US 20060294042A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2006/0294042 A1**

Kapadia et al. (43) **Pub. Date: Dec. 28, 2006**

(54) **DISPARATE DATA STORE SERVICES CATALOGUED FOR UNIFIED ACCESS**

(75) Inventors: **Arshish C. Kapadia**, Issaquah, WA (US); **Jonah S. Burke**, Seattle, WA (US); **Howard M. Crow**, Sammamish, WA (US)

Correspondence Address:
MICROSOFT CORPORATION
ATTN: PATENT GROUP DOCKETING DEPARTMENT
ONE MICROSOFT WAY
REDMOND, WA 98052-6399 (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA

(21) Appl. No.: **11/165,748**

(22) Filed: **Jun. 23, 2005**

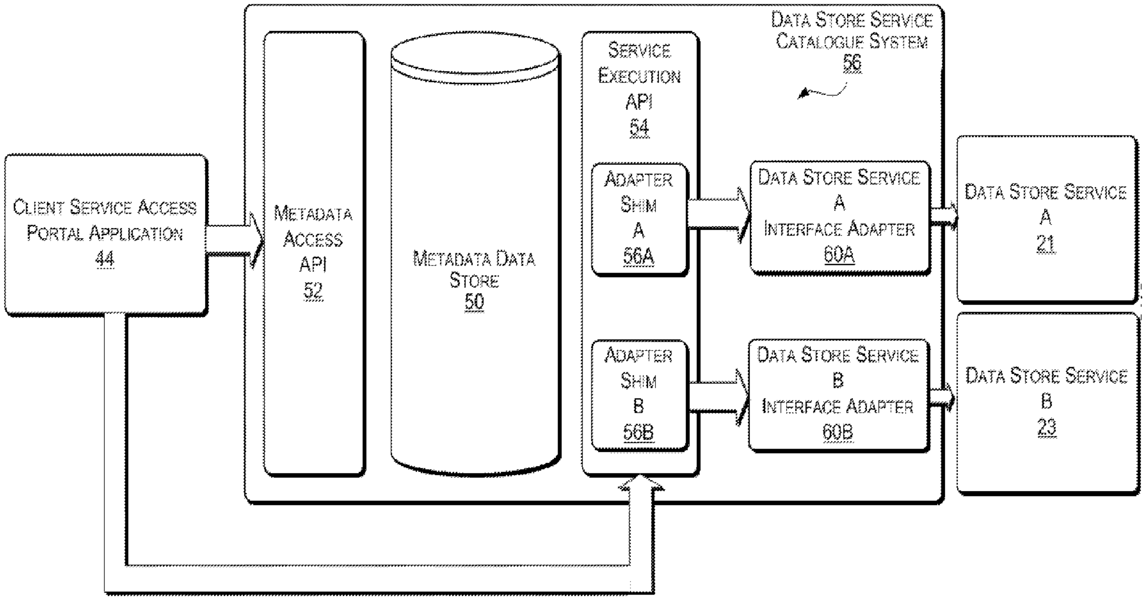
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/1**

(57) **ABSTRACT**

A data store catalogue service model is disclosed for exposing disparate data store services to clients without requiring explicit knowledge on how to interact with each disparate service. The data store catalogue service model may maintain interaction details for each data store service that clients may access for interacting with those services. The model may also maintain metadata that may describe the different types of services available for access by clients, the data types used by those services, how to access the data provided by the services, and how to communicate semantically with the services for accessing the service's data. Additionally, the data store may include metadata that enables clients to interact with one or more registered data stores in a number of ways, such as for discovering registered data store services, entities, classes, and/or any associations between related entities within the same or among disparate services.



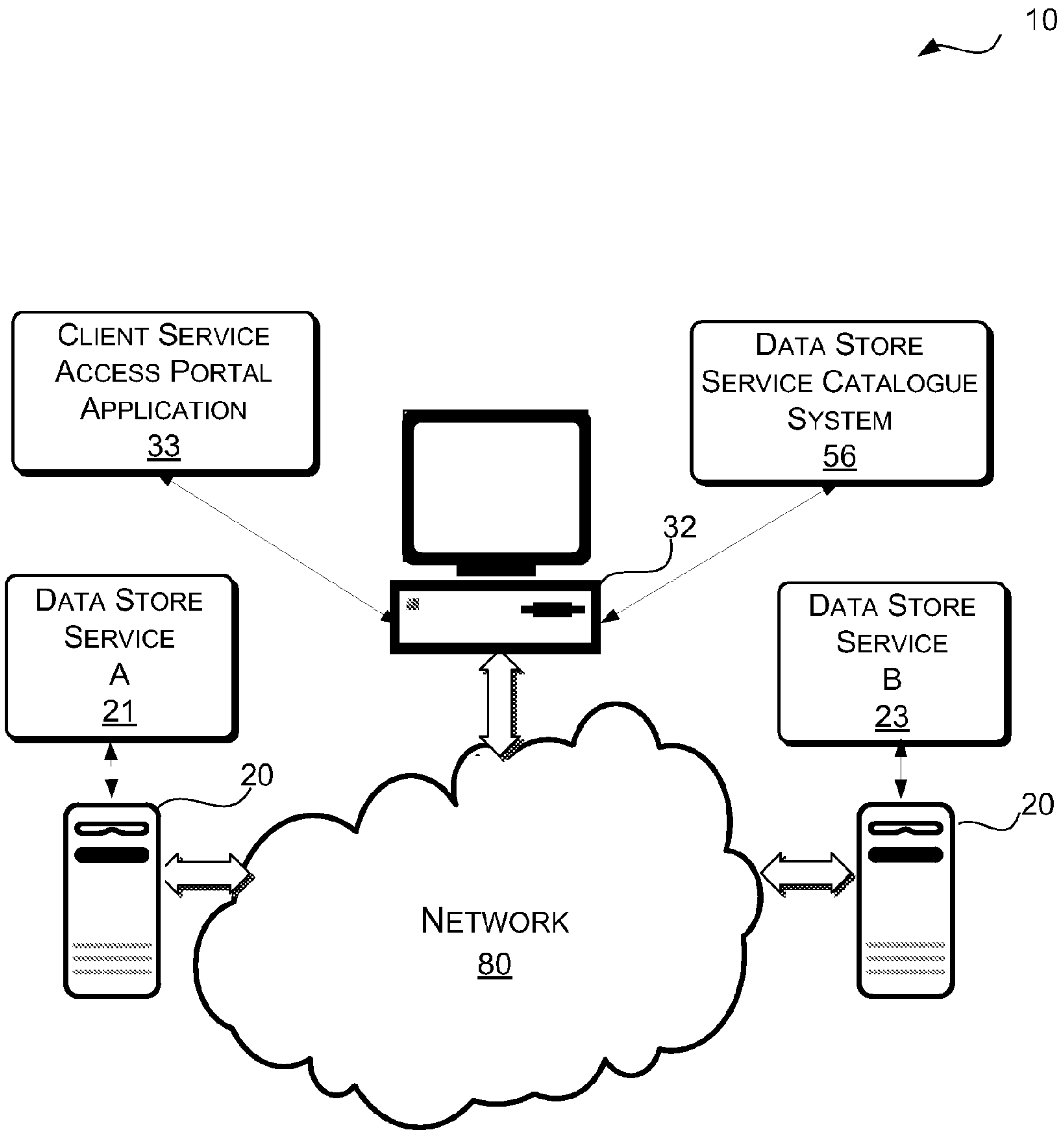

10
Client Service Access Portal Application
33
Data Store Service Catalogue System
56
32
Data Store Service A
21
Data Store Service B
23
20
20
Network
80

FIG. 1

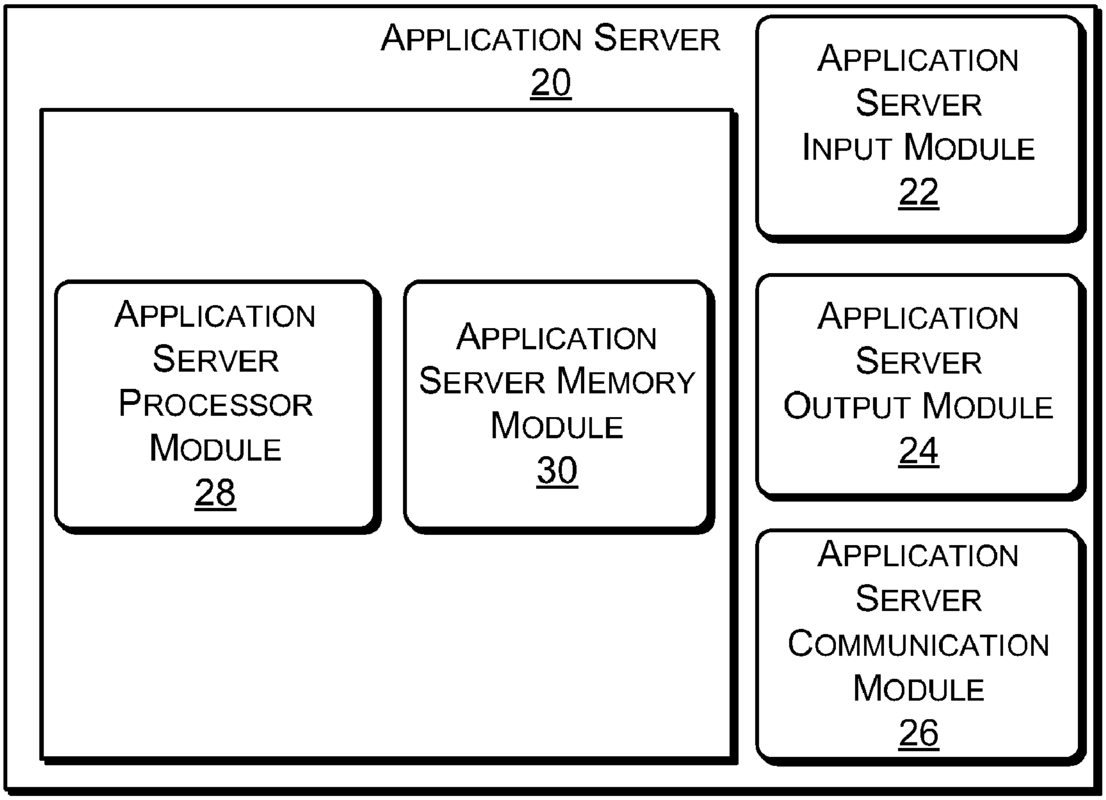

APPLICATION SERVER
20
APPLICATION SERVER INPUT MODULE
22
APPLICATION SERVER PROCESSOR MODULE
28
APPLICATION SERVER MEMORY MODULE
30
APPLICATION SERVER OUTPUT MODULE
24
APPLICATION SERVER COMMUNICATION MODULE
26

FIG. 2

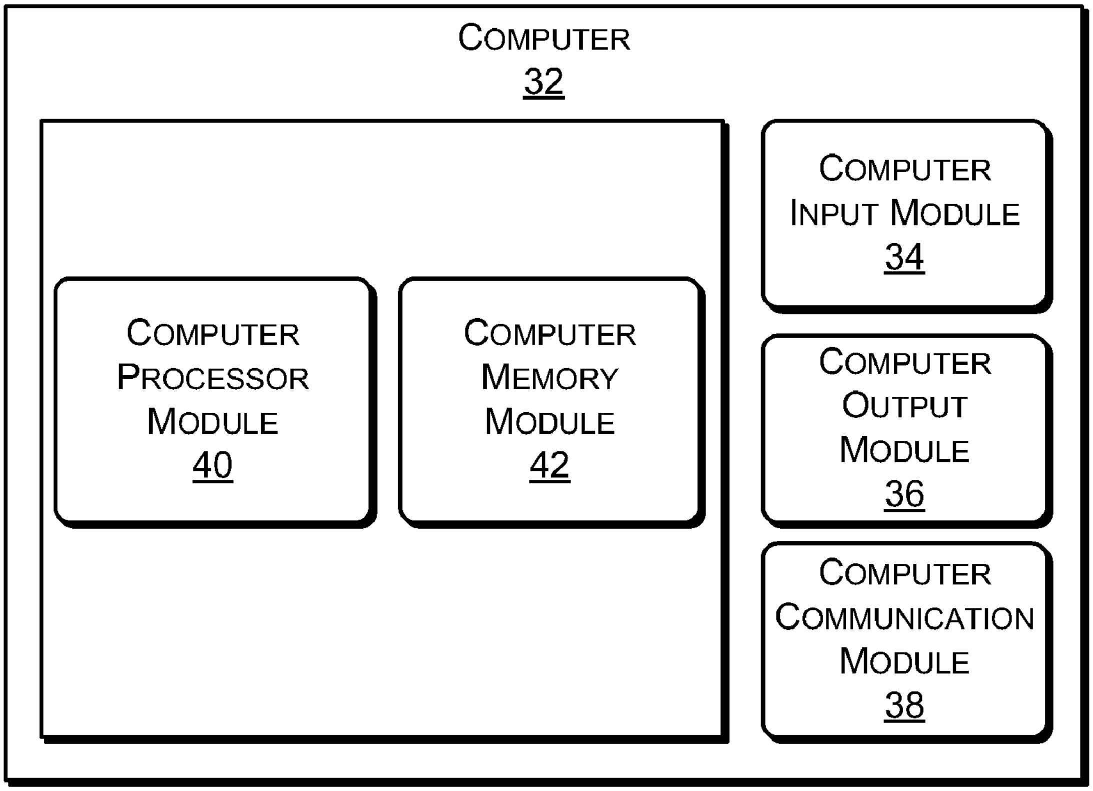

COMPUTER
32
COMPUTER PROCESSOR MODULE
40
COMPUTER MEMORY MODULE
42
COMPUTER INPUT MODULE
34
COMPUTER OUTPUT MODULE
36
COMPUTER COMMUNICATION MODULE
38

FIG. 3

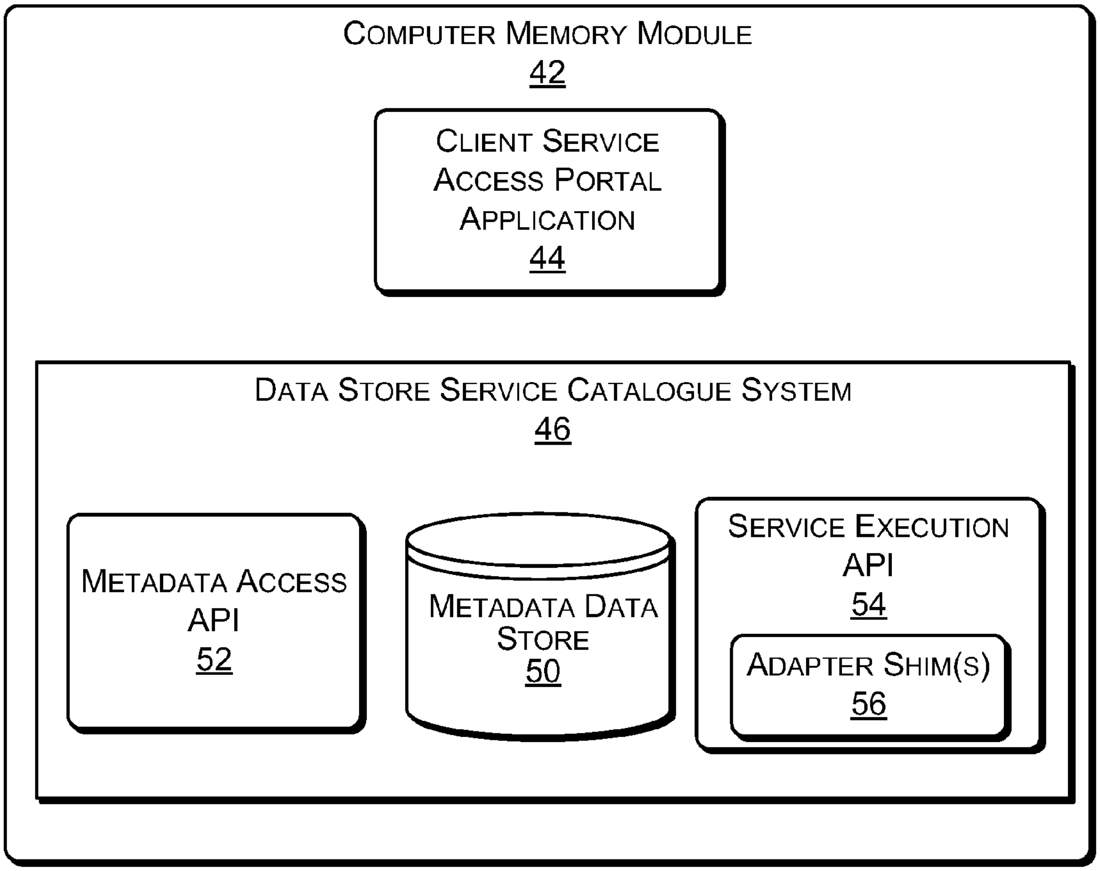
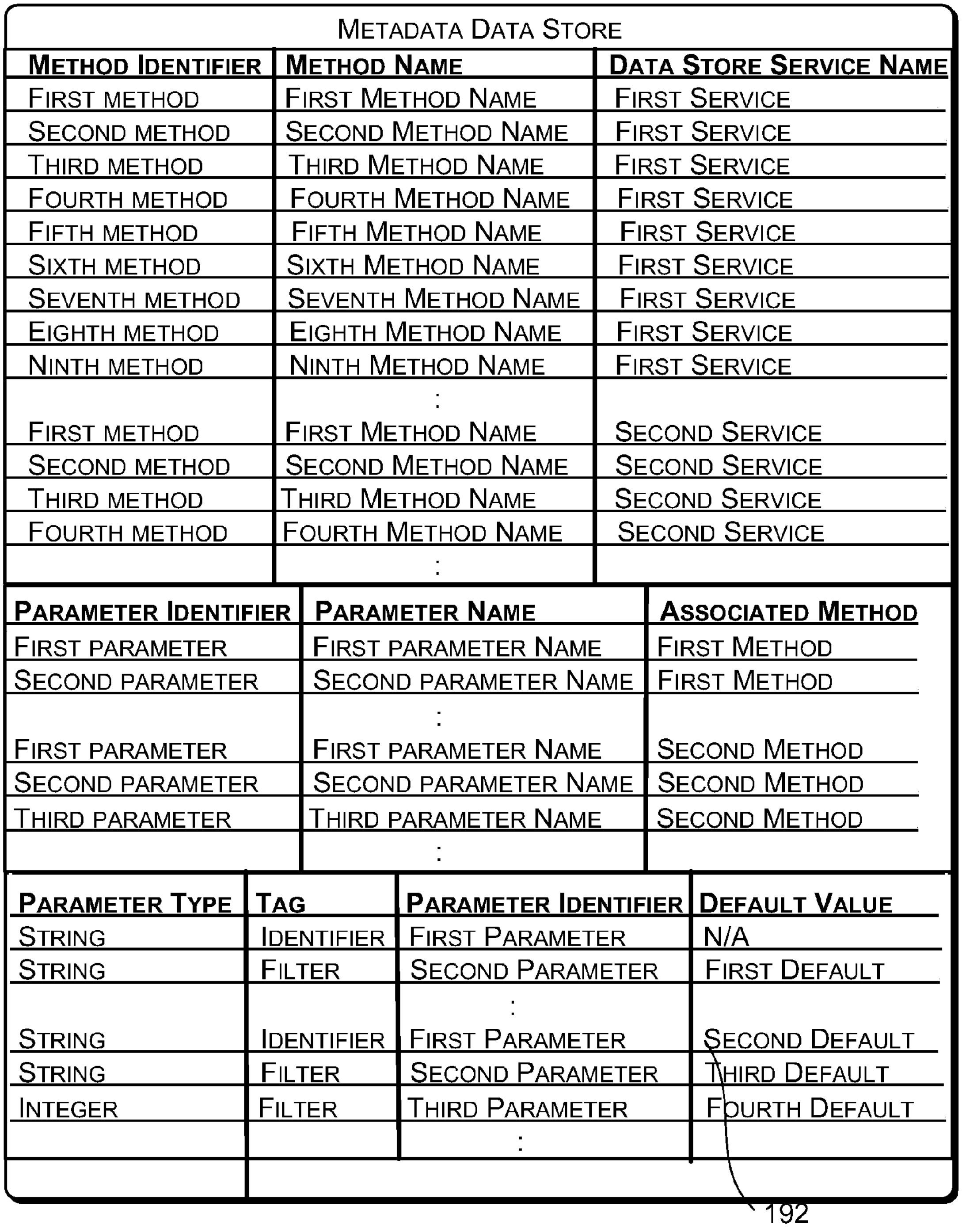

COMPUTER MEMORY MODULE
42
CLIENT SERVICE ACCESS PORTAL APPLICATION
44
DATA STORE SERVICE CATALOGUE SYSTEM
46
METADATA ACCESS API
52
METADATA DATA STORE
50
SERVICE EXECUTION API
54
ADAPTER SHIM(S)
56

FIG. 4



METADATA DATA STORE

METHOD IDENTIFIER	METHOD NAME	DATA STORE SERVICE NAME
FIRST METHOD	FIRST METHOD NAME	FIRST SERVICE
SECOND METHOD	SECOND METHOD NAME	FIRST SERVICE
THIRD METHOD	THIRD METHOD NAME	FIRST SERVICE
FOURTH METHOD	FOURTH METHOD NAME	FIRST SERVICE
FIFTH METHOD	FIFTH METHOD NAME	FIRST SERVICE
SIXTH METHOD	SIXTH METHOD NAME	FIRST SERVICE
SEVENTH METHOD	SEVENTH METHOD NAME	FIRST SERVICE
EIGHTH METHOD	EIGHTH METHOD NAME	FIRST SERVICE
NINTH METHOD	NINTH METHOD NAME	FIRST SERVICE
	:	
FIRST METHOD	FIRST METHOD NAME	SECOND SERVICE
SECOND METHOD	SECOND METHOD NAME	SECOND SERVICE
THIRD METHOD	THIRD METHOD NAME	SECOND SERVICE
FOURTH METHOD	FOURTH METHOD NAME	SECOND SERVICE
	:	

PARAMETER IDENTIFIER	PARAMETER NAME	ASSOCIATED METHOD
FIRST PARAMETER	FIRST PARAMETER NAME	FIRST METHOD
SECOND PARAMETER	SECOND PARAMETER NAME	FIRST METHOD
	:	
FIRST PARAMETER	FIRST PARAMETER NAME	SECOND METHOD
SECOND PARAMETER	SECOND PARAMETER NAME	SECOND METHOD
THIRD PARAMETER	THIRD PARAMETER NAME	SECOND METHOD
	:	

PARAMETER TYPE	TAG	PARAMETER IDENTIFIER	DEFAULT VALUE
STRING	IDENTIFIER	FIRST PARAMETER	N/A
STRING	FILTER	SECOND PARAMETER	FIRST DEFAULT
		:	
STRING	IDENTIFIER	FIRST PARAMETER	SECOND DEFAULT
STRING	FILTER	SECOND PARAMETER	THIRD DEFAULT
INTEGER	FILTER	THIRD PARAMETER	FOURTH DEFAULT
		:	

FIG. 5

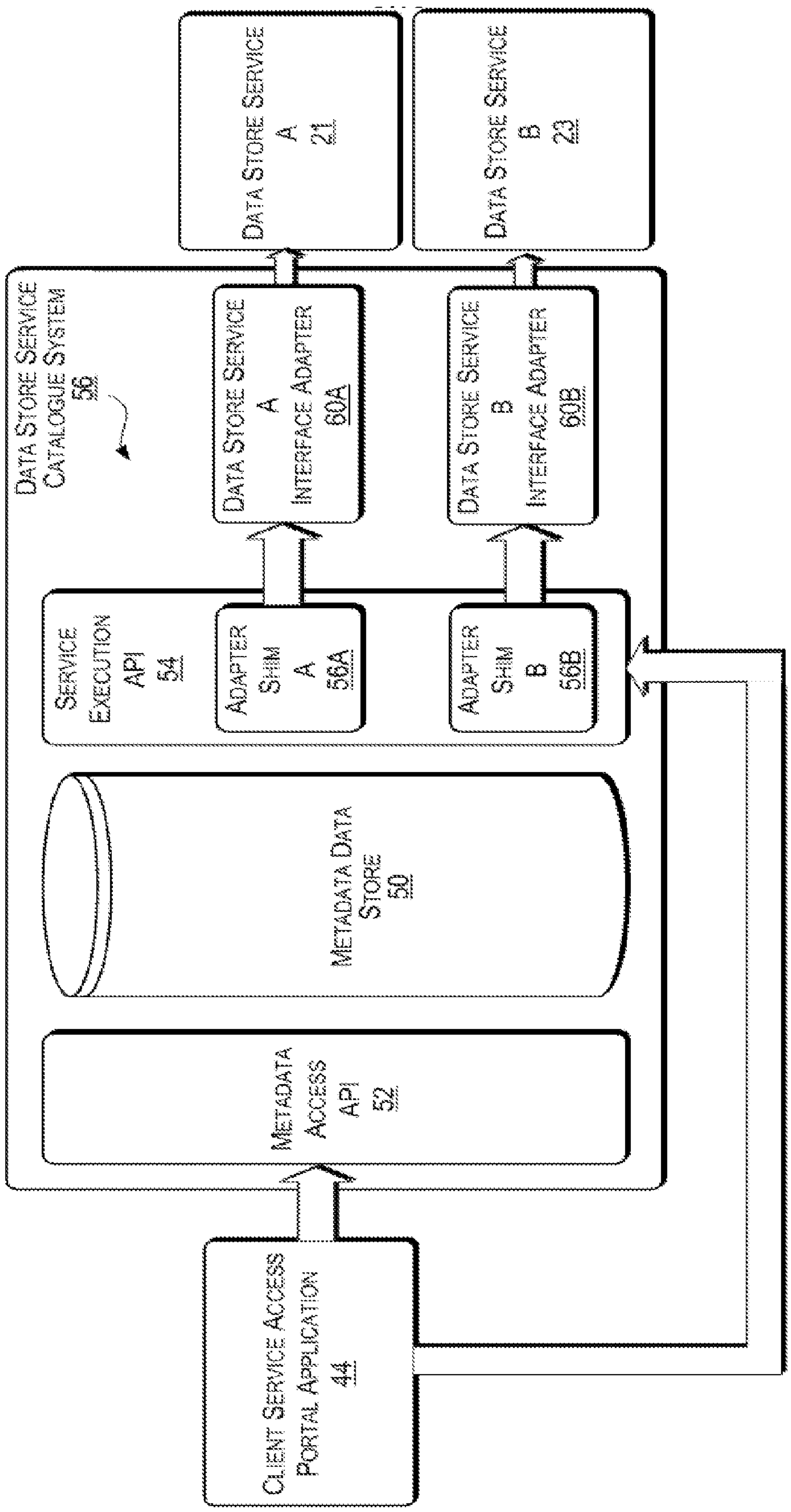

Data Store Service A 21
Data Store Service B 23
Data Store Service Catalogue System 56
Data Store Service A Interface Adapter 60A
Data Store Service B Interface Adapter 60B
Service Execution API 54
Adapter Shim A 56A
Adapter Shim B 56B
Metadata Data Store 50
Metadata Access API 52
Client Service Access Portal Application 44

FIG. 6

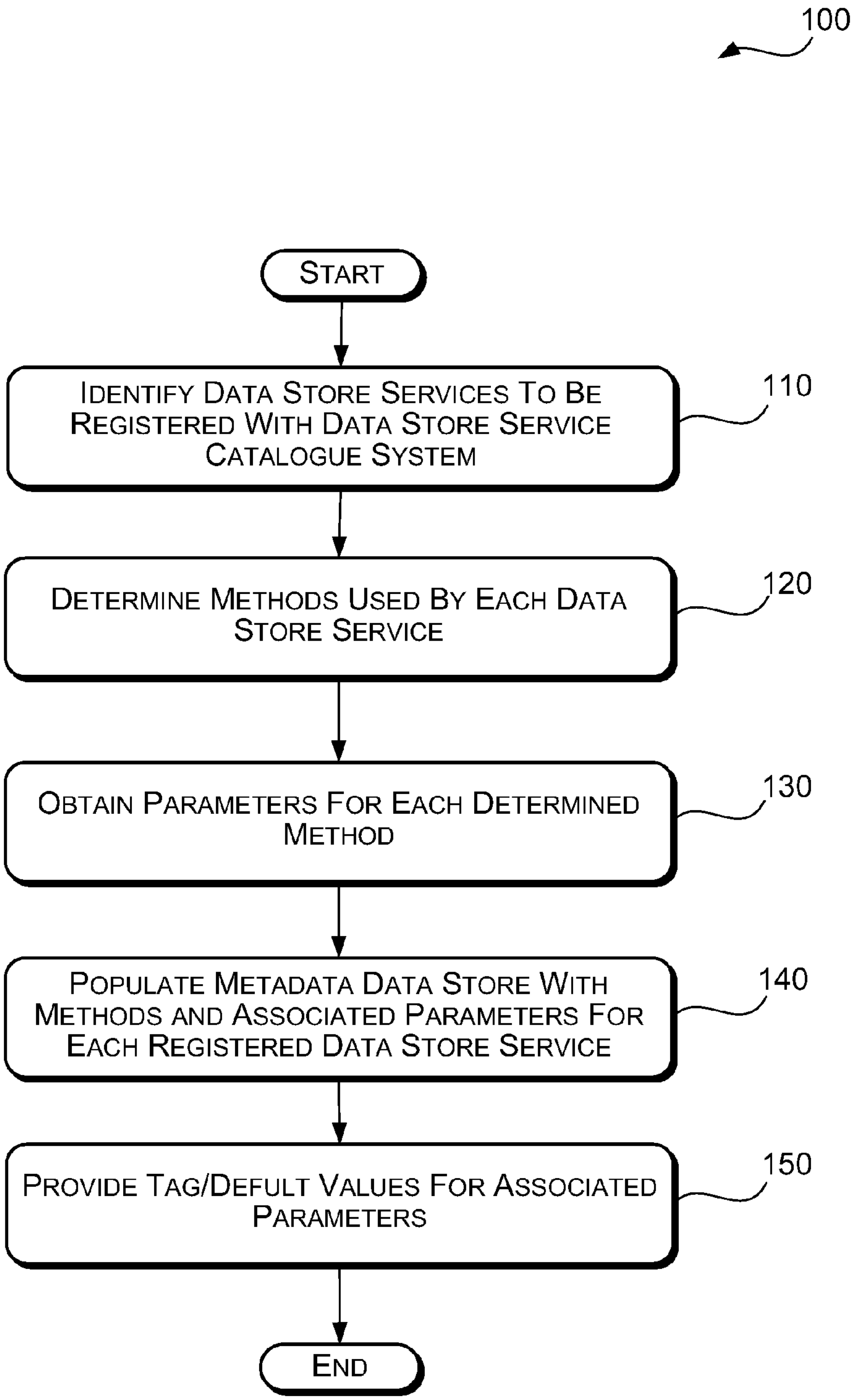

100
START
IDENTIFY DATA STORE SERVICES TO BE REGISTERED WITH DATA STORE SERVICE CATALOGUE SYSTEM
110
DETERMINE METHODS USED BY EACH DATA STORE SERVICE
120
OBTAIN PARAMETERS FOR EACH DETERMINED METHOD
130
POPULATE METADATA DATA STORE WITH METHODS AND ASSOCIATED PARAMETERS FOR EACH REGISTERED DATA STORE SERVICE
140
PROVIDE TAG/DEFULT VALUES FOR ASSOCIATED PARAMETERS
150
END

FIG. 7

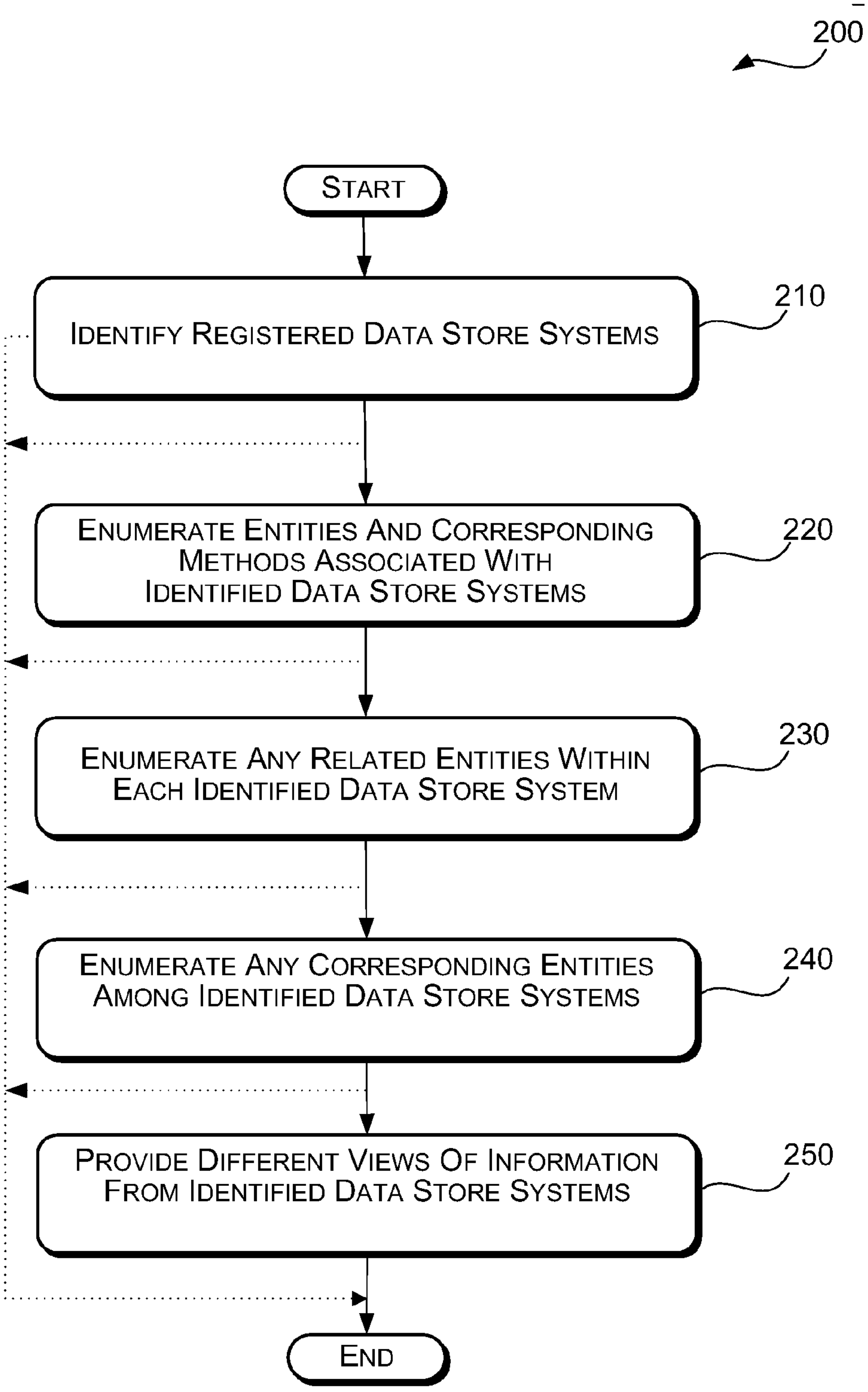
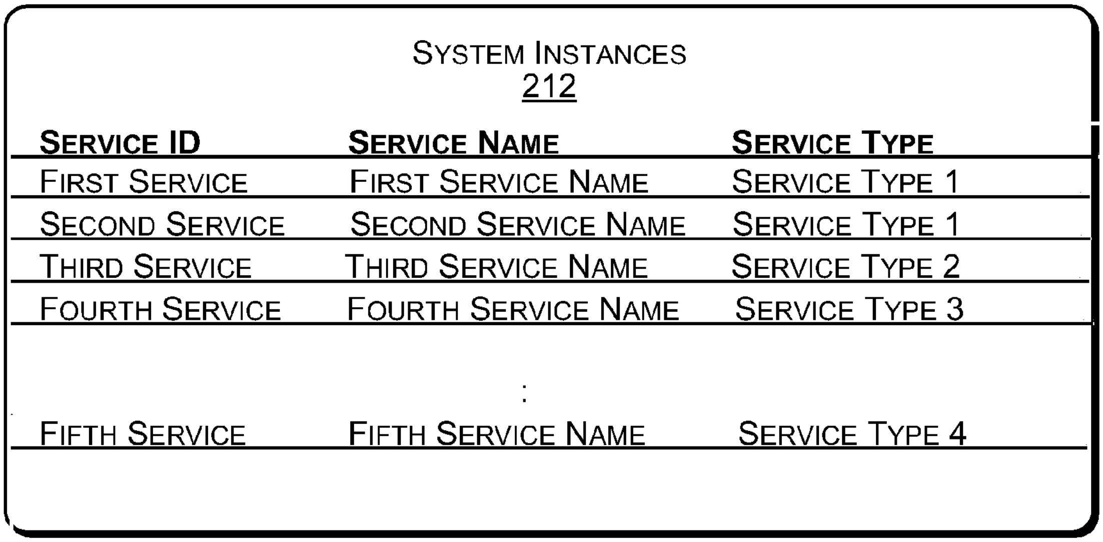

200
START
IDENTIFY REGISTERED DATA STORE SYSTEMS
210
ENUMERATE ENTITIES AND CORRESPONDING METHODS ASSOCIATED WITH IDENTIFIED DATA STORE SYSTEMS
220
ENUMERATE ANY RELATED ENTITIES WITHIN EACH IDENTIFIED DATA STORE SYSTEM
230
ENUMERATE ANY CORRESPONDING ENTITIES AMONG IDENTIFIED DATA STORE SYSTEMS
240
PROVIDE DIFFERENT VIEWS OF INFORMATION FROM IDENTIFIED DATA STORE SYSTEMS
250
END

FIG. 8



SYSTEM INSTANCES
212

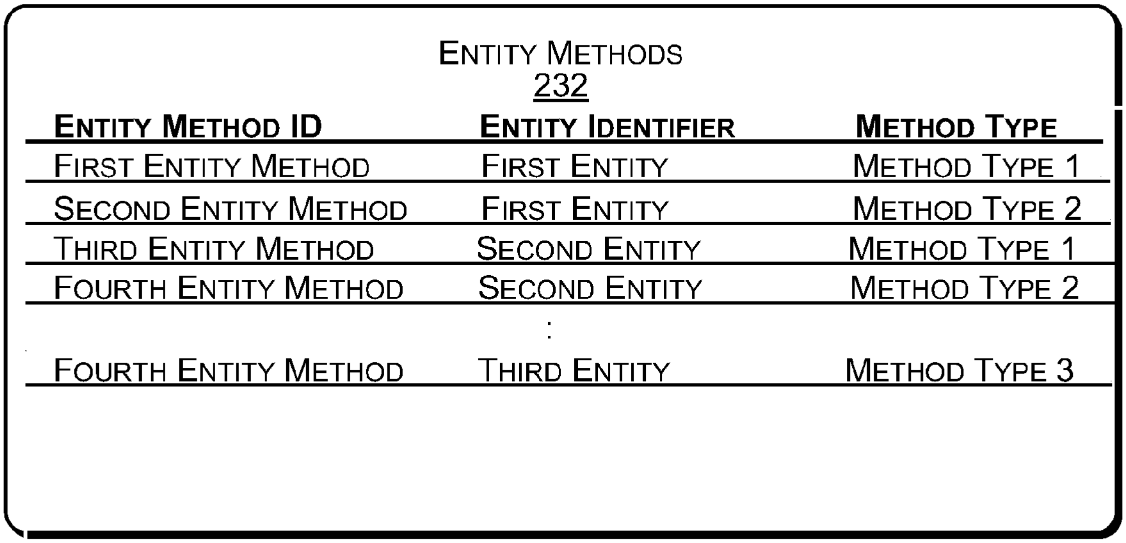
SERVICE ID	SERVICE NAME	SERVICE TYPE
FIRST SERVICE	FIRST SERVICE NAME	SERVICE TYPE 1
SECOND SERVICE	SECOND SERVICE NAME	SERVICE TYPE 1
THIRD SERVICE	THIRD SERVICE NAME	SERVICE TYPE 2
FOURTH SERVICE	FOURTH SERVICE NAME	SERVICE TYPE 3
	:	
FIFTH SERVICE	FIFTH SERVICE NAME	SERVICE TYPE 4

FIG. 9

ENTITIES
222

ENTITY ID	SYSTEM IDENTIFIER	ENTITY TYPE
FIRST ENTITY	FIRST SERVICE	ENTITY TYPE 1
SECOND ENTITY	FIRST SERVICE	ENTITY TYPE 2
THIRD ENTITY	SECOND SERVICE	ENTITY TYPE 1
FOURTH ENTITY	SECOND SERVICE	ENTITY TYPE 2
	:	
FIFTH ENTITY	THIRD SERVICE	ENTITY TYPE 3

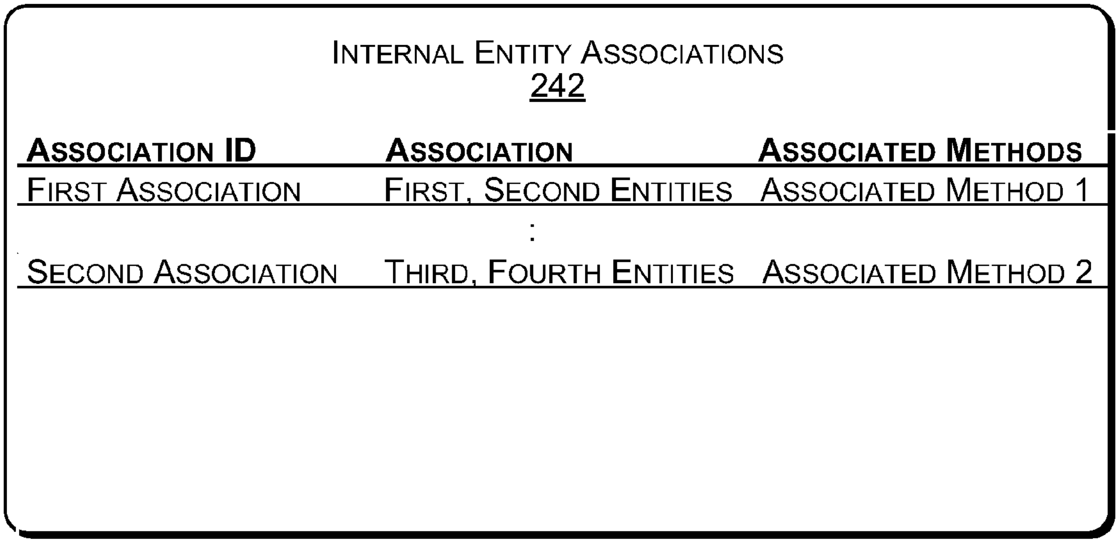
FIG. 10



ENTITY METHODS
232

ENTITY METHOD ID	ENTITY IDENTIFIER	METHOD TYPE
FIRST ENTITY METHOD	FIRST ENTITY	METHOD TYPE 1
SECOND ENTITY METHOD	FIRST ENTITY	METHOD TYPE 2
THIRD ENTITY METHOD	SECOND ENTITY	METHOD TYPE 1
FOURTH ENTITY METHOD	SECOND ENTITY	METHOD TYPE 2
	:	
FOURTH ENTITY METHOD	THIRD ENTITY	METHOD TYPE 3

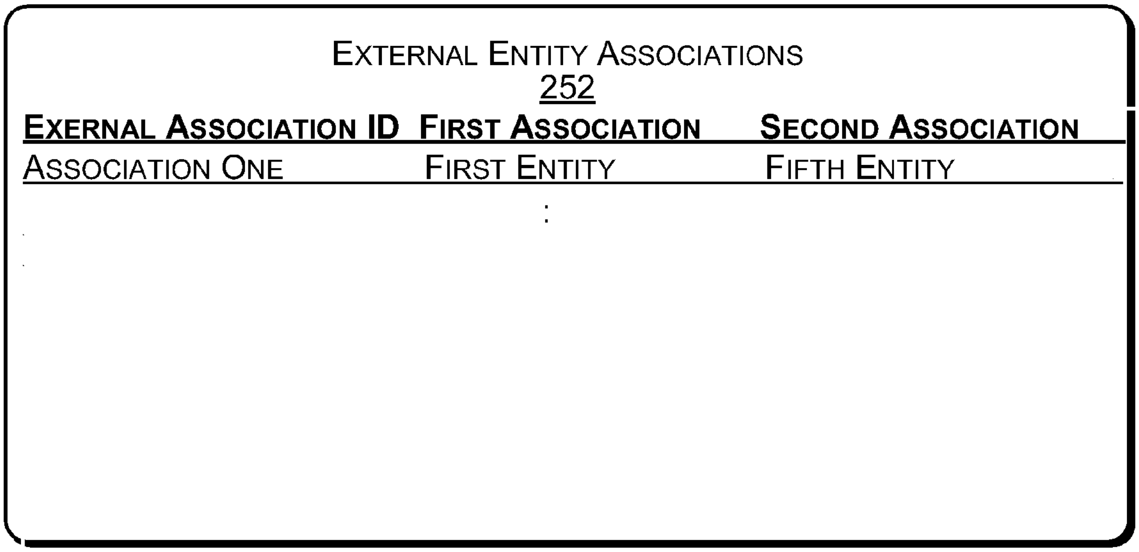
FIG. 11



INTERNAL ENTITY ASSOCIATIONS
242

ASSOCIATION ID	ASSOCIATION	ASSOCIATED METHODS
FIRST ASSOCIATION	FIRST, SECOND ENTITIES	ASSOCIATED METHOD 1
	:	
SECOND ASSOCIATION	THIRD, FOURTH ENTITIES	ASSOCIATED METHOD 2

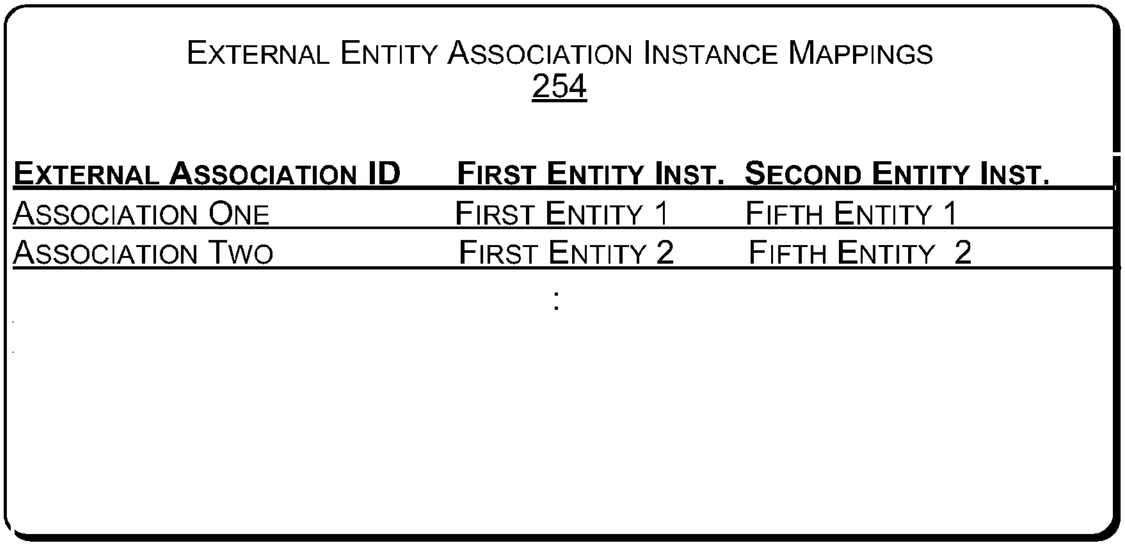
FIG. 12



EXTERNAL ENTITY ASSOCIATIONS
252

EXERNAL ASSOCIATION ID	FIRST ASSOCIATION	SECOND ASSOCIATION
ASSOCIATION ONE	FIRST ENTITY	FIFTH ENTITY
	:	

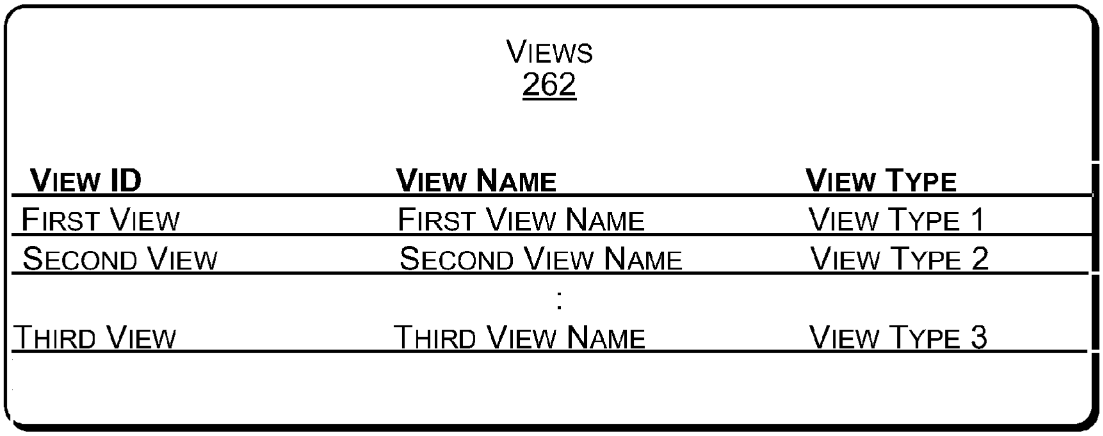
FIG. 13



External Entity Association Instance Mappings
254

External Association ID	First Entity Inst.	Second Entity Inst.
Association One	First Entity 1	Fifth Entity 1
Association Two	First Entity 2	Fifth Entity 2
	:	

FIG. 14



VIEWS
262

VIEW ID	VIEW NAME	VIEW TYPE
FIRST VIEW	FIRST VIEW NAME	VIEW TYPE 1
SECOND VIEW	SECOND VIEW NAME	VIEW TYPE 2
	:	
THIRD VIEW	THIRD VIEW NAME	VIEW TYPE 3

FIG. 15

VIEW DEFINITIONS
264

VIEW TYPE	**VIEW FIELDS**
VIEW TYPE 1	FIELD 1, … ,FIELD N
VIEW TYPE 2	FIELD 7, …, FIELD N
:	
VIEW TYPE 3	FIELD 1

FIG. 16

DISPARATE DATA STORE SERVICES CATALOGUED FOR UNIFIED ACCESS

PRIORITY CLAIM

[0001] This application is a Continuation-In-Part under 35 U.S.C. §120 of U.S. patent application Ser. No. 11/165,748 filed on Jun. 23, 2005, which is incorporated by reference herein in its entirety.

TECHNICAL FIELD

[0002] The disclosed technology relates generally to data processing and, more particularly, to enabling clients to discover and interact with one or more entities in one or more registered data store services in a number of ways using one or more uniform access interfaces.

BACKGROUND

[0003] Organizations may often use or desire using line-of-business ("LOB") systems to carry out daily operations, for example. Further, many organizations may implement a number of different types of LOB systems to perform particular tasks, such as SAP, SIEBLE and other types of LOB systems. While these LOB systems may potentially offer a great deal of benefits, at the same time organizations may find themselves unexpectedly devoting other resources just to be able to utilize these systems.

[0004] For instance, an organization's developer(s) may need to familiarize themselves with the semantics for interacting with each type of different LOB system. While SAP and SIEBLE were mentioned above as examples of LOB systems, any one organization may desire employing other types of LOB systems. Moreover, organizations may desire utilizing several instantiations of a particular LOB system (e.g., SAP) each dedicated to handling particular business related operations in addition to those other types of systems mentioned above. As a result, still more of an organization's resources would need to be invested to be able to leverage those systems.

SUMMARY

[0005] The following presents a simplified summary of the subject matter disclosed in further detail herein to provide a basic understanding to the reader. This summary is not an exhaustive or limiting overview of the disclosed subject matter, and is not provided for identifying key and/or critical elements of the subject matter or delineating the scope of the claimed subject matter. Thus, the scope of the claimed subject matter should not be limited in any way by this summary. Its sole purpose is to present some of the concepts in a simplified form as an introduction to the more detailed description that is presented later.

[0006] The present example provides a data store catalogue service model that may be implemented as a data store catalogue service system **56** in the manner described herein with regard to **FIGS. 5 and 7**. The data store catalogue service model may be implemented to expose one or more disparate data store related services to one or more clients without requiring the clients to have explicit knowledge on how to interact with each or any of the disparate services. The data store catalogue service system **56** may maintain interaction details for the data store related services, which the clients may access for interacting with those services. In particular, the data store catalogue service system **56** may implement a metadata data store **50** that may describe the different types of services available for accessing by clients, the data types used by those services, how to access the data provided by the services, and how to communicate semantically with the services for accessing the service's data. The clients may issue one or more requests to access those particular services defined in the metadata data store via one or more application program interfaces ("APIs") exposed to the clients by a data store service catalogue system in a unified manner.

[0007] Many of the attendant features will be more readily appreciated as the same becomes better understood by reference to the following detailed description considered in connection with the accompanying drawings.

DESCRIPTION OF THE DRAWINGS

[0008] The foregoing summary will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

[0009] **FIG. 1** is a block diagram of at least a portion of a system for registering data store services in cataloguing systems to provide clients with access to disparate data store services in a unified manner;

[0010] **FIG. 2** is a block diagram of an application server that may be used in the system illustrated in **FIG. 1**;

[0011] **FIG. 3** is a block diagram of a computer that may be used in the system illustrated in **FIG. 1**;

[0012] **FIG. 4** is a block diagram of a client service access portal application and a data store service catalogue system that may be implemented in the system illustrated in **FIG. 1**;

[0013] **FIG. 5** is a diagram of at least a portion of a metadata data store that may be implemented in the data store service catalogue system illustrated in **FIG. 4**;

[0014] **FIG. 6** is a functional block diagram illustrating at least a portion of a process for registering data store services in cataloguing systems to provide clients with access to disparate data store services in a unified manner;

[0015] **FIG. 7** is a flow chart of a process for registering data store services in cataloguing systems to provide clients with access to disparate data store services in a unified manner;

[0016] **FIG. 8** is a flow chart of a process for enabling clients to discover and interact with one or more entities in one or more registered data store services; and

[0017] **FIGS. 9-16** are diagrams one or more other portions of a metadata data store that may be implemented in the data store service catalogue system illustrated in **FIG. 4** that may be used in accordance with the process for enabling clients to discover and interact with one or more entities in one or more registered data store services.

[0018] Like reference numerals are used to designate like parts in the accompanying drawings.

DETAILED DESCRIPTION

[0019] The detailed description provided below in connection with the appended drawings is intended as a description of the present examples and is not intended to represent the only forms in which the present example may be constructed or utilized. The description sets forth the functions of the example and the sequence of steps for constructing and operating the example. However, the same or equivalent functions and sequences may be accomplished by different examples.

[0020] An exemplary data store catalogue service system **56** implemented in a system **10** and method **100** for registering data store services in cataloguing systems to provide clients with access to disparate data store services in a unified manner is generally shown in **FIGS. 1, 4**, **6** and **7**. As mentioned above earlier, organizations may often use or desire utilizing a number of different types of LOB systems to perform particular tasks, although organizations may desire implementing other types of software applications besides LOB systems, such as Web services, which may involve the same resource investments for enabling an organization's client applications to use the different types of systems.

[0021] In particular, an organization may wind up devoting resources for finding the different systems and identifying their purpose. In addition, these organizations may find themselves devoting even more resources just to be able to programmatically access each of the respective systems' methods, such as for allowing them to develop their own user or other types of access interfaces for accessing the systems. Unfortunately, many such systems typically involve uniquely formatted ways for accessing information from each system's respective data store. For instance, unique parameters, filters and other access nuances often make it difficult for organizations to develop interfaces for accessing the various systems unless the organization's software application developers possess intimate knowledge of each system's semantics.

[0022] Data store catalogue service system **56** may be implemented in system **10** in an attempt to address at least some of the issues noted above by providing client applications with one or more unified interfaces for interacting with one or more disparate systems without requiring specific knowledge of the inner workings of those systems, although there may be other uses as well. Basically, data store catalogue service system **56** may provide an environment where one or more disparate systems may be registered in one or more data stores using metadata, for example. The metadata may define methods, parameters and/or default values for one or more of the data store services that may allow clients to be able to interact with the registered services. As a result, application developers may be insulated from having to explicitly program their applications to be able to invoke the appropriate method calls and to process the various types of return values produced by each of the different types of systems.

[0023] In particular, data store catalogue service system **56** may implement a metadata data store **50** that may describe the different types of services available for accessing by clients, the data types used by those services, how to access the data provided by the services, and how to communicate semantically with the services for accessing the service's data. The clients may issue one or more requests to access those particular services defined in the metadata data store via one or more application program interfaces ("APIs") exposed to the clients by a data store service catalogue system in a unified manner.

[0024] Referring now specifically to **FIG. 1**, an example of a suitable operating environment in which system **10** may be implemented is illustrated. As shown in **FIG. 1**, system **10** may generally include at least one computer **32** and one or more application servers **20** that may be coupled together via network **80**, although system **10** may include a lesser or greater number and other types of devices.

[0025] It should be noted, however, that **FIG. 1** is not intended to suggest any limitation as to the scope of use or functionality of the system **10**. Other types of computing systems, environments, and/or configurations that may be suitable for use with the system **10** include, but are not limited to, hand-held, notebook or laptop devices, multiprocessor systems, microprocessor-based systems, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0026] Referring now to **FIG. 2**, an exemplary configuration for one or more of the application servers **20** is illustrated. In its most basic configuration, application server **20** may comprise application server input module **22**, application server output module **24**, application server communication module **26**, application server processor module **28** and application server memory module **30**, which may be coupled together by one or more bus systems or other communication links, although application server **20** may comprise other modules in other arrangements. Modules **22**, **24**, **26**, **28** and **30** will now be described below with continued reference to **FIG. 2**.

[0027] Application server input module **22** may comprise one or more user input devices, such as a keyboard and/or mouse, and any supporting hardware. Application server input module **22** may enable a user who is operating application server **20** to generate and transmit signals or commands to application server processor module **28**, although other types of user input devices may be used.

[0028] Application server output module **24** may comprise one or more user output devices, such as a computer monitor (e.g., CRT, LCD or plasma display), and any supporting hardware, although other types of output devices may be used. Application server output module **24** may present one or more results from application server processor module **28** executing instructions stored in application server memory module **30** as described in further detail herein below.

[0029] Application server communication module **26** may comprise one or more communication interface devices, such as wire-based (e.g., Ethernet) or wireless network adapters, and any supporting hardware, although other types of communication interface devices may be used, such as serial port interfaces (e.g., RS-232). Application server communication module **26** may enable application server **20** to transmit data to and receive data from other devices via network **80**, such as computer **32**, although application server communication module **26** may transmit/receive data to/from other computing systems or peripherals (e.g., external memory storage device or printer) via other communication media, such as direct cable connections, for example.

[**0030**] Application server processor module **28** may access data and may execute instructions stored in application server memory module **30** for controlling, monitoring and managing (hereinafter referred to as "operating") application server input module **22**, application server output module **24**, application server communication module **26** and application server memory module **30** as described herein, although some or all of the data and instructions may be stored in and/or executed by the modules themselves.

[**0031**] Additionally, application server processor module **28** may access data and may execute instructions stored in application server memory module **30** to perform functions for implementing at least a portion of the method **100** as described herein and illustrated in **FIG. 7**, although application server processor module **28** may perform other functions, one or more other processing devices or systems may perform some or all of these functions, and application server processor module **28** may comprise circuitry configured to perform the functions described herein.

[**0032**] Application server memory module **30** may comprise one or more types of fixed and/or portable memory accessible by application server processor module **28**, such as ROM, RAM, SRAM, DRAM, DDRAM, hard and floppy-disks, optical disks (e.g., CDs, DVDs), magnetic tape, ferroelectric and ferromagnetic memory, electrically erasable programmable read only memory, flash memory, charge coupled devices, smart cards, or any other type of computer-readable media, which may be read from and/or written to by one or more magnetic, optical, or other appropriate reading and/or writing systems coupled to application server processor module **28** and/or one or more other processing devices or systems.

[**0033**] Application server memory module **30** may store at least a portion of the data and instructions that may be accessed and/or executed by application server processor module **28** for operating application server communication module **26** and application server memory module **30**, although some or all of the data and instructions may be stored elsewhere, such as in the modules themselves and/or the application server processor module **28**.

[**0034**] Application server memory module **30** in each application server **20** may also store one or more modules, such as data store service "A"**21** and/or data store service "B"**23** shown in **FIG. 1**, although the modules may be stored elsewhere. Further, data store service "A"**21** and/or data store service "B"**23** may comprise an LOB system, such as SAP or SIEBLE, although data store service "A"**21** and/or data store service "B"**23** may comprise other types of LOB systems or other types of data store services, such as Web services, or any other type of server application. LOB systems, such as SAP and SIEBEL, are types of enterprise or business related software.

[**0035**] Referring to **FIG. 3**, in its most basic configuration, computer **32** may comprise computer input module **34**, computer output module **36**, computer communication module **38**, computer processor module **40**, and computer memory module **42**, which may be coupled together by one or more bus systems or other communication links, although computer **32** may comprise other elements in other arrangements. Modules **34**, **36**, **38**, **40** and **42** will now be described below with continued reference to **FIG. 4**.

[**0036**] Computer input module **34** may comprise one or more user input devices, such as a keyboard and/or mouse, and any supporting hardware. Computer input module **34** may enable a user who is operating computer **32** to generate and transmit signals or commands to computer processor module **40**, such as commands for operating client service access portal application **44**, although other types of user input devices may be used.

[**0037**] Computer output module **36** may comprise one or more user output devices, such as a computer monitor (e.g., CRT, LCD or plasma display), and any supporting hardware, although other types of output devices may be used. Computer output module **36** may present one or more results from computer processor module **40** executing instructions stored in computer memory module **42** as described in further detail herein below.

[**0038**] Computer communication module **38** may comprise one or more communication interface devices, such as a network interface card (e.g., Ethernet card or wireless network card), and any supporting hardware, although other types of communication interface devices may be used, such as a serial interface (e.g., RS-232 interface). Computer communication module **38** may enable computer **32** to transmit data to or receive data from other devices via network **80**, such as one or more application servers **20**, although computer communication module **38** may transmit/receive data to/from other computing systems or peripherals (e.g., external memory storage device or printer) via other communication media, such as a direct cable connection, for example.

[**0039**] Computer processor module **40** may access and/or may execute data and instructions stored in computer memory module **42** for operating computer input module **34**, computer output module **36**, computer communication module **38** and computer memory module **42** as described herein, although some or all of the data and instructions may be stored in and/or executed by the modules themselves.

[**0040**] Additionally, computer processor module **40** may access and/or may execute data and instructions stored in computer memory module **42** to perform functions for implementing at least a portion of the method **100** described with reference to **FIG. 7**, although computer processor module **40** may perform other functions, one or more other processing devices or systems may perform some or all of these functions, and computer processor module **40** may comprise circuitry configured to perform the functions described herein.

[**0041**] Computer memory module **42** may comprise the same types of memory storage devices as application server memory module **30** in application server **20**, although other types of computer-readable media may be used, which may be read from and/or written to by one or more magnetic, optical, or other appropriate reading and/or writing systems coupled to computer processor module **40** and/or one or more other processing devices or systems.

[**0042**] Computer memory module **42** may store at least a portion of the data and instructions that may be accessed and/or executed by computer processor module **40** for operating computer input module **34**, computer output module **36**, computer communication module **38**, computer processor module **40** and computer memory module **42**, although some or all of the data and instructions may be stored elsewhere, such as in the modules themselves and/or the computer processor module **40**.

[**0043**] Computer memory module **42** may also store client service access portal application **44** and data store service catalogue system **46**, although the application **44** and system **46** may be stored elsewhere. Client service access portal application **44** and data store service catalogue system **46** may comprise data and/or instructions written in a number of programming languages, which when accessed and/or executed by computer processor module **40**, may cause computer **32** to implement at least a portion of the method **100** described with reference to **FIG. 7**, although the modules may comprise circuitry configured to operate in the manner described herein.

[**0044**] Client service access portal application **44** may access metadata access API **52** to interact with at least one of data store service "A"**21** or data store service "B"**23** implemented on one or more servers **20** as will be described in further detail herein below. Data store service catalogue system **46** may comprise metadata access API **52**, metadata data store **50**, and service execution API **54**, although system **56** may comprise other components in other arrangements. Metadata data store **50** may comprise one or more metadata entries, which may define one or more methods, parameters associated with the methods, and other information related to the semantics for interacting with one or more of data store services "A"**21** and/or "B"**23** that may be implemented on servers **20**, although data store **50** may comprise one or more other metadata entries for defining the semantics for interacting with other services or other types of entries besides metadata entries.

[**0045**] Metadata access API **52** may comprise one or more application program interfaces that may accept one or more methods calls and/or parameters for accessing one or more entities that may be defined in metadata data store **50** and which may correspond to one or more entities associated with one or more of data store services "A"**21** and/or "B"**23** that may be implemented on one or more application servers **20**.

[**0046**] Service execution API **54** may comprise one or more application program interfaces that may accept one or more methods calls and/or parameters from data store service catalogue system **56** based on the metadata obtained from metadata data store **50** for a particular service **21** and/or **23** that may be implemented on servers **20**. Further, service execution API **54** may comprise one or more adapter shims **56** for each of the services **21** and/or **23** that may be implemented on servers **20**, such as adapter shim "A"**56**A and adapter shim "B"**56**B shown in **FIG. 7**. Adapter shims **56** may comprise low level information that service execution API **54** may use to communicate with data store service interface adapters **60**A, **60**B shown in **FIG. 6**, such as information that may specify how to invoke one or more methods associated with data store services **21** and/or **23** that may be exposed by adapter **70**. An example includes an SAP NET Connector software component where data store services **21** and/or **23** may comprise an SAP LOB service, for instance.

[**0047**] It should be appreciated that client service access portal application **44**, data store service catalogue system **46** and their associated modules as described above are depicted in the manner shown in **FIG. 4** for ease of description and exemplary purposes only. However, client service access portal application **44** and data store service catalogue system **46** may comprise a fewer or greater number and other types of modules that may reside on one or more other computing systems or devices.

[**0048**] Referring back to **FIG. 1**, network **80** may comprise a Wide Area Network ("WAN"), such as the Internet, although a variety of other communication systems and/or methods using appropriate protocols may be used, including other types of WANs, local area networks, wireless networks, phone lines, serial and/or parallel bus cables, coaxial cables, and combinations thereof.

[**0049**] An example of a method **100** for registering a data store service "A"**21** and a data store service "B"**23** will now be described with reference to **FIGS. 6-7** in the context of being carried out by the system **10** described above in connection with **FIGS. 1-5**.

[**0050**] Referring to **FIG. 7** and beginning at step **110**, by way of example only, a user of computer **32** may operate client service access portal application **44** using the computer's input system, in conjunction with operation of the computer processor module **40**, computer memory module **42** and computer communication module **38**, to identify which services (e.g., data store service "A"**21**, data store service "B"**23**) may be available in network **80** for registration with data store service catalogue system **56**.

[**0051**] At step **120**, the user of computer **32**, such as a developer, may determine which one or more methods may be used by each data store service identified above at step **110**.

[**0052**] At step **130**, the user may identify one or more parameters that may be used for each method identified above at step **120**.

[**0053**] At step **140**, the user may populate metadata data store metadata data store **50** in the data store service catalogue system **46** with the methods and associated parameters for each data store service to register each service, an example of which is illustrated in **FIG. 5**. It should be noted that the exemplary parameters and manner in which the tables are organized in **FIG. 5** are provided for ease of illustration and descriptive purposes only, as the particular parameters provided and the manner in which they are organized may vary depending on the particular environment they are employed. Further, the examples provided in **FIG. 5** may include more or less information for each parameter as required for the particular data services each parameter may be associated with.

[**0054**] At step **150**, client service access portal application **44** may provide tag values for associated parameters, such as whether the particular parameter represents an identifier or a filter, although other parameters may be provided. Further, client service access portal application **44** may provide a default value for one or more of the associated parameters included in metadata data store **50**. Any default value provided for a particular parameter may depend on the particular data store service that the parameter may be associated with, such as data store service "A"**21** or data store service "B"**23**, as well as the type of parameter it is provided for as identified by the parameter's "Tag" entry in the metadata data store **50**. shown in **FIG. 5**.

[**0055**] For instance, a first default value **192** shown in **FIG. 5** may specify a particular format for specifying the format in which date values may be filtered when sent back to the client application **44**, for example. As an example, the default value **192** shown in **FIG. 5** may specify "[Date format=xx/xx/xxxx]," although any other values may be provided. Upon defining each of the tag values and any default values for the parameters, the method **100** may end.

[**0056**] An example of a method **200** for enabling clients to discover and interact with one or more entities in one or more registered data store services, such as data store service "A"**21** and a data store service "B"**23**, will now be described with reference to **FIGS. 8-16** in the context of being carried out by the system **10** described above in connection with **FIGS. 1-7**.

[**0057**] Referring to **FIG. 8** and beginning at step **210**, by way of example only, a user of computer **32**, such as an organization's developer, may operate client service access portal application **44** using the computer's input system, in conjunction with operation of the computer processor module **40**, computer memory module **42** and computer communication module **38**, to access the data store service catalogue system **46** for a number of reasons, such as for querying the module **38** to identify which services (e.g., data store service "A"**21**, data store service "B"**23**) may be registered in metadata data store **50**, for instance.

[**0058**] Further, a developer may desire querying the module **38** to identify which data store services may be registered in metadata data store **50** so that the developer can develop code that may enable one or more other client computers on the network **80** to access the data store services in a particular manner. However, the developer may not be well versed in coding for one or more of the various data store services that may be registered within the metadata data store **50** for which providing access to the one or more other clients may be desired. Thus, the metadata that may be maintained in metadata data store **50** for each registered data store service may enable the developer with limited coding knowledge to develop code for leveraging one or more of the registered data services.

[**0059**] Further, the metadata data store **50** may be populated with metadata describing one or more services (e.g., data store service "A"**21**, data store service "B"**23**) registered therein using any number of methods, such as the process described above in connection with method **100**, for example, although other methods could be used. For instance, the developer may provide the following line of code to the client service access portal application **44**, which in turn would send the code to the data store service catalogue system **46** for further processing as described herein:

[**0060**] IList e=System.GetSystem;

[**0061**] The Systeminstance.GetSysteminstances( ) method may be made available to the client service access portal application **44** by the data store service catalogue system **46** via the metadata access API, for example. Responsive to receiving the above-identified from the client application **44**, the data store service catalogue system **46** may search for any registered data store services in the metadata data store **50**.

[**0062**] Referring to **FIG. 9**, an exemplary portion of information that may be maintained in the metadata data store **50** to represent one or more registered instances of data store services is illustrated as a system instances **212** table. It should be appreciated that the information illustrated in **FIG. 9** is provided for ease of illustration and descriptive purposes only. As shown in **FIG. 9**, one or more data store service instances are identified under a "Service ID" column and/or a "Service Name" column.

[**0063**] Further, a "Service Type" column in the system instances **212** table may identify the particular type of data store service (e.g., SAP, Sieble). For instance, system instances **212** table shows that there are two registered data store services of "Service Type"="Service Type 1," which represents two instances of the same type of data store service. When the data store service catalogue system **46** identifies the data store services registered in the metadata data store **50**, a reference to a listing or enumeration of the services may be made available to the client service access portal application **44**, and hence the developer, for example.

[**0064**] At step **220**, the developer operating the client service access portal application **44** running on computer **32** may desire determining which "entities" may be associated with and/or exposed by one or more of the registered data store service instances that may have been identified above at step **210** via one or more API's exposed by those services, for example. An entity may represent data and/or one or more methods or logic defined within the registered data store service's native environment that can be implemented by data store service catalogue system **56**. For instance, an SAP data store service may define a "Customer" entity and an "Employee" entity that may be associated with one or more methods or logic, for instance, although other entities may be defined. Further, the one or more methods, associated parameters and/or default parameter values that may be associated with each entity associated with a particular data store service instance may be defined by the data store service catalogue system **46** in the metadata data store **50**, such as in the exemplary metadata provided in **FIG. 5** earlier, for instance.

[**0065**] For a specific example, the developer may provide the following line of code to the client service access portal application **44**, which in turn would send the code to the data store service catalogue system **46** for further processing as described herein:

[**0066**] IList e=System.GetSystem.

[**0067**] GetClasses;

[**0068**] Referring to **FIG. 10**, an exemplary portion of information that may be maintained in the metadata data store **50** to represent one or more entities associated with each of the one or more registered data store services identified above at step **210** is illustrated as an entities instances **222** table. It should be appreciated that the information illustrated in **FIG. 10** is provided for ease of illustration and descriptive purposes only. As shown in **FIG. 10**, one or more data store service instances are identified under an "Entity ID" column. Further, a "System Identifier" column in the entities instances **222** table identifies which of the one or more registered data store services identified above at step **210** the particular entity may be associated with.

[**0069**] Referring back to the example provided above in **FIG. 9**, since there may be two different instances of the same registered data store services identified above at step **210**, each service instance may be associated with the same types of entities (e.g., entity types 1 and 2), although each entity may be defined within metadata data store **50** differently and referred to independently, if desired, for example.

[**0070**] Referring to **FIG. 11**, another exemplary portion of information that may be maintained in the metadata data store **50** is illustrated as an entity methods **232** table, which may identify one or more methods associated with each of the entities associated with the one or more registered data store services shown in **FIG. 10** discussed above. Again, the information illustrated in **FIG. 11** is provided for ease of illustration and descriptive purposes only, and the methods identified in the entity methods table **232** may be described or further annotated in the metadata data store **50** in the same manner shown and described above in connection with **FIG. 5**, although the methods may be defined in other ways.

[**0071**] When the data store service catalogue system **46** identifies the one or more entities and one or more methods associated with each of the one or more data store services registered in the metadata data store **50**, a reference to a listing or enumeration of these entities and associated methods may be made available to the client service access portal application **44**, and hence the developer, for example. Furthermore, each of the identified entities may support a number of methods, including but not limited to an instantiate( ) method for instantiating a particular entity, one or more find( ) and/or findSpecific methods for finding one or more instances of entities in one or more identified data store services registered in the metadata data store **50**, one or more getAssociation( ) methods for identifying one or more associations among one or more entities defined in the metadata data store **50**, one or more getExternalAssociation( ) methods for identifying one or more associations among one or more corresponding entities among one or more different data store services registered in the metadata data store **50**, and/or GetViews methods.

[**0072**] At step **230**, the developer operating the client service access portal application **44** running on computer **32** may desire determining whether any of the entities that may have been identified above at step **220** for each of the registered data store services that may have identified above at step **210** are related to each other. The relationships between the one or more entities may be described in the metadata data store **50** and may be based on one or more relationships defined in the native environment of the particular data store service that the entity may be associated with, for instance, although the metadata data store **50** may be extensible to define non-native entity associates for particular implementation environments, if desired. For example, an SAP data store service may define a relationship between a "Customer" entity and a "Sales Order" entity.

[**0073**] For a specific example, the developer may provide the following line of code to the client service access portal application **44** to enable retrieving any defined associations between the entities in one or more particular data store services, which in turn would send the code to the data store service catalogue system **46** for further processing as described herein:

[**0074**] IList e=Systeminstance.GetAssociations.

[**0075**] Referring to **FIG. 12**, an exemplary portion of information that may be maintained in the metadata data store **50** to represent one or more associations between one or more entities associated with each of the one or more registered data store services identified above at step **210** is illustrated as internal entities associations **242** table. It should be appreciated that the information illustrated in **FIG. 12** is provided for ease of illustration and descriptive purposes only. As shown in **FIG. 12**, one or more entity associations are identified under an "Association ID" column. Further, an "Association" column in the internal entities associations **242** table identifies which of the one or more entities are associated. Finally, an "Associated Methods" column in the internal entities associations **242** identifies one or more associated methods may be called to retrieve information representing the related entities.

[**0076**] For instance, if the "First Entity" represents a "Customer" and the "Second Entity" represents "Sales Orders" in an SAP data store service environment, then data store service catalogue system **56** would implement the appropriate SAP code that ay be defined in the metadata data store **50** to obtain "Sales Order" information for each "Customer" entity maintained by the SAP service, for example. When the data store service catalogue system **46** identifies any associations or relationships among the one or more entities that may be defined in the metadata data store **50**, a reference to a listing or enumeration of these entity associations may be made available to the client service access portal application **44**, and hence the developer, for example.

[**0077**] At step **240**, the developer operating the client service access portal application **44** running on computer **32** may desire determining whether any of the entities that may have been identified above at step **220** for each of the registered data store services that may have identified above at step **210** are related to any other corresponding entities in one or more different data store services that may have been identified above at step **210**. For instance, the developer may desire determining and/or leveraging the correspondence between SAP data store service "Customers" and any corresponding Sieble data store service "Customers," for example.

[**0078**] The relationships between the one or more corresponding entities in one or more different data store services may be described in the metadata data store **50**. However, these corresponding relationships often may not be defined in the native environment of the particular data store services that the corresponding entities may be associated with and thus this identifying these types of relationships may prove utility to the developer, for instance. For a specific example, the developer may provide the following line of code to the client service access portal application **44** to enable retrieving any defined associations between the entities in one or more particular data store services, which in turn would send the code to the data store service catalogue system **46** for further processing as described herein:

[**0079**] IList e=Systeminstance.GetExternalAssociations.

[**0080**] Referring to **FIG. 13**, an exemplary portion of information that may be maintained in the metadata data store **50** to represent one or more external associations between one or more corresponding entities in one or more different data store services identified above at step **210** is illustrated as external entity associations **252** table. It should be appreciated that the information illustrated in **FIG. 13** is provided for ease of illustration and descriptive purposes only. As shown in **FIG. 13**, one or more entity associations are identified under an "External Association ID" column.

[**0081**] Further, a "First Association" column in the external entities associations **252** table identifies which of the one or more entities from a first data store services may have an external association. Moreover, a "Second Association" column in the external entities associations **252** table identifies another one or more of the entities from a second data store service that may have an external association with the entity identified under the "First Association" in the same row. For instance, the "First Entity" may represent a "Customer" entity in an SAP data store service environment and the "Second Entity" may represents a corresponding "Customer" in a Sieble data store service environment.

[**0082**] Referring to **FIG. 14**, another exemplary portion of information that may be maintained in the metadata data store **50** is shown as external entity association instance mappings **254** in **FIG. 14** and may describe one or more instances of the associations shown in **FIG. 13**, for example. When the data store service catalogue system **46** identifies any associations or relationships among the one or more entities that may be defined in the metadata data store **50**, a reference to a listing or enumeration of these external entity associations may be made available to the client service access portal application **44**.

[**0083**] At step **250**, the developer operating the client service access portal application **44** running on computer **32** may desire specifying which properties or fields associated with a particular entity may be output and/or provided to clients that a client data service application may be coded for accessing by the clients. For instance, a "Customer" entity in an SAP data service environment may be associated with over 250 properties. Thus, the developer may desire limited the number of fields that may be displayed to clients to just a few relevant fields that may be appropriate in a particular environment, for example.

[**0084**] For a specific example, the developer may provide the following line of code to the client service access portal application **44** to enable using one or more predefined "views" that may be defined in the metadata data store **50**, which in turn would send the code to the data store service catalogue system **46** for further processing as described herein:

[**0085**] System.GetViews.ViewName.

[**0086**] Referring to **FIG. 15**, an exemplary portion of information that may be maintained in the metadata data store **50** to represent one or more available views that may be defined in the metadata data store **50** is illustrated as views **262** table. It should be appreciated that the information illustrated in **FIG. 13** is provided for ease of illustration and descriptive purposes only. As shown in **FIG. 15**, one or more views are identified under "View ID" and "View Name" columns. Further, a "View Type" column in the views **262** table identifies the particular type of view.

[**0087**] Referring to **FIG. 16**, another exemplary portion of information that may be maintained in the metadata data store **50** is shown as view definitions **264** table and may describe properties associated with one or more fields defined for each field type defined in the views **262** table shown in **FIG. 15**, for example. Thus, one or more views may be called for particular instances to allow the developer to manipulate which portions of information associated with one or more entities may be returned and/or presented.

[**0088**] It should be appreciated that steps **210-250** may be performed in any particular order and do not need to be performed in the manner depicted in **FIG. 8**. If the developer has input all desired portions of code in the manner described in one or more steps **210-250** the method **200** may end, although one or more portions may be repeated as desired.

[**0089**] It should be appreciated that while application server memory module **30** and computer memory module **42** illustrated in **FIGS. 2 and 3**, respectively, have been described above as comprising computer storage media, the memory modules **30** and **42** should be broadly interpreted to cover communication media as well. Communication media may embody computer-readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example only, communication media includes wired media, such as a wired network or direct-wired connection, and wireless media, such as acoustic, RF, infrared, other wireless media, and combinations thereof.

[**0090**] Further, while the present examples are described and illustrated herein as being implemented in a data store catalogue service system **56**, the system **56** described is provided as an example and not a limitation. As those skilled in the art will appreciate, the present examples are suitable for application in a variety of different types of data store catalogue service systems **56** systems.

[**0091**] Those skilled in the art will realize that storage devices utilized to store program instructions can be distributed across a network. For example a remote computer may store an example of the process described as software. A local or terminal computer may access the remote computer and download a part or all of the software to run the program. Alternatively the local computer may download pieces of the software as needed, or distributively process by executing some software instructions at the local terminal and some at the remote computer (or computer network). Those skilled in the art will also realize that by utilizing conventional techniques known to those skilled in the art that all, or a portion of the software instructions may be carried out by a dedicated circuit, such as a DSP, programmable logic array, or the like.

[**0092**] Further, while particular examples have been described, alternatives, modifications, variations, improvements, and substantial equivalents that are or may be presently unforeseen may arise to applicants or others skilled in the art. Accordingly, the appended claims as filed, and as they may be amended, are intended to embrace all such alternatives, modifications, variations, improvements, and substantial equivalents. Further, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims.

What is claimed is:

1. At least one computer-readable medium having at least one instruction stored thereon, which when executed by at least one processing system in conjunction with at least one application program, causes the at least one application program to access metadata maintained for one or more registered data store services, the at least one medium comprising at least one instruction for:

at least one first parameter in an access interface for requesting at least a portion of the metadata describing the one or more registered data store services; and

at least one second parameter in the unified access interface for providing the requested metadata describing the one or more registered data store services for enabling at least one requesting client to interact with the one or more described data stores services without requiring data store-specific code.

2. The at least one medium as set forth in claim 1 wherein the at least one first parameter comprises information requesting enumeration of the one or more registered data store services.

3. The at least one medium as set forth in claim 1 further comprising at least one third parameter for returning information identifying one or more registered data store services.

4. The at least one medium as set forth in claim 1 wherein the at least one first parameter comprises information requesting enumeration of at least one of either one or more entities or one or more classes associated with the one or more registered data store services.

5. The at least one medium as set forth in claim 1 further comprising at least one third parameter for returning information enumerating at least one of either one or more entities or one or more classes associated with the one or more registered data store services.

6. The at least one medium as set forth in claim 1 wherein the at least one first parameter comprises information requesting enumeration of one or more related entities.

7. The at least one medium as set forth in claim 6 wherein the one or more related entities comprise at least one of either corresponding entities from any one of the one or more registered data store services or corresponding entities from at least one of the registered data store services and at least another one of the registered data store services.

8. The at least one medium as set forth in claim 1 further comprising at least one third parameter for returning information enumerating at least one of either one or more related entities from any one of the one or more registered data store services or one or more related entities from at least one of the registered data store service and at least another one of the registered data store services.

9. The at least one medium as set forth in claim 1 wherein the at least one first parameter comprises view information identifying one or more property fields of one or more entities for which information is being requested.

10. At least one computer-readable medium having at least one instruction stored thereon, which when executed by at least one processing system, causes the at least one processing system to provide metadata maintained for one or more registered data store services that enables interacting with the registered data store services, the at least one medium comprising at least one instruction for:

identifying the one or more registered data store services maintained in a data store service registration repository that are associated with at least one interaction request from one or more clients; and

providing at least a portion of the metadata associated with the one or more identified data store services that can be used by the one or more clients for requesting at least one server to implement the at least one interaction request without requiring the clients to provide data store-specific code for implementing the at least one interaction request.

11. The medium as set forth in claim 10 further comprising obtaining data store-specific code for execution by the at least one server for implementing the at least one interaction request responsive to receiving at least another portion of the metadata associated with the one or more registered data store services in the data store service registration repository.

12. The medium as set forth in claim 11 wherein the at least one server executes the data store-specific code on behalf of the one or more clients to implement the at least one interaction request.

13. The medium as set forth in claim 10 wherein providing at least a portion of the metadata associated with the one or more identified data store services further comprises returning information naming the one or more identified data store services.

14. The medium as set forth in claim 10 wherein providing at least a portion of the metadata associated with the one or more identified data store services further comprises returning information enumerating at least one of either one or more entities or one or more classes associated with the one or more identified data store services.

15. The medium as set forth in claim 10 wherein providing at least a portion of the metadata associated with the one or more identified data store services further comprises returning information enumerating at least one of either corresponding entities from any one of the one or more identified data store services or corresponding entities from at least one of the identified data store services and at least another one of the identified data store services.

16. The medium as set forth in claim 10 wherein providing at least a portion of the metadata associated with the one or more identified data store services further comprises providing one or more property fields describing one or more entities for which information is being requested based upon one or more views identified by the one or more clients.

17. A method for discovering one or more objects associated with one or more registered data store services, the method comprising:

identifying one or more data store services registered in at least one data store repository that are associated with at least one client request for information describing the one or more data store services; and

providing metadata obtained from the at least one data store repository that describes the one or more registered data store services.

18. The method as set forth in claim 17 wherein providing metadata obtained from the at least one data store repository that describes the one or more registered data store services further comprises enumerating at least one of either one or more instances of the registered data store services or one or more entities or one or more classes associated with the one or more identified data store services.

19. The method as set forth in claim 17 wherein providing metadata obtained from the at least one data store repository that describes the one or more registered data store services further comprises enumerating at least one of either corresponding entities from any one of the one or more registered data store services or corresponding entities from at least one of the registered data store services and at least another one of the registered data store services.

20. The method as set forth in claim 17 wherein providing metadata obtained from the at least one data store repository that describes the one or more registered data store services further comprises providing one or more property fields describing one or more entities for which metadata is being requested based upon one or more views identified by the one or more clients.

* * * * *