



(12) **DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION**

(13) **A1**

(86) Date de dépôt PCT/PCT Filing Date: 2016/10/14
 (87) Date publication PCT/PCT Publication Date: 2017/04/20
 (85) Entrée phase nationale/National Entry: 2018/04/13
 (86) N° demande PCT/PCT Application No.: US 2016/057213
 (87) N° publication PCT/PCT Publication No.: 2017/066700
 (30) Priorités/Priorities: 2015/10/15 (US62/242,253);
 2016/10/14 (US15/294,341)

(51) Cl.Int./Int.Cl. *G06F 9/44* (2018.01)

(71) Demandeur/Applicant:
POKITDOK, INC., US

(72) Inventeurs/Inventors:
CORBIN, BRIAN, US;
DUNLEVY, TIM, US;
GOSNELL, DENISE K., US;
ALDRIDGE, MATT, US;
THOMAS, DOUG, US;
TANNER, TED, US

(74) Agent: DLA PIPER (CANADA) LLP

(54) Titre : SYSTEME ET PROCEDE POUR LA PERSISTENCE ET LA CORRELATION DYNAMIQUES DE METADONNEES SUR DES TRANSACTIONS D'API

(54) Title: SYSTEM AND METHOD FOR DYNAMIC METADATA PERSISTENCE AND CORRELATION ON API TRANSACTIONS

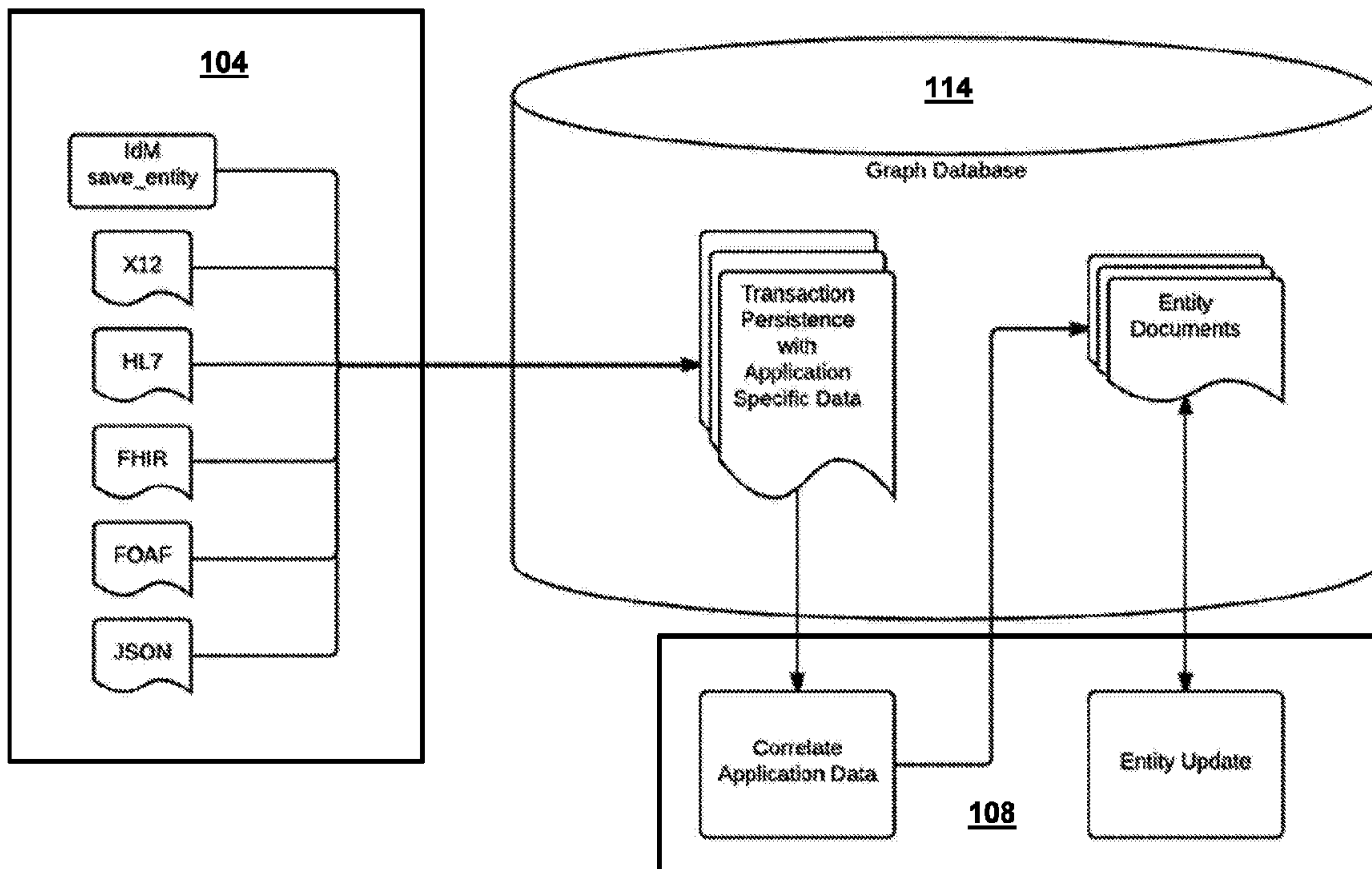


FIGURE 2

(57) Abrégé/Abstract:

A dynamic metadata persistence and correlation system and method are disclosed. The system and method provide a means of tracking and relating transactional metadata from application API calls to internal data models. This system pairs application level flexibility with dynamic correlation management for entity evolution, data retrieval, and analytics.

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

(19) World Intellectual Property
Organization
International Bureau(10) International Publication Number
WO 2017/066700 A1(43) International Publication Date
20 April 2017 (20.04.2017)(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:

PCT/US2016/057213

(22) International Filing Date:

14 October 2016 (14.10.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/242,253 15 October 2015 (15.10.2015) US
15/294,341 14 October 2016 (14.10.2016) US(71) Applicant: **POKITDOK, INC.** [US/US]; 28 E. 3rd Avenue, Suite 300, San Mateo, CA 94401 (US).(72) Inventors: **CORBIN, Brian**; 28 E. 3rd Avenue, Suite 300, San Mateo, CA 94401 (US). **DUNLEVY, Tim**; 28 E. 3rd Avenue, Suite 300, San Mateo, CA 94401 (US). **GOSNELL, Denise, K.**; 28 E. 3rd Avenue, Suite 300, San Mateo, CA 94401 (US). **ALDRIDGE, Matt**; 28 E. 3rd Avenue, Suite 300, San Mateo, CA 94401 (US). **THOMAS, Doug**; 28 E. 3rd Avenue, Suite 300, San Mateo, CA 94401 (US). **TANNER, Ted**; 28 E. 3rd Avenue, Suite 300, San Mateo, CA 94401 (US).(74) Agent: **LOHSE, Timothy, W.**; DLA Piper LLP (US), 2000 University Avenue, East Palo Alto, CA 94303 (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, JP, KE, KG, 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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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).

Published:

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

(54) Title: SYSTEM AND METHOD FOR DYNAMIC METADATA PERSISTENCE AND CORRELATION ON API TRANSACTIONS

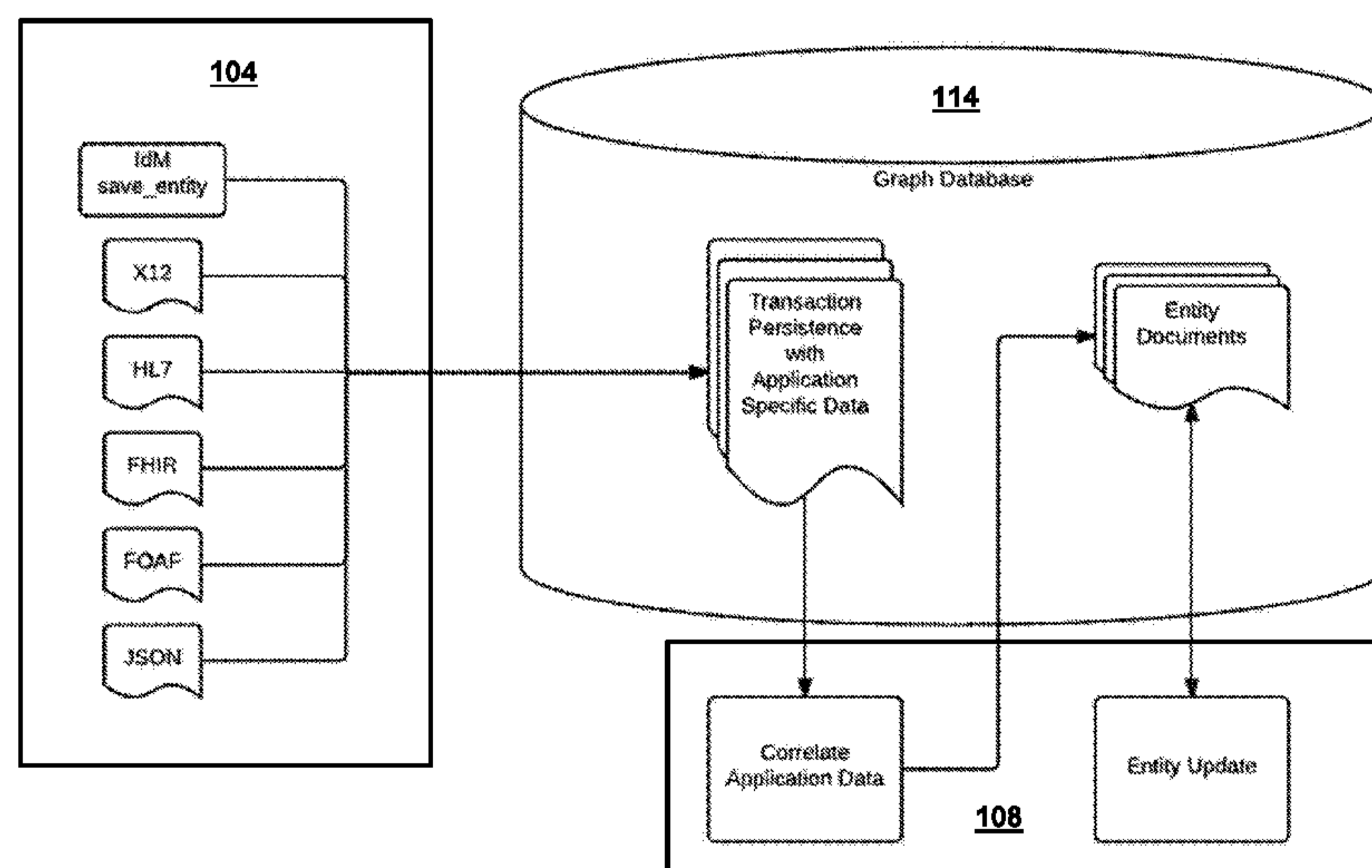


FIGURE 2

(57) Abstract: A dynamic metadata persistence and correlation system and method are disclosed. The system and method provide a means of tracking and relating transactional metadata from application API calls to internal data models. This system pairs application level flexibility with dynamic correlation management for entity evolution, data retrieval, and analytics.

WO 2017/066700 A1

**SYSTEM AND METHOD FOR DYNAMIC METADATA PERSISTENCE AND
CORRELATION ON API TRANSACTIONS**

5

10 **Field**

The disclosure relates generally to tracking and relating transactional metadata from application API calls to internal data models.

Background

15 Application level API calls may require a level of user flexibility in order to fully encapsulate the data of each request. Additionally, each API call contains transaction data which details the specifics of the data exchange. Both the application level API calls and transaction metadata dynamically create data keys for transfer and persistence into a database. It is desirable to provide to provide a mechanism to facilitate the transfer, persistence and correlation of the application data and the API transactional metadata.

20 **Brief Description of the Drawings**

Figure 1 illustrates an example of a healthcare system that may incorporate a data persistence system;

Figure 2 illustrates an example of an implementation of a data persistence system;

25 Figure 3 illustrates an example of a health graph database that may be used by the data persistence system;

Figure 4 details an example of a method for data persistence using the system having the three portions of the architecture shown in Figures 1-2;

Figure 5 illustrates a level to which application level users can provide custom fields for persistence in the graph database;

-2-

Figure 6 illustrates the historical correlation linking process that is part of the data persistence method shown in Figure 4; and

Figures 7A and 7B illustrate an example of the dynamic provenance tracking that occurs during the correlation process.

5 Detailed Description of One or More Embodiments

The disclosure is particularly applicable to a data persistence system and method that is used with the PokitDok healthcare system and method and its database shown in Figure 1 and it is in this context that the disclosure will be described. It will be appreciated, however, that the system and method has greater utility since it may be used as a standalone system and it may be used with or integrated into other data systems in which it is desirable to facilitate the transfer, persistence and correlation of the application data and the API transactional metadata.

Application level API calls through the PokitDok platform require a level of user flexibility in order to fully encapsulate the data of each request. Additionally, each API call contains transaction data which details the specifics of the data exchange. Both the application level API calls and transaction metadata dynamically create data keys for transfer and persistence into a database. The system and method herein details the transfer, persistence and correlation of a dynamic set of application specific data in addition to API transaction metadata.

The dynamic metadata persistence and correlation system described below provides a means of tracking and relating transactional metadata from application API calls to internal data models. This system pairs application level flexibility with dynamic correlation management for entity evolution, data retrieval, and analytics.

Figure 1 illustrates an example of a healthcare system 100 that may incorporate a data persistence system that may be implemented in a backend component 108 or a health graph store 114 or in a combination of the backend component 108 and the health graph store 114. The healthcare marketplace system 100 may have one or more computing devices 102 that connect over a communication path 106 to a backend system 108. Each computing device 102, such as computing devices 102a, 102b, ..., 102n as shown in Figure 1, may be a processor based device with memory, persistent storage, wired or wireless communication circuits and a display that allows each computing device to connect to and couple over the communication path 106 to a backend system 108. For example, each computing device may be a smartphone device, such as

-3-

an Apple Computer product, Android OS based product, etc., a tablet computer, a personal computer, a terminal device, a laptop computer and the like. In one embodiment shown in Figure 12, each computing device 102 may store an application 104 in memory and then execute that application using the processor of the computing device to interface with the backend system.

5 For example, the application may be a typical browser application or may be a mobile application. The communication path 106 may be a wired or wireless communication path that uses a secure protocol or an unsecure protocol. For example, the communication path 106 may be the Internet, Ethernet, a wireless data network, a cellular digital data network, a WiFi network and the like.

10 The backend system 108 may also have an API processor component 110 and a health marketplace engine 112 that may be coupled together. Each of these components of the backend system may be implemented using one or more computing resources, such as one or more server computers, one or more cloud computing resources and the like. In one embodiment, the API processor 110 and health marketplace engine 112 may each be implemented in software in which
15 each has a plurality of lines of computer code that are executed by a processor of the one or more computing resources of the backend system. In other embodiments, each of the API processor 110 and health marketplace engine 112 may be implemented in hardware such as a programmed logic device, a programmed processor or microcontroller and the like. The backend system 108 may be coupled to a health graph store 114 that stores the various data and software modules that
20 make up the healthcare system as well as the health graph database that is used to store the health data of the backend component 108. The store 114 may be implemented as a hardware database system, a software database system or any other storage system. In this example implementation, the data persistence system may be incorporated into the backend system 108 or may be coupled to the backend system 108, but located remotely.

25 The API processor 110 may receive application programming interface (API) requests and communications from an application 104 of one of the computing devices 102 and may in part perform the data persistence process described below with reference to Figures 4-5. The health marketplace engine 112 may allow practitioners that have joined the healthcare social community to reach potential clients in ways unimaginable even a few years ago. In addition to
30 giving practitioners a social portal with which to communicate and market themselves with

-4-

consumers, the marketplace gives each healthcare practitioner the ability to offer their services in an environment that is familiar to users of Groupon, Living Social, or other social marketplaces.

In the system 100, each of the entries for a user may be stored in the store 114 in the health graph with a user's globally unique identifier and the identifiers of other users of the healthcare system that are allowed to view the information to provide, among other things, access control for the data in the healthstream of each user. In the system, each entry may default to being private among the users explicitly specified on the data. Users can choose to change the privacy level to 'anonymous' when they want to share information they've learned about a particular health issue with the community without revealing their identity.

Figure 2 illustrates an example of an implementation of a data persistence system that may be implemented in the backend 108 and/or the store 114 depending on the particular implementation of the data persistence system. In the data persistence system, a piece of dynamic and flexible application level data enters the system through an API call from an application 104 being executed by one of the computing devices 102 as shown in Figure 2. An example of the API call and the application level data that is contained in the API call is shown in Figure 5. As shown in Figure 2, the piece of data may be various types of data in various formats, such as save_entity data, X12 health data, HL7 data (described in more detail at www.hl7.org/implement/standards which is incorporated herein by reference), FHIR data (described in more detail at https://en.wikipedia.org/wiki/Fast_Healthcare_Interoperability_Resources which is incorporated herein by reference), FOAF data (described in more detail at www.foaf-project.org/ which is incorporated herein by reference) and/or JSON data. The API call contains metadata about the request. For example, the metadata could be represented within the "application_data" key field and corresponding JSON payload shown in Figure 5. The disclosure is not limited to the example in Figure 5 and the system may utilize other types of metadata that the example shown in Figure 5.

The data persistence system persists the user specific application data and API metadata in the graph database 114 as shown. The data persistence system correlates each persisted transaction to a unique identifier which tracks the provenance of the dynamic data fields. For example, Figure 5 contains an example payload to be used in the correlation process. In the

-5-

process, various correlation techniques may be used. For example, the correlation technique may include strong identifier linkage, partial or full document hashing, streaming weighted clustering, ... etc.

The data persistence system utilizes this identifier to dynamically correlate the inflow of application data and metadata as shown in Figure 2 to evolve the entities contained within the database (shown as a graph database for in the example in Figure 2). An example of the evolving correlation process is shown in Figure 6 and described below. Figure 2 details the flow of this procedure from application level to data persistence and concludes with entity correlation. Figure 3 illustrates an example of a health graph database that may be interfaced by the data persistence system and may be modified/updated by the data persistence system.

Figure 4 details an example of a method 400 for data persistence using the system having the three portions of the architecture shown in Figures 1-2. As shown, the method may be performed/implemented with the application 104 executed on a computing system, the backend system 108 and the healthgraph 114. Initially, an application 104 initializes the system through an API request (402 – init()). An example of this API request is described above. In the API request, the user declares application specific data (application level data) in the API call which is processed by the system 108 in Figures 1 and 2. The system 108 processes the API call and transforms it for the intended internal recipient (404- process_request()). When an API call is received at the backend 108, it is transformed into an internal data model and run against an ensemble of correlation metrics that are executed by the streaming API processor 110. The data from the API may then be further combined with known data models from the health marketplace 112 and the input data stream is connected to the evolving entity (as described below) which is stored in the evolving entity persistence database 114.

The system 108 may then appends the user specific data fields and API metadata fields to the internal transfer (406- insert_application_data()). An example of the appending of the data fields to the internal transfer and extraction of the identity models from a data stream and correlation to a known identity is shown in Figures 7A and 7B where Figure 7A shows an example of the known identity and the data fields from the API call and Figure 7B shows the correlation of the data fields in the API call to the known entity.

-6-

In the method, the backend system 108 may persist this data in an internal database (408-
save_transaction()) and a unique identifier that tracks the provenance of the dynamic data fields.
The unique identifier created for each observed entity can be vector of length n which represents
the centroid of the entity's correlated data payloads. As additional payloads are processed and
5 linked, this unique identifier, which can be represented as a centroid vector, is adjusted to
account for the appended data. The chaining and linking process is shown in Figure 6 and the
features observed and correlated in Figures 7A and 7B represent the evolution of the vector space
and the calculable entity centroid for dynamic provenance tracking.

During this process, the fields of the data may be checked against a whitelist for permitted
10 external data fields and persisted according to each key's level of accessibility. The whitelist is
autonomously updated to learn, identify and pass user specific fields. Once the database has
received and persisted the immutable transaction based on the application level data and the
metadata, a process is initialized which will update internal data models via the transactions
unique identifier as shown for example in Figures 7A and 7B. Furthermore, as additional
15 streaming entities are processed, they are linked to the existing entities which are calculated to
have the highest match. This process extracts the application specific data and API metadata and
correlates the data from the single request to the entities which it contains (412 –
correlate_application_data()). The internal data models are updated and evolved according to the
newly observed application data and metadata (414 – update_entities()). In the process, once the
20 data has been saved into the health graph 114, the system 108 may notify the application 104 that
the process has been completed (410-notifying()).

Figure 5 illustrates a level to which application level users can provide custom fields for
persistence in the graph database. In particular, Figure 5 shows that the system allows for
dynamic key value declaration at the API endpoint in the application_data field. The API
25 interface allows for custom key value pairs in this field which, when combined with the metadata
from the transaction, are persisted in the database. For example, the transaction's activity_id is
used to correlate the dynamic key/values pairs across the transaction database that is part of the
health graph database 114. This data is correlated and joined to update the identity of each entity
present within the transaction.

-7-

Figure 6 illustrates the evolving correlation process. As payloads 602 are processed in a streaming manner, they are linked with existing and known strong identifiers 604. Simultaneously, these streaming payloads are measured for correlation according to the observed meta data and optimized correlation algorithm as described above. An example payload with strong identifiers and meta data for correlation is shown in Figure 5.

The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the disclosure to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the disclosure and its practical applications, to thereby enable others skilled in the art to best utilize the disclosure and various embodiments with various modifications as are suited to the particular use contemplated.

The system and method disclosed herein may be implemented via one or more components, systems, servers, appliances, other subcomponents, or distributed between such elements. When implemented as a system, such systems may include an/or involve, inter alia, components such as software modules, general-purpose CPU, RAM, etc. found in general-purpose computers. In implementations where the innovations reside on a server, such a server may include or involve components such as CPU, RAM, etc., such as those found in general-purpose computers.

Additionally, the system and method herein may be achieved via implementations with disparate or entirely different software, hardware and/or firmware components, beyond that set forth above. With regard to such other components (e.g., software, processing components, etc.) and/or computer-readable media associated with or embodying the present inventions, for example, aspects of the innovations herein may be implemented consistent with numerous general purpose or special purpose computing systems or configurations. Various exemplary computing systems, environments, and/or configurations that may be suitable for use with the innovations herein may include, but are not limited to: software or other components within or embodied on personal computers, servers or server computing devices such as routing/connectivity components, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, consumer electronic devices, network PCs, other

-8-

existing computer platforms, distributed computing environments that include one or more of the above systems or devices, etc.

In some instances, aspects of the system and method may be achieved via or performed by logic and/or logic instructions including program modules, executed in association with such components or circuitry, for example. In general, program modules may include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular instructions herein. The inventions may also be practiced in the context of distributed software, computer, or circuit settings where circuitry is connected via communication buses, circuitry or links. In distributed settings, control/instructions may occur from both local and remote computer storage media including memory storage devices.

The software, circuitry and components herein may also include and/or utilize one or more type of computer readable media. Computer readable media can be any available media that is resident on, associable with, or can be accessed by such circuits and/or computing components. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and can accessed by computing component. Communication media may comprise computer readable instructions, data structures, program modules and/or other components. Further, communication media may include wired media such as a wired network or direct-wired connection, however no media of any such type herein includes transitory media. Combinations of the any of the above are also included within the scope of computer readable media.

In the present description, the terms component, module, device, etc. may refer to any type of logical or functional software elements, circuits, blocks and/or processes that may be implemented in a variety of ways. For example, the functions of various circuits and/or blocks can be combined with one another into any other number of modules. Each module may even be

-9-

implemented as a software program stored on a tangible memory (e.g., random access memory, read only memory, CD-ROM memory, hard disk drive, etc.) to be read by a central processing unit to implement the functions of the innovations herein. Or, the modules can comprise programming instructions transmitted to a general purpose computer or to processing/graphics hardware via a transmission carrier wave. Also, the modules can be implemented as hardware logic circuitry implementing the functions encompassed by the innovations herein. Finally, the modules can be implemented using special purpose instructions (SIMD instructions), field programmable logic arrays or any mix thereof which provides the desired level performance and cost.

10 As disclosed herein, features consistent with the disclosure may be implemented via computer-hardware, software and/or firmware. For example, the systems and methods disclosed herein may be embodied in various forms including, for example, a data processor, such as a computer that also includes a database, digital electronic circuitry, firmware, software, or in combinations of them. Further, while some of the disclosed implementations describe specific hardware components, systems and methods consistent with the innovations herein may be implemented with any combination of hardware, software and/or firmware. Moreover, the above-noted features and other aspects and principles of the innovations herein may be implemented in various environments. Such environments and related applications may be specially constructed for performing the various routines, processes and/or operations according to the invention or they may include a general-purpose computer or computing platform selectively activated or reconfigured by code to provide the necessary functionality. The processes disclosed herein are not inherently related to any particular computer, network, architecture, environment, or other apparatus, and may be implemented by a suitable combination of hardware, software, and/or firmware. For example, various general-purpose machines may be used with programs written in accordance with teachings of the invention, or it may be more convenient to construct a specialized apparatus or system to perform the required methods and techniques.

Aspects of the method and system described herein, such as the logic, may also be implemented as functionality programmed into any of a variety of circuitry, including programmable logic devices ("PLDs"), such as field programmable gate arrays ("FPGAs"), programmable array logic ("PAL") devices, electrically programmable logic and memory devices

-10-

and standard cell-based devices, as well as application specific integrated circuits. Some other possibilities for implementing aspects include: memory devices, microcontrollers with memory (such as EEPROM), embedded microprocessors, firmware, software, etc. Furthermore, aspects may be embodied in microprocessors having software-based circuit emulation, discrete logic
5 (sequential and combinatorial), custom devices, fuzzy (neural) logic, quantum devices, and hybrids of any of the above device types. The underlying device technologies may be provided in a variety of component types, e.g., metal-oxide semiconductor field-effect transistor ("MOSFET") technologies like complementary metal-oxide semiconductor ("CMOS"), bipolar technologies like emitter-coupled logic ("ECL"), polymer technologies (e.g., silicon-conjugated
10 polymer and metal-conjugated polymer-metal structures), mixed analog and digital, and so on.

It should also be noted that the various logic and/or functions disclosed herein may be enabled using any number of combinations of hardware, firmware, and/or as data and/or instructions embodied in various machine-readable or computer-readable media, in terms of their behavioral, register transfer, logic component, and/or other characteristics. Computer-readable
15 media in which such formatted data and/or instructions may be embodied include, but are not limited to, non-volatile storage media in various forms (e.g., optical, magnetic or semiconductor storage media) though again does not include transitory media. Unless the context clearly requires otherwise, throughout the description, the words "comprise," "comprising," and the like are to be construed in an inclusive sense as opposed to an exclusive or exhaustive sense; that is to
20 say, in a sense of "including, but not limited to." Words using the singular or plural number also include the plural or singular number respectively. Additionally, the words "herein," "hereunder," "above," "below," and words of similar import refer to this application as a whole and not to any particular portions of this application. When the word "or" is used in reference to a list of two or more items, that word covers all of the following interpretations of the word: any of the items in
25 the list, all of the items in the list and any combination of the items in the list.

Although certain presently preferred implementations of the invention have been specifically described herein, it will be apparent to those skilled in the art to which the invention pertains that variations and modifications of the various implementations shown and described herein may be made without departing from the spirit and scope of the invention. Accordingly, it
30 is intended that the invention be limited only to the extent required by the applicable rules of law.

-11-

While the foregoing has been with reference to a particular embodiment of the disclosure, it will be appreciated by those skilled in the art that changes in this embodiment may be made without departing from the principles and spirit of the disclosure, the scope of which is defined by the appended claims.

-12-

Claims:

1. A system, comprising: 1
a healthcare system that receives an application programming interface call containing application level data for a request, the healthcare system having a health graph database that stores data about one or more entities associated with the healthcare system, the application
5 programming interface call containing metadata about the request;
a data persistence system that persists application level data and metadata into the health graph database and correlates each piece of persisted application level data and metadata to an entity stored in the health graph database; and
an entity in the health graph database being updated in response to the persisted
10 application level data and metadata.
2. The system of claim 1, wherein the healthcare system receives data about a health care transaction through the API call.
3. The system of claim 2, wherein the data about a health care transaction is one of a
15 X12 data format, a HL7 data format, a FHIR data format, a FOAF data format and a JSON data format.
4. The system of claim 1, wherein the data persistence system correlates each piece of persisted application level data to a unique identifier in the health graph database to correlate the persisted application level data.
5. The system of claim 4, wherein the data persistence system tracks a provenance of
20 the application level data using the unique identifier.
6. A data persistence method, comprising:
receiving application level data for a request through an application programming
interface call, the application programming interface call having metadata about the request;
persisting the application level data and the metadata in a health graph database;
25 correlating each piece of persisted application level data and metadata to an entity stored in the health graph database; and
updating the entity in the health graph database based on the persisted application level data and metadata.

-13-

7. The method of claim 6, wherein receiving the application level data further comprises receiving data about a health care transaction.

8. The method of claim 7, wherein the data about a health care transaction is one of a X12 data format, a HL7 data format, a FHIR data format, a FOAF data format and a JSON data
5 format.

9. The method of claim 6, wherein correlating each piece of persisted application level data and metadata further comprises correlating each piece of persisted application level data to a unique identifier in the health graph database to correlate the persisted application level data.

10 10. The method of claim 9, wherein corrected each piece of persisted application level data to a unique identifier further comprises tracking a provenance of the application level data using the unique identifier.

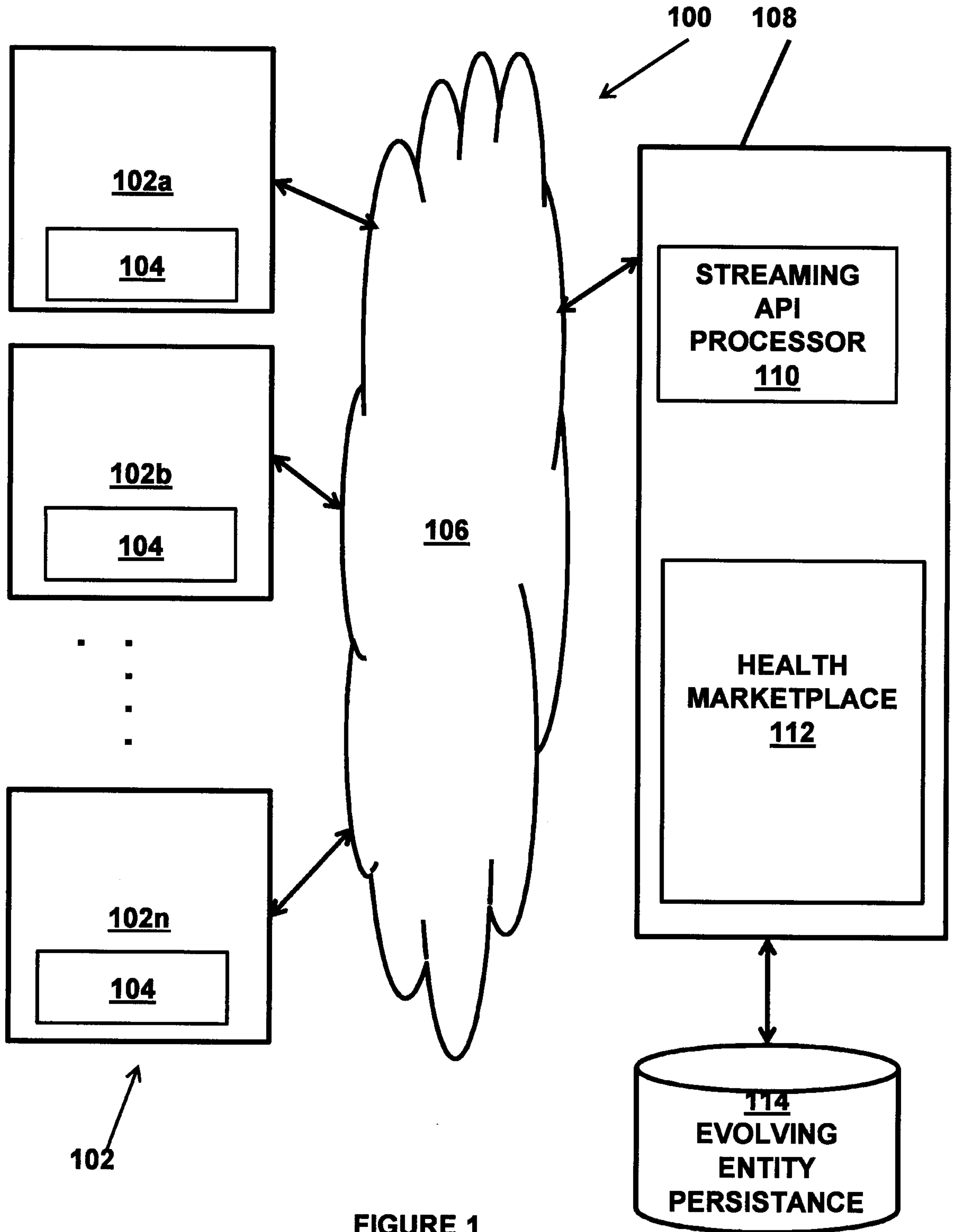


FIGURE 1

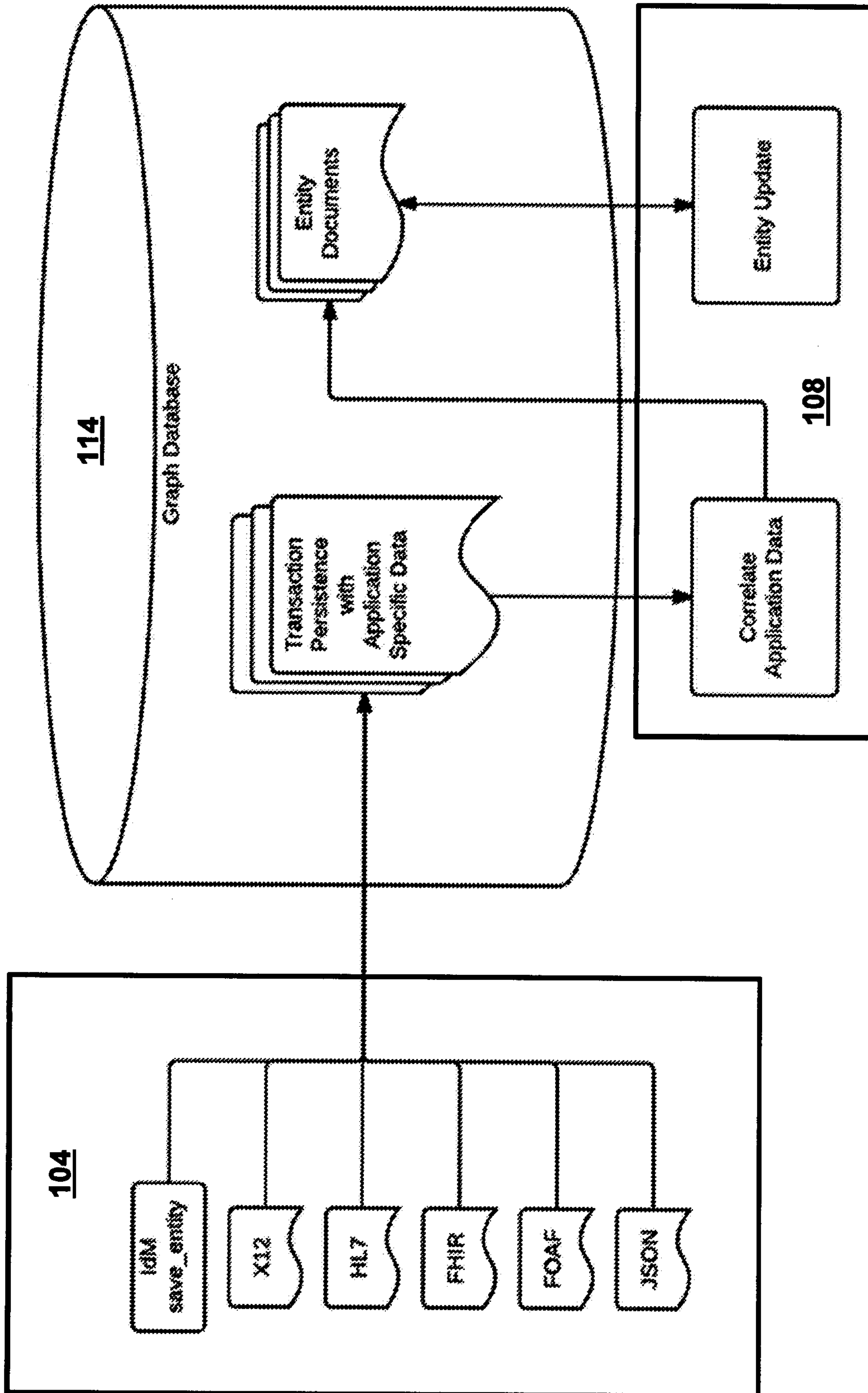


FIGURE 2

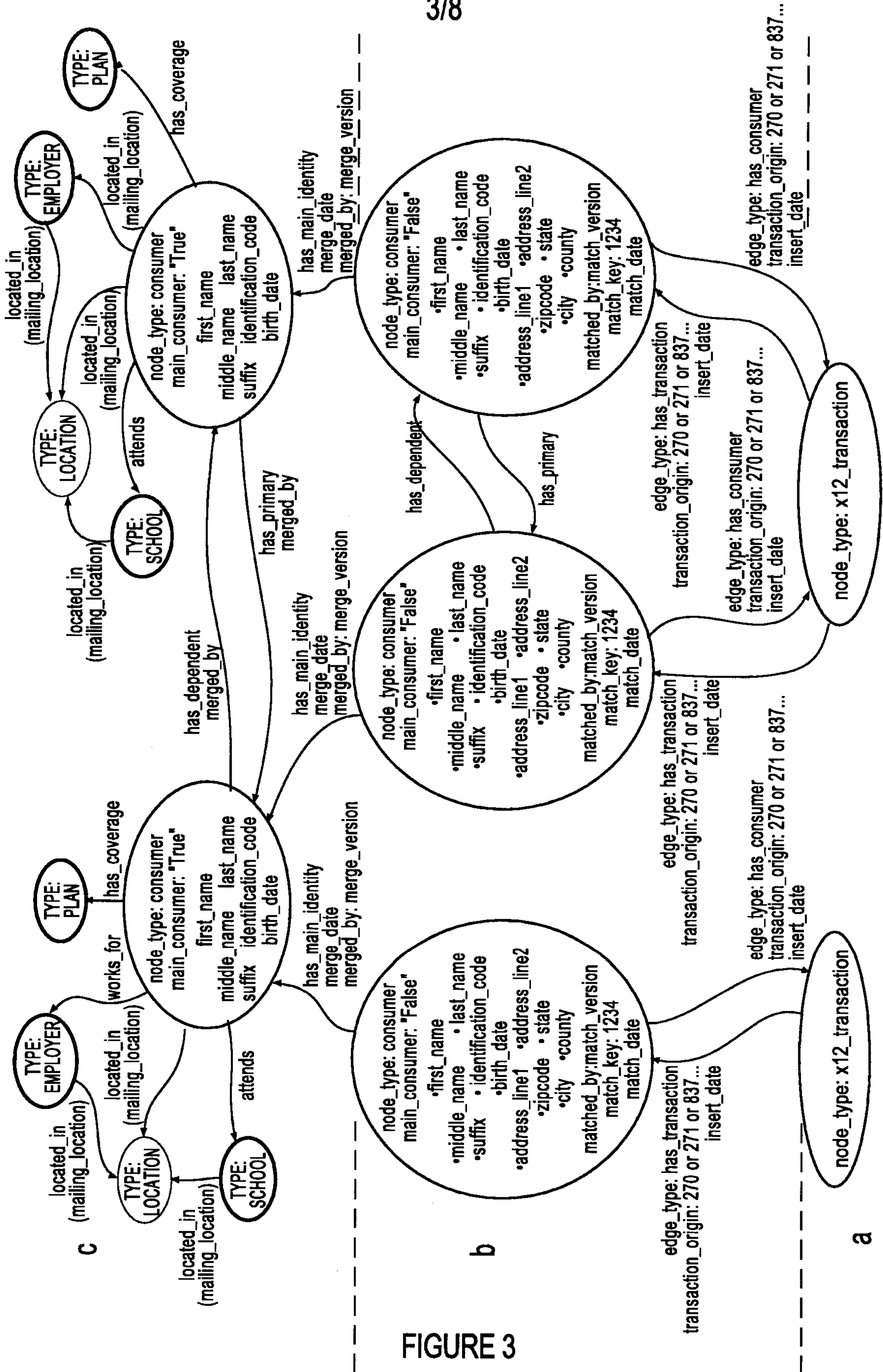


FIGURE 3

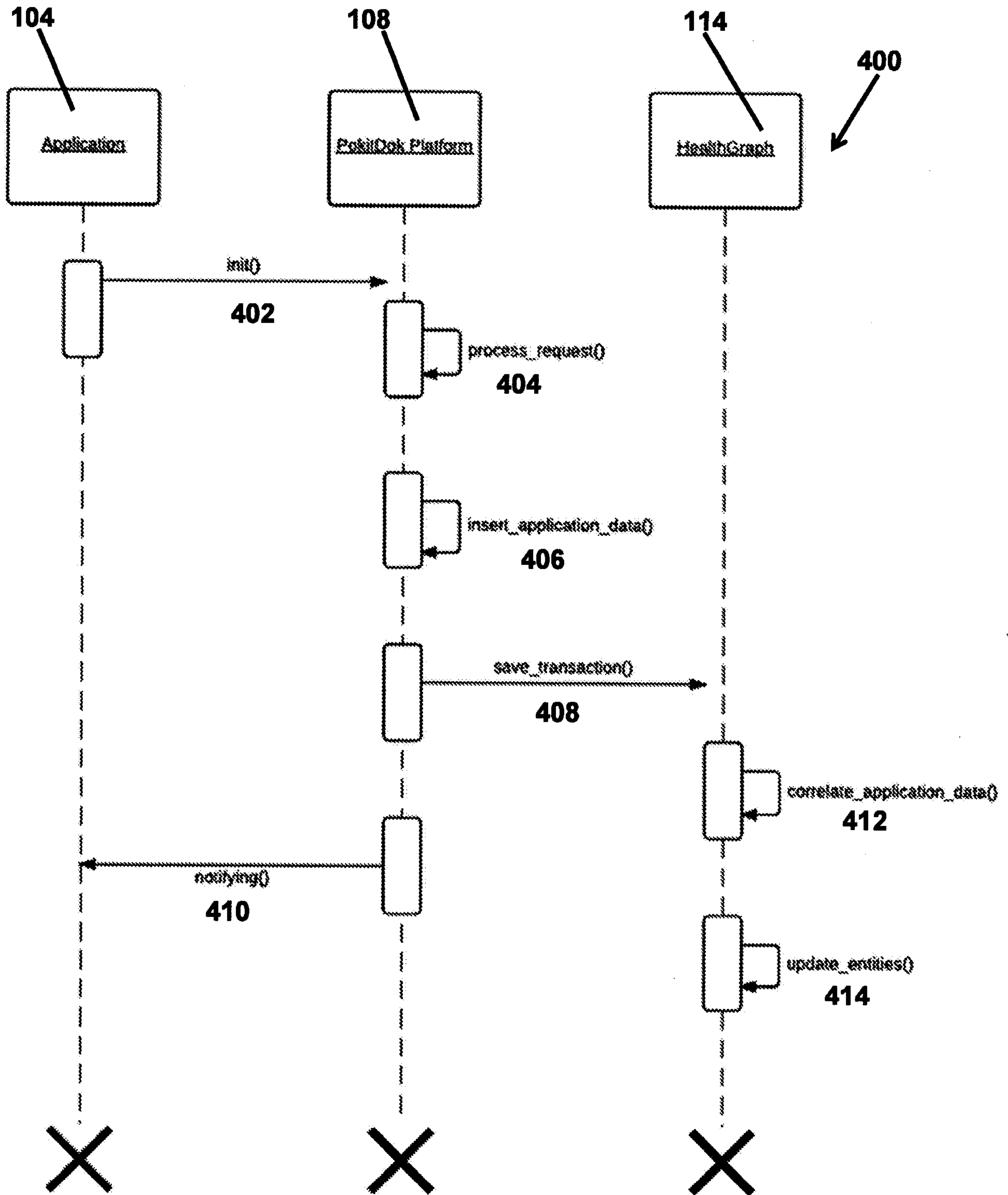


FIGURE 4

```

{
  "IEA": {
    "segment": [
      ....
    ]
  },
  "ISA": {
    "segment": [
      ....
    ]
  },
  "element_delimiter": "*",
  "field_separator": ":",
  "functional_groups": [
    .....
  ],
  "repetition_symbol": "A",
  "x12_transaction": "837",
  "application_data": [
    {
      "activity_id": "2KU1JCLRTNSWJTHZCNA",
      "patient_id": "9fd77995-6f62-49f4-bc65-09ddb7f2649e-0",
      "provider_id": "5ba623cd-5d8f-4a0c-b7af-8b65651f3a52-0",
      "other_data": "{\\"notification_id\\":
        \\"4319e686-1a10-4178-8d48-2f9733f41f9d-0\\", \\"filename_prefix\\":
        \\"2015090909090909_TEST_PROVIDER_5561ecb6-7e6f-4867-ba12-a7e09b45_837_\\\"}",
      "site_id": "b6f7021e-9654-431c-9943-efa2fd902291-0",
      "visit_id": "5561ecb6-7e6f-4867-ba12-a7e09b457620-9"
    }
  ],
}

```

FIGURE 5

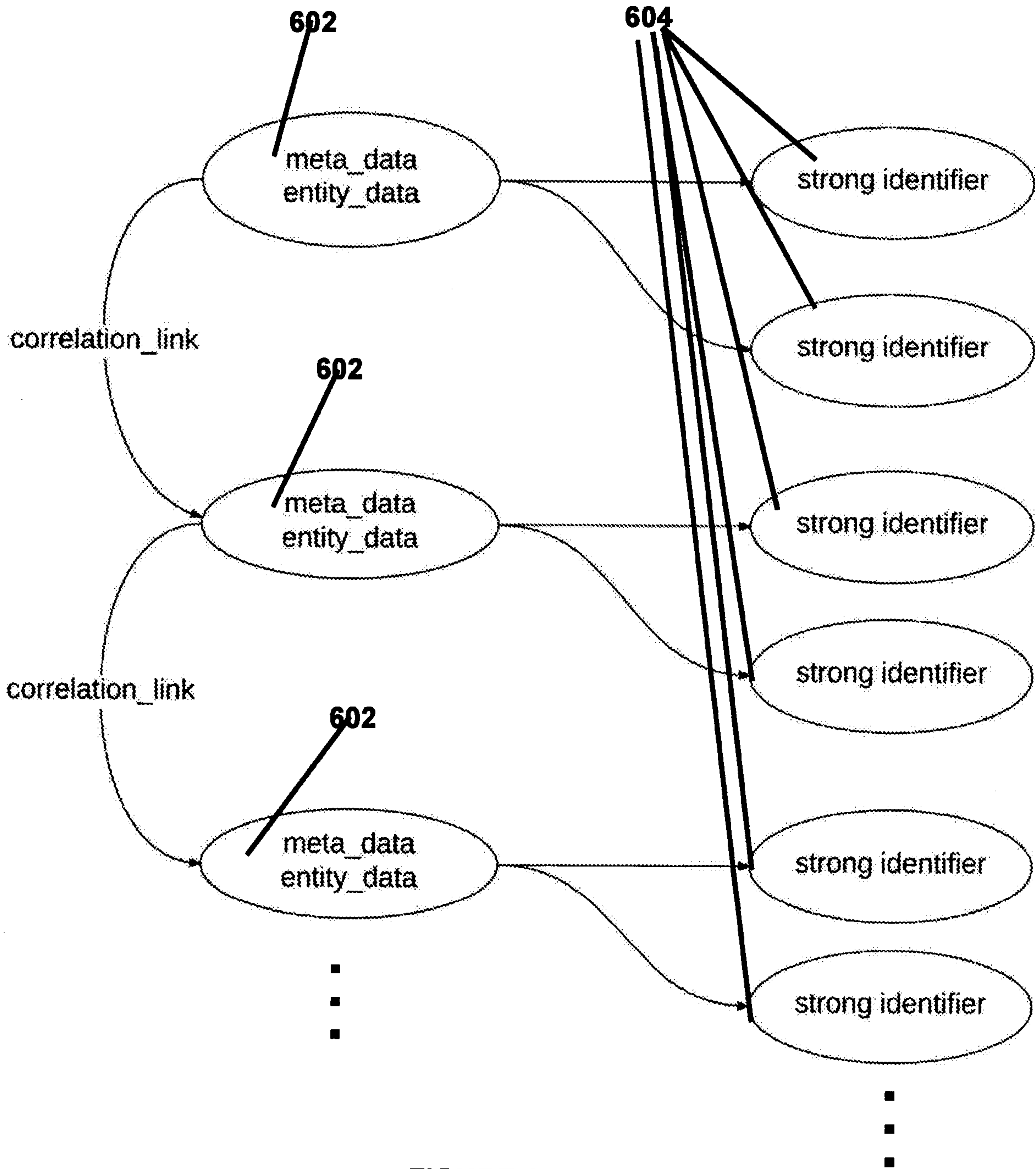


FIGURE 6

/Identity/PUT
Historical Chains

Example Graph Mutation



/Identity/PUT

```
{
  app_id: ABCD1234
  app_name: DignityHealth
  uuid: abcd-efgh-1234

  address_line1: 1 new address lane
  city: Atlanta
  email: oscar-the-dog@pokitdok.com
  first_name: Oscar
  last_name: Whitmire
  middle_name: Harold
  prefix: Mr.
  social_security_nmber: 333-22-4444
  state: GA
}
```

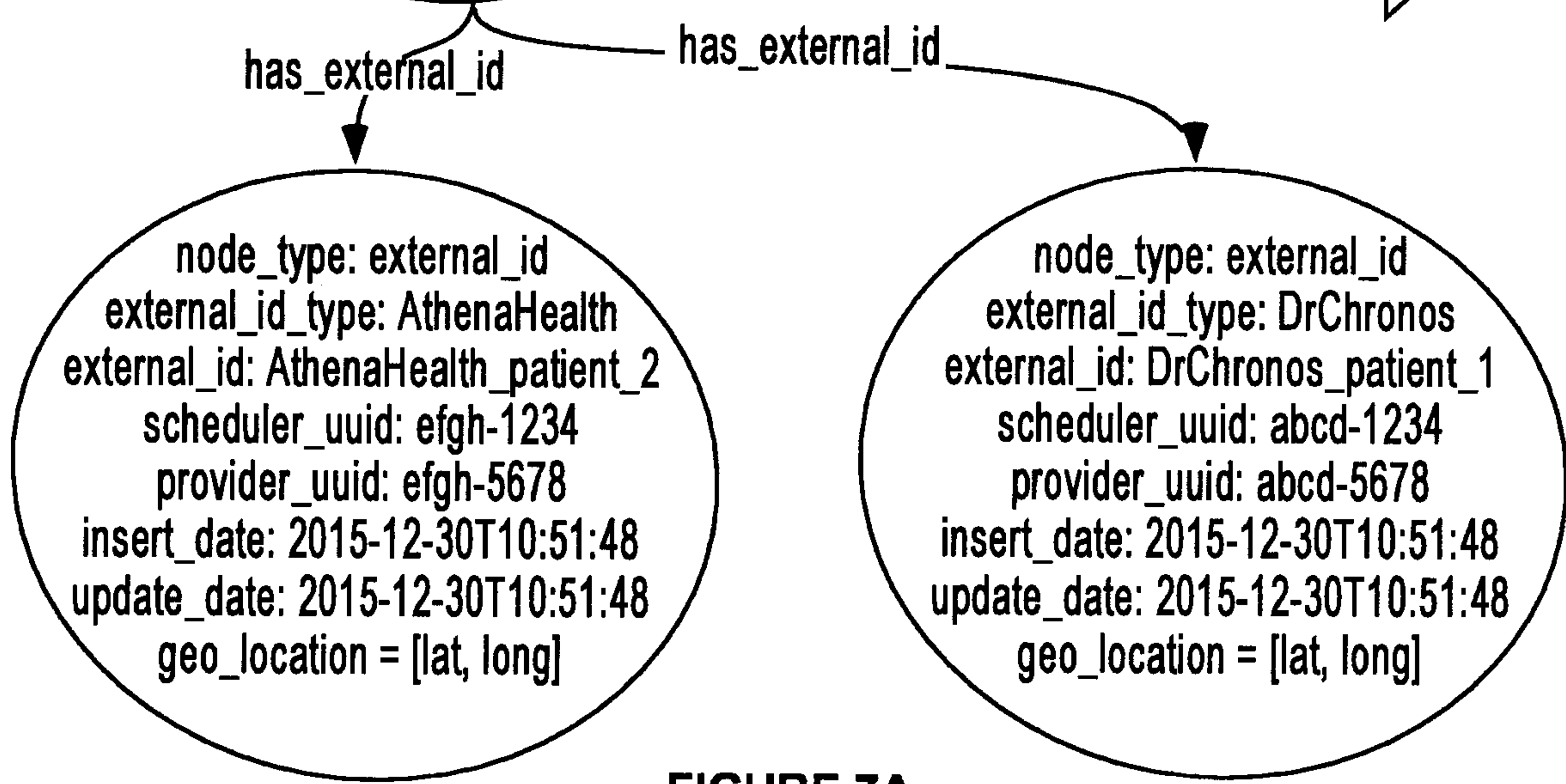
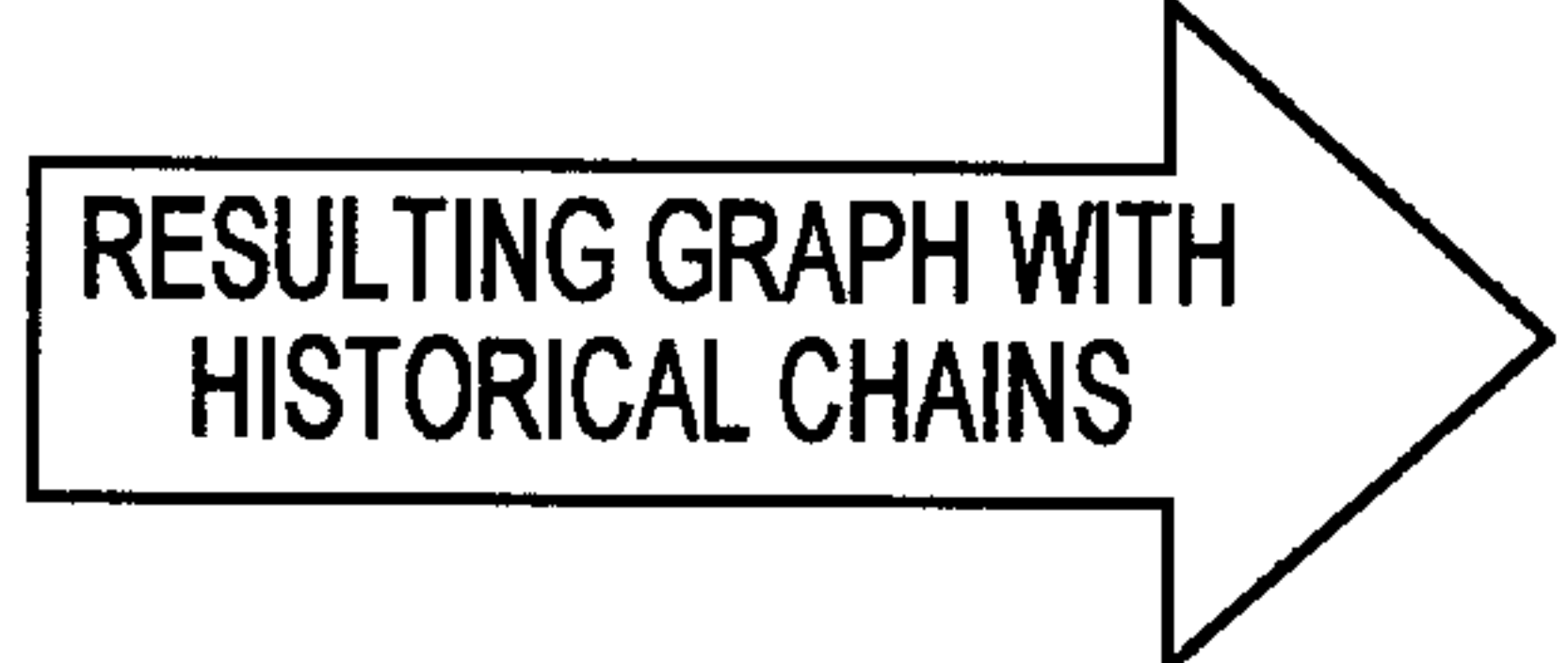


FIGURE 7A

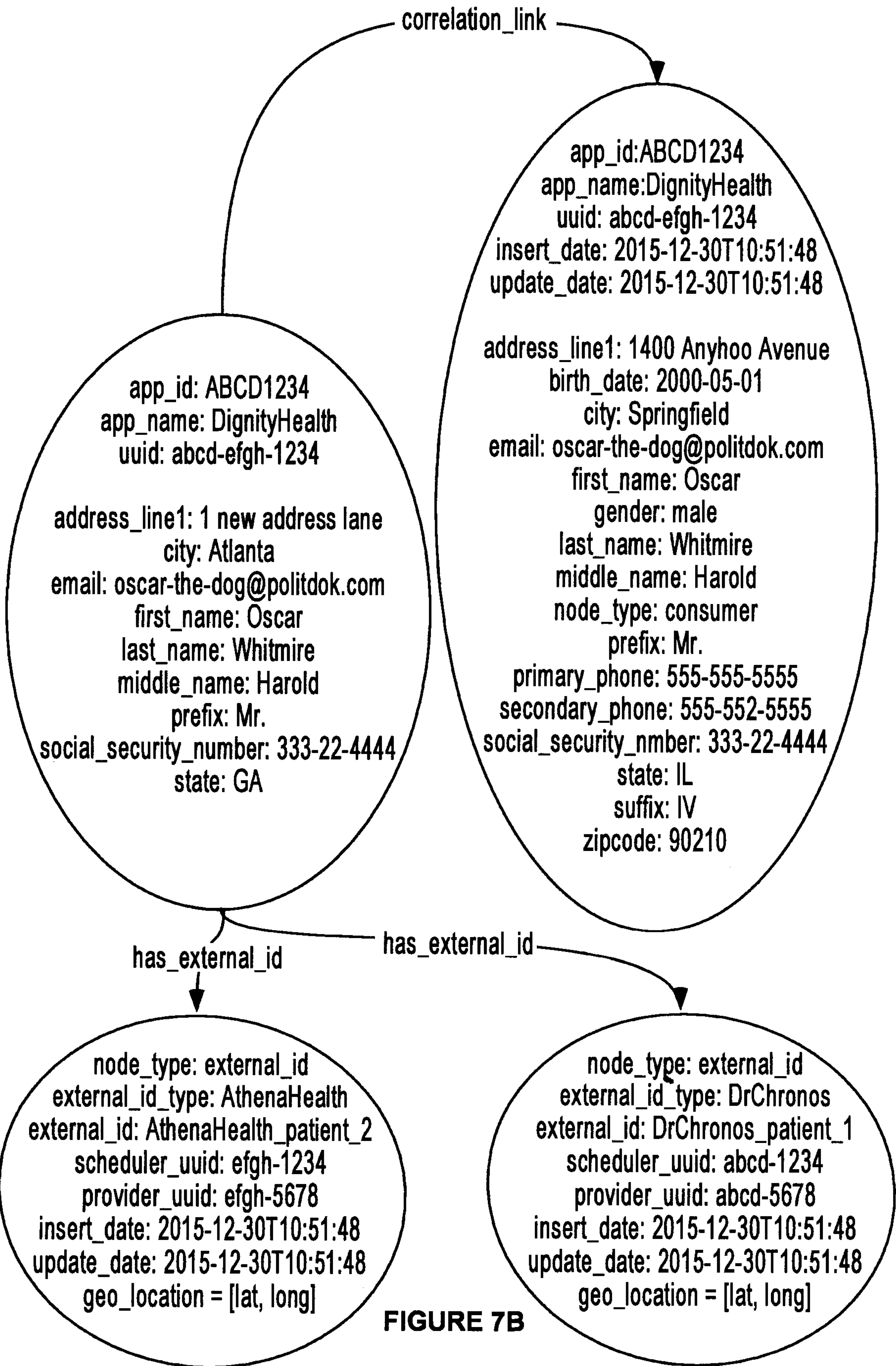


FIGURE 7B

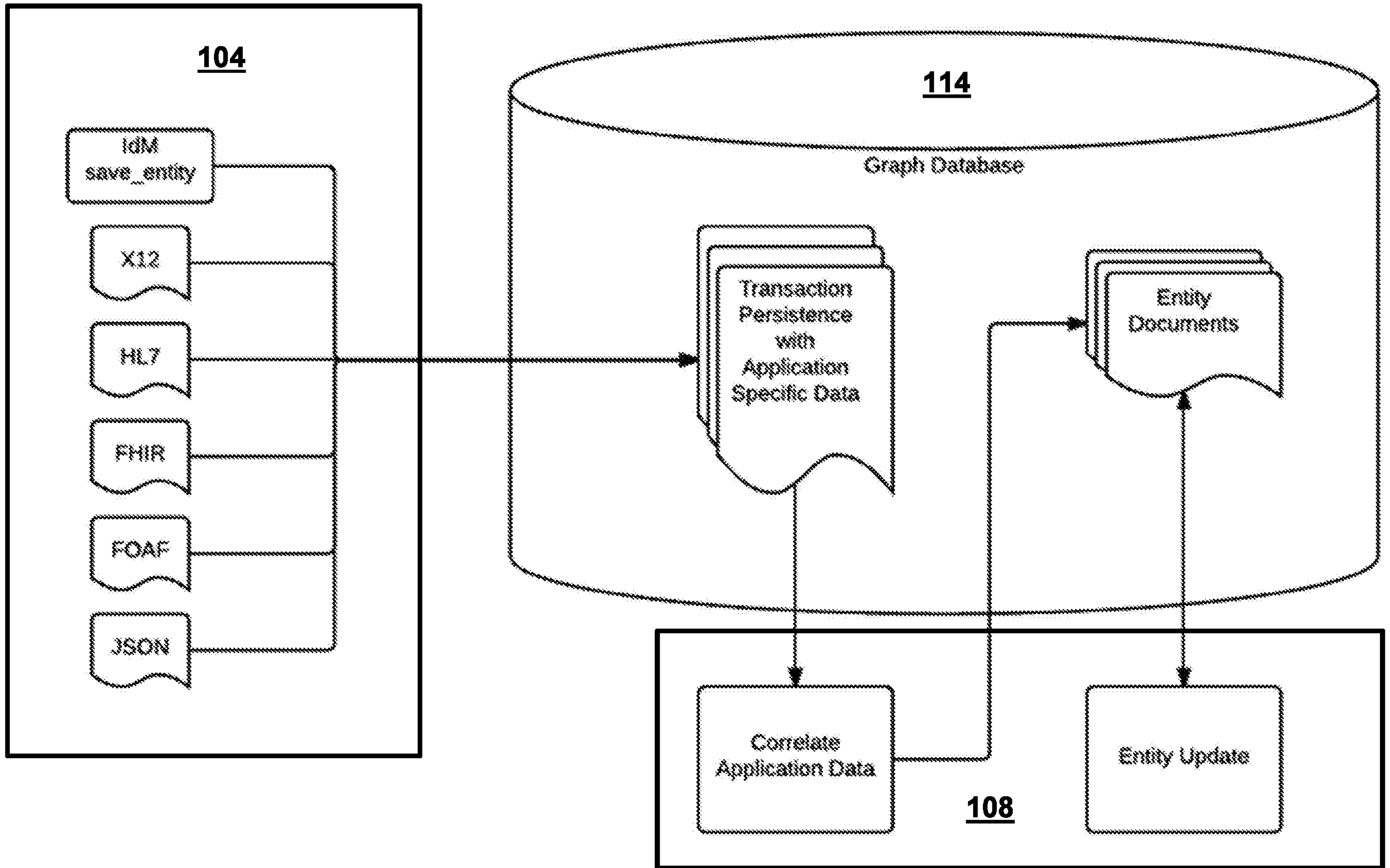


FIGURE 2