



(51) International Patent Classification:

H04L 29/12 (2006.01) **G06F 21/00** (2006.01)
G06F 17/30 (2006.01) **H04L 29/06** (2006.01)

(21) International Application Number:

PCT/FI2009/050640

(22) International Filing Date:

29 July 2009 (29.07.2009)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

12/231,291 28 August 2008 (28.08.2008) US

(71) Applicant (for all designated States except US): **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FI-02150 Espoo (FI).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **OLIVER, Ian** [GB/FI]; Hitävägen 153, FI-01150 Söderkulla (FI). **HONKOLA, Jukka** [FI/FI]; Ensontie 5A, FI-02140 Espoo (FI).

(74) Agent: **NOKIA CORPORATION**; IPR Department, Virpi Tognetty, Keilalahdentie 4, FI-02150 Espoo (FI).

(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, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: METHOD, APPARATUS AND COMPUTER PROGRAM TO GENERATE A UNIQUE NODE IDENTIFIER

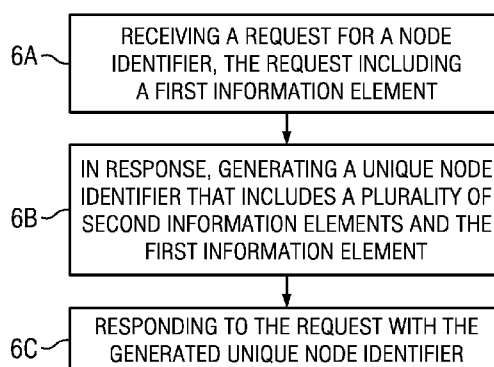


FIG. 6

(57) Abstract: A method includes receiving a request for a node identifier, the request including a first information element. In response, the method further includes generating a unique node identifier that includes a plurality of second information elements and the first information element; and responding to the request with the generated unique node identifier. Also disclosed are apparatus for performing the method, as well as a computer program embodied on a computer-readable media for executing the method.

METHOD, APPARATUS AND COMPUTER PROGRAM TO GENERATE A UNIQUE NODE IDENTIFIER

TECHNICAL FIELD:

5

The exemplary and non-limiting embodiments of this invention relate generally to the generation of a unique identifier for a node, such as one bidirectionally coupled with an information-containing system, such as the semantic web.

10 BACKGROUND:

Various abbreviations that appear in the specification and/or in the drawing figures are defined as follows:

15	GSM	global system for mobile communication, a mobile telecommunication protocol
	3G	third generation, descriptive of a class of mobile telecommunication protocols
	3GPP	3G positioning protocol (part of the 3G protocol)
	GPS	global positioning system
	IP	internet protocol
20	RDF	resource description framework
	SIB	semantic information broker
	SIM	subscriber identity module
	SSAP	smart space application protocol
	TCP	transmission control protocol
25	UUID	universal unique identifier
	URI	uniform resource identifier
	WLAN	wireless local area network
	WWW	world wide web
	XML	extensible markup language

30

The "Semantic Web" in its current form represents information that is either static or monotonically changing. It has its roots in the WWW and thus takes on a "web-wide" persona. The information and ontologies used have standard web-wide semantics grounded in generally accepted real-world concepts.

35

Much of the usage of information, that will invariably become part of the Semantic Web, will be

personal and/or local. The semantics, structure and adherence to real-world concepts will be grounded by the local users of that information, thus resulting in many localized Semantic Webs.

These local Semantic Webs will contain information that is highly dynamic, non-monotonically changing, adhering loosely (if at all) to their stated ontologies, or not to any standardized, written and commonly understood ontology, and will behave and be reasoned about according to localized, non-standard and non-intuitive logics.

The dynamicity and monotonicity of information in and forming the Semantic Web will vary depending upon locality and be organized according to person, usage, etc., as what may be termed as "spaces" or, more descriptively, as "SmartSpaces". Of particular interest herein are such small, localized information or knowledge spaces by which persons, via autonomous 'agents', interact through control-flow-free mechanisms.

A problem that arises is how to generate unique and meaningful identifiers for nodes or agents in a SmartSpace-based system, such that these unique identifiers can be read (through decryption if necessary) for information that cannot be readily faked or misrepresented. The information derived from such identifiers may be used, for example, for identification, trust and policy enforcement purposes.

SUMMARY

The foregoing and other problems are overcome, and other advantages are realized, by the use of the exemplary embodiments of this invention.

In a first aspect thereof the exemplary embodiments of this invention provide a method that includes receiving a request for a node identifier, the request including a first information element; in response, generating a unique node identifier that includes a plurality of second information elements and the first information element; and responding to the request with the generated unique node identifier.

In a further aspect thereof the exemplary embodiments of this invention provide a computer-readable memory media that stores a computer program instructions, where execution of the computer program instructions results in the performance of operations that comprise receiving a request for a node identifier, the request including a first information element; in response, generating a unique node identifier that includes a plurality of second information elements and

the first information element; and responding to the request with the generated unique node identifier.

In another aspect thereof the exemplary embodiments of this invention provide an apparatus that
5 comprises a receiver configured to receive a request for a node identifier, the request including a first information element; a generator configured to respond to the received request to produce a unique node identifier that includes a plurality of second information elements and the first information element; and a transmitter configured to send the generated unique node identifier.

10 BRIEF DESCRIPTION OF THE DRAWINGS

In the attached Drawing Figures:

Figure 1 illustrates an exemplary architecture within which the embodiments of this invention
15 may be implemented.

Figure 2 depicts the overall behavior of a system that includes the exemplary embodiments of this invention.

Figure 3 shows an example implementation of the system.

Figure 4 shows an exemplary message sequence flow between the units shown in Figure 1 and
20 Figure 3.

Figure 5 is an example of a Sedvice SmartSpace architecture that is useful in providing a context and technological environment for the exemplary embodiments depicted in Figures 1-4.

Figure 6 is a logic flow diagram that illustrates the operation of a method, and a result of execution of computer program instructions, in accordance with the exemplary embodiments of
25 this invention.

DETAILED DESCRIPTION

Reference may be had to a publication entitled: "Personal Semantic Web Through A Space Based
30 Computing Environment", by Ian Oliver and Jukka Honkola, in proceedings: Middleware for the Semantic Web, Second IEEE International Conference on Semantic Computing, Santa Clara, CA, USA, August 4-7, 2008, which is incorporated by reference herein in its entirety.

The generation of a unique identifier can be achieved by a number of means, such as by
35 randomly generating unique identifiers, or by using serial numbers of hardware. At present, currently known generation methods do not provide for encryption of such an identifier (which

is typically random in its information content).

The exemplary embodiments of this invention pertain to the generation of unique identifiers for various multiple agents or nodes, such as multiple agents or nodes in a SmartSpace-based system. The exemplary embodiments also pertain to other use cases and scenarios, such as RDF blank node identifier generation for the secure inclusion of additional information.

These exemplary embodiments combine various identifier generation techniques and apparatus with the inclusion of random number and unique identifier generation techniques, as well as internal information about a host application, to produce a unique identifier which may also contain semantic information if desired.

In certain exemplary embodiments it should be noted that the described identifier generation code returns an UUID4 number via a unique ID generator component (item 28 in Figure 1). This particular approach is made for simplicity, and is not to be construed as a limitation. For example, also described is an example of the generation of a much richer and robust identifier.

General reference to UUID may be made to, for example, RFC 4122, "A Universally Unique Identifier (UUID) URN Namespace", P. Leach et al., July 2005.

Also described is a scenario in which a node identification may be used within the context of an RDF graph for uniquely identifying "blank nodes" with a meaningful unique identifier, which may carry meta-information in a secure form.

General reference with regard to RDF may be made to, for example, a document entitled "RDF Primer", W3C Recommendation 10 February 2004, eds. F. Manola et al. (www.w3.org).

In the context of these exemplary embodiments a "knowledge processor" contains one or more nodes to facilitate connection to a Space, e.g., a SmartSpace. A SmartSpace is constructed from a number of SIBs which represent and perform the processing of the SmartSpace.

A node joins a SmartSpace by passing a set of credentials which are then evaluated by a SIB. If successful the node is enabled to join the SmartSpace, and the ID of the node is noted for subsequent transactions between the node and any SIB in that SmartSpace. Such transactions between the node and the SmartSpace include information insertion, retraction, querying and subscriptions, as described more fully below with respect to Figure 5.

There is no restriction as to whether each node must present a unique identifier, although a SmartSpace may decide to degrade the trust and functionality if two or more nodes attempt a join with the same identifier.

5

In order to provide further context for the exemplary embodiments of this invention, reference may be made to Figure 5.

The inventors have developed a system, which may be referred to as "Sedvice", that takes
10 conventional agent, blackboard and publish/subscribe concepts and reimplements them in a lightweight manner suitable for small, mobile devices. These agents, termed "knowledge processors" (KPs), operate autonomously and anonymously by sharing information through spaces. These spaces contain a store of information to which the KPs have access and can change at will, processing capabilities for reasoning, modifying and analyzing that information,
15 as well as security and policy functions.

The Sedvice system assumes that information is shared through localized (potentially) personal spaces; that the information can be highly dynamic (changing), and that the information has a semantics primarily given through the interpretation of the information by any given KP. No
20 control flow mechanism need be present, and KPs may share information concerning how external communication, synchronization and coordination can be achieved.

The Sedvice architecture at its simplest is a publish/subscribe system having distributed KPs. Each KP includes a user interface (UI), as well as the logic and/or internal workings of the
25 knowledge processor and nodes. Spaces contain information brokers (known as SIBs), an information store and reasoners. Figure 5 shows the relationship between these various elements.

A KP is considered as a composite structure which is capable of running on a single device, e.g.,
30 a mobile device (such as a wireless communication device), a personal computer or a sensor, as a few examples. A particular device may host more than one KP depending on, for example, the operating system and the available memory. The UI is not required in all cases and, if present, may be very simple in nature, e.g., an LCD display, a single button. A node is the part of the KP which contains the logic and functionality to connect to some Space, including the logic for
35 parsing messages and pointers to subscription handlers between the KP and the Space. A node may potentially connect to more than one Space at a time, thus distributing and synchronizing

No attempt is made by the Space to enforce consistency or integrity of information according to the stated ontologies (through the RDF typeOf relation). Internal reasoning nodes may be present which perform this activity if the Space has been instantiated in this manner. Information is explicitly semi-structured and may take on any form that a KP inserts/retracts. While certain typing constructs and namespaces may be present (if this information is inserted), these do not mean that a node querying for that information will necessarily interpret that information according to the implied ontology. The semantics of the information is interpreted by the reader, merely implied by the writer, and is grounded in the real-world context of the node (any two given KPs may disagree about the ultimate interpretation of the information).

The Space provides further functionality regarding the joining and leaving of KPs/Nodes, as well as policy management. KPs have a set of credentials which are passed during the join operation.

The dual of the KP/node instantiated leave and join operations are Space instantiated invite and remove operations. These operations are not necessarily provided by every Space, nor understood by every KP/node.

Connectivity is provided through a set of "listeners" that provide access via any given specified protocol. One suitable protocol is TCP/IP through standard socket mechanisms, although other protocols can be used to provide, for example, a BluetoothTM enabled listener, or a listener that uses HTTP/S. Listeners can provide preprocessing of the incoming messages if necessary, for example, with BluetoothTM profiles. Any number of listeners may be provided at any time, although it is assumed that there is always at least one present.

A Space is constructed from a number (at least one) of the SIBs which contain the aforementioned schedulers, management information, listeners and information store. A Space is represented by these SIBs, and the total possible connectivity methods are defined by the distributed union of all the represented SIB listeners. The SIBs communicate internally to ensure and verify the membership and credentials of the KP/nodes that have joined that Space, which may connect via any listener and any SIB in that Space.

From the point of view of any KP, the information available is always the distributed union over the transitive closure of the routes between all the SIBs (each SIB contains routing tables to other SIBs), and within a given Space all of the SIBs are totally routable (but not necessarily totally connected).

Having thus provided an overview of one non-limiting type of environment in which the exemplary embodiments of this invention may be practiced, reference is now made to the basic architecture of the system that is shown in Figure 1. A node 5 (which may be embodied in a mobile device, such as a wireless communication device) includes a node identifier generator component 10 that may include, or be interfaced with, a node manager or knowledge processor 12; some specific hardware 14, such as a SIM card; a time source 16, such as a clock or GPS system; and a positioning unit 18, such as a GPS or a WLAN/GSM/3G triangulation module. The node identifier generator component 10 may be configured to include all or a sub-set of a node ID constructor 20, an encryption engine 22, a secondary identifier generator 24, a hardware ID generator 26 (configured for coupling with the hardware 14), a unique ID generator 28, a temporal ID generator 30 (configured for coupling with the time source 16) and a positional ID generator 32 (configured for coupling with the positioning unit 18). Note that not all of these particular modules/units/functions may be present in a particular implementation, and furthermore that the node identifier generator component 10 may include other, additional modules/units/functions in some implementations. In other words, the architecture depicted in Figure 1 is not intended to be viewed as limiting in any respect the exemplary embodiments of this invention.

Also shown in Figure 1 are exemplary apparatus and components that may host the functionality of the node 5, including at least one controller 1, such as a computer or data processor, a computer-readable memory media, or memory 2, that stores a program of computer executable instructions (PROG) 2A, an interface (I/F) 3 for coupling with any needed hardware devices, such as the hardware 14, clock 16 and positioning unit 18 (if not internal to the node 5), and at least one transmitter/receiver (transceiver) 4 for bidirectional communication with other components of the Space, such as a SIB. In some embodiments the transceiver 4 may be a wired-type of transceiver adapted for communication through a cable or fiber optic network, while in other embodiments the transceiver 4 may be wireless-type of transceiver, such as one adapted for radio frequency (e.g., cellular, or WLAN, or mesh network, or WiMAX, or Bluetooth) or optical communications. Note that the various components of the node 5, such as the encryption engine 22, the constructor 20 and the various generators 24, 26, 28, 30 and 32 may be implemented entirely as software that forms a part of the program 2A, or as hardware circuits (e.g., dedicated encryption engine logic), or as a combination of software and hardware (and firmware).

During the construction of a message to be sent to a Space (e.g., a SmartSpace) a node identifier

(ID) is needed. This is achieved by at least two mechanisms: creation of the ID and getting the ID. Both mechanisms return an ID in some suitable form, such as a string of bytes/characters, or in a more structured form such as an XML representation.

- 5 Both mechanisms take a parameter (or parameters if not expressed as a single parameter with structure) that is used during the final processing of the node identifier.

Technically, a create() is a wrapper for a get() that stores a completely new ID based upon triggering the generation mechanism. Both take the form:

10

Create(parameter):NodeID

Get(parameter):NodeID

If one were to embed this in some program:

15

Str var = Create(1)

Print AId is = A, var

the output may appear as follows:

20

Id is = 0x38abc876d8e787ff,

which assumes the non-limiting case of a hexadecimal representation.

- 25 For a case of identifier embedding in a message, a typical message between the node 5 and a SIB, in the non-limiting case of the Sedvice SmartSpace described above with respect to Figure 5, takes the form in XML:

<SSAP_message>

30

<icv>ICV</icv>

<node_id>ID</node_id>

<space_id>ID</space_id>

<transaction_id>ID</transaction_id>

<transaction_type>INSERT</transaction_type>

35

<message_type>REQUEST</message_type>

<parameter name = "triples" type = "RDF-XML | RDF-M3">

STRING

</parameter>

<parameter name = "confirm">TRUE | FALSE</parameter>

</SSAP_message>

5

A typical instantiation of this can be seen below in the exemplary SSAP message with the node ID highlighted:

<SSAP_message>

10 <transaction_type>INSERT</transaction_type>

<message_type>REQUEST</message_type>

<transaction_id>62355249-9807-4c91-95ff-21640cebc266</transaction_id>

<node_id>**1ddcb584-96e4-49e1-a901-7e2cf1d1e642_bootHSS**</node_id>

<space_id>x</space_id>

15 <parameter name = "triples" type = "rdf-m3">

<triple_list>

<triple tag = "0">

20 All communication between the node 5 and the SIB/SmartSpace may be performed in this manner, using the explicitly embedded node_id.

Discussed now in further detail are the constituent components of the node ID generator 10 shown in Figure 1.

25 The primary interface is via the node ID constructor 20 which provides the create and get functions to the outside world, in this case the message constructor 40 (see Figure 2) of the knowledge processor 12. The node ID constructor 20 provides coordination between these functional components and has the responsibility for storing any temporary and transient information related to the process, such as any passed parameters. The node ID constructor 20, if implemented in hardware, may include at least logic and persistent memory. Non-persistent memory may be used as well, so long as it can be reloaded from storage upon power-up or reboot. The storage may be located within-component or off-component using suitable storage technology, such as storing the information to a file and reloading from the file.

35 The encryption engine 22 encrypts the generated identifier before usage, and provides a mechanism for protecting the information stored within the identifier, if the identifier contains

such information that could be used to ascertain certain aspects of the device and/or the hardware or software used for creating the identifier (e.g., the node 5). If the identifier is generated from a single unique source, for example UUID4, then the encryption engine 22 may not be needed. However, if the identifier is generated from a multiplicity of sources, such as
5 other UUID standards or clocks, network addresses, and so forth, then the use of the encryption engine 22 is desirable, but not mandatory. In other words, the encryption engine 22 may be viewed as an optional component in a given implementation.

The encryption engine 22, if present, may use any suitable technology, such as a one-time pad or
10 a public-private key. It may be assumed that the encrypted key can always be decrypted by a trusted recipient, and any information that the identifier contains can thus be read by the trusted recipient.

The secondary identifier generator 24 may be used to add any hard coded information into the
15 key. Such information may be, by example, a serial number of the node identifier generator component 10. While node identifier generation may also be influenced by any passed parameters to the node ID constructor 20, the secondary identifier generator 24 may override this information, or it may add to this information. The secondary identifier generator 24 may be operated so that the information stored by this component is obtained from some other source.

20 The node manager or knowledge processor 12 may be located outside of the node ID generator 10, although information pertaining to the generation of the identifier can be obtained dynamically at run-time from hardware, software or other components, in particular the node manager 12 (which contains information about its run-time environment and which nodes/agents
25 are running), or a particular node/knowledge processor itself.

The hardware ID generator 26 generates identifier information based upon reading some hardware component, such as the SIM card 14, or a CPU serial number, as two non-limiting
30 examples.

The unique ID generator 28 is connected to, or contains, a random number generator of suitable quality to generate a "guaranteed" unique identifier. An example of such a scheme is the UUID4 identifier. Other identifier schemes are also valid. This information can be gained also by exploiting certain characteristics of hardware, such as specific fault or test instructions, for
35 example, floating-point division bugs in Pentium™ class processors, AMD x86 processor serial numbers, CPU microcode configurations and so forth.

The temporal ID generator 30 generates identifiers based upon the current time (e.g., some aspect of the current time such as the year, full date, hour/minute/second, Unix-time, etc.) This information is typically obtained from the external time source 16, such as an on-board hardware clock, a software call to a function such as POSIX time(), GPS (or other positioning system) clock signals, or national broadcast radio clock signals, as non-limiting examples.

The positional ID generator 32 generates identifiers based upon the current position of the device. The positional ID generator 32 takes its information from the positioning unit 18, such as the above-noted GPS or WLAN triangulation, GSM/3G/mobile device base station position triangulation, 3GPP, attitude sensors and/or accelerometers, as non-limiting examples.

As was noted, the foregoing components are not to be construed as constituting a full set of identifier generation components or modules. In general, any unit which can supply information suitable for these devices, as well as others not described, can be used for identifier generation. Non-limiting examples of such units include, but are not limited to, sensors such as temperature sensors, humidity sensors, light meters and camera functionality.

Example Behavior (1)

In this example there is described typical behavior and messaging that can be seen in a system of interest (one including the node identifier generator component 10). Reference is made to Figure 2 where the message constructor component 40 of some node requests an ID as part of a SmartSpace insert operation. The message constructor 40 receives a request to insert into a given space 42 the RDF triple (a,b,c). The message constructor 40 requires a unique identifier to be sent along with the above information to the space 42, and makes a Get() request with some static information ("12345") to object1 44. The node ID generator 10 associated with object1 44 returns to the message constructor 40 an identifier based upon the current state of the system and containing (in some processed form) the static information. The message constructor 40 then sends a completed message onward to the space 42 (SIB) for subsequent processing.

Example Behavior (2)

In this additional non-limiting example the internal operation of the various components of the node ID generator 10 shown in Figure 1 are further explained by the communication between them. The instantiation of this particular unit is shown in Figure 3, and the exemplary message

flow is shown in Figure 4. Note in Figure 4 that the instantiation of the hardware component 14 is shown twice, once as the CPU 14A, and once as the SIM card 14B.

Initially node 5 makes the call Get("12345") in order to receive a node identifier. The following messages are numbered as Message *x*, where *x* is a number 1-16 in the message flow diagram shown in Figure 4. An explanation of each of the Messages 1-16 is as follows:

1. Call to the hardware ID generator 26 to return an identifier
 2. Call to the CPU 14A to return (in this example) its serial number
 - 10 3. Return of the CPU 14A serial number in an internal/device specific format
 4. Call to the SIM card 14B in the device for its serial number
 5. Return of the SIM Card 14B serial number in an internal/device specific format
 6. Return of the hardware identifier in some node generator format (see below)
 7. Call to the temporal ID generator 30 for the current time
 - 15 8. Call to the hardware clock 16 in the system
 9. Return of the time in some internal format, e.g.: 32-bits containing the BIOS date
 10. Return of the time in some node generator format (see below)
 11. Call to the secondary ID generator with parameters ("12345")
 12. Return of the secondary ID (see below)
 - 20 13. Processing of the returned information by the node ID constructor 20
 14. Passing of the processed information to the encryption engine 22
 15. Return of the encrypted information to the node ID constructor 20
 16. Return of the final unique (meaningful) identifier to the node 5
- 25 The identifier returned to the node ID constructor 20 from the hardware ID generator 26 may take the following form, if represented using XML:

```

<hardware id>
  <cpu serial number = "abcdef">
30  <simcard number = "eeeeee">
  </hardware id>

```

The identifier returned to the node ID constructor 20 from the temporal ID generator 30 may take the following form, if represented using XML:

```

35  <temporal id>

```

```

<time = 20:31:28>
<date = 18/6/2007>
<timezone = GMT+2>
</temporal id>

```

5

Note that in both examples above the hardware ID generator 26 and temporal ID generator 30 process the incoming information from their respective sources 14, 16 in order to produce a format that is usable and understandable by the node ID constructor 20.

10 The actual functionality of the secondary ID generator 24 is relatively unconstrained. For example, in one non-limiting embodiment it takes the parameters supplied using the Get() call to the node ID constructor 20 of the node ID generator 10 and processes the parameters with (potentially) some statically coded information. In this example the return may take the form:

15 <secondary_id = 12345_aaaaa>,

where "aaaaa" represents the potentially statically coded information.

The node ID constructor 20 performs some function on all of the returned information. In one non-limiting embodiment the function may be a simple concatenation of the information, while another, more complex function may be to further refine and process the information to produce the identifier. For example, the node ID constructor 20 may perform concatenation and MD5 production. MD5 (Message-Digest algorithm 5) is a widely used, partially insecure cryptographic hash function with a 128 bit hash value.

25

In this non-limiting example the node ID constructor 20 concatenates the information received from the various components of the node ID generator 10 and computes a suitable hashing function (e.g., a MD5 hashing function) over the concatenation:

30 C: = "<hardware id><cpu serial number = "abcdef"><simcard number = "eeeeee"></hardware id><temporal id><time = 20:31:28><date = 18/6/2007><timezone = GMT+2></temporal id><secondary_id = 12345_aaaaa>"
H: = hash_as_md5(C)

35 The value in H may then be passed to the encryption engine 22 (message 14 in Figure 4) which returns an encrypted form of H (message 15) which can be following:

Encrypt(H) = "f2fedbcf87487dafcbcd762374fadfafde010182387213bcfdeb345b62f2deaf"

This encrypted value is returned to the node 5 and used in whatever way necessary, typically as the node ID (highlighted below) in some message to the space 42 (e.g., an SSAP message to a SmartSpace), for example:

```

<SSAP_message>
  <transaction_type>INSERT</transaction_type>
10  <message_type>REQUEST</message_type>
  <transaction_id>62355249-9807-4c91-95ff-21640cebc266</transaction_id>
  <node_id>    f2fedbcf87487dafcbcd762374fadfafde010182387213bcfdeb345b62f2deaf
</node_id>
  <space_id>x</space_id>
15  <parameter name = "triples" type = "rdf-m3">

```

Further with regard to the uses of these exemplary embodiments, the RDF standard defines the notion of blank node. As all nodes in an RDF graph must be uniquely identified this means that a "blank node" has a uniquely generated identifier. Using the exemplary embodiments of this invention it is possible to generate unique identifiers that contain encoded information. Such information can then only be read by certain parties having knowledge of the required decryption key. This is fully within the remit, scope and official definition of the universal resource identifier scheme.

Such an RDF triple is described below using the XML representation; where the generated identifier is highlighted:

```

<subject>http://hss.nrc.nokia.com/hssuser#10c191a5-d0d5-4cf7-9694-83d7ab28fb35</subject>
><predicate>!rdf:type</predicate><object type = "uri">
30  http://hss.nrc.nokia.com/hssuser#FamilyMember</object>

```

Note the addition of the generated identifier to the namespace declaration.

As should be appreciated, this describes a method of "secretly" encoding meta-information or other information into the URI of a node, such as a blank RDF node.

Described now are examples of instantiated messages. In these example messages the generated node_id is highlighted. Also highlighted are other identifiers which may be generated in the same or similar manner, in accordance with the exemplary embodiments of this invention.

5 Message example 1:

```

<SSAP_message>
  <transaction_type>INSERT</transaction_type>
  <message_type>REQUEST</message_type>
10  <transaction_id>62355249-9807-4c91-95ff-21640cebc266</transaction_id>
  <node_id>1ddcb584-96e4-49e1-a901-7e2cf1d1e642_bootHSS</node_id>
  <space_id>x</space_id>
  <parameter name = "triples" type = "rdf-m3">
    <triple_list>
15    <triple tag = "0">
      <subject>http://hss.nrc.nokia.com/hssuser#FamilyMember</subject>
      <predicate>!rdfs:subClassOf</predicate>
      <object type = "uri">http://hss.nrc.nokia.com/hssuser#HomeMember</object>
    </triple>
20    <triple tag = "1">
      <subject>http://hss.nrc.nokia.com/hssuser#FamilyMember</subject>
      <predicate>!rdf:type</predicate>
      <object type = "uri">!rdfs:Class</object>
    </triple>
25    <triple tag = "2">
      <subject>http://hss.nrc.nokia.com/hssuser#HomeMember</subject>
      <predicate>!rdf:type</predicate>
      <object type = "uri">!rdfs:Class</object>
    </triple>
30    <triple tag = "3">
      <subject>http://hss.nrc.nokia.com/hssuser#Visitor</subject>
      <predicate>!rdfs:subClassOf</predicate>
      <object type = "uri">http://hss.nrc.nokia.com/hssuser#HomeMember</object>
    </triple>
35    </triple_list>
  </parameter>

```

```

    <parameter name = "confirm">
      True
    </parameter>
  </SSAP_message>

```

5

Message Example 2:

```

<SSAP_message><transaction_type>INSERT</transaction_type><message_type>
REQUEST</message_type><transaction_id>bc5c694a-2406-427c-8193-7b58a4e7442a</trans
10 action_id><node_id>1ddcb584-96e4-49e1-a901-7e2cf1d1e642_bootHSS</node_id><space_i
d>x</space_id><parameter name = "triples" type = "rdf-m3"><triple_list><triple tag = "0">
<subject>
http://hss.nrc.nokia.com/mediaCDS#MediaTrack</subject><predicate>!rdfs:subClassOf</predic
ate><object
                                type
                                =
                                "uri">
15 http://hss.nrc.nokia.com/mediaCDS#MediaItem</object></triple><triple
tag
=
"1"><subject>http://hss.nrc.nokia.com/mediaCDS#MediaTrack</subject><predicate>!rdf:type<
/predicate><object
type
=
"uri">!rdfs:Class</object></triple><triple
tag
=
"2"><subject>http://hss.nrc.nokia.com/mediaCDS#MediaItem</subject><predicate>!rdf:type<
/predicate><object
type
=
"uri">!rdfs:Class</object></triple><triple
tag
=
20 "3"><subject>http://hss.nrc.nokia.com/mediaCDS#MediaItem</subject><predicate>!rdfs:subCl
assOf</predicate><object
                                type
                                =
                                "uri">http://hss.nrc.nokia.com/mediaCDS#MediaObject</object></triple><triple
tag
=
"4"><subject>http://hss.nrc.nokia.com/mediaCDS#MediaItem</subject><predicate>!rdf:type<
/predicate><object
type
=
"uri">!rdfs:Class</object></triple><triple
tag
=
25 "5"><subject>http://hss.nrc.nokia.com/mediaCDS#MediaObject</subject><predicate>!rdf:type
</predicate><object
                                type
                                =
                                "uri">!rdfs:Class</object></triple></triple_list></parameter><parameter
name
=
"confirm">True</parameter></SSAP_message>

```

30 The forgoing message elements, construction and syntax are deemed to be illustrative and exemplary, and are not intended to be construed as restricting in any manner the scope and use of the exemplary embodiments of this invention.

It can be noted that there may be different latencies in the response times of the various units
 35 shown in Figure 1. For example, the response of the temporal ID generator 30 and/or the hardware ID generator 26 may be microseconds, while the response time of the positional ID

generator 32 may be milliseconds or seconds, depending on the nature of the positioning unit 18. These latency differences may be mitigated by caching (storing) certain results, such as a last known position, and in response to a time-out occurring (e.g., when the requested information cannot be found) substituting suitable other information in its place.

5

One non-limiting advantage that may be realized by the use of these exemplary embodiments is that the identifier structure returned may be highly structured and readable by systems capable of supporting semantic web constructs (as well as those systems that do not). The identifier may also be encrypted, giving a degree of protection to the identifier such that systems that cannot

10 use the identifier information or read the information contained within the identifier can still use the identifier and guarantee its uniqueness.

Based on the foregoing it should be apparent that the exemplary embodiments of this invention provide a method, apparatus and computer program(s) to generate a structured and possibly

15 encrypted identifier of a node, such as a node configured to be coupled with a space containing information brokers and other nodes.

Referring to Figure 6, a method in accordance with these exemplary embodiments thus includes (Block 6A) receiving a request for a node identifier, the request including a first information

20 element; (Block 6B) in response, generating a unique node identifier that includes a plurality of second information elements and the first information element; and (Block 6C) responding to the request with the generated unique node identifier.

The method of the preceding paragraph, where the first information element comprises static

25 information.

The method of the preceding paragraphs, further comprising encrypting the generated unique node identifier.

30 The method of the preceding paragraphs, where the second information elements comprise a temporal information element.

The method of the preceding paragraphs, where the second information elements comprise a serial number associated with the node.

35

The method of the preceding paragraphs, where the second information elements comprise a

random number.

The method of the preceding paragraph, where the random number is a universal unique identifier.

5

The method of the preceding paragraphs, where the second information elements comprise a value obtained from a characteristic of a data processor or a data processor instruction set associated with the node.

10 The method of the preceding paragraphs, where the second information elements comprise a value obtained from a hardware element that forms a part of, or that is coupled with, the node.

The method of the preceding paragraphs, where the second information elements comprise a value obtained from a sensor.

15

The method of the preceding paragraphs, where generating comprises executing a function using the plurality of second information elements and the first information element.

The method of the preceding paragraph, where the function is concatenation.

20

The method of the preceding two paragraphs, further comprising executing a hashing operation on a result of the execution of the function.

25 The method of the preceding paragraph, further comprising encrypting a result of the hashing function.

The method of the preceding paragraphs, where in response to responding further comprising including the generated unique node identifier in a message.

30 The method of the preceding paragraph, further comprising sending the message to a semantic information broker.

The method of the preceding paragraphs, where the message is a smartspace application protocol message.

35

The method of the preceding paragraphs, where the message is a resource description

framework message.

The method of the preceding paragraph, where the resource description framework message is associated with a blank node.

5

The various blocks shown in Figure 6 may be viewed as method steps, and/or as operations that result from operation of computer program code stored in a computer-readable media (e.g., the program 2A stored in the memory 2), and/or as a plurality of coupled logic circuit elements constructed to carry out the associated function(s).

10

These exemplary embodiments also encompass an apparatus that includes means for receiving a request for a node identifier, the request including a first information element; means, responsive to the request, for generating a unique node identifier that includes a plurality of second information elements and the first information element; and means for responding to the request with the generated unique node identifier.

15

In general, the various exemplary embodiments may be implemented in hardware or special purpose circuits, software, logic or any combination thereof. For example, some aspects may be implemented in hardware, while other aspects may be implemented in firmware or software which may be executed by a controller, microprocessor or other computing device, although the invention is not limited thereto. The various embodiments may be implemented at least partially by computer program instructions that are stored or otherwise embodied in computer-readable memory media, such as semiconductor or disk-based memory devices or modules or systems.

20

While various aspects of the exemplary embodiments of this invention may be illustrated and described as block diagrams, logic flow diagrams, message flow diagrams, or by using some other pictorial representation, it is well understood that these blocks, apparatus, systems, techniques or methods described herein may be implemented in, as non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof. As such, it should be appreciated that at least some aspects of the exemplary embodiments of the inventions may be practiced in various components such as integrated circuit chips and modules. For example, the node ID generator 10 may be wholly (or partially) implemented within an integrated circuit package, either alone or in combination with other functional units.

30

35

Various modifications and adaptations to the foregoing exemplary embodiments of this invention

may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings. However, any and all modifications will still fall within the scope of the non-limiting and exemplary embodiments of this invention.

- 5 For example, while the exemplary embodiments have been described above in the context generally of the Semantic Web and RDF, it should be appreciated that the exemplary embodiments of this invention are not limited for use with only these particular types of systems and standards, and that they may be used to advantage in other systems and standards. In addition, the exemplary embodiments of this invention may be used in applications and systems
10 other than the above described SmartSpace and/or the Sedvice applications and systems.

- Further, the various names used for the described parameters are not intended to be limiting in any respect, as these parameters may be identified by any suitable names. Further, any formulas and logical expressions, including message formats and the like, that use these various
15 parameters may differ from those expressly disclosed. Further, the various names assigned to different components and units (e.g., node ID constructor, unique ID generator, etc.) are not intended to be limiting in any respect, as these various components and units may be identified by any suitable names.

- 20 It should be noted that the terms "connected," "coupled," or any variant thereof, mean any connection or coupling, either direct or indirect, between two or more elements, and may encompass the presence of one or more intermediate elements between two elements that are "connected" or "coupled" together. The coupling or connection between the elements can be physical, logical, or a combination thereof. As employed herein two elements may be considered
25 to be "connected" or "coupled" together by the use of one or more wires, cables and/or printed electrical connections, as well as by the use of electromagnetic energy, such as electromagnetic energy having wavelengths in the radio frequency region, the microwave region and the optical (both visible and invisible) region, as several nonlimiting and non-exhaustive examples.

- 30 Furthermore, some of the features of the various non-limiting and exemplary embodiments of this invention may be used to advantage without the corresponding use of other features. As such, the foregoing description should be considered as merely illustrative of the principles, teachings and exemplary embodiments of this invention, and not in limitation thereof.

CLAIMS

What is claimed is:

1. A method, comprising:

receiving a request for a node identifier, the request including a first information element;
in response, generating a unique node identifier that includes a plurality of second information elements and the first information element; and
responding to the request with the generated unique node identifier.

2. The method of claim 1, where the first information element comprises static information.

3. The method of claim 1, further comprising encrypting the generated unique node identifier.

4. The method of claim 1, where the second information elements comprise a temporal information element.

5. The method of claim 1, where the second information elements comprise a serial number associated with the node.

6. The method of claim 1, where the second information elements comprise a random number.

7. The method of claim 6, where the random number is a universal unique identifier.

8. The method of claim 1, where the second information elements comprise a value obtained from a characteristic of a data processor or a data processor instruction set associated with the node.

9. The method of claim 1, where the second information elements comprise a value obtained from a hardware element that forms a part of, or that is coupled with, the node.

10. The method of claim 1, where the second information elements comprise a value obtained from a sensor.

11. The method of claim 1, where generating comprises executing a function using the plurality

of second information elements and the first information element.

12. The method of claim 11, where the function is concatenation.

13. The method of claim 11, further comprising executing a hashing operation on a result of the execution of the function.

14. The method of claim 13, further comprising encrypting a result of the hashing operation.

15. The method of claim 1, where in response to responding further comprising including the generated unique node identifier in a message.

16. The method of claim 15, further comprising sending the message to a semantic information broker.

17. The method of claim 15, where the message is a smartspace application protocol message.

18. The method of claim 15, where the message is a resource description framework message.

19. The method of claim 18, where the resource description framework message is associated with a blank node.

20. A computer-readable memory media that stores a computer program instructions, where execution of the computer program instructions results in the performance of operations that comprise:

receiving a request for a node identifier, the request including a first information element;

in response, generating a unique node identifier that includes a plurality of second information elements and the first information element; and

responding to the request with the generated unique node identifier.

21. The computer-readable memory media of claim 20, where the first information element comprises static information.

22. The computer-readable memory media of claim 20, further comprising an operation of encrypting the generated unique node identifier.

23. The computer-readable memory media of claim 20, where the second information elements

comprise a temporal information element.

24. The computer-readable memory media of claim 20, where the second information elements comprise a serial number associated with the node.

25. The computer-readable memory media of claim 20, where the second information elements comprise a random number.

26. The computer-readable memory media of claim 25, where the random number is a universal unique identifier.

27. The computer-readable memory media of claim 20, where the second information elements comprise a value obtained from a characteristic of a data processor or a data processor instruction set associated with the node.

28. The computer-readable memory media of claim 20, where the second information elements comprise a value obtained from a hardware element that forms a part of, or that is coupled with, the node.

29. The computer-readable memory media of claim 20, where the second information elements comprise a value obtained from a sensor.

30. The computer-readable memory media of claim 20, where the operation of generating comprises executing a function using the plurality of second information elements and the first information element.

31. The computer-readable memory media of claim 30, where the function is concatenation.

32. The computer-readable memory media of claim 30, further comprising executing a hashing operation on a result of the execution of the function.

33. The computer-readable memory media of claim 32, further comprising encrypting a result of the hashing operation.

34. The computer-readable memory media of claim 20, where in response to responding further comprising including the generated unique node identifier in a message.

35. The computer-readable memory media of claim 34, further comprising sending the message to a semantic information broker.
36. The computer-readable memory media of claim 34, where the message is a smartspace application protocol message.
37. The computer-readable memory media of claim 34, where the message is a resource description framework message.
38. The computer-readable memory media of claim 37, where the resource description framework message is associated with a blank node.
39. An apparatus, comprising:
a receiver configured to receive a request for a node identifier, the request including a first information element;
a generator configured to respond to the received request to produce a unique node identifier that includes a plurality of second information elements and the first information element; and
a transmitter configured to send the generated unique node identifier.
40. The apparatus of claim 39, where the first information element comprises static information.
41. The apparatus of claim 39, further comprising an encryption engine configured to encrypt the generated unique node identifier prior to the identifier being sent.
42. The apparatus of claim 39, where the second information elements comprise at least one of a temporal information element, a serial number associated with the node, a random number, a value obtained from a characteristic of a data processor or a data processor instruction set, a value obtained from a hardware element that forms a part of, or that is coupled with, the node, and a value obtained from a sensor.
43. The apparatus of claim 42, where the random number is a universal unique identifier.
44. The apparatus of claim 39, where said generator is configured to execute a function using the plurality of second information elements and the first information element.
45. The apparatus of claim 44, where the function is concatenation.

46. The apparatus of claim 44, said generator being further configured to execute a hashing operation on a result of the execution of the function.
47. The apparatus of claim 46, further comprising an encryption engine configured to encrypt a result of the hashing operation prior to the identifier being sent.
48. The apparatus of claim 39, where the generated unique node identifier is included as part of a message that is sent by said transmitter.
49. The apparatus of claim 48, where said transmitter sends the message to a semantic information broker.
50. The apparatus of claim 48, where the message is a smartspace application protocol message.
51. The apparatus of claim 48, where the message is a resource description framework message.
52. The apparatus of claim 51, where the resource description framework message is associated with a blank node.

1/4

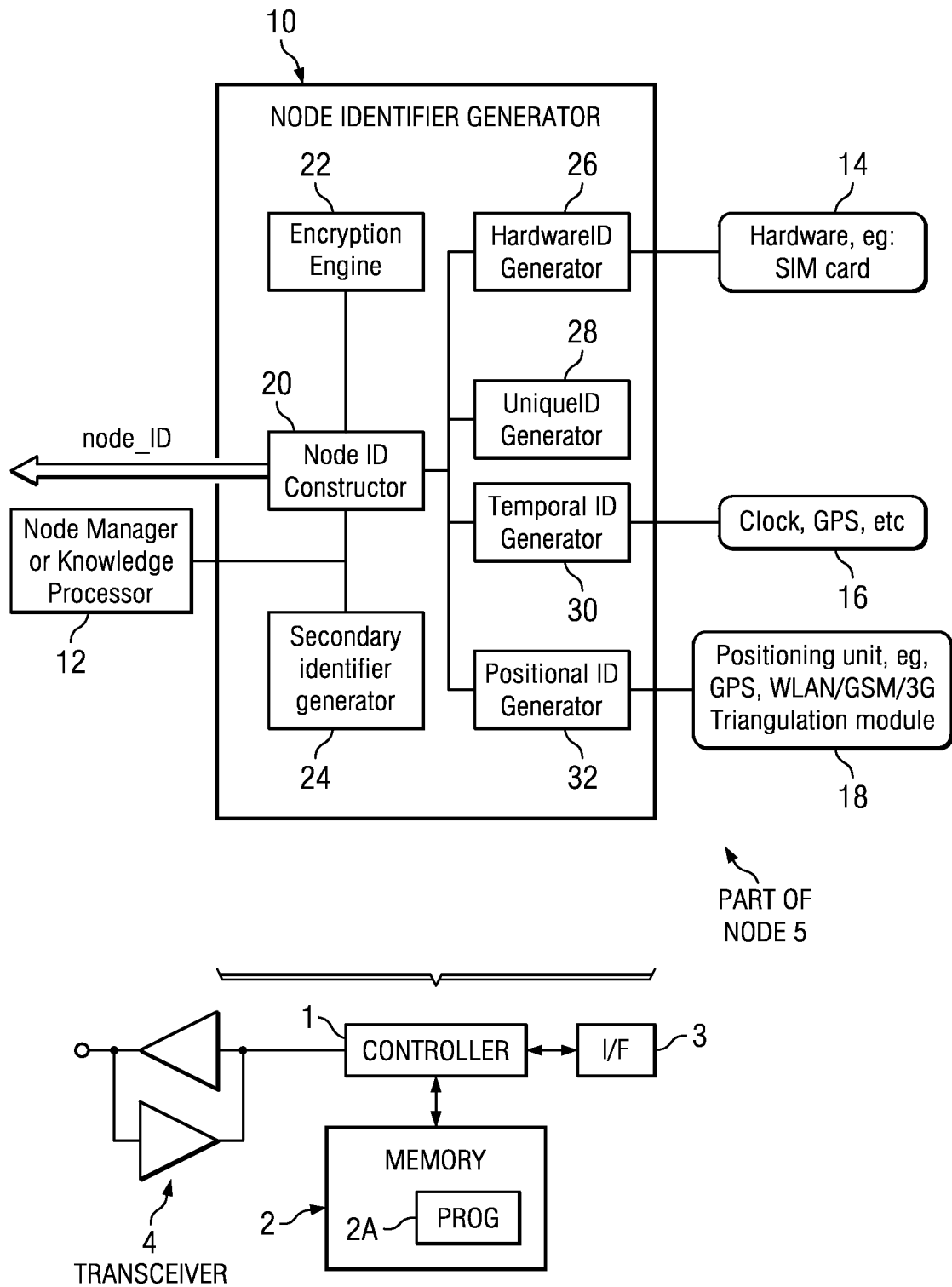


FIG. 1

2/4

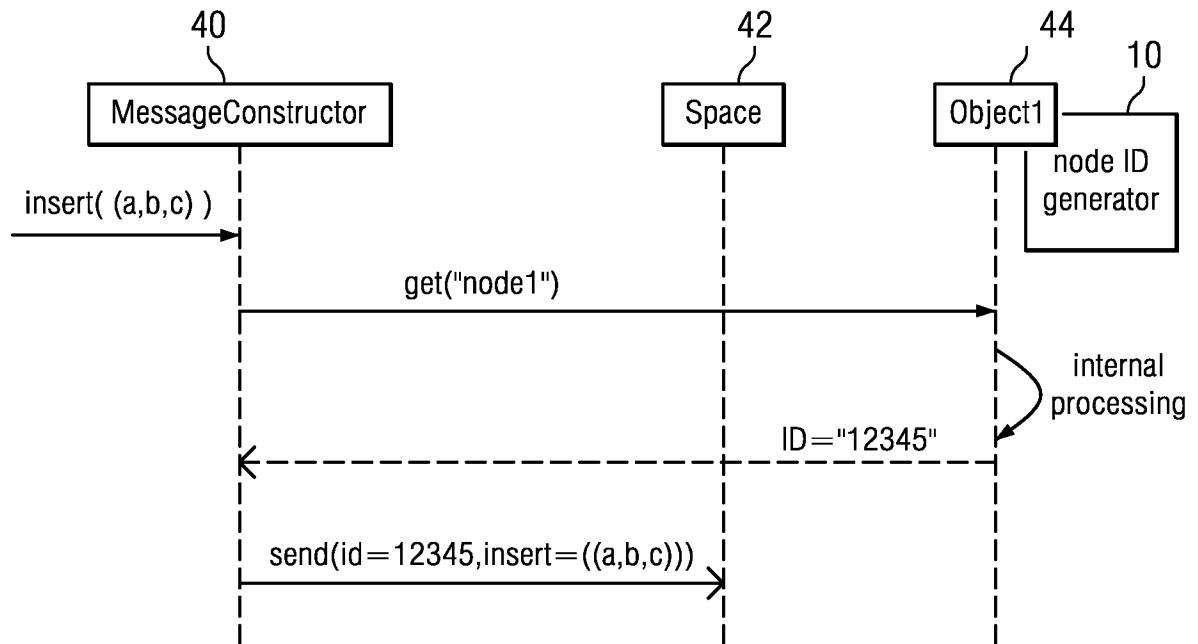


FIG. 2

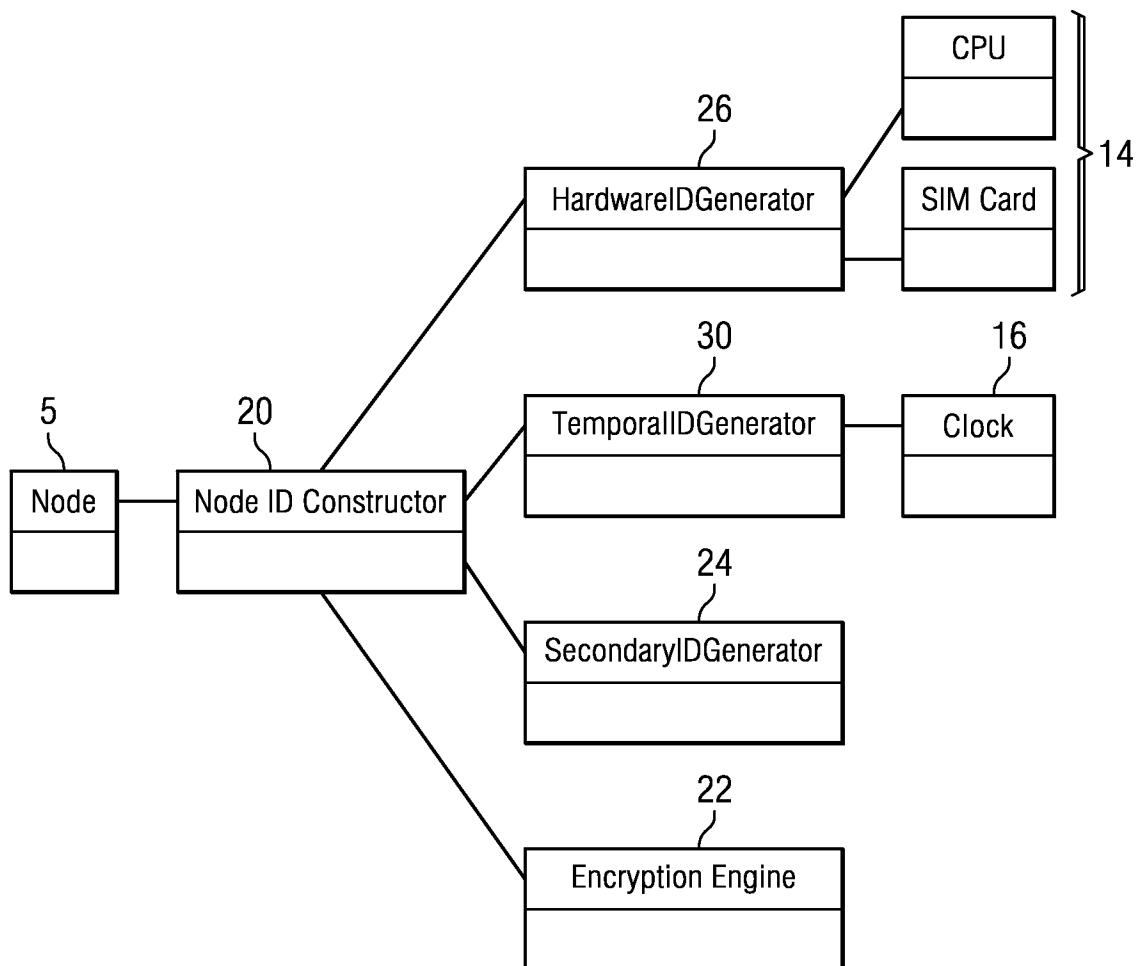


FIG. 3

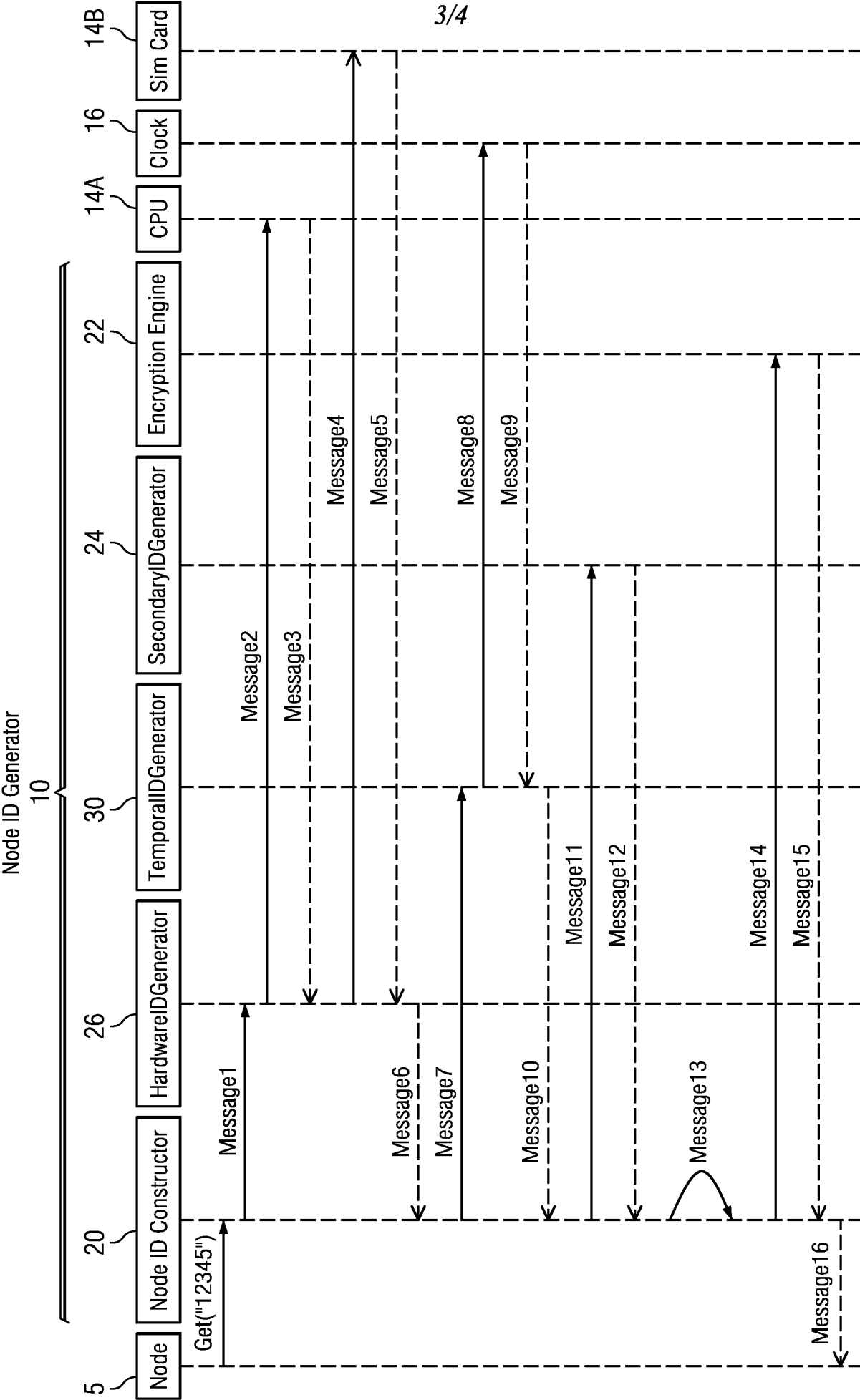


FIG. 4

4/4

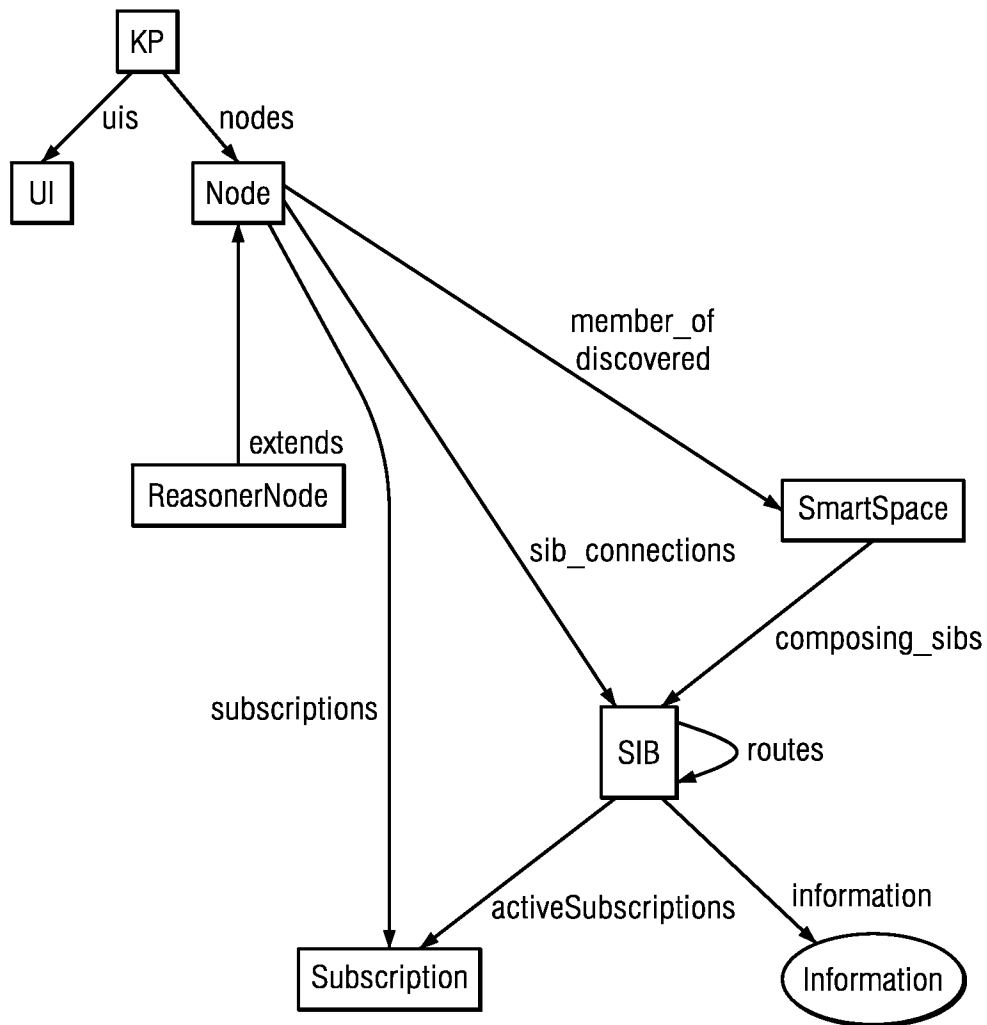


FIG. 5

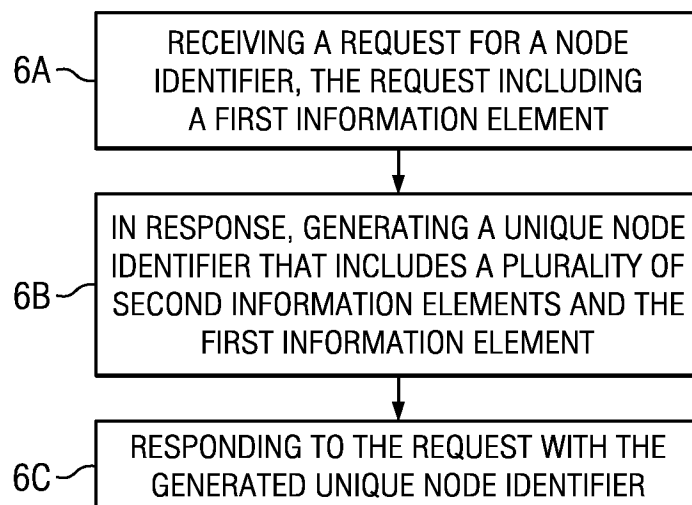


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2009/050640

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04L, G06F

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

FI, SE, NO, DK

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

EPO-Internal, WPI, XPESP, XPIEE, XPIETF, XPI3E, XPMISC, XPRD, XP3GPP, COMPDX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006106802 A1 (GIBLIN CHRISTOPHER J et al.) 18 May 2006 (18.05.2006), figure 1; [0023], [0025]-[0027], [0044]-[0049]	1-15, 20-34, 39-48
X	US 2004199762 A1 (CARLSON MICHAEL PIERRE et al.) 07 October 2004 (07.10.2004), abstract; figure 2; [0001], [0022], [0039], [0048]	1-3, 11, 15, 20-22, 30, 39-41, 44
X	US 2005033768 A1 (SAYERS CRAIG P et al.) 10 February 2005 (10.02.2005), abstract; figures 3-5; [0004]-[0006], [0039]-[0054]	1, 5-10, 16-20, 24-29, 35-39, 42, 43, 49-52
X	SHANKAR V: "URI Link Encryption" [Online] 14 November 2007 (2007-11-14), Retrieved from the Internet: <URL:http://www.utdallas.edu/~vns082000/publishes_files/URI_Link_Encryption.pdf> [retrieved on 2009-11-05], abstract; figure 1; sections 3.2 (Accessing unauthorized information), 4 (URL Encryption as a solution)	1-4, 20-23, 39-42

☒ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

05 November 2009 (05.11.2009)

Date of mailing of the international search report

11 November 2009 (11.11.2009)

Name and mailing address of the ISA/FI
National Board of Patents and Registration of Finland
P.O. Box 1160, FI-00101 HELSINKI, Finland

Facsimile No. +358 9 6939 5328

Authorized officer

Vesa-Matti Louekoski

Telephone No. +358 9 6939 500

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2009/050640

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SAYERS C et al.: "The case for generating URIs by hashing RDF content" [Online] 26 December 2002 (2002-12-26), Retrieved from the Internet: <URL:http://web.archive.org/web/20040104224011/www.hpl.hp.com/techreports/2002/HPL-2002-216.pdf> [retrieved on 2009-11-05], abstract; sections 3, 4	
A	FEATHER C: "The Hashed URI", IETF INTERNET-DRAFT, 17 September 2002 (2002-09-17), XP015000872, whole document	
A	WANG X et al.: "Semantic Space: An Infrastructure for Smart Spaces", In: IEEE Pervasive Computing, IEEE Service Center, Los Alamitos, CA, US, Vol. 3, Iss. 3, 1 July 2004 (2004-07-01), pp. 32 - 39, XP011116167, figure 2; sections "The context model", "The Semantic Space infrastructure"; 2nd last paragraph of subsection "Performance" on pages 38-39	
A	SUDHA R et al.: "Ubiquitous Semantic Space: A context-aware and coordination middleware for Ubiquitous Computing" In proceedings: 2nd International Conference on Communication Systems Software and Middleware, 1 January 2007 (2007-01-01), pp. 1-7, XP031113861, abstract; figure 1; sections III, IV	
A	OLIVER I et al.: "Personal Semantic Web Through A Space Based Computing Environment". In proceedings: Middleware for the Semantic Web, 2nd IEEE International Conference on Semantic Computing, Santa Clara, CA, USA, 4-7 August 2008. Retrieved from the Internet: <URL:http://arxiv.org/abs/0808.1455> [Retrieved on 2009-11-09], whole document	

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2009/050640

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2006106802 A1	18/05/2006	US 2008313469 A1 KR 20060055314 A CN 1777090 A	18/12/2008 23/05/2006 24/05/2006
US 2004199762 A1	07/10/2004	None	
US 2005033768 A1	10/02/2005	None	

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2009/050640

CLASSIFICATION OF SUBJECT MATTER

Int.Cl.

H04L 29/12 (2006.01)

G06F 17/30 (2006.01)

G06F 21/00 (2006.01)

H04L 29/06 (2006.01)