

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
27 February 2003 (27.02.2003)

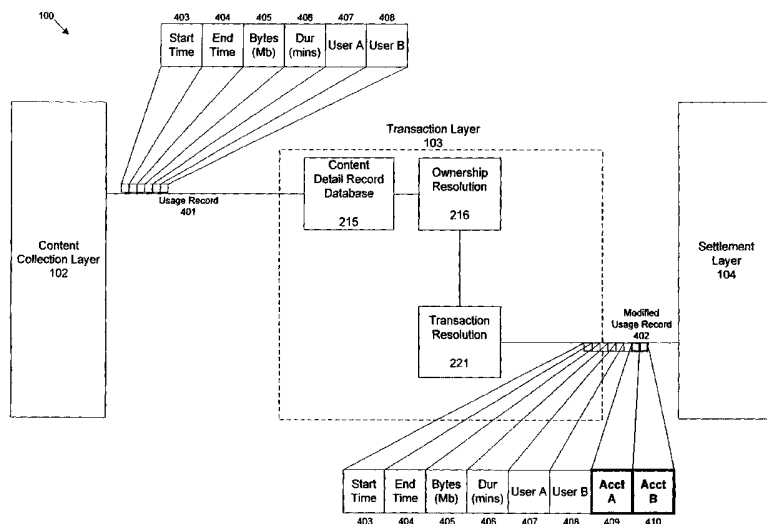
PCT

(10) International Publication Number
WO 03/017064 A2

- (51) International Patent Classification⁷: **G06F** (74) Agents: **DUNNAM, Michael, P.** et al.; Woodcock Washburn LLP, 46th floor, One Liberty Place, Philadelphia, PA 19103 (US).
- (21) International Application Number: PCT/US02/26594
- (22) International Filing Date: 20 August 2002 (20.08.2002) (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/934,285 21 August 2001 (21.08.2001) US
- (71) Applicant (*for all designated States except US*): **APOGEE NETWORKS** [US/US]; 1455 Broad Street, 4th Floor, Bloomfield, NJ 07003 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (*for US only*): **DESHMUKH, Abhi** [IN/US]; 253A, Kinderkamack Road, River Edge, NJ 07661 (US). **HERNANDEZ, Milton** [DM/US]; Apt. D, 1 Cononial Court, River Edge, NJ 07661 (US). **PITCHAIKANI, Balaji** [IN/US]; 13048 Cumbra Vista Court, Los Altos Hills, CA 94022 (US).
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:**
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: CONTENT TRANSACTION RESOLUTION



(57) Abstract: The invention describes a method, system, and computer-readable medium having computer-executable instructions for identifying a relationship among the parties to a transaction. The inventive method includes receiving data related to an event, where the data includes one or more data elements. The inventive method also includes selecting the data elements that identify the relationship among the parties to the transaction, determining whether additional data elements are required to identify the relationship among the parties to the transaction, and identifying the additional data elements. The event may be an exchange of data between the parties over a communication network, and the data related to the event may be stored in a usage record. The additional data element may identify the relationship among the parties, called a contract.



WO 03/017064 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

CONTENT TRANSACTION RESOLUTION

Technical Field

The invention relates generally to the processing of data, and more particularly to
5 identifying the relationships among parties to a transaction.

Background of the Invention

Recently, the collection and processing of data transmitted over communication
networks, like the Internet, has moved to the forefront of business objectives. In fact,
10 with the advent of the Internet, new revenue generating business models have been
created to account for the consumption of content received from a data network (*i.e.*,
content-based billing). For example, content distributors, application service providers
(ASPs), Internet service providers (ISPs), and wireless Internet providers have realized
new opportunities based on the value of the content and services that they deliver. As a
15 result of this content-and-services-billing initiative, it has become increasingly important
to resolve intelligently and flexibly the corresponding transactions according to the
business needs of the customer.

Often, even in traditional commerce streams, there a many different participants
between the maker and the consumer of the goods. Some of these so-called “middlemen”
20 simply may facilitate access to the good, while others may add useful characteristics to
the good. This is especially true in a distributed network environment, like the Internet,

that impose additional burdens on capturing the transaction process because of the unlimited number of data sources and the correspondingly unlimited number of data types. As a result, adequately describing the transacting of the goods and services in such an environment, or even across any of the traditional environments requires a technique
5 that is capable of understanding and processing the participants and the various types of data.

To date, resolving the characteristics of a transaction has been accomplished over data-specific environments using certain application-specific (*i.e.*, “non-generic” or custom) methods. For example, a typical telephone bill captures the characteristics of a
10 voice-based telecommunications transaction. In particular, the participants may be identified as the party initiating the call (*i.e.*, the “calling party” telephone number), the party receiving the call (*i.e.*, the “called party” telephone number), and one or more parties connecting the communication (*i.e.*, the “local” or “long distance carrier”). This simple example adequately captures all of the necessary aspects of the telephone call to
15 properly account for the transaction by requiring payment from certain parties (*e.g.*, the calling and/or the called party) to certain other parties (*e.g.*, the local and/or long distance carrier). While these “hard-coded” transaction resolution techniques are sufficient to accommodate only specific environments (*e.g.*, telecommunications system) involving well-defined participants, they are correspondingly too rigid to efficiently satisfy the
20 ever-changing face of a communication network like the Internet. Also, these “hard-

coded” techniques are too specific to operate across the many and varied transaction environments and domains that exist throughout the business world.

Therefore, there exists a need to provide a flexible technique that can capture and resolve transactions of any type, for any business domain and involving any number and
5 kind of participants without requiring modification of the technique for each type. Such a technique would serve to accommodate multi-faceted networked environments like the Internet, as well as to provide a single solution that would apply to any business environment.

10

Summary of the Invention

The invention describes a method, system, and computer-readable medium having computer-executable instructions for identifying a relationship among the parties to a transaction. The inventive method includes receiving data related to an event, where the data includes one or more data elements. The inventive method also includes selecting
15 the data elements that identify the relationship among the parties to the transaction, determining whether additional data elements are required to identify the relationship among the parties to the transaction, and identifying the additional data elements. The event may be an exchange of data between the parties over a communication network, and the data related to the event may be stored in a usage record. The additional data
20 element may identify the relationship among the parties, called a contract. Determining whether additional elements are required may be a function of the received data, and/or a

function of the requirements of a customer settlement system. The customer settlement system may be a data security system, a network capacity planning system, and/or a content monitoring system or a billing application, for example. The identifying of additional data elements or accountables may be initiated by a building block known as a primitive that consists of multiple specific methods including a database lookup, a direct value assignment, a regular expression parsing or match, a CDR database lookup, hard-coded, network address, area code, substring manipulation or a custom code execution. All or some of the inventive method steps may be conducted on a computer-readable medium using computer-executable instructions.

10 The inventive system for determining parties to a transaction includes an instrumentation layer, a content collection layer in communication with the instrumentation layer, and a transaction layer in communication with the content collection layer. The instrumentation layer provides raw data, while the content collection layer generically processes the raw data, and the transaction layer identifies the relationships among the parties to the transaction. Also, the content collection layer may provide the raw data to the transaction layer in a usage record, and the transaction layer may modify the usage record as a function of identifying the parties to the transaction. The inventive system may further include a settlement layer in communication with the transaction layer, wherein the settlement layer rates the modified usage record. The transaction layer may include a database for storing the data processed by the content collection layer.

Brief Description of the Drawings

Other features of the invention are further apparent from the following detailed description of the embodiments of the invention taken in conjunction with the accompanying drawings, of which:

Figure 1 is a block diagram of a system for analyzing content transmitted over a communication network, according to the invention;

Figure 2 is a block diagram further describing the components of the system described in Figure 1;

Figures 3A and 3B provide a flow diagram further detailing the operation the system described in Figure 1;

Figure 4 provides a block diagram of the system described in Figure 1, providing greater detail of a configuration and operation of a transaction layer;

Figures 5A and 5B provide a flow diagram of a method for identifying the characteristics of a transaction, according to the invention;

Figure 6 is a block diagram further detailing the operation and configuration of the transaction layer, according to the invention; and

Figure 7 is a flow diagram that details the operation and configuration of the transaction layer of transaction layer, according to the invention.

Detailed Description of the Invention

System Overview

Figure 1 is a block diagram of a system 100 for analyzing content transmitted over a communication network. Although the following description will be discussed in the context of collecting, processing and billing for data transmitted over the Internet, it should be appreciated that the invention is no so limited. In fact, the invention may be applied to any type of network, including a private local area network, for example. Also, the invention may be used for purposes other than billing for the usage of content. For example, the invention may be used to analyze the type of data transmitted over a particular network, or to determine the routing patterns of data on a network. Furthermore, the invention may be used to facilitate the intelligent collection and aggregation of data relevant to a particular industry. In addition, the invention may be used to track specific Internet protocol (IP) network resources and to detect fraud, for example. It should also be appreciated that the invention may contemplate non-network and non-computer related business domains like commodity trading (*e.g.*, grains, crude oil) where multiple parties are involved. Therefore, although the invention is often described in the context of the Internet, it should be appreciated that the invention may equally applied to traditional business environments.

In addition, it should be appreciated that the term “content” may be defined as data that is transmitted over the network. In the context of the Internet, content may include .mp3 files, hypertext markup language (html) pages, videoconferencing data, and

streaming audio, for example. The terms “content provider” and “customer” will be used throughout the description as well. Content provider may refer to the primary creator or provider of the content, while customer is the primary recipient of the content. Both the producer and customer may be a human or a computer-based system.

5 As shown in Figure 1, an instrumentation layer 101 provides raw data to a content collection layer 102. Instrumentation layer 101 may consist of various data sources, for example, network routers. The network routers may provide information regarding the various types of routed data, including for example, data format, originating Internet protocol (IP) address, and destination IP address. One example of such information is
10 Cisco System's NetFlow™.

Content collection layer 102 collects information about the delivery of the content, as well as the substance of the content itself. Content collection layer 102 also may sort, filter, aggregate, aggregate, and store the content according to the particular needs of the end user. In effect, content collection layer is responsible for extracting
15 meaningful information about IP traffic, and so it is provided with an understanding of the data sources in instrumentation layer 101. Content collection layer 102 also may transform the data from the plurality of sources in instrumentation layer 101 into standard formats for use in a transaction layer 103.

Content collection layer 102 is in communication with transaction layer 103.
20 Generally, content collection layer 102 reports to transaction layer 103 that a relevant communication event has occurred and should be considered by the remainder of system

100. A communication event may be defined as any transfer of data between systems. Transaction layer 103 captures the predetermined agreements among all of the parties involved in system 100 regarding the value of the transferred content, as well as the value added by each of the individual parties in transferring such content. Therefore, 5 transaction layer 103 is charged with understanding the nature of the parties, as well as the understanding the actions that one or more parties perform and the influence of such action on the respective parties.

Transaction layer 103 is in communication with a settlement layer 104. Settlement layer 104 captures the operations that are necessary to understand the 10 significance of the transaction defined by transaction layer 103. For example, settlement layer 104 may rate a particular transaction by assigning a monetary value to the transaction. Settlement layer 104 also may divide the burdens and benefits of the monetary value among the relevant parties. In this way, settlement layer 104 ensures that certain parties receive a particular portion of the payment made by the other parties. 15 Settlement layer 104 also may be responsible for delivering this burden and benefit information to the appropriate parties with the appropriate identifiers (*e.g.*, account numbers).

Figure 2 is a block diagram further describing the components of system 100. As shown in Figure 2, instrumentation layer 101 includes data sources 201-203 that each 20 provide raw data 204-206, respectively, to collection layer 102. As discussed, data sources 201-203 may include various internetworking devices like routers, bridges, and

network switches. Data sources 201-203 provide raw data to an input data adapter 207. Accordingly, input data adapter 207 understands the operation of and the data provided by data sources 201-203. Although one input data adapter is shown in Figure 2, it should be appreciated that more than one input data adapter may be used in system 100. For
5 example, each data source may have a dedicated input data adapter.

Input data adapter 207 understands the incoming data and extracts the data into the appropriate fields. This field-delimited data, called flow objects 208, are sets of attributes. The attributes may be any characteristics that are provided by, or can be derived from, raw data 204-206 provided by data sources 201-203, respectively. For
10 example, flow objects 208 may include a set of attributes describing the source and destination, including source IP address, destination IP address, source interface, and destination interface. Because input data adapter 207 is charged with understanding raw data 204-206 from data sources 201-203, as well as the required flow objects 208 of system 100, it is capable of transforming the raw data to the flow objects, where the flow
15 objects may all be of a standard format.

Input data adapter 207 provides flow objects 208 to a content data language 209. Content data language 209 may transform the attributes in flow objects 208 into other attributes that are desired by a particular customer. For example, content data language 209 may derive a network identifier attribute that is not readily available from a data
20 source, from a source address attribute and a destination address attribute that are provided by flow object 208 attributes from input data adapter 207. This derivation may

be based on a customer's desire to determine which network conveyed the transaction between the source and the destination. Therefore, following this example, content data language 209 will know to extract the source address attribute and the destination address attribute from flow objects 208.

5 Content data language 209 may perform other functions as well. For example, content data language 209 may perform a generic lookup function 219 that is built into content data language 209. Generally, generic lookup 219 describes a technique for mapping any number of attributes to any number of other derived attributes. For example, generic lookup 219 may be used to map a unique resource locator (URL) attribute to a particular content-type attribute.

10 Content data language 209 also is in communication with a correlator 211. Generally, correlator 211 connects the many daily network content events from various network devices, like routers for example. Often, the connected data may come from distinctly different data sources at distinctly unrelated times. Correlator 211 allows this data to be intelligently connected to each other, regardless of how different the sources or of how disparate the time received. For example, a Netflow™ enabled router and a Radius™ enabled network access switch may each provide certain data that is relevant to one particular transaction. However, because portions of the data come from different devices, the data may arrive at system 100 at different times, and in different formats.

15 Also, because each provides data that is necessary to complete one transaction, the data from each cannot be considered separately. Correlator 211 allows this data to be

20

intelligently grouped regardless of the format of the data or of the time the data pieces are received.

Furthermore, correlator 209 may rearrange the order of the received flow objects 208 to suit the needs of the remainder of system 100. By performing such correlation
5 without having to first store all of the data on a disk (*e.g.*, a database), significantly faster processing is achieved. Correlator 209 may perform this correlation in real-time, for example.

Although system 100 has been described using content data language 209 and correlator 211, it should be appreciated that flow objects 208 may proceed directly to a
10 filter 212, if no correlation is required and if no attribute derivation is required, for example. Filter 212 analyzes flow objects 208 to ensure that the provided attributes are desired by system 100. If flow objects 208 are not needed (*i.e.*, a mismatch), filter 212 may prevent flow objects 208 from passing to an aggregator 213. Also, although filter 212 has been shown as a separate device in system 100, it should be appreciated that the
15 functionality of filter 212 may be incorporated into aggregator 213.

Filter 212 passes the matching flow objects to aggregator 213. Aggregator 213 may provide additional filtering and classification of the multitude of daily network content transactions, based on user criteria. Aggregator 213 may perform such filtering and classification in real-time. Aggregator 213 will be discussed in greater detail with
20 reference to Figures 4-7. Aggregator 213 provides the filtered and classified information to an output data adapter 214. Output data adapter 214 may convert the data from

aggregator 213 into one or more content detail records for storage in CDR database 215. Therefore, CDR database 215 stores a complete description of a content event.

Transaction layer 103 includes CDR database 215, an ownership resolution component 216, and a transaction resolution component 221. CDR database 215 passes
5 the CDR onto ownership resolution component 216. A proper accounting of a communication event, for any purpose including billing, requires both identifying the participants and defining the relationships among those participants. These tasks are accomplished by ownership resolution component 216 and transaction resolution component 221.

10 Ownership resolution component 216 serves to define the direct and indirect participants of the communication event. In particular, because raw data 204-206 may not provide sufficient "ownership" data relating to the parties involved in the content transaction, such "ownership" data must be provided by system 100 to properly account for the entire transaction. Ownership resolution component 216 operates to provide such
15 ownership-based data to system 100.

Ownership resolution component 216 is in communication with transaction resolution component 221. Transaction resolution component 221 serves to capture the predetermined relationships and/or agreements among the parties defined by ownership resolution component 216 regarding the value of the transferred content, as well as the
20 value added by each of the individual parties in transferring such content. Such predetermined relationships may be previously agreed upon by the participants, or may

be created and agreed upon with the implementation of system 100. Therefore, transaction component 103 understands the nature of the parties, the actions that one or more parties perform, and the influence of such action on the respective parties.

Ownership resolution component 216 and transaction resolution component 221 will be
5 discussed in greater detail.

Transaction component 216 provides the transaction information to a rating component 217. Rating component 217 provides a weight or significance (*e.g.*, a price) to the transaction, so as to provide a tangible value to the transaction. Rating component 217 may make this determination based on various metrics including the type of the
10 content, the quantity of content consumed, the place and time delivered, or the quality of the content, for example. Therefore, rating component 217 allows system 100 to provide some contextual value, indicating the significance or relevance that certain content or information has to the individual customer.

Rating component 217 provides the rated transaction to a presentment component
15 218. Presentment component 218 may provide the capability for a customer 220 to view their real-time billing information, for example, over the network. Presentment component 218 also may attach relevant identifiers to the bill (*e.g.*, account numbers, etc.).

Figures 3A and 3B provide a flow diagram further detailing the operation 300 of
20 system 100. As shown in Figure 3A, in step 301, raw data 204-206 is received from data sources 201-203. In step 302, input data adapter 207 converts raw data 204-206 to flow

objects 208, where flow objects 208 are sets of attributes, determined from raw data 204-206. In step 303, it is determined whether there is a need to derive new attributes from the existing attributes found in flow objects 208. If there is such a need, in step 304, CDL 209 is used to derive new attributes from existing attributes. Also, as discussed
5 above, attributes may be correlated by correlator 211.

In step 305, flow objects 208 are filtered by filter 212. In step 306, the matching flow objects (*i.e.*, those passed by filter 212) are further filtered and classified by aggregator 213 (as discussed more fully with reference to Figures 4-7). In step 307, output data adapter 214 converts the data aggregated by aggregator 213 to a format
10 compatible with transaction layer 103.

As shown in Figure 3B, in step 308, output adapter 214 may format the aggregated data into one or more content data records for storage in CDR database 215. In step 309, ownership resolution component 216 identifies the parties involved in the transaction. In step 310, transaction resolution component 221 captures the
15 predetermined agreements among the parties, as well as the value added by each of the individual parties. In step 311, the CDR is rated based on predetermined metrics (*e.g.*, time of transmission, quality of content). In step 312, a bill is presented to the customer.

Content Transaction Layer

A transaction, including a business transaction for example, may be any activity between two or more participants that affects certain parties. The affected parties may be considered the participants to the transaction, although parties and participants may be mutually exclusive entities. It should be appreciated that the terms “parties” and “owners” may be used interchangeably throughout. Also, a transaction may include one or more events, where an event may be some action (*e.g.*, the transfer of a service or good) that the participants conduct with each other. Such events may be the exchange of data over a communication network, for example. Also, it should be appreciated that the same event or action may take place in any direction (*e.g.*, from party A to party B or from party B to party A). The “subject” of the action is the party who performs the activity, while the “object” of the action is the party who is affected or influenced in any way by the action.

Transaction layer 103 is responsible for capturing the characteristics of the transaction, including the parties, participants, subject, object, event/action type, direction of the action, for example. The scope of the characteristics captured by transaction layer 103 will vary with the nature of instrumentation layer 101, the capability of content collection layer 102, and the requirements of settlement layer 104. Generally, transaction layer 103 captures the predetermined agreements among all of the parties involved in system 100 regarding the value of the transferred content, as well as the value added by each of the individual parties in transferring such content. Therefore, transaction layer

103 is charged with understanding the nature of the parties, as well as the understanding the actions that one or more parties perform and the influence of such action on the respective parties.

Transaction layer 103 includes CDR database 215, an ownership resolution component 216, and a transaction resolution component 221. CDR database 215 is a data store that holds each CDR for further processing by ownership resolution component 216 and transaction resolution component 221. In the context of a billable electronic business transaction, a proper accounting of a communication event, for any purpose including billing, requires both identifying the participants and defining the relationships among those participants. As discussed, ownership resolution component 216 serves to define the participants of the communication event, while transaction resolution component 221 serves to capture the predetermined relationships and/or agreements among the parties defined by ownership resolution component 216.

Transaction layer 103 is generic in that it operates without requiring an understanding of instrumentation layer 101 and content collection layer 102. Instead, transaction layer 103 takes the relevant characteristics of the transaction provided by instrumentation layer 101 and content collection layer 102, and selects and generates those characteristics required by settlement layer 104.

Although the operation of transaction layer 103 will be described in the context of an electronic event over a communication network for the purposes of billing for the event, it should be appreciated that the invention is not so limited, but is so described the

purpose of clarity. For example, it should be appreciated that the invention may include functions other than billing, including security, network capacity planning, and content monitoring, for example. Also, the generic techniques described may be applied beyond the context of an electronic event over a communication network.

5 Figure 4 provides a block diagram of system 100 providing greater detail of the configuration and operation of transaction layer 103. As shown in Figure 4, content collection layer 102 provides a usage record 401 to transaction layer 103. A series of fields 403-408 comprise usage record 401. Although fields 403-408 will be identified by certain labels for the purpose of clarity, it should be appreciated that fields 403-408 may
10 identify any of the many possible characteristics of the transaction.

 In the example provided by Figure 4, fields 403 and 404 identify the parties or participants to the transaction. In particular, field 403 identifies "User A" and field 404 identifies "User B." User A may be the subject party and User B may be the object party. For example, in the context of a telecommunication event, User A may be the "calling"
15 party and User B may be the "called" party. Field 405 and field 406 identify a measurable quantity of the event. In particular, following the example of the telecommunications event, field 405 identifies a duration (in minutes) of the event and field 406 identifies the quantity of data exchanged during the event (in megabytes). Field 407 and field 408 may identify a "start time" and an "end time," respectively, of the
20 communication event.

Although fields 403-408 provide just one example, in any case, it should be appreciated that user record 401 will include certain characteristics provided by instrumentation layer 101 and/or derived by content collection layer 102. These characteristics typically are provided to transaction layer 103 because they have some particular significance to identifying the characteristics necessary to allow settlement layer 104 to provide some accounting of the transaction. Therefore, usage record 401 may not necessarily include all of the characteristics provide by instrumentation layer 101 and/or derived by content collection layer 102, because certain characteristics may not relevant to the settlement of the transaction (*e.g.*, network type). Yet, usage record 401 will include those available characteristics that pertain to permitting settlement layer 104 to properly account for the transaction.

Once usage record 401 passes through transaction layer 103, a modified usage record 402 may be provided to settlement layer 104. Modified usage record 402 may be modified by transaction layer 103 to include additional details of the transaction event required by settlement layer 104 and/or required by the customer's system to which the usage record applies (*e.g.*, a usage billing system). As shown for modified usage record 402, an account A field 409 and an account B field 410 may be added to further identify the transaction. In one example, account A field 409 may include an alphanumeric designator that identifies the account number that has been assigned by the customer's billing system for User A. In the context of the telecommunications system, account A field 409 may hold User A's ten-digit home telephone number, for example.

In any case, the components of transaction layer 103 operate to identify these additional fields, based on the requirements of the customer's system and of settlement layer 104. In particular, ownership resolution component 216 operates to ensure that the necessary participants or parties to the transaction are identified, while transaction
5 resolution component 221 operates to ensure that the relationships among the parties or participants are identified.

Identification of the relevant entities and their relationships may be accomplished using "lookups" in data stores located within ownership resolution component 216, user-defined actions specified on external systems and applications, and transaction resolution
10 component 221, for example. For example, a data table may have a "User" data field and an associated "Account" data field. When User A is received by ownership resolution component 216, it will conduct a lookup in the data table and identify Account A as the appropriate account associated with User A. Also, transaction resolution component 221 may have a data table that permits a lookup of the relationship between User A and User
15 B, identified in usage record 401. For example, upon receiving field 407 designating User A and field 408 designating User B, transaction resolution component 221 may conduct a lookup of an algorithm that identifies the appropriate burdens and benefits (*e.g.*, debit and credit of money) assigned to each of the users with respect to each other.

It also should be appreciated that transaction layer 103 may not provide additional
20 characteristics to usage record 401, as described but may select the relevant characteristics (*i.e.*, those characteristics needed by settlement layer 104) from the

characteristics made available by instrumentation layer 101 and/or derived by content collection layer 102. In fact, usage record 401 may include all of the necessary characteristics to properly account for the transaction. In this case, ownership resolution component 216 and transaction resolution component 221 select the relevant characteristics from the available characteristics.

Figures 5A and 5B provide a flow diagram of a method 500 for identifying the characteristics of a transaction using system 100. As shown in Figure 5A, in step 501, raw data 204-206 is received into system 100. Raw data 204-206 includes certain characteristics provided by data sources 201-203, respectively, in instrumentation layer 101. In step 502, content collection layer 102 processes (*e.g.*, aggregates, correlates and filters) raw data 204-206. In step 503, content collection layer 102 provides the processed data to CDR database 215 in transaction layer 103. CDR database 215 stores the data for further processing by transaction layer 103.

In step 505, it is determined whether the contents of the data provided to transaction layer 103 already include the needed ownership data. If the data includes the needed ownership data, the relevant data is selected by ownership resolution component 216 and provided to transaction resolution component 221, in step 507. If, on the other hand, the data provided to transaction layer 103 does not already include the needed ownership data, the required data is identified (*e.g.*, using database lookups and accountables) by ownership resolution component 216, in step 506. Once ownership resolution component 216 identifies the required ownership data, the relevant data is

selected by ownership resolution component 216 and provided to transaction resolution component 221, in step 507.

As shown in Figure 5B, in step 508 it is determined whether the contents of the data provided to transaction layer 103 already include the needed relationship data (*i.e.*, data defining the burdens and benefits encountered by the parties/participants to the transaction). If the data includes the needed relationship data, the relevant data is selected by transaction resolution component 221 and provided to settlement layer 104, in step 510. If, on the other hand, the data provided to transaction layer 103 does not already include the needed relationship data, the required data is identified (*e.g.*, using database lookups) by transaction resolution component 221, in step 509. Once ownership resolution component 216 identifies the required ownership data, the relevant data is selected by ownership resolution component 216 and provided to transaction resolution component 221, in step 510.

15 Ownership and Transaction Resolution

Figure 6 is a block diagram further detailing the operation and configuration of transaction layer 103. It should be appreciated that elements described for transaction layer 103 are not exclusive, but provide just one example of a configuration for transaction layer 103. In particular, the functionality of the previously described ownership resolution component 216 and transaction resolution component 221 will be described in greater detail in the various components described with reference to Figure 6.

In practice, however, it should be appreciated that transaction layer 103 may comprise of any number of components that ensure that the necessary parties to a transaction are identified generically.

As shown in Figure 6, transaction layer 103 includes a configuration repository 601, a configuration database 610, a configuration manager 603, and a fail over detector 604. In addition, transaction layer 103 includes a transaction layer engine 608 in communication with configuration repository 601, configuration manager 603, and fail over detector 604.

Transaction layer engine 608 includes a rules processor 605, a CDR reader 606, and a transaction writer 607. Transaction layer engine 608 processes raw data 204-206 in usage record 401 and creates a modified usage record 402, based upon the requirements of settlement layer 104. It should be appreciated that there may be more than one transaction layer engine 608 for each transaction layer 103. However, a single transaction layer engine 608 is shown for the purpose of clarity and brevity.

CDR reader 106 understands the nature of the transaction and operates to read a particular usage record 401 stored in CDR database 215. CDR reader 106 then submits usage record 401 to rules processor 605. Rules processor 605 applies rules (*e.g.*, written in XML) to usage record 401 so as to determine ownership and the relationships among the ownership, for each individual usage record 401. Rules processor 605 may then generate one or more modified usage records 402, based on the application of the rules to usage record 401. Once rules processor 605 generates the modified usage record 402,

transaction writer 607 operates to write a transaction (based on the modified usage record 402) to CDR database 215. Modified usage record 402 is then used by settlement layer 104 to further process the transaction (*e.g.*, rate the transactions monetarily).

Fail over detector 604 operates to provide fault tolerance and initiate a fail over engine (not shown) for a failed transaction layer engine, in accordance with one or more mathematical algorithms, the discussion of which is beyond the scope of the invention. Configuration manager 603 provides an external interface to transaction layer 103 using a business editor 609. Business editor 609 may be manipulated by a user or customer using a graphics user interface (GUI). In particular, the GUI of business editor 609 receives commands from a user and translates them into actions to be performed on usage record 401, for example, by rules processor 605. Configuration repository 601 operates to provide an abstraction to configuration database 610. Configuration database 610 may be a relational database or a set of extensible markup language (XML) computer-readable files, for example.

Figure 7 is a flow diagram that details the operation of transaction layer 103, as shown in Figure 6. As discussed, it should be appreciated that multiple transaction engines may be used to implement a process. Where the invention is instituted using computer-readable instruction, transaction layer engine 608 may be implemented as a computer-executable instruction or thread. Therefore, multiple transaction layer engines may be implemented by having multiple computer-readable instructions.

In step 701, transaction layer engine 608 reads usage record 401 using CDR reader 606. In step 702, rules processor 605 applies one or more rules to usage record 401 read by CDR reader 606, so as to determine one or more “accountables.”

Accountables are labels that are used to facilitate identifying the parties or owners to a transaction. Accountables may be defined and/or modified by an external user (*e.g.*, a customer) using the graphical user interface provided by business editor 609. Also, it should be appreciated that accountables may be defined for a new usage types for which no rules in rules processor 605 are provided. The identification of accountables is the function of ownership resolution component 216, as described with reference to Figure 4.

In step 703, “contracts” are identified that define the relationships or agreements between one or more accountables. In the context of a billing system, a contract identifies which parties pays (*i.e.*, the billed party) and which party receives payment (*i.e.*, the billing party). A particular contract may be identified and executed when one or more “conditions” are satisfied, however, conditions are not required. The identification of contracts is conducted by transaction resolution component 221, as described with reference to Figures 2 and 4. A condition is an expression of a logical assertion that can be evaluated as true or false. In transaction layer 103, for example, a condition may be true if a certain data element is present and false if the data element is not present. The following is an example of a condition in XML:

```
<condition name="condition1" function="IsPresent" outcome="true"
type="Accountable" id="ColoProvider"/>
```

In step 704, once the contracts are determined, they are executed. In step 705,
5 usage record 401 is changed to modified usage record 402 based upon the execution of
the contract and the identification of the accountables. In step 706, modified usage
record 402 may be written to a data table using transaction writer 607, so that settlement
layer 104 can rate it, in step 707. Therefore, a transaction may be represented as modified
10 usage record 402, where modified usage record may include the elements of usage record
401 in addition to the accountable and contract elements (*e.g.*, fields 409-410 as
discussed with reference to Figure 4).

Although the discussion provided an example of a single contract and two
accountables for a single usage record, it should be appreciated that many contracts and
many accountables (*i.e.*, and therefore many “transactions”) may be derived from a single
15 usage record, because there may be multiple transactions implicated from a single usage
record 401. For example, usage record 401 may involve three parties each having a
contract with the other respective parties. In this example, therefore, there may be three
separate contracts for a single usage record 401, and three individual accountables for
each party or owner.

20 In addition to requiring a certain accountable, it may be necessary to determine a
value of the accountable. For example, transaction layer 103 may determine not only that
the “called party” accountable be a part of the transaction, but also that a value for the

called party (*e.g.*, “856-555-1212” or “John Doe”) be identified. Also, it may be necessary to transform one accountable into another accountable to satisfy the requirement of settlement layer 104. For example, settlement layer 104 may require the called party accountable (either determined by transaction layer 103 or provided in usage record 401) be transformed into a network identifier accountable.

In either case, the value or transformation of accountables or contracts may be determined using certain predetermined functions or “primitives.” A primitive allows the user to specify how to find the value of a particular an accountable, and/or how to find billing and/or billed account numbers associated with a particular contract. As discussed, transformations may allow a value of an accountable to be determined using another accountable (called “linked accountables”), for example. Primitives define and direct such transformations. The input to a primitive can be one of fields 403-408 from usage record 401 or a previously determined accountable, for example.

Primitives may be described and distinguished based on their prescribed functionality. For example, one type of primitive may be an external database lookup. A database lookup is a primitive that operates to transform one accountable or contract to another accountable or contract. Also, a database lookup primitive may function to assign a value to an accountable or contract, as a result of conducting a lookup of the accountable or contract in a database. It should be appreciated that the database lookup primitive also allows flexibility in that any database that has Java Database Connectivity (JDBC) drivers (*e.g.*, Oracle, Microsoft SQL Server, Sybase, Database 2) can be used to

perform complex lookups. For native databases that do not have JDBC connectivity, native databases may be accommodated through native methods that may be used through primitives. In this way, therefore, previous accountable or contact transformations can be linked to find new accountables or contracts, respectively.

5 Other types of primitives may include direct value assignment, and a function primitive. A function primitive provides flexibility and extensibility by allowing a third party based library (*e.g.*, Java-based or native libraries) to be used. The arguments of a function primitive may be text strings. Also, the arguments of a function primitive may be expressed in terms of fields in CDR database 215 (*i.e.*, string-based fields in usage
10 record 401), and other previously determined accountables. Therefore, function primitives permit the integration of proprietary and domain-specific algorithms to be implemented to perform ownership related processing.

Another primitive may be a “regular expression” based primitive that allows parsing of complex text-based fields with a full feature (*e.g.*, PERL5 compliant) pattern
15 specification. Common used techniques would include parsing URLs, filenames and paths, calling/called telephone numbers, email address, for example. In addition, a CDR database 215 lookup primitive allows holding on to an ownership entity if no further processing is required. This primitive is used where usage record 401 provides the information needed by settlement layer 104 without any need for more data correlation
20 and/or manipulation by transaction layer 103.

A default primitive may be a hard-coded primitive that is used where the value is known and/or is not dependent on usage record 401. For example, such a primitive may be used where a party is always present in a consumption event, regardless of the other relevant parties. A network address primitive allows determination of a network entity given an arbitrary string. An area code primitive may be used to determine an area code of a telephone number given the complete telephone number. This primitive may be relevant for a Voice-over-IP application. These examples primitives are not exclusive, but are meant to provide a few examples of the manipulation and correlation that may be achieved by the primitives on the accountables and contracts.

The following is an example of an accountable primitive in XML:

```
<accountable>
  <name> ColoProvider </name>
  <role> ServiceProvider </role>
  <usageType> RMONHost </usageType>
  <valid from="past" to="present" />
  <primitive type="default">
    <value> Exodus </value>
  </primitive>
</accountable>
```

Each accountable that is involved in a contract will have an assigned "role." It follows, therefore, that an accountable that is not involved in a contract may not need to have an assigned role. The role associates the particular accountable for integration with settlement layer 104. This means that the concept of a role is used for the TL/AMI integration.

Similar to the accountable, a contract is implemented using a set of primitives. The following is an example of a contract primitive in XML, and is provided to further describe the invention. In the example, a Web hosting scenario is present where a content provider (*e.g.*, CNN) places content on one or more Web servers and pays a Web Hoster for doing so based on some predefined pricing plan,

In order to define the accountables, certain elements of usage record 401 may be assumed as follows: Web Hit Time, Web Server Name/IP address, Client IP address, bytes, Mime type, and URL. In this example, if a rule in the business world requires the "Content Provider" of the Web Server (*e.g.*, CNN) to pay to the Web Hoster, there are two accountables: Web Hoster and Content Provider. The XML definition of a contract that captures this business rule may be as follows:

```

01    <contract>
02    <name> ContentProvider-Pays-WebHoster </name>
15    <usageType> WebHosting </usageType>
04    <valid from="past" to="present" />
05    <condition function="IsPresent" type="Accountable"
      name="condition1" id=" ContentProvider " />
06    <condition function="IsPresent" type="Accountable"
20    name="condition2" id="WebHoster" />

07    <billedFunction>
08    <status> enabled </status>
09    <primitive type="internal-account-lookup">
25    <database name="GlobalDB"/>
11    <argument order="1" type="Accountable"
      name="ContentProvider"/>
12    </primitive>
13    </billedFunction>

```

```
14     <billingFunction>
15     <status> disabled </status>
16     <primitive type="internal-account-lookup">
5      17     <database name="GlobalDB"/>
18     <argument order="1" type="Accountable" name="WebHoster"/>
19     </primitive>
20     </billingFunction>
10    21     </contract>
```

In this example, a contract has a name defined in line 02 as “ContentProvider-Pays-WebHoster.” The contract also defines an element or usage type called “WebHosting” in line 03. The contract also has two sets of conditions in lines 05 and 06, called “IsPresent,” which consider whether “ContentProvider” and “WebHoster” elements are present, respectively. Therefore, this particular contract will be selected and executed by transaction layer 103 if these elements are present.

Also, in this contract example, there are two primitives that are specified. There is a first primitive for finding the account number of the billed entity “ContentProvider,” which begins on line 07 and ends on line 13, and there is a primitive for determining the account of the billing entity “WebHoster” beginning on line 14 and ending on line 20. Therefore, the example contract contemplates the “who pays to whom” logic, and results in the execution of a contract that creates a modified usage record 402, which captures the transaction. Notably, in lines 09 and 16, both primitives are defined as database lookups.

The invention is directed to a system and method for of determining the relationships among parties to a transaction. The invention often was described in the

context of the Internet, but is not so limited to the Internet, regardless of any specific description in the drawing or examples set forth herein. For example, the invention may be applied to wireless networks, as well as non-traditional networks like Voice-over-IP-based networks, private networks, and/or traditional business environments like
5 commodity trading. It will be understood that the invention is not limited to the use of any of the particular components or devices herein. Indeed, this invention can be used in any application that requires the determining of relationships among parties to a transaction. Further, the system disclosed in the invention can be used with the method of the invention or a variety of other applications.

10 While the invention has been particularly shown and described with reference to the embodiments thereof, it will be understood by those skilled in the art that the invention is not limited to the embodiments specifically disclosed herein. Those skilled in the art will appreciate that various changes and adaptations of the invention may be made in the form and details of these embodiments without departing from the true spirit
15 and scope of the invention as defined by the following claims.

What is Claimed is:

1. A method of determining a relationship among parties to a transaction, comprising:
 - receiving data related to an event, wherein said data comprises one or more data elements;
 - selecting said data elements that identify a relationship among the parties to the transaction;
 - determining whether additional data elements are required to identify the relationship among the parties to the transaction; and
 - identifying the additional data elements.
2. The method of claim 1, wherein the relationships define an action to be taken among the parties.
3. The method of claim 2, wherein the action includes at least one of the following:
 - payment of money, extension of credit, exchange of data, transfer of ownership, exchange of services.
4. The method of claim 1, wherein said event is an exchange of data between the parties over a communication network.

5. The method of claim 1, wherein said data related to said event is stored in a usage record.
6. The method of claim 1, wherein said determining is a function of the received data.
7. The method of claim 1, wherein said determining is a function of the requirements of a customer settlement system.
8. The method of claim 7, wherein said customer settlement system includes at least one of the following: a data security system, a network capacity planning system, and a content monitoring system.
9. The method of claim 1, wherein said data element is a contract.
10. The method of claim 1, wherein said identifying of additional data elements is initiated by a primitive.
11. The method of claim 10, wherein said primitive includes at least one of the following: a database lookup, direct value assignment, regular expression, CDR database lookup, hard-coded, network address, substring manipulation, area code, and a user-specified action.

12. A system for determining parties to a transaction, comprising:
- an instrumentation layer, wherein the instrumentation layer provides raw data;
 - a content collection layer in communication with the instrumentation layer,
- wherein the content collection layer generically processes the raw data; and
- a transaction layer in communication with the content collection layer, wherein the transaction layer identifies relationships among the parties to the transaction.
13. The system of claim 12, wherein said content collection layer provides said raw data to said transaction layer in a usage record.
14. The system of claim 13, wherein said transaction layer modifies said usage record as a function of identifying relationships among the parties to the transaction.
15. The system of claim 12, further comprising a settlement layer in communication with said transaction layer, wherein the settlement layer rates the modified usage record.
16. The system of claim 13, wherein said transaction layer comprises a database for storing said data processed by said content collection layer.
17. A computer-readable medium having computer-executable instructions for:

receiving data related to an event, wherein said data comprises one or more data elements;

selecting said data elements that identify a relationship among the parties to the transaction;

determining whether additional data elements are required to identify the relationship among the parties to the transaction; and

identifying the additional data elements.

18. The computer-readable medium of claim 17, wherein the relationships define an action to be taken among the parties.

19. The computer-readable medium of claim 18, wherein the action includes at least one of the following: payment of money, extension of credit, exchange of data, transfer of ownership, exchange of services.

20. The computer-readable medium of claim 17, wherein said event is an exchange of data between the parties over a communication network.

21. The computer-readable medium of claim 17, wherein said data related to said event is stored in a usage record.

22. The computer-readable medium of claim 17, wherein said determining is a function of the received data.
23. The computer-readable medium of claim 17, wherein said determining is a function of the requirements of a customer settlement system.
24. The computer-readable medium of claim 23, wherein said customer settlement system includes at least one of the following: a data security system, a network capacity planning system, and a content monitoring system.
25. The computer-readable medium of claim 17, wherein said data element is a contract.
26. The computer-readable medium of claim 17, wherein said identifying of additional data elements is initiated by a primitive.
27. The computer-readable medium of claim 26, wherein said primitive includes at least one of the following: a database lookup, direct value assignment, regular expression, CDR database lookup, hard-coded, network address, substring manipulation, area code, and a user-specified action.

100 ↗

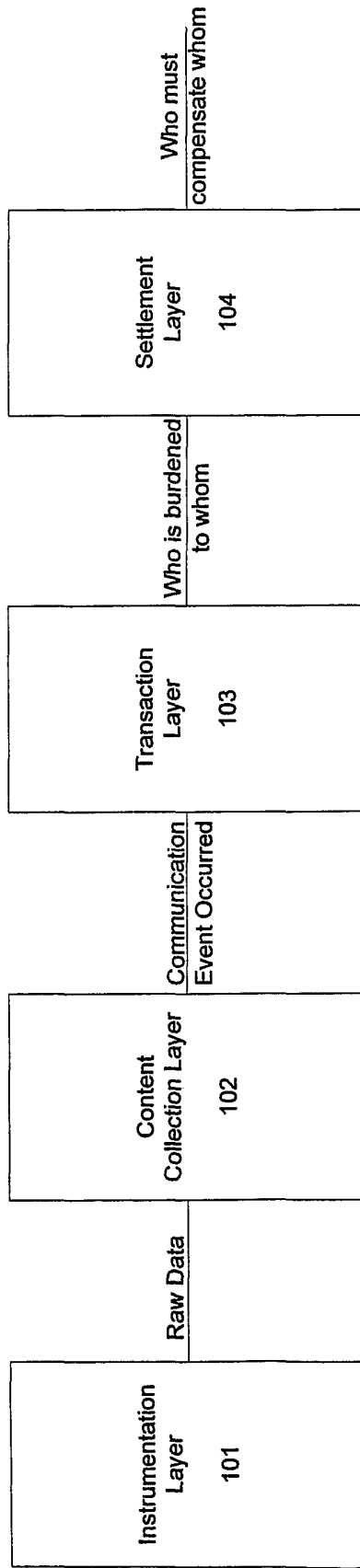


Figure 1

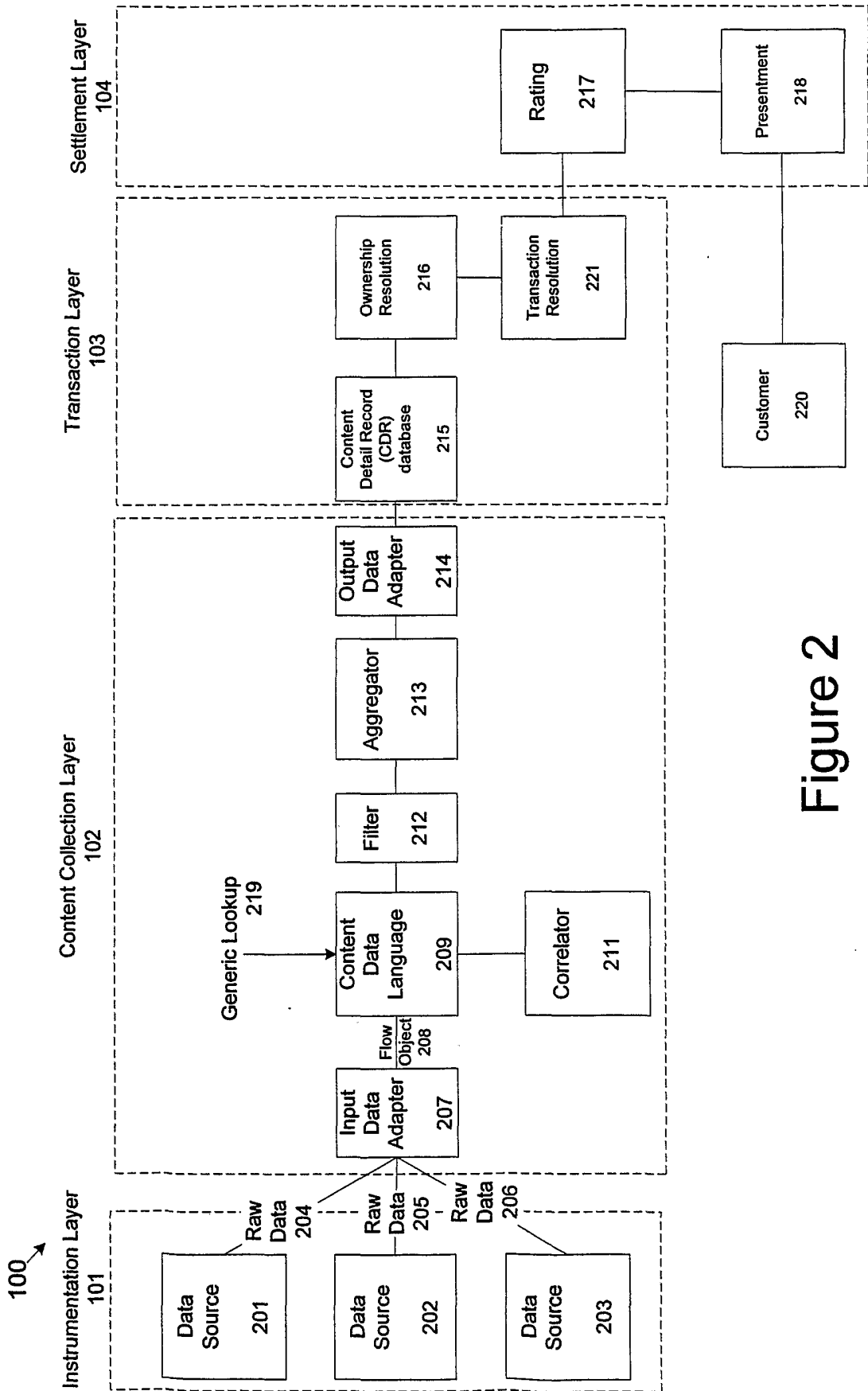


Figure 2

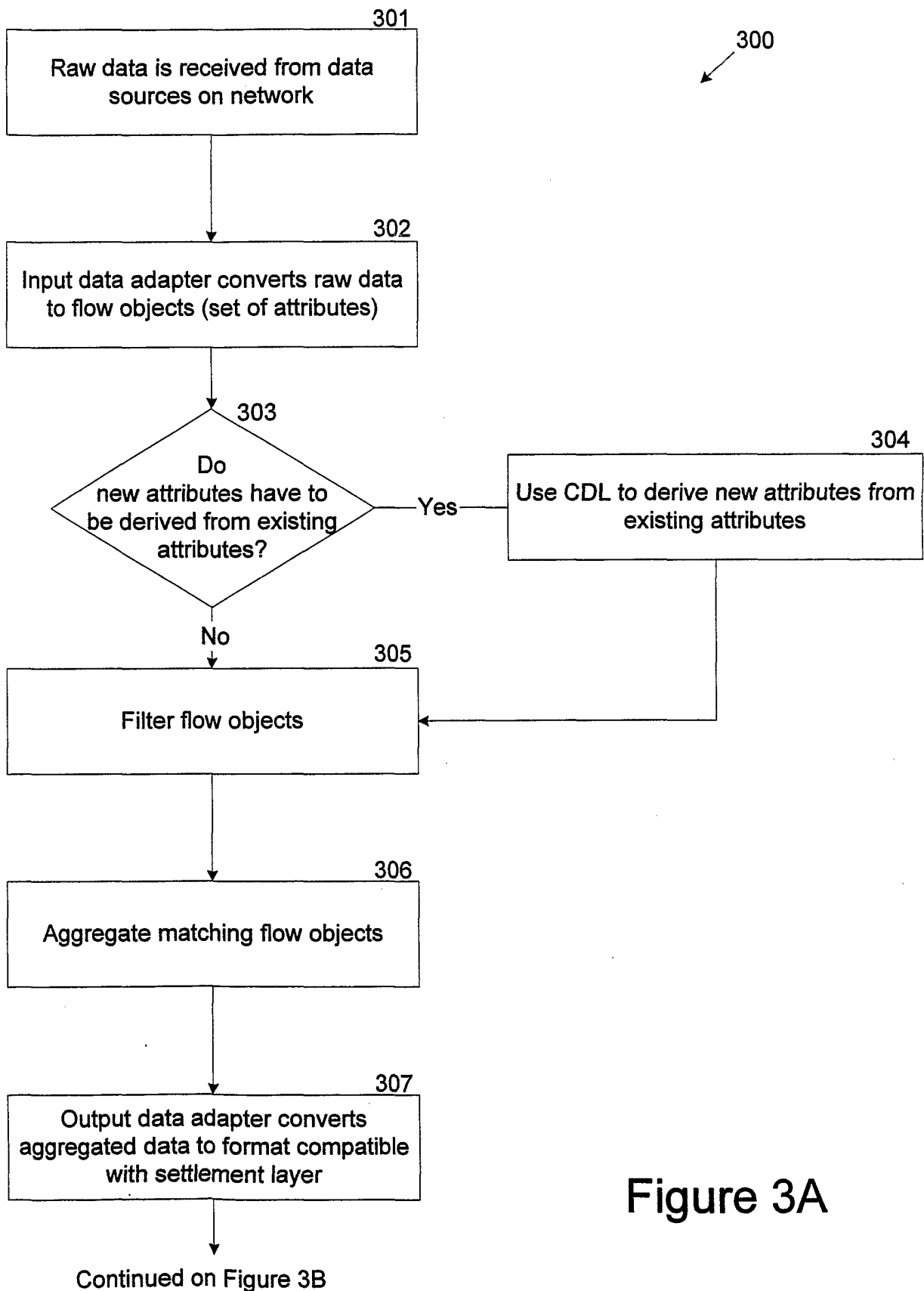


Figure 3A

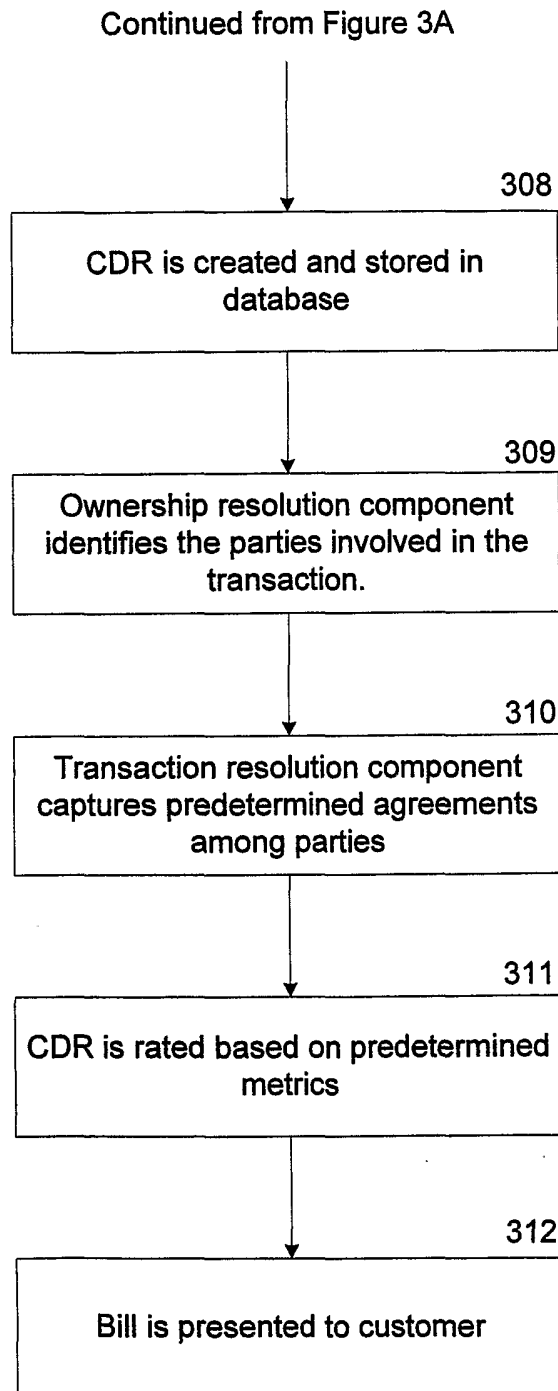


Figure 3B

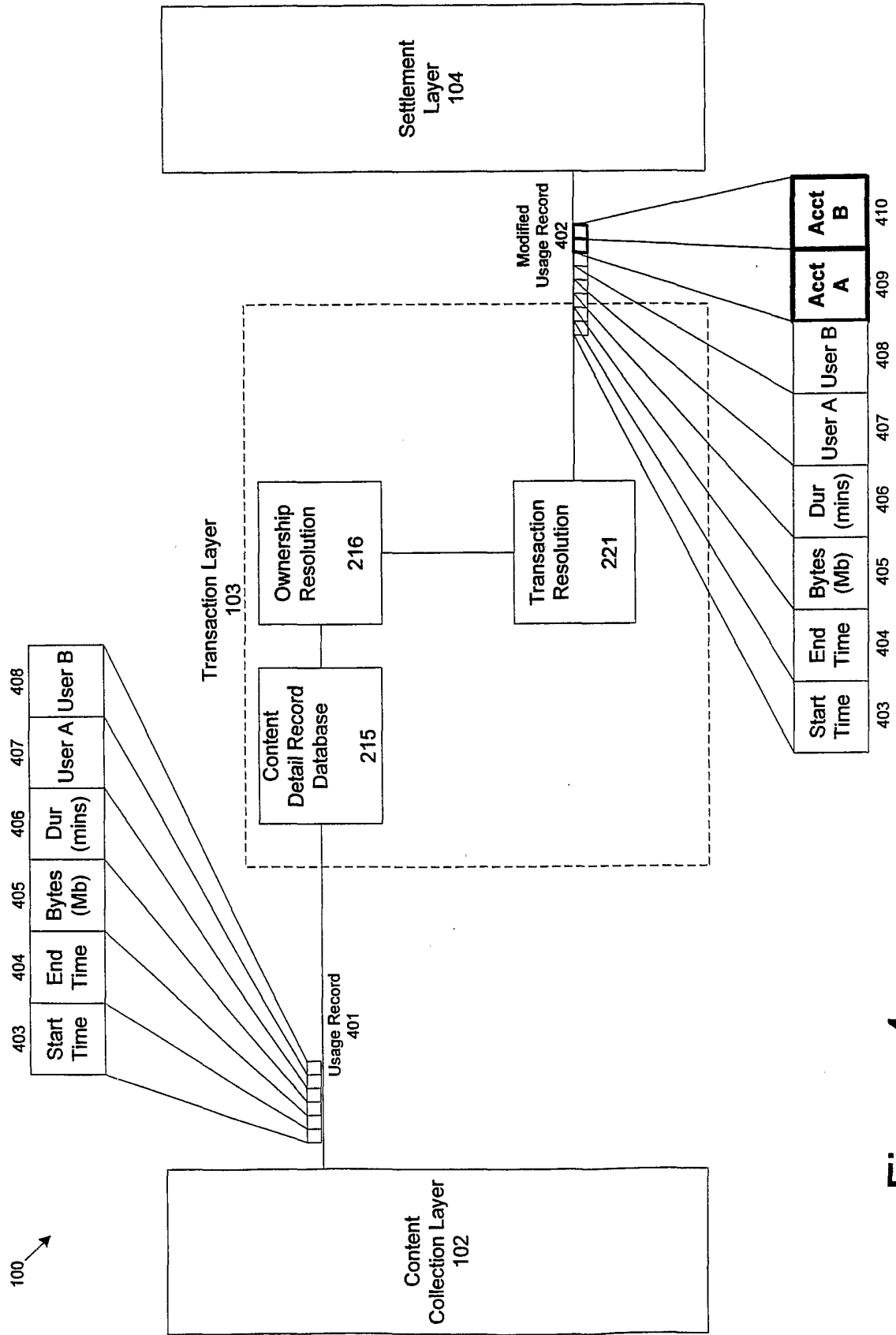


Figure 4

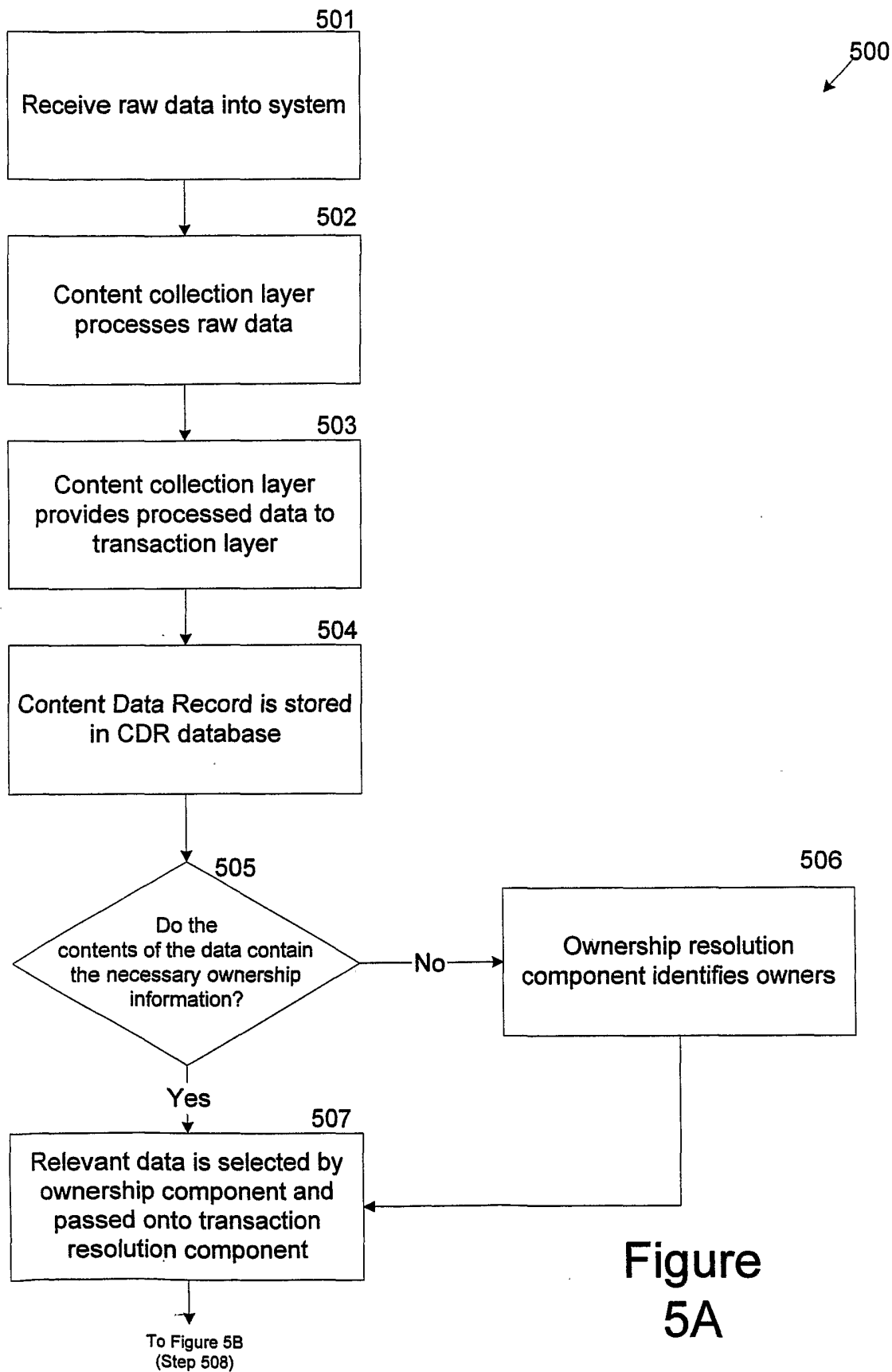


Figure 5A

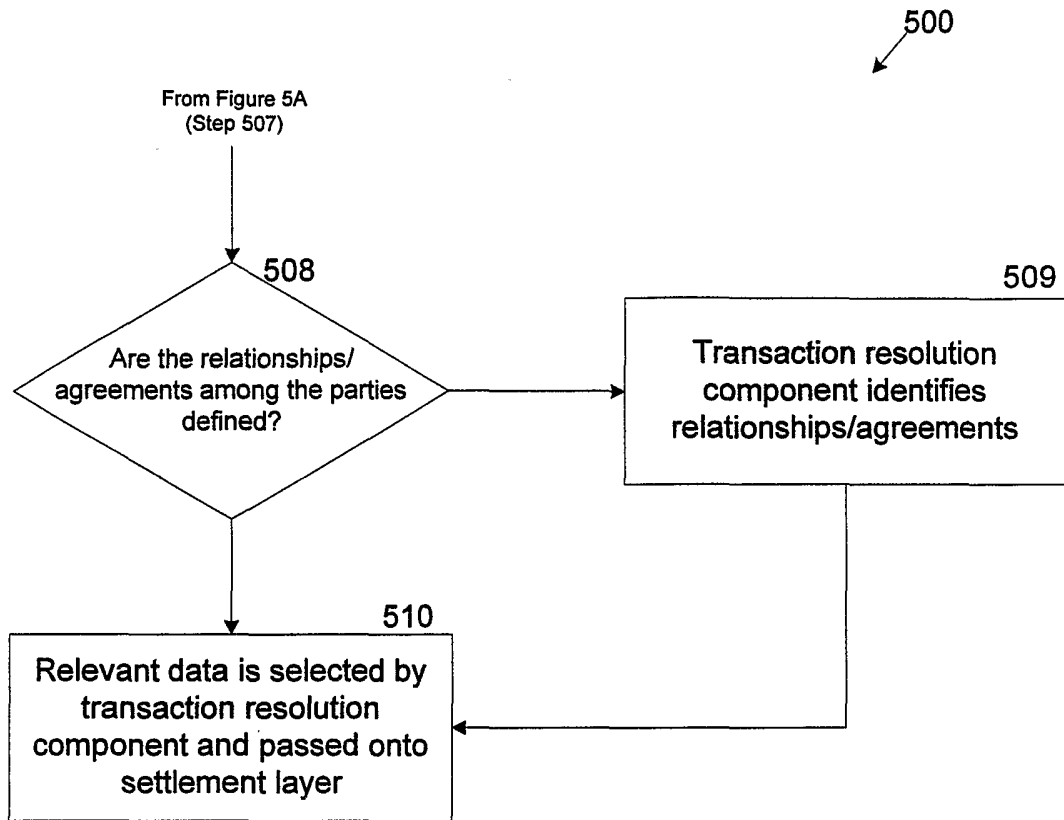


Figure 5B

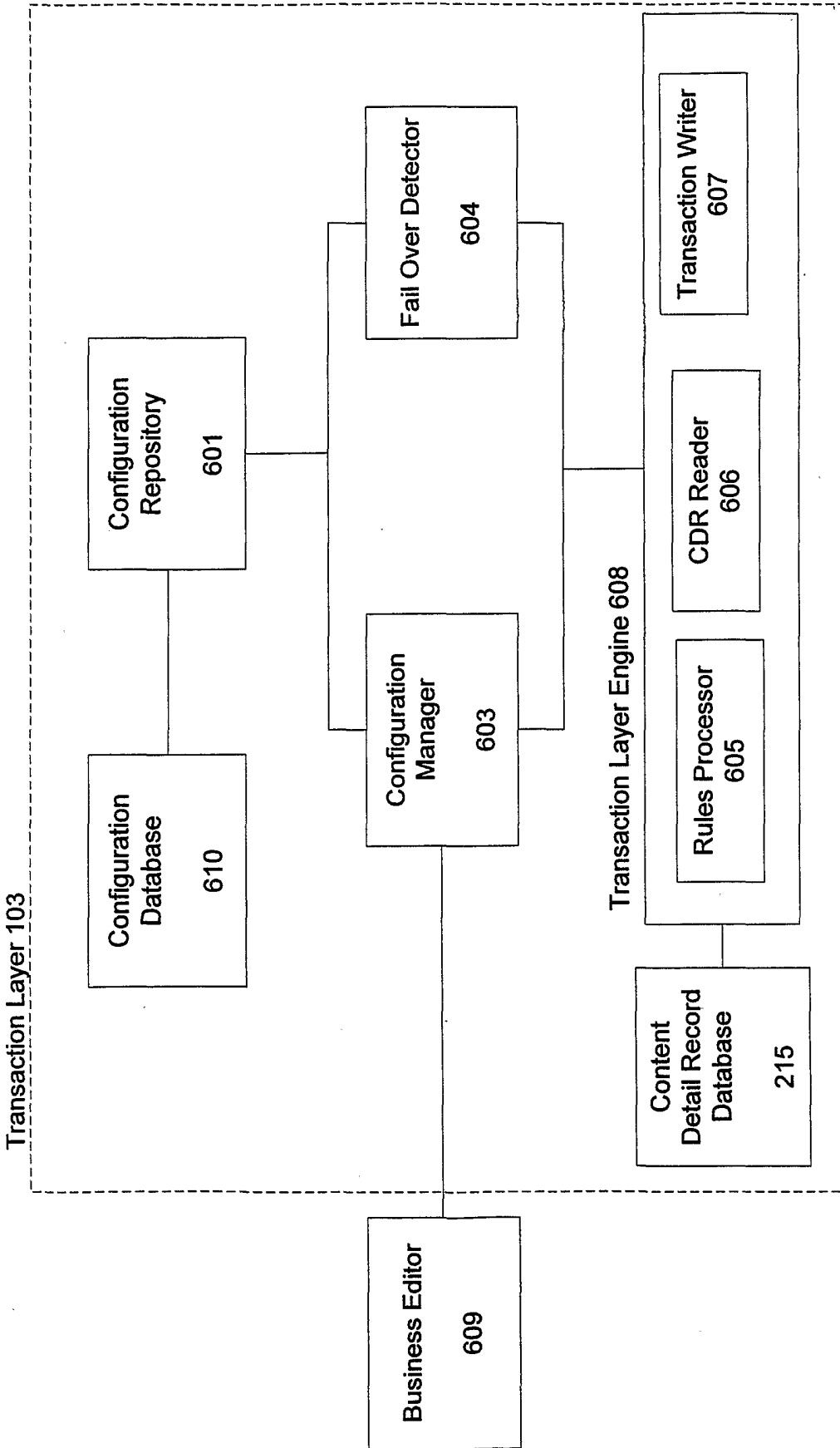


Figure 6

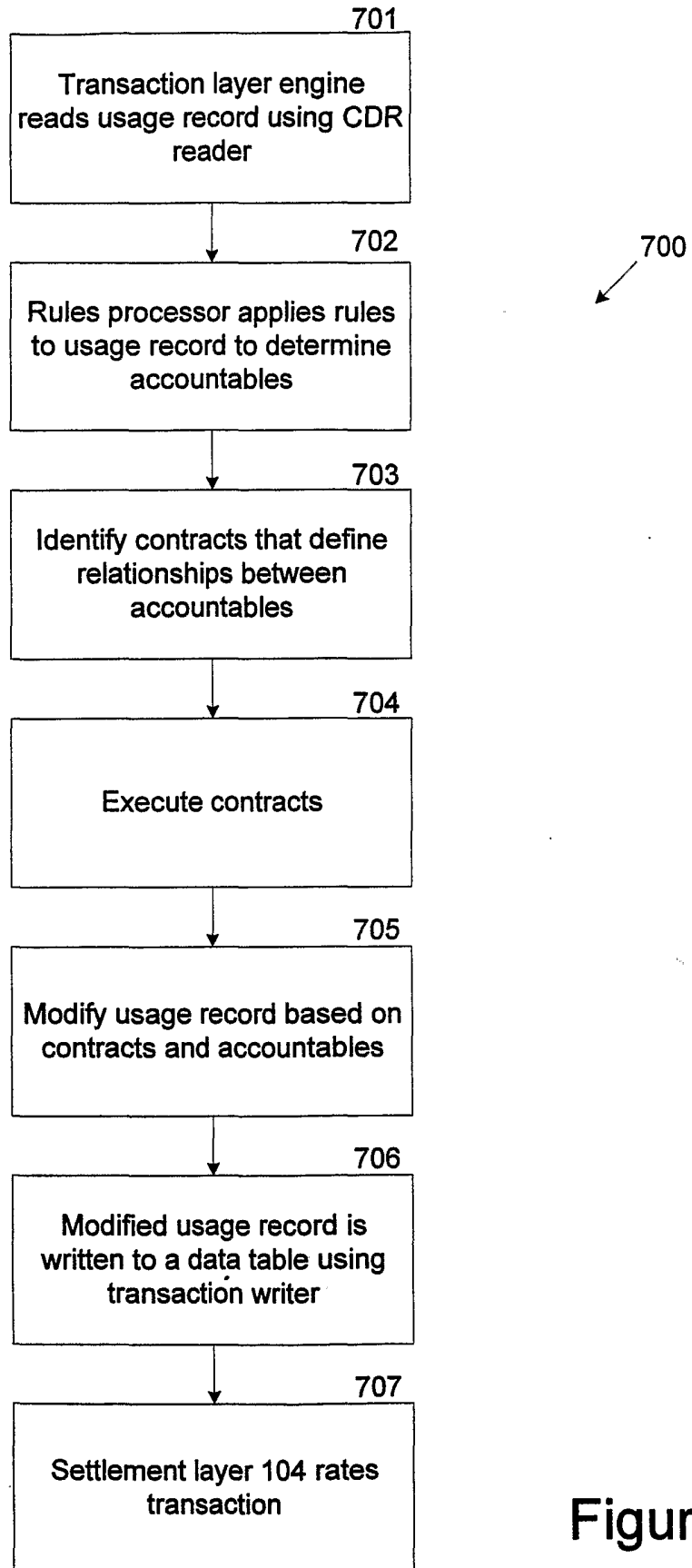


Figure 7