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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 February 2012 (16.02.2012)

PCT

(10) International Publication Number
WO 2012/021427 A2

(51) **International Patent Classification**:
G06F 21/24 (2006.01) *G06F 21/20* (2006.01)

(21) **International Application Number**:
PCT/US2011/046888

(22) **International Filing Date**:
8 August 2011 (08.08.2011)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
61/372,293 10 August 2010 (10.08.2010) US
12/943,765 10 November 2010 (10.11.2010) US

(71) **Applicant** *(for all designated States except US)*: **BENEFITFOCUS.COM** [US/US]; 100 Benefitfocus Way, Charleston, South Carolina 29492 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **JAIN, Amit** [IN/IN]; Flat 50106, Pine Block, Indu Fortune Fields, 13th Ph, KPHB, Hyderabad 500072 (IN). **TANNER, JR., Theodore C.** [US/US]; 5272 7th Green Drive, Hollwood, South Carolina 29449 (US).

(74) **Agent**: **THOMPSON, John R.**; Stoel Rives LLP, 201 So. Main Street, Suite 1100, One Utah Center, Salt Lake City, Utah 84111 (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, 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, IS, JP, KE, KG, KM, 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, PE, PG, PH, PL, PT, QA, RO, RS, RU, 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, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) **Title**: SYSTEMS AND METHODS FOR SECURE AGENT INFORMATION

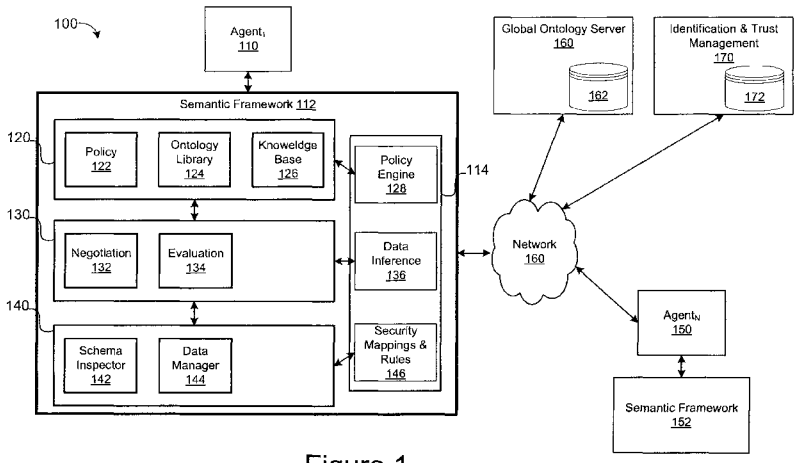


Figure 1

(57) **Abstract**: Semantic information may be secured by an agent using one or more semantic security labels (*e.g.*, security predicates). The agent may be configured to allow other agents to access the semantic information according to a set of semantically expressed policies, strategies, and/or rules. A request to receive information may be mapped to a negotiation policy of the agent. The agent may evaluate the request against a semantic information sharing policy. If the information is accessible under the information sharing policy, the information may be provided. If not, the agent may negotiate information sharing terms using the negotiation ontology, strategy, and rules. Similarly, the agent may request information from other entities. Terms of the information requests may be negotiated using the negotiation ontology, strategy, and rules.

WO 2012/021427 A2

SYSTEMS AND METHODS FOR SECURE AGENT INFORMATION

TECHNICAL FIELD

[0001] This disclosure relates to systems and methods for sharing semantic data and, in particular, to systems and methods for applying semantic-based security predicates to shared structured and unstructured data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Additional aspects and advantages will be apparent from the following detailed description of preferred embodiments, which proceeds with reference to the accompanying drawings, in which:

[0003] **Figure 1** is a block diagram of one embodiment of a system for semantically securing and sharing semantic information;

[0004] **Figure 2** is a flow diagram of one embodiment of a method for sharing semantically secured information;

[0005] **Figure 3** is a flow diagram of one embodiment of a method for negotiating sharing of semantically secured information; and

[0006] **Figure 4** is a flow diagram of one embodiment of a method for requesting semantically secured information.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0007] There is an increasing amount of data available via data communications networks. This data may be controlled by one or more entities. In order to control access to the data, the entity may store the data in a data storage system (*e.g.,* database storage systems, directory systems, and the like), that is accessible only in a controlled network environment, such as within a Local Area Network (LAN) controlled by the entity and/or within the entity's Virtual/Private Local Area Network (VLAN). Where the entity makes the data available on a more widely accessible network (*e.g.,* the Internet), access to the data may be controlled by the entity's security system and/or access control rules, such as role-based access control (RBAC), discretionary access control (DAC), or the like.

[0008] An entity may wish to share information with other entities. However, data sharing between entities may be problematic, since access to the data may be secured using entity-specific security and access control rules. Translation between security systems may be a complicated, manual process which may be time-consuming and problematic. Moreover, the translation may result in data security problems, since the receiving entity may choose to disregard the security directives of the sending entity.

[0009] In addition, the data shared by entities may be expressed semantically. The semantic expression of data may allow it to be processed autonomously (*e.g.,* by expert systems, agents, or the like), which may significantly increase the value of the data and allow for sophisticated data mining, data analysis, and other applications. The Resource Description Framework (RDF) provides a machine understandable description data, including resources, resource properties, and relationships. Other examples of semantic data formats include, Web Ontology Language (OWL), OWL 2, Resource Description Format Schema (RDFS), and the like.

[0010] As used herein, an ontology and/or semantic data may refer to an organized collection of knowledge referring to both the meaning of terms (*e.g.,* concepts) and relationships between concepts. In some embodiments of this disclosure, semantic data (*e.g.,* ontological data) may comprise a plurality of vertices (*e.g.,* nodes) interconnected by one or more edges. The vertices within the ontology graph may be concepts within the ontology, and the edges interconnecting the vertices may represent relationships between related concepts within the ontology.

[0011] Traditional security and/or access control systems may be ineffective at properly controlling access to semantic data. For example, many semantic data formats (*e.g.,* RDF), may allow the same "statement" (*e.g.,* resource, property, and/or value) to be expressed in many different ways using many different syntaxes. Traditional security systems may be incapable of applying access control rules to all of the possible permutations and/or expression formats of the data. Moreover, manually applying such rules would quickly become untenable.

[0012] In addition, semantic data may be used to infer and/or entail other related semantic data. A security and/or access control system must apply consistent access control rules to inferred and/or entailed information (*e.g.,* prevent protected data from being "inferred" and/or "entailed" from less protected data).

[0013] Moreover, the negotiation between entities relating to security policies may be streamlined or even automated. Streamlining the security negotiation process between entities may allow data sharing between a large number of entities, which, in turn, may significantly increase the value of the data controlled by the entities.

[0014] In some embodiments disclosed herein, access to semantic data (*e.g.,* RDF content) may be controlled by a Secure Resource Description Framework (SRDF). RDF semantic data may comprise a set of triples organized into a named, directed graph structure including resources, properties, references, literals, and variables. An RDF statement is a

resource rectifying triple, comprising three properties: an RDF subject, object, and predicate. The SRDF may operate by applying security predicates to RDF statements using an RDF pattern.

[0015] An RDF pattern may comprise a triple, where each component is either a data constant or a variable. A pattern mapping may be a mapping between a first RDF pattern PT_1 and a second RDF pattern PT_2 such that, for the mapping V, $V: PT_1 \rightarrow PT_2$. An SRDF security policy may comprise a set of pairs $SP = \{ sp_1, ..., sp_n \}$, such that every security policy comprises a respective security label and corresponding pattern. The security label (*e.g.,* a security predicate) may be applied to RDF statements matching the pattern. The security predicate may control access to the RDF statement, assign security classifications to newly generated statements (*e.g.,* to which the security policy pattern corresponds), and check for unauthorized inferences (discussed below).

[0016] An SRDF security cover may be a set of all the RDF security policy mappings. The security cover may further comprise rules describing how security label conflicts are managed (*e.g.,* in the event that multiple patterns match a particular RDF statement).

[0017] The SRDF may include conflict resolution rules. Some of these rules may define which security label (*e.g.,* security predicate) is mapped to a particular RDF statement in the event that several security labels map thereto. The conflict resolution rules may further include rules to handle inference conflicts, which may occur due to newly entailed RDF statements. In one embodiment, the conflict resolution rules may be comprised of a table of mappings to such statements. However, a generated RDF statement may already be within the security cover and have a different security label. The difference in the security labels may create a security policy violation (*e.g.,* if the existing RDF statement has a "higher" or different security level than the inferred RDF statement). In some embodiments, this type of "inferred" conflict may be detected if the security label of an inferred statement is "lower" (*e.g.,* less restrictive) than an existing statement.

[0018] An inferred conflict may be resolved in a number of different ways, according to the preferences and/or security requirements of the entity. For example, the security label of the inferred statement may be completely subsumed (*e.g.,* replaced) by the original, "higher" security label. Alternatively, in some embodiments, the security policy may allow the inferred statement to retain the less restrictive security level. If the new label is "higher" (*e.g.,* more restrictive) than the original label, the new label may be retained. In some cases, the security level may be demoted to the original level. Alternatively, or in addition, the

resulting security label may comprise a combination of the original security label and the new, inferred label.

[0019] In some embodiments, a security policy (*e.g.,* set of security labels and associated mappings) may be checked for consistency. This may comprise detecting direct and inferred conflicts within the policy. If the checking indicates that conflicts exist (*e.g.,* either direct and/or inferred), a policy maintainer may be informed and appropriate action recommended. Alternatively, or in addition, the security policy may be automatically modified to remove the direct and/or inferred conflicts substantially as discussed above (*e.g.,* by subsuming one or more conflicting policies, retaining an inferred security label, and/or by combining security labels).

[0020] Using the security predicates discussed above, systems and methods may be developed to facilitate secure data sharing between entities. The data shared between the entities may comprise unstructured data and/or structured data (*e.g.,* semantic data formatted according to RDF).

[0021] **Figure 1** is one embodiment of a system for secure sharing of agent information. An $agent_1$ 110 may be communicatively coupled to a network 160. The network 160 may comprise any communications network known in the art including, but not limited to: an Internet protocol (IP) network (*e.g.,* a Transmission Control Protocol IP network (TCP/IP) or the like), a local area network (LAN), a wide area network (WAN), a virtual private network (VPN), a wireless network (*e.g.,* IEEE 802.11a-n, Bluetooth®, EDGE, G3, or the like), a public switched telephone network (PSTN), a cellular telephone network, or the like.

[0022] The $agent_1$ 110 may comprise a human agent, an expert system, or any other entity capable of processing and/or controlling the flow of structured and/or unstructured data. In some embodiments, the $agent_1$ 110 may comprise a computing device (not shown). The computing device of the $agent_1$ 110 may comprise one or more processors (not shown) communicatively coupled to memory (not shown), data storage, such as a hard disc, storage area network, or the like (not shown), one or more input/output devices, such as a keyboard, a mouse, a display, and the like (not shown), a communications interface (not shown), and the like.

[0023] The computing device may be substantially autonomously controlled by a software $agent_1$ 110 (*e.g.,* an expert system or the like). The software $agent_1$ 110 may comprise a user interface by which a human operator (not shown) may control the operation of the $agent_1$ 110. Alternatively, or in addition, the $agent_1$ 110 may be substantially directly controlled by a human operator.

[0024] The $agent_1$ 110 may be communicatively coupled to a semantic framework 112. In some embodiments, the semantic framework 112 may be a component of the $agent_1$ 110 (*e.g.,* be hosted on the computing device of the $agent_1$ 110 or the like). Alternatively, or in addition, portions of the semantic framework 112 may reside on one or more separate computing devices (not shown). In this case, the computing device of the $agent_1$ 110 may be communicatively coupled to the one or more computing devices comprising the semantic framework 112 via a communications network, such as the network 160, a separate LAN (not shown), a VPN (not shown), or the like.

[0025] The semantic framework 112 may comprise information managed and/or controlled by the $agent_1$ 110. For example, the $agent_1$ 110 may represent a healthcare provider (*e.g.,* a health maintenance organization), and, as such, the information managed by semantic framework 112 of the $agent_1$ 110 may include healthcare statistics, patient records, billing information, and the like.

[0026] The semantic framework 112 may comprise a data, ontology, and rules layer 120 comprising a policy module 122, an ontology library 124, and a knowledge base 126. The policy module 122 may comprise a security policy defining a security and/or access control policy of the semantic framework 112.

[0027] The policy module 122 may comprise rules governing information sharing between entities in the network 160 (*e.g.,* between the $agent_1$ 110 and other, external entities, such as the $agent_N$ 150). The policy module 122 may comprise policies represented using a policy language having different language constructs, such as actors, actions, obligations, authorizations, delegation, privacy, and so on. For example, an "actor" within the policy module 122 may correspond to a particular agent (*e.g.,* $agent_N$ 150) on the network 160 and/or an individual within the $agent_N$ 150 (*e.g.,* a sub-agent of the $agent_N$ 150, such as an employee, a delegee, or the like).

[0028] The policy information of the policy module 122 may be divided into various, different sub-policies according to the requirements of the $agent_1$ 110. For example, the policy module 122 may comprise one or more sharing policies defining how the $agent_1$ 110 is to share information with entities on the network 160. The policy module may further comprise policies directed to trust management (*e.g.,* which entities the $agent_1$ 110 trusts), delegation policies, privacy policies, authentication policies, authorization policies, and the like.

[0029] The policy module 122 may comprise ontology-based policies (*e.g.,* semantic policies), which may be used to represent the dynamic entities (*e.g.,* conceptual information)

managed by the semantic framework 112 (*e.g.,* within the ontology library 124). These ontology-based policies may define policy elements, such as subjects, resources, credentials, and obligations between ontological elements (*e.g.,* between resources, properties, values, and the like within the ontology library 124 and/or accessible in an ontology 162 of the global ontology server 160).

[0030] The policies information of the policy module 122 may comprise security labeling information (*e.g.,* may comprise security predicates, such as the SRDF security labels discussed above). The labels may indicate whether a particular policy is available to other agents on the network 150 and/or whether the policy is private.

[0031] The ontology library 124 may comprise a domain-specific ontology of the $agent_1$ 110. The ontology library 124 may be used to represent domain information of the $agent_1$ 110. The ontology library 124 may comprise ontology information in RDF, OWL, OWL2, or another related format. The ontology library 124 may be used to semantically classify unstructured data and/or to import semantic data from a third-party, external source (*e.g.,* $agent_N$ 150).

[0032] Various systems and methods for mapping unstructured data to concepts within an ontology are disclosed in co-pending provisional application entitled "Systems and Methods for Concept Mapping," Serial Number No. 60/086,335, filed August 5, 2008, to Gregory Jorstad et al., which is hereby incorporated by reference in its entirety.

[0033] The ontology library 124 may be used to semantically represent the data managed and/or controlled by the $agent_1$ 110. Accordingly, the ontology library 124 may be a domain-specific ontology tailored to the particular application provided by the $agent_1$ 110. For instance, as discussed above, the $agent_1$ 110 may be associated with a healthcare provider. As such, the ontology library 124 may comprise detailed health-care related concepts and relationships. In contrast, an ontology library 124 of an $agent_1$ 110 associated with another type of entity (*e.g.,* an investment banking institute) may comprise detailed concepts and relationships related to other areas. Moreover, one or more of the agents 110 or 150 may operate using a sparse ontology library 124 to improve efficiency (*e.g.,* an ontology where concepts that are unrelated to the application area of the agent 110 or 150 are removed).

[0034] The ontology library 124 may be a secure ontology, such that the information stored therein may be mapped to a security policy (*e.g.,* a security label and/or predicate). The security policy of the semantic framework 112 may comprise a plurality of security patterns, including a default security pattern, which may leave no data within the ontology library 124 unclassified and/or without a security label. The security label applied to the data

within the ontology library 124 may be periodically updated as the security policies of the semantic framework 112 are revised.

[0035] As will be discussed below, some applications of the system 100 may require that information corresponding to an ontology of a first agent (*e.g.,* $agent_1$ 110) be mapped or translated to an ontology of a second agent (*e.g.,* $agent_N$ 150). As discussed above, however, the domain-specific ontology library 124 of the agents 110 and 150 may differ, which may make a straight-forward, one-to-one mapping between the respective ontology libraries problematic. In this case, one or more of the agents 110 and 150 may access an ontology 162 of the global ontology server 160. The global ontology 162 may be a more "complete" (non-sparse) ontology library. An agent 110 or 150 receiving information corresponding to an unknown concept (*e.g.,* a concept that does not appear in the local ontology library 124 of the agent) may determine a mapping into the local ontology 124 using the concepts and/or relationships provided by the global ontology 162. Alternatively, or in addition, an agent (*e.g.,* $agent_1$ 110) may perform a "one-time" import of a portion of the global ontology 162 provided by the global ontology server 160 in order to effectively process data from a third-party, external agent (*e.g.,* $agent_N$ 150).

[0036] The knowledge base module 126 may comprise the semantic data managed and/or controlled by the semantic framework 112. As discussed above, the knowledge base 126 may comprise data in semantic form (*e.g.,* in RDF, OWL, OWL2, or the like). The data within the knowledge base 126 may comprise the security labels discussed above in conjunction with the ontology library 124.

[0037] The semantic framework 112 may comprise a negotiation layer 130. The negotiation layer 130 may allow the semantic framework 112 to securely share information with other agents accessible via the network 160 (*e.g.,* $agent_N$ 150 discussed below).

[0038] The negotiation layer 130 may comprise a negotiation module 132, which may be configured to negotiate security and/or access control policy between the $agent_1$ 110 and another, third-party agent. For example, the negotiation module 132 may determine what, if any, access another agent ($agent_N$ 150) is given to information managed and/or controlled by the $agent_1$ 110. Similarly, the negotiation module 132 may determine whether information provided by the $agent_N$ 150 may be stored in the knowledge base 126 of the $agent_1$ 110.

[0039] In some embodiments, the negotiation module 132 may track multiple iterations of trust establishment between entities (*e.g.,* between the $agent_1$ 110 and other, external agents, such as $agent_N$ 150). Therefore, the negotiation module 132 may be configured to access negotiation ontologies, negotiation strategies, and negotiation rules applicable to a

particular entity. For instance, a data sharing negotiation between the $agent_1$ 110 and the $agent_N$ 150 may comprise evaluating credentials provided by the $agent_N$ 150, an authorization level of the $agent_N$ 150, a trust status of the $agent_N$ 150, and the like. As is discussed below, the credentials provided by the $agent_N$ 150 as part of a data sharing negotiation may be verified using an identification and trust management service 130. This may allow the $agent_1$ 110 to verify the identity of the $agent_N$ 150 (*e.g.,* verify the authenticity of messages transmitted from the $agent_N$ 150).

[0040] A data sharing negotiation between the $agent_1$ 110 and the $agent_N$ 150 may involve several iterations comprising offers and counteroffers between the entities. For example, the $agent_1$ 110 may provide access to the information it manages and/or controls to other agents (*e.g.,* $agent_N$ 150) on the network 160. The $agent_1$ 110 may impose conditions on this data sharing. For example, the $agent_1$ 110 may require that the data sharing be reciprocal, may require that the $agent_N$ 150 have particular data security measures in place to protect the information, the $agent_1$ 110 may require that the $agent_N$ 150 only allow the data to be used in a certain way (*e.g.,* according to a license), and so on. Accordingly, the negotiation module 132 may use the negotiation ontologies, strategies, and/or rules (*e.g.,* business rules) discussed above to autonomously negotiate data sharing with the $agent_N$ 150 (*e.g.,* using a matchmaking engine, such as the evaluation module 134 discussed below). Alternatively, or in addition, one or more users (*e.g.,* human actors) may intervene in the negotiation process by establishing a trust relationship, data sharing rules, and the like (*e.g.,* may bootstrap a negotiation between the $agent_1$ 110 and the $agent_N$ 150).

[0041] The evaluation module 134 may be used to evaluate whether an information sharing offer (*e.g.,* negotiation) from an external entity (*e.g.,* the $agent_N$ 150) should be accepted and/or whether a counteroffer should be returned. The evaluation module 134 may evaluate the negotiation ontology, strategy, and/or rules of the negotiation module 132 against an offer and/or information request from an external entity (*e.g.,* $agent_N$ 150) and may determine whether the offer will be accepted and/or whether a counteroffer will be returned. Accordingly, the evaluation module 134 may comprise a matchmaking engine and/or other reasoning system.

[0042] In making negotiation decisions, the evaluation module 134 may access an identification and trust management service 170 The identification and trust management service 170 may comprise and/or may be communicatively coupled to a user store 172 (*e.g.,* user identification database or the like) to allow the evaluation module 134 and/or other components within the semantic framework 112 to verify (*e.g.,* authenticate) the identity of

external, third-party entities (*e.g.,* $agent_N$ 150) and/or authentication message transmitted by the entities. For example, an $agent_N$ 150 may submit a request for information to the $agent_1$ 110. The semantic framework 112 of the $agent_1$ 110 may be used to negotiate access to the desired information with the $agent_N$ 150. Alternatively, or in addition, if a negotiation has already occurred, the semantic framework 112 may determine whether the information sharing arrangement negotiated with the $agent_N$ 150 allows for access to the requested information. As discussed above, this evaluation may be based upon a negotiation between the $agent_1$ 110 and the $agent_N$ 150 comprising a negotiation ontology, strategy, and rules. The nature of the negotiation and/or subsequent access to information may depend upon the identity of the $agent_N$ 150.

[0043] For instance, the $agent_1$ 110 may have a long-standing, established relationship with the $agent_N$ 150. As such, the $agent_N$ 150 may negotiate a relatively high level of access to the information managed and/or controlled by the $agent_1$ 110. Similarly, the $agent_1$ 110 may trust information provided by the $agent_N$ 150. Alternatively, the $agent_N$ 150 may be a relative unknown to the $agent_1$ 110. In fact, in some cases, the $agent_1$ 110 may not have any established business relationship with the $agent_N$ 150. In this case, the $agent_N$ 150 may be granted a relative lower level of access to the information managed and/or controlled by the $agent_1$ 110. However, as data is shared between the $agent_1$ 110 and the $agent_N$ 150, the information sharing policy negotiated therebetween may be adapted to reflect increased trust between the $agent_1$ 110 and the $agent_N$ 150.

[0044] The identity of the $agent_N$ 150 may be determined using a number of different mechanisms. In some embodiments, messages transmitted between the $agent_1$ 110 and the $agent_N$ 150 may comprise a signature (*e.g.,* an X.509 signature or the like). In this embodiment, the identification and trust service 130 may be communicatively coupled to a data store 132 comprising a plurality of trusted public key certificates to verify digital signatures from trusted entities. In other embodiments, the messages transmitted between the agents 110 and 150 may comprise an authentication token, such as a WS-Security token, a Kerberos ticket, a Security Assertions Markup Language (SAML) assertion, or the like.

[0045] The secure framework 112 may further comprise harvesting module 140, which may include a schema inspector 142 and data manager 144. The schema inspector 142 may provide a mapping between unstructured and/or structured data and the ontology library 124 of the $agent_1$ 110, substantially as described above (*e.g.,* using a concept mapping system or method). In addition, the schema inspector may be configured to apply a security label (*e.g.,* security predicate) to the mapped data. As discussed above, this may comprise matching

concepts identified in the incoming data (*e.g.,* RDF statements) against a security pattern (stored in the security mappings and rules module 146 discussed below). Security label(s) associated with a security pattern matching may be applied to the concept. Mapped and secured concepts may be transmitted to and stored by the knowledge base module 126.

[0046] The data manager 144 may manage the retrieval and dissemination of data by the $agent_1$ 110. The $agent_1$ 110 may be configured to acquire data from an unstructured data source, via data mining, or another data retrieval process. In addition, the data manager 144 may allow the $agent_1$ 110 to transmit requests for information to other agents via the network 160, such as the $agent_N$ 150. Similarly, the data manager 144 may allow the $agent_1$ 110 to respond to requests for information from other agents on the network 160. Accordingly, the data manager 144 may be configured to communicate with various data sources, such as databases (*e.g.,* Structured Query Language (SQL) databases, XML databases, or the like), directories (*e.g.,* X.509 directories, Lightweight Directory Access Protocol (LDAP) directories, or the like), file systems (*e.g.,* a hierarchical file system, flat file system, distributed file system, or the like), or other data storage systems. In addition, the data manager 144 may be configured to communicate with agents over the network 144 using a communication protocol, such as Remote Method Invocation (RMI), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), or another communications protocol.

[0047] A security layer 114 may be used to control the operation of the semantic framework 112 and, in particular, to apply security labels (*e.g.,* security predicates) to the information managed and/or controlled by the $agent_1$ 110, to translate between security labels of the $agent_1$ 110 and those of other, external agents (*e.g.,* $agent_N$ 150), and to negotiate security policies between the external agents.

[0048] The security layer 114 may comprise a security mappings and rules module 146. As unstructured data is retrieved by the data manager 144, it may be semantically mapped into an ontology as described above (*e.g.,* using the schema inspector 142, the ontology library 124, and/or the global ontology service 160). Similarly, semantically structured data retrieved by the data manger may be mapped to the ontology of the $agent_1$ 110. The security mapping and rules module 146 may apply security labels to the imported semantic data by applying one or more security patterns to the data (*e.g.,* security labels may be applied to the data per the security patterns discussed above). The data, along with the associated security labels (*e.g.,* security predicates) may be stored in the knowledge base 126.

[0049] The security layer 114 may further comprise a data inference module 136. As discussed above, semantic information may be expressed in various different ways. Accordingly, information inferred and/or entailed from semantic information may be given a different security label than the same data already within the knowledge base 126. If the security label applied to the inferred and/or entailed information is less-restrictive than that of the information already within the semantic framework 112, the inferred and/or entailed information may represent a security leak (*e.g.,* information that is intended to be accessible according to a "high" level security label may be accessible via a "lower" level security label). Similarly, in some cases, multiple security labels may map to a particular semantic data statement. The data inference module 136 may be configured to resolve these conflicts according to a security policy of the security layer 114 (*e.g.,* as stored in the security mapping and rules module 146).

[0050] The policy engine 128 of the security layer may be configured to interpret the policies stored in the policy module 122. As discussed above, the policy module 122 may comprise a plurality of different policy types including, but not limited to policies relating to: information sharing (*e.g.,* between agents on the network 160); trust management policies, delegation policies, privacy policies, authentication policies, authorization policies, and ontological policies, and the like. The ontological policies may be used to represent dynamic elements within the semantic framework 112, such as subjects, resources, credentials and obligations (*e.g.,* according to the ontology library 124). Accordingly, the policy 122 may comprise policy information developed according to an information sharing negotiation between the $agent_1$ 110 and external, third-party agents, such as $agent_N$ 150. This negotiated policy may define the information sharing relationship between the $agent_1$ 110 and the $agent_N$ 150, including, but not limited to: to the subjects with which information may be shared, resources to be shared (*e.g.,* including the security level available to the $agent_N$ 150), credentials required by the $agent_N$ 150, and obligations of the $agent_N$ 150 and/or $agent_1$ 110.

[0051] The policy engine 128 may be configured to interpret the policy information stored in the policy module 122 to accept credentials from external agents, such as the $agent_N$ 150, and make access decisions accordingly (*e.g.,* based on the identity of the agent, the credentials supplied by the agent, a security label of the information requested by agent, and the like).

[0052] **Figure 2** is a flow diagram of one embodiment of a method 200 for negotiating an information sharing policy between two or more agents.

[0053] At step 210, the method 200 may be initialized. This may comprise accessing information required for the negotiation process, such as a negotiation ontology, negotiation strategy, and/or negotiation rules. The initialization of step 210 may further comprise accessing a global ontology service 160 and/or identification and trust service, such as the services 160 and 170 of **Figure 1**. These services may allow the method 200 to translate semantic information from an external agent into a domain-specific ontology accessible to the method 200 (*e.g.,* negotiation ontology of the method 200). The identification and trust service may allow the method 200 to authenticate messages transmitted from external entities.

[0054] At step 220, the method 200 may receive a request for information managed and/or controlled by the method 200. The request may comprise a request to establish an information sharing policy between the semantic framework of the method 200 and an external entity.

[0055] At step 230, the method 200 may determine an identity of the submitter of the request. This may comprise accessing identifying information stored in the message (*e.g.,* a digital signature, key, password, SAML assertion, or the like). In addition, authenticating the identity of the requester may comprise accessing an identification and trust management service (*e.g.,* an OpenID service, a certificate authority, or the like).

[0056] The identity of the requester may be unknown to the method 200 (*e.g.,* this may be the first information request from the requester). In this case, the method 200 may dynamically identify the requester using a dynamic identification system, such as OpenID discussed above, or other federated identity service (*e.g.,* a trusted root certification authority, or other dynamic credential issuance and verification framework). This may allow the method 200 to provide access to requesters without manually establishing a trust relationship between the method 200 and the requester.

[0057] At step 240, the method 200 may access an information sharing policy associated with the requester. In some embodiments, several iterations of trust establishment may be required between the method 200 and the other entity before a mutually accepted level of trust is established therebetween. The trust level may evolve over time as the method 200 interacts with particular entities. For example, new, relatively unknown entities (*e.g.,* requesters) may be granted a relatively low level of access to the information managed and/or controlled by the method 200. This access level may be expressed within the context of a negotiation ontology as negotiation strategies and/or rules. However, as the method 200 successfully interacts with the entity and/or by manual intervention, the trust level with the

particular entity may increase (*e.g.,* the requester may be granted additional rights to the information managed and/or controlled by the method 200). In some embodiments, the access level of the entity may be modified according to a negotiation process (described below). For instance, a requester may be granted increased access by allowing the method 200 access to information managed and/or controlled by the requesting entity (*e.g.,* by establishing a reciprocal information sharing relationship). Similarly, information access may be predicated upon an agreement (*e.g.,* license agreement) between the method 200 and the requester that the data will be used in a prescribed manner (*e.g.,* as read only, not disseminated to other entities, and so on). One of skill in the art would recognize that an information sharing policy between the method 200 and the requester could be modified and/or predicated on any number of different factors.

[0058] If the method 200 determines that an information sharing policy exists, the flow may continue to step 260, where the method 200 may negotiate the request of step 220; otherwise, the flow may continue to step 250, where a new information sharing policy for the requester may be established.

[0059] At step 250, a new information sharing policy for the requester may be initialized. The information sharing policy may be based upon the negotiation ontology, strategy, and rules discussed above. In some embodiments, the information sharing policy of step 250 may be initialized according to the identity of the requester established at step 230. For example, the requester may be known to the method 200 in another context (*e.g.,* in a business context, such as in a vendor-vendee relationship). The information sharing policy for the requester may be initialized according to the nature of the pre-existing relationship (*e.g.,* as defined by the negotiation ontology, strategy, and/or rules). Similarly, an unknown requester whose identity and/or trust level is established by a highly trusted source (*e.g.,* such as an identification and trust management service, such as the service 170 of **Figure 1**) may affect the nature of the information sharing policy initialized at step 250 (*e.g.,* a requester whose identity is verified by a trusted source may be granted a higher level of initial access, than a requester whose identity is self-verified and/or verified by a less-trusted source). As would be appreciated by one of skill in the art, the negotiation ontology, strategy, and rules discussed above could be used to initialize the information sharing policy based on virtually any property in the request of step 220 and/or the identity of the requester established at step 230.

[0060] At step 260, the method 200 may negotiate the request according to the information sharing policy. In some embodiments, the negotiation of step 260 may comprise

comparing one or more security labels (*e.g.,* security predicates) applied to information requested by the request of step 220 to the information sharing policy obtained at step 250 or 252. The negotiation may further comprise determining the nature of the access sought by the requester (*e.g.,* read-only access, read and disseminate, terms of use, etc.). Negotiating the information request at step 260 may comprise accessing a negotiation ontology, a negotiation strategy, and one or more negotiation rules as described above. In addition, the negotiation may comprise determining the nature the request received at step 220. For instance, the method 200 may determine a security label (*e.g.,* security predicate) associated with the information requested by the requester. As such, the method 200 may map the requested data onto the ontology of the method 200 (*e.g.,* to the ontology library 124 of **Figure 1**). Similarly, the negotiation may require determining the nature of the access requested (*e.g.,* read access, write access, access according to a license, and so on). The requested access type may be mapped to the negotiation ontology. The negotiation may further comprise evaluating the terms under which the information is to be shared (*e.g.,* licensing terms, or the like), whether the information sharing is reciprocal, or the like.

[0061] In some embodiments, the negotiation of step 260 may comprise a simple comparison to result in a "yes" or "no" decision. In other embodiments, the negotiation of step 260 may comprise the communication of one or more offers and counteroffers, wherein the information to be shared, how the information is to be shared, the terms under which the information is to be shared, and the like may be negotiated.

[0062] **Figure 3** depicts one embodiment of a method 300 for negotiating access to semantically secured information. At step 305, the method 300 may be initialized substantially as discussed above. The initialization may comprise accessing an ontology library associated with the method 300 (*e.g.,* the ontology library 124 of **Figure 1**), accessing a negotiation ontology, strategy, and/or rules (*e.g.,* in the policy 122 of **Figure 1**), accessing one or more security mappings and rules (*e.g.,* in module 146 of **Figure 1**), and the like.

[0063] At step 310, a request for information managed and/or controlled by the method 300 may be received. An information sharing policy associated with the requester may also be received and/or obtained at step 310. The information sharing policy may be obtained and/or generated substantially as described above in conjunction with **Figure 2**.

[0064] At step 320, the method 300 may determine whether the request of step 310 requests information semantically or in an unstructured format. The determination of step 320 may allow the method 300 to map the requested information to an ontology of the method 300. For example, the information request of step 310 may refer to requested

information semantically (*e.g.,* to data within a particular ontology, in RDF format, in OWL, or the like), or, alternatively, the request may be expressed in an unstructured format (*e.g.,* as a natural language search, in text terms, or the like). If the information request is expressed semantically, the flow may continue to step 322; otherwise, the flow may continue to step 324.

[0065] At step 322, the semantic information request may be mapped to the ontology of the method 300 (*e.g.,* to the ontology library 124 of **Figure 1**). If the method 300 uses a sparse and/or domain-specific ontology, the mapping of step 322 may comprise mapping the semantic data in the request to the domain-specific ontology of method 300 (*e.g.,* the ontology library 124 of **Figure 1**). In some embodiments, the method 300 may access a global ontology service, such as the global ontology service 160 of **Figure 1** to perform the mapping. The global ontology service may be used to translate between an ontology used by the requester to a domain-specific and/or sparse ontology of method 300. Alternatively, or in addition, the method 300 may perform the mapping using other techniques (*e.g.,* term matching, disambiguation, or the like).

[0066] At step 324, the unstructured data in the information request may be mapped to an ontology of the method 300. The mapping of step 326 may comprise a natural language to ontology mapping (*e.g.,* concept mapping) substantially as discussed above.

[0067] At step 330, one or more security labels associated with the information requested by request of step 310 may be obtained (*e.g.,* security labels assigned to the semantic data determined at step 322 or 324). As discussed above, semantic information managed and/or controlled by the method 300 may comprise respective security labels (*e.g.,* security predicates) applied using one or more security patterns. Upon mapping the request data to the ontology of method 300, these security patterns may be applied to the semantic information mapped at step 322 or 324. The resulting security labels may be obtained at step 330 and used in the negotiation process of steps 350-370 discussed below.

[0068] At step 340, the method 300 may determine whether the request of step 310 requests an access level to the information in a semantic or unstructured format. The access level may indicate the nature of the access requested at step 310. For example, a "read" access request may allow the requester to read information managed and/or controlled by the method 300, but not use the information in other ways (*e.g.,* disseminate and/or modify the information). "Read/write" access may imply the ability to modify information managed and/or controlled by the method 300. Similarly, a "read and disseminate" access may allow the requester to read the information as well as make the information available to other

entities. Various different requesters may have different terms and/or means for expressing the various requested access levels (*e.g.,* as an Access Control List (ACL), as Unix access bit mask, or the like). In addition, the access level of step 330 may comprise terms-of-use and/or licensing information relating to the information (*e.g.,* how the requester may use the information, etc.).

[0069] This access level information may be expressed semantically and/or in an unstructured format. If the information is expressed semantically, the flow may continue to step 342; otherwise, the flow may continue to step 344.

[0070] At step 342, the semantic access-level information in the request may be mapped to an ontology of the method 300 (*e.g.,* to the negotiation ontology). This may allow the method 300 to determine whether the requested access level should be granted (*e.g.,* at step 350). The mapping of step 342 may be performed substantially as described above in conjunction with step 322 and may comprise accessing a global ontology service to translate between a negotiation ontology (or other ontology) of the method 300 and an ontology used in the request.

[0071] At step 344, the unstructured (*e.g.,* natural text or other format) access information may be mapped to the negotiation ontology (or other ontology) of the method 300. The mapping may comprise a concept mapping process discussed above (*e.g.,* may comprise term matching, disambiguation, and the like).

[0072] Although **Figure 3** depicts translating particular portions of the request to an ontology of method 300 (*e.g.,* the information referred to by the request and the requested access level to the information), one skilled in the art would recognize the request could comprise additional information which could similarly be translated (*e.g.,* mapped) to the ontology of the method 300. Such information could include, but is not limited to: an identity of the requester, an attribute of the requester (*e.g.,* an employee identifier of the requester, a social security number of the requester, contact information, etc.), a "role" of the requester within an organization (*e.g.,* human resources, claims processing, etc.), reciprocal permissions to access data managed and/or controlled by the requester, and the like. As such, this disclosure should not be read as limited to any particular set of information with a request (*e.g.,* within the request of step 310) and/or any particular semantic mapping of request information.

[0073] At step 350, the semantic information from the request may be used (along with the information sharing policy, negotiation ontology, strategy and rules) to determine whether access to the information, under the proposed terms, should be granted. This determination

of step 350 may comprise comparing the security label (*e.g.,* security predicate) applied to the information sought by the requester (obtained at step 330) and/or a semantic access level and/or usage terms sought by the requester (obtained at step 342 or 344), along with other information in the request, to the information sharing policy of the requester. As discussed above, in some embodiments, the negotiation evaluation of step 350 may be performed by a reasoning engine (*e.g.,* matchmaking engine).

[0074] The evaluation of step 350 may be determined by the information sharing policy of the requester and/or negotiation policy, strategy, and/or rules of the method 300. For example, the information sharing policy and/or negotiation rules, may indicate that the requester is allowed read access to semantic information having a particular security label (*e.g.,* a "public" security label). As such, if the security label on the information sought by the request is not "public," the request may be rejected (as discussed below). Similarly, information may include a security label indicating that the information is only available to particular entities (*e.g.,* personal information about a person may only be available to particular requesters, such as the person him/herself, an entity having a particular role (*e.g.,* human resources), or the like). Similarly, the security label of the information may indicate that particular types of information may only be accessed under particular terms (*e.g.,* as "read-only," under a particular license, or the like). As could be appreciated by one of skill in the art, any number of different security labels, information sharing policy information, and/or request attributes, could be used in the semantic evaluation of step 350.

[0075] If the evaluation of step 350 indicates that the information may be shared according to the parameters within the request (*e.g.,* the access level and/or terms-of-use within the request of step 310), the flow may continue to step 360; otherwise, the flow may continue to step 352.

[0076] At step 352, a counteroffer may be transmitted to the requester. The counteroffer may indicate what should be in the request of step 310 to cause the method 300 to allow the method 300 to accept the request. The counteroffer may be established using the negotiation ontology, strategy, and rules discussed above. For example, the negotiation strategy and rules may indicate that in order to provide access to information to a particular entity, the entity must agree to use the information under a particular set of licensing terms. The counteroffer may comprise the licensing terms and indicate that if the request is re-submitted with the terms, the request may be accepted. Alternatively, or in addition, the counteroffer may indicate that particular entity may only access information if reciprocal access is granted to another entity (*e.g.,* information may be shared if, and only if, the entity allows its

information to be shared). Similarly, the counteroffer may indicate that the information requested is secured and/or private and that only entities having a higher trust level may access it. In this case, the counteroffer may indicate one or more ways in which the entity may increase its trust level (*e.g.,* by using a better authentication mechanism, using an out-of-band exchange, or the like). One skilled in the art could recognize that the counteroffer of step 352 could comprise any modification to the request known in the art (*e.g.,* information type, information access level, licensing agreement, payment, etc.). As such, this disclosure should not be read as limited to any particular type or types of counteroffers.

[0077] Also at step 352, the counteroffer may be returned to the requester via a network transmission (*e.g.,* message), out-of-band communication, or the like.

[0078] At step 360, the method 300 may receive a response to the counteroffer. The response may comprise an indication of the acceptance of one or more terms of the counteroffer of step 352 and/or a request for information. The request for information in the response of step 360 may be processed at step 310 substantially as described above.

[0079] At step 370, the information requested by the request received at step 310 may be provided to the requester. This may comprise transmitting a message comprising the information to the requester via a network (or other data transmission means). Alternatively, providing the data may comprise providing the requester with an authentication and/or authorization credential to allow the requester to access the data at another time and/or using another communications mechanism.

[0080] **Figure 4** is a flow diagram of one embodiment of a method 400 for requesting semantic information from another entity. At step 405, the method 400 may be initialized, which may comprise accessing one or more information sharing policies, a negotiation policy, strategy, and/or rules substantially as described above.

[0081] At step 410, a request for information may be generated. In some embodiments, the request for information may be tailored to a particular entity (*e.g.,* to $Agent_N$ 150 of **Figure 1**). In this case, the contents of the request may conform to an information policy established and/or negotiated with the entity. The information sharing policy may indicate the nature of the access available to the method 400 (*e.g.,* read, write, disseminate, terms-of-use, and the like). Similarly, the information sharing policy may indicate the nature of the information the method 400 may request, how the information is to be expressed (*e.g.,* unstructured, semantically, in a particular ontology, or the like). If no information sharing policy for the entity exists and/or the information request is not directed to any particular entity (*e.g.,* the information request is to be "broadcast" to a plurality of entities), the

information request 410 may be general. A generalized information request may be refined during a negotiation process described below in conjunction with step 430-470.

[0082] The information request may indicate information requested by the method 400. The information request may indicate the requested information semantically and/or as an unstructured format (*e.g.,* as a natural language search or the like). Information requested semantically may be expressed within the ontology of the method 400 (*e.g.,* in a domain-specific ontology of the method 400, in a global ontology, within an ontology of the requestee, or the like). Similarly, the desired access level, the terms-of-use, and the like may be similarly expressed. An unstructured request may comprise natural language search terms, access control primitives, license terms, or the like.

[0083] At step 420, the information request may be transmitted and/or broadcast to one or more entities and one or more responses may be received. Each response may be processed as described below in steps 430-470.

[0084] At step 430, the method 400 may determine whether the response is an acceptance of the request. The determination of step 430 may comprise examining the response to determine whether the response includes the information requested at step 420. Alternatively, the response may comprise a key, token, or the like allowing the method to retrieve the information using another communications method (*e.g.,* via a secure back-channel connection, such as mutually authenticated SSL or the like). If the response is an acceptance, the flow may continue to step 432.

[0085] Alternatively, portions of the response may comprise a counteroffer and/or rejection. A counteroffer and/or rejection may be processed at step 440.

[0086] At step 440, the terms of the counteroffer and/or rejection may be mapped to a negotiation ontology of the method 400. For example, a counteroffer may include licensing terms, which must be accepted for access to the requested information. The negotiation ontology may comprise a concept or other semantic data corresponding to the license. Similarly, the counteroffer and/or rejection may comprise an access level under which the information is available (*e.g.,* only as "read-only"). This access level may be similarly mapped to the negotiation ontology. As could be appreciated by one of skill in the art, any counteroffer and/or rejection property and/or attribute could be mapped to the negotiation ontology including, but not limited to: a reciprocity request, a security label (*e.g.,* security predicate) indicator, a trust-level requirement, a payment requirement, or the like.

[0087] At step 450, the counteroffer and/or rejection may be evaluated to determine whether the terms in the counteroffer and/or rejection may be accepted. This may comprise

comparing the terms of the counteroffer and/or rejection (as expressed in the negotiation ontology) to one or more negotiation rules and/or strategies of the method 400. If the terms of the counteroffer and/or rejection are not acceptable and/or may not be modified to be acceptable, the flow may continue to step 462, where the request may be terminated; otherwise, the flow may continue to step 470.

[0088] At step 470, the negotiation strategy and/or rules may be used to respond to the counteroffer and/or rejection (*e.g.,* using a reasoning and/or matchmaking engine). The response of step 470 may comprise acceptance of the terms of the counteroffer (*e.g.,* by modifying a requested access-level, agreeing to terms-of-service, agreeing to provide reciprocal information access, providing a higher level of authentication credential, transferring a payment to the entity, or the like). Alternatively, or in addition, the response may modify one or more of the terms of the counteroffer and/or rejection (*e.g.,* propose an intermediary access level to the data, propose a different payment amount, propose an alternative reciprocal information sharing arrangement, or the like). As can be appreciated by one of skill in the art, any term within the counteroffer and/or rejection of step 420 could be modified in the response of step 470.

[0089] At step 475, the response to the counteroffer and/or rejection generated at step 470 may be transmitted and/or otherwise made available to the entity and a response thereto may be received. Upon receiving a response, the flow may continue to step 430, where the response may be further processed as described above (*e.g.,* may be further negotiated if required).

[0090] The above description provides numerous specific details for a thorough understanding of the embodiments described herein. However, those of skill in the art will recognize that one or more of the specific details may be omitted, or other methods, components, or materials may be used. In some cases, operations are not shown or described in detail.

[0091] Furthermore, the described features, operations, or characteristics may be combined in any suitable manner in one or more embodiments. It will also be readily understood that the order of the steps or actions of the methods described in connection with the embodiments disclosed may be changed as would be apparent to those skilled in the art. Thus, any order in the drawings or Detailed Description is for illustrative purposes only and is not meant to imply a required order, unless specified to require an order.

[0092] Embodiments may include various steps, which may be embodied in machine/computer-executable instructions stored on a memory configured to be executed by

a processor within a general-purpose or special-purpose computing device, such as a personal computer, a laptop computer, a mobile computer, a personal digital assistant, smart phone, or the like. Alternatively, the steps may be performed by hardware components that include specific logic for performing the steps, or by a combination of hardware, software, and/or firmware.

[0093] Embodiments may also be provided as a computer program product including a computer-readable medium having stored instructions thereon that may be used to program a computer (or other electronic device) to perform processes described herein. The computer-readable medium may include, but is not limited to: hard drives, floppy diskettes, optical disks, CD-ROMs, DVD-ROMs, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, solid-state memory devices, or other types of media/machine-readable medium suitable for storing electronic instructions.

[0094] As used herein, a software module or component may include any type of computer instruction or computer executable code located within a memory device and/or transmitted as electronic signals over a system bus or wired or wireless network. A software module may, for instance, comprise one or more physical or logical blocks of computer instructions, which may be organized as a routine, program, object, component, data structure, etc., that perform one or more tasks or implements particular abstract data types.

[0095] In certain embodiments, a particular software module may comprise disparate instructions stored in different locations of a memory device, which together implement the described functionality of the module. Indeed, a module may comprise a single instruction or many instructions, and may be distributed over several different code segments, among different programs, and across several memory devices. Some embodiments may be practiced in a distributed computing environment where tasks are performed by a remote processing device linked through a communications network. In a distributed computing environment, software modules may be located in local and/or remote memory storage devices. In addition, data being tied or rendered together in a database record may be resident in the same memory device, or across several memory devices, and may be linked together in fields of a record in a database across a network.

[0096] It will be understood by those having skill in the art that many changes may be made to the details of the above-described embodiments without departing from the underlying principles of this disclosure. The scope of the present invention should, therefore, be determined only by the following claims.

WHAT IS CLAIMED IS:

1. A method for sharing semantic information performed by a system including a processor and a memory communicatively coupled to the processor, the memory storing instructions executable by the processor to cause the system to perform the method, the method comprising:

applying semantic security labels to semantic data in a data storage medium;

receiving a request by a requester to access a portion of the semantic data;

obtaining a semantic security label of the portion of the semantic data;

comparing the semantic security label to an information sharing policy;

determining whether the portion of the semantic data can be accessed by the requester based on the semantic security label and the information sharing policy;

providing access to the portion of the semantic data based on the determination; and

responding to the request with a counteroffer if the information sharing policy disallows access to the requester.

2. The method of claim 1, wherein the method further includes receiving the information sharing policy, the information sharing policy being associated with the requester.

3. The method of claim 1, wherein the request to access a portion of the semantic data is expressed semantically.

4. There method of claim 1, wherein the request to access a portion of the semantic data is expressed in an unstructured format.

5. The method of claim 1, wherein the method further includes mapping the request to an ontology included in an ontology library.

6. The method of claim 5, wherein the semantic security label of the portion of the semantic data is obtained based at least in part on the mapping of the request to the ontology included in the ontology library.

7. The method of claim 1, wherein the semantic security label is applied to the portion of the semantic data using at least one security pattern.

8. The method of claim 1, wherein the request to access a portion of the semantic data includes information identifying the portion of the semantic data and information related to the nature of the request.

9. The method of claim 8, wherein the information related to the nature of the request includes a requested access level.

10. The method of claim 8, wherein the information related to the nature of the request includes information related to the identity of the requester.

11. The method of claim 8, wherein the information related to the nature of the request level is expressed semantically.

12. The method of claim 8, wherein the information related to the nature of the request is expressed in an unstructured format.

13. The method of claim 8, wherein the method further includes mapping the information related to the nature of the request to a negotiation ontology.

14. The method of claim 13, wherein determining whether the portion of the semantic data can be accessed by the requester is further based on the mapping of the information related to the nature of the request to a negotiation ontology.

15. A system for sharing semantic information, the system comprising:

a processor; and

a memory communicatively coupled to the processor storing instructions that when executed by the processor cause the processor to perform a method comprising:

applying semantic security labels to semantic data in a data storage medium;

receiving a request by a requester to access a portion of the semantic data;

obtaining a semantic security label of the portion of the semantic data;

comparing the semantic security label to an information sharing policy;

determining whether the portion of the semantic data can be accessed by the requester based on the semantic security label and the information sharing policy;

providing access to the portion of the semantic data based on the determination; and

responding to the request with a counteroffer if the information sharing policy disallows access to the requester.

16. The system of claim 15, wherein the method further includes receiving the information sharing policy, the information sharing policy being associated with the requester.

17. The system of claim 15, wherein the request to access a portion of the semantic data is expressed semantically.

18. There system of claim 15, wherein the request to access a portion of the semantic data is expressed in an unstructured format.

19. The system of claim 15, wherein the method further includes mapping the request to an ontology included in an ontology library.

20. The system of claim 19, wherein the semantic security label of the portion of the semantic data is obtained based at least in part on the mapping of the request to the ontology included in the ontology library.

21. The system of claim 15, wherein the semantic security label is applied to the portion of the semantic data using at least one security pattern.

22. The system of claim 15, wherein the request to access a portion of the semantic data includes information identifying the portion of the semantic data and information related to the nature of the request.

23. The system of claim 22, wherein the information related to the nature of the request includes a requested access level.

24. The system of claim 22, wherein the information related to the nature of the request includes information related to the identity of the requester.

25. The system of claim 22, wherein the information related to the nature of the request level is expressed semantically.

26. The system of claim 22, wherein the information related to the nature of the request is expressed in an unstructured format.

27. The system of claim 22, wherein the method further includes mapping the information related to the nature of the request to a negotiation ontology.

28. The system of claim 27, wherein determining whether the portion of the semantic data can be accessed by the requester is further based on the mapping of the information related to the nature of the request to a negotiation ontology.

29. A computer-readable medium storing instructions that when executed by a processor cause a computer system to perform a method for sharing semantic information, the method comprising:

applying semantic security labels to semantic data in a data storage medium;

receiving a request by a requester to access a portion of the semantic data;

obtaining a semantic security label of the portion of the semantic data;

comparing the semantic security label to an information sharing policy;

determining whether the portion of the semantic data can be accessed by the requester based on the semantic security label and the information sharing policy;

providing access to the portion of the semantic data based on the determination; and

responding to the request with a counteroffer if the information sharing policy disallows access to the requester.

30. The computer-readable medium of claim 29, wherein the method further includes receiving the information sharing policy, the information sharing policy being associated with the requester.

31. The computer-readable medium of claim 29, wherein the request to access a portion of the semantic data is expressed semantically.

32. There computer-readable medium of claim 29, wherein the request to access a portion of the semantic data is expressed in an unstructured format.

33. The computer-readable medium of claim 29, wherein the method further includes mapping the request to an ontology included in an ontology library.

34. The computer-readable medium of claim 33, wherein the semantic security label of the portion of the semantic data is obtained based at least in part on the mapping of the request to the ontology included in the ontology library.

35. The computer-readable medium of claim 29, wherein the semantic security label is applied to the portion of the semantic data using at least one security pattern.

36. The computer-readable medium of claim 29, wherein the request to access a portion of the semantic data includes information identifying the portion of the semantic data and information related to the nature of the request.

37. The computer-readable medium of claim 36, wherein the information related to the nature of the request includes a requested access level.

38. The computer-readable medium of claim 36, wherein the information related to the nature of the request includes information related to the identity of the requester.

39. The computer-readable medium of claim 36, wherein the information related to the nature of the request level is expressed semantically.

40. The computer-readable medium of claim 36, wherein the information related to the nature of the request is expressed in an unstructured format.

41. The computer-readable medium of claim 36, wherein the method further includes mapping the information related to the nature of the request to a negotiation ontology.

42. The computer-readable medium of claim 41, wherein determining whether the portion of the semantic data can be accessed by the requester is further based on the mapping of the information related to the nature of the request to a negotiation ontology.

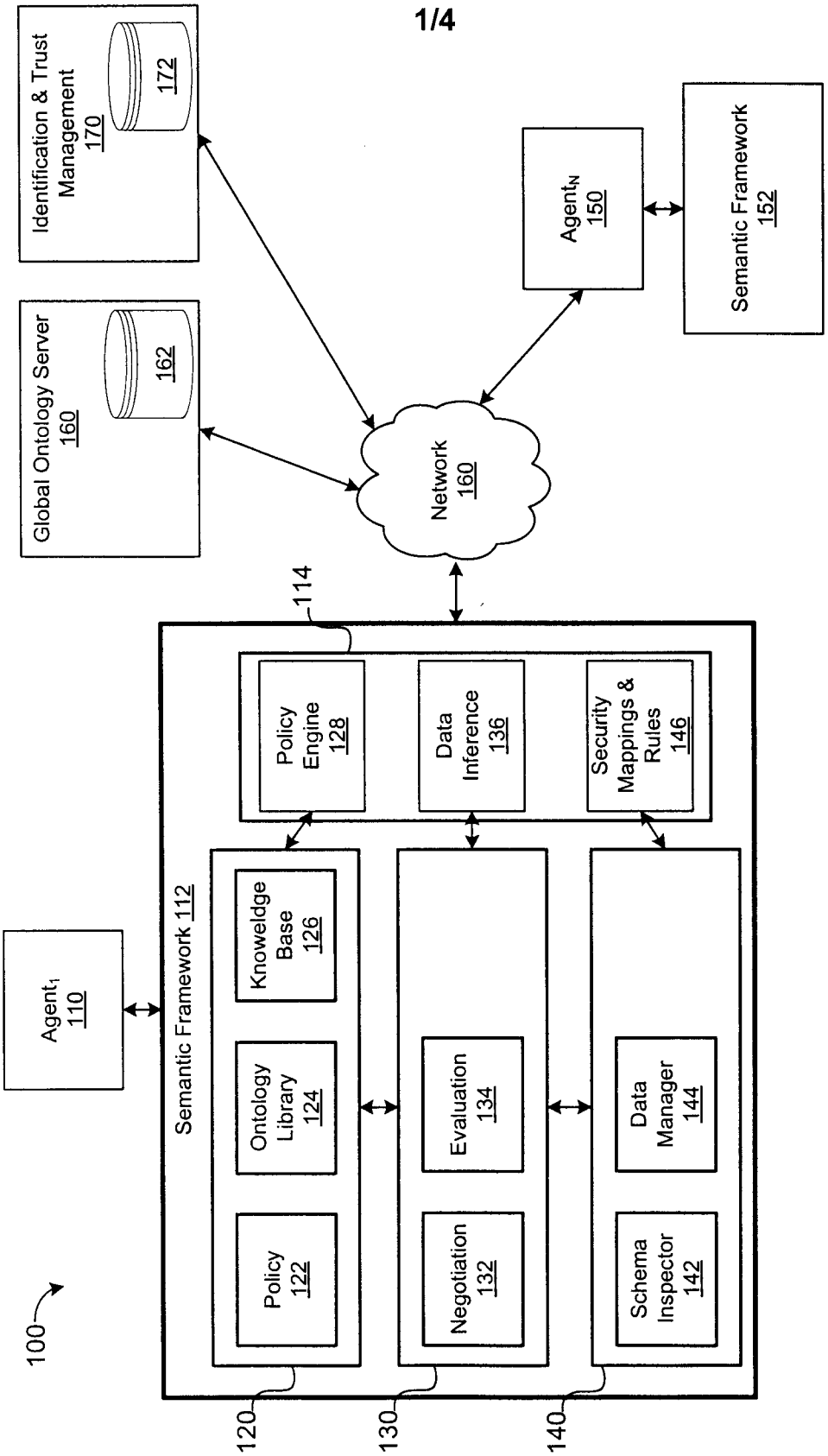

Identification & Trust Management 170
172
Global Ontology Server 160
162
Network 160
Agent$_N$ 150
Semantic Framework 152
114
Agent$_1$ 110
Semantic Framework 112
Policy Engine 128
Data Inference 136
Security Mappings & Rules 146
Knowledge Base 126
Ontology Library 124
Policy 122
Evaluation 134
Negotiation 132
Data Manager 144
Schema Inspector 142
100
120
130
140

Figure 1

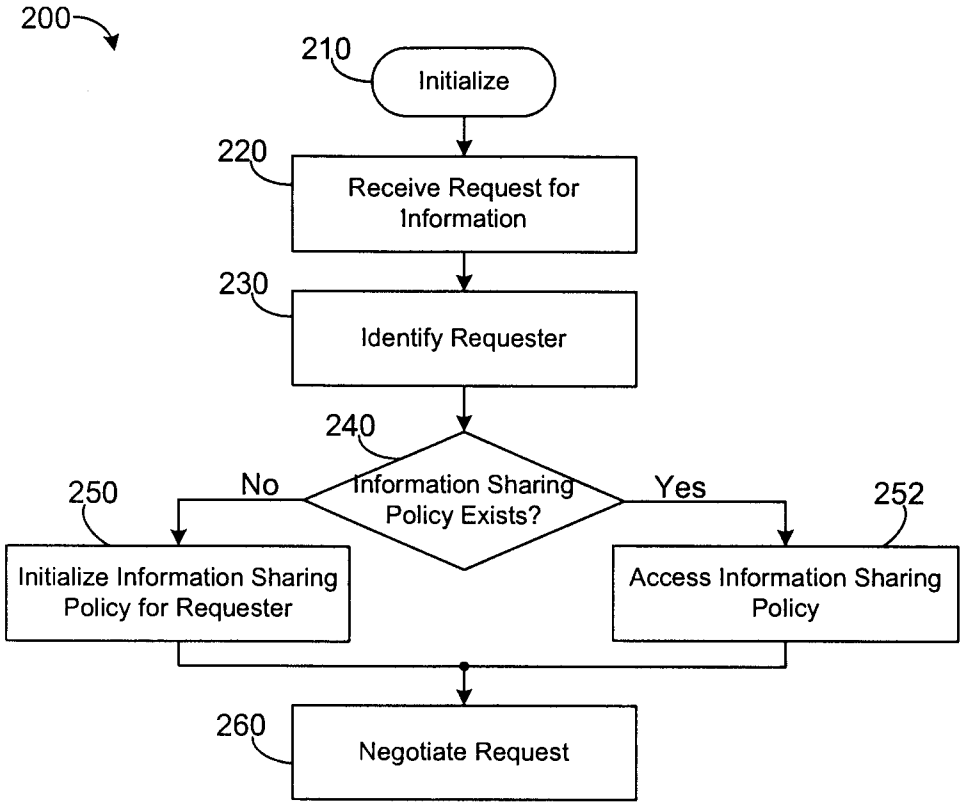

200
210
Initialize
220
Receive Request for Information
230
Identify Requester
240
Information Sharing Policy Exists?
No
Yes
250
Initialize Information Sharing Policy for Requester
252
Access Information Sharing Policy
260
Negotiate Request

Figure 2

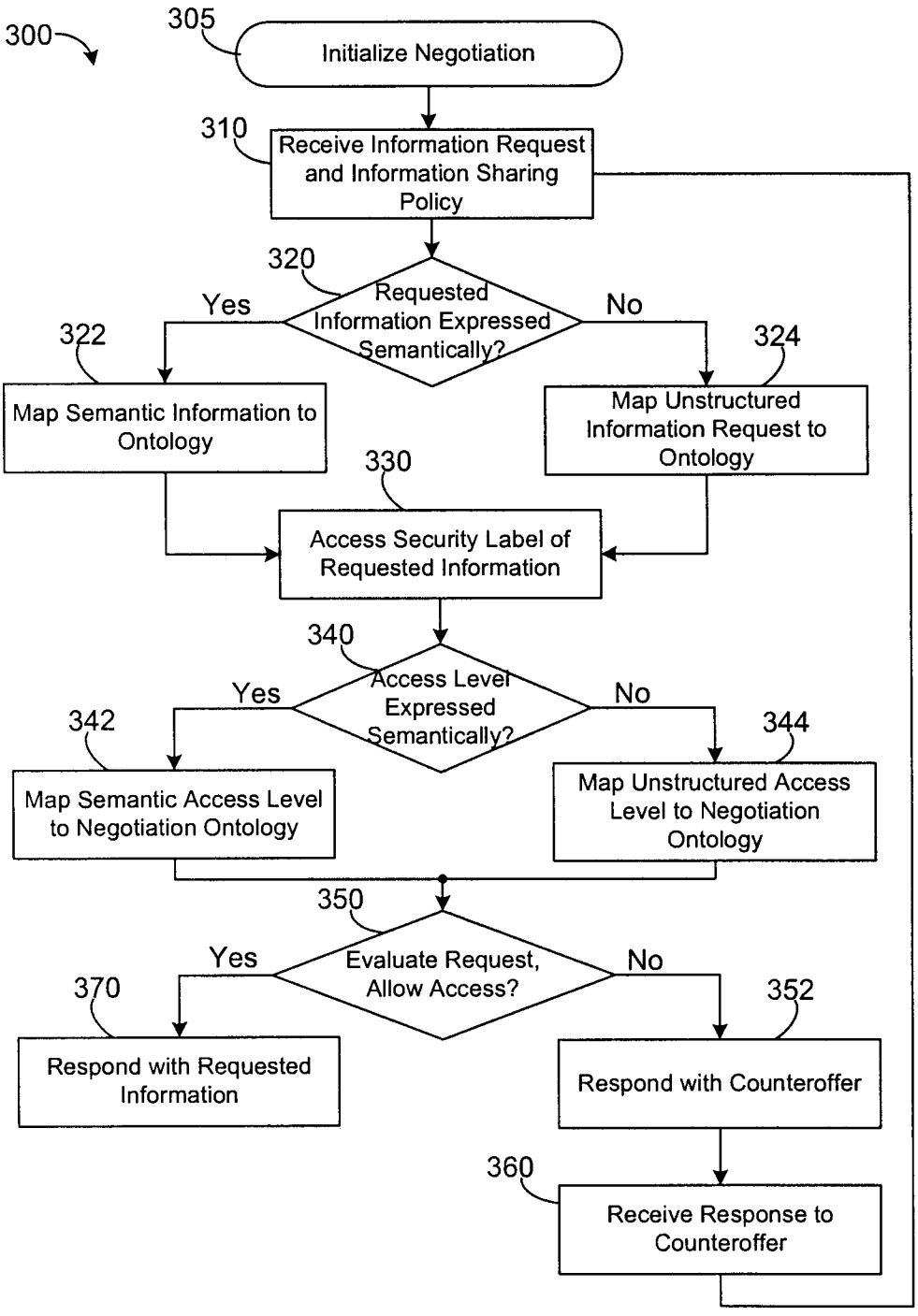

300
305
Initialize Negotiation
310
Receive Information Request and Information Sharing Policy
320
Requested Information Expressed Semantically?
Yes
No
322
Map Semantic Information to Ontology
324
Map Unstructured Information Request to Ontology
330
Access Security Label of Requested Information
340
Access Level Expressed Semantically?
Yes
No
342
Map Semantic Access Level to Negotiation Ontology
344
Map Unstructured Access Level to Negotiation Ontology
350
Evaluate Request, Allow Access?
Yes
No
370
Respond with Requested Information
352
Respond with Counteroffer
360
Receive Response to Counteroffer

Figure 3

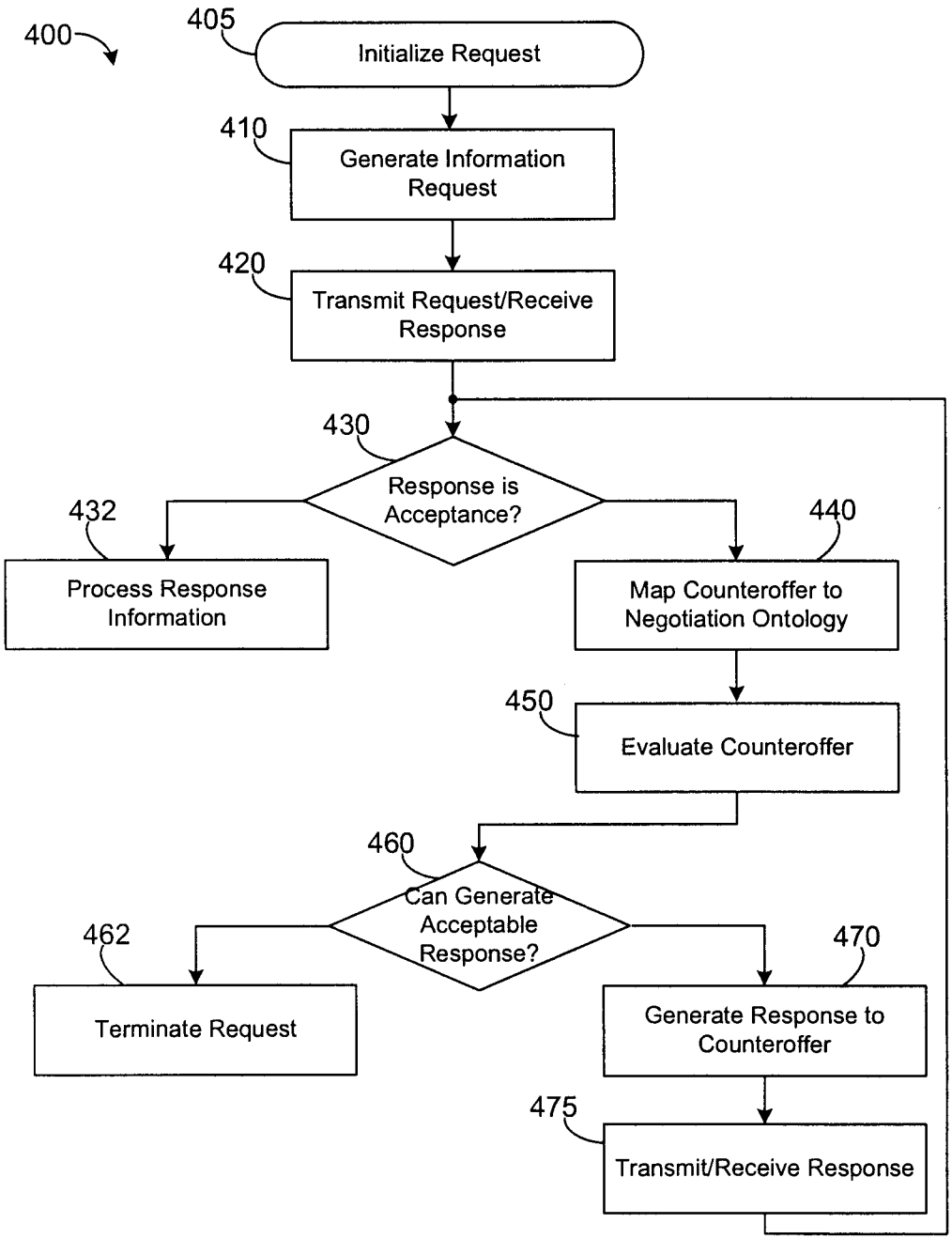

400
405
Initialize Request
410
Generate Information Request
420
Transmit Request/Receive Response
430
Response is Acceptance?
432
Process Response Information
440
Map Counteroffer to Negotiation Ontology
450
Evaluate Counteroffer
460
Can Generate Acceptable Response?
462
Terminate Request
470
Generate Response to Counteroffer
475
Transmit/Receive Response

Figure 4