



(86) **Date de dépôt PCT/PCT Filing Date:** 2014/08/26
 (87) **Date publication PCT/PCT Publication Date:** 2015/07/23
 (85) **Entrée phase nationale/National Entry:** 2016/07/12
 (86) **N° demande PCT/PCT Application No.:** US 2014/052764
 (87) **N° publication PCT/PCT Publication No.:** 2015/108572
 (30) **Priorités/Priorities:** 2014/01/14 (US61/927,261);
 2014/08/22 (US14/466,907)

(51) **Cl.Int./Int.Cl. G06F 19/00** (2011.01),
G06Q 50/22 (2012.01)
 (71) **Demandeur/Applicant:**
 POKITDOK, INC., US
 (72) **Inventeurs/Inventors:**
 CORBIN, BRIAN SCOTT, US;
 WHITMIRE, THOMAS DIXON, IV, US;
 TANNER, THEODORE CALHOUN, JR., US
 (74) **Agent:** SMART & BIGGAR

(54) **Titre : SYSTEME ET PROCEDE DE LECTURE EN FLUX CONTINU DYNAMIQUE DE DONNEES TRANSACTIONNELLES**
 (54) **Title: SYSTEM AND METHOD FOR DYNAMIC TRANSACTIONAL DATA STREAMING**

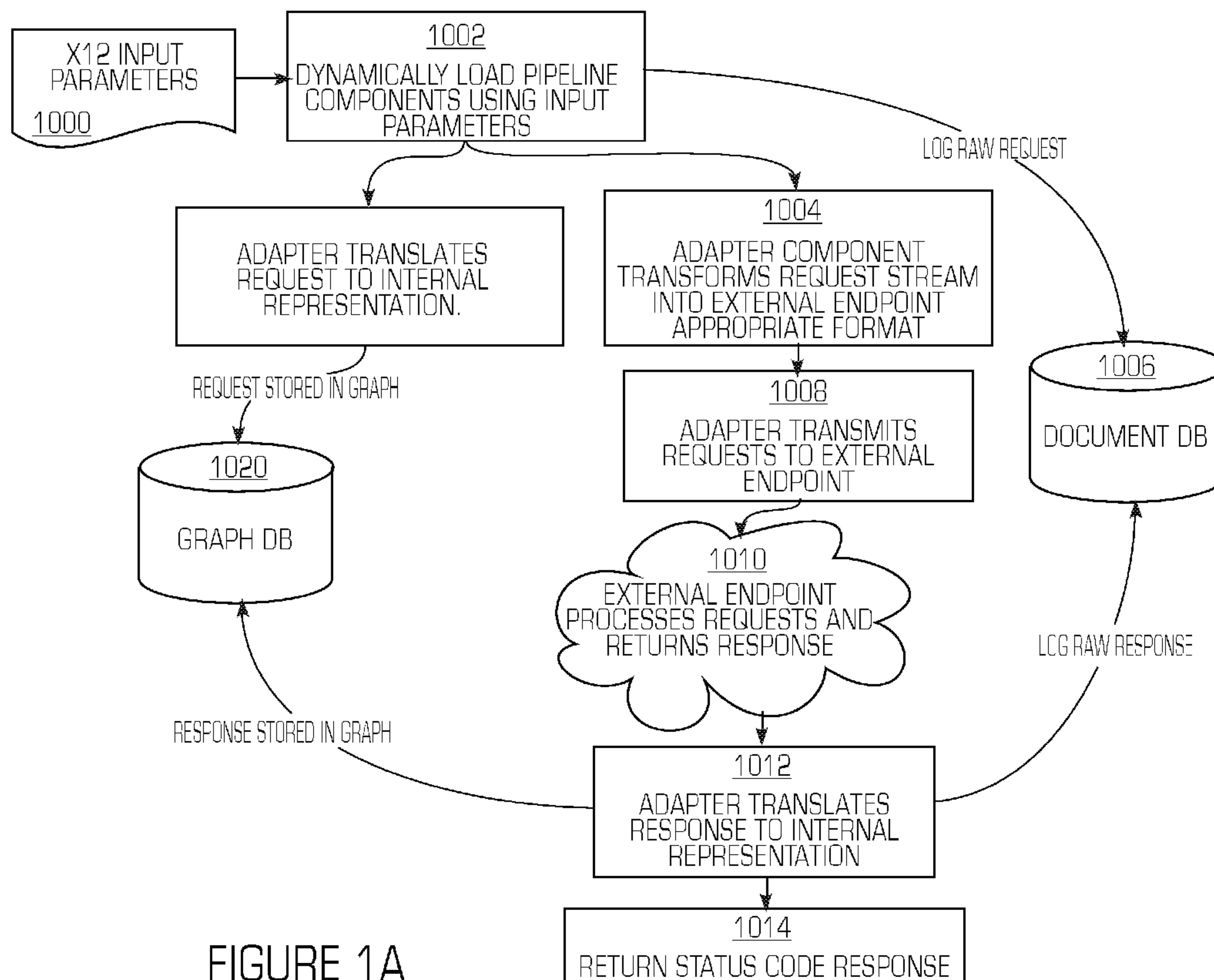


FIGURE 1A

(57) **Abrégé/Abstract:**
 A system and method for dynamic transactional data streaming are provided.

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

(19) World Intellectual Property
Organization
International Bureau(10) International Publication Number
WO 2015/108572 A1(43) International Publication Date
23 July 2015 (23.07.2015)

(51) International Patent Classification:

G06F 15/16 (2006.01) G06Q 50/22 (2012.01)
G06F 17/30 (2006.01)

(21) International Application Number:

PCT/US2014/052764

(22) International Filing Date:

26 August 2014 (26.08.2014)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/927,261 14 January 2014 (14.01.2014) US
14/466,907 22 August 2014 (22.08.2014) US(71) Applicant: **POKITDOK, INC.** [US/US]; 28 E. 3rd Avenue Suite 201, San Mateo, CA 94401 (US).(72) Inventors: **CORBIN, Brian, Scott**; 28 E. 3rd Avenue Suite 201, San Mateo, CA 94401 (US). **WHITMIRE, Thomas, Dixon, IV**; 28 E. 3rd Avenue Suite 201, San Mateo, CA 94401 (US). **TANNER, Theodore, Calhoun, Jr.**; 28 E. 3rd Avenue Suite 201, 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, 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, KZ, LA, LC, LK, LR, LS, LT, 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 TRANSACTIONAL DATA STREAMING

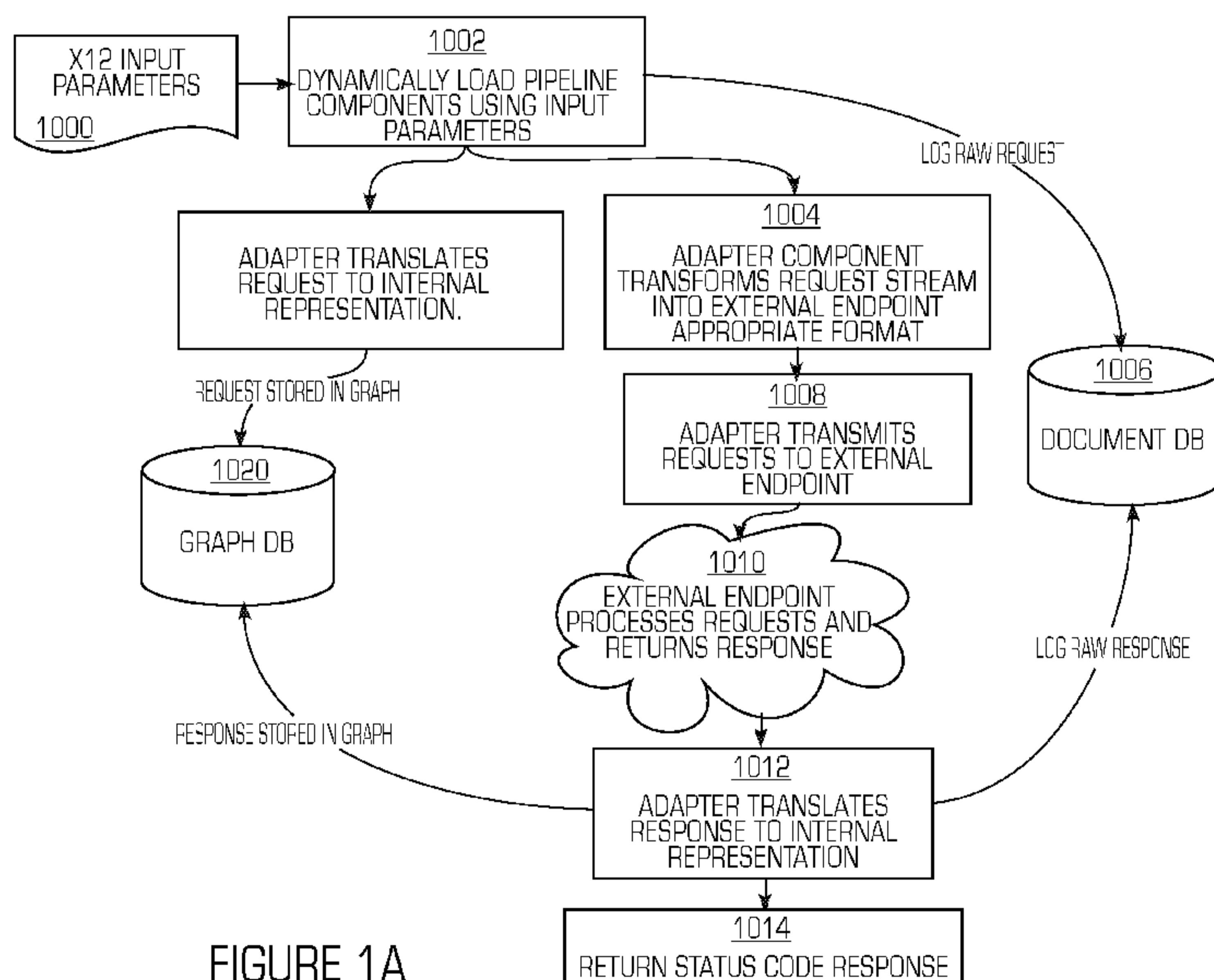


FIGURE 1A

(57) Abstract: A system and method for dynamic transactional data streaming are provided.

WO 2015/108572 A1

-1-

SYSTEM AND METHOD FOR DYNAMIC TRANSACTIONAL DATA STREAMING

Brian Scott Corbin
Thomas Dixon Whitmire IV
Theodore Calhoun Tanner, Jr.

Priority Claims/Related Applications

This application claims the benefit and priority under 35 USC 119(e) to U.S. Provisional Patent Application Serial No. 61/927,261, filed on January 14, 2014 and entitled
5 “System and Method for Dynamic Transactional Data Streaming”, the entirety of which is incorporated herein by reference.

Field

The disclosure relates generally to a transactional data streaming system and in particular to a data streaming system that may be used for health care related data.

10 Background

With the desire to lower health care costs, the United States has implemented a new health care system that, among other things, fosters competition among health care providers using healthcare marketplaces. However, the true costs of health care lie hidden in the inefficient access, delivery and payment systems that have escaped innovation and disruption
15 until now. More specifically, the transmission and reception of health care information within the context of care for the consumer has languished in past orthodoxies of obfuscated technologies and data standards. Thus, the *data* pipelines for fluid enablement of base costs for health care have been the main impediment to accessible transparent costs within the Health Industry. The current industry standard for electronic transmission of health care data
20 is called HIPAA ASC X12 5010 per the Health Insurance Portability & Accountability Act (HIPAA) of 1996.

HIPAA was supposed to make the health care system in the United States more efficient by standardizing health care transactions. To wit the acronym says *Portability not Privacy*. HIPAA added a new Part C titled "Administrative Simplification" to Title XI of the
25 Social Security Act. This effort was created to supposedly simplify health care transactions by requiring all health plans to engage in health care transactions in a standardized way.

-2-

However, HIPAA has actually created a market for IT companies that have in fact inflated the true cost of care services. More specifically, there are several companies that charge for basic Application Programming Interfaces (API), Claims eligibility and enrollment processing under HIPAA. Furthermore, there are several clearinghouses that charge for basic
5 Electronic Data Interchange (EDI).

Thus, many companies that support HIPAA ASC x12 5010 standards inject huge hidden costs within the Health IT processing flows. This is both an infrastructure cost and a usage charge. It is desirable for these costs to be as close to zero as possible in order for the consumer to truly know the cost of care for a service that is based on the current day and date.

10 The majority of existing Health IT systems are legacy data systems, conveying transactional information by means of sub-optimal file formats. The transactional data contained in these files can be difficult to generate and parse. Some systems are unable to work with these files unless they can be completely loaded into memory at once.

Brief Description of the Drawings

15 Figure 1A illustrates an example of an ASCX12 5010 Processing Pipeline;

Figure 1B illustrates an implementation of a dynamic transactional data stream introspection and processing system that may be implemented in the adapter;

Figure 2 illustrates a dynamic transactional data stream introspection and processing method that may be implemented, for example, by the system implementation in Figure 1B;

20 Figure 3 illustrates the distributed and asynchronous process infrastructure used by the dynamic transactional data stream introspection and processing system of Figure 1B;

Figure 4 illustrates an outbound transactional data stream processing method that may be implemented, for example, by the system shown in Figure 5;

25 Figure 5 illustrates an example of an implementation of dynamic data transactional system;

Figure 6 illustrates an example of a 270 Provenance Graph that may be part of the system in Figures 1B and 5;

Figure 7 illustrates an example of a 271 Provenance Graph that may be part of the system in Figures 1B and 5;

-3-

Figure 8 illustrates an example of a 276 Provenance Graph that may be part of the system in Figures 1B and 5;

Figure 9 illustrates an example of a 277 Provenance Graph that may be part of the system in Figures 1B and 5;

5 Figure 10 illustrates an example of a 834 Provenance Graph that may be part of the system in Figures 1B and 5;

Figure 11 illustrates an example of a ASCX12 5010 Inter-Domain Graph that may be part of the system in Figures 1B and 5;

10 Figure 12 illustrates an example of an implementation of a healthcare services marketplace system that may utilize the dynamic transaction system and method;

Figure 13 illustrates a ASCX12 5010 sample input file, containing both eligibility (270) and enrollment (834) transactions; and

Figure 14 illustrates how rule sets are dynamically loaded and applied to transaction sets.

15 Detailed Description of One or More Embodiments

The disclosure is particularly applicable to a healthcare marketplace system that uses the dynamic transactional data streaming and it is in this context that the disclosure will be described. It will be appreciated, however, that the dynamic transactional data streaming has greater utility since it may be implemented in different ways than those described below and
20 may be used without the healthcare marketplace system or used with other systems that can take advantage of the benefits of the dynamic transactional data streaming.

The HIPPA ASC X12 5010 standard defines a number of different health care related transactions and code sets including:

- (270/271) for health care eligibility inquiry and response transactions
- 25 • (274) for health care provider information
- (275) for patient information
- (276/277) for health care claims, status, and acknowledgement transactions
- (278) for health care services review

-4-

- **(834/835/837)** for Benefit Enrollment, Claim Payment and Care Claim transactions.

Although the disclosure below is directed to an example of the system that uses HIPPA ASC X12 5010 standard, the transactional data streaming system and method may be used with various standards or later revisions of the X12 standard since the system and method could be easily adjusted to work with the various standards or later revisions of the X12 standard.

As described above, the current systems inject huge health care IT costs into the process and thus the system and method provide a low cost pipeline that would enable the above transactions. The system and method may also provide a topology that enables lower costs along with a “graph based” representation of several data sources that includes the entire HIPPA ASC X12 5010 standard.

The system and method provide a technique to efficiently process this health care related information as dynamic transactional data streams. In one aspect of the system, each file may be processed using streaming to minimize the memory footprint. As the transactional data moves through the system, the transactional data may be dynamically analyzed so that rules for parsing and processing the data can be loaded up at run time. As the context changes in the data stream, new rules can be injected to adjust the data processing. Dynamic rule injection also provides for minimized system downtime as new behaviors can be added while data is streaming through the system.

To understand the dynamic transactional data streaming system and method, first an example of the processing pipeline for the ASC X12 5010 standard is described with reference to Figure 1A. As shown in Figure 1A, the processing pipeline may receive one or more x12 input parameters (1000) that may be dynamically loaded into pipeline components (1002.) The system logs the raw request in a document database, prior to executing the adapter. The system may use an adapter component 1004 to transform the request stream to separate internal and external endpoint data formats as described below with reference to Figures 1B and 2-4. Examples of data formats include, but are not limited to XML, SOAP messages, JSON, delimited data formats, etc. The system’s internal data format is used throughout the system to ensure that data is processed in a uniform manner. External endpoint data formats vary based on the requirements of the external endpoint, and are generated based

-5-

on the adapter component's configuration. The adapter component 1004 transmits the transformed request stream to the external endpoint. The external endpoint then processes the request, and returns a response. The system logs the response stream into a document database. The adapter component may then translate the response to the internal
5 representation, which is then persisted in the graph database. Then, the status code response is returned to the original requestor to complete the processing pipeline.

Figure 1B illustrates an implementation of a dynamic transactional data stream introspection and processing system 1500 that may be implemented, for example in the adapter 1004 shown in Figure 1A. The system 1500 may have a processing and messaging
10 component 3000 that is described in more detail with reference to Figure 3 that may be used to facilitate the functions of the components 1500A, 1500B, 1500C and 1500D of the system. The system 1500 and each component of the system may be implemented in a combination of hardware and software using one or more computing resources, such as one or more server computers or one or more cloud computing resources that have one or more processors and a
15 memory connected to the one or more processors. In this implementation, each of the components of the system may be a plurality of lines of code that may be executed by the processor of the computing resources. Alternatively, each of the components of the system may be a hardware device (or all of the components implemented in one hardware device) wherein the hardware device, based on logic programmed to implement the functions and
20 operations of the components described below, may implement each component. The hardware device may be a microcontroller, a processor, a programmable logic device, a field programmable gate array, an application specific integrated circuit and the like. For example, each component may be implemented by a hardware processor that is configured to perform the operations and functions of the component.

25 Returning to Figure 1B, the system 1500 may further have a data stream examining component 1500A that performs initial analysis of the transactional data stream as described in more detail in process 2004 in Figure 2. The system 1500 may further have a data stream segment processing component 1500B that analyzes and processes each segment of each data stream and selects an appropriate rule set to use for the processing based on the
30 information/data in the particular transaction datastream being currently processed as described in more detail in processes 2006-2022 in Figure 2. The system 1500 may further

-6-

have a transactional data rule set store 1500C that stores a plurality of rule sets that may be selected by the data stream segment processing component 1500B. Thus, using the rule set store 1500C and the data stream segment processing component 1500B, as the transactional data moves through the system, the transactional data may be dynamically analyzed so that rules for parsing and processing the data (the rule sets in the rule set store 1500C) can be loaded up at run time. Furthermore, as the context changes in the data stream, new rules can be injected into the rule set store 1500C or selected by the data stream segment processing component 1500B from the already existing rule sets in the rule set store 1500C to adjust the data processing. Dynamic rule injection also provides for minimized system downtime as new behaviors can be added while data is streaming through the system.

To illustrate how rule sets are dynamically applied to transaction sets, consider a ASCX12 5010 file containing both eligibility and enrollment transaction sets as shown in Figure 13. In the ASC X12 5010 file specification, functional groups are used to group transactions by transaction types. Transaction data is contained within a functional group within a transaction set. The functional group acts as an outer envelope for one or more transaction sets, and is delimited using GS and GE segments. ST and SE segments enclose data within a transaction set. The system parses the GS and ST segments and infers which rule sets are used, based on the transaction type as follows in this example:

ASCX12 5010 Control Segments	Transaction Set Identifiers
GS*HS* 1234567890* 654456654*20130810*0409*1*X*005010X279A1~	005010X279A1 (Transaction Version)
ST*270*0001*005010X279A1	270 (Transaction Set Type) 005010X279A1 (Transaction Version)

Figure 14 illustrates the how the system dynamically applies rules to transaction sets. A data stream containing multiple transaction sets, enters the system and is processed by a stream parser. The stream parser parses the input stream into the data into a structured format such as JSON. The transaction sets from the input stream are preserved in the structured format. The rules engine inspects each transaction set in the input stream, and determines which rule sets to apply, based on transaction set identifiers. Thus if an input stream contains

-7-

multiple transaction sets, the rules engine is capable of dynamically loading the rule set on demand to process the transaction. Once the transaction set has been identified, the rules engine determines if the rule set is loaded. If the rule set is not loaded, the rules engine loads the rule set from the rule store. Finally, the rule set is applied to the transaction set. The rule set processing channels are illustrated in Figure 3. The system 1500 may further comprise a data stream output component 1500D (details of the process is shown in Figure 4) that generates an output data stream.

Figure 2 illustrates the dynamic transactional data stream introspection and processing method (2000) utilized by the dynamic transactional data stream introspection and processing system 1500 of Figure 1B. For example, the dynamic transactional data stream introspection and processing method (2000) may be implemented in the adapter component (1004) in the processing pipeline in Figure 1A. The system may also receive input parameters identifying the input data source and its transmission properties, such as the EDI sender, receiver, and payor identifiers, as well as data element separators. Transmission properties define the channel for the EDI transmission, specifying the transmission source and endpoint. As shown in Figure 1A, the dynamic transactional data stream may have the adapter and connector components that the system inspects its input parameters and dynamically instantiates the adapter and connector components to handle the inbound request. The adapter forms requests and parse responses in a manner compatible with the channel's endpoint destination. The adapter also may transmit requests and receives responses via the connector component. The connector components provide data transmission services using protocols such as FTP, HTTP, SOAP. The process depicted in Figure 2 activates rule sets based on the current transaction stream. The rule sets are used to read and parse data into a data structure which is then returned back to the caller.

Figure 3 illustrates the processing environment used by the dynamic transactional data stream introspection and processing system shown in Figure 1B. The processing environment is distributed across multiple processing channels. Processing operations are non-blocking due to the use of asynchronous messaging. Parallel processing is achieved through the use of multiple processing channels, where separate channels could refer to physically separate processing nodes, separate threads of execution, or a hybrid model where processing occurs on multiple nodes with each node using a multi-threaded environment. The inbound i/o

-8-

stream (3002) is segmented into units of work by a stream parser (3004). The unit of work is specific to each transaction set, such as eligibility (270), enrollment (834), etc. Within each transaction set, the units of work are composed of segments which are composed of fields. The transaction set specific rule set is used to parse the unit of work from a transaction set.

5 The stream parser determines the appropriate unit of work by interrogating the inbound i/o stream, and then forwards the unit onto a processing channel (3006). A processing channel provides the services and execution environment for transforming the unit of work to the appropriate format. Within the processing channel, asynchronous message queues (3008) are used to support parallel processing within the distribution channel. Finally, 10 a rule set processor (3010) is used to retrieve the appropriate rules from the system's rule engine (3012), and then apply these rules as described below to the unit of work as is described in more detail below in Figure 3. Transformed units of work are coalesced into an outbound i/o stream (3014), which is then returned to the calling process.

 Referring back to Figure 2, in the method, the system shown in Figures 1B or 5 may 15 inspect an incoming stream for the segments and type of data contained within the payload and may also receive a transaction data stream 2002 and the system may examine the transaction data to determine the element separator and segment terminator (2004.) The known ASC X12 5010 standard (more details of which may be found at 20 http://www.cms.gov/Regulations-and-Guidance/HIPAA-Administrative-Simplification/Versions5010andD0/Version_5010.html, which is incorporated herein by reference) conveys data using a delimited format. In this format, data elements are separated using an element delimiter. Data elements are grouped into a larger structure referred to as a segment. Segments are delimited from one another using a segment terminator. The control characters used as the element separator and segment terminator are machine readable, and 25 embedded within the ASC X12 5010 data stream. The system also may inspect the respective stream to determine which components are used to process the payload and components are associated with data elements by configuration in a configuration data store. Thus, the detection of the delimiters identifies each segment of the transactional datastream.

 The system may activate an appropriate rule set based on the transaction set 30 information (2006.) An example of the rules set based on transaction set information is discussed below with reference to the health care marketplace system scenario that shows

how a healthcare marketplace system might use dynamic transactional data streaming during a consumer/health care provider interaction. The system may then read a next element from the data stream using the termination identifiers (2008.) The system may look up parse expression(s) for the data (2010) and parse the data based on the parse expression (2102.) An example of the parsing process is described below in the health care marketplace system scenario. The system may then apply rules to the parsed data (2014.) An example of this rule application process is described below in the health care marketplace system scenario.

The system may then determine if an end of the transaction set has been reached (2016) and loops back to 2008 to read the next segment if the end of the transaction set has not been reached. If the end of the transaction set has been reached, then the system may store the data generated by the rules into a graph form (2018) that may be stored in a graph data store 2020. For example, a graph model may be built with respect to the types of data that can be inferred across concerning the streams. Once the data is stored in a graph form, the system determines if more transaction sets need to be processed (2022) and the transaction data stream processing is completed if there are not any more transaction sets to be processed. If there are more transaction sets to be processed, then the system loops back to process 2006 to activate the appropriate rule set.

To better understand how this might work in practice; consider a scenario where a consumer and a health care provider are interacting within a health care marketplace system that utilizes this dynamic transactional data streaming technique. When the consumer first engages with the health care provider about a condition, the provider can utilize the health care marketplace system's eligibility check functionality to verify the consumer's health insurance and access their deductible information. The eligibility check functionality transmits a X12 270 transaction set (defined by the ASC X12 5010 standard above) to the consumer's insurance company to inquire about their deductible and general coverage information. When the insurance company responds with a X12 271 transaction set (defined by the ASC X12 5010 standard above) for benefit eligibility information, the system will use parse expressions based on the X12 file specifications defined by the ASC X12 5010 standard above in order to decompose segments in the transaction set into data elements. The parsed segment data is passed through a rules engine that is initialized with rules from a rule set that has been defined for the specific transaction set and trading partner. The following table

-10-

includes some example parse expressions that would be used to parse segments found in a X12 271 transaction set:

<i>Example Parse Expressions for segments</i>	<i>Example Eligibility Inquiry Data Segments</i>
HL*{HL01}*{HL02}*{HL03}*{HL04}~	HL*3*2*22*0~
NM1*{NM101}*{NM102}*{NM103}*{NM104}* {NM105}*{NM106}*{NM107}*{NM108}*{NM109}* {NM110}*{NM111}*{NM112}~	NM1*IL*1*SMITH*JOHN*B***MI*W199100000~
EB*{EB01}*{EB02}*{EB03}*{EB04}*{EB05}* {EB06}*{EB07}*{EB08}*{EB09}*{EB10}*{EB11}*{ EB12}*{EB13}*{EB14}~	EB*C*IND*30***23*3000*****Y~

Segment element IDs are defined in brackets, { }, inside each parse expression and may be optionally mapped to more meaningful variable names such that they may be referenced in rules by the more friendly variable name instead of the element ID.

A sample rule subset that can process segments found in a benefit eligibility inquiry response (271) transaction set is included in the table below along with sample data segments that would cause the rule(s) to execute:

<i>Example Rules for an Eligibility Inquiry</i> <i>Rule Set</i>	<i>Example Eligibility Inquiry Data Segments</i>
“Hierarchical level rule”: if the current segment has an id of “HL”, then store the HL segment information in working memory for later use	HL*3*2*22*0~
“Subscriber identification rule”: if the current segment has an id of “NM1” and the “hierarchical level code” (or HL03 element id) in the current “HL” segment is “22” and the “entity	NM1*IL*1*SMITH*JOHN*B***MI*W199100000~

<i>Example Rules for an Eligibility Inquiry Rule Set</i>	<i>Example Eligibility Inquiry Data Segments</i>
identifier code” (or NM101 element id) in the “NM1” segment is “IL”, then store the “NM1” segment information on the X12 domain model	
“Eligibility deductible information rule”: if the current segment has an id of “EB” and the “eligibility/benefit information” (or EB01 element id) in the segment has a value of “C” then store the deductible information found in the segment on the X12 domain model	EB*C*IND*30***23*3000*****Y~

After verifying the consumer’s health insurance, the provider examines the patient and makes a diagnosis. Since the provider did a general eligibility inquiry (X12 270) to determine the consumer’s current deductible information, the provider is now equipped to recommend a set of treatment options that the consumer can pay for with cash or insurance. With a diagnosis and treatment(s) identified, the provider can initiate a more specific eligibility inquiry (X12 270) with codes (typically CPT or ICD-10) for the treatments to determine if the recommended treatments are covered by the consumer’s insurance plan. This allows the consumer to make informed decisions regarding the treatments and their costs while they’re still meeting with their health care provider. Once a treatment is selected, the health care marketplace system will record the treatment purchase transaction and submit the necessary X12 837 claims to the insurance company if the consumer elects to (partially) pay with insurance. If there is a portion of the treatment cost remaining after processing the X12 835 health care claim payment response, the health care marketplace system can then bill the consumer via their credit card on file and deposit the funds in the provider’s bank account along with the insurance payment that was delivered in the X12 835 claim payment transaction set. Each X12 transaction set received by the health care marketplace system can

-12-

make use of the dynamic transactional data streaming outlined in Figure 2. For example, when a X12 271 transaction set is received following the benefits eligibility inquiry for the consumer visiting the health care provider, the 271 transaction data is examined (2004). After this examination, the appropriate rule set for the X12 271 transaction set is loaded into a rules engine (2006). For each data segment present in the 271 transaction set, the system will stream the segments through a stream based parser that will determine the appropriate parse expression based on the current segment's identifier (2008 and 2010). The segment data will be parsed according to the matched parse expression (2012) and then the parsed segment data will be loaded into the working memory for the rules engine so that the rules may be applied to the segment data (2014). The process of streaming in segments from the 271 transaction set, parsing them, and applying rules continues until the end of the transaction set is reached (2016). The eligibility information is stored in a graph data store once the transaction set is processed (2018 and 2020). This process may also be repeated for additional transaction sets that may be received from the insurance company (2022). The graph data stored in 2020 is available to the consumer and health care provider as part of the health care marketplace system.

Figure 4 illustrates an outbound transactional data stream processing method 4000 that may be implemented, for example, by the system shown in Figures 1B and 5. In the method, the system may select sub-graphs that contain transactional information for streaming (4002) from the graph data store 1020 described above with reference to Figure 1A. The system may then load appropriate stream generation rules based on current graph data in query results (4004) and generate appropriate records on the output stream by executing the loaded rules against the graph data (4006) which is communicated as a data file stream 4008.

The table below illustrates how the system utilizes rules to generate a claim status request (276) output stream using the graph data store domain.

Claim Status Request Sample Rules	276 X12 Output Stream Segments
<p>“New Patient Record Rule”: When the rules engine encounters a patient data attribute on the X12 domain model, add a new HL segment to the output stream to indicate the start of a new</p>	HL*4*3*22*0~

patient record.	
"Patient Birth Date and Gender Rule": When the patient has a date of birth and gender on the X12 domain model, add a new DMG segment to the output stream	DMG*D8*19191029*M~
"Patient Identification Rule": When the patient has a name and member id, generate a NM1 segment and add it to the output stream	NM1*QC*1*DOE*JOHN****MI*R11056841~

After the appropriate records are generated, the system may determine if the end of the graph query results has been reached (4010) and delivers the generated data file(s) to a trading partner(s) to communicate the transaction (4012) if the end of the graph query results has
5 been reached. If the end of the graph query results has not been reached, then the system may loop to process 4004 to load the appropriate rules.

Figure 5 illustrates an example of an implementation of a dynamic data transactional system 5000 that may implement the methods described above. The system may be implemented using one or more computing resources (for example, cloud computing
10 component(s) or server computer(s) with typical elements like processors and memory) and one or more components (including the adapter and connector components and some of the components in Figure 5) that may be partially implemented by a plurality of lines of computer code. In another implementation, each of the components in Figure 4 (and the adapter and connector components) may be implemented in hardware or a hardware device, such as a
15 programmable logic device, an ASIC and the like.

Inbound requests are routed to a load balancer that distributes the processing load between one or more Enterprise Service Bus, ESB, and components. Each ESB is a software platform that hosts the ASC X12 Processing Pipeline and makes it available to external clients. The ESB utilizes two data stores, an operational database and a statistics cache to
20 support its operations. The operational database tracks the services and components which are supported by the ESB. The statistics cache records service metrics such as the number of times a service has been executed, or a service's average response time. Both databases may be used to provide insight into the current health of the overall system. The system uses document and graph databases as depicted in Figure1. The ESB environment is maintained

-14-

using the ESB administration application. The administration application is an operations console used to add , configure, and remove services, view processing statistics, and schedule recurring tasks. Each of the stores and the cache may be implemented as a software database or a hardware database.

5 Using the system, a transactional data file is communicated/received by a trading partner participating in data exchange. Trading Partner named herein, any other person or entity who will be receiving EDI information from or providing EDI information to another entity and who requires a submitter number because of a business need. Such person or entity may be an individual provider, a clinic composed of multiple providers, a clearinghouse or a
10 billing agent. In the system, the graph model described above allows the X12 system that is historically row/column key-value based to be queried in a more knowledge domain model behavior. Furthermore, the system dynamically generates inter-domain relationships for nodes within the graph model. These graph relationships provide a holistic view of a member's data within a single system. In addition, the system may use Dynamic query based
15 models that do not need traditional relational algebras.

Figures 6 - 10 illustrate the system's graph representation of ASC X12 5010 data segments. The transaction sets represented included: 270 (Figure 6), 271 (Figure 7), 276 (Figure 8), 277 (Figure 9), and 834 (Figure 10). Each transaction set's graph representation includes schema and data nodes. Schema nodes are used to map ASC X12 5010 segment
20 fields to domain names. The following example, from Figure 6 maps "name" fields from the ASC X12 5010 name segment.

Field Name	Domain Name
NM101	Entity Identifier Code
NM102	Entity Type Qualifier
NM103	lastname
NM104	firstname

Data nodes store instances of persisted ASC X12 5010 data segments.

Field Name	Data Value
NM101	P
NM102	1

-15-

NM103	LEE
NM104	GARY

The graph system models the relationships between nodes. This information is used to associate node segments with one another using well defined relationships such as “hasDependent”, “hasReceiverAddress”, and “isEntityType”. The system also establishes provenance, or origin, relationships which are used to associate nodes with parent or container nodes. These relationships preserve the ASC X12 5010 control envelopes used to convey transaction sets.

An example of a cypher graph queries that could be used to access graph data for outbound transactional data streaming may be:

```

10      start subGPProv = node:SubGProv(type= SubGProv) match tranSetProv<-
      [:hasTransactionSetProv]-subGPProv-[:hasSubGProv]-member,member-
      [:hasMemberNameInfo]-memberNameInfo-[:hasMemberName]-
      memberName,member-[:hasMemberEmployerInfo]-
      >memberEmployerInfo,memberEmployerName<-[:hasMemEmployerName]-
15      memberEmployerInfo-[:hasMemEmpCommunicationNumber]-
      >memberEmpCommunicationNumber ,memberEmployerInfo-
      [:hasMemEmpStreetAddress]->memberEmpStreetAddress where
      tranSetProv.ST01 = 834 and memberName.NM109 = 123456789 return
      memberEmpCommunicationNumber,memberEmpStreetAddress,memberEmployer
20      Name;

      start subGPProv = node:SubGProv(type= SubGProv) match tranSetProv<-
      [:hasTransactionSetProv]-subGPProv<-[:hasSubGProv]-member-
      [:hasMemberNameInfo]->memberNameInfo-[:hasMemberName]-
      >memberName,member-[:hasMemberHealthCoverageInfo]-
25      >memberHealthCoverageInfo-[:hasHealthCoverage]->healthCoverage where
      tranSetProv.ST01 = 834 and memberName.NM109 = 123456789 return
      healthCoverage;

```

A healthcare marketplace system may provide a transparent health services marketplace with clear descriptions and posted prices. Many health care providers and payers

-16-

use legacy systems to communicate information for a variety of transactions: eligibility checks, claims processing and benefits enrollment. To integrate the healthcare marketplace system capabilities with existing systems in the health care space, it's important that it be able to process massive streams of transactional data related to health care services. The ability to process these transaction streams enables: real-time eligibility checks for quote requests, submitting a claim for a service after paying cash so that the service cost can contribute toward a deductible, enrolling a consumer in new health benefits so that they might save money on expensive services. Integrating all of these transaction capabilities with the health service marketplace provides consumers with easy access to information to help them make informed decisions concerning their health care spending. It also provides health care providers and payers with more efficiencies so that administrative costs for processing health care transactions approach zero. Without the dynamic transactional data streaming capabilities, consumers would only be able to use the healthcare marketplace system for cash based transactions and would have to consult other systems for insurance based pricing. The dynamic transactional data streaming may provide the best possible user experience for health care consumers and providers participating in the health care services marketplace.

Figure 12 illustrates an example of an implementation of a healthcare services marketplace system 100 that may utilize the dynamic transaction system and method. 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 12, 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 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. 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

-17-

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.

The backend system 108 may also have a health marketplace engine 110, a request for quote engine 112 and a predictive pricing engine 113 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 health marketplace engine 110, the request for quote engine 112 and the predictive modeling engine 113 may each be implemented in software in which 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 health marketplace engine 110, the request for quote engine 112 and the predictive modeling engine 113 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 store 114 that stores the various data and software modules that make up the healthcare system. 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 dynamic transaction system components described above may be incorporated into the backend system 108 or may be coupled to the backend system 108, but located remotely.

The health marketplace engine 110 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 giving practitioners a social portal with which to communicate and market themselves with 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.

The request for quote engine 112, in the example shown in Figure 12 in which the request for quote engine 112 is part of the health marketplace system 110, allows a user of the health marketplace system to search for practitioners in their area that treat their conditions or practice in a desired specialty and request a quote for the service they need. Further details of the request for quote engine 112 are provided in U.S. Patent Application Serial No. 61/871,195, filed on August 28, 2013 and U.S. Patent Application Serial No. 14/328,591 filed

-18-

on July 10, 2014, the entirety of both of which are incorporated herein by reference. The predictive modeling engine 113 may generate healthcare service prices based on predictive modeling. Further details of the predictive modeling engine 113 are provided in U.S. Patent Application Serial No. 61/881,918, filed on September 24, 2013 and U.S. Patent Application
5 Serial No. 14/455,341 filed on August 8, 2014, the entirety of both of which are incorporated herein by reference.

The health marketplace system 110 may further have a health management engine 110a that may generate a healthstream for each member who is a user of the health system 100 and who has logged into the health system 100. The healthstream groups health related
10 information and events into a timeline that can be shared among a patient, healthcare provider(s), and approved family members/friends of each member/user of the system. The information for each user/member may be entered directly into the healthstream using the application 104. Alternatively or in addition, the information about the user/member may also be imported from entries made in other systems including Facebook, Twitter,
15 Foursquare and other web sites. Although the keeping of a detailed health journal requires a lot of discipline and work and busy lives don't often have time to keep up with it, a lot of important health information about the user/member may be recorded all the time in social networks and web sites. The healthstream may provide users an easy way to import and organize information that has already been recorded in these other systems so that it can be
20 visualized as a timeline of health information.

The health timeline (that is part of the healthstream) may always be available for review by approved members of the PokitDok community (healthcare providers, family members, friends) who have the required access rights and privileges that may be set up by the user. The system may allow a user/member to link an account in the health system, such
25 as the health system provided by PokitDok, with other accounts that have been established by the user/member using a known OAuth authorization flow (further details of which may be found at <http://en.wikipedia.org/wiki/OAuth> which is incorporated herein by reference.) The OAuth flow links the accounts of the user (social network and web accounts) so that the system is able to gather health related information. Once the accounts are linked, the
30 application 104 on the computing device may make API calls to each linked system to import

-19-

health related posts into their healthstream. In addition, users may be able to drag and drop entries to and from their healthstream to quickly manage what's permanently stored there.

In the health system, the timeline view generated for each user/member may support multiple cases in the timeline view. Each case may be like a directory on a filesystem where events may be stored together. For example, a mother may have cases defined for herself
5 and for her young children that are not yet old enough to manage their own healthstream. When that mother uses the PokitDok healthstream, her checkins at the doctor's office will be imported into her case folder. When she posts about her child running a fever on facebook, that event will be imported into her child's case folder that may be done by utilizing the APIs
10 provided by the various social networks and systems that can be linked to a PokitDok account. Each time a PokitDok user returns to the system, asynchronous tasks are queued to process the latest data from their linked accounts. The results of the above tasks may be presented in a list in the application. Each entry in that list may be manually added to a case using the drag and drop capabilities of HTML5. In addition, when possible, imported entries
15 may be automatically added to cases by analyzing the imported content for health related keywords.

The health system may support a variety of healthstream entry types. For example, when a checkin is added to the healthstream, a link to the location is stored along with geolocation information so it can be quickly displayed on a map view. As another example,
20 when entries with photos are added to the healthstream, a link to the original photo is stored along with cached versions of the photo at various resolutions for display in different contexts. The healthstream may also include entries about medications that may be added to the timeline that include information about when a medication was started/stopped along dosage details for the medication. In addition, doctor appointments may also be added to the
25 healthstream. Furthermore, healthstream video entries contain links to the original video so that it may be embedded in the health timeline along with the other information. In addition to the specific healthstream entry types above, a generic entry type may be available that supports miscellaneous file attachments. For example, a user may have a PDF of blood results that were emailed to them that they want to add to their healthstream so that it can be
30 shared with other health providers also on that case. The healthstream may allow a user to drag the PDF attachment from their email to the appropriate case in their timeline.

-20-

In the system 100, each of the entries for a user may be stored in the store 114 with a user's globally unique identifier and the identifiers of other PokitDok users 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
5 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.

Healthcare providers that are part of a healthstream case for a user can also add events to it. For example, a provider can review a user's healthstream when meeting with
10 them and recommend a service they've posted on the health marketplace as part of their treatment plan. This may add that service to the healthstream with a date/time stamp so the patient and other healthcare providers are all up-to-date with currently active treatments and medications. If multiple providers are participating on a case, email and SMS alerts can be triggered to alert them that new information is available for their review.

15 While the foregoing has been with reference to a particular embodiment of the invention, 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.

-21-

Claims:

1. A dynamic transactional data streaming system, comprising:
a data streaming system;
the data streaming system having a storage device that stores a plurality of rule sets,
wherein each rule set processes a transactional datastream having a particular set of
5 transactional data;
the data streaming system having a data stream examination component configured to
receive a transactional datastream having a plurality of segments wherein each segment has a
plurality of data elements, the data stream examination component also configured to identify
each of the segments in the transactional datastream; and
10 the data streaming system having a data stream segment processing component
configured to receive each segment of the transactional datastream identified by the data
stream examination component, configured to select a rule set from the plurality of rule sets
based on the data elements in each segment of the transactional datastream and configured to
generate a set of translated data based on the selected rule set and the plurality of segments
15 and the plurality of data elements of the transactional datastream.
2. The system of claim 1, wherein the data stream segment processing component
is configured to parse each data element in each segment based on a parse expression to
generate a parsed data element and configured to apply the selected rule set to the parsed data
element.
- 20 3. The system of claim 2, wherein the set of translated data is stored in a graph.
4. The system of claim 1, wherein the data stream examination component that is
configured to identify each of the segments in the transactional datastream is further
configured to identify an element delimiter that separates each data element and a segment
terminator that separates each segment in the transactional datastream.
- 25 5. The system of claim 1 further comprising a data stream output component
configured to load a stream generation rule based on the set of translated data and configured
to generate an output datastream using the stream generation rule and the set of translated
data.

-22-

6. The system of claim 5, wherein the data stream output component loads the stream generation rules based on a set of translated data stored in graph.

7. The system of claim 1, wherein the transactional datastream is a healthcare transactional datastream.

5 8. The system of claim 7, wherein the healthcare transactional datastream uses an ASC X12 5010 standard delimited format.

9. The system of claim 1, wherein the data stream system further comprises a data processing component having a plurality of processing channels wherein each processing channel processes a segment of the transactional data so that the data segments of the
10 transactional datastream are processed in parallel.

10. The system of claim 9, wherein each processing channel has a message queue so that each processing channel communicates with the other processing channels.

11. The system of claim 9, wherein each processing channel is a thread executed by a processor.

15 12. The system of claim 9, wherein each processing channel has a rule set processor that performs the operations of the data stream segment processing component for each segment assigned to the processing channel.

13. A dynamic transactional data streaming method, comprising:
storing, in a storage device, a plurality of rule sets, wherein each rule set processes a
20 transactional datastream having a particular set of transactional data;

receiving, by a computer system coupled to the storage device, a transactional datastream having a plurality of segments wherein each segment has a plurality of data elements;

25 identifying, by the computer system, each of the segments in the transactional datastream;

selecting, by the computer system, a rule set from the plurality of rule sets based on the data elements in each segment of the transactional datastream; and

-23-

generating, by the computer system, a set of translated data based on the selected rule set and the plurality of segments and the plurality of data elements of the transactional datastream.

14. The method of claim 13 further comprising parsing, by the computer system, each data element in each segment based on a parse expression to generate a parsed data element and applying, by the computer system, the selected rule set to the parsed data element.

15. The method of claim 14 further comprising storing the set of translated data in a graph.

16. The method of claim 13, wherein identifying each segment further comprises identifying, by the computer system, an element delimiter that separates each data element and a segment terminator that separates each segment in the transactional datastream.

17. The method of claim 13 further comprising loading, by the computer system, a stream generation rule based on the set of translated data and generating, by the computer system, an output datastream using the stream generation rule and the set of translated data.

18. The method of claim 17, wherein loading the stream generation rules further comprising loading, by the computer system, the stream generation rule based on a set of translated data stored in graph.

19. The method of claim 13, wherein the transactional datastream is a healthcare transactional datastream.

20. The method of claim 19, wherein the healthcare transactional datastream uses an ASC X12 5010 standard delimited format.

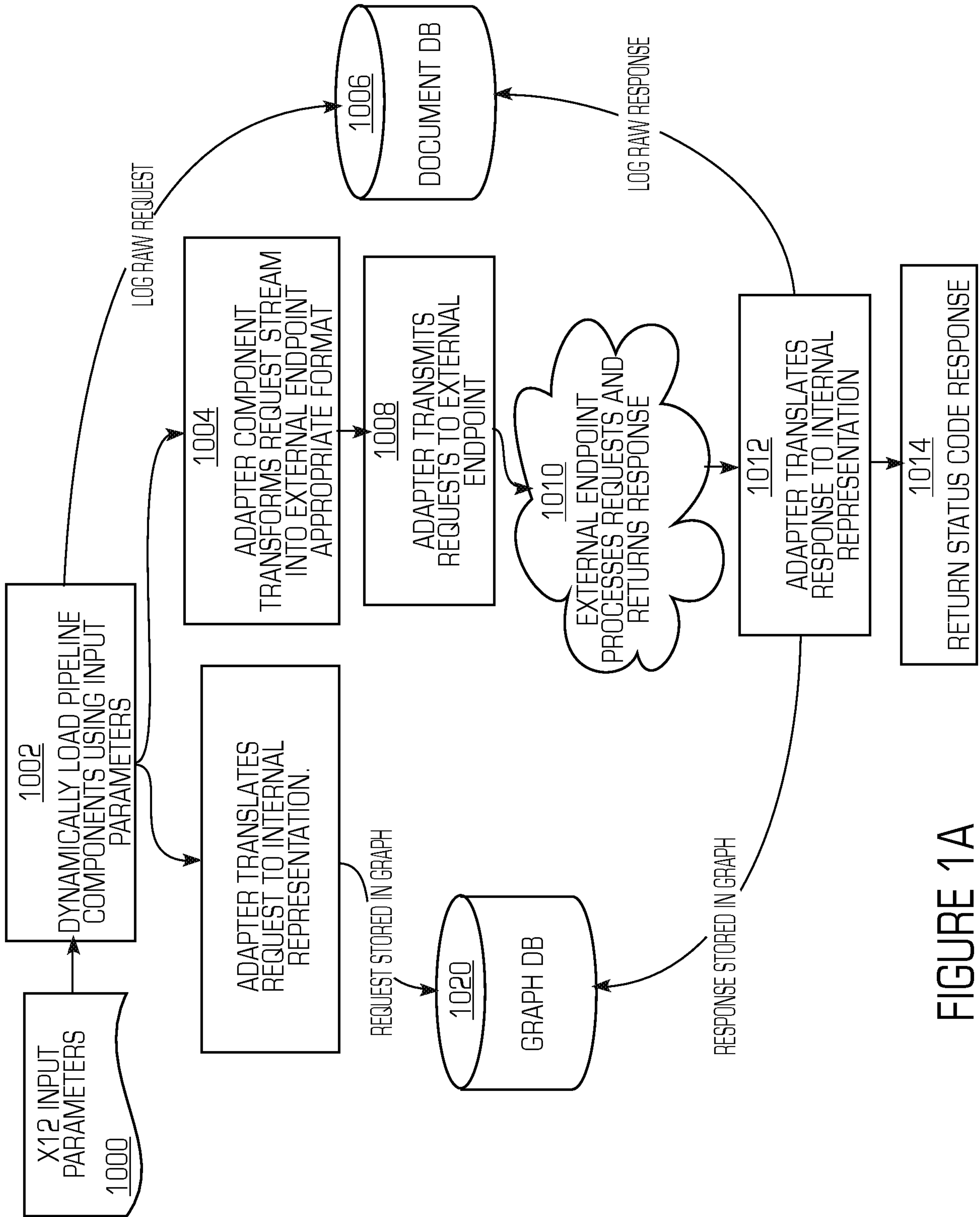


FIGURE 1A

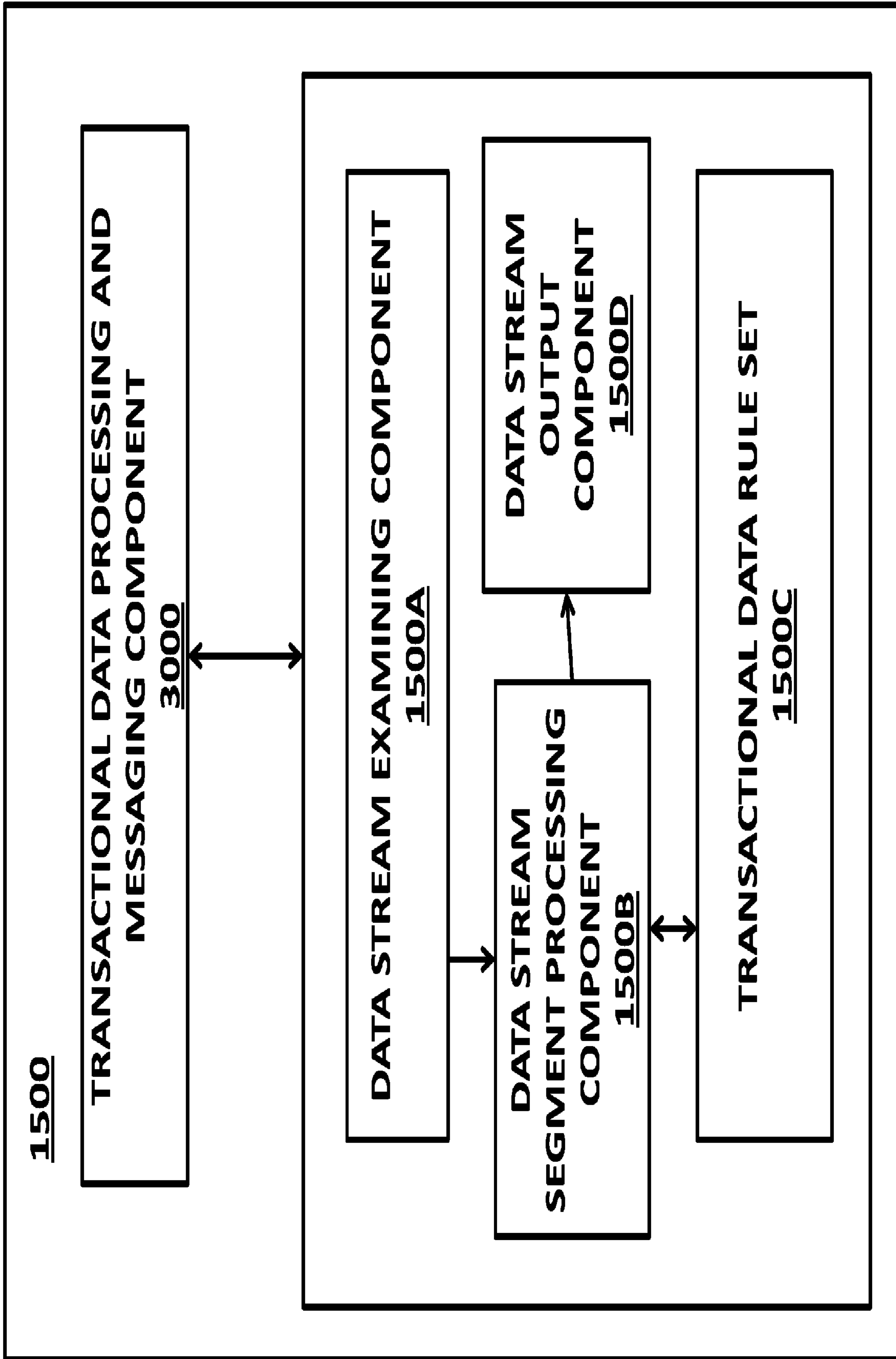
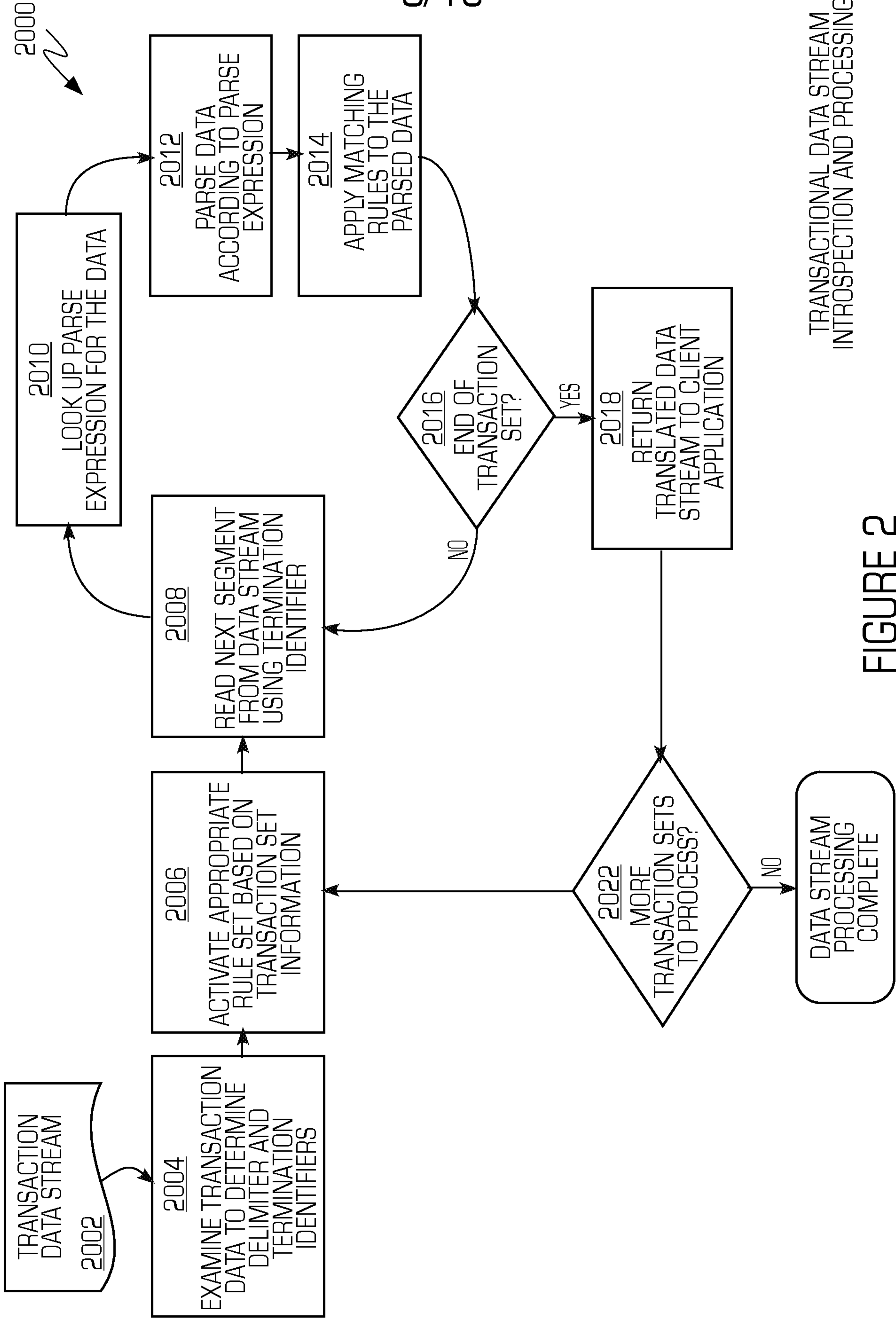


FIGURE 1B



TRANSACTIONAL DATA STREAM
INTROSPECTION AND PROCESSING

FIGURE 2

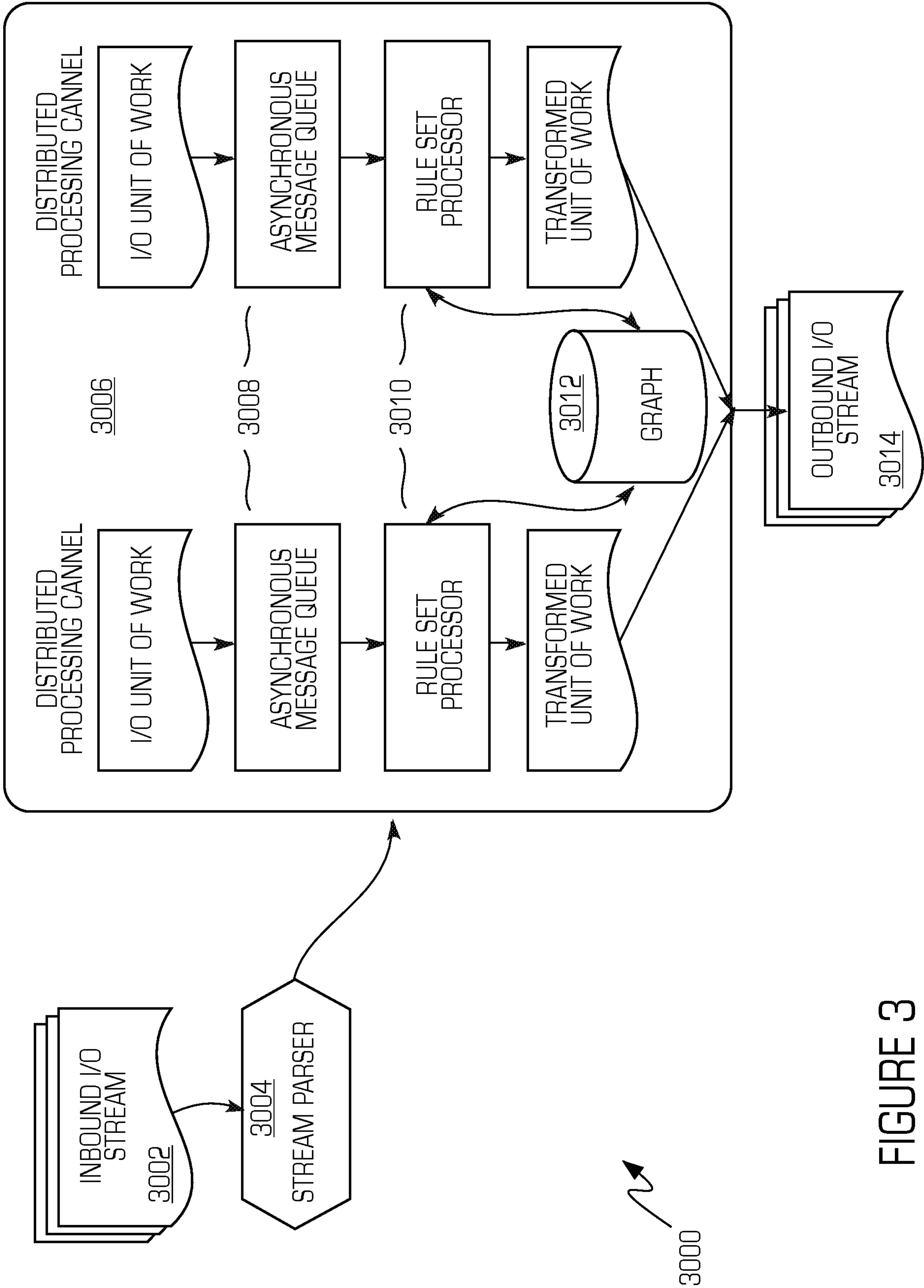


FIGURE 3

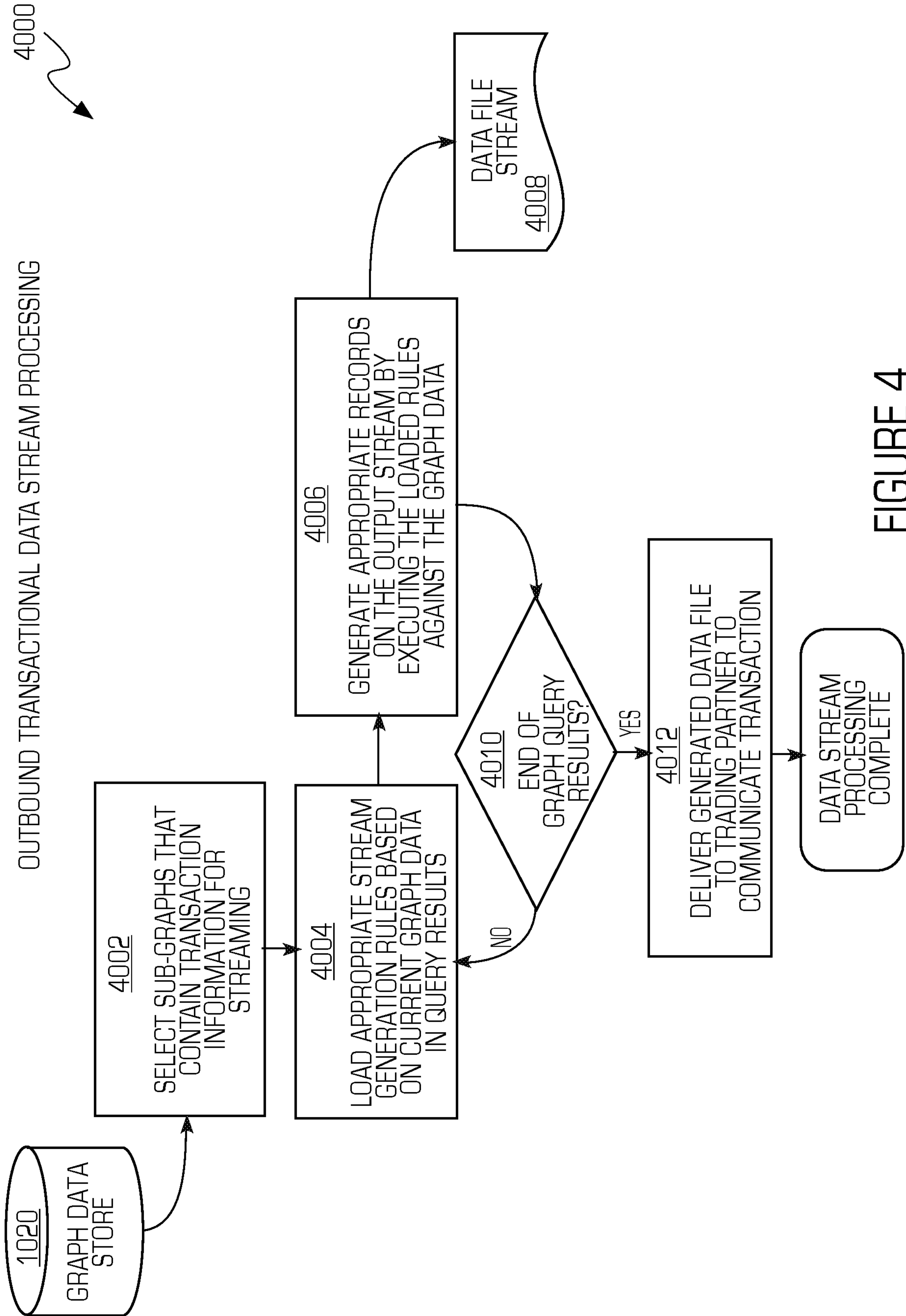


FIGURE 4

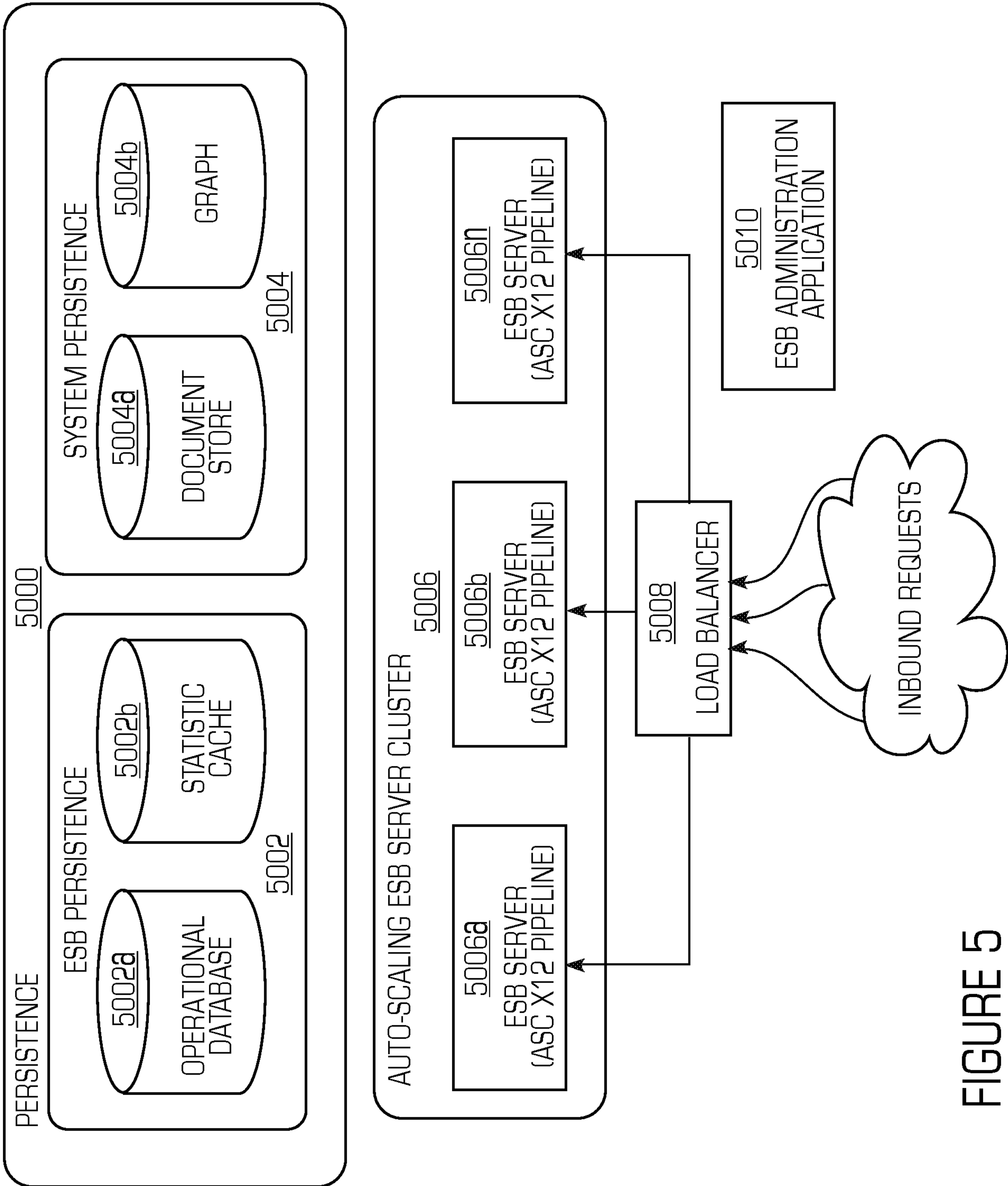


FIGURE 5

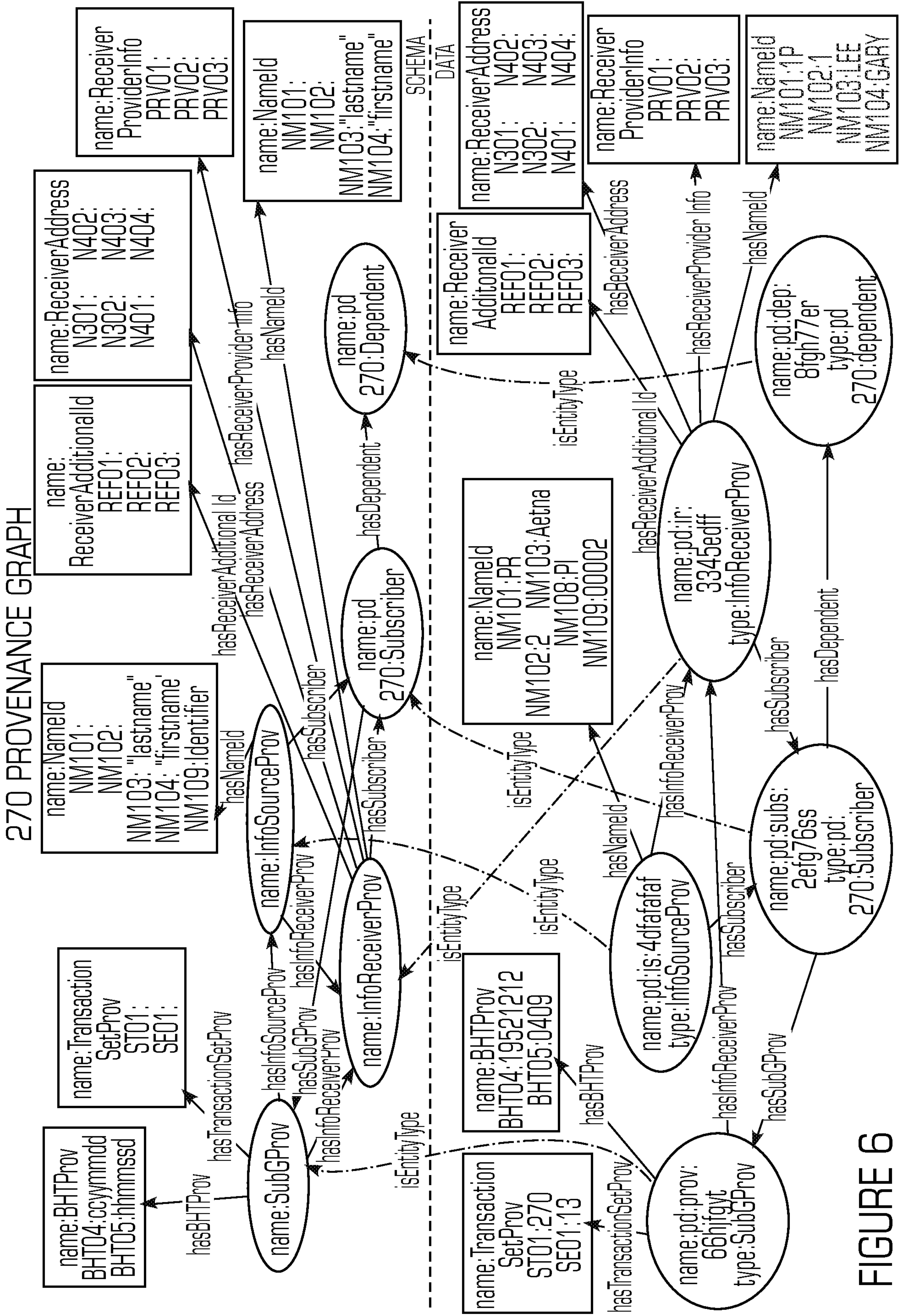


FIGURE 6

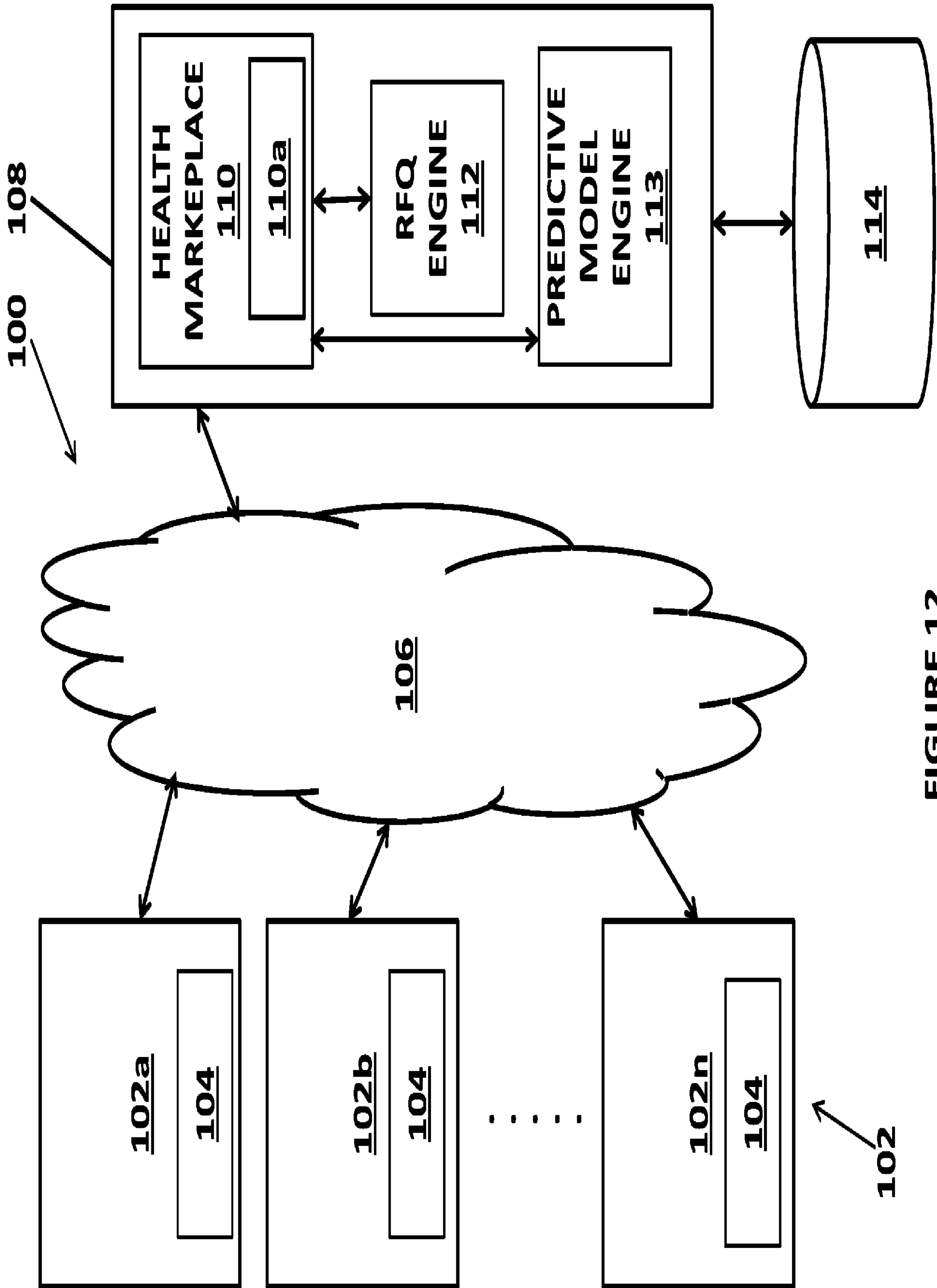


FIGURE 12

14/15

ISA*00* *00* *ZZ* 1234567890 *ZZ* 654456654
 *130810*0409*^*00501*10000001*0*P*~
 GS*HS* 1234567890* 654456654*20130810*0409*1*X*005010X279A1~
 ST*270*0001*005010X279A1~
 BHT*0022*13*137083739083716126837*20130810*0409~
 HL*1**20*1~
 NM1*PR*2*MOCK PAYER FOR TESTING*****PI*MOCKPAYER~
 HL*2*1*21*1~
 NM1*1P*1*LEE*GARY*****XX*1497834352~
 HL*3*2*22*0~
 TRN*1*1*1453915417~
 NM1*IL*1*BUSHHOG*BILLYBOB*****MI*W199138913~
 DMG*D8*19770101~
 DTP*291*D8*20130810~
 EQ*30~
 SE*13*0001~
 GE*1*1~
 GS*BE*1234567890*654456654*20130809*1150*2*X*005010X220A1~
 ST*834*0002*005010X220A1~
 BGN*00*12456*20130809*1150*ET***2~
 N1*P5*KOELPIN AND SONS*FI*999888777~
 N1*IN*REICHERT-SCHROEDER*FI*654456654~
 INS*Y*18*021*20*A***FT~
 REF*0F*123456789~
 REF*1L*123456001~
 DTP*356*D8*20130809~
 NM1*IL*1*BROWN*ADELIA****34*123456789~
 PER*IP**HP*951-768-7732 x17235*WP*728.783.0678 x2392***~
 N3*9702 NICOLE MEADOWS*SUIITE 370~
 N4*EAST WYMANMOUTH*LA*67175*****~
 DMG*D8*19870914*F~
 HD*021**HLT~
 DTP*348*D8*20130809~
 COB*P*890111*5~
 HD*021**DEN~
 DTP*348*D8*20130809~
 HD*021**VIS~
 DTP*348*D8*20130809~
 SE*21*0002~
 GE*1**2~
 IEA*1*100000001~

FIGURE 13

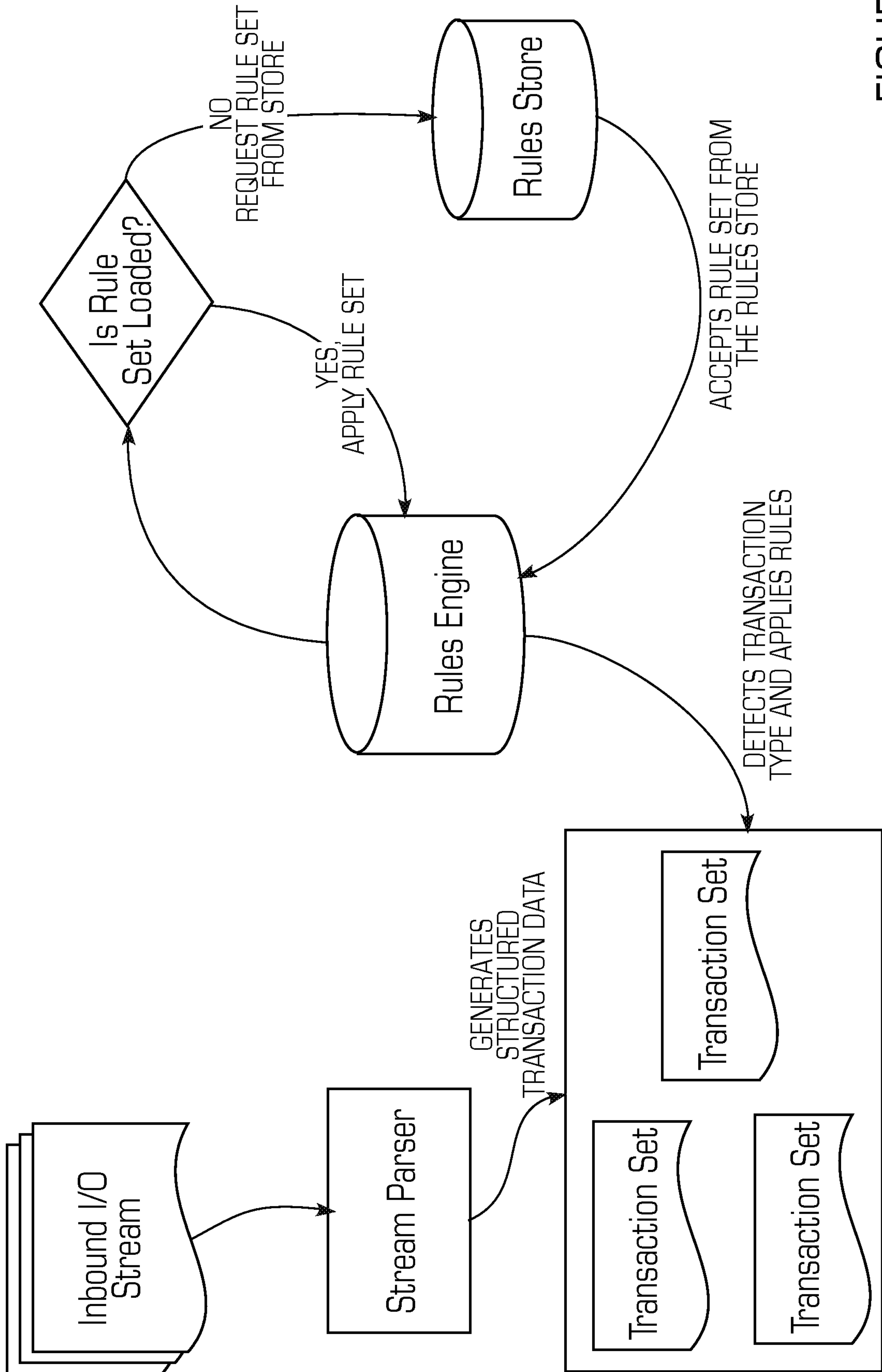


FIGURE 14

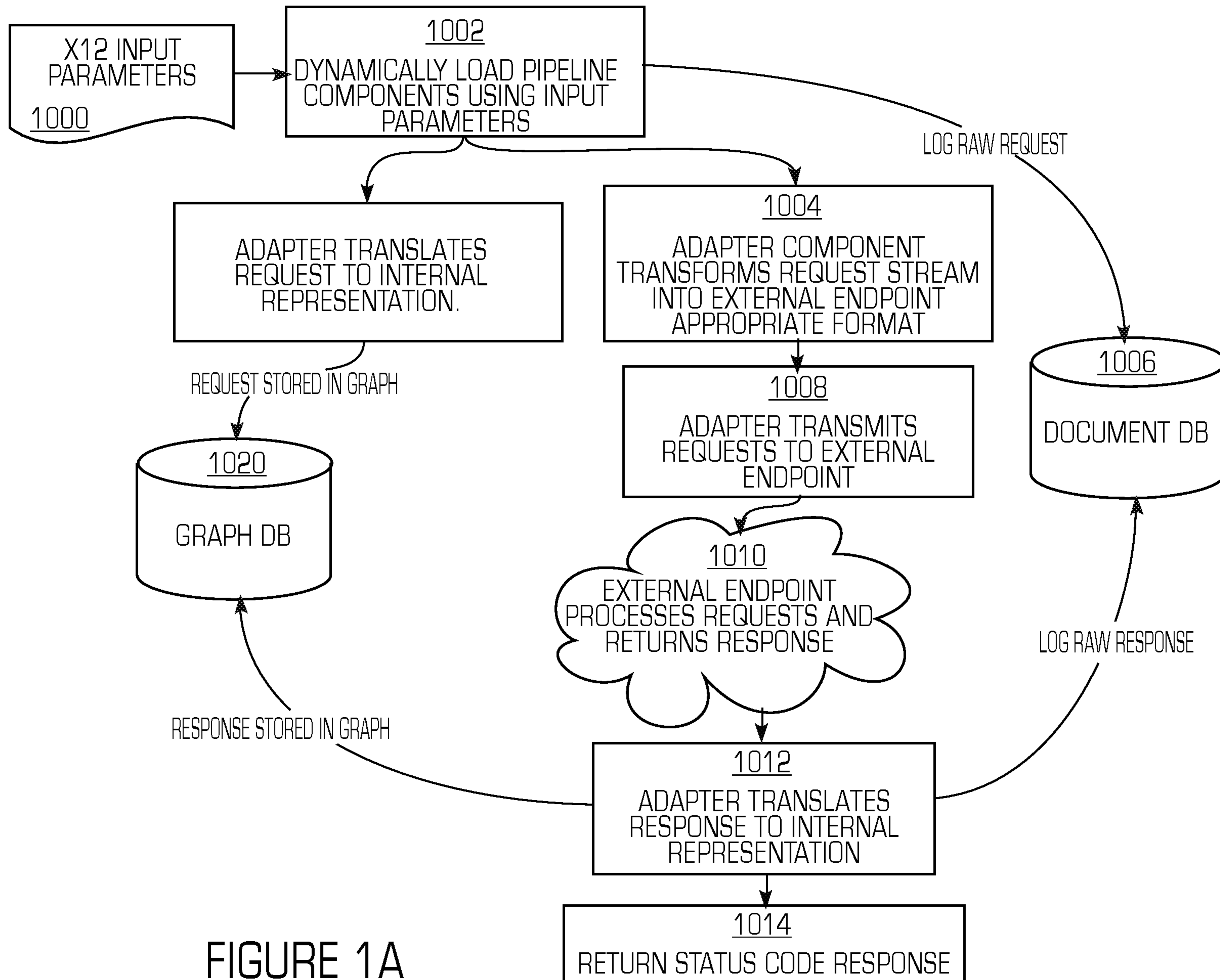


FIGURE 1A