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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 January 2022 (20.01.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/015295 A1

(51) International Patent Classification:
G06F 21/55 (2013.01) *G06Q 30/02* (2012.01)
G06F 21/62 (2013.01)

(21) International Application Number:
PCT/US2020/042007

(22) International Filing Date:
14 July 2020 (14.07.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **GOOGLE LLC** [US/US]; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(72) Inventor: **ZAAROUR, Charbel**; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(74) Agent: **BATAVIA, Neil, M.** et al.; Dority & Manning, PA, P.O. Box 1449, Greenville, SC 29602 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: SYSTEMS AND METHODS OF DELEGATED ANALYTICS COLLECTION

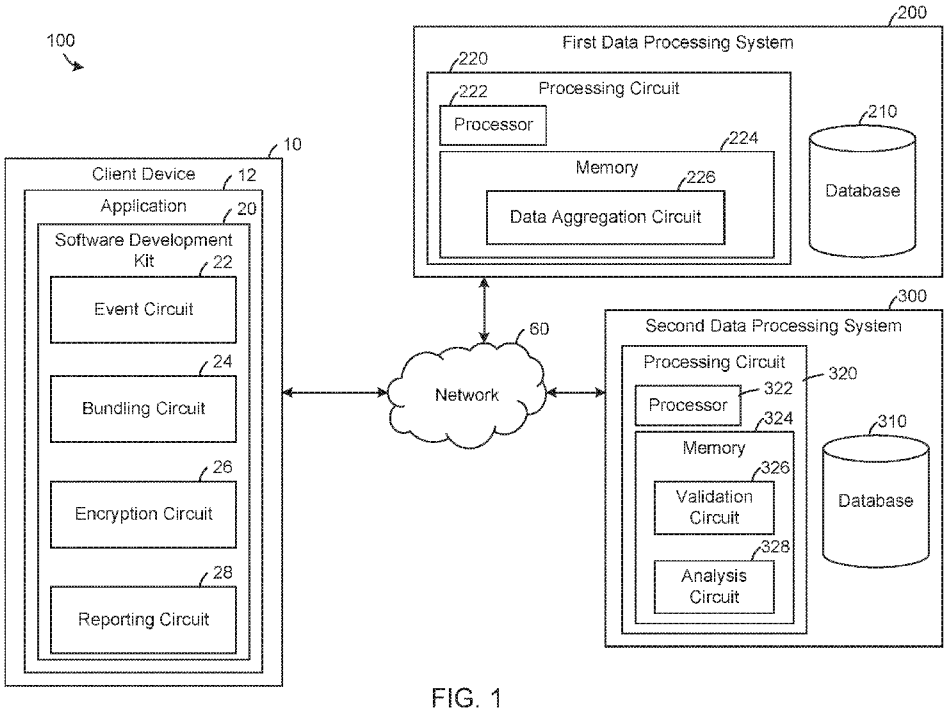

FIG. 1

(57) Abstract: A method including tagging a feature of a mobile application as an event generator, in response to a user interacting with the feature of the mobile application on a mobile device, generating an event having an event type, requesting, by an interaction measurement software development kit (SDK) from the mobile application, interaction data associated with the user interaction with the mobile application based on the event type, and securely transmitting, by the interaction measurement SDK, the interaction data to a first computing device indicated by the interaction measurement SDK.

WO 2022/015295 A1

[Continued on next page]

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

SYSTEMS AND METHODS OF DELEGATED ANALYTICS COLLECTION

BACKGROUND

[0001] It can be helpful for analytics systems to be able to determine information about data from client devices, such as how many devices interacted with a particular content item. To do so, data is often collected at an aggregator and then forwarded to an analysis service. However, forwarding the data from the client device to the aggregator to the analysis service introduces a time delay. Furthermore, if the aggregator becomes unavailable (e.g., the client device cannot send data to the aggregator), then the analysis service will not receive the data and no results will be produced.

SUMMARY

[0002] One implementation of the disclosure relates to a method including tagging a feature of a mobile application as an event generator, in response to a user interacting with the feature of the mobile application on a mobile device, generating an event having an event type, requesting, by an interaction measurement software development kit (SDK) from the mobile application, interaction data associated with the user interaction with the mobile application based on the event type, and securely transmitting, by the interaction measurement SDK, the interaction data to a first computing device indicated by the interaction measurement SDK.

[0003] In some implementations, the interaction measurement SDK bundles interaction data from a number of events and transmits the bundled interaction data in response to a threshold. In some implementations, the threshold is a threshold period of time or a threshold number of events. In some implementations, the interaction SDK generates a signature of the interaction data using a hashing algorithm, and wherein the first computing device validates an authenticity of the interaction data using the signature. In some implementations, the method further includes securely transmitting the interaction data to a second computing device associated with the mobile application, and wherein the interaction SDK removes personal identifiers from the interaction data prior to securely transmitting the interaction data to the first computing device. In some implementations, the interaction data includes an

intermediary identifier, and wherein method further includes identifying, by the second computing device, supplemental data using the intermediary identifier, transmitting, by the second computing device to the first computing device, the supplemental data, and correlating, by the first computing device, the interaction data with the supplemental data. In some implementations, the first computing device does not have access to personally identifying information. In some implementations, the method further includes analyzing, by the first computing device, at least the interaction data to generate data describing usage of the mobile application in real time. In some implementations, analyzing at least the interaction data includes identifying a previous interaction of the user that was not performed using the mobile application.

[0004] Another implementation of the disclosure relates to a system for correlating data from different entities, including a client device, a first computing device, and a second computing device, wherein the client device includes a processing circuit having one or more processors and memory, the memory having instructions stored thereon that, when executed by the one or more processors, cause the processing circuit to tag a feature in a mobile application as an event generator, in response to a user interacting with the feature of the mobile application, generate an event having an event type, request interaction data associated with the user interaction with the mobile application based on the event type, and securely transmit the interaction data to the first computing device, and wherein the instructions constitute a software development kit (SDK).

[0005] In some implementations, wherein the instructions cause the one or more processors to bundle interaction data from a number of events and transmit the bundled interaction data in response to a threshold. In some implementations, the threshold is a threshold period of time or a threshold number of events. In some implementations, the instructions cause the one or more processors to generate a signature of the interaction data using a hashing algorithm, and wherein the first computing device validates an authenticity of the interaction data using the signature. In some implementations, the instructions further cause the one or more processors to securely transmit the interaction data to the second computing device and remove personal identifiers from the interaction data prior to securely transmitting the interaction data to the first computing device. In some implementations,

wherein the interaction data includes an intermediary identifier, wherein the second computing device identifies supplemental data using the intermediary identifier, wherein second computing device transmits the supplemental data to the first computing device, and wherein the first computing device correlates the interaction data with the supplemental data. In some implementations, the first computing device does not have access to personally identifying information. In some implementations, the first computing device analyzes at least the interaction data to generate data describing usage of the mobile application in real time. In some implementations, analyzing at least the interaction data includes identifying a previous interaction of the user that was not performed using the mobile application.

[0006] Another implementation of the disclosure relates to a computer-readable medium having instructions constituting a software development kit (SDK) for a mobile device stored thereon that, when executed by one or more processors, cause the one or more processors to tag a feature of a mobile application as an event generator, in response to a user interacting with the feature of the mobile application on a mobile device, generate an event having an event type, request interaction data from the mobile device associated with the user interaction with the mobile application based on the event type, and securely transmit the interaction data to a first computing device.

[0007] In some implementations, wherein the SDK bundles interaction data from a number of events and transmits the bundled interaction data in response to a threshold, wherein the threshold is a threshold period of time or a threshold number of events, wherein the interaction SDK generates a signature of the interaction data using a hashing algorithm, and wherein the first computing device validates an authenticity of the interaction data using the signature. In some implementations, the instructions further cause the one or more processors to securely transmit the interaction data to a second computing device and remove personal identifiers from the interaction data prior to securely transmitting the interaction data to the first computing device.

[0008] The various aspects and implementations may be combined where appropriate.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 is a diagram illustrating various entities interacting over a network, according to an illustrative implementation.

[0010] FIG. 2 is a diagram illustrating an architecture for data transfer and correlation, according to an implementation.

[0011] FIG. 3 is a flow diagram illustrating a method of generating and transmitting event data using the architecture of FIG. 2, according to an illustrative implementation.

[0012] FIG. 4 is a flow diagram illustrating a method of determining supplemental data using the architecture of FIG. 2, according to an illustrative implementation.

[0013] FIG. 5 is a flow diagram illustrating a method of generating analytics results using the architecture of FIG. 2, according to an illustrative implementation.

[0014] FIG. 6 is a block diagram of a computing system, according to an illustrative implementation.

DETAILED DESCRIPTION

[0015] Following below are more detailed descriptions of various concepts related to, and implementations of, methods, apparatuses, and systems for collecting event data in a timely and redundant manner while preserving privacy. The various concepts introduced above and discussed in greater detail below may be implemented in any of numerous ways, as the described concepts are not limited to any particular manner of implementation.

[0016] In many domains, it may be desirable to collect event data from mobile devices. For example, in the event of a malfunction (e.g., a software crash, a hardware failure, etc.), it is often desirable to collect event data such as device settings, operating parameters, and the like in order to diagnose and fix the malfunction. In other domains, it may be desirable for a provider of third party content to determine a rate of interaction with that third party content.

[0017] System and methods of the present disclosure relate generally to reporting mobile device event data. More specifically, systems and methods of the present disclosure relate to

a unique software development kit (SDK) and computer architecture methodologies to collect and correlate event data from mobile devices.

[0018] Typically, event data from mobile devices is sent to a single endpoint (e.g., an aggregator, etc.) and then forwarded to an analysis system for analysis (e.g., correlation, etc.). For example, event data generated by a first mobile application may be transmitted to servers controlled by a developer of the first mobile application which may collect the data and forward it to an analysis system which determines aggregate statistics associated with the event data. However, the single endpoint may not always be available. For example, the endpoint may experience unexpected failures or routine maintenance which prevent it from receiving data from mobile devices and/or forwarding data to the analysis system. It is desirable to have a robust architecture that facilitates analysis of event data even if the first endpoint becomes unavailable. Therefore, there is a need for a unique SDK and computer architecture methodology to facilitate transmission of event data directly to the analysis system.

[0019] As discussed above, event data from mobile devices is generally sent to a single endpoint (e.g., an aggregator, etc.) and then forwarded to an analysis system for analysis. For example, a developer may receive event data and send the event data to a third party for analysis (e.g., because analyzing the event data is prohibitively difficult, complex, and/or expensive for the developer, etc.). However, transmitting event data from a client device to an aggregator and then to an analysis system introduces a time delay. It may be undesirable to introduce a time delay. For example, an analysis system may be used to identify bugs in application 12, such as link rot. The time delay introduced by forwarding data from client devices to an aggregator and then to the analysis system may use additional system and network resources which could be more efficiently deployed elsewhere and may increase the amount of time it takes to identify a bug in application 12. For example, the analysis system may detect a stale hyperlink several days after it goes stale (e.g., ceases to function properly, etc.). Therefore, there is a need for a unique SDK and computer architecture methodology to facilitate timely analysis of application events.

[0020] Aspects of the present disclosure provide improved data reporting and analysis pathways and computer architectures. The pathways and architectures may be used to report

and analyze mobile device event data in a robust and timely manner that prevents exposing PII to third parties.

[0021] To ensure robustness against endpoint failure, systems and methods of the present disclosure introduce an SDK that facilitates sending data to a second endpoint. A non-limiting example implementation is as follows: a user using a desktop device may view a first content item. A first computing system may register that the user viewed the first content item. The user using a mobile application may then interact with a second content item, thereby causing the mobile application to transmit event data associated with the interaction to the first computing device. A SDK embedded in the mobile application may register that the user interacted with the second content item and may prepare event data associated with the interaction for transmission. For example, the event data may include the content item, a time, and a source of the content item (e.g., a website, etc.). The SDK may remove any personal identifiers from the event data, such as a mobile device identifier. The SDK may then securely transmit the event data to a second computing device. The first computing device may identify the user and prepare supplemental data associated with the user's interaction with the second content item. For example, the first computing device may identify data corresponding to the user's interaction with the first content item. The first computing device may remove any personal identifiers from the supplemental data and transmit the supplemental data to the second computing device. The second computing device may correlate the event data with the supplemental data to determine a result.

[0022] In various embodiments, the SDK of the present disclosure removes identifiers from event data before transmitting the event data to the analysis system. In various embodiments, the SDK of the present disclosure facilitates correlation of data through intermediary identifiers. For example, the SDK of the present disclosure may transmit event data having an intermediary identifier (e.g., an event identifier, etc.) to a first endpoint (e.g., an aggregator, etc.) and the analysis system. The first endpoint may be associated with an application (e.g., a server maintained by the developers of the application, etc.). In some implementations, the first endpoint may independently maintain user information such as account balances associated with user accounts of the application. The first endpoint may identify supplemental data. For example, the first endpoint may maintain a database having

interaction data and may search the database for interaction data associated with a user. The supplemental data may include an intermediary identifier. The first endpoint may transmit the supplemental data to the analysis system which may correlate the supplemental data with the event data using the intermediary identifier, thereby preserving PII.

[0023] Referring now to FIG. 1, a system 100 for collecting event data from mobile devices is shown, according to an illustrative implementation. System 100 includes client device 10, first data processing system 200, and second data processing system 300. In various implementations, components of system 100 communicate over network 60. Network 60 may include computer networks such as the Internet, local, wide, metro or other area networks, intranets, satellite networks, other computer networks such as voice or data mobile phone communication networks, combinations thereof, or any other type of electronic communications network. Network 60 may include or constitute a display network (e.g., a subset of information resources available on the Internet that are associated with a content placement or search engine results system, or that are eligible to include third party content items as part of a content item placement campaign). In various implementations, network 60 facilitates secure communication between components of system 100. As a non-limiting example, network 60 may implement transport layer security (TLS), secure sockets layer (SSL), hypertext transfer protocol secure (HTTPS), and/or any other secure communication protocol.

[0024] Client device 10 may be a mobile computing device, smartphone, tablet, or any other device configured to facilitate receiving, displaying, and interacting with content (e.g., web pages, mobile applications, etc.). Client device 10 may include an application 20 to receive and display online content and to receive user interaction with the online content. For example, application 20 may be a web browser. Additionally or alternatively, application 20 may be a mobile application associated with a specific merchant.

[0025] In various implementations, application 20 interacts with a content publisher to receive online content. For example, application 20 may receive an information resource from a content publisher. The information resources may include web-based content items such as a web page or other online documents. The information resources may include

instructions (e.g., scripts, executable code, etc.) that when interpreted by application 20 cause application 20 to display a graphical user interface such as an interactable web page to a user.

[0026] Application 20 is shown to include software development kit 20 having event circuit 22, bundling circuit 24, encryption circuit 26, and reporting circuit 28. Software development kit (SDK) 20 may include a collection of software development tools contained in a package. SDK 20 may include an application programming interface (API). In some implementations, SDK 20 includes one or more libraries having reusable functions that interface with a particular system software (e.g., iOS, Android, etc.). SDK 20 may facilitate embedding functionality in application 12. For example, a developer may use SDK 20 to automatically transmit event data whenever an event of a specific type occurs on application 12. As a further example, SDK 20 may include a reusable function configured to collect and report device analytics and a developer may insert the reusable function into the instructions of application 12 to cause the reusable function to be called during specific actions of application 12. In some implementations, event circuit 22, building circuit 24, encryption circuit 26, and/or reporting circuit 28 are functionalities provided by SDK 20 (e.g., reusable functions, etc.).

[0027] Event circuit 22 may detect events within application 12. In various implementations, event circuit 22 may be configured to trigger other functionality based on detecting specific events (e.g., transactions, in-app purchases, achieving a certain level in an in-app game, performing a certain number of actions, spending a certain amount of time interacting with an application, etc.). For example, event circuit 22 may trigger bundling circuit 24 upon detecting an event within application 12. In various implementations, SDK 20 includes a function that is embedded in application 12 to trigger event circuit 22. For example, a developer may include a function of SKD 20 in a transaction confirmation functionality of application 12 that causes event circuit 22 to detect a confirmed transaction. It should be understood that events may include any action important to a developer within an application and are not limited to the examples expressly contemplated herein. In various implementations, event circuit 22 is configured to differentiate between different types of events. For example, event circuit 22 may trigger a first set of actions based on a first type of detected event and may trigger a second set of actions based on a second type of detected

event. In various implementations, event circuit 22 is configured to collect event data associated with the detected event and transmit the collected event data to bundling circuit 24.

[0028] Bundling circuit 24 may bundle (e.g., aggregate, etc.) event data. In various implementations, bundling circuit 24 receives event data associated with a detected event from event circuit 22. Bundling circuit 24 may collect event data from one or more events and bundle the event data for transmission. For example, bundling circuit 24 may collect event data from ten events and combine the event data into a single bundle. Event data may include a timestamp of the event, a name of the event, and/or parameters of the event (e.g., a purchased item, a price, a currency, discounts, subscription information, etc.). In some implementations, bundling circuit 24 transmits bundles to encryption circuit 26. Additionally or alternatively, bundling circuit 24 may transmit bundles to reporting circuit 28. In various implementations, bundling circuit 24 generates a data bundle. The data bundle may include a bundle index, a bundle timestamp, bundle data, and a bundle signature. In various implementations, the bundle signature is generated by encryption circuit 26, discussed below. In some implementations, the bundle index specifies where specific event data is located in the bundle data. For example, the bundle data may include a byte array and the bundle index may include an index to the byte array.

[0029] Encryption circuit 26 may encrypt data to produce encrypted data. For example, encryption circuit 26 may encrypt bundled event data. Additionally or alternatively, encryption circuit 26 may perform various obliviating functions on received data. For example, encryption circuit 26 may remove identifiers (e.g., IP address, device identifiers, etc.), fragment event data, add noise, or perform other functions to anonymize data. In various implementations, encryption circuit 26 implements asymmetric encryption. For example, encryption circuit 26 may implement a Rivest-Shamir-Adleman (RSA) cryptosystem. In various implementations, encryption circuit 26 may receive a public key from first data processing system 200 and use the public key to encrypt received data. Additionally or alternatively, encryption circuit 26 may generate a signature associated with received data. For example, encryption circuit 26 may generate a hash of bundled event data received from bundling circuit 24. One or more hashing functions may be used. For

example, encryption circuit 26 may implement a SHA-2, Scrypt, Balloon, and/or Argon2 hashing function.

[0030] Reporting circuit 28 may transmit bundled event data to first data processing system 200 and/or second data processing system 300. In various implementations, reporting circuit 28 transmits data via network 60. Reporting circuit 28 may confirm the transmission of data. For example, reporting circuit 28 may transmit bundled event data to first data processing system 200 and receive a confirmation that the bundled event data was received successfully. In some implementations, reporting circuit 28 first attempts to transmit bundled event data to first data processing system 200 and if that fails then attempts to transmit bundled event data to second data processing system 300. Additionally or alternatively, reporting circuit 28 may transmit data to first data processing system 200 and second data processing system 300 in parallel. In some implementations, reporting circuit 28 transmits different data to first data processing system 200 and second data processing system 300. For example, reporting circuit 28 may transmit encrypted event data having device identifiers to first data processing system 200 and may transmit unencrypted event data without device identifiers to second data processing system 300. Additionally or alternatively, reporting circuit 28 may transmit the same data to first data processing system 200 and second data processing system 300.

[0031] In various implementations, reporting circuit 28 transmits data periodically. For example, reporting circuit 28 may transmit data at a predefined time. As another example, reporting circuit 28 may transmit data on an interval (e.g., every ten minutes, every ten hours, etc.). Additionally or alternatively, reporting circuit 28 may transmit data in response to a threshold. For example, reporting circuit 28 may transmit data in response to bundling circuit receiving a threshold number of event data from events (e.g., ten events, one-hundred events, etc.). In some implementations, reporting circuit 28 transmits data dynamically. For example, reporting circuit 28 may transmit data in response to client device 10 being connected to a charging source. As a further example, reporting circuit 28 may transmit data in response to the transmission bundle reaching a specified data size.

[0032] In various implementations, reporting circuit 28 reports metrics. For example, reporting circuit 28 may transmit metrics alongside each data bundle. The metrics may

include a size of the data bundle, a timestamp of the transmission and/or generation of the data bundle, a data bundle index, an SDK identifier, and/or a signature of the data bundle. In various implementations, the SDK identifier includes information associated with SDK 20. For example, the SDK identifier may include a version number of SDK 20. The signature of the data bundle may include a hash of the data bundle contents as discussed above with reference to encryption circuit 26.

[0033] First data processing system 200 may receive event data from SDK 20 and/or generate supplemental data. In various implementations, first data processing system 200 receives a data bundle from SDK 20, identifies supplemental data based on the contents of the data bundle, and transmits the supplemental data to second data processing system 300. In some implementations, first data processing system 200 anonymizes the supplemental data before transmission to second data processing system 300. For example, first data processing system 200 may remove personal identifiers from the supplemental data before transmission to second data processing system 300. First data processing system 200 may be a server, distributed processing cluster, cloud processing system, or any other computing device. First data processing system 200 may include or execute at least one computer program or at least one script. In some implementations, first data processing system 200 includes combinations of software and hardware, such as one or more processors configured to execute one or more scripts.

[0034] First data processing system 200 is shown to include database 210 and processing circuit 220. Database 210 may store supplemental data. For example, database 210 may include information associated with previous content interactions. As an additional example, a user using a desktop computer may navigate to a website and interact with a content item. First data processing system 200 may receive information associated with the user's interaction with the content item and store the information in database 210. The supplemental data may include content identifiers, device identifiers, user identifiers, click-streams, and/or the like. Database 210 may include one or more storage mediums. The storage mediums may include but are not limited to magnetic storage, optical storage, flash storage, and/or RAM. First data processing system 200 may implement or facilitate various APIs to perform database functions (i.e., managing data stored in database 210). The APIs

can be but are not limited to SQL, ODBC, JDBC, and/or any other data storage and manipulation API.

[0035] Processing circuit 220 may include processor 222 and memory 224. Memory 224 may have instructions stored thereon that, when executed by processor 222, cause processing circuit 220 to perform the various operations described herein. The operations described herein may be implemented using software, hardware, or a combination thereof. Processor 222 may include a microprocessor, ASIC, FPGA, etc., or combinations thereof. In many implementations, processor 222 may be a multi-core processor or an array of processors. Memory 224 may include, but is not limited to, electronic, optical, magnetic, or any other storage devices capable of providing processor 222 with program instructions. Memory 224 may include a floppy disk, CD-ROM, DVD, magnetic disk, memory chip, ROM, RAM, EEPROM, EPROM, flash memory, optical media, or any other suitable memory from which processor 222 can read instructions. The instructions may include code from any suitable computer programming language such as, but not limited to, C, C++, C#, Java, JavaScript, Perl, HTML, XML, Python and Visual Basic.

[0036] Memory 224 may include data aggregation circuit 226. Data aggregation circuit 226 may identify supplemental data from database 210. In various embodiments, data aggregation circuit 226 receives a data bundle from SDK 20 and analyzes contents of the data bundle to identify supplemental data. For example, aggregation circuit 226 may receive a data bundle including an event associated with a first device identifier, may identify supplemental data including previous content interactions associated with the device identifier, and may transmit the supplemental data to second data processing system 300. In various implementations, supplemental data is associated with contents of a data bundle. For example, the supplemental data may be related to event data of an event included in a data bundle through an intermediate identifier. In some implementations, data aggregation circuit 226 verifies the contents of a received data bundle. For example, data aggregation circuit 226 may verify a signature included in the data bundle. Additionally or alternatively, data aggregation circuit 226 may decrypt the contents of the data bundle. For example, data aggregation circuit 226 may receive an asymmetrically encrypted data bundle and decrypt the asymmetrically encrypted data bundle using a private key.

[0037] Second data processing system 300 may receive event data from SDK 20 and facilitate performing analysis on received data to generate information. For example, second data processing system 300 may receive a data bundle including event data from SDK 20 and supplemental data from first data processing system 200 and securely correlate the received data to generate information. As another example, second data processing system 300 may receive first data associated with a transaction from SDK 20 and second data associated with a user interaction with a content item from first data processing system 200 and correlate the first and second data.

[0038] In various embodiments, second data processing system 300 generates aggregate information. For example, second data processing system 300 may determine how many users completed a transaction after interacting with a content item. The aggregate information may describe a number or grouping of online interactions (e.g., interactions with a number of content items). Additionally or alternatively, the aggregate information may describe an individual online interaction (e.g., a single interaction with a single content item). Aggregate information may include a unique identifier. In some implementations, the identifier identifies a marketing campaign. Additionally or alternatively, the identifier may uniquely identify each online interaction. In some implementations, the aggregate information describes one or more interactions associated with content items. For example, aggregate information may include a time, date, and/or location of online interactions. The interactions described by the anonymous interaction data may include viewing a content item (e.g., navigating to a webpage in which a content item is presented and/or determining that the item or a portion of the item is presented within a viewport of the device upon which the webpage is viewed, etc.), selecting/clicking a content item, hovering over a content item, and/or other interactions with a content item.

[0039] Second data processing system 300 may be a server, distributed processing cluster, cloud processing system, or any other computing device. Second data processing system 300 may include or execute at least one computer program or at least one script. In some implementations, second data processing system 300 includes combinations of software and hardware, such as one or more processors configured to execute one or more scripts.

[0040] Second data processing system 300 is shown to include database 310 and processing circuit 320. Database 310 may store received data. For example, database 310 may store event data received from SDK 20 and/or supplemental data received from first data processing system 200. In some implementations, database 310 stores identifiers. For example, database 310 may store event data and supplemental data sharing an intermediary identifier. The identifier may be used later for correlation of anonymous interaction data. Database 310 may include one or more storage mediums. The storage mediums may include but are not limited to magnetic storage, optical storage, flash storage, and/or RAM. Second data processing system 300 may implement or facilitate various APIs to perform database functions (i.e., managing data stored in database 310). The APIs can be but are not limited to SQL, ODBC, JDBC, and/or any other data storage and manipulation API.

[0041] Processing circuit 320 includes processor 322 and memory 324. Memory 324 may have instructions stored thereon that, when executed by processor 322, cause processing circuit 320 to perform the various operations described herein. The operations described herein may be implemented using software, hardware, or a combination thereof. Processor 322 may include a microprocessor, ASIC, FPGA, etc., or combinations thereof. In many implementations, processor 322 may be a multi-core processor or an array of processors. Memory 324 may include, but is not limited to, electronic, optical, magnetic, or any other storage devices capable of providing processor 322 with program instructions. Memory 324 may include a floppy disk, CD-ROM, DVD, magnetic disk, memory chip, ROM, RAM, EEPROM, EPROM, flash memory, optical media, or any other suitable memory from which processor 322 can read instructions. The instructions may include code from any suitable computer programming language such as, but not limited to, C, C++, C#, Java, JavaScript, Perl, HTML, XML, Python and Visual Basic.

[0042] Memory 324 may include validation circuit 326 and analysis circuit 328. Validation circuit 326 may validate data bundles received from SDK 20. In various implementations, validation circuit 326 validates data bundles by verifying a signature included in the data bundles. For example, a signature may include a hash of contents of the data bundle and/or event data and validation circuit 326 may hash the contents of the data bundle and compare the generated hash to the received signature to determine whether the

contents of the data bundle have been modified. In some implementations, if validation circuit 326 determines a data bundle (or specific event data included in the data bundle) to be invalid, it discards the invalid data. Additionally or alternatively, validation circuit 326 may flag the data as invalid which can be used by later systems to generate a confidence metric associated with the analysis results.

[0043] Analysis circuit 328 may receive data and produce information regarding the data. In various implementations, analysis circuit 328 performs statistical operations on received data to produce statistical measurements describing the received dataaa. For example, analysis circuit 328 may determine an interaction rate associated with a marketing campaign. In some implementations, analysis circuit 328 generates demographic information (e.g., user distributions, etc.), geographic results (e.g., location distributions, etc.), and/or audiences (e.g., a target group of users based on one or more parameters, for example users who purchased more than a threshold amount, etc.). In some implementations, analysis circuit 328 correlates event data with supplemental data. For example, analysis circuit 328 may correlate event data associated with an event with supplemental data associated with a content interaction using an intermediate identifier to determine an effect of the content interaction on causing the event. In various implementations, analysis circuit 328 generates information. The information may include an interaction rate, data describing an operation of application 12, and/or the like.

[0044] Referring now to FIG. 2, an improved computer architecture for securely transmitting and correlating data from mobile devices is shown, according to an illustrative implementation. In brief summary, developers may utilize reusable functions of SDK 20 to generate and transmit event data from client device 10 to first data processing system 200 and second data processing system 300 in response to events. For example, functions of SDK 20 embedded in a mobile application (e.g., application 12, etc.) may cause client device 10 to collect and transmit data associated with client device 10 in response to a user confirming a transaction, selecting a menu option, viewing a content item, and/or the like. In response to detecting the event, SDK 20 may collect, bundle, and transmit data to first data processing system 200 and second data processing system 300 for analysis.

[0045] In various implementations, first data processing system 200 may store additional or supplemental data. For example, first data processing system 200 may store data associated with past customers (e.g., user preferences, etc.). As an additional example, first data processing system 200 may store data associated with previous user interactions with content. First data processing system 200 may identify supplemental data based on the data received from client device 10 and may transmit the supplemental data to second data processing system 300. Supplemental data may include previous user interactions (e.g., a record of a user viewing a content item such as a video, a previous transaction, etc.), user demographic information, user preferences, and/or any other data. It should be understood that supplemental data may include any data that a developer has access to that is associated with a user and is not limited to the specific examples explicitly contemplated herein. In various implementations, second data processing system 300 correlates the data from client device and first data processing system 200 to determine a result. Additionally or alternatively, second data processing system 300 may analyze the data received from client device 10 to determine a result. As a non-limiting example, a user shown a video may click on the video. A content provider (e.g., first data processing system 200, etc.) providing the video may wish to know how many users clicked on the video. In some implementations, users may interact with other content provided by the content provider as a result of their interaction with the first content item. For example, a user shown the video may later visit a website maintained by the content provider to purchase an item featured in the video. In some implementations, the interaction is or is associated with an online conversion. SDK 20 may facilitate robust reporting and analysis of user interactions with online content while preserving PII.

[0046] FIG. 2 illustrates a system 110 for transmitting and analyzing event data from mobile devices. In various implementations, client device 10 implements SDK 20 to facilitate collection and transmission of event data. For example, a developer may embed a reusable function of SDK 20 into application 12 on client device 10 to cause client device 10 to transmit data related to specific events. In various implementations, SDK 20 facilitates redundancy in reporting event data. For example, SDK 20 may enable a fallback pathway to report event data. In various implementations, application 12 may natively report application

data to first data processing system 200. SDK 20 may facilitate transmitting event data, redundantly and/or in parallel, to second data processing system 300.

[0047] In various implementations, SDK 20 embedded in application 12 on client device 10 generates event data. For example, a user using application 12 may update a notification preference and SDK 20 may detect the change in preference and generate event data. The event data may include device settings (e.g., software version, hardware settings, configurations, etc.), user account settings (e.g., preferences, activity, etc.), action-sequences (e.g., clickstreams, etc.), data related to the specific event, such as the changed notification preferences, and/or the like. In some implementations, the event data describes a series of actions (e.g., clicks) the user made leading up to a purchase as well as information regarding the purchase itself (e.g., price, item purchased, etc.). In various implementations, SDK 20 bundles event data from a number of events. For example, SDK 20 may bundle event data from 10 events to reduce the power consumption associated with transmitting each event individually.

[0048] At step 404, SDK 20 and/or application 12 may transmit the bundled event data to first data processing system 200. In some implementations, the event data transmitted to first data processing system 200 includes identifiers. In various implementations, the event data transmitted to first data processing system 200 is encrypted (e.g., asymmetrically encrypted, etc.). In some implementations, SDK 20 may determine whether first data processing system 200 received the bundled event data successfully. In some implementations, if SDK 20 determines first data process system 200 did not receive the bundled event data successfully, then SDK 20 transmits the bundled event data to second data processing system 300. Additionally or alternatively, SDK 20 may transmit the bundled event data to second data processing system 300 in parallel (step 402). In various implementations, SDK 20 removes personal identifiers from the bundled event data before transmitting to second data processing system 300. For example, SDK 20 may remove device identifiers and/or user identifiers before transmitting the bundled event data to second data processing system 300.

[0049] In various implementations, first data processing system 200 identifies supplemental data based on the received bundled event data. For example, first data processing system 200 may identify previous interaction data associated with a user identified

by a user identifier included in the bundled event data. At step 406, first data processing system 200 may transmit the supplemental data to second data processing system 300. In various implementations, first data processing system 200 removes personal identifiers before transmitting the supplemental data to second data processing system 300. For example, first data processing system 200 may remove device identifiers and/or user identifiers before transmitting the supplemental data to second data processing system 300.

[0050] Second data processing system 300 may receive the event data from client device 10 (e.g., via SDK 20, etc.) and/or the supplemental data from first data processing system 200. In various implementations, second data processing system 300 analyzes (e.g., performs various correlations and/or statistical operations on, etc.) the event data and/or the supplemental data to generate results. For example, second data processing system 300 may correlate previous interaction data included in the supplemental data with a transaction included in the event data using an intermediary identifier to determine a conversion rate.

[0051] Referring now to FIG. 3, method 500 of reporting event data is shown, according to an implementation. In brief summary, SDK 20 and/or application 12 generates event data in response to interactions associated with a user (e.g., in-application interactions, etc.) and SDK 20 reports the event data to an endpoint. In various implementations, SDK 20 performs method 500. At step 502, SDK 20 determines a connection to first data processing system 200 is unavailable. For example, SDK 20 may ping first data processing system 200 to determine if it is online. In some implementations, SDK 20 determines whether first data processing system 200 confirms receipt of a data package (e.g., previous event data, etc.). In some implementations, step 502 is optional.

[0052] At step 504, SDK 20 detects an event. SDK 20 may include various reusable functions that are embedded by developers in application 12. When application 12 performs a particular operation, the reusable functions of SDK 20 may be invoked, thereby causing SDK 20 to perform various actions. For example, a reusable function of SDK 20 may be embedded in a checkout process of application 12 and may cause SDK 20 to report event data (e.g., data associated with the transaction, etc.) when the checkout is completed. Additionally or alternatively, SDK 20 may detect an event by listening for various indications from

application 12. For example, application 12 may generate a flag when an application event occurs and SDK 20 may detect the flag to identify the occurrence of an event.

[0053] At step 506, SDK 20 may collect event data from application 12 and/or client device 10. For example, SDK 20 may collect data related to a transaction associated with the event. At step 508, SDK 20 bundles the event data. Step 508 may include combining event data from a number of events into a single data package. For example, event data from one-hundred events may be combined into bundled event data. In some implementations, SDK 20 bundles event data from events occurring over a specified time period. For example, SDK 20 may bundle events from each day (e.g., 24-hour period, etc.). It should be understood that various bundling techniques are possible and SDK 20 is not limited to the specific techniques expressly enumerated herein. In various implementations, SDK 20 labels the event data. For example, SDK 20 may label the event data with an event type. The event type may correspond to a source of the event data. For example, SDK 20 may determine first event data corresponding to an in-application transaction is of a "transaction" type and second event data corresponding to visiting a web-page is of a "content interaction" type. In various implementations, SDK 20 generates different event data based on the event type. For example, SDK 20 may generate event data including an item price and quantity for a "transaction" event and may generate event data including a content identifier for "content interaction" type data. In should be understood that many event types are possible and are not limited to the types expressly contemplated herein.

[0054] At step 510, SDK 20 may prepare and encrypt event data. In various implementations, step 510 includes removing identifiers. For example, SDK 20 may remove personal identifiers (e.g., device identifiers, user identifiers, etc.) from bundled event data. Additionally or alternatively, SDK 20 may encrypt the bundled event data. For example, SDK 20 may asymmetrically encrypt the bundled event data using a public key received from first data processing system 200. In some implementations, SDK 20 removes identifiers from bundled event data that is transmitted to second data processing system 300. In some implementations, SDK 20 encrypts bundled event data that is transmitted to first data processing system 200. In some implementations, step 510 includes generating a signature of

the bundled event data. For example, SDK 20 may generate a signature by hashing the bundled event data. In some implementations, step 510 is optional.

[0055] At step 512, SDK 20 securely transmits event data to second data processing system 300. In some implementations, SDK 20 transmits the event data to second data processing system 300 in response to determining that first data processing system 200 is unavailable (e.g., based on step 502, etc.). In various implementations, SDK 20 transmits event data to second data processing system 300 that is free of personal identifiers, thereby preserving PII. In some implementations, SDK 20 transmits the bundled event data to second data processing system 300 in response to a threshold. For example, SDK 20 may transmit the bundled event data to second data processing system 300 in response to a threshold number of events being included in the event data bundle. In various implementations, SDK 20 facilitates specifying a destination (e.g., second data processing system 300, etc.). For example, SDK 20 may facilitate receiving a URL and/or an IP address of second data processing system 300 specifying where to transmit event data to.

[0056] At step 514, SDK 20 may securely transmit encrypted event data to first data processing system 300. In various implementations, SDK 20 transmits encrypted event data including one or more identifiers to first data processing system 200. In some implementations, step 514 is optional.

[0057] Referring now to FIG. 4, method 600 of identifying supplemental data is shown, according to an implementation. In some implementations, first data processing system 200 performs method 600. In some implementations, first data processing system 200 performs method 600 in response to receiving event data from client device 10 and/or SDK 20. At step 602, first data processing system 200 receives event data from client device 10. In various implementations, first data processing system 200 receives event data including identifiers. For example, first data processing system 200 may receive event data including device identifiers. Additionally or alternatively, first data processing system 200 may receive encrypted event data.

[0058] At step 602, first data processing system 200 may decrypt the event data. In various implementations, step 602 includes asymmetrically decrypting the event data using a

private key held by first data processing system 200 to produce unencrypted event data. At step 606, first data processing system 200 may select supplemental data from stored data using the event data. In various implementations, step 606 includes identifying supplemental data using an identifier included in the event data. In various implementations, first data processing system 200 searches database 210. Database 210 may include data associated with previous content interactions. For example, first data processing system 200 may identify previous content interactions associated with a user. As an additional example, first data processing system 200 may identify a video that a user previously watched.

[0059] At step 608, first data processing system 200 may prepare the supplemental data. For example, first data processing system 200 may remove personal identifiers associated with the supplemental data. In some implementations, first data processing system 200 removes IP addresses, user identifiers, and/or device identifiers associated with the supplemental data. Additionally or alternatively, first data processing system 200 may perform various obliviating functions on the supplemental data. For example, first data processing system 200 may fragment the supplemental data data, add noise, or perform other functions to anonymize data. At step 610, first data processing system 200 may securely transmit the supplemental data to second data processing system 300. In some implementations, first data processing system 200 transmits the supplemental data to second data processing system 300 in response to a signal. For example, second data processing system 300 may send a request for supplemental data to first data processing system 200.

[0060] Referring now to FIG. 5, method 700 of analyzing event data and/or supplemental data is shown, according to an implementation. In various implementations, second data processing system 300 performs method 700. In some implementations, second data processing system 300 performs method 700 in response to receiving data from client device 10. Additionally or alternatively, second data processing system 300 may perform method 700 in response to receiving a request for results from an external system (e.g., first data processing system 200, etc.).

[0061] At step 702, second data processing system 300 may receive event data from client device 10 (e.g., via SDK 20, etc.). In various implementations, the event data is bundled event data. In various implementations, the event data does not include personal identifiers.

At step 704, second data processing system 300 may validate the received event data. In various implementations, the event data includes a signature. Second data processing system 300 may validate the event data using the signature. For example, second data processing system 300 may generate a hash of the received event data and compare the generated hash to a signature included in the event data. In some implementations, step 704 is optional.

[0062] At step 706, second data processing system 300 may receive supplemental data from first data processing system 200. In some implementations, second data processing system 300 receives supplemental data from other sources. In various implementations, the supplemental data includes interaction data associated with the event data. For example, the supplemental data may include a content interaction sharing an intermediary identifier with an event of the event data. As another example, the supplemental data may include a content item identifier that can be correlated to a transaction included in the event data. In some implementations, step 706 is optional. For example, in some implementations, first data processing system 200 does not transmit supplemental data to second data processing system 300.

[0063] At step 708, second data processing system 300 may correlate the event data to the supplemental data. For example, second data processing system 300 may match an identifier of a content item with an intermediary identifier of the event data to determine that the user has interacted with the content item. As an additional example, second data processing system 300 may determine a URL of a content item matches a website URL included in the event data. In some implementations, step 708 is optional. For example, second data processing system 300 may not receive supplemental data corresponding to every event within the event data.

[0064] At step 710, second data processing system 300 may analyze data to generate results. In various implementations, second data processing system 300 analyzes event data. Additionally or alternatively, second data processing system 300 may analyze supplemental data. In various implementations, second data processing system 300 performs various analysis operations to produce the results. In various implementations, the results include a count of the number of interactions and/or event data sharing a first characteristic. Additionally or alternatively, the results may include a sum of interactions (e.g., transactions,

clicks, phone calls, etc.), a sum of the value associated with each interaction (e.g., a dollar amount), and/or metadata. In various implementations, the results are aggregate information. For example, the results may include statistical information associated with event data from a number of events. In some implementations, the results are saved in database 310. In various implementations, the results include data describing usage of application 12 in real time. For example, the results may include a count of a number of users currently viewing a specific content item. Additionally or alternatively, second data processing system 300 may identify a previous interaction of the user that was not performed using application 12 (e.g., cross-platform attribution, etc.). For example, second data processing system 300 may correlate a content interaction included in supplemental data to event data to determine a user previously interacted with the content on a different device than client device 10.

[0065] At step 712, second data processing system 300 may securely transmit the results to first data processing system 200. Additionally or alternatively, second data processing system 300 may securely transmit the results to a different destination. In some implementations, step 712 is optional.

[0066] FIG. 6 illustrates a depiction of a computing system 800 that can be used, for example, to implement any of the illustrative systems (e.g., system 110, etc.) described in the present disclosure. The computing system 800 includes a bus 805 or other communication component for communicating information and a processor 810 coupled to the bus 805 for processing information. The computing system 800 also includes main memory 815, such as a random access memory ("RAM") or other dynamic storage device, coupled to the bus 805 for storing information, and instructions to be executed by the processor 810. Main memory 815 can also be used for storing position information, temporary variables, or other intermediate information during execution of instructions by the processor 810. The computing system 800 may further include a read only memory ("ROM") 820 or other static storage device coupled to the bus 805 for storing static information and instructions for the processor 810. A storage device 825, such as a solid state device, magnetic disk or optical disk, is coupled to the bus 805 for persistently storing information and instructions.

[0067] The computing system 800 may be coupled via the bus 805 to a display 835, such as a liquid crystal display, or active matrix display, for displaying information to a user. An

input device 830, such as a keyboard including alphanumeric and other keys, may be coupled to the bus 805 for communicating information, and command selections to the processor 810. In another implementation, the input device 830 has a touch screen display 835. The input device 830 can include a cursor control, such as a mouse, a trackball, or cursor direction keys, for communicating direction information and command selections to the processor 810 and for controlling cursor movement on the display 835.

[0068] In some implementations, the computing system 800 may include a communications adapter 840, such as a networking adapter. Communications adapter 840 may be coupled to bus 805 and may be configured to enable communications with a computing or communications network 845 and/or other computing systems. In various illustrative implementations, any type of networking configuration may be achieved using communications adapter 840, such as wired (e.g., via Ethernet), wireless (e.g., via WiFi, Bluetooth, etc.), pre-configured, ad-hoc, LAN, WAN, etc.

[0069] According to various implementations, the processes that effectuate illustrative implementations that are described herein can be achieved by the computing system 800 in response to the processor 810 executing an arrangement of instructions contained in main memory 815. Such instructions can be read into main memory 815 from another computer-readable medium, such as the storage device 825. Execution of the arrangement of instructions contained in main memory 815 causes the computing system 800 to perform the illustrative processes described herein. One or more processors in a multi-processing arrangement may also be employed to execute the instructions contained in main memory 815. In alternative implementations, hard-wired circuitry may be used in place of or in combination with software instructions to implement illustrative implementations. Thus, implementations are not limited to any specific combination of hardware circuitry and software.

[0070] Although an example processing system has been described in FIG. 6, implementations of the subject matter and the functional operations described in this specification can be carried out using other types of digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them.

[0071] Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, or features described herein may enable collection of user information (e.g., information about a user's social network, social actions, or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user. In situations in which the systems described herein collect personal information about users or applications installed on a user device, or make use of personal information, the users are provided with an opportunity to control whether programs or features collect user information (e.g., information about a user's social network, social actions, or activities, profession, a user's preferences, or a user's current location). In addition or in the alternative, certain data may be treated in one or more ways before it is stored or used, so that personal information is removed.

[0072] System and methods of the present disclosure offer many benefits over existing systems. Typically, data from client device applications is collected by an aggregator such as a server associated with the operation of the application before being forwarded to an analysis system for analysis. However, the connection between the client device and the aggregator is not always reliable. For example, the client device may be unable to transmit data to the aggregator and/or the aggregator may become unavailable (e.g., go offline for maintenance, etc.). The novel SDK and computer architectures described herein facilitate a redundant path for reporting event data. Specifically, the SDK of the present disclosure facilitates transmitting event data to a separate endpoint in response to a failure in transmitting data to the aggregator. Therefore, the SDK of the present disclosure improves existing systems by facilitating redundant reporting of event data and thereby reducing lost data.

[0073] Furthermore, conventional systems introduce a time delay between when client devices generate event data and when the analysis system receives the event data. As discussed earlier, typically client devices transmit event data to an aggregator that forwards the data to the analysis system. However, there may be a time delay between when the aggregator receives the data and when the analysis system receives the data. This time delay may reduce the responsiveness of an application to changes. For example, if an application has a software bug that is causing transactions to prematurely fail, a conventional system may take some time to detect the software bug. However, the novel SDK and computer architectures described herein facilitate direct transmission of event data to the analysis system. Specifically, the SDK of the present disclosure facilitates increased responsiveness of application events (e.g., crashes, bugs, etc.) by reducing the time delay between when data is generated by client devices and when an analysis system receives the data.

[0074] Moreover, generally systems that correlate data from different entities match the data using identifiers. For example, a system may match a user interaction with a user transaction using a user identifier. The novel SDK and computer architectures described herein facilitate correlation of data from different entities without revealing PII. Specifically, the SDK and computer architectures of the present disclosure facilitate correlating data from different entities using an intermediate identifier that does not reveal PII. Furthermore, content providers (e.g., application developers, etc.) may improve analytics results without having to transmit sensitive data to the analysis system. For example, the content provider may identify supplemental data from their own databases and remove sensitive identifiers (e.g., personal identifiers, etc.) before transmitting the data to the analysis system.

[0075] Implementations of the subject matter and the operations described in this specification can be carried out using digital electronic circuitry, or in computer software embodied on a tangible medium, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions, encoded on one or more computer storage medium for execution by, or to control the operation of, data processing apparatus. Alternatively or in addition, the program instructions

can be encoded on an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. A computer-readable storage medium can be, or be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium can be a source or destination of computer program instructions encoded in an artificially-generated propagated signal. The computer storage medium can also be, or be included in, one or more separate components or media (e.g., multiple CDs, disks, or other storage devices). The computer storage medium may be tangible and/or may be non-transitory.

[0076] The operations described in this specification can be implemented as operations performed by a data processing apparatus on data stored on one or more computer-readable storage devices or received from other sources.

[0077] The term "data processing apparatus" or "computing device" encompasses all kinds of apparatus, devices, and machines for processing data, including by way of example, a programmable processor, a computer, a system on a chip, or multiple ones, or combinations of the foregoing. The apparatus can include special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). The apparatus can also include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, a cross-platform runtime environment, a virtual machine, or a combination of one or more of them. The apparatus and execution environment can realize various different computing model infrastructures, such as web services, distributed computing and grid computing infrastructures.

[0078] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, declarative or procedural languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, object, or

other unit suitable for use in a computing environment. A computer program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0079] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform actions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). Circuit as utilized herein, may be implemented using hardware circuitry (e.g., FPGAs, ASICs, etc.), software (instructions stored on one or more computer readable storage media and executable by one or more processors), or any combination thereof.

[0080] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for performing actions in accordance with instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant ("PDA"), a mobile audio or video player, a game console, a Global Positioning System ("GPS") receiver, or a portable storage device (e.g., a universal serial bus ("USB") flash drive), to name just a few. Devices suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example,

semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0081] To provide for interaction with a user, implementations of the subject matter described in this specification can be carried out using a computer having a display device, e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's client device in response to requests received from the web browser.

[0082] Implementations of the subject matter described in this specification can be carried out using a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such backend, middleware, or frontend components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), an inter-network (e.g., the Internet), and peer-to-peer networks (e.g., ad hoc peer-to-peer networks).

[0083] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some implementations, a server transmits data (e.g., an HTML page) to a client device (e.g., for

purposes of displaying data to and receiving user input from a user interacting with the client device). Data generated at the client device (e.g., a result of the user interaction) can be received from the client device at the server.

[0084] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any inventions or of what may be claimed, but rather as descriptions of features specific to particular implementations of particular inventions. Certain features that are described in this specification in the context of separate implementations can also be carried out in combination or in a single implementation. Conversely, various features that are described in the context of a single implementation can also be carried out in multiple implementations, separately, or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can, in some cases, be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination. Additionally, features described with respect to particular headings may be utilized with respect to and/or in combination with illustrative implementations described under other headings; headings, where provided, are included solely for the purpose of readability and should not be construed as limiting any features provided with respect to such headings.

[0085] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the implementations described above should not be understood as requiring such separation in all implementations, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products embodied on tangible media.

[0086] Thus, particular implementations of the subject matter have been described. Other implementations are within the scope of the following claims. In some cases, the actions recited in the claims can be performed in a different order and still achieve desirable results.

In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous.

WHAT IS CLAIMED IS:

1. A method comprising:

tagging a feature of a mobile application as an event generator;

in response to a user interacting with the feature of the mobile application on a mobile device, generating an event having an event type;

requesting, by an interaction measurement software development kit (SDK) from the mobile application, interaction data associated with the user interaction with the mobile application based on the event type; and

securely transmitting, by the interaction measurement SDK, the interaction data to a first computing device indicated by the interaction measurement SDK.

2. The method of Claim 1, wherein the interaction measurement SDK bundles interaction data from a number of events and transmits the bundled interaction data in response to a threshold.

3. The method of Claim 2, wherein the threshold is a threshold period of time or a threshold number of events.

4. The method of any preceding Claim, wherein the interaction SDK generates a signature of the interaction data using a hashing algorithm, and wherein the first computing device validates an authenticity of the interaction data using the signature.

5. The method of any preceding Claim, further comprising securely transmitting the interaction data to a second computing device associated with the mobile application, and wherein the interaction SDK removes personal identifiers from the interaction data prior to securely transmitting the interaction data to the first computing device.

6. The method of Claim 5, wherein the interaction data includes an intermediary identifier, and wherein method further comprises:

identifying, by the second computing device, supplemental data using the intermediary identifier;

transmitting, by the second computing device to the first computing device, the supplemental data; and

correlating, by the first computing device, the interaction data with the supplemental data.

7. The method of Claim 6, wherein the first computing device does not have access to personally identifying information.

8. The method of any preceding Claim, further comprising analyzing, by the first computing device, at least the interaction data to generate data describing usage of the mobile application in real time.

9. The method of Claim 8, wherein analyzing at least the interaction data includes identifying a previous interaction of the user that was not performed using the mobile application.

10. A system for correlating data from different entities, comprising:

a client device including a processing circuit having one or more processors and memory, the memory having instructions stored thereon that, when executed by the one or more processors, cause the processing circuit to:

tag a feature in a mobile application as an event generator;

in response to a user interacting with the feature of the mobile application, generate an event having an event type;

request interaction data associated with the user interaction with the mobile application based on the event type; and

securely transmit the interaction data to a first computing device; and

wherein the instructions constitute a software development kit (SDK).

11. The system of Claim 10, wherein the instructions cause the one or more processors to bundle interaction data from a number of events and transmit the bundled interaction data in response to a threshold.

12. The system of Claim 11, wherein the threshold is a threshold period of time or a threshold number of events.

13. The system of any of Claims 10 to 12, further comprising the first computing device, wherein the instructions cause the one or more processors to generate a signature of the interaction data using a hashing algorithm, and wherein the first computing device validates an authenticity of the interaction data using the signature.

14. The system of any of Claims 10 to 13, wherein the instructions further cause the one or more processors to securely transmit the interaction data to a second computing device and remove personal identifiers from the interaction data prior to securely transmitting the interaction data to the first computing device.

15. The system of Claim 14, further comprising the first computing device and the second computing device, wherein the interaction data includes an intermediary identifier, wherein the second computing device identifies supplemental data using the intermediary identifier;

wherein the second computing device transmits the supplemental data to the first computing device; and

wherein the first computing device correlates the interaction data with the supplemental data.

16. The system of Claim 15, wherein the first computing device does not have access to personally identifying information.

17. The system of any of Claims 10 to 16, further comprising the first computing device, wherein the first computing device analyzes at least the interaction data to generate data describing usage of the mobile application in real time.

18. The system of Claim 17, wherein analyzing at least the interaction data includes identifying a previous interaction of the user that was not performed using the mobile application.

19. A computer-readable medium having instructions constituting a software development kit (SDK) for a mobile device stored thereon that, when executed by one or more processors, cause the one or more processors to:

tag a feature of a mobile application as an event generator;

in response to a user interacting with the feature of the mobile application on a mobile device, generate an event having an event type;

request interaction data from the mobile device associated with the user interaction with the mobile application based on the event type; and

securely transmit the interaction data to a first computing device.

20. The non-transitory computer-readable medium of Claim 19, wherein the SDK bundles interaction data from a number of events and transmits the bundled interaction data in response to a threshold, wherein the threshold is a threshold period of time or a threshold number of events, wherein the interaction SDK generates a signature of the interaction data using a hashing algorithm.

21. The non-transitory computer-readable medium of Claim 19, wherein the instructions further cause the one or more processors to securely transmit the interaction data to a second computing device and remove personal identifiers from the interaction data prior to securely transmitting the interaction data to the first computing device.

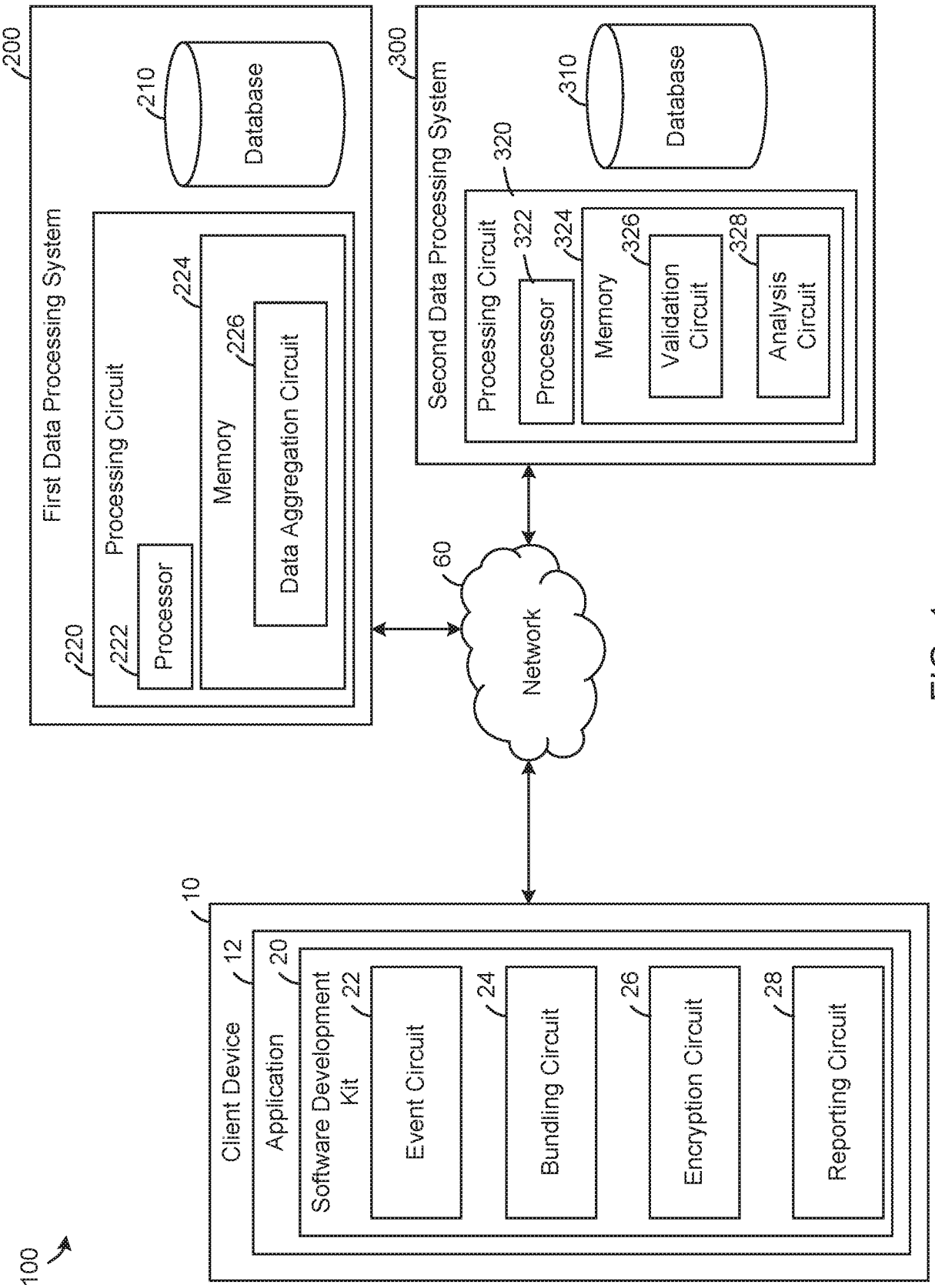
100
10
Client Device
12
Application
20
Software Development Kit
22
Event Circuit
24
Bundling Circuit
26
Encryption Circuit
28
Reporting Circuit
60
Network
200
First Data Processing System
210
Database
220
Processing Circuit
222
Processor
224
Memory
226
Data Aggregation Circuit
300
Second Data Processing System
310
Database
320
Processing Circuit
322
Processor
324
Memory
326
Validation Circuit
328
Analysis Circuit

FIG. 1

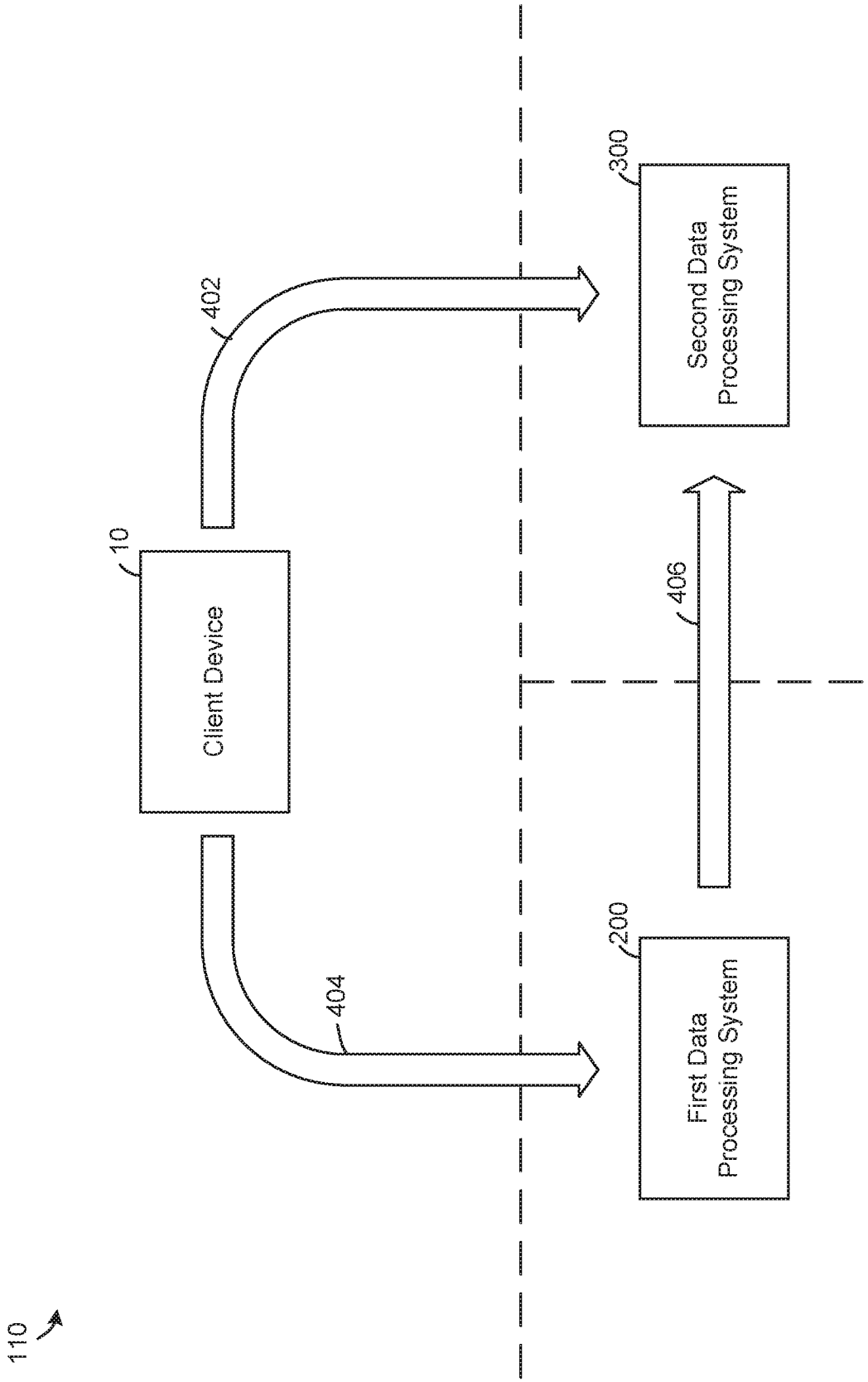
110
10
Client Device
402
404
406
200
First Data Processing System
300
Second Data Processing System

FIG. 2

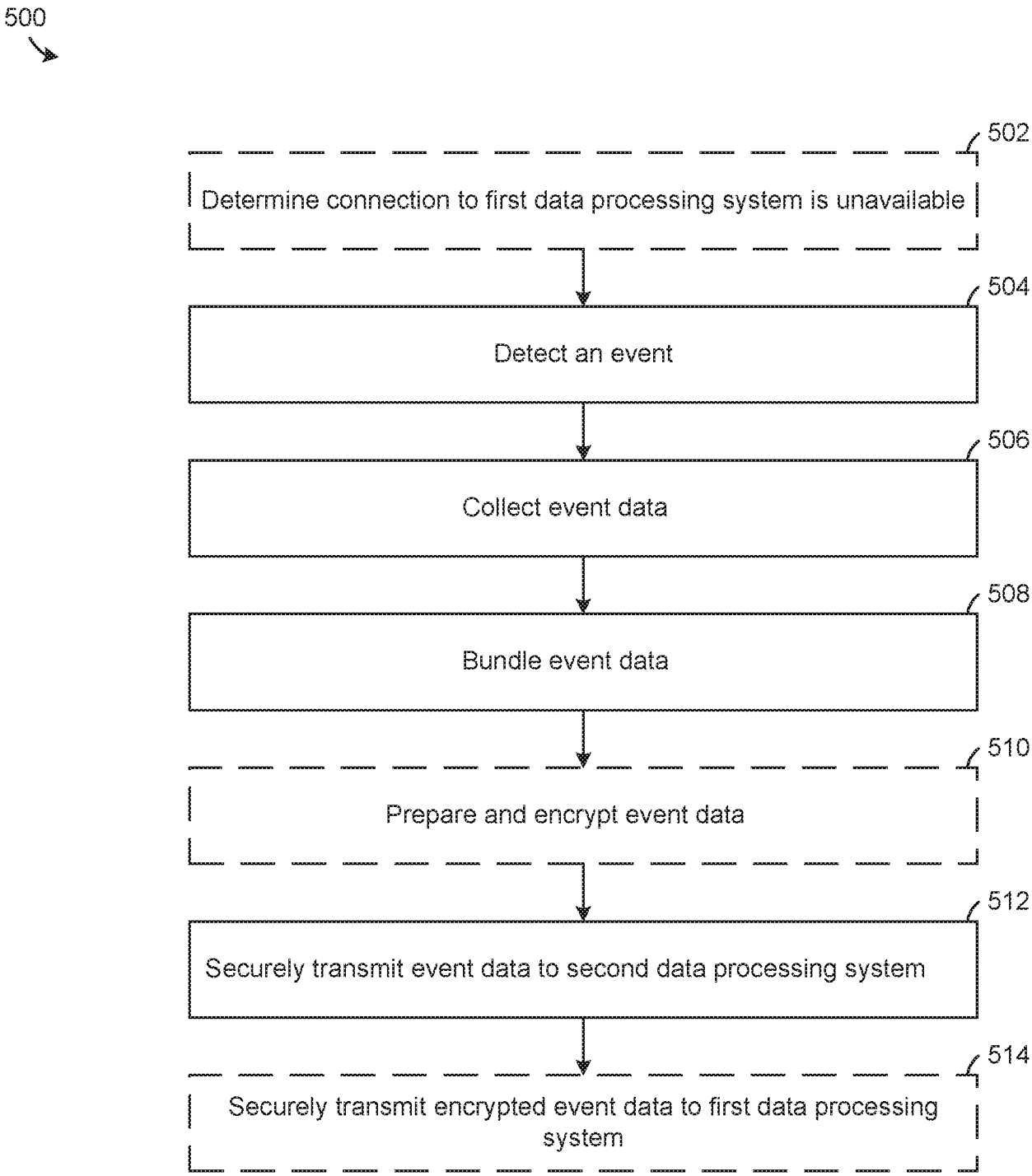
500
502
Determine connection to first data processing system is unavailable
504
Detect an event
506
Collect event data
508
Bundle event data
510
Prepare and encrypt event data
512
Securely transmit event data to second data processing system
514
Securely transmit encrypted event data to first data processing system

FIG. 3

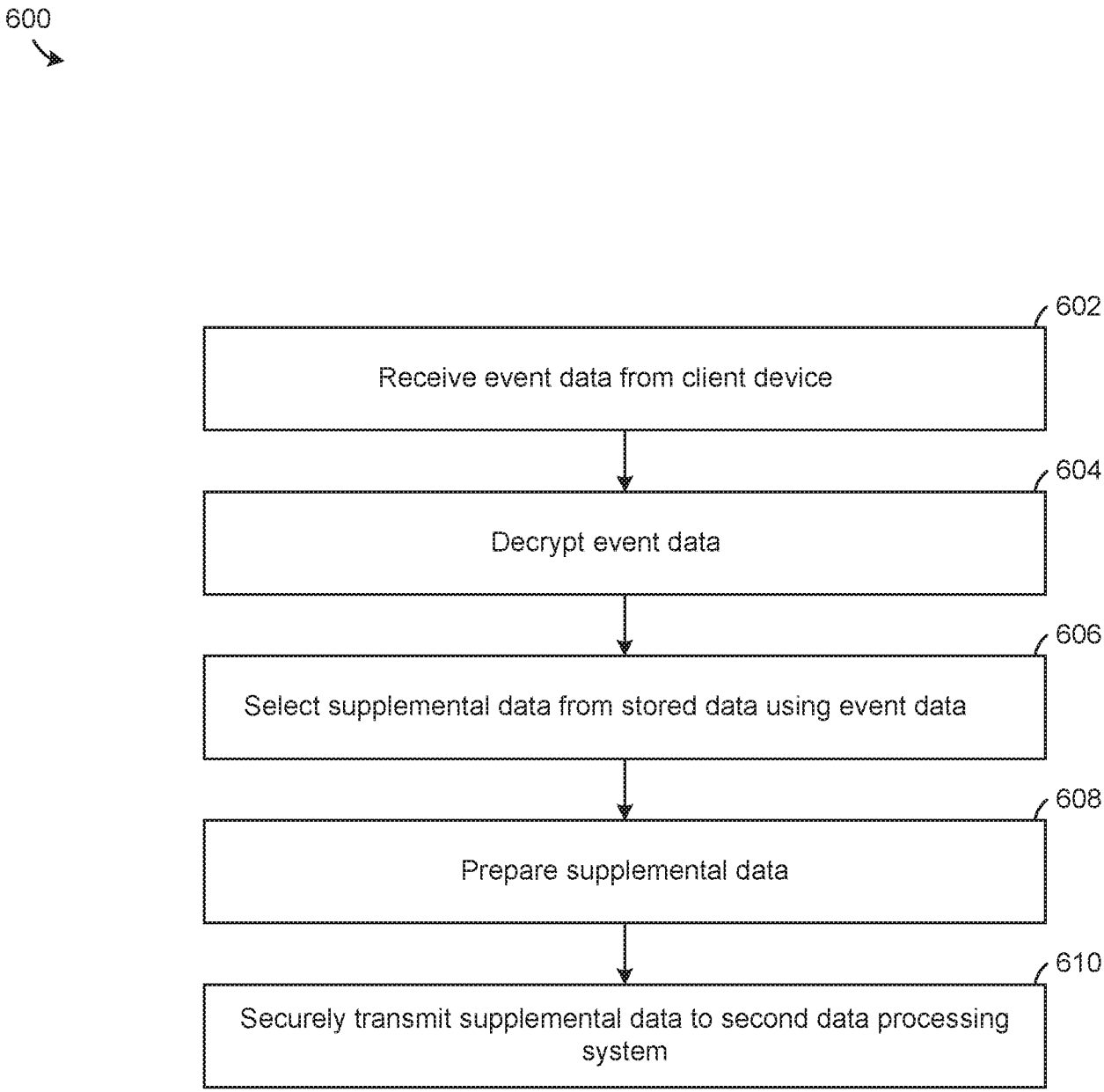
600
602
Receive event data from client device
604
Decrypt event data
606
Select supplemental data from stored data using event data
608
Prepare supplemental data
610
Securely transmit supplemental data to second data processing system

FIG. 4

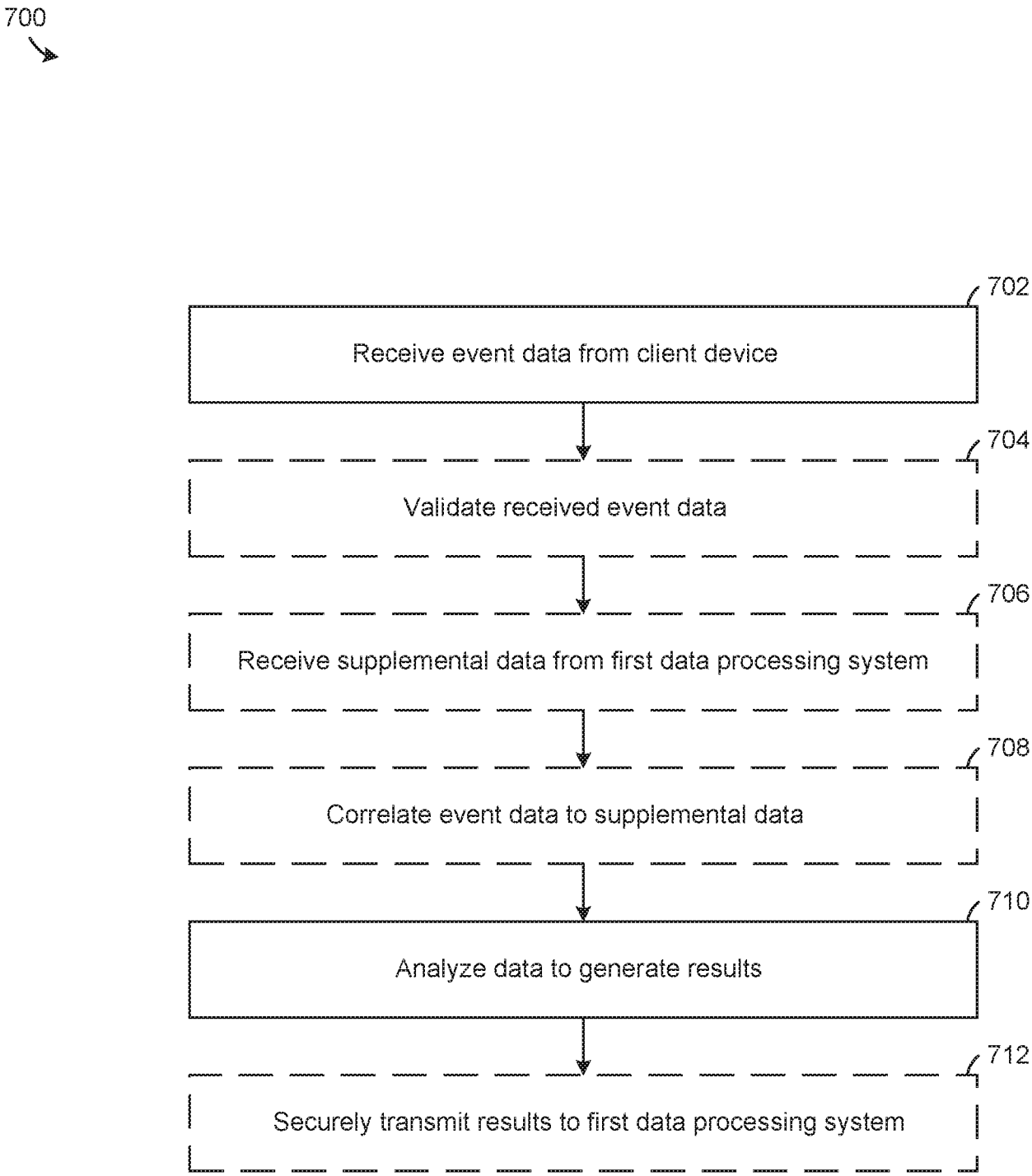
700
702
Receive event data from client device
704
Validate received event data
706
Receive supplemental data from first data processing system
708
Correlate event data to supplemental data
710
Analyze data to generate results
712
Securely transmit results to first data processing system

FIG. 5

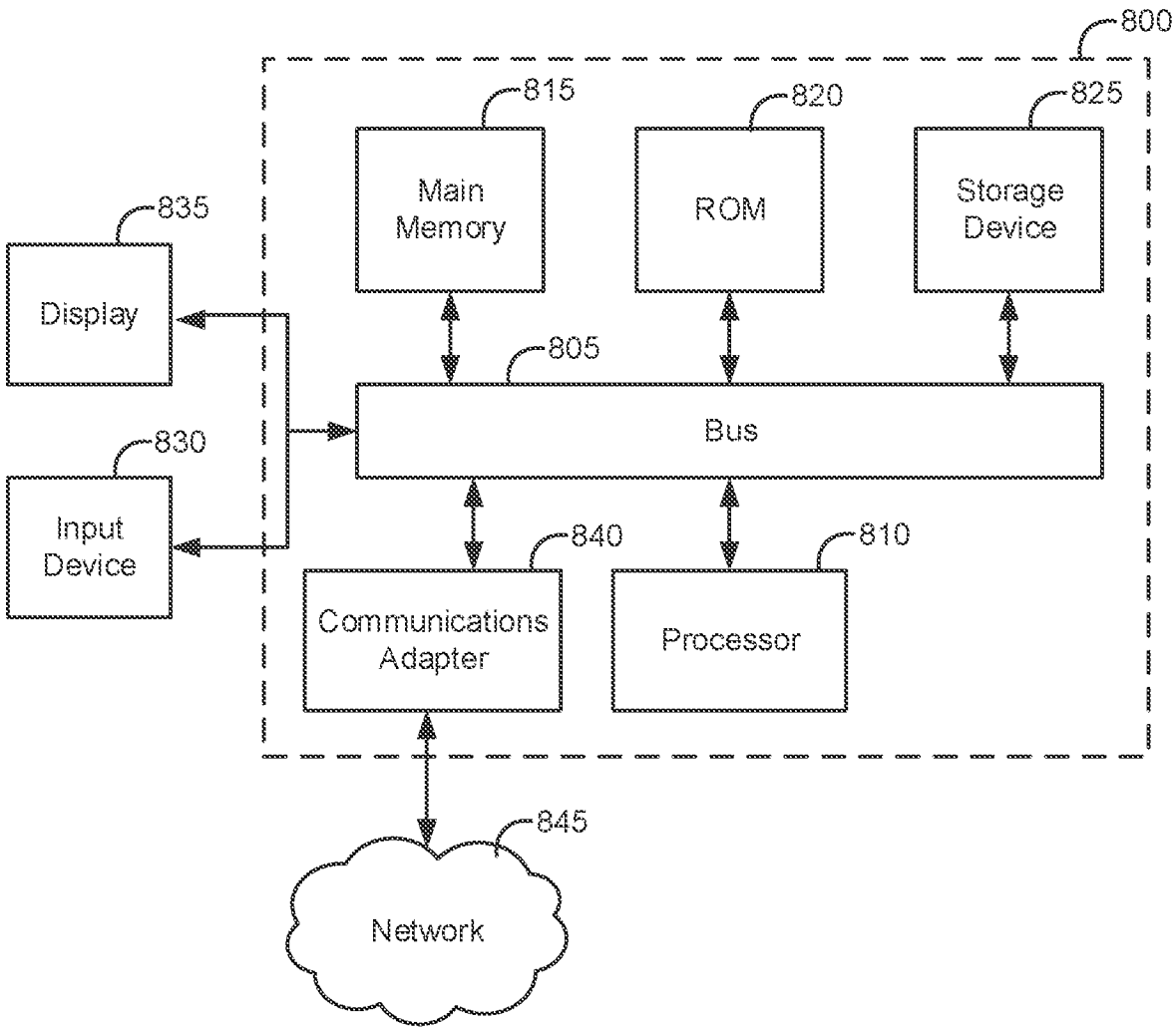
800
815
820
825
Main Memory
ROM
Storage Device
835
Display
805
Bus
830
Input Device
840
810
Communications Adapter
Processor
845
Network

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2020/042007

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/55 G06F21/62 G06Q30/02
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F G06Q H04L H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US 2015/088635 A1 (MAYCOTTE HIGINIO O [US] ET AL) 26 March 2015 (2015-03-26) paragraphs [0023] - [0024], [0048] - [0049], [0062] - [0064]; figures 1-2	1,10,19, 21 2-9, 11-18,20
X Y	US 2020/160388 A1 (SABEG MEHDI ERIC ARNAUD [FR] ET AL) 21 May 2020 (2020-05-21) paragraphs [0071] - [0077], [0080], [0083], [0084], [0088], [0102] - [0112], [0114] - [0118]; figure 7	1-21 5-9, 14-18
Y	WO 2016/148840 A1 (QUALCOMM INC [US]) 22 September 2016 (2016-09-22) paragraphs [0034], [0054], [0057], [0088]	2-9, 11-18,20

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 March 2021	12/03/2021

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Veillas, Erik

2

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2020/042007

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2015/213288 A1 (BILODEAU MICHAEL [US] ET AL) 30 July 2015 (2015-07-30) paragraphs [0068] - [0075]	5,6
Y	US 2015/121080 A1 (DAYKA JOHN C [US] ET AL) 30 April 2015 (2015-04-30) paragraphs [0013] - [0015]	2-4

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2020/042007

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015088635 A1	26-03-2015	US 2015088635 A1	26-03-2015
		US 2015235259 A1	20-08-2015
US 2020160388 A1	21-05-2020	US 2020160388 A1	21-05-2020
		WO 2020100118 A1	22-05-2020
WO 2016148840 A1	22-09-2016	CN 107430660 A	01-12-2017
		EP 3271860 A1	24-01-2018
		JP 2018513467 A	24-05-2018
		KR 20170128300 A	22-11-2017
		US 2016277435 A1	22-09-2016
		WO 2016148840 A1	22-09-2016
US 2015213288 A1	30-07-2015	CN 105940410 A	14-09-2016
		EP 3100203 A1	07-12-2016
		KR 20160114077 A	04-10-2016
		US 2015213288 A1	30-07-2015
		US 2017177904 A1	22-06-2017
		WO 2015116478 A1	06-08-2015
US 2015121080 A1	30-04-2015	US 2015121080 A1	30-04-2015
		US 2015121081 A1	30-04-2015