US 20220405377A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2022/0405377 A1**

KAKUI (43) **Pub. Date: Dec. 22, 2022**

(54) **DATA ACCESS CONTROL METHOD, DATA ACCESS CONTROL APPARATUS, AND DATA ACCESS CONTROL PROGRAM**

(71) Applicant: **Hitachi, Ltd.**, Tokyo (JP)

(72) Inventor: **Kentaro KAKUI**, Tokyo (JP)

(21) Appl. No.: **17/842,897**

(22) Filed: **Jun. 17, 2022**

(30) **Foreign Application Priority Data**

Jun. 21, 2021 (JP) 2021-102711

Publication Classification

(51) **Int. Cl.**

G06F 21/44	(2006.01)
G06F 21/62	(2006.01)
G06F 21/60	(2006.01)
G06F 9/455	(2006.01)
G06F 16/2452	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 21/44*** (2013.01); ***G06F 21/6218*** (2013.01); ***G06F 21/604*** (2013.01); ***G06F 9/455*** (2013.01); ***G06F 16/2452*** (2019.01); *G06F 2221/2141* (2013.01)

(57) **ABSTRACT**

A policy determination unit acquires a rule for a request for accessing data based on a preset access control policy, and selects whether to acquire attribute information about an attribute of each record of the data from the outside of a database in which the data is stored. As a result, when selecting acquisition of the attribute information, the attribute information is acquired and the rule based on the attribute information is evaluated, and when selecting no acquisition of the attribute information, the database is caused to execute filtering of the data based on the rule. Then, based on the evaluation result of the rule or the filtering execution result, a record of the data corresponding to the access request is acquired from the database.

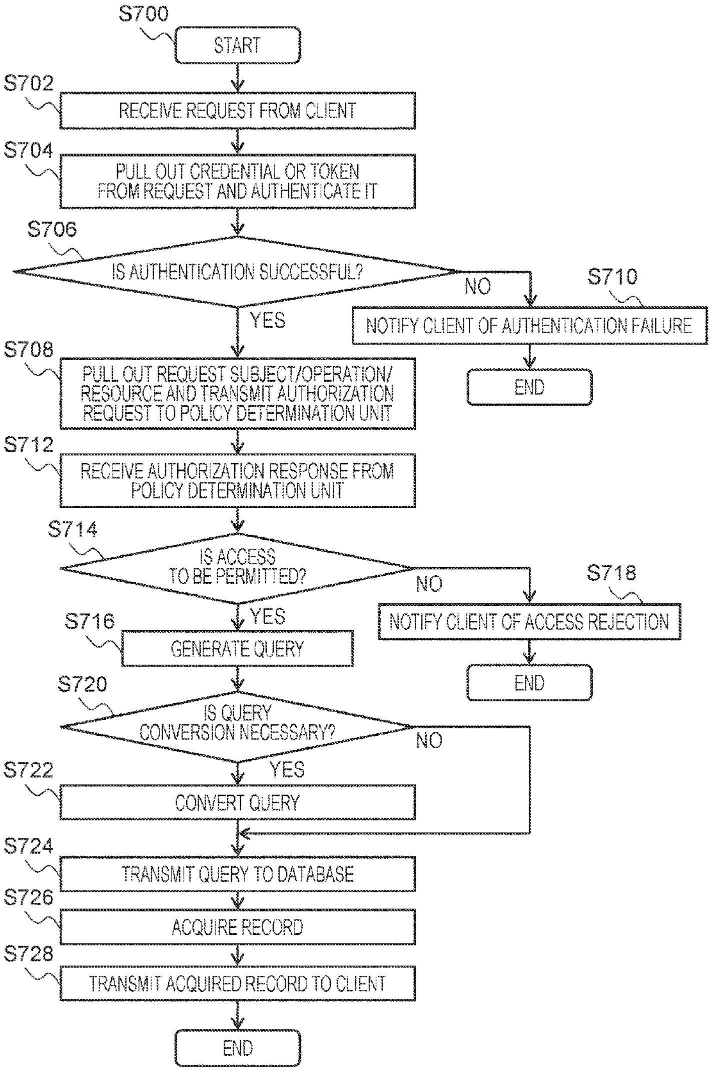

S700
START
S702
RECEIVE REQUEST FROM CLIENT
S704
PULL OUT CREDENTIAL OR TOKEN FROM REQUEST AND AUTHENTICATE IT
S706
IS AUTHENTICATION SUCCESSFUL?
NO
S710
NOTIFY CLIENT OF AUTHENTICATION FAILURE
END
YES
S708
PULL OUT REQUEST SUBJECT/OPERATION/ RESOURCE AND TRANSMIT AUTHORIZATION REQUEST TO POLICY DETERMINATION UNIT
S712
RECEIVE AUTHORIZATION RESPONSE FROM POLICY DETERMINATION UNIT
S714
IS ACCESS TO BE PERMITTED?
NO
S718
NOTIFY CLIENT OF ACCESS REJECTION
END
YES
S716
GENERATE QUERY
S720
IS QUERY CONVERSION NECESSARY?
NO
YES
S722
CONVERT QUERY
S724
TRANSMIT QUERY TO DATABASE
S726
ACQUIRE RECORD
S728
TRANSMIT ACQUIRED RECORD TO CLIENT
END

FIG. 1

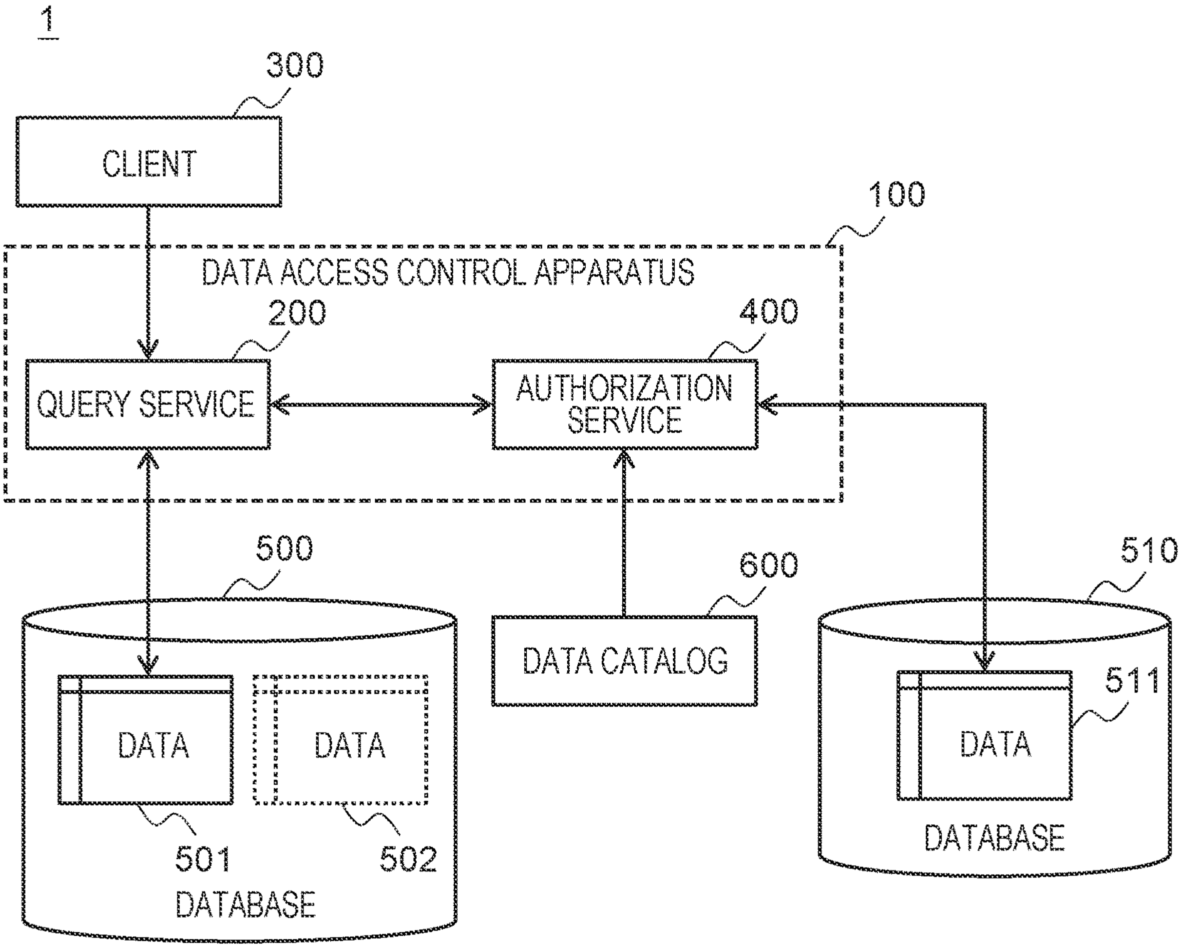

1
300
CLIENT
100
DATA ACCESS CONTROL APPARATUS
200
QUERY SERVICE
400
AUTHORIZATION SERVICE
500
DATA
DATA
501
502
DATABASE
600
DATA CATALOG
510
DATA
511
DATABASE

FIG. 2

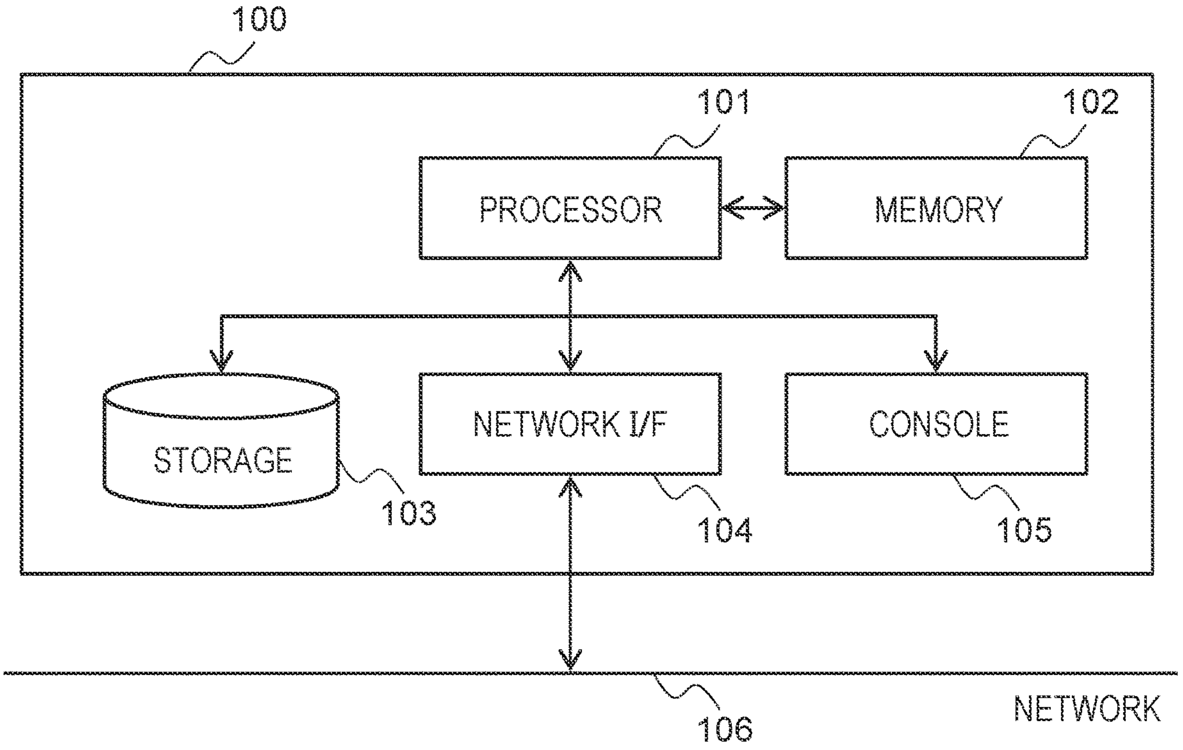

100
101
102
PROCESSOR
MEMORY
STORAGE
NETWORK I/F
CONSOLE
103
104
105
106
NETWORK

FIG. 3

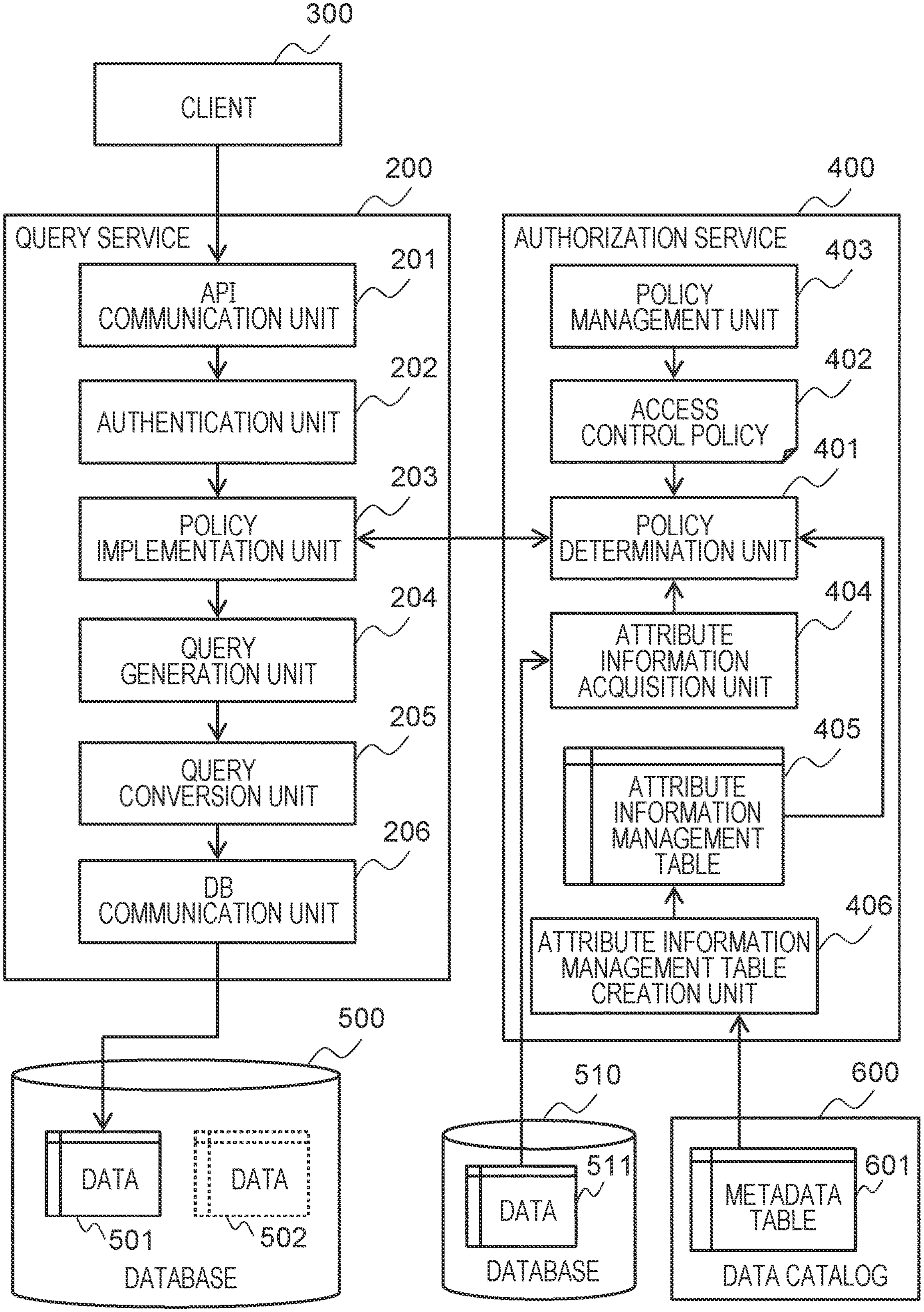

300
CLIENT
200
QUERY SERVICE
201
API COMMUNICATION UNIT
202
AUTHENTICATION UNIT
203
POLICY IMPLEMENTATION UNIT
204
QUERY GENERATION UNIT
205
QUERY CONVERSION UNIT
206
DB COMMUNICATION UNIT
400
AUTHORIZATION SERVICE
403
POLICY MANAGEMENT UNIT
402
ACCESS CONTROL POLICY
401
POLICY DETERMINATION UNIT
404
ATTRIBUTE INFORMATION ACQUISITION UNIT
405
ATTRIBUTE INFORMATION MANAGEMENT TABLE
406
ATTRIBUTE INFORMATION MANAGEMENT TABLE CREATION UNIT
500
DATA
DATA
501
502
DATABASE
510
511
DATA
DATABASE
600
601
METADATA TABLE
DATA CATALOG

FIG. 4

501

ORDER RECEPTION SLIP TABLE: ORDERS

521

SLIP No	ORDERER	ORDER RECEPTION DATE	...
ORD001	CUS001	2020-10-01	
ORD002	CUS002	2020-10-02	
ORD003	CUS001	2020-10-03	
⋮			

502, 511

CUSTOMER TABLE: CUSTOMERS

522

CUSTOMER No	CUSTOMER NAME	PERSON IN CHARGE	...
CUS001	Acme Co.	1001	
CUS002	Beep Inc.	1002	
CUS003	Coyote Ltd.	1001	
⋮			

FIG. 5

601A

METADATA TABLE

id	TYPE	NAME	CAPABILITIES	BELONGS TO	...
D500	DB	DB01	SQL	-	
D501	Table	ORDERS	-	D500	
D502	Table	CUSTOMERS	-	D500	
⋮					
D510	DB	DB02	SQL	-	
D511	Table	CUSTOMERS	-	D510	
⋮					

611

601B

METADATA TABLE

id	TYPE	NAME	CAPABILITIES	BELONGS TO	...
D500	DB	DB01	SQL	-	
D501	Table	ORDERS	-	D500	
⋮					
D510	DB	DB02	SQL	-	
D511	Table	CUSTOMERS	-	D510	
⋮					

FIG. 6

405A

ATTRIBUTE INFORMATION MANAGEMENT TABLE

id	TARGET	NAME	PLACE	CAPABILITIES	ACCESS PATH	...
A501	RESOURCE	ORDER RECEPTION SLIP	DB01	SQL	ORDERS. SLIP No	
A502	SUBJECT	EMPLOYEE	DB01	SQL	CUSTOMERS. PERSON IN CHARGE	
⋮						

411A

405B

ATTRIBUTE INFORMATION MANAGEMENT TABLE

id	TARGET	NAME	PLACE	CAPABILITIES	ACCESS PATH	...
A501	RESOURCE	ORDER RECEPTION SLIP	DB01	SQL	ORDERS. SLIP No	
A511	SUBJECT	EMPLOYEE	SV02	REST	/customers/${id}	
⋮						

411B

FIG. 7

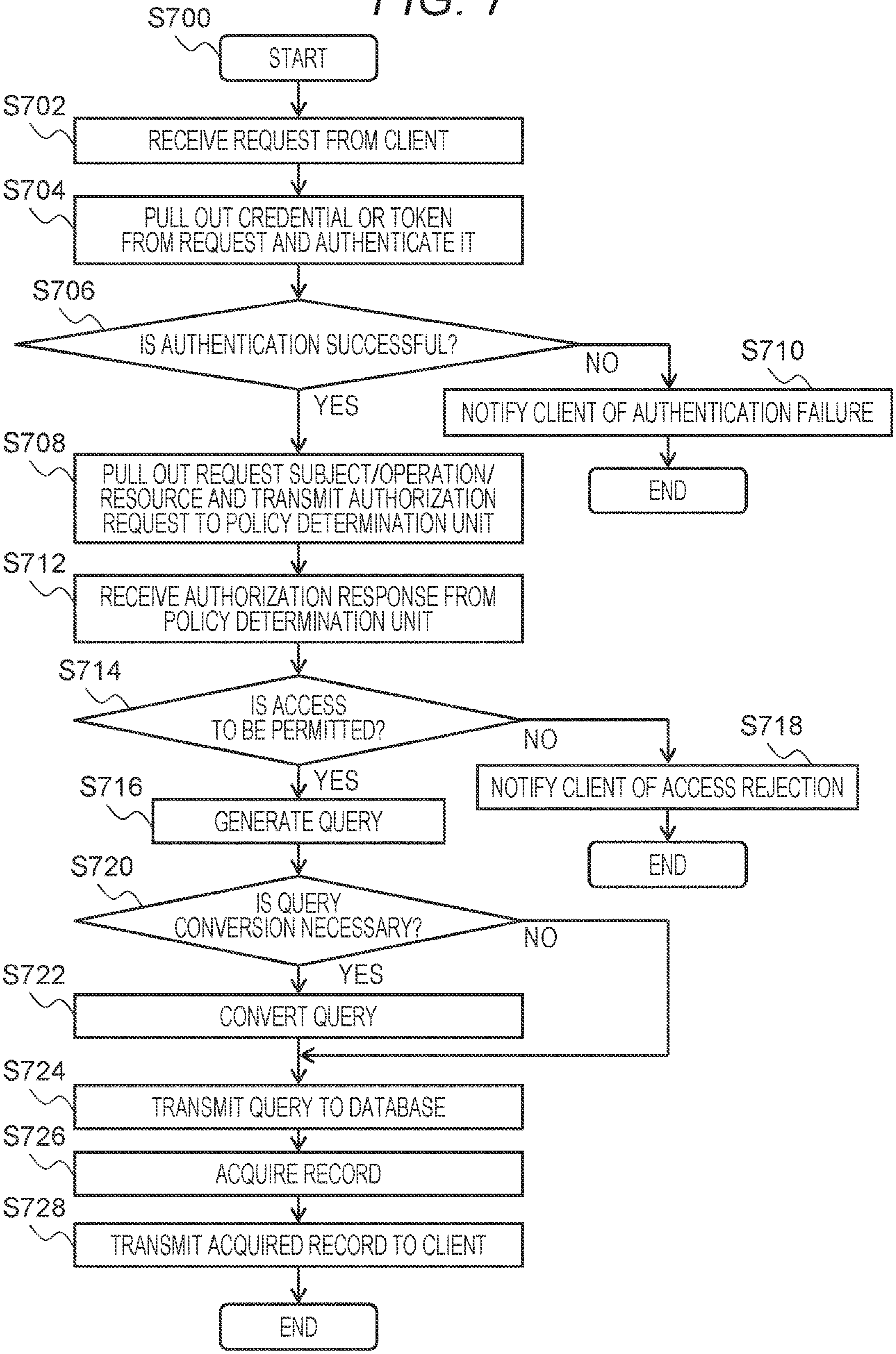

S700
START
S702
RECEIVE REQUEST FROM CLIENT
S704
PULL OUT CREDENTIAL OR TOKEN FROM REQUEST AND AUTHENTICATE IT
S706
IS AUTHENTICATION SUCCESSFUL?
NO
S710
NOTIFY CLIENT OF AUTHENTICATION FAILURE
END
YES
S708
PULL OUT REQUEST SUBJECT/OPERATION/ RESOURCE AND TRANSMIT AUTHORIZATION REQUEST TO POLICY DETERMINATION UNIT
S712
RECEIVE AUTHORIZATION RESPONSE FROM POLICY DETERMINATION UNIT
S714
IS ACCESS TO BE PERMITTED?
NO
S718
NOTIFY CLIENT OF ACCESS REJECTION
END
YES
S716
GENERATE QUERY
S720
IS QUERY CONVERSION NECESSARY?
NO
YES
S722
CONVERT QUERY
S724
TRANSMIT QUERY TO DATABASE
S726
ACQUIRE RECORD
S728
TRANSMIT ACQUIRED RECORD TO CLIENT
END

FIG. 8

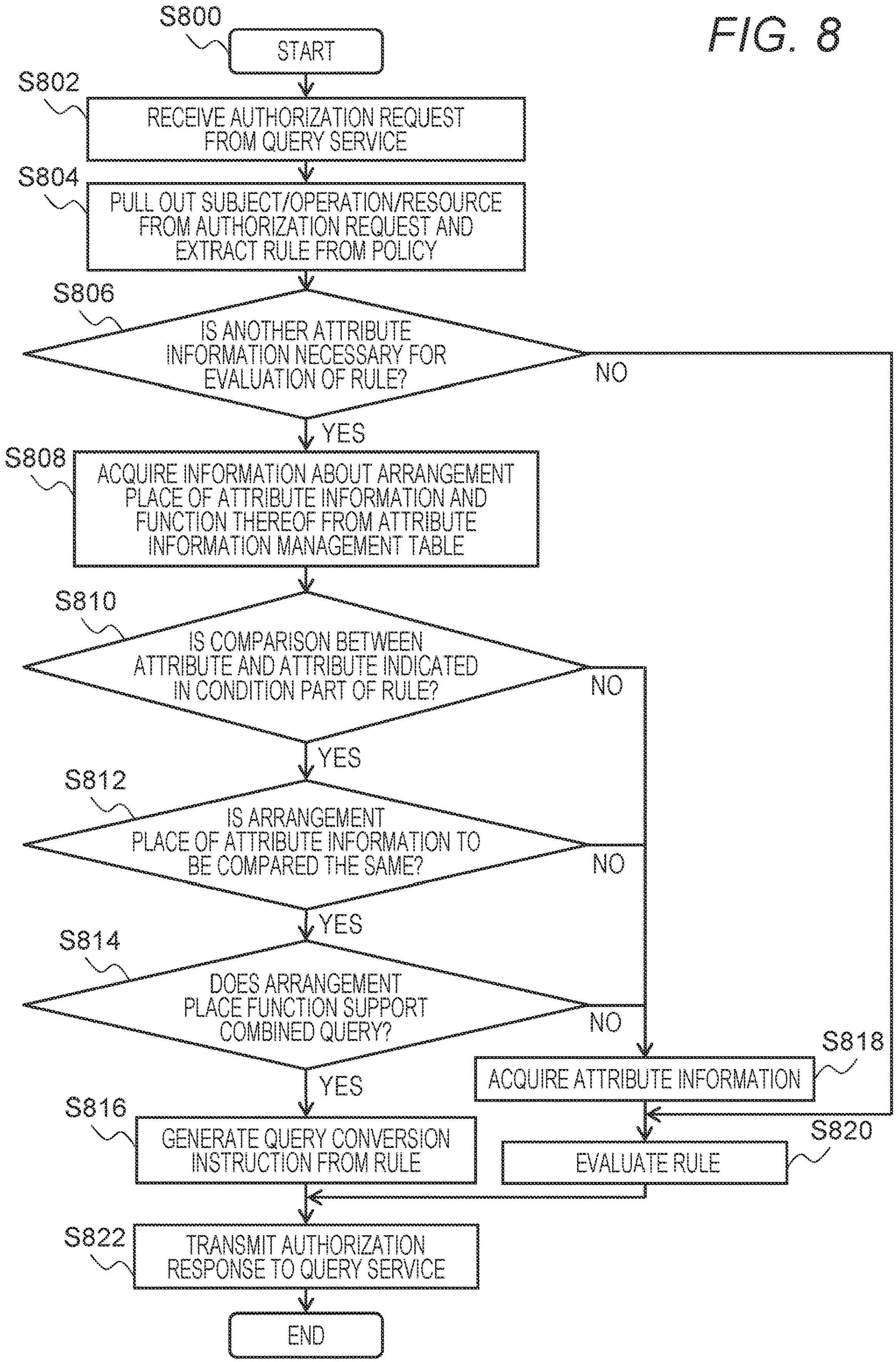

S800
START
S802
RECEIVE AUTHORIZATION REQUEST FROM QUERY SERVICE
S804
PULL OUT SUBJECT/OPERATION/RESOURCE FROM AUTHORIZATION REQUEST AND EXTRACT RULE FROM POLICY
S806
IS ANOTHER ATTRIBUTE INFORMATION NECESSARY FOR EVALUATION OF RULE?
NO
YES
S808
ACQUIRE INFORMATION ABOUT ARRANGEMENT PLACE OF ATTRIBUTE INFORMATION AND FUNCTION THEREOF FROM ATTRIBUTE INFORMATION MANAGEMENT TABLE
S810
IS COMPARISON BETWEEN ATTRIBUTE AND ATTRIBUTE INDICATED IN CONDITION PART OF RULE?
NO
YES
S812
IS ARRANGEMENT PLACE OF ATTRIBUTE INFORMATION TO BE COMPARED THE SAME?
NO
YES
S814
DOES ARRANGEMENT PLACE FUNCTION SUPPORT COMBINED QUERY?
NO
YES
S818
ACQUIRE ATTRIBUTE INFORMATION
S816
GENERATE QUERY CONVERSION INSTRUCTION FROM RULE
S820
EVALUATE RULE
S822
TRANSMIT AUTHORIZATION RESPONSE TO QUERY SERVICE
END

FIG. 9

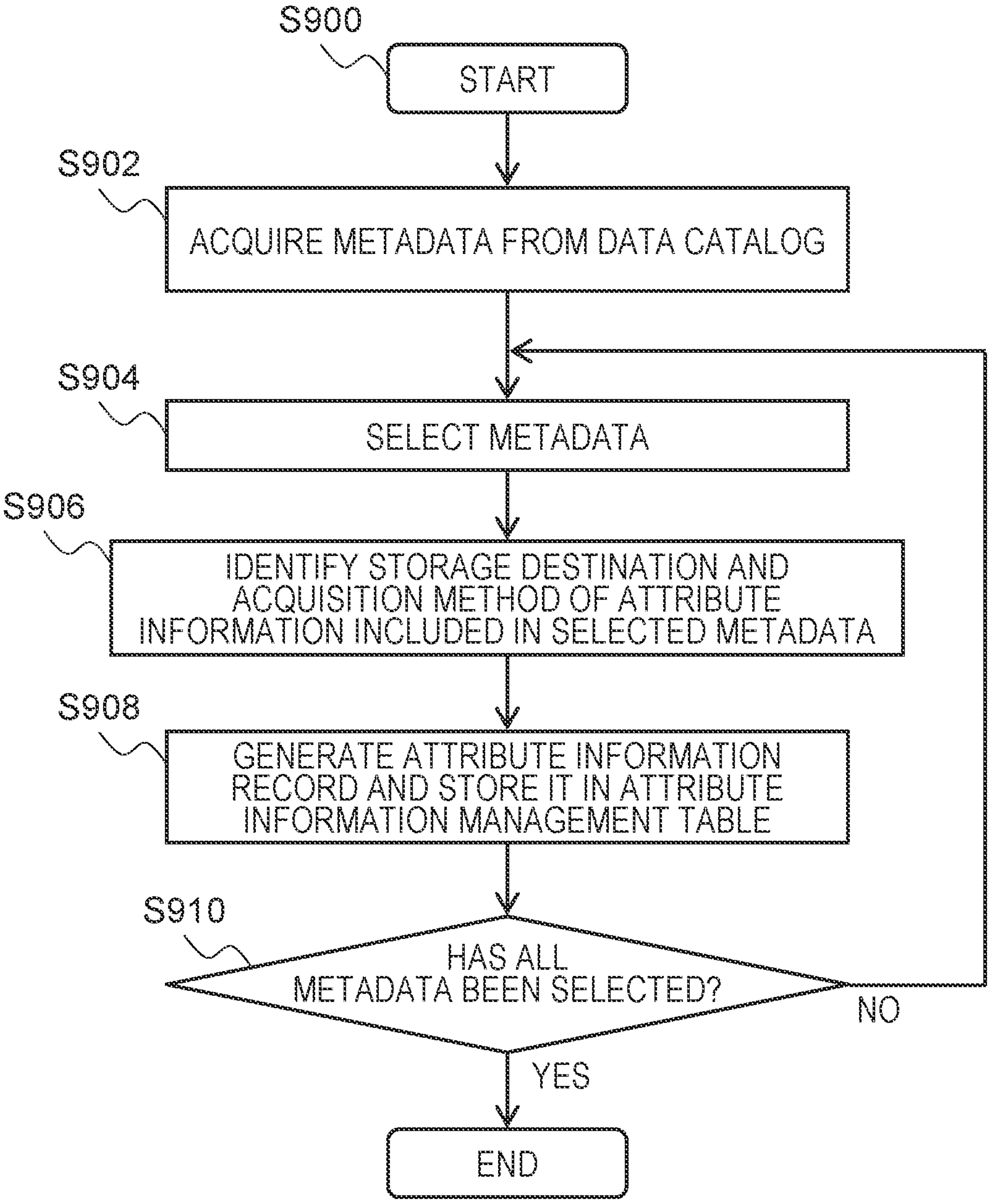

S900
START
S902
ACQUIRE METADATA FROM DATA CATALOG
S904
SELECT METADATA
S906
IDENTIFY STORAGE DESTINATION AND ACQUISITION METHOD OF ATTRIBUTE INFORMATION INCLUDED IN SELECTED METADATA
S908
GENERATE ATTRIBUTE INFORMATION RECORD AND STORE IT IN ATTRIBUTE INFORMATION MANAGEMENT TABLE
S910
HAS ALL METADATA BEEN SELECTED?
NO
YES
END

FIG. 10

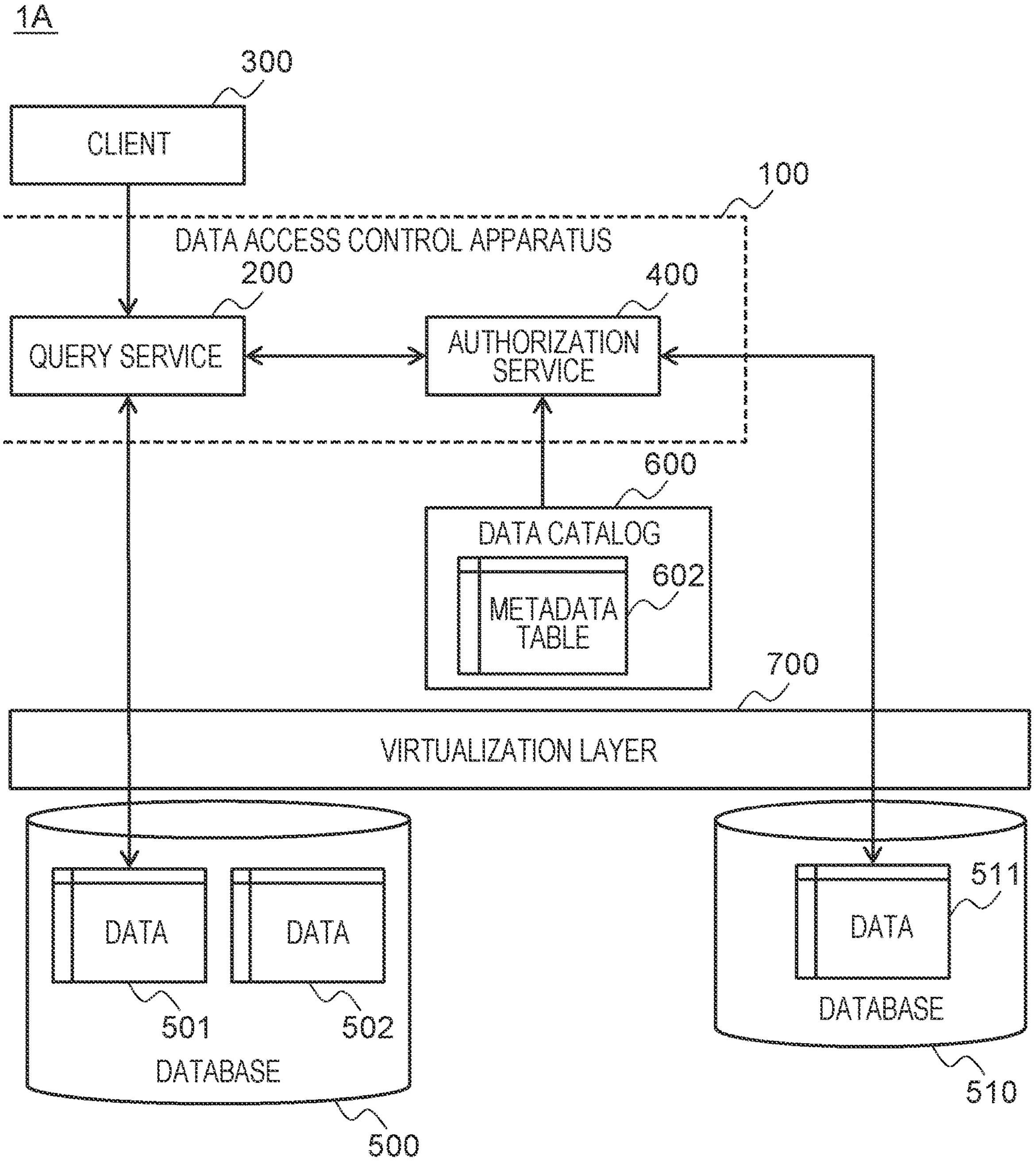
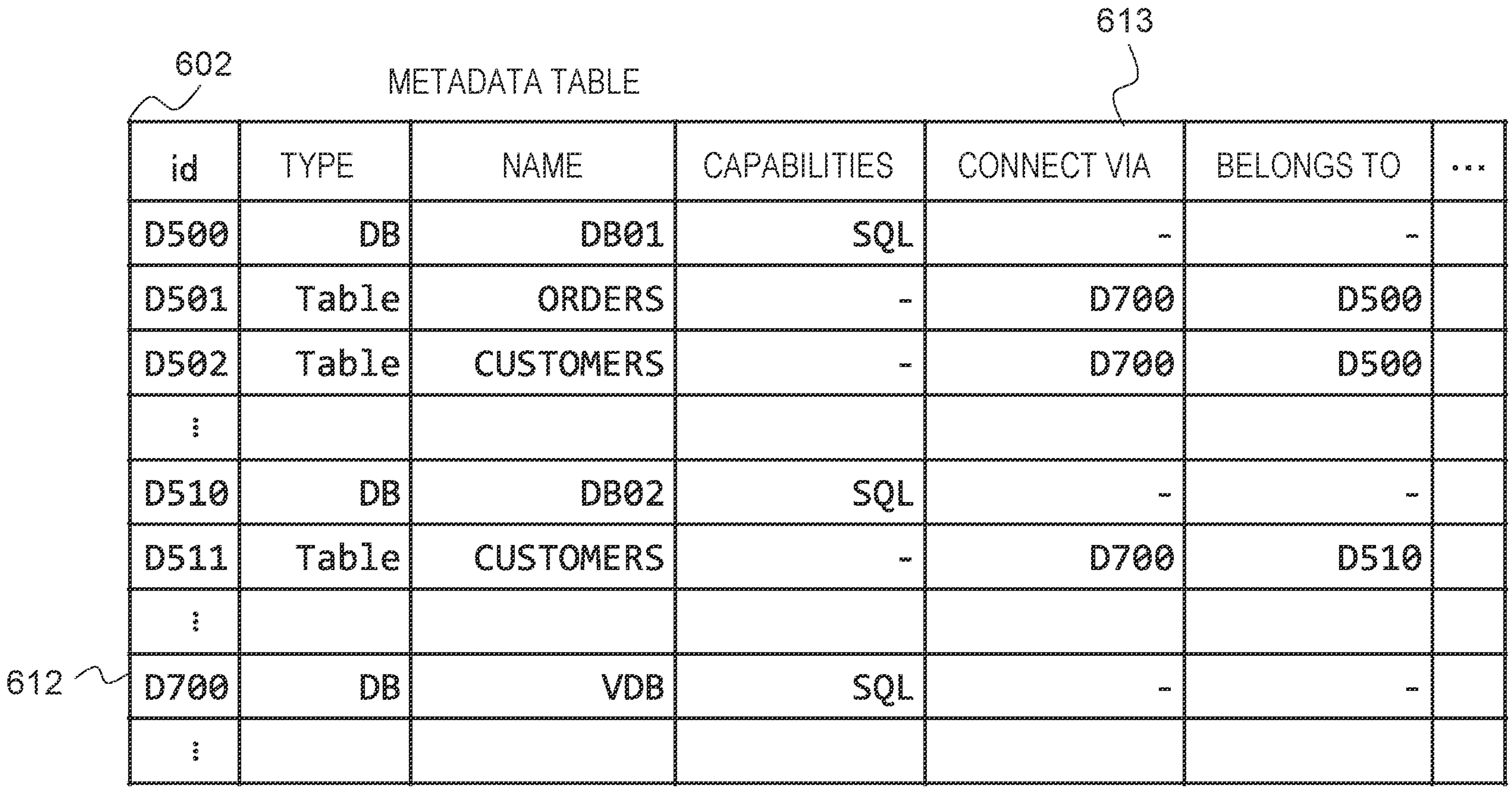

1A
300
CLIENT
100
DATA ACCESS CONTROL APPARATUS
200
QUERY SERVICE
400
AUTHORIZATION SERVICE
600
DATA CATALOG
602
METADATA TABLE
700
VIRTUALIZATION LAYER
DATA
DATA
501
502
DATABASE
500
DATA
511
DATABASE
510

FIG. 11



METADATA TABLE

id	TYPE	NAME	CAPABILITIES	CONNECT VIA	BELONGS TO	...
D500	DB	DB01	SQL	-	-	
D501	Table	ORDERS	-	D700	D500	
D502	Table	CUSTOMERS	-	D700	D500	
⋮						
D510	DB	DB02	SQL	-	-	
D511	Table	CUSTOMERS	-	D700	D510	
⋮						
D700	DB	VDB	SQL	-	-	
⋮						

DATA ACCESS CONTROL METHOD, DATA ACCESS CONTROL APPARATUS, AND DATA ACCESS CONTROL PROGRAM

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority from Japanese application JP2021-102711, filed on Jun. 21, 2021, the contents of which is hereby incorporated by reference into this application.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0002] The present invention relates to a method, an apparatus, and a program for controlling data access.

2. Description of the Related Art

[0003] As a document disclosing the background art of the present invention, US 2020/0358823 A is known. US 2020/0358823 A describes a technique of generating an authentication rule according to a specific context based on a policy defined by an external policy service, and evaluating the authentication rule in an application to determine whether access to resources such as various databases is allowed for each user.

SUMMARY OF THE INVENTION

[0004] In conventional data access control, depending on a structure of a database to be accessed by a user, when a policy for access control is defined, it is necessary to enumerate and describe combinations of a user who is allowed or denied to access and data for each piece of data. In such a case, the description of the policy is enlarged, and the combination of the user and the data to be described in the policy is updated every day according to the update of the database, which causes a problem that it is difficult to maintain and manage the policy. The technique described in US 2020/0358823 A is insufficient in coping with such a problem.

[0005] Therefore, an object of the present invention is to facilitate maintenance management of an access control policy in data access control.

Solution to Problem

[0006] A data access control method according to the present invention is a data access control method performed by a computer, and the method includes acquiring, with the computer, a rule for a request for accessing first data, based on a preset access control policy, selecting, with the computer, whether to acquire attribute information about an attribute of each record of the first data from outside of a first database in which the first data is stored, when selecting acquisition of the attribute information, acquiring the attribute information and evaluating the rule based on the attribute information by the computer, and when selecting no acquisition of the attribute information, instructing, with the computer, the first database to execute filtering of the first data based on the rule, and outputting, with the computer, a record of the first data corresponding to the access request, based on an evaluation result of the rule or an execution result of the filtering.

[0007] A data access control apparatus according to the present invention includes a policy determination unit that acquires a rule for a request for accessing first data, based on a preset access control policy, and selects whether to acquire attribute information about an attribute of each record of the first data from outside of a first database in which the first data is stored, an attribute information acquisition unit that acquires the attribute information, a query unit that causes the first database to execute filtering of the first data based on the rule, and a communication unit that outputs a record of the first data corresponding to the access request, wherein the attribute information acquisition unit acquires the attribute information when the policy determination unit selects acquisition of the attribute information, wherein the policy determination unit evaluates the rule based on the attribute information acquired by the attribute information acquisition unit, wherein the query unit causes the first database to execute the filtering when the policy determination unit selects no acquisition of the attribute information, and wherein the communication unit outputs a record of the first data corresponding to the access request based on an evaluation result of the rule from the policy determination unit or an execution result of the filtering in the first database.

[0008] A data access control program according to the present invention causes a computer to execute a first process of acquiring a rule for a request for accessing first data, based on a preset access control policy, a second process of selecting whether to acquire attribute information about an attribute of each record of the first data from outside of a first database in which the first data is stored, a third process of, when selecting acquisition of the attribute information in the second process, acquiring the attribute information and evaluating the rule based on the attribute information, a fourth process of, when selecting no acquisition of the attribute information in the second process, instructing the first database to execute filtering of the first data based on the rule, and a fifth process of outputting a record of the first data corresponding to the access request based on an evaluation result of the rule by the third process or an execution result of the filtering by the fourth process.

Advantageous Effects of Invention

[0009] According to the present invention, it is possible to provide a data access control technology in which maintenance management of an access control policy is easy.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. **1** is a block diagram illustrating a configuration of a service providing system including a data access control apparatus according to a first embodiment of the present invention;

[0011] FIG. **2** is a diagram illustrating an example of a hardware configuration of a data access control apparatus;

[0012] FIG. **3** is a functional block diagram illustrating functional configurations of a query service and an authorization service;

[0013] FIG. **4** is a diagram illustrating an example of data stored in a database;

[0014] FIG. **5** is a diagram illustrating an example of a metadata table stored in a data catalog according to the first embodiment of the present invention;

[0015] FIG. **6** is a diagram illustrating an example of an attribute information management table created by an attribute information management table creation unit;

[0016] FIG. **7** is a flowchart of a process executed by a query service;

[0017] FIG. **8** is a flowchart of a process executed by an authorization service;

[0018] FIG. **9** is a flowchart of processing executed when an attribute information management table is created in an authorization service;

[0019] FIG. **10** is a block diagram illustrating a configuration of a service providing system including a data access control apparatus according to a second embodiment of the present invention; and

[0020] FIG. **11** is a diagram illustrating an example of a metadata table stored in a data catalog according to the second embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0021] Hereinafter, embodiments of the present invention will be described with reference to the drawings. For clarity of description, the following description and drawings are omitted and simplified as appropriate. The present invention is not limited to the present embodiment, and all application examples consistent with the idea of the present invention are included in the technical scope of the present invention. Unless otherwise specified, each component may be singular or plural.

[0022] In the following description, for example, various types of information may be described by expression of an "xxx list" or an "xxx table", but the various types of information may be expressed by a data structure other than a list or a table. In order to indicate that various types of information do not depend on the data structure, the "xxx list" or the "xxx table" may be referred to as "xxx information".

[0023] In addition, in the following description, reference numerals (or common portions in reference numerals) may be used when the same kind of elements are not distinguished from each other, and an ID of an element (or a reference numeral of an element) may be used when the same kind of elements are distinguished from each other.

[0024] In the following description, a "service providing system" is a system including one or more computers. Therefore, the "service providing system" may be one computer, may be a plurality of computers, or may include a device other than the computer in addition to the computer. The one or more computers typically include at least one physical computer.

[0025] The one or more computers may include at least one virtual computer.

[0026] In addition, in the following description, the "data access control apparatus" and the "client" may include one or more computers.

[0027] In the following description, processing may be described with a "program" or a process thereof as a subject. However, the subject of the processing may be a processor since the program is executed by a processor (for example, a central processing unit (CPU)) to perform predetermined processing appropriately using a storage resource (for example, memory) and/or a communication interface device (for example, a communication port). The processor operates as a function unit that realizes a predetermined function by operating according to the program. An apparatus and a system including a processor are an apparatus and a system each including the function unit.

[0028] In the following description, a "database" means a computer having a physical storage device, and may typically be a computer having a non-volatile storage device (for example, an auxiliary storage device). The auxiliary storage device may be, for example, a hard disk drive (HDD) or a solid state drive (SSD). Different types of storage devices may be mixed in the database.

First Embodiment

[0029] Hereinafter, a first embodiment of the present invention will be described.

[0030] FIG. **1** is a block diagram illustrating a configuration of a service providing system including a data access control apparatus according to a first embodiment of the present invention. A service providing system **1** illustrated in FIG. **1** is a system that provides various services to a user, and includes a data access control apparatus **100**, a client **300**, databases **500** and **510**, and a data catalog **600**. In the service providing system **1**, these components are connected to each other via a network such as the Internet, a local area network (LAN), or a wide area network (WAN).

[0031] The data access control apparatus **100** has a query service **200** and an authorization service **400**. The query service **200** receives a request transmitted from the client **300** when the user operates the client **300** including a personal computer (PC) or the like, and provides a service corresponding to the content of the request. For example, part or all of the data **501** specified in the request is acquired from the database **500**, and is transmitted to the client **300** as a response to the request. Note that the request and the response in the query service **200** are typically defined as a RESTful API.

[0032] The authorization service **400** provides an authorization process for restricting access to the data **501** by the query service **200** for each user, depending on the attribute of the user transmitting the request from the client **300**. The authorization service **400** may perform an authorization process based on the metadata acquired from the data catalog **600** or the data **511** stored in the database **510** and provide a result thereof.

[0033] The database **500** stores the data **501** and the data **502**. The data **501**, **502** is information in which a large number of records including a plurality of preset data items are collectively tabulated, and different data items are set. However, the data **502** is not necessarily stored in the database **500**, and there is a case where the data **502** is not stored in the database **500** due to design or operation of an application for realizing the service providing system **1**. Therefore, in the block diagram of FIG. **1**, the data **502** is indicated by a broken line.

[0034] The database **510** stores the data **511**. As in the data **501**, **502** stored in the database **500**, the data **511** is information in which a large number of records including a plurality of preset data items is collectively tabulated, and is tabulated based on a schema common to the data **502**. Typically, the data **511** is the master and the data **502** is a duplicate thereof. As described above, there is a case where the data **502** is not stored in the database **500**, but even in this case, the data **511** can be acquired from the database **510**

instead of the data **502** and used in the authorization process since the authorization service **400** transmits a request to the database **510**.

[0035] Whether to store the data **502** in the database **500** in the service providing system **1** is determined by, for example, a trade-off between a cost required for ensuring consistency between the data **502** and the data **511** and an overhead in acquiring the data **511** from the database **510**. Which setting state is set may be selected at the time of designing or starting operation of an application for realizing the service providing system **1**, or may be changed during operation.

[0036] The data catalog **600** collects and accumulates metadata that is information about the databases **500**, **510** and the data **501**, **502** and **511** stored therein. The authorization service **400** can determine whether to acquire the data **511** from the database **510** by referring to the metadata acquired from the data catalog **600**.

[0037] FIG. **2** is a diagram illustrating an example of a hardware configuration of the data access control apparatus **100**. As illustrated in FIG. **2**, the data access control apparatus **100** includes, for example, a computer including a processor **101**, a memory **102**, a storage **103**, a network interface (I/F) **104**, and a console **105**.

[0038] The processor **101** executes a predetermined program or application stored in the storage **103** by using the memory **102** as a work area, thereby performing arithmetic processing for realizing the query service **200** and the authorization service **400** in FIG. **1**. The network I/F **104** is connected to the client **300**, the databases **500** and **510**, and the data catalog **600** in FIG. **1** via a network **106**, and performs information communication with them in accordance with control of the processor **101**. The console **105** is an input/output device used by an administrator who manages the service providing system **1**, and includes, for example, a display, a mouse, a keyboard, and the like.

[0039] Note that, in the example of FIG. **2**, a hardware configuration example in a case where the query service **200** and the authorization service **400** are implemented by one data access control apparatus **100** has been illustrated. However, the service may be implemented by separate hardware, or may be implemented by combining a plurality of pieces of hardware. Any hardware configuration can implement the data access control apparatus **100** that provides the query service **200** and the authorization service **400** to the client **300**.

[0040] FIG. **3** is a functional block diagram illustrating a functional configuration of the query service **200** and the authorization service **400**. In FIG. **3**, the query service **200** includes functional blocks of an application programming interface (API) communication unit **201**, an authentication unit **202**, a policy implementation unit **203**, a query generation unit **204**, a query conversion unit **205**, and a DB communication unit **206**. In addition, the authorization service **400** includes functional blocks of a policy determination unit **401**, a policy management unit **403**, an attribute information acquisition unit **404**, and an attribute information management table creation unit **406**. These functional blocks are realized by, for example, arithmetic processing performed by the processor **101** in FIG. **2**.

[0041] In the query service **200**, the API communication unit **201** receives a request transmitted from the client **300**. The authentication unit **202** authenticates the request received by the API communication unit **201** and identifies the user who has made the request, thereby identifying the subject of the request. Based on the result of the authorization process provided from the authorization service **400**, the policy implementation unit **203** permits or denies access to the data **501** according to the subject of the request, and determines whether query conversion is necessary.

[0042] When access to the data **501** is permitted by the policy implementation unit **203**, the query generation unit **204** generates a query for requesting the database **500** to acquire the data **501**. The query conversion unit **205** converts the query generated by the query generation unit **204** in a case where the policy implementation unit **203** determines that query conversion is necessary. The DB communication unit **206** transmits the query generated by the query generation unit **204** and further converted by the query conversion unit **205** to the database **500** as necessary, thereby extracting and acquiring the record of the data **501** according to the request from the database **500**. The record acquired by the DB communication unit **206** is transmitted to the client **300** by the API communication unit **201** as a response to the request.

[0043] In the authorization service **400**, the policy management unit **403** manages an access control policy **402** used for the authorization process executed in the policy determination unit **401**. The access control policy **402** is information indicating a rule for defining, for each subject, which record among a plurality of records included in the data **501** is permitted to be accessed by various users (subjects) who access the data **501** using the client **300**, and for example, one set in advance in a predetermined file format such as an eXtensible access control markup language (XACML) is stored in the data access control apparatus **100**. The policy management unit **403** acquires the access control policy **402** stored and held in the storage **103** of FIG. **2**, for example, to output the access control policy **402** to the policy determination unit **401**.

[0044] The attribute information acquisition unit **404** acquires attribute information about an attribute of each record of the data **511** from the data **501** stored in the database **510** as necessary. When the data **502** common to the data **511** is stored in the database **500**, the attribute information acquisition unit **404** may not acquire the attribute information from the data **511**.

[0045] The attribute information management table creation unit **406** acquires a metadata table **601** stored in the data catalog **600** and creates an attribute information management table **405** based on the metadata table **601**. The metadata table **601** is acquired by tabulating the above-described metadata, and includes information about functions supported by various databases including the databases **500**, **510** and information about data stored in each database. The attribute information management table creation unit **406** can create the attribute information management table **405** indicating a storage destination of the data by extracting and acquiring the information about the data **501**, **502**, and **511** from the metadata table **601**. Note that details of the attribute information management table **405** will be described later.

[0046] The policy determination unit **401** performs an authorization process for a request from the client **300** based on the access control policy **402** output from the policy management unit **403**, the attribute information acquired by the attribute information acquisition unit **404**, and the attribute information management table **405** created by the attribute information management table creation unit **406**. In this authorization process, it is determined whether to permit the access to the data **501** to the request, and which record of the data **501** is permitted to be accessed in a case where the access is permitted. A result of the authorization process by the policy determination unit **401** is transmitted to the policy implementation unit **203** of the query service **200**.

[0047] FIG. **4** is a diagram illustrating an example of the data **501**, **502**, and **511** stored in the databases **500**, **510**. For example, an order reception slip table and a customer table having the data structure illustrated in FIG. **4** are stored as the data **501**, **502** in the database **500**, and a customer table having the same structure as the data **502** is stored as the data **511** in the database **510**. These pieces of data are configured by a plurality of records, and a value of a data item set for each column corresponding to the field is stored in a field of each record. For example, in the order reception slip table of the data **501**, values corresponding to data items such as “slip No.”, “orderer”, and “order reception date” are stored for a plurality of fields constituting each record, and in the customer table of the data **502**, **511**, values corresponding to data items such as “customer No.”, “customer name”, and “person in charge” are stored for a plurality of fields constituting each record. Among these data items, the “orderer” and the “customer No.” have a correspondence relationship, and each record of the data **501** and each record of the data **502**, **511** are associated with each other by the correspondence relationship.

[0048] The attribute information acquisition unit **404** can extract and acquire necessary information from the data **501** as the attribute information about the attribute of each record of the data **502** by the above association. For example, for the record **521** in which the value of “orderer” is “CUS001” in the data **501**, the record **522** in which the value of “customer No.” is the same “CUS001” can be extracted from the data **502**, and the value of each field of the record **522** can be acquired as the attribute information about the record **521**.

[0049] FIG. **5** is a diagram illustrating an example of the metadata table **601** stored in the data catalog **600** in the first embodiment of the present invention. As described above, the data **502** and the data **511** respectively stored in the databases **500**, **510** are tabulated based on a common schema, but there is a case where the data **502** does not exist in the database **500** due to application design or operational reasons. Therefore, the configuration of the metadata table **601** in which the metadata is tabulated differs depending on whether the data **502** exists. In the example of FIG. **5**, an example of the metadata table **601** when the data **502** exists is illustrated as a metadata table **601**A, and an example of the metadata table **601** when the data **502** does not exist is illustrated as a metadata table **601**B.

[0050] The difference between the metadata table **601**A and the metadata table **601**B is the presence or absence of a record **611** indicating the feature of the data **502** and the database to which the data **502** belongs. That is, the metadata table **601**A in which the data **502** exists is configured to include the record **611** representing the metadata of the data **502**, but the metadata table **601**B in which the data **502** does not exist does not include the record **611** representing the metadata of the data **502**.

[0051] FIG. **6** is a diagram illustrating an example of the attribute information management table **405** created by the attribute information management table creation unit **406**. As described with reference to FIG. **5**, the configuration of the metadata table **601** differs depending on whether the data **502** exists. Therefore, the configuration of the attribute information management table **405** created based on the metadata table **601** similarly differs depending on whether the data **502** exists. In the example of FIG. **6**, an example of the attribute information management table **405** when the data **502** exists is illustrated as an attribute information management table **405**A, and an example of the attribute information management table **405** when the data **502** does not exist is illustrated as an attribute information management table **405**B.

[0052] In the attribute information management table **405**A and the attribute information management table **405**B, contents of records **411**A and **411**B respectively corresponding to the data **502** and the data **511** are different. That is, in the attribute information management table **405**A in which the data **502** exists, the storage destination and the acquisition method of the attribute information based on the data **502** are recorded in a record **411**A, and in the attribute information management table **405**B in which the data **502** does not exist, the storage destination and the acquisition method of the attribute information based on the data **511** are recorded in a record **411**B.

[0053] Specifically, in the attribute information management table **405**A in FIG. **6**, “DB01” and “SQL” stored in “arrangement place” and “function”, respectively, of the record **411**A indicate that the data **502** including the attribute information is stored in the database **500**, and the attribute information can be acquired by an SQL instruction. Furthermore, in the attribute information management table **405**B, “SV02” and “REST” stored in “arrangement place” and “function”, respectively, of the record **411**B indicate that the attribute information can be acquired by a service defined by a RESTful API.

[0054] Further, in the attribute information management table **405**A, **405**B, a descriptor for uniquely identifying an acquisition destination of the attribute information in the databases **500**, **510** is stored in the “access path” of each record. Specifically, in each record of the attribute information management table **405**A and each record, except for the record **411**B, of the attribute information management table **405**B, a concatenation of the table name and the column name of the data **501**, **502** is stored in the “access path”, and in the record **411**B of the attribute information management table **405**B, the path parameter specified in the request of the RESTful API is stored in the “access path”.

[0055] The attribute information management table **405**A is generated, for example, when the administrator of the service providing system **1** refers to the metadata table **601**A and selects the data **502** as the attribute information about the access subject. On the other hand, the attribute information management table **405**B is generated, for example, when the administrator of the service providing system **1** refers to the metadata table **601**B and selects the data **511** as the attribute information about the access subject.

[0056] Next, details of processing performed by the data access control apparatus **100** will be described below with reference to a flowchart.

[0057] FIG. **7** is a flowchart of processing executed by the query service **200** of the data access control apparatus **100**. When a request instructing access to the data **501** is transmitted from the client **300**, the query service **200** starts the processing illustrated in the flowchart of FIG. (step **S700**).

[0058] The API communication unit **201** receives a request from the client **300** (step S**702**). The request includes information indicating which record among the plurality of records constituting the data **501** is to be acquired. For example, a request for acquiring a record whose value of "slip No." is "ORD001" in the data **501** is described as "GET/orders/ORD001 HTTP/1.1".

[0059] The authentication unit **202** pulls out the credential or the token included in the request received by the API communication unit **201** in step S**702**, and authenticates the request using the pulled-out credential or token (step S**704**). For example, when the received request is an API based on HTTP, a credential or a token is typically included in a header field of the request. When the request is successfully authenticated in step S**704**, the query service **200** can identify the user (subject) who has transmitted the request.

[0060] For the authentication performed in step S**704**, for example, Basic authentication, JWT authentication, or the like can be used. As a specific authentication execution method, for example, Basic authentication is to check a password included in a credential, and JWT authentication is to verify a signature of a token. As described above, the processing content of step S**704** may be different depending on the authentication method, but in any case, the authentication is successful if the subject of the request can be identified. When the subject of the request can be identified in step S**704**, it is determined that the authentication is successful (step S**706**: Yes), and the process proceeds to step S**708**. On the other hand, when the subject of the request cannot be identified in step S**704**, it is determined that the authentication has failed (step S**706**: No), and the process proceeds to step S**710**.

[0061] When the authentication succeeds, the policy implementation unit **203** transmits the information on the subject identified by the authentication unit **202** in step S**704** and the information on the operation and resource pulled out from the received request to the policy determination unit **401** as an authorization request (step S**708**). On the other hand, when the authentication fails, the API communication unit **201** notifies the client **300** that the authentication fails (step S**710**), and ends the processing illustrated in the flowchart of FIG. **7**.

[0062] When receiving the authorization request transmitted from the policy implementation unit **203** in step S**708**, the policy determination unit **401** performs an authorization process to transmit a result thereof to the policy implementation unit **203** as an authorization response. The policy implementation unit **203** receives the authorization response transmitted from the policy determination unit **401** (step S**712**).

[0063] Based on the content of the authorization response received from the policy determination unit **401** in step S**712**, the policy implementation unit **203** determines whether to permit access to the data **501** with respect to the request the API communication unit **201** receives from the client **300** in step S**702** (step S**714**). When the policy determination unit **401** determines to permit access to the data **501** and an authorization response indicating the determination result is received in step S**714**, it is determined that access is permitted (step S**714**: Yes), and the process proceeds to step S**716**. On the other hand, when the policy determination unit **401** determines to deny access to the data **501** and the authorization response indicating the determination result is received in step S**714**, it is determined that the access is denied (step S**714**: No), and the process proceeds to step S**718**.

[0064] When an authorization response indicating that access to the data **501** is permitted is received from the policy determination unit **401**, the query generation unit **204** generates a query for acquiring a desired record from the data **501** stored in the database **500** with respect to the request received by the API communication unit **201** from the client **300** in step S**702** (step S**716**). For example, when the database **500** is a database using SQL as a query language, and a request for acquiring a record whose value of "slip No." is "ORD001" in the data **501** is transmitted from the client **300**, a query instructing this to the database **500** is described as "select * from orders where slip No='ORD001';". As a result, the query service **200** generates a query according to the access request from the client **300** to the data **501**. On the other hand, when the policy determination unit **401** receives an authorization response indicating that access to the data **501** is denied, the API communication unit **201** notifies the client **300** that access to the data **501** is denied (step S**718**), and ends the processing illustrated in the flowchart of FIG. **7**.

[0065] When the query is generated by the query generation unit **204** in step S**716**, the query conversion unit **205** determines whether query conversion is necessary (step S**720**). Here, the policy implementation unit **203** determines the presence or absence of query conversion based on the authorization response received from the policy determination unit **401** in step S**712**. That is, when the received authorization response includes the query conversion instruction, the query conversion unit **205** determines that the query conversion is necessary (step S**720**: Yes), and executes the processing of step S**722**. On the other hand, when the received authorization response does not include the query conversion instruction, the query conversion unit **205** determines that the query conversion is unnecessary (step S**720**: No), and the process proceeds to step S**724** without executing the process of step S**722**.

[0066] When it is determined that the query conversion is necessary, the query conversion unit **205** converts the query generated by the query generation unit **204** in step S**716** into a query for combining and filtering the data **501** and the data **502** stored in the database **500** (step S**722**). For example, in a case where the value of the "person in charge" corresponding to the user (subject) identified by the authentication in step S**704** is "1001", the query acquired by converting the query by SQL described in the above example is described as "select * from orders inner join customers on (orders. orderer=customers. customer No) where customers. person in charge='1001' and orders. slip No='ORD001';". As a result, in the query service **200**, the data **501** to be accessed and the data **502** including the attribute information about the data **501** are combined from the query generated in response to the request in step S**716**, and a query for causing the database **500** to execute filtering based on the rule described in the access control policy **402** is generated.

[0067] The DB communication unit **206** transmits, to the database **500**, the query generated by the query generation unit **204** in step S**716** (when the processing in step S**722** is not executed) or the query after conversion generated by the query conversion unit **205** converting the query generated by the query generation unit **204** in step S**722** (step S**724**).

[0068] The DB communication unit **206** acquires a record of data **501** output by the database **500** in response to the query transmitted in step S**724** (step S**726**). The API communication unit **201** outputs a record of the data **501** acquired by the DB communication unit **206** in step S**726** as a response to the request received in step S**702** to transmit the record to the client **300** (step S**728**). When the API communication unit **201** transmits the response to the request to the client **300**, the query service **200** ends the processing flow of FIG. **7**.

[0069] FIG. **8** is a flowchart of processing executed by the authorization service **400** of the data access control apparatus **100**. When the authorization request is transmitted from the policy implementation unit **203** in step S**708** of FIG. **7**, the authorization service **400** starts the processing illustrated in the flowchart of FIG. **8** (step S**800**).

[0070] The policy determination unit **401** receives an authorization request from the policy implementation unit **203** of the query service **200** (step S**802**). As described above, the authorization request includes information about subject, operation, and resource in the request received by the query service **200** from the client **300**.

[0071] The policy determination unit **401** pulls out each piece of information about a subject, an operation, and a resource from the authorization request received in step S**802**. In addition, by acquiring the access control policy **402** from the policy management unit **403** and pulling out a rule defined in the access control policy **402**, a rule for a request for accessing the data **501** performed from the client **300** is acquired (step S**804**).

[0072] The policy determination unit **401** compares each piece of information about the subject, the operation, and the resource pulled out from the authorization request in step S**804** with the rule extracted in step S**804**, and determines whether attribute information different from the attribute information included in the resource is required for evaluation of the rule for the operation (step S**806**). Here, for example, it is determined whether another piece of attribute information is necessary as follows.

[0073] As described above, it is assumed that a request for acquiring a record whose value of "slip No." is "ORD001" in the data **501** is transmitted from the client **300**, and an authorization request for the request is received from the query service **200**. In addition, it is assumed that a rule defined in the access control policy **402** includes matching between a value of "orderer" of the record specified in the request and a value of "customer No." corresponding to a person in charge number assigned to the user who has transmitted the request through the client **300**. In such a case, in order to evaluate the rule, it is necessary to determine the attribute of the user who has transmitted the request using a value of "person in charge" that is attribute information included in data **502**, **511** different from data **501** that is a resource. Therefore, in such a case, it is determined in step S**806** that another piece of attribute information is necessary. On the other hand, when comparison of records between different data is not necessary in evaluation of the rule, it is determined in step S**806** that the different attribute information is unnecessary, and the process proceeds to step S**820**. Note that the above example is merely an example, and it is possible to determine in step S**806** whether attribute information different from the attribute information included in the resource is necessary for evaluation of various rules.

[0074] As a result of the determination in step S**806**, when it is determined that another attribute information is necessary (step S**806**: Yes), the process proceeds to step S**808**, and when it is determined that the another attribute information is unnecessary (step S**806**: No), the process proceeds to step S**820**.

[0075] When it is determined in step S**806** that another attribute information is necessary, the policy determination unit **401** refers to the attribute information management table **405** created by the attribute information management table creation unit **406** to acquire information on the arrangement place of the attribute information and the function thereof from the attribute information management table **405** (step S**808**). For example, in the above-described example, by referring to the record **411**A of the attribute information management table **405**A or the record **411**B of the attribute information management table **405**B, it is possible to acquire the information indicating the data **502** stored in the database **500** or the information indicating the data **511** stored in the database **510** as the arrangement place of the attribute information necessary for the evaluation of the rule. Furthermore, as the function of the arrangement place of the attribute information, information about the function supported by the database **500** or information about the function supported by the database **510** can be acquired.

[0076] The policy determination unit **401** determines whether the content indicated in the condition part of the rule extracted in step S**804** is a comparison between an attribute and an attribute (step S**810**). For example, as described above, in the case of a rule that the value of "orderer" of the record specified in the request matches the value of "customer No." corresponding to the person in charge number assigned to the user who has transmitted the request by the client **300** match, the condition part of the rule describes a remark in which the attribute indicated by the value of "orderer" of the data **501** is compared with the attribute indicated by the value of "customer No." of the data **502**, **511**. Therefore, in this case, it is determined that the content indicated in the condition part of the rule is the comparison between the attribute and the attribute (step S**810**: Yes), and the process proceeds to step S**812**. On the other hand, when it is determined that the content indicated in the condition part of the rule is not the comparison between the attribute and the attribute (step S**810**: No), the process proceeds to step S**818**.

[0077] The policy determination unit **401** determines whether the arrangement place of the attribute information to be compared in the evaluation of the rule is the same, based on the information about the arrangement place acquired from the attribute information management table **405** in step S**808** (step S**812**). For example, as described above, in a case where the information that the arrangement place of the attribute information necessary for the evaluation of the rule is in the database **500** that is served as the storage destination of the data **501** is acquired by referring to the record **411**A of the attribute information management table **405**A, it is determined that the arrangement place is the same (step S**812**: Yes), and the process proceeds to step S**814**. On the other hand, when it is determined that the arrangement place is not the same (step S**812**: No), the process proceeds to step S**818**.

[0078] The policy determination unit **401** determines whether the function of the arrangement place of the attribute information to be compared in the evaluation of the rule supports a combined query of combining a plurality of pieces of data tables, based on the information about the function of the arrangement place acquired from the attribute information management table **405** in step S**808** (step S**814**). For example, as described above, in a case where the database **500**, which is the arrangement place of the attribute information necessary for the evaluation of the rule, acquires the information that the operation is performed by the SQL that can use the combined query by referring to the record **411**A of the attribute information management table **405**A, it is determined that the combined query is supported (step S**814**: Yes), and the process proceeds to step S**816**. On the other hand, when it is determined that the combined query is not supported (step S**814**: No), the process proceeds to step S**818**.

[0079] When all the determination results of steps S**810**, S**812**, and S**814** are positive determinations, the policy determination unit **401** generates a query conversion instruction for the query service **200** from the rule extracted in step S**804** (step S**816**). Here, the query conversion instruction is generated so as to convert the query generated by the query generation unit **204** in step S**716** of FIG. **7** into the query as described above including the combined query according to the rule. As a result, the query service **200** is instructed to combine the data **501** and the data **502** and cause the database **500** to execute filtering based on the rule. When the query conversion instruction is generated in step S**816**, the process proceeds to step S**822**.

[0080] When the determination result of any one of steps S**810**, S**812**, and S**814** is a negative determination, the policy determination unit **401** acquires attribute information necessary for evaluation of the rule based on the information acquired in step S**808** (step S**818**). Here, for example, the data **511** is read from the database **510** and the attribute information is acquired based on the information acquired by referring to the record **411**B of the attribute information management table **405**B.

[0081] When step S**818** is executed or when the determination result of step S**806** is a negative determination, the policy determination unit **401** evaluates the rule extracted in step S**804** (step S**820**). Here, when step S**818** is executed, the rule is evaluated using the attribute information acquired in step S**818** together with each piece of information about the subject, the operation, and the resource acquired from the authorization request in step S**804**. On the other hand, when it is determined in step S**806** that another attribute information is unnecessary, the rule is evaluated based on each piece of information about the subject, the operation, and the resource acquired from the authorization request in step S**804**. As a result, it is determined whether to permit access to the data **501** for the authorization request received from the query service **200**. When the rule can be evaluated in step S**820**, the process proceeds to step S**822**.

[0082] The policy determination unit **401** transmits an authorization response to the authorization request received in step S**802** to the query service **200** (step S**822**). The authorization response includes either the query conversion instruction generated in step S**816** or the evaluation result of the rule performed in step S**820**. When the policy determination unit **401** transmits the authorization response, the authorization service **400** ends the processing flow of FIG. **8**.

[0083] Note that, in the query service **200**, the policy implementation unit **203** receives the authorization response transmitted from the policy determination unit **401** in step S**822** of FIG. **8** in step S**712** of FIG. **7**. As a result, the query service **200** can proceed to processing of step S**714** and subsequent steps.

[0084] FIG. **9** is a flowchart of processing executed when the attribute information management table creation unit **406** creates the attribute information management table **405** in the authorization service **400** of the data access control apparatus **100**. The attribute information management table creation unit **406** starts the processing illustrated in the flowchart of FIG. **9** at a predetermined timing before the processing of FIG. **8** is executed in the authorization service **400** (step S**900**).

[0085] The attribute information management table creation unit **406** acquires the metadata from the data catalog **600** (step S**902**). Here, metadata is acquired by reading the metadata table **601**A or the metadata table **601**B illustrated in FIG. **5** from the data catalog **600**.

[0086] The attribute information management table creation unit **406** selects any of the pieces of metadata acquired in step S**902** (step S**904**). Here, metadata is selected by selecting any record in the metadata table **601**A, **601**B.

[0087] The attribute information management table creation unit **406** identifies a storage destination and an acquisition method of the attribute information included in the metadata selected in step S**904** (step S**906**). Here, for example, the storage destination and the acquisition method of the attribute information can be identified by identifying the identification name of the data including the attribute information indicated by the metadata, the database in which the data is stored, the function of the database, and the like from the content of each item included in the selected record. Alternatively, the storage destination and the acquisition method of the attribute information may be identified from the information input by the administrator of the service providing system **1**, or the identifying methods may be combined. In addition to this, the storage destination and the acquisition method of the attribute information can be identified by any method.

[0088] The attribute information management table creation unit **406** creates an attribute information record that is a record representing information about the attribute information based on the storage destination and the acquisition method of the attribute information acquired in step S**906** to store the attribute information record in the attribute information management table **405**A or the attribute information management table **405**B (step S**908**).

[0089] When the attribute information record is stored in the attribute information management table **405**A, **405**B in step S**908**, the attribute information management table creation unit **406** determines whether all the pieces of metadata have been selected in step S**904** (step S**910**). When all the records have been selected in the metadata table **601**A, **601**B, it is determined that all the metadata has been selected (step S**910**: Yes), and the processing flow of FIG. **9** ends. On the other hand, when there is an unselected record in the metadata table **601**A, **601**B, it is determined that not all the pieces of metadata have been selected (step S**910**: No), and the process returns to step S**904** to continue the selection of the metadata.

[0090] The attribute information management table creation unit **406** can generate the attribute information management table **405**A from the metadata table **601**A or the attribute information management table **405**B from the metadata table **601**B by executing the processing described above.

[0091] According to the first embodiment of the present invention described above, the following operational effects are achieved.

[0092] (1) The data access control apparatus **100** acquires a rule for a request for accessing the data **501** based on the preset access control policy **402** (step S**804**), and selects whether to acquire attribute information about an attribute of each record of the data **501** from the outside of the database **500** in which the data **501** is stored (steps S**810** to S**814**). As a result, when selecting acquisition of the attribute information (step S**810**: No, step S**812**: No, or step S**814**: No), the attribute information is acquired (step S**818**), and the rule based on the attribute information is evaluated (step S**820**). When selecting not acquisition of the attribute information (Steps S**810**, S**812**, and S**814**: Yes), the database **500** is caused to execute filtering of the data **501** based on the rule (Steps S**816**, S**722**, and S**724**). Then, a record of the data **501** corresponding to the access request is acquired from the database **500** based on the evaluation result of the rule performed in step S**820** or the execution result of the filtering by the database **500** (step S**726**). With this configuration, it is possible to reliably evaluate the rule regardless of whether information necessary for evaluation of the rule defined in the access control policy **402** is stored in the database **500**. Therefore, it is possible to provide a data access control technology that facilitates maintenance and management of an access control policy.

[0093] (2) The data access control apparatus **100** selects no acquisition of the attribute information from the outside of the database **500** when the data **502** including the attribute information is stored in the database **500** (step S**812**: Yes), and selects acquisition of the attribute information from the outside of the database **500** when the data **502** is not stored in the database **500** (step S**812**: No). Thus, it is possible to appropriately select whether to acquire the attribute information from the outside of the database **500** according to the presence or absence of the data **502**.

[0094] (3) When selecting no acquisition of the attribute information from the outside of the database **500**, the data access control apparatus **100** combines the data **501** and the data **502** to cause the database **500** to execute filtering (steps S**816**, S**722**, and S**724**). In this way, since it is not necessary to evaluate the rule in the data access control apparatus **100**, it is possible to reduce the overhead at the time of accessing the data **501** and to realize high-speed access control.

[0095] (4) However, even when the data **502** is stored in the database **500**, in a case where the data **501** and the data **502** cannot be combined in the database **500** (step S**814**: No), it selects acquisition of the attribute information from the outside of the database **500**. With this configuration, even when the database **500** does not have a function of combining the plurality of data tables, it is possible to reliably evaluate the rule.

[0096] (5) The data access control apparatus **100** provides, by a processor **101** that is, for example, a computer, the query service **200** that generates a query according to an access request, and the authorization service **400** that performs the authorization process based on the access control policy **402** for the access request. In the authorization service **400**, the policy determination unit **401** selects whether to acquire the attribute information from the outside of the database **500** (steps S**810** to S**814**). Then, in a case of selecting acquisition of the attribute information, the acquisition of the attribute information and the evaluation of the rule are performed (steps S**818** and S**820**), and in a case of selecting no acquisition of the attribute information, a query for causing the database **500** to execute filtering is output from the query service **200** (steps S**816**, S**722**, and S**724**). With this configuration, the data access control apparatus **100** can be implemented using a computer.

[0097] (6) In the query service **200**, the query conversion unit **205** converts the query generated by the query generation unit **204** into a query for causing the database **500** to execute filtering in accordance with an instruction from the authorization service **400** (step S**722**). Then, the DB communication unit **206** outputs the data to the database **500** (step S**724**). With this configuration, the filtering of the data **501** based on the rule can be reliably executed by the database **500**.

[0098] (7) The data access control apparatus **100** refers to the attribute information management table **405** indicating the arrangement place and the function of the attribute information (step S**808**), and selects whether to acquire the attribute information from the outside of the database **500**. With this configuration, it is possible to reliably and easily select whether to acquire the attribute information from the outside of the database **500**.

[0099] (8) The data access control apparatus **100** acquires the metadata regarding the database **500** storing the data **501** and the databases **500**, **510** storing the data **502**, **511** including the attribute information from the metadata table **601**A, **601**B by the attribute information management table creation unit **406** (steps S**902** and S**904**). Then, the attribute information management table **405**A, **405**B is created based on the acquired metadata (steps S**906** and S**908**). With this configuration, it is possible to reliably and easily create the attribute information management table indicating the arrangement place of the attribute information and the function thereof.

[0100] (9) The metadata used to create the attribute information management table **405**A, **405**B includes information on a function supported by each of the database **500** and the database **510**, and information on the database to which the data **501** and the data **511** belong, as exemplified in each record of the metadata table **601**A, **601**B of FIG. **5**. In this way, information necessary for creating the attribute information management table **405**A, **405**B can be reliably acquired from the metadata.

Second Embodiment

[0101] Next, a second embodiment of the present invention will be described. Hereinafter, differences from the first embodiment will be mainly described, and description of common points with the first embodiment will be omitted or simplified unless otherwise necessary.

[0102] In the present embodiment, an example in which the database **500** and the database **510** are connected to the data access control apparatus **100** via the virtualization layer will be described.

[0103] FIG. **10** is a block diagram illustrating a configuration of a service providing system including a data access control apparatus according to the second embodiment of the present invention. A service providing system **1**A illustrated in FIG. **10** includes a data access control apparatus **100**, a client **300**, databases **500** and **510**, and a data catalog

600, as in the service providing system **1** of FIG. **1** described in the first embodiment. The present embodiment is different from the first embodiment in that the databases **500**, **510** are connected to the data access control apparatus **100** via a virtualization layer **700**, and a metadata table **602** is stored in a data catalog **600**. The virtualization layer **700** provides an interface through which the data access control apparatus **100** can issue a query to the databases **500**, **510** as if the data **501**, **502**, and **511** stored in separate databases **500**, **510** were stored in a single database.

[0104] FIG. **11** is a diagram illustrating an example of the metadata table **602** stored in the data catalog **600** in the second embodiment of the present invention. In the metadata table **602** illustrated in FIG. **11**, a record **612** indicating information about the virtualization layer **700** is added unlike the metadata table **601**A, **601**B of FIG. **5** described in the first embodiment. Further, in the record of the data **501**, **502**, and **511** among the records of the metadata table **602**, information indicating that the connection destination is the virtualization layer **700** is added to a column **613**.

[0105] In the present embodiment, the attribute information management table creation unit **406** creates the attribute information management table **405** based on the metadata table **602**. At this time, the attribute information management table **405** is created so as to include the information about the virtualization layer **700** in the record indicating the storage destination and the acquisition method of the attribute information based on the data **511**.

[0106] Furthermore, in the present embodiment, the query service **200** converts the query generated by the query generation unit **204** in step S**716** into a query for causing the database **500** to execute filtering based on the rule described in the access control policy **402** by combining the data **501** and the data **511** via the virtualization layer **700** in step S**722** of FIG. **7**. As a result, even when the data **501** and the data **511** are stored in different databases **500**, **510**, it is possible to combine these data and execute filtering based on the rule.

[0107] As described above, in the service providing system **1**A of the present embodiment, the database **500** and the database **510** are connected to the data access control apparatus **100**, which is a computer that creates the attribute information management table **405**, via the virtualization layer **700**. The metadata table **602** stored in the data catalog **600** further includes information about the virtualization layer **700**. The data access control apparatus **100** combines the data **501** and the data **511** via the virtualization layer **700** to cause the database **500** to execute filtering. In this way, even when the data **501** and the data **511** are stored in different databases **500**, **510**, it is possible to combine these data and execute filtering based on the rule.

[0108] Note that each record of the metadata table **602** may further include information indicating validity of data combination via the virtualization layer **700** and information on priority of data combination. In this way, it is possible to more appropriately determine whether to execute filtering by data combination via the virtualization layer **700** between different databases for each data.

[0109] The present invention is not limited to the above embodiments, and can be implemented using any components without departing from the gist of the present invention. For example, the functional blocks of the query service **200** and the authorization service **400** illustrated in FIG. **3** may be realized by different hardware. Alternatively, any functional block may be included in the client **300**, or may be included in the database **500** or the database **510**. Further, some or all of the functional blocks in FIG. **3** may be realized by hardware such as a field-programmable gate array (FPGA).

[0110] The embodiments and modifications described above are merely examples, and the present invention is not limited to these contents as long as the features of the invention are not impaired. Although various embodiments and modifications have been described above, the present invention is not limited to these contents. Other aspects conceivable within the scope of the technical idea of the present invention are also included within the scope of the present invention.

What is claimed is:

1. A data access control method performed by a computer, the method comprising:

acquiring, with the computer, a rule for a request for accessing first data, based on a preset access control policy;

selecting, with the computer, whether to acquire attribute information about an attribute of each record of the first data from outside of a first database in which the first data is stored;

when selecting acquisition of the attribute information, acquiring the attribute information and evaluating the rule based on the attribute information with the computer;

when selecting no acquisition of the attribute information, instructing, with the computer, the first database to execute filtering of the first data based on the rule; and

acquiring, with the computer, a record of the first data corresponding to the access request from the first database, based on an evaluation result of the rule or an execution result of the filtering.

2. The data access control method according to claim **1**, wherein

the method includes

selecting no acquisition of the attribute information when second data including the attribute information is stored in the first database, and

selecting acquisition of the attribute information when the second data is not stored in the first database.

3. The data access control method according to claim **2**, wherein

when selecting no acquisition of the attribute information, the method includes combining the first data and the second data to cause the first database to execute the filtering.

4. The data access control method according to claim **3**, wherein

even when the second data is stored in the first database, in a case where the first data and the second data cannot be combined in the first database, the method includes selecting acquisition of the attribute information.

5. The data access control method according to claim **1**,

wherein the method includes using the computer to provide a query service that generates a first query according to the access request and an authorization service that performs an authorization process based on the access control policy for the access request, and

wherein the authorization service

selects whether to acquire the attribute information from outside of the first database, and when selecting acquisition of the attribute information, acquires the attribute information and evaluates the rule, and

when selecting no acquisition of the attribute information, causes the query service to output a second query for causing the first database to execute the filtering.

6. The data access control method according to claim **5**, wherein

the query service converts the generated first query into the second query according to an instruction from the authorization service and outputs the second query to the first database.

7. The data access control method according to claim **1**, wherein

the method includes referring to an attribute information management table indicating an arrangement place of the attribute information and a function thereof to select whether to acquire the attribute information from outside of the first database.

8. The data access control method according to claim **7**, wherein

the method includes acquiring, with the computer, metadata regarding the first database in which the first data is stored and a second database in which second data including the attribute information is stored, and creating the attribute information management table, based on the acquired metadata.

9. The data access control method according to claim **8**, wherein

the metadata includes information about a function supported by each of the first database and the second database, and information about databases to which the first data and the second data belong.

10. The data access control method according to claim **9**, wherein

the first database and the second database are each connected, via a virtualization layer, to the computer that creates the attribute information management table, and

the metadata further includes information about the virtualization layer.

11. The data access control method according to claim **10**, wherein

the method includes combining the first data and the second data via the virtualization layer to cause the first database to perform the filtering.

12. The data access control method according to claim **11**, wherein

the metadata further includes information about validity and priority of data combination via the virtualization layer.

13. A data access control apparatus comprising:

a policy determination unit that acquires a rule for a request for accessing first data, based on a preset access control policy, and selects whether to acquire attribute information about an attribute of each record of the first data from outside of a first database in which the first data is stored;

an attribute information acquisition unit that acquires the attribute information;

a query unit that causes the first database to execute filtering of the first data based on the rule; and

a communication unit that acquires a record of the first data corresponding to the access request from the first database, wherein

the attribute information acquisition unit acquires the attribute information when the policy determination unit selects acquisition of the attribute information,

the policy determination unit evaluates the rule, based on the attribute information acquired by the attribute information acquisition unit,

the query unit causes the first database to execute the filtering when the policy determination unit selects no acquisition of the attribute information, and

the communication unit acquires a record of the first data corresponding to the access request from the first database, based on an evaluation result of the rule from the policy determination unit or an execution result of the filtering in the first database.

14. A data access control program for causing a computer to execute:

a first process of acquiring a rule for a request for accessing first data, based on a preset access control policy;

a second process of selecting whether to acquire attribute information about an attribute of each record of the first data from outside of a first database in which the first data is stored;

a third process of, when selecting acquisition of the attribute information in the second process, acquiring the attribute information and evaluating the rule based on the attribute information;

a fourth process of, when selecting no acquisition of the attribute information in the second process, instructing the first database to execute filtering of the first data based on the rule; and

a fifth process of acquiring a record of the first data corresponding to the access request from the first database, based on an evaluation result of the rule by the third process or an execution result of the filtering by the fourth process.

* * * * *