also be a date/timestamp when the application was last or most recently updated on the computing device 10. The second metadata type may also be a date/timestamp when the application was first used by the user on the computing device 10. The second metadata type may also be a date/timestamp when the application transitioned from a free and/or trial version to a full and/or paid version on the computing device 10. The second metadata type may also be a name of

the application. The second metadata type may also be a name of the developer of the application. The second metadata type may also be a version number of the application. The second metadata type may also be a number of times the application has been used, launched, and/or selected by the user within a predetermined time period. The second metadata type may also be a duration of use of the application within a predetermined time period. The second metadata type may also be a location of the application (e.g., position on page, page number, etc.), etc. The second metadata type may be the same as or different from the first metadata type.

[0093] In an example embodiment, the second stage metadata processor 322 is also configurable or configured to select a third metadata type for the final group of applications processor 330 (and/or one or more other metadata types). More specifically, the second stage metadata processor 322 selects a third metadata type for each ranked group of applications (which is provided to the final group of applications processor 330). Such selection of the third metadata type may be based on a variety of considerations including, but not limited to: information pertaining to the ranked group of applications selected by the ranked group of applications selector 320 (e.g., an assessment of one or more metadata of the applications); the first metadata type selected by the first stage metadata processor 312; the second metadata type selected by the second stage metadata processor 312; the application selection criteria generated by the application selection criteria generator 302; a predetermined, dynamically determined, and/or randomly generated criteria/criterion selected by and/or for the website 20, application 20, platform 20, system 20, etc.. Alternatively, the third metadata type may be a default or common metadata type used for all ranked groups of applications selected by the ranked group of applications selector 320.

[0094] The third metadata type may be any metadata type. For example, the third metadata type may be a date/timestamp when the application was installed on the computing device 10. The third metadata type may also be a date/timestamp when the application was uninstalled from the computing device 10. The third metadata type may also be a date/timestamp when the application was last or most recently updated on the computing device 10. The third metadata type may also be a date/timestamp when the application was first used by the user on the computing device 10. The third metadata type may also be a date/timestamp when the application transitioned from a free and/or trial version to a full and/or paid version on the computing device 10. The third metadata type may also be a name of the application. The third metadata type may also be a name of the developer of the application. The third metadata

type may also be a version number of the application. The third metadata type may also be a number of times the application has been used, launched, and/or selected by the user within a predetermined time period. The third metadata type may also be a duration of use of the application within a predetermined time period. The third metadata type may also be a location of the application (e.g., position on page, page number, etc.), etc. The third metadata type may be the same as or different from the first metadata type. However, the third metadata type is a different type of metadata from the second metadata type.

[0095] In example embodiments where a second metadata type and third metadata type are selected by the second stage metadata processor 322 for the ranked group of applications, the second and third metadata types may be one of the following: the name of the application and the date/timestamp when the application was installed; the name of the application and the date/timestamp when the application was uninstalled; the name of the application and the date/timestamp when the application was updated; the name of the application and the date/timestamp when the application was first used/launched on the computing device; the name of the application and the date/timestamp when the application transitioned from a free trial to a full or paid version; the name of the developer of the application and the date/timestamp when the application was installed; the name of the developer of the application and the date/timestamp when the application was uninstalled; the name of the developer of the application and the date/timestamp when the application was updated; the name of the developer of the application and the date/timestamp when the application was first used by the user on the computing device; or the name of the developer of the application and the date/timestamp when the application transitioned from a free trial to a full or paid version. Other metadata types and combinations are also contemplated without departing from the teachings of the present disclosure.

[0096] After selecting a second metadata type for each ranked group of applications (as selected by the ranked group of applications selector 320), an example embodiment of the second stage metadata processor 322 is configurable or configured to obtain (for either two or more applications in the ranked group of applications or all of the applications in the ranked group of applications) a second metadata of the application corresponding to the selected second metadata type. The second stage metadata processor 322 is then configurable or configured to provide the obtained second metadata (that corresponds to the selected second metadata type) of each application in the ranked group of applications to the final group of

applications generator 330 and/or one or more other elements of the unique device identifier processor 300.

[0097] Although the present disclosure may describe the second stage metadata processor 322 as selecting a second metadata type for each ranked group of applications and obtaining a second metadata for each application in each ranked group of applications, it is to be understood that the second stage metadata processor 322 may also select two or more metadata types (e.g., a second metadata type and third metadata type; second metadata type, third metadata type, and fourth metadata type; etc.) for each ranked group of applications and obtain two or more metadata (each corresponding to the selected two or more metadata types) for each application in each ranked group of applications without departing from the teachings of the present disclosure.

[0098] It is also to be understood that the second stage metadata processor 332 is not limited to only selecting metadata types and obtaining metadata of applications in the ranked group of applications. For example, the second stage metadata processor 322 may also select, for the ranked group of applications, information pertaining to the computing device 10 (e.g., brand of the computing device 10, model of the computing device 10, operating system of the computing device 10, date of last update of the operating system of the computing device 10, etc.). In such example embodiments where information pertaining to the computing device 10 are selected, the second stage metadata processor 322 is configurable or configured to obtain such information pertaining to the computing device 10 (e.g., if the brand of the computing device 10 is selected, then the information obtained in action 580 may be Apple, Samsung, Google, Xiaomi, etc.; if the model of the computing device 10 is selected, the information obtained in action 580 may be iPhone 12, Note, Mi MX2, etc.; etc.).

[0099] The second stage metadata processor 332 is then configurable or configured to provide results (e.g., second metadata type, second metadata of each application in the ranked group of applications, third metadata type, third metadata of each application in the ranked group of applications, etc.) to the final group of applications selector, the unique device identifier generator 340, and/or other elements of the main processor 200.

[00100] *The final group of applications selector (e.g., final group of applications selector 330).*

[00101] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more final group of applications selectors (e.g., final group of applications selector 330). The final group of applications selector 330 is configurable or configured to

communicate with other elements of the unique device identifier processor 300. For example, the final group of applications selector 330 receives ranked groups of applications generated by the ranked group of applications generator 320. The final group of applications selector 330 also receives selected second metadata types (and in some embodiments, third and/or other metadata types) for ranked groups of applications from the second stage metadata processor 322. The final group of applications selector 330 also receives obtained second metadata (and in some embodiments, third and other metadata) for ranked groups of applications from the second stage metadata processor 322. The final group of applications processor 330 also provides final groups of applications to the unique device identifier generator 340.

[00102] In an example embodiment, the final group of applications selector 330 is configurable or configured to select a final group of applications from the ranked group of applications. Two or more applications (or all of the applications) in the ranked group of applications may be selected by the final group of applications selector 330 based on the results from the second stage metadata processor 322 (e.g., one or more of the following: second metadata type, second metadata, third metadata type, third metadata, information pertaining to the computing device 10, etc.). For example, the final group of applications may be a predetermined number of the applications selected from the ranked group of applications (as received from the ranked group of applications processor 320) based on the ranking of the applications in the ranked group of applications. As another example, the final group of applications selected may be all of the applications in the ranked group of applications. The final group of applications selector 330 then obtains and/or identifies the results from the second stage metadata processor 322 (e.g., one or more of the following: second metadata type, second metadata, third metadata type, third metadata, information pertaining to the computing device 10, etc.) for each of the applications selected in the final group of applications, and provides such identified results to the unique device identifier generator 340.

[00103] For example, if the second metadata type selected by the second stage metadata processor 322 is a name of the application, then the final group of applications selector 330 will provide to the unique device identifier generator 340 the name of each application in the final group of applications.

[00104] **FIGURE 4E** illustrates an example of the results, as provided by the final group of applications selector 330 to the unique device identifier generator 340, which include the name of each application in the final group of applications and information pertaining to the computing device 442. Figure 4E also illustrates an example of performing a concatenating

and/or combining 444 of such results (in the example illustrated in Figure 4E, such results being information pertaining to the computing device 442 and the name of the selected applications in the final group of applications). It is to be understood that such concatenating and/or combining may be performed by the final group of applications selector 330 and/or the unique device identifier generator 340).

[00105] The final group of applications selector 330 is then configurable or configured to provide the selected final group of applications to the unique device identifier generator 340 and/or other elements of the main processor 200.

[00106] *The unique device identifier generator (e.g., unique device identifier generator 340).*

[00107] As illustrated in at least Figure 3, the unique device identifier processor 300 includes one or more unique device identifier generators (e.g., unique device identifier generator 340). The unique device identifier generator 340 is configurable or configured to communicate with other elements of the main processor 200, including the final group applications selector 330, the second stage metadata processor 322, the unique device identifier validator 400, and the results processor 220.

[00108] In an example embodiment, the unique device identifier generator 340 is configurable or configured to receive results from the final group of applications selector 330 (and/or the second stage metadata processor 322). Such results may include metadata type(s), metadata of each application in the final group of applications, and/or information pertaining to the computing device 10. In example embodiments, such results received from the final group of applications selector 330 may be concatenated and/or combined by the unique device identifier generator 340 (and/or the final group of applications selector 330). The unique device identifier generator 340 is then configurable or configured to generate a unique device identifier for the computing device 10 based on the results received from the final group of applications selector 330. For example, the unique device identifier for the computing device 10 may be a concatenation and/or combination of the results received from the final group of applications selector 330 (e.g., see 444 in Figure 4E). As another example, the unique device identifier for the computing device 10 may be a hash of the results received from the final group of applications selector 330. As another example, the unique device identifier for the computing device 10, as generated by the unique device identifier generator 340, may be a hash of a concatenation and/or combination of the results received from the final group of applications selector 330. **FIGURE 4F** illustrates an example of a unique device identifier 452 generated

by hashing a concatenation and/or combination of the results received from the final group of applications selector 330 using an MD5 hashing algorithm. Other hashing algorithms are also contemplated without departing from the teachings of the present disclosure. As another example, the unique device identifier for the computing device 10 may be an encryption of the results received from the final group of applications selector 330. As another example, the unique device identifier for the computing device 10, as generated by the unique device identifier generator 340, may be an encryption of a concatenation and/or combination of the results received from the final group of applications selector 330. As another example, the unique device identifier for the computing device 10 may be any other form of codification and/or transformation of the results received from the final group of applications selector 330. As another example, the unique device identifier for the computing device 10, as generated by the unique device identifier generator 340, may be any other form of codification and/or transformation of a concatenation and/or combination of the results received from the final group of applications selector 330.

[00109] The unique device identifier generator 340 is then configurable or configured to provide the generated unique device identifier to the unique device identifier validator 400 (e.g., when a validation and/or verification of a computing device 10 and/or previously generated unique device identifier is required), the results processor 220 (e.g., for providing, sharing, and/or making available the generated unique device identifier to the website 20, application 20, platform 20, system 20, or the like; for storing the generated unique device identifier in one or more databases 40; for providing, sharing, and/or making available the generated unique device identifier to the computing device 10; for providing, sharing, and/or making available the generated unique device identifier to other main processors 200; for determining a requested/required conclusion, such as whether or not the computing device 10 had previously visited and/or interacted with a particular website, submitted a particular request form, received a particular advertisement, received a particular message, received a particular notification, is an approved, recognized, and/or known computing device, etc.), other elements of the main processor 200, and/or other elements of the system 200.

[00110] The unique device identifier validator (e.g., unique device identifier validator 400).

[00111] As illustrated in at least Figure 2, the main processor 200 for managing computing device identification, verification, and validation includes one or more unique device identifier validators (e.g., unique device identifier validators 400). The unique device

identifier validator 400 is configurable or configured to communicate with other elements of the main processor 200, including the main interface 210, the unique device identifier processor 300, and the results processor 400. The unique device identifier validator 400 is configurable or configured to validate and/or verify unique device identifiers that have been generated for a plurality of computing devices 10.

[00112] In an example embodiment, after the unique device identifier processor 300 generates a unique device identifier for a particular computing device 10 (e.g., a computing device 10), the unique device identifier validator 400 is configurable or configured to validate and/or verify the generated unique device identifier of that particular computing device 10 for a particular purpose 20, website 20, application 20, platform 20, system 20, or the like. More specifically, the unique device identifier validator 400 is configurable or configured to determine whether or not a particular computing device 10 had previously, recently, historically (e.g., within a specific time period), or the like, visited, interacted with, submitted a form using, etc. such particular purposes 20, websites 20, applications 20, platforms 20, systems 20, etc. This is achieved by comparing previously, recently, historically (e.g., within a specific time period), or the like, generated (e.g., as generated by main unique device identifier processor 300) and/or stored (e.g., as stored in database 40) unique device identifiers for such particular purposes 20, websites 20, applications 20, platforms 20, systems 20, etc.

[00113] The unique device identifier validator 400 is then configurable or configured to provide results of the validation and/or verification to the results processor 220 and/or other elements of the main processor 200.

[00114] The results processor (e.g., results processor 220).

[00115] As illustrated in at least Figure 2, the main processor 200 for managing computing device identification, verification, and validation includes one or more results processors (e.g., results processor 220). The results processor 220 is configurable or configured to communicate with other elements of the main processor 200 and system 100, including the main interface 210, the unique device identifier processor 300, the unique device identifier validator 400, the computing device 10, the website 20, the application 20, the platform 20, the system 20, the network 30, the communication channel 30, and/or the database 40.

[00116] In an example embodiment, the results processor 220 is configurable or configured to generate a result and/or perform an action based on the generating of the unique device identifier performed by the unique device identifier processor 300 and/or the validation/verification performed by the unique device identifier validator 400. The results

processor 220 is then configurable or configured to communicate such result to the website 20, application 20, platform 20, system 20, or the like, that requested the performing of the generating, validation, and/or verification of the unique device identifier. For example, the results processor 220 may perform one or more of the following: provide, share, and/or make available a generated unique device identifier to the website 20, application 20, platform 20, system 20, or the like; store a generated unique device identifier in one or more databases 40; provide, share, and/or make available a generated unique device identifier to the computing device 10; provide, share, and/or make available a generated unique device identifier to other main processors 200; provide, share, and/or make available a requested/required conclusion, such as whether or not the computing device 10 had previously visited and/or interacted with a particular website, submitted a particular request form, received a particular advertisement, received a particular message, received a particular notification, is an approved, recognized, and/or known computing device, etc.).

[00117] **Example embodiments of a method for managing computing device identification, verification, and validation (e.g., method 500).**

[00118] **FIGURE 5** illustrates an example embodiment of a method (e.g., method 500) for managing computing device identification, verification, and validation. The method 500 includes establishing a communication channel 30 with a computing device 10 (e.g., action 510). Such communication channel 30 may be established between the computing device 10 and an example embodiment of the main processor 200.

[00119] The method 500 may also include identifying an initial set of applications for the computing device 10 (e.g., action 520). For example, the initial set of applications may be all applications that have ever been installed on the computing device 10. As another example, the initial set of applications may be applications that were installed on the computing device 10 within a particular period of time. As another example, the initial set of applications may be applications that were installed but not uninstalled on the computing device 10 within a particular period of time. As another example, the initial set of applications may be applications that were installed and updated on the computing device 10 within a particular period of time. Other examples of initial sets of applications are also contemplated without departing from the teachings of the present disclosure. Figure 4A illustrates an example of an initial set of applications 402 for a particular computing device 10 (including applications that were factory installed (i.e., those with zero milliseconds in date/time of installation) and those that were installed by the user (i.e., those with non-zero milliseconds in date/time of

installation). Figure 4A also illustrates the date/time of installation 404 of each application and whether or not the application was installed by the user 406 (and not factory installed).

[00120] The method 500 further includes selecting a first group of applications from among the initial set of applications (e.g., action 530). The first group of applications may include applications previously and/or currently installed on the computing device 10. The selecting of the first group of applications may be based on an application selection criteria. For example, the application selection criteria may limit the first group of applications to applications that were installed on the computing device 10 by the user within a particular period of time (e.g., within 1 week of activating or first use of the computing device 10). As another example, the application selection criteria may limit the first group of applications to the first 10 (or more or less) applications that were installed on the computing device 10 by the user. As another example, the application selection criteria may limit the first group of applications to only applications that were installed on the computing device 10 by the user (e.g., only those installed by the user, and excluding factory-installed applications; this can be determined in a variety of ways, for example, by identifying whether a date/timestamp for an installation date/time of an application has any milliseconds (i.e., non-zero milliseconds) in the date/timestamp). Other examples of application selection criteria are also contemplated without departing from the teachings of the present disclosure. Figure 4B illustrates an example of selecting the first group of applications 412 based on applications that were installed by the user (and not factory installed). Figure 4B also illustrates the date/time of installation 414 of each application and whether or not the application was installed by the user 416 (and not factory installed).

[00121] After selecting the first group of applications, the method 500 includes selecting a first metadata type for the first group of applications (e.g., action 540). The first metadata type may include, for example, a date/timestamp when the application was installed on the computing device 10; date/timestamp when the application was uninstalled from the computing device 10; date/timestamp when the application was last updated on the computing device 10; date/timestamp when the application was first used by the user on the computing device 10; date/timestamp when the application transitioned from a free trial to a full or paid version on the computing device 10; name of the application; name of the developer of the application; version number of the application; number of times the application has been used, launched, and/or selected by the user within a predetermined time period; duration of use of the

application within a predetermined time period; location of the application (e.g., position on page, page number, etc.), etc.

[00122] The method 500 further includes obtaining (and/or identifying), for each application in the first group of applications, a first metadata of the application (e.g., action 550). The first metadata of each application is the metadata of the application that corresponds to the first metadata type. For example, if the selected first metadata type is a date/timestamp when the application was installed on the computing device 10, then the first metadata of each application in the first group of applications obtained in action 550 is the date/timestamp when the application was installed on the computing device 10. As another example, if the selected first metadata type is a date/timestamp when the application was first used/launched by the user on the computing device 10, then the first metadata of each application in the first group of applications obtained in action 550 is the date/timestamp when the application was first used/launched by the user on the computing device 10. As another example, if the selected first metadata type is a name of the application, then the first metadata of each application in the first group of applications obtained in action 550 is the name of the application.

[00123] After obtaining and/or identifying the first metadata for each application in the first group of application, the method 500 includes generating a ranked group of applications (e.g., action 560). The ranked group of applications may be generated by ranking two or more applications (or all of the applications) in the first group of applications based on the obtained first metadata of the two or more applications in the first group of applications. For example, if the selected first metadata type is a date/timestamp when the application was installed on the computing device 10, then the ranked group of applications is a ranking of applications based on the obtained first metadata of each application (i.e., ranking based on the date/timestamp when the application was installed on the computing device 10). It is to be understood that two or more applications may first be selected from the first group of applications before performing the action 550 of obtaining a first metadata of the application without departing from the teachings of the present disclosure. Figure 4C illustrates an example of a ranked group of applications 422 (for all applications in the first group of applications) generated based on the first metadata type being the date/timestamp of installation of the application and the first metadata being details of the date/timestamp of installation 424. Figure 4C also illustrates whether or not the application was installed by the user 426 (and not factory installed). Figure 4D illustrates an example of a ranked group of applications 432 (for only a predetermined number of applications in the first group of applications) generated based on the first metadata

type being the date/timestamp of installation of the application and the first metadata being details of the date/timestamp of installation 434. Figure 4D also illustrates whether or not the application was installed by the user 436 (and not factory installed).

[00124] The method 500 further includes selecting, for the ranked group of applications, a second metadata type from among a plurality of metadata types (e.g., action 570). The method 500 may also include selecting, for the ranked group of applications, a third metadata type from among a plurality of metadata types. In example embodiments where the second and third metadata types are selected, the second metadata type may be a different metadata type than (or same metadata type as) the third metadata type. The second and third metadata types may include, but is not limited to, the name of each application in the ranked group of applications; name of the developer of the application; date/timestamp when the application was installed on the computing device 10; date/timestamp when the application was uninstalled from the computing device 10; date/timestamp when the application was last updated on the computing device 10; date/timestamp when the application was first used by the user on the computing device 10; date/timestamp when the application transitioned from a free trial to a full or paid version on the computing device 10; name of the application; name of the developer of the application; version number of the application; number of times the application has been used, launched, and/or selected by the user within a predetermined time period; duration of use of the application within a predetermined time period; location of the application (e.g., position on page, page number, etc.), etc. For example, in example embodiments where both the second and third metadata types are selected for each application, the second and third metadata types may be one of the following: the name of the application and the date/timestamp when the application was installed; the name of the application and the date/timestamp when the application was uninstalled; the name of the application and the date/timestamp when the application was updated; the name of the application and the date/timestamp when the application was first used/launched on the computing device; the name of the application and the date/timestamp when the application transitioned from a free trial to a full or paid version; the name of the developer of the application and the date/timestamp when the application was installed; the name of the developer of the application and the date/timestamp when the application was uninstalled; the name of the developer of the application and the date/timestamp when the application was updated; the name of the developer of the application and the date/timestamp when the application was first used by the user on the computing device; or the name of the developer of the application and the date/timestamp when the

application transitioned from a free trial to a full or paid version. It is to be understood in the present disclosure that one or more other metadata types, such as a fourth metadata type, fifth metadata type, etc., may also be selected for the ranked group of applications without departing from the teachings of the present disclosure. The method 500 may also include selecting, for the ranked group of applications, information pertaining to the computing device 10 (e.g., brand of the computing device 10, model of the computing device 10, etc.) in addition to the second metadata type (and third metadata type, if selected in addition to the second metadata type).

[00125] The method 500 further includes obtaining a second metadata of each application in the ranked group of applications (e.g., action 580). The second metadata of each application in the ranked group of applications is a metadata corresponding to the second metadata type. In example embodiments where the second and third metadata types are selected (i.e., in action 570), the method 500 may also include obtaining a third metadata of each application in the ranked group of applications. The third metadata of each application in the ranked group of applications is a metadata corresponding to the third metadata type. In example embodiments where information pertaining to the computing device 10 are also selected (i.e., in action 570), the method 500 may also include obtaining such information pertaining to the computing device 10 (e.g., if the brand of the computing device 10 is selected, then the information obtained in action 580 may be Apple, Samsung, Google, Xiaomi, etc.; if the model of the computing device 10 is selected, the information obtained in action 580 may be iPhone 12, Note, Mi MX2, etc.; etc.). Figure 4E illustrates an example of selecting, for action 580, the second metadata (name of application) and information pertaining to the computing device 442. Figure 4E also illustrates an example of concatenating or combining 444 information pertaining to the computing device 442 and the second metadata of the selected applications in the ranked group of applications.

[00126] The method 500 further includes generating a unique device identifier for the computing device 10 (e.g., action 590). The unique device identifier is generated based on at least the second metadata and the third metadata. The unique device identifier may be generated in one or more of a plurality of ways. For example, as illustrated in Figure 4E, the information obtained for action 580 may be concatenated or combined together into a string 444. As illustrated in Figure 4F, the string of information concatenated or combined together 444 may be hashed (e.g., MD5 hash, etc.) to form the unique device identifier.

[00127] **Example embodiments of a method for managing computing device identification, verification, and validation (e.g., method 600).**

[00128] **FIGURE 6** illustrates an example embodiment of a method (e.g., method 600) for managing computing device identification, verification, and validation. The method 600 includes establishing a communication channel 30 with a computing device 10 (e.g., action 610). Such communication channel 30 may be established between the computing device 10 and an example embodiment of the main processor 200.

[00129] The method 600 further includes selecting a first group of applications (e.g., action 620). The first group of applications may include applications previously and/or currently installed on the computing device 10. The selecting of the first group of applications may be based on an application selection criteria. For example, the application selection criteria may limit the first group of applications to applications that were installed on the computing device 10 by the user within a particular period of time (e.g., within 1 week of activating or first use of the computing device 10). As another example, the application selection criteria may limit the first group of applications to the first 10 (or more or less) applications that were installed on the computing device 10 by the user. As another example, the application selection criteria may limit the first group of applications to only applications that were installed on the computing device 10 by the user (e.g., only those installed by the user, and excluding factory-installed applications; this can be determined in a variety of ways, for example, by identifying whether a date/timestamp for an installation date/time of an application has any milliseconds (i.e., non-zero milliseconds) in the date/timestamp). Other examples of application selection criteria are also contemplated without departing from the teachings of the present disclosure. Figure 4B illustrates an example of selecting the first group of applications 412 based on applications that were installed by the user (and not factory installed). Figure 4B also illustrates the date/time of installation 414 of each application and whether or not the application was installed by the user 416 (and not factory installed).

[00130] The method 600 further includes generating a ranked group of applications (e.g., action 630). The ranked group of applications may be generated by ranking two or more applications (or all of the applications) in the first group of applications based on a predetermined metadata of the two or more applications in the first group of applications. For example, if the predetermined metadata is a date/timestamp when the application was installed on the computing device 10, then the ranked group of applications is a ranking of applications based on the date/timestamp when the application was installed on the computing device 10. Figure 4C illustrates an example of a ranked group of applications 422 (for all applications in the first group of applications) generated based on a predetermined metadata being the

date/timestamp of installation of the application 424. Figure 4C also illustrates whether or not the application was installed by the user 426 (and not factory installed). Figure 4D illustrates an example of a ranked group of applications 432 (for only a predetermined number of applications in the first group of applications) generated based on a predetermined metadata type being the date/timestamp of installation of the application 434. Figure 4D also illustrates whether or not the application was installed by the user 436 (and not factory installed).

[00131] The method 600 further includes selecting, for the ranked group of applications, a first metadata type from among a plurality of metadata types (e.g., action 640). The method 600 may also include selecting, for the ranked group of applications, a second metadata type from among a plurality of metadata types. In example embodiments where the first and second metadata types are selected, the first metadata type may be a different metadata type than (or same metadata type as) the second metadata type. The first and second metadata types may include, but is not limited to, the name of each application in the ranked group of applications; name of the developer of the application; date/timestamp when the application was installed on the computing device 10; date/timestamp when the application was uninstalled from the computing device 10; date/timestamp when the application was last updated on the computing device 10; date/timestamp when the application was first used by the user on the computing device 10; date/timestamp when the application transitioned from a free trial to a full or paid version on the computing device 10; name of the application; name of the developer of the application; version number of the application; number of times the application has been used, launched, and/or selected by the user within a predetermined time period; duration of use of the application within a predetermined time period; location of the application (e.g., position on page, page number, etc.), etc. For example, in example embodiments where both the first and second metadata types are selected for each application, the first and second metadata types may be one of the following: the name of the application and the date/timestamp when the application was installed; the name of the application and the date/timestamp when the application was uninstalled; the name of the application and the date/timestamp when the application was updated; the name of the application and the date/timestamp when the application was first used/launched on the computing device; the name of the application and the date/timestamp when the application transitioned from a free trial to a full or paid version; the name of the developer of the application and the date/timestamp when the application was installed; the name of the developer of the application and the date/timestamp when the application was uninstalled; the name of the developer of the application and the

date/timestamp when the application was updated; the name of the developer of the application and the date/timestamp when the application was first used by the user on the computing device; or the name of the developer of the application and the date/timestamp when the application transitioned from a free trial to a full or paid version. It is to be understood in the present disclosure that one or more other metadata types, such as a third metadata type, fourth metadata type, etc., may also be selected for the ranked group of applications without departing from the teachings of the present disclosure. The method 600 may also include selecting, for the ranked group of applications, information pertaining to the computing device 10 (e.g., brand of the computing device 10, model of the computing device 10, etc.) in addition to the first metadata type (and second metadata type, if selected in addition to the first metadata type).

[00132] The method 600 further includes obtaining a first metadata of each application in the ranked group of applications (e.g., action 650). The first metadata of each application in the ranked group of applications is a metadata corresponding to the first metadata type. In example embodiments where the first and second metadata types are selected (i.e., in action 640), the method 600 may also include obtaining a second metadata of each application in the ranked group of applications. The second metadata of each application in the ranked group of applications is a metadata corresponding to the second metadata type. In example embodiments where information pertaining to the computing device 10 are also selected (i.e., in action 640), the method 600 may also include obtaining such information pertaining to the computing device 10 (e.g., if the brand of the computing device 10 is selected, then the information obtained in action 580 may be Apple, Samsung, Google, Xiaomi, etc.; if the model of the computing device 10 is selected, the information obtained in action 580 may be iPhone 12, Note, Mi MX2, etc.; etc.). Figure 4E illustrates an example of selecting, for action 650, the first metadata (name of application) and information pertaining to the computing device 442. Figure 4E also illustrates an example of concatenating or combining 444 information pertaining to the computing device 442 and the second metadata of the selected applications in the ranked group of applications.

[00133] The method 600 further includes generating an information payload (e.g., action 660). The information payload includes at least the first metadata of each application in the ranked group of applications.

[00134] The method 600 further includes generating the unique device identifier for the computing device 10 (e.g., action 670). The unique device identifier is generated based on at least the information payload. The unique device identifier may be generated in one or more

of a plurality of ways. For example, as illustrated in Figure 4E, the information obtained for action 650 may be concatenated or combined together into a string 444, followed by generating of an information payload in action 660. As illustrated in Figure 4F, the information payload may then be hashed (e.g., MD5 hash, etc.) to form the unique device identifier.

[00135] **Example embodiments of a method for managing computing device identification, verification, and validation (e.g., method 700).**

[00136] **FIGURE 7** illustrates an example embodiment of a method (e.g., method 700) for managing computing device identification, verification, and validation. The method 700 includes establishing a communication channel 30 with a computing device 10 (e.g., action 710). Such communication channel 30 may be established between the computing device 10 and an example embodiment of the main processor 200.

[00137] The method 700 may also include selecting a first group of applications (e.g., action 720). The first group of applications may include applications previously and/or currently installed on the computing device 10. For example, the first group of applications may be applications that were installed on the computing device 10 by the user within a particular period of time (e.g., within 1 week of activating or first use of the computing device 10). As another example, the first group of applications may be the first 10 (or more or less) applications that were installed on the computing device 10 by the user. As another example, the first group of applications may be only applications that were installed on the computing device 10 by the user (e.g., only those installed by the user, and excluding factory-installed applications; this can be determined in a variety of ways, for example, by identifying whether a date/timestamp for an installation date/time of an application has any milliseconds (i.e., non-zero milliseconds) in the date/timestamp). Other examples of selecting the first group of applications are also contemplated without departing from the teachings of the present disclosure. Figure 4B illustrates an example of selecting the first group of applications 412 based on applications that were installed by the user (and not factory installed). Figure 4B also illustrates the date/time of installation 414 of each application and whether or not the application was installed by the user 416 (and not factory installed).

[00138] After selecting the first group of applications, the method 700 includes generating a ranked group of applications (e.g., action 730). The ranked group of applications is generated by ranking two or more applications that have been selected to be in the first group of applications. For example, the method 700 may first select two or more applications (or all of the applications) from the first group of applications. Thereafter, the method 700 may

include ranking such two or more applications (or all of the applications) from the first group of applications based on one or more criteria, including those described above and in the present disclosure. For example, the ranking of applications may be based on a date/timestamp when the application was installed on the computing device 10. Alternatively or in addition, the ranking of applications may be based on alphabetical order of the name of the application and/or name of the developer of the application. Other ranking approaches for the ranked group of applications are also contemplated without departing from the teachings of the present disclosure. Figure 4C illustrates an example of a ranked group of applications 422 (for all applications in the first group of applications) ranked based on a date/timestamp of installation of the application 424. Figure 4C also illustrates whether or not the application was installed by the user 426 (and not factory installed). Figure 4D illustrates an example of a ranked group of applications 432 (for only a predetermined number of applications in the first group of applications) ranked based on the date/timestamp of installation of the application 434. Figure 4D also illustrates whether or not the application was installed by the user 436 (and not factory installed).

[00139] The method 700 further includes obtaining a first metadata of each application in the ranked group of applications (e.g., action 740). The first metadata of each application in the ranked group of applications is a metadata corresponding to a first metadata type. For example, the first metadata may include one or more of the following: the name of the application; the date/timestamp when the application was installed; the date/timestamp when the application was uninstalled; the date/timestamp when the application was updated; the date/timestamp when the application was first used/launched on the computing device; the date/timestamp when the application transitioned from a free trial to a full or paid version; the name of the developer of the application; etc. In example embodiments, the method 700 may also include obtaining information pertaining to the computing device 10 (e.g., the brand of the computing device 10 such as Apple, Samsung, Google, Xiaomi, etc.; the model of the computing device 10, such as iPhone 12, Note, Mi MX2, etc.; etc.). Figure 4E illustrates an example of obtaining the name of application and information pertaining to the computing device 442. Figure 4E also illustrates an example of concatenating or combining 444 information pertaining to the computing device 442 and the second metadata of the selected applications in the ranked group of applications.

[00140] The method 700 further includes generating the unique device identifier for the computing device 10 (e.g., action 750). The unique device identifier is generated based on at

least the first metadata of each application in the ranked group of applications. The unique device identifier may be generated in one or more of a plurality of ways. For example, as illustrated in Figure 4E, the information obtained for action 740 may be concatenated or combined together into a string 444. As illustrated in Figure 4F, the string of information concatenated or combined together 444 may be hashed (e.g., using MD5 hashing algorithm, SHA-256 hashing algorithm, etc.) to generate the unique device identifier.

[00141] While various embodiments in accordance with the disclosed principles have been described above, it should be understood that they have been presented by way of example only, and are not limiting. Thus, the breadth and scope of the example embodiments described in the present disclosure should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the claims and their equivalents issuing from this disclosure. Furthermore, the above advantages and features are provided in described embodiments, but shall not limit the application of such issued claims to processes and structures accomplishing any or all of the above advantages.

[00142] For example, "communication," "communicate," "connection," "connect," "call," "calling," or other similar terms should generally be construed broadly to mean a wired, wireless, and/or other form of, as applicable, connection between elements, devices, computing devices, telephones, processors, controllers, servers, networks, telephone networks, the cloud, and/or the like, which enable voice and/or data to be sent, transmitted, broadcasted, received, intercepted, acquired, and/or transferred (each as applicable).

[00143] Also, as referred to herein, a processor, device, computing device, telephone, phone, server, gateway server, communication gateway server, and/or controller, may be any processor, computing device, and/or communication device, and may include a virtual machine, computer, node, instance, host, or machine in a networked computing environment. Also as referred to herein, a network or cloud may be or include a collection of machines connected by communication channels that facilitate communications between machines and allow for machines to share resources. Network may also refer to a communication medium between processes on the same machine. Also as referred to herein, a network element, node, or server may be a machine deployed to execute a program operating as a socket listener and may include software instances.

[00144] Database (or memory or storage) may comprise any collection and/or arrangement of volatile and/or non-volatile components suitable for storing data. For example, memory may comprise random access memory (RAM) devices, read-only memory (ROM)

devices, magnetic storage devices, optical storage devices, solid state devices, and/or any other suitable data storage devices. In particular embodiments, database may represent, in part, computer-readable storage media on which computer instructions and/or logic are encoded. Database may represent any number of memory components within, local to, and/or accessible by a processor and/or computing device.

[00145] Various terms used herein have special meanings within the present technical field. Whether a particular term should be construed as such a "term of art" depends on the context in which that term is used. Such terms are to be construed in light of the context in which they are used in the present disclosure and as one of ordinary skill in the art would understand those terms in the disclosed context. The above definitions are not exclusive of other meanings that might be imparted to those terms based on the disclosed context.

[00146] Words of comparison, measurement, and timing such as "at the time," "equivalent," "during," "complete," and the like should be understood to mean "substantially at the time," "substantially equivalent," "substantially during," "substantially complete," etc., where "substantially" means that such comparisons, measurements, and timings are practicable to accomplish the implicitly or expressly stated desired result.

[00147] Additionally, the section headings and topic headings herein are provided for consistency with the suggestions under various patent regulations and practice, or otherwise to provide organizational cues. These headings shall not limit or characterize the embodiments set out in any claims that may issue from this disclosure. Specifically, a description of a technology in the "Background" is not to be construed as an admission that technology is prior art to any embodiments in this disclosure. Furthermore, any reference in this disclosure to "invention" in the singular should not be used to argue that there is only a single point of novelty in this disclosure. Multiple inventions may be set forth according to the limitations of the claims issuing from this disclosure, and such claims accordingly define the invention(s), and their equivalents, that are protected thereby. In all instances, the scope of such claims shall be considered on their own merits in light of this disclosure, but should not be constrained by the headings herein.

Claims

1. A method of identifying computing devices, the method comprising:

establishing a first communication channel between a processor and a first computing device;

selecting, by the processor, a first group of applications installed on the first computing device, the first group of applications selected based on an application selection criteria;

generating, by the processor, a ranked group of applications, the ranked group of applications generated by ranking two or more applications in the first group of applications based on a predetermined metadata of the two or more applications in the first group of applications;

selecting, by the processor for the ranked group of applications, at least a first metadata type and second metadata type from among a plurality of metadata types, wherein the first metadata type is different from the second metadata type;

obtaining, by the processor, a first metadata and second metadata of each application in the ranked group of applications, the first metadata of each application in the ranked group of applications being metadata corresponding to the first metadata type, the second metadata of each application in the ranked group of applications being metadata corresponding to the second metadata type;

generating, by the processor, an information payload, the information payload including at least the first metadata and second metadata of each application in the ranked group of applications; and

generating, by the processor, a unique device identifier for the first computing device, the unique device identifier generated based on at least the information payload.

2. The method of claim 1, wherein:

the first communication channel is established by the processor based on a request received, by the processor, from the first computing device.

3. The method of claim 1, wherein the application selection criteria is generated based on at least one of the following:

the application selection criteria is based on only applications that were installed on the first computing device by a user of the first computing device after a date on which the first computing device was available for sale;

the application selection criteria is based on applications that were not factory-installed on the first computing device;

the application selection criteria is based on only applications installed on the first computing device within a predetermined period of time;

the application selection criteria is based on only applications previously installed on the first computing device that have been uninstalled;

the application selection criteria is based on only applications previously installed on the first computing device that have been uninstalled within a predetermined period of time;

the application selection criteria is based on only applications installed on the first computing device that have been updated within a predetermined period of time;

the application selection criteria is based on only applications installed on the first computing device that have not been updated within a predetermined period of time; and

the application selection criteria is based on only applications installed on the first computing device that have been reinstalled after previously being uninstalled.

4. The method of claim 1, wherein the first metadata type includes at least one of the following:

a name of an application installed on the first computing device;

a name of an application that was previously uninstalled from the first computing device;

a name of an application that was updated on the first computing device; and

a name of a developer of an application installed on the first computing device.

5. The method of claim 1, wherein the first metadata type is generated based on at least one of the following:

the initial set of applications;

the first group of application selected from among the initial set of applications;

the application selection criteria;

the first time period;

the first metadata selection criteria; and

the ranked group of applications.

6. The method of claim 1, wherein the second metadata type includes at least one of the following:

a date/timestamp when an application was installed on the first computing device;

a date/timestamp when an application was uninstalled from the first computing device;

a date/timestamp when an application was first updated on the first computing device;

a date/timestamp when an application was last updated on the first computing device;

a date/timestamp when an application was first launched on the first computing device; and

a date/timestamp when an application transitioned from a free trial status to a full or paid status on the first computing device.

7. The method of claim 1, wherein the second metadata type is generated based on at least one of the following:

the initial set of applications;

the first group of application selected from among the initial set of applications;

the application selection criteria;

the first time period;

the first metadata selection criteria;

the ranked group of applications; and

the second metadata selection criteria.

8. The method of claim 1, wherein:

the first metadata type and second metadata type are different from one another.

9. The method of claim 1, wherein:

the unique device identifier includes at least a portion of the information payload.

10. The method of claim 1, wherein the generating of the unique device identifier further includes:

generating a hash of the information payload.

11. The method of claim 10, wherein:

the unique device identifier includes the hash of the information payload.

12. The method of claim 1, further comprising:

establishing a second communication channel between the processor and a second computing device;

selecting, by the processor, a second group of applications installed on the second computing device, the second group of applications selected based on the application selection criteria;

generating, by the processor, a second ranked group of applications, the second ranked group of applications generated by ranking two or more applications in the second

group of applications based on a predetermined metadata of the two or more applications in the second group of applications;

selecting, by the processor for the second ranked group of applications, at least the first metadata type and second metadata type from among the plurality of metadata types, wherein the first metadata type is different from the second metadata type;

obtaining, by the processor, a first metadata and second metadata of each application in the second ranked group of applications, the first metadata of each application in the second ranked group of applications being metadata corresponding to the first metadata type, the second metadata of each application in the second ranked group of applications being metadata corresponding to the second metadata type;

generating a second information payload, the second information payload including at least the first metadata and second metadata of each application in the second ranked group of applications; and

generating, by the processor, a second unique device identifier for the second computing device, the second unique device identifier generated based on at least the second information payload.

13. The method of claim 12, further comprising:

comparing, by the processor, the first unique device identifier and the second unique device identifier;

responsive to a determination that the first unique device identifier is different from the second unique device identifier:

identifying the first computing device to be a different computing device than the second computing device.

14. The method of claim 13, further comprising:

responsive to a determination that the first unique device identifier is the same as the second unique device identifier:

identifying the first computing device to be the same computing device as the second computing device.