Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3036711 A1 2018/04/12

(21) 3 036 711

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2017/09/29
(87) Date publication PCT/PCT Publication Date: 2018/04/12
(85) Entrée phase nationale/National Entry: 2019/03/12
(86) N° demande PCT/PCT Application No.: US 2017/054416
(87) N° publication PCT/PCT Publication No.: 2018/067399
(30) Priorités/Priorities: 2016/10/03 (US62/403,587);
2017/05/09 (US62/503,562)

(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01)

(71) Demandeur/Applicant:
STRATUS DIGITAL SYSTEMS, US

(72) Inventeurs/Inventors:
TORKELSON, CARY, US;
CHANIN, KENNETH ARI, US;
SULLIVAN, PATRICK J., US;
GEANKOPLIS, BRAD, US

(74) Agent: FASKEN MARTINEAU DUMOULIN LLP

(54) Titre : SERVEUR DE TRANSACTION TRANSITOIRE
(54) Title: TRANSIENT TRANSACTION SERVER

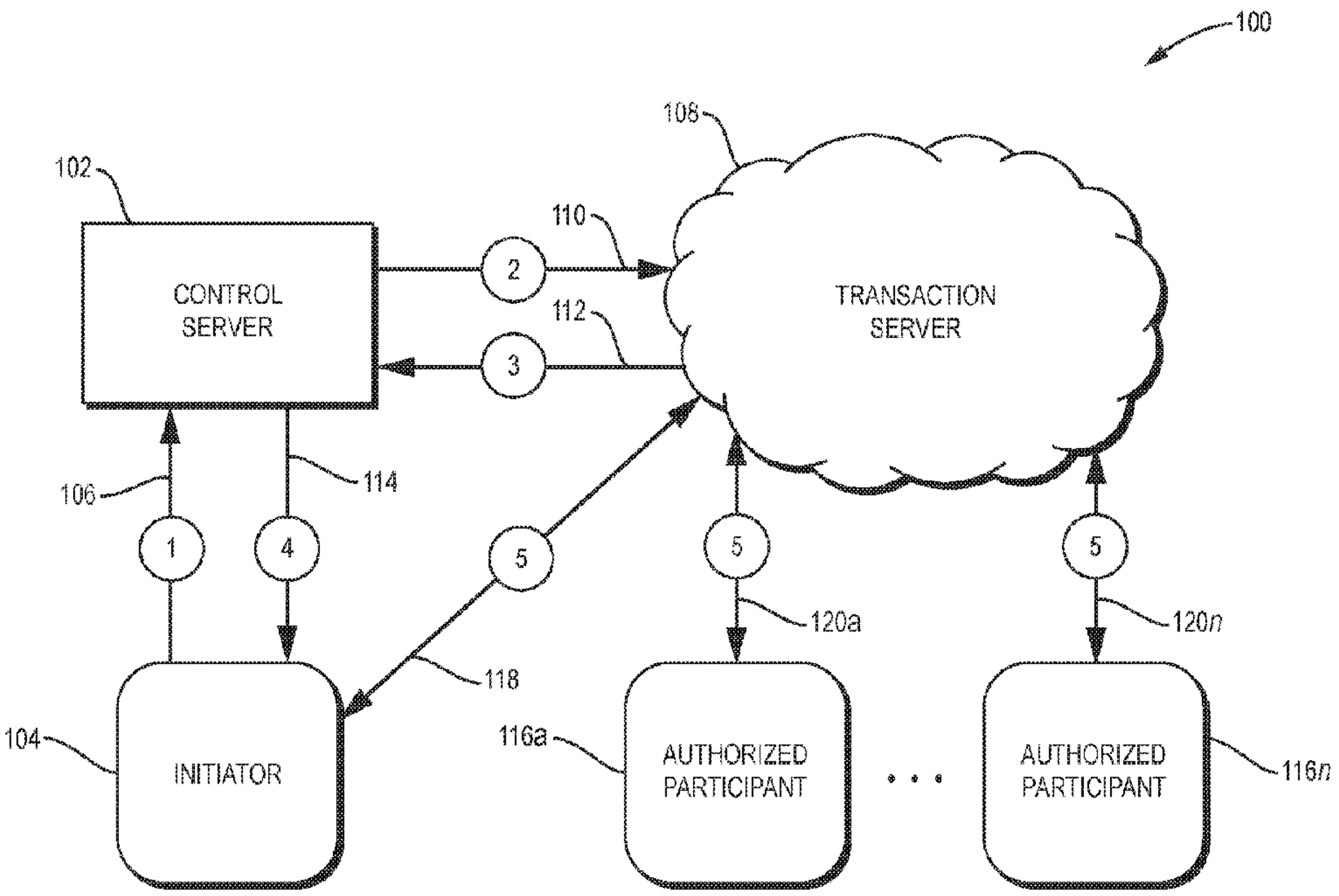

FIG. 1

(57) Abrégé/Abstract:
A networked computer system enables one or more transactions to be executed securely. An initiator sends a service request to a control server. The control server creates or selects an existing transaction server for the sole purpose of executing the transaction requested by the service request. If the transaction server is pre-existing, it may be in an inaccessible state and then be made accessible in response to receiving the service request. The control server informs the initiator of the created transaction server. The initiator (and possibly one or more other authorized participants) engages in the transaction with the transaction server, independently of the control server. The transaction server terminates, such as upon completion of the transaction, the expiration of a timeout period, or satisfaction of another server termination criterion. The use of such a one-time transaction server increases security, privacy, and anonymity.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 April 2018 (12.04.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/067399 A1

(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/US2017/054416

(22) International Filing Date:
29 September 2017 (29.09.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/403,587	03 October 2016 (03.10.2016)	US
62/503,562	09 May 2017 (09.05.2017)	US

(71) Applicant: **STRATUS DIGITAL SYSTEMS** [US/US]; 4675 Goodpasture Loop #83, Eugene, Oregon 97401 (US).

(72) Inventors: **TORKELSON, Cary**; 522 Fulvue Dr., Eugene, Oregon 97405 (US). **CHANIN, Kenneth Ari**; PO Box 14297, Savannah, Georgia 31416 (US). **SULLIVAN, Patrick J.**; 4675 Goodpasture Loop #83, Eugene, Oregon 97401 (US). **GEANKOPLIS, Brad**; 11354 Mother Lode Circle, Gold River, California 95670 (US).

(74) Agent: **PLOTKIN, Robert**; BLUESHIFT IP LLC, 1 Broadway, 14th Floor, Cambridge, Massachusetts 02142 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,

(54) Title: TRANSIENT TRANSACTION SERVER

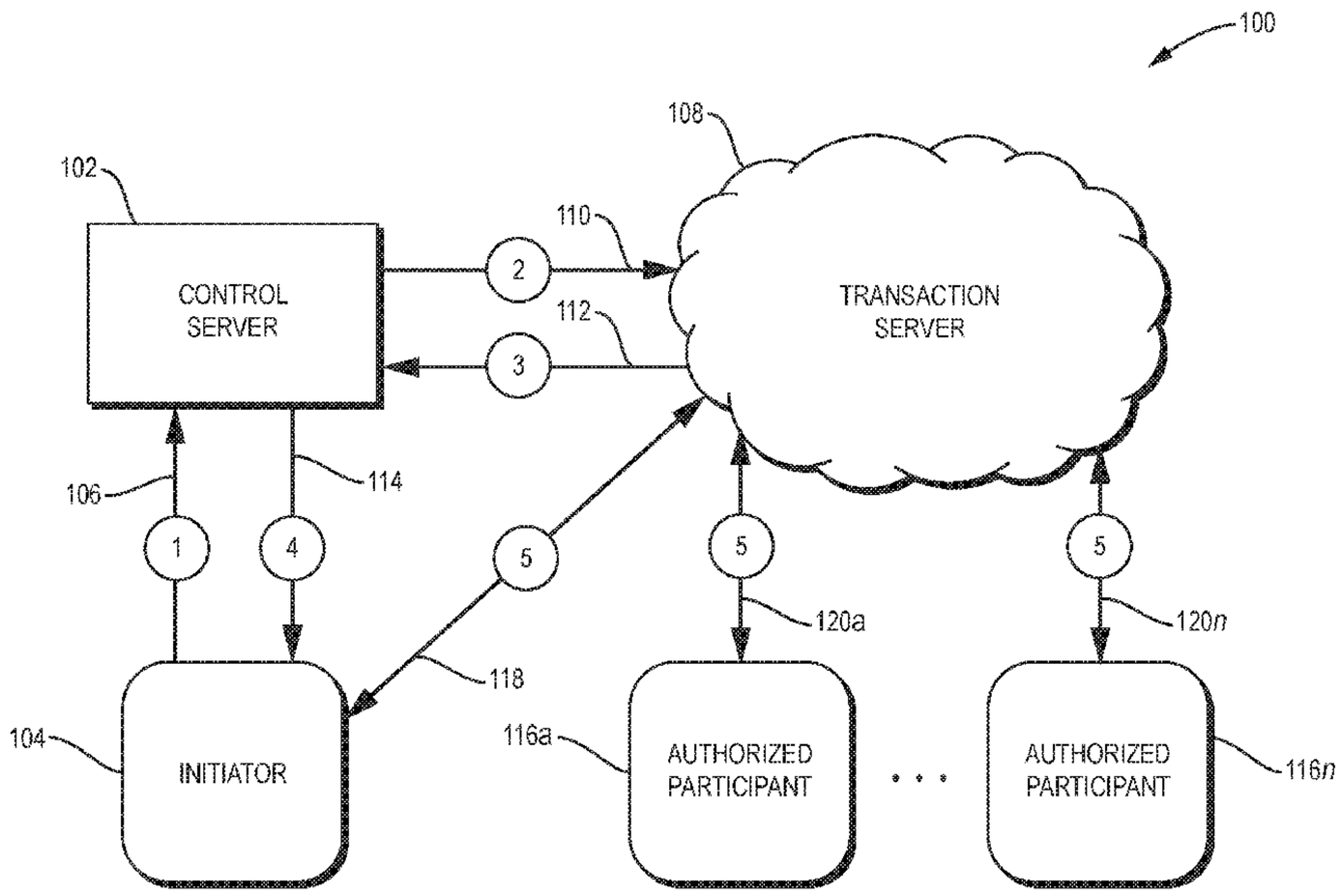

FIG. 1

(57) Abstract: A networked computer system enables one or more transactions to be executed securely. An initiator sends a service request to a control server. The control server creates or selects an existing transaction server for the sole purpose of executing the transaction requested by the service request. If the transaction server is pre-existing, it may be in an inaccessible state and then be made accessible in response to receiving the service request. The control server informs the initiator of the created transaction server. The initiator (and possibly one or more other authorized participants) engages in the transaction with the transaction server, independently of the control server. The transaction server terminates, such as upon completion of the transaction, the expiration of a timeout period, or satisfaction of another server termination criterion. The use of such a one-time transaction server increases security, privacy, and anonymity.

[Continued on next page]

WO 2018/067399 A1

WO 2018/067399 A1

OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

Transient Transaction Server

BACKGROUND

Transactions performed by computer systems have a variety of security vulnerabilities. As transactions having increasing value are performed over the Internet and by computers accessible via the Internet, such transactions become increasingly vulnerable to attack.

What is needed, therefore, are improved techniques for protecting transactions, and the data accessed by such transactions, against attacks.

SUMMARY

A networked computer system enables one or more transactions to be executed securely. An initiator sends a service request to a control server. The control server creates or selects an existing transaction server for the sole purpose of executing the transaction requested by the service request. If the transaction server is pre-existing, it may be in an inaccessible state and then be made accessible in response to receiving the service request. The control server informs the initiator of the created transaction server. The initiator (and possibly one or more other authorized participants) engages in the transaction with the transaction server, independently of the control server. The transaction server terminates, such as upon completion of the transaction, the expiration of a timeout period, or satisfaction of another server termination criterion. The use of such a one-time transaction server increases security, privacy, and anonymity.

Other features and advantages of various aspects and embodiments of the present invention will become apparent from the following description and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a dataflow diagram of a system for using a transient server to execute one or more transactions according to one embodiment of the present invention;

FIG. 2 is a flowchart of a method performed by the system of FIG. 1 according to one embodiment of the present invention;

FIG. 3 is a dataflow diagram of a system for making an existing but unaddressable transaction server addressable in order to execute one or more transactions according to one embodiment of the present invention;

FIG. 4 is a flowchart of a method performed by the system of FIG. 3 according to one embodiment of the present invention;

FIG. 5 is a dataflow diagram of a system for using a pool of IP addresses to change the IP addresses of transaction servers over time according to one embodiment of the present invention;

FIG. 6 is a flowchart of a method perform by the system of FIG. 5;

FIG. 7 is a dataflow diagram of a system for continuously provisioning transaction servers according to one embodiment of the present invention;

FIG. 8 is a flowchart of a method performed by the system of FIG. 7 according to one embodiment of the present invention.

DETAILED DESCRIPTION

Embodiments of the present invention include systems and methods for provisioning a virtual server that is used to process a finite number of (e.g., one) transactions, after which the server is de-provisioned. Such systems and methods have a variety of advantages, such as the ability to better secure transactions from outside interference, and to allow more anonymity and privacy for such transactions and the data they access.

For example, referring to FIG. 1, a dataflow diagram is shown of a system 100 for executing transactions according to one embodiment of the present invention. Referring to FIG. 2, a flowchart is shown of a method 200 performed by the system 100 of FIG. 1 according to one embodiment of the present invention.

The system 100 includes a control server 102. The control server 102 is a "computer system," as that term is used herein. The term "computer system" refers herein to any one or more computers acting in coordination with each other to execute instructions to perform operations on input and/or stored data to produce output. A computer system, such as the control server 102 may, for example, consist of a single computer of any kind (e.g., server, desktop PC, laptop PC, smartphone, tablet computer, PDA) or any combination of any number of computers of any kinds(s). A computer

system and/or any of its component computers may be a physical machine or a virtual machine. A computer system may, therefore, include any one or more physical machines and/or any one or more virtual machines in any combination, in any location relative to each other, including configurations in which two or more machines are either local to each other or accessible to each other remotely over a network. The component computers in a computer system may communicate with each other via any mechanism(s), such as one or more wires, buses, cables, wired network connections, wireless network connections, application program interfaces (APIs), or any combination thereof.

Although the computer system 102 is referred to herein as a "server," the control server 102 need not operate according to a client-server architecture. The term "server" is used herein to refer to the control server 102, and to other servers disclosed herein, merely for convenience and to indicate that the control server 102 responds to requests to provide services. The term "server," as used herein, refers to any kind of computer system, whether or not that computer system operates according to a client-server architecture. Therefore, the control server 102, and other servers disclosed herein, need not be a "server" as that term is used conventionally, but more generally may be any kind of computer system.

The system 100 also includes an initiator 104. The initiator 104 may, but need not be, a computer system. The initiator 104 provides a request 106 for a transaction to the control server 102, and the control server 102 receives the request 106 (FIG. 2, operation 202). The terms "transaction" and "service" are used interchangeably herein. The terms "service request" and "transaction request" are used herein to refer to requests to perform a service/transaction.

The control server 102 may have an address to which communications, such as the request 106, may be addressed. The control server 102's address may, for example, be a public Internet Protocol (IP) address and/or port, although this is not a limitation of the present invention. As another example, the control server 102's address may be an address that is behind a corporate or other private firewall. The control server 102 may or may not require and engage in secure communications. For example, the request 106 and other communications engaged in by the control server 102 may be secure communications. As another example, the control server 102 may limit its connections to

only well-known initiators, or have other security measures in place to authenticate the initiator 104 and other initiators, such as hardware key devices and/or biometrics.

The initiator 104 may, but need not be, the end user or machine that intends to engage in the transaction requested by the request 106. For example, a proxy (not shown) may initiate the transaction, which may cause the initiator 104 to provide the request 106 to the control server 102.

The initiator 104 may provide (e.g., transmit) the request 106 to the control server 102 by addressing the request 106 to the control server 102's address (e.g., IP address and/or port), in response to which the system 100 may deliver the request 106 to the control server 102 at the control server 102's address. The initiator 104 may provide the request 106 to the control server 102 in any of a variety of ways and using any of a variety of mechanisms, such as by providing (e.g., transmitting) the request 106 via one or more wires, buses, cables, wired network connections, wireless network connections, application program interfaces (APIs), or any combination thereof. The same applies equally to any other communication disclosed herein.

In general, the request 106 contains data indicating a request to perform a particular transaction. Such data may, for example, indicate any one or more of the following in any combination: the type of transaction to be performed, one or more parameters of the transaction, and one or more values of each such parameter. The request 106 may request any type of transaction, such as any type of Software as a Service (SaaS) transaction. Examples of such transactions include, but are not limited to, digital asset exchange services, health care record services, secure messaging services, and industrial internet of things software/firmware update services.

The request 106 may, for example, include data representing any one or more of the following properties:

Transaction Type. This is the type of transaction requested by the request 106. The type of transaction determines what kind of transaction server 108 should be created. Examples include transfer of digital assets, secure message delivery, software distribution, and data manipulation and analysis.

Timeout. The timeout specifies how long, at most, the transaction server 108 should remain in existence. Typically, the transaction server 108 will self-terminate after the transaction is complete, but if for some reason the transaction does not complete

within the timeout period, the transaction server 108 will terminate prior to the transaction being complete.

Transaction Specific Properties. Certain transaction types may have additional properties that are relevant to the transaction being processed, and need to be specified prior to the transaction server 108 being created. Examples include the maximum allowed message size for transfer and the maximum allowed amount of currency for exchange. These parameters apply immediately before any authorized participant communicates with the transaction server.

Privacy. Other transaction properties may be private, or may only be delivered directly to the transaction server 108, bypassing the control server 102. Example of these properties include maximum number of downloads for a file and a password needed for a secondary participant to enter the transaction.

Authorized Participants. A list of authorized participants can be specified at service request time. If so, only those participants are allowed to communicate with the transaction server 108 for participation in the transaction. Authorized participants may be specified by machine IP address, user name and password, or other ways by which a participant may contact the transaction server 108 (browser type, device type, etc.).

Authentication Method. This determines how entities authenticate with the transaction server 108. Examples of authentication methods include that no authentication is needed (public service), machine address authentication (e.g., IP address), software authentication (user/password pair), two-factor authentication, confirming a code or token, using an authorized browser extension, biometric authentication, or a combination of the above.

Privacy Level/Logging. Since the transaction server 108 self-terminates after the transaction is complete, no record of the transaction is stored permanently. It may be that the initiator 104 would like a record of the transaction to persist. If so, the transaction server 108 may send, to the control server 102 or another server (not shown), information to create a permanent record of the transaction server 108 and/or the transaction, such as copying log files to a persistent storage location and/or taking a snapshot of the transaction server 108 (or a portion thereof) as it existed before it was terminated.

In the example of FIG. 1, a single control server 102 is shown. Such a control server 102 may, for example, be general purpose, meaning that it may be capable of

processing many (e.g., any) type of request 106. As another example, the control server 102 may be special purpose, meaning that it may be capable of processing only one or a finite set of request types. In the latter case, the system 100 may include a plurality of control servers 102, each of which may be capable of processing one or more types of requests.

The system 100 also includes a transaction server 108. The transaction server 108 may not exist (e.g., may not have been provisioned) before the initiator 104 provides the request 106 to the control server 102. Instead, the control server 102, in response to receiving the request 106, may create 110 (e.g., provision) the transaction server 108 (FIG. 2, operation 204). The control server 102 may, for example, create 110 the transaction server 108 based on contents of the request 106. For example, the control server 102 may generate the transaction server 108 to be capable of executing the type of transaction indicated by the request 106. Additionally or alternatively, the request 106 may include data representing one or more properties of a server that is required and/or desired to perform the requested transaction, in which case the control server 102 may create the transaction server 108 to have the properties represented by the request 106. Additionally or alternatively, the control server 102 may have access to a number of pre-defined transaction server images, from which the control server 102 may create the transaction server 108. Different transaction server images may, for example, be used to execute different types of transactions (e.g., one transaction server image to execute transactions of a first type and another transaction server image to execute transactions of a second type), or to execute the same type of transaction but with different properties as specified in the transaction request 106.

Element 110 in FIG. 1 represents the creation of the transaction server 108 by the control server 102. Such creation may be performed in any of a variety of ways. For example, the control server 102 may provide (e.g., transmit) a creation request to a cloud service provider (not shown) specifying that the transaction server 108 be provisioned to handle request 106. The transaction server 108 may, for example, be created using an industry standard machine image that was previously constructed to handle the type of SaaS request represented by the request 106 (such as may be identified using information in the request 106 which indicates the type of the request 106). Such a machine image may, for example, contain all of the software required to execute the transaction requested

by the request 106, including all necessary customizations and configurations. Any transaction-specific parameters, such as the participants in the transaction, the types of data and/or assets being exchanged by the transaction, the agreed-upon exchange rates in the transaction, and others, may be specified either by a control file made available to the transaction server 108 at startup, or in the process of creating the transaction server 108.

The transaction server 108 is an example of a "computer system," as that term is used herein. The transaction server 108 may, for example, be a physical or virtual server (e.g., a computer system that is accessible over the Internet via a public IP address), or may include more than one virtual or cloud server. The transaction server 108 may include on-demand resources for assisting the transaction server 108 in executing the transaction requested by the request 106, such as private databases, high-speed disk caches, and/or third-party services and/or software.

The control server 102 identifies 112 (e.g., receives) an address at which the transaction server 108 is accessible (FIG. 2, operation 206). The control server 102 may, for example, receive the address of the transaction server 108 in a message transmitted by the transaction server 108 to the control server 102. As another example, the control server 102 may generate the address of the transaction server 108 and assign the generated address to the transaction server 108. The transaction server 108 may also indicate to the control server 102 (such as by transmitting an appropriate message to the control server 102) that the transaction server 108 is ready to execute the transaction requested by the initiator 104 in the request 106. Therefore, in general, element 112 in FIG. 1 indicates that, after creation of the transaction server 108, the transaction server 108 provides, to the control server 102, all information that the initiator 104 and/or authorized participants 116a-*n* will need to connect to and communicate with the transaction server 108.

The control server 102 provides 114 the address of the transaction server 108 to the initiator 104, such as by transmitting a message containing the address of the transaction server 108 to the initiator 104 (FIG. 2, operation 208). Such a message may be transmitted over a digital communication network, such as the Internet. This message may, for example, be transmitted in response to the request 106 from the initiator 104. More generally, element 114 in FIG. 1 indicates that the control server 102 indicates to the initiator 104 that the transaction server 108 is ready to execute the transaction

requested by the request 106 and provides the initiator 104 with all information necessary for the initiator 104 to communicate with the transaction server 108, such as the transaction server 108's address.

As briefly mentioned above, the system 100 may also include, in addition to the initiator 104, one or more participants 116a-*n* who are authorized to participate in the transaction requested by the request 106. Note that *n* may be any number, such as 0, 1, 2, or higher. If *n*=0, then the initiator 104 is the only participant in the transaction requested by the request 106. If *n*=1, then only the initiator 104 and the authorized participant 116a participate in the transaction requested by the request 106, and the system 100 would not include authorized participant 116*n*. In certain embodiments of the present invention, the initiator 104 participates in the creation of the transaction server 108, but does not participate in transactions executed by the transaction server 108. In such embodiments, the authorized participant(s) 116a-*n* participate in the transaction(s) executed by the transaction server 108, while the initiator 104 does not participate in such transactions.

The control server 102 may, in certain embodiments, notify the authorized participants 116a-*n* about the transaction server 108, such as in the same way that the control server 102 notifies the initiator 104 about the transaction server 108. Alternatively, for example, the initiator 104 may inform the authorized participants 116a-*n* about the transaction server 108 after the control server 102 notifies the initiator 104 about the transaction server 108.

Once the transaction server 108 has been created and the initiator 104 (and possibly the authorized participants 116a-*n*) have been informed about the transaction server 108, the initiator 104 may engage 118 in the transaction requested by the request 106 with the transaction server 108 (FIG. 2, operation 210). The initiator 104 may provide (e.g., transmit) information to the transaction server 108 by addressing the transaction server 108 using the address received 114 by the initiator 104 from the control server 102. The same is true of communications, if any, between the authorized participants 116a-*n* and the transaction server 108.

If the authorized participants 116a-*n* are to participate in the transaction, then the authorized participants 116a-*n* may also engage 120a-*n* in the transaction requested by the request 106 with the transaction server 108. In general, elements 118 and 120a-*n* in FIG. 1 represent all interactions between and among the initiator 104, authorized participants

116a-*n*, and the transaction server 108 that are involved in the execution of the transaction by the transaction server 108. Such interactions may include transmitting and receiving messages (e.g., from the transaction server 108 to the initiator 104 and/or other participants, or from the initiator and/or other participants to the transaction server 108) over a network (such as the Internet), where such messages may include commands and/or data.

At some point after the completion of the execution of the transaction by the transaction server 108, the transaction server 108 terminates (e.g., is de-provisioned or otherwise deleted, destroyed, or inactivated) (FIG. 2, operation 212). For example, the transaction server 108 may terminate immediately upon completion of the transaction or otherwise in response to completion of the transaction, or after (and in response to) the expiration of some predetermined timeout period after initiation of the transaction, even if the transaction does not complete. These server termination criteria are merely examples and are not limitations of the present invention. The transaction server 108 may terminate in response to any one or more server termination criteria being satisfied, such as terminating in response to:

completion of some predetermined number of transactions by the transaction server 108;

determining that a transaction being executed by the transaction server 108 satisfies an anomalous condition or otherwise is aberrant in some way (e.g., has one or more parameter values which fall outside permissible ranges of values);

determining that at least some predetermined number of participants have participated in a transaction with the transaction server 108;

satisfaction of any binary criterion in relation to a transaction executed by the transaction server 108;

determining that a transaction executed by the transaction server 108 (or the transaction server 108 as a whole) has transferred at least some predetermined amount of data; and

determining that a transaction executed by the transaction server 108 (or the transaction server as a whole) has transferred at least some predetermined amount of files.

As just described, in certain embodiments, the transaction server 108 may execute at most one (e.g., exactly one) transaction before terminating. In other embodiments of

the present invention, the transaction server 108 executes a finite number of transactions, greater than one, before terminating, or executes any number of transactions for no more than some predetermined finite amount of time (referred to herein as the timeout period), before terminating. As another example, the transaction server 108 may terminate after executing a transaction having a specified property (possibly after having executed one or more additional transactions). More generally, the system 100 may apply one or more terminating criteria to the transaction server 108. The system 100, in response to determining that the termination criteria have been satisfied, may terminate the transaction server 108. The particular examples of termination criteria explicitly disclosed herein (e.g., maximum number of transactions, maximum amount of time, transaction type) are merely examples and are not limitations of the present invention.

As the description above implies, the control server 102 is involved in the creation of the transaction server 108, but does not otherwise participate in the execution of the transaction by the transaction server 108. For example, the control server 102 may not send or receive any information (e.g., commands and/or data) to or from the transaction server 108 other than that required to create the transaction server 108 and to obtain information about the creation and availability of the transaction server 108, as indicated by elements 110, 112, and 114 in FIG. 1. In some embodiments of the present invention, the control server 102 does not send or receive information to or from the transaction server 108 in the process of creating the transaction server 108, in which case the control server 102 does not send or receive any information to or from the transaction server 108 at any point.

In particular, after the transaction server 108 has been created, the transaction server 108 executes the transaction requested by the request 106 without the involvement of the control server 102. For example, the transaction server 108 engages in all communication 118 and 120a-*n* with the initiator 104 and authorized participants 116a-*n* without the involvement of the control server 102. As another example, the transaction server 108 terminates without the involvement of the control server 102. As another example, no commands or data involved in the transaction, such as critical, private, or sensitive data, may pass through the control server 102.

The control server 102 may or may not be involved in the termination of the transaction server 108. For example, the control server 102 may cause the transaction

server 108 to terminate, such as by sending a termination instruction to the transaction server 108 or through another mechanism. Alternatively, for example, the transaction server 108 may terminate itself, without the involvement of the control server 102.

As mentioned above, the control server 102 and the transaction server 108 each have at least one address (e.g., IP address) at which they are addressable. The control server 102's address is distinct and different from the transaction server 108's address. For example, the control server 102 may have one IP address and the transaction server 108 may have another IP address that is different from the control server 102's IP address. As another example, the control server 102 and the transaction server 108 may be accessible at different ports of the same IP address. The control server 102 may be a physical machine and the transaction server 108 may be another physical machine that is distinct from the control server 102. As yet another example, the control server 102 and transaction server 108 may be different virtual machines residing on the same physical computer system.

One advantage of embodiments of the present invention, such as the system 100 of FIG. 1 and the method 200 of FIG. 2, is that they enable the transaction requested by the request 106 to be performed more securely than in prior art systems. Creating and using the transaction server 108 solely for a finite number of transactions (e.g., one transaction), and executing the transaction independently of the control server 102, protects against the risk that a successful attack on that transaction will also compromise other transactions and the data accessible to such transactions, because any attack on the transaction execution by the transaction server 108 has no access to other transactions or to the data accessible to such transactions. Embodiments of the present invention, therefore, include improved computer systems and methods which address a previously-unsolved technical problem in computer security, namely the technical problem of how to protect servers against being compromised by attacks on other servers. This problem, and its solution by embodiments of the present invention, are inherently rooted in computer technology, represent an improvement to computer technology, and use particular combinations of non-conventional computer technology to produce previously unrealized technical benefits.

One advantage of embodiments of the present invention illustrated in FIGS. 3-8 is that they enable transactions to executed by computers more quickly and efficiently than

in prior art systems by provisioning transaction servers before they are needed, thereby enabling such transaction servers to be available for use without the delay that would be incurred if such transaction servers were not provisioned until they were needed. The embodiments illustrated in FIGS. 3-8, therefore, solve the technical problem of how to enable virtual servers to execute transactions more efficiently, and do so using a solution which is inherently rooted in computer technology, which is an improvement to computer technology, and which uses particular combinations of non-conventional computer technology to produce previously unrealized technical benefits.

As described above, in some embodiments of the present invention, the transaction server 108 may not exist (e.g., be provisioned) before the control server 102 receives the request 106. In other embodiments of the present invention, however, the transaction server 108 may exist before the control server receives the request 106. In such embodiments, however, the transaction server 108 may be inaccessible, at least to any component of the system 100 other than the control server 102, before the request 106 is received. The transaction server 108 may even be inaccessible to any component of the system 100, including the control 102, before the request 106 is received. As will be described in more detail below, in such embodiments, in response to receiving the request 106, the control server 102 may make the transaction server 108 accessible (e.g., to the initiator 104 and/or the authorized participants 116a-n), so that the transaction server 108 becomes available to execute one or more transactions in any of the ways described above.

More specifically, referring to FIG. 3, a dataflow diagram is shown of a system 300, including one or more "dark" transaction servers 308a-*m*, for making one of the dark transaction servers 308a-*m* accessible for use as a transaction server 308. Referring to FIG. 4, a flowchart is shown of a method 400 performed by the system 300 of FIG. 3 according to one embodiment of the present invention. Although the description herein shows a plurality of dark transaction servers 308a-*m* and the description refers herein to the plurality of dark transaction servers 308a-*m*, all such references should be understood to refer to any number of transaction servers, including as few as one transaction server.

The system 300 of FIG. 3 may perform some or all of the functions of the system 100 of FIG. 1. For ease of illustration and explanation, only the aspects of the system 300 of FIG. 3 that differ from the system 100 of FIG. 1 are described herein. The absence of a

description herein of aspects of the system 100 of FIG. 1 in connection with the system 300 of FIG. 3 does not imply that the system 300 of FIG. 3 does not also have those aspects. The same is true of the method 400 of FIG. 4 relative to the method 200 of FIG. 2.

The control server 102 may provision 310 one or more transaction servers 308a-*m* (referred to herein as "dark" transaction servers), even before the initiator 104 provides the request 106 to the control server 102 (FIG. 4, operation 402). One reason why it may be beneficial to provision the dark transaction servers 308a-*m* before the control server 102 receives the request 106 is that there may be a noticeable delay between the time at which the control server 102 receives the request 106 and the time at which a transaction server (e.g., the transaction server 108 in FIG. 1) may be provisioned and become fully available to execute transactions. Such delays may be undesirable and it may be beneficial to mitigate or eliminate such delays when using a transaction server to execute time-sensitive transactions. Delays of five seconds or more are undesirable in many environments, and in some situations such delays may be as long as two or three minutes, or even longer. As described in more detail below, embodiments of the present invention may be used to mitigate or eliminate such delays while maintaining the security benefits of the transient nature of the transaction server 108.

The control server 102 provisions the dark transaction servers 308a-*m* in a manner that makes the dark transaction servers 308a-*m* unaddressable. This unaddressibility is what makes the dark transaction servers 308a-*m* "dark." The control server 102 may make the dark transaction servers 308a-*m* unaddressable immediately upon provisioning them, such that the dark transactions servers 308a-*m* are not addressable unless and until they are subsequently made addressable, such as in any of the ways described below.

The control server 102 may make the dark transaction servers 308a-*m* unaddressable in any of a variety of ways. For example, the control server 102 may configure one or more security policies of a server provider that provides the dark transaction servers 308a-*m* to make the dark transaction servers 308a-*m* unaddressable.

Even while the dark transaction servers are unaddressable, they may still be capable of making outgoing requests. While a server is unaddressable, it may poll the control server 302 to determine whether the control server 302 has received a transaction request that has not yet been assigned to a transaction server. As described in more detail

below, the control server 302 may assign transaction requests to dark transactions servers and provide transaction parameters and other information to the dark transaction servers when they are needed to service transaction requests.

In general, regardless of the particular mechanism that the control server 102 uses to make the dark transaction servers 308a-*m* unaddressable, the effect of such unaddressability is to prohibit elements of the system 300 (such as the initiator 104, the authorized participants 116a-*n*, and even the control server 102 itself) from seeing the dark transaction servers 308a-*m* on the network and from sending network traffic to or receiving network traffic from the dark transaction servers 308a-*m*. As a result, such elements of the system 300 cannot alter the configurations of the dark transaction servers 308a-*m* or alter the predetermined behaviors of the dark transaction servers 308a-*m* while the dark transaction servers are in their unaddressable (dark) state.

Now assume that the control server 102 receives the transaction request 106, as shown and described above in connection with FIGS. 1 and 2. In response to receiving the request 106, the control server 102 selects 312 one of the dark transaction servers 308a-*m* and makes 314 the selected dark transaction server addressable over the network (FIG. 4, operation 404). The resulting addressable (non-dark) transaction server is shown as transaction server 308 in FIG. 3.

The control server 102 may make the transaction server 308 addressable in any of a variety of ways, such as by updating the server provider's security policy to allow network traffic to be sent to and from the transaction server 308. Such updates may specify particular entities (e.g., the initiator 104 and/or authorized participants 116a-*n*) who are authorized to communicate with the transaction server 308, so that only those entities, and no other entities, can communicate with the transaction server 308. As a result, elements of the system 300, such as the initiator 104, the authorized participants 116a-*n*, and the control server 102 itself, may see the transaction server 308 on the network and may send network traffic to and receive network traffic from the transaction server 308 while the transaction server 308 is in its addressable (non-dark) state. The effect of provisioning the transaction server 308 in an unaddressable state before the control server 302 receives the request 106, and then making the transaction server 308 addressable in response to receiving the request 106, is similar, from a security perspective, to provisioning the transaction server 308 in response to receiving the request

106, because the transaction server 308 is not accessible and therefore cannot be tampered with before the request 106 is received, but results in making the transaction server 308 accessible to transaction participants (e.g., the initiator 104 and/or authorized participants 116a-*n*) more quickly than if the transaction server 308 were not provisioned until after the request 106 is received, by eliminating the time required to provision the transaction server 308 after receiving the request 106.

The control server 302 may provision 316 a new dark transaction server and include the newly provisioned server within the pool of dark transaction servers 308a-*m* (FIG. 4, operation 406). The control server 302 may provision the new dark transaction server in any of the ways disclosed above in connection with the provisioning 310 of the dark transaction servers 308a-*m* (FIG. 4, operation 402). The newly provisioned server is then available for selection as a transaction server the next time operation 404 of method 400 is performed.

At a subsequent time, the transaction server 308 terminates 316 (FIG. 4, operation 408). The transaction server 308 may terminate, for example, in any of a variety of ways, such as any of the ways disclosed herein in connection with termination of the transaction server 108 of FIG. 1 (FIG. 2, operation 212), such as by terminating the transaction server 308 in response to the transaction server 308 completing its transaction or in response to a timeout period elapsing before the transaction server 308 has completed its transaction. The control server 302 may provision the new dark transaction server in response to any of a variety of conditions being satisfied, including complex conditions. For example, the control server 302 may let the existing transition server 308 operate during business hours and terminate the existing transaction server 308 after it has operated in standby mode for two hours during off-hours.

The control server 302 may provision a new dark transaction server within the pool of dark transaction servers 308a-*m* in response to terminating the transaction server 308, in any of the ways disclosed herein. The newly provisioned server is then available for selection as a transaction server the next time operation 404 of method 400 is performed.

Terminating the transaction server 308 in operation 408 is only one possibility and is not a limitation of the present invention. Alternatively, for example, instead of terminating the transaction server 308, the transaction server 308 may be put back into an

unaddressable state (using any of the techniques disclosed herein), and thereby again become part of the pool of dark transaction servers 308a-*m*, where it would again become available for selection to perform one or more additional transactions. Although the resulting dark transaction server would not be pristine, because it would already have performed a transaction, it would again be in an unaddressable state, thereby preventing any outside entity from affecting its function. Such an embodiment would reduce the overall cost of operating the system 300 compared to embodiments in which transaction servers are terminated, and would provide high-volume transaction systems with the ability to handle such high volumes without needing to provision and terminate servers at a rapid rate.

Certain embodiments of the present invention may maintain both a pool of dark transaction servers (such as the pool of dark transaction servers 308a-*m* shown in FIG. 3) and a pool of IP addresses. Such an embodiment is illustrated by the system 500 of FIG. 5, and the corresponding method 600 of FIG. 6.

The system 500 of FIG. 5 includes a control server 502, which may be the same as or similar to the control server 302 of FIG. 3. The system 500 of FIG. 5 also includes a pool of dark transaction servers 508a-*m*, which may be the same as or similar to the pool of dark transaction servers 308a-*m* of FIG. 3. For ease of illustration and explanation, only the aspects of the system 500 of FIG. 5 that differ from the system 300 of FIG. 3 are described herein. The absence of a description herein of aspects of the system 300 of FIG. 3 in connection with the system 500 of FIG. 5 does not imply that the system 500 of FIG. 5 does not also have those aspects. The same is true of the method 600 of FIG. 6 relative to the method 400 of FIG. 4.

As in the system 300 of FIG. 3 and the method 400 of FIG. 4, in the system 500 of FIG. 5, the control server 502 may provision one or more of the transaction servers 508a-*m*, even before the initiator 104 provides the request 106 to the control server 502, using any of the techniques disclosed above (FIG. 6, operation 602). Similarly, as in the system 300 of FIG. 3 and the method 400 of FIG. 4, in the system 500 of FIG. 5, the control server 502 may receive the transaction request 106, as shown and described above in connection with FIGS. 1 and 2. In response to receiving the request 106, the control server 502 selects 510 one of the dark transaction servers 508a-*m* and makes 314 the selected dark transaction server addressable over the network (FIG. 6, operation 604).

The resulting addressable (non-dark) transaction server is shown as transaction server 508 in FIG. 5.

The control server 502 also selects 512 (e.g., randomly) one of the IP addresses from a pool 520 of IP addresses in the system 500, assigns the selected IP address to the selected transaction server 508 (FIG. 6, operation 606). The selected IP address is then no longer available for selection or assignment to any other servers.

The control server 102 enables 514 the selected transaction server 508 to be addressable at the selected IP address, such as by updating the server provider's security policy to allow network traffic to be sent to and from the transaction server 508 at the selected IP address (FIG. 6, operation 608).

At a subsequent time (such as in response to the transaction server 508 completing its transaction or in response to the lapse of a timeout period without completion of the transaction by the transaction server 508), the control server 102 two actions 516: (1) disassociates the selected IP address from the transaction server 508 and may or may not make the selected IP address again available for selection from within the pool 520 of IP addresses (FIG. 6, operation 610); and (2) makes the transaction server 508 unaddressable and makes the transaction server 508 again available for selection for use from the pool of dark transaction servers 508a-*m* (FIG. 6, operation 612).

The system 500 of FIG. 5 and the method 600 of FIG. 6 make it possible for the transaction servers 508a-*m* to be used more than once (whether or not they actually are used more than once), thereby obtaining the benefit of reducing or eliminating the time required to provision new servers in response to each new transaction request, while randomizing the IP addresses that are assigned to transaction servers as they are brought into use, thereby reducing or eliminating the security risks introduced by reusing servers. Furthermore, regardless of the number of transactions performed by each of the transaction servers 508a-*m* before terminating, the system 500 of FIG. 5 and the method 600 of FIG. 6 have the benefit that they make a dark transaction server's IP address unknowable until it is activated for use in a transaction.

Yet another embodiment of the present invention is illustrated by FIG. 7, which is a dataflow diagram 700 of a system for continuously provisioning transaction servers, and FIG. 8, which is a flowchart of a method 800 performed by the system 700 of FIG. 7 according to one embodiment of the present invention.

The control server 702 (which may be the same as or similar to any of the other control servers 102, 302, and 502 disclosed herein) repeatedly (e.g., periodically) provisions 712 new dark transaction servers in a pool 706 of dark transaction servers (FIG. 8, operation 802). The control server 702 may, for example, provision one dark transaction server and add it to the pool 706 of dark transaction servers at one time, and then, at a later time, provision another dark transaction server and add it to the pool 706 of dark transaction servers. The control server 702 may repeat this process for any amount of time and for any number of dark transaction servers. The control server 702 may provision each of the dark transaction servers in the pool 706 as "dark" (unaddressable) transaction servers in any of the ways disclosed herein.

An initiator 704 (which may be the same as or similar to the initiator 104 of FIG. 1) requests 714 a transaction server, such as in any of the ways disclosed above in connection with FIGS. 1 and 2 (FIG. 8, operation 804).

At any time, when a new dark transaction server in the transaction server pool 706 has finished initializing and is ready to perform a transaction, that transaction server informs 716 the control server 702 that the transaction server is ready to perform a transaction (FIG. 8, operation 806). Note that the newly-initiated transaction server may inform the control server 702 of its availability at any time, such as before and/or after the initiator 704 requests 714 a transaction server from the control server 702.

If, at the time the control server 702 is informed by the newly-initiated transaction server that the newly-initiated transaction server is available to perform a transaction, there are no pending transaction requests at the control server 702, then the transaction server terminates 716 (FIG. 8, operation 808). The control server 702 may, for example, determine that it does not have any pending transaction requests and, in response to such a determination, terminate the transaction server or instruct the transaction server to terminate.

In response to receiving the request 714, the control server 702: (1) selects dark transaction server, if any, that is available to perform transactions, which may include waiting for a dark transaction server to become available; (2) makes the selected transaction server addressable and otherwise enables the selected transaction server for communication; and (3) informs 716 the initiator 704 of the address of the selected transaction server (FIG. 8, operation 810). The initiator 704 and the selected transaction

server (which is no longer "dark") may then communicate with each other, and the transaction server may perform one or more transactions on behalf of the initiator 704 in any of the ways disclosed herein.

By staggering the number of servers being provisioned continuously over time in the system 700 of FIG. 7, any delay between a service request 714 and the availability of a server for servicing that request can be eliminated, while maintaining the security benefits of the transient transaction servers disclosed herein.

Embodiments of the present invention disclosed herein may be combined with each other in various ways. For example, the system 100 and method of FIGS. 1 and 2, respectively, may be combined with the systems and methods of FIGS. 3-8 in various ways, as will be apparent to those having ordinary skill in the art.

Embodiments of the present invention may be used in a variety of applications, such as the following, which are merely illustrative and not exhaustive.

Push of Software Update or Patch from an Original Equipment Manufacturer (OEM). A transaction server 108 may be used by a smart machine OEM 104, or an authorized third party, to authenticate and transmit software updates to remote smart machines 116n.

Machines with microprocessors, programmable logic controls (PLCs), supervisory control and data acquisition (SCADA) systems and distributed control systems (DCS), that are connected to the internet, known as smart machines 116n, need periodic software updates. These smart machines often lack systems for secure communication or a system of authenticating the sender or software package, and they may receive unauthorized software updates from bad actors that seek to secure unauthorized control over the smart machine or install damaging software.

Embodiments of the present invention may be used to install a software update on the diaspora of their remote smart machines 116n. The OEM acts as an Initiator 108 to establish a transaction server 108 by sending request 106 to a Control Server 102 to cause the transaction server 108 to be created using specific parameters. In this case, the Transaction Server 108 parameters may be or include authentication information. The Transaction Server 108, once created, authenticates the OEM 104, and the software is transferred 118 from the OEM 104 to the transaction server 108. The authenticated

software is then transmitted 120n from the transaction server 108 to the smart machines 116n.

Advantages of the software update/patch embodiment described above include the following: (1) protects against installation of malicious software; (2) scalable—unlimited numbers of transaction servers may be created; (3) unlimited smart machines may be updated from the single or multiple transactional servers; and (4) redundancy—transaction servers may be created with redundant updates to ensure updates communicate with intended devices on various platforms or in various environments or via various transmission methods.

Transmission of Health Records. Embodiments of the present invention may use a Transaction Server 108 to mitigate the likelihood and magnitude of risk of unauthorized access and transfer of digital health records from a generator 116a of the records to another authorized participant 116b. The Initiator 104 may be the medical group that generated the records, the patient, an insurance company or another medical facility that needs the records.

In this example, the Initiator 104 may be the insurance company for the insured that needs an insured patient's medical records, and the Authorized Participant 116a may be the medical group that generated the records. The insurance company (Initiator 104) contacts 106 a Control Server 102 to establish 110 a Transaction Server 108 with specified parameters. These parameters may be authentication, cybersecurity, and data record keeping and reporting requirements in accordance with insurance and HIPPA or other health regulatory and industry requirements. The Transaction Server 108 notifies (118 and 120a) the Initiator 104 and the medical group or Authorized Participant 116a that it has been created in accordance with the specified parameters, the medical group 116a transmits 120a the health records to the Transaction Server 108. The Transaction Server 108 then forwards 118 the health records to the insurance company 104. The Transaction Server 108 is then terminated 212. All communication and data transfer is protected with encryption keys.

Advantages of the transmission of health records embodiment described above include: (1) prevents a man-in-the-middle attack where a bad actor targets the records during transit; (2) eliminates direct repeated continuous communication links vulnerabilities; (3) establishes a foundation for big data storage in an encrypted format;

and (4) significantly reduces risk by limiting the records available on the transaction server to only those needed for the specific transaction being executed.

Transient Asset Escrow and Exchange. A Transaction Server 108 may be used to securely exchange digital assets for trusts and banking companies, digital exchange operators, title companies, and securities exchanges, brokerages or clearing agencies.

As an illustrative example, a digital exchange operator, the Initiator 104, identifies two traders or Authorized Participants 116a and 116b, that wish to exchange digital assets (via trade order matching). The digital assets may be crypto currencies such as Bitcoin or Litecoin, digital fiat currency, digital deeds, or any other digital asset of value. It is not necessary for the digital exchange to take custody of either the digital assets or the asset's private cryptographic keys. The assets remain with the two owners until the trade order match is made and the transaction is ready to proceed.

When Authorized Participant 116a and Authorized Participant 116b are ready to exchange assets, the digital exchange operator or Initiator 104 communicates 118 to the Control Server 102, which creates 110 the Transaction Server 108 and escrow. The Transaction Server 108 confirms it is established according to Initiator's 104 specified parameters by communicating (112) appropriate messages to Control Server 102, Initiator 104 and the traders or Authorized Participants (116a and 116b). Each Authorized Participant (116a and 116b) send their assets to the escrow and Transaction Server 108. The Transaction Server 108 will confirm the details and authenticate the parties and the transaction according to the specified parameters. Once custody of the assets is confirmed on relevant blockchains for the specified period (of blocks, based on the level of transaction surety or the settlement risk specified by the Control Server 102), the Transaction Server 108 will transfer (120a and 120b) the assets to the respective recipients, Authorized Participants (116a and 116b). The Transaction Server 108 may or may not confirm that the intended recipients have received the new digital assets, according to the parameters. Once the transaction is complete, the Transaction Server 108 transmits the information that the Control Server 102 has specified in the parameters, and then terminates 212. The Transaction Server 108 may control one transaction or multiple transactions during its existence.

Advantages of the transient asset escrow and exchange embodiment described above include: (1) the system establishes a true escrow; (2) bad actors do not have enough

time to locate, target and penetrate the Transaction Server, because the Transaction Server exists only for a brief period of time; (3) bad actors do not know the location or address of the server hosting the Transaction Server, because the Transaction Server is created on a random outside server; (4) because the exchange Operator does not have access to Parties' private keys, no assets are aggregated, and phishing the digital exchange Operator does not provide access to a pool of aggregated deposits, as it currently does in exchange Operator systems; and (5) reduces reserve currency requirements and cyber security costs for digital exchange operator.

It is to be understood that although the invention has been described above in terms of particular embodiments, the foregoing embodiments are provided as illustrative only, and do not limit or define the scope of the invention. Various other embodiments, including but not limited to the following, are also within the scope of the claims. For example, elements and components described herein may be further divided into additional components or joined together to form fewer components for performing the same functions.

Any of the functions disclosed herein may be implemented using means for performing those functions. Such means include, but are not limited to, any of the components disclosed herein, such as the computer-related components described below.

The techniques described above may be implemented, for example, in hardware, one or more computer programs tangibly stored on one or more computer-readable media, firmware, or any combination thereof. The techniques described above may be implemented in one or more computer programs executing on (or executable by) a programmable computer including any combination of any number of the following: a processor, a storage medium readable and/or writable by the processor (including, for example, volatile and non-volatile memory and/or storage elements), an input device, and an output device. Program code may be applied to input entered using the input device to perform the functions described and to generate output using the output device.

Embodiments of the present invention include features which are only possible and/or feasible to implement with the use of one or more computers, computer processors, and/or other elements of a computer system. Such features are either impossible or impractical to implement mentally and/or manually. For example, embodiments of the present invention include computer systems which include a variety

of components, such as the control server 102 and transaction server 108, which are themselves computer systems, and which communicate with each other over digital communication networks, such as the Internet. Embodiments of the present invention, therefore, are directed to improvements to computer technology.

Any claims herein which affirmatively require a computer, a processor, a memory, or similar computer-related elements, are intended to require such elements, and should not be interpreted as if such elements are not present in or required by such claims. Such claims are not intended, and should not be interpreted, to cover methods and/or systems which lack the recited computer-related elements. For example, any method claim herein which recites that the claimed method is performed by a computer, a processor, a memory, and/or similar computer-related element, is intended to, and should only be interpreted to, encompass methods which are performed by the recited computer-related element(s). Such a method claim should not be interpreted, for example, to encompass a method that is performed mentally or by hand (e.g., using pencil and paper). Similarly, any product claim herein which recites that the claimed product includes a computer, a processor, a memory, and/or similar computer-related element, is intended to, and should only be interpreted to, encompass products which include the recited computer-related element(s). Such a product claim should not be interpreted, for example, to encompass a product that does not include the recited computer-related element(s).

Each computer program within the scope of the claims below may be implemented in any programming language, such as assembly language, machine language, a high-level procedural programming language, or an object-oriented programming language. The programming language may, for example, be a compiled or interpreted programming language.

Each such computer program may be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a computer processor. Method steps of the invention may be performed by one or more computer processors executing a program tangibly embodied on a computer-readable medium to perform functions of the invention by operating on input and generating output. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, the processor receives (reads) instructions and data from a memory (such as a read-only memory and/or a random access memory) and writes

(stores) instructions and data to the memory. Storage devices suitable for tangibly embodying computer program instructions and data include, for example, all forms of non-volatile memory, such as semiconductor memory devices, including EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROMs. Any of the foregoing may be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits) or FPGAs (Field-Programmable Gate Arrays). A computer can generally also receive (read) programs and data from, and write (store) programs and data to, a non-transitory computer-readable storage medium such as an internal disk (not shown) or a removable disk. These elements will also be found in a conventional desktop or workstation computer as well as other computers suitable for executing computer programs implementing the methods described herein, which may be used in conjunction with any digital print engine or marking engine, display monitor, or other raster output device capable of producing color or gray scale pixels on paper, film, display screen, or other output medium.

Any data disclosed herein may be implemented, for example, in one or more data structures tangibly stored on a non-transitory computer-readable medium. Embodiments of the invention may store such data in such data structure(s) and read such data from such data structure(s).

What is claimed is:

CLAIMS

1. A method performed by at least one computer processor executing computer program instructions tangibly stored on at least one non-transitory computer-readable medium, the method comprising:

(A) receiving, over a network at a control server, from an initiator, a request to execute a transaction;

(B) creating a transaction server based on the request;

(C) providing the initiator with information about the transaction server, including an address of the transaction server; and

(D) using the transaction server to execute the transaction independently of the control server.

2. The method of claim 1, wherein (C) comprises transmitting a message, containing the information about the transaction server, to the initiator over a network.

3. The method of claim 1, further comprising:

(E) in response to determining that the transaction has completed, terminating the transaction server.

4. The method of claim 1, further comprising:

(E) in response to determining that a timeout period has expired, terminating the transaction server.

5. The method of claim 1, further comprising:

(E) in response to determining that the transaction satisfies an anomalous condition, terminating the transaction server.

6. The method of claim 1, wherein using the transaction server to execute the transaction independently of the control server comprises executing the transaction without communicating with the control server.

7. The method of claim 1, wherein the control server has a first IP address and the transaction server has a second IP address, wherein the first IP address differs from the second IP address.

8. The method of claim 1, wherein (B) comprises identifying a type of the transaction and creating the transaction server to be capable of executing a transaction of the identified type.

9. The method of claim 1, wherein (B) comprises identifying a property specified by the request and creating the transaction server to have the identified property.

10. A system comprising at least one non-transitory computer-readable medium, the at least one non-transitory computer-readable medium comprising computer program instructions executable by at least one computer processor to execute a method, the method comprising:

(A) receiving, over a network at a control server, from an initiator, a request to execute a transaction;

(B) creating a transaction server based on the request;

(C) providing the initiator with information about the transaction server, including an address of the transaction server; and

(D) using the transaction server to execute the transaction independently of the control server.

11. The system of claim 10, wherein (C) comprises transmitting a message, containing the information about the transaction server, to the initiator over a network.

12. The system of claim 10, wherein the method further comprises:

(E) in response to determining that the transaction has completed, terminating the transaction server.

13. The system of claim 10, wherein the method further comprises:

(E) in response to determining that a timeout period has expired, terminating the transaction server.

14. The system of claim 10, wherein the method further comprises:

(E) in response to determining that the transaction satisfies an anomalous condition, terminating the transaction server.

15. The system of claim 10, wherein using the transaction server to execute the transaction independently of the control server comprises executing the transaction without communicating with the control server.

16. The system of claim 10, wherein the control server has a first IP address and the transaction server has a second IP address, wherein the first IP address differs from the second IP address.

17. The system of claim 10, wherein (B) comprises identifying a type of the transaction and creating the transaction server to be capable of executing a transaction of the identified type.

18. The system of claim 10, wherein (B) comprises identifying a property specified by the request and creating the transaction server to have the identified property.

19. A computer-implemented method for use with a transaction server, the transaction server being unaddressable over a network, the method comprising:

(A) receiving, over a network at a control server, from an initiator, a request to execute a transaction;

(B) making the transaction server addressable in response to the request;

(C) providing the initiator with information about the transaction server, including an address of the transaction server; and

(D) using the transaction server to execute the transaction independently of the control server.

20. The method of claim 19, wherein (B) comprises:

at the control server, making the transaction server addressable in response to the request.

21. The method of claim 19, further comprising:

(E) assigning an IP address to the transaction server in response to the request.

22. The method of claim 19, further comprising:

(E) after (B), provisioning a new transaction server;

(F) making the new transaction server unaddressable over the network.

23. The method of claim 19, further comprising:

(E) in response to determining that a server termination criterion has been satisfied, terminating the transaction server.

24. The method of claim 19, further comprising:

(E) before (B), selecting the transaction server from a plurality of transaction servers.

25. A system comprising at least one non-transitory computer-readable medium, the at least one non-transitory computer-readable medium comprising computer program instructions executable by at least one computer processor to execute a method, the method comprising:

(A) receiving, over a network at a control server, from an initiator, a request to execute a transaction;

(B) making the transaction server addressable in response to the request;

(C) providing the initiator with information about the transaction server, including an address of the transaction server; and

(D) using the transaction server to execute the transaction independently of the control server.

26. The system of claim 25, wherein (B) comprises:

at the control server, making the transaction server addressable in response to the request.

27. The system of claim 25, wherein the method further comprises:

(E) assigning an IP address to the transaction server in response to the request.

28. The system of claim 25, wherein the method further comprises:

(E) after (B), provisioning a new transaction server;

(F) making the new transaction server unaddressable over the network.

29. The system of claim 25, wherein the method further comprises:

(E) in response to determining that a server termination criterion has been satisfied, terminating the transaction server.

30. The system of claim 25, wherein the method further comprises:

(E) before (B), selecting the transaction server from a plurality of transaction servers.

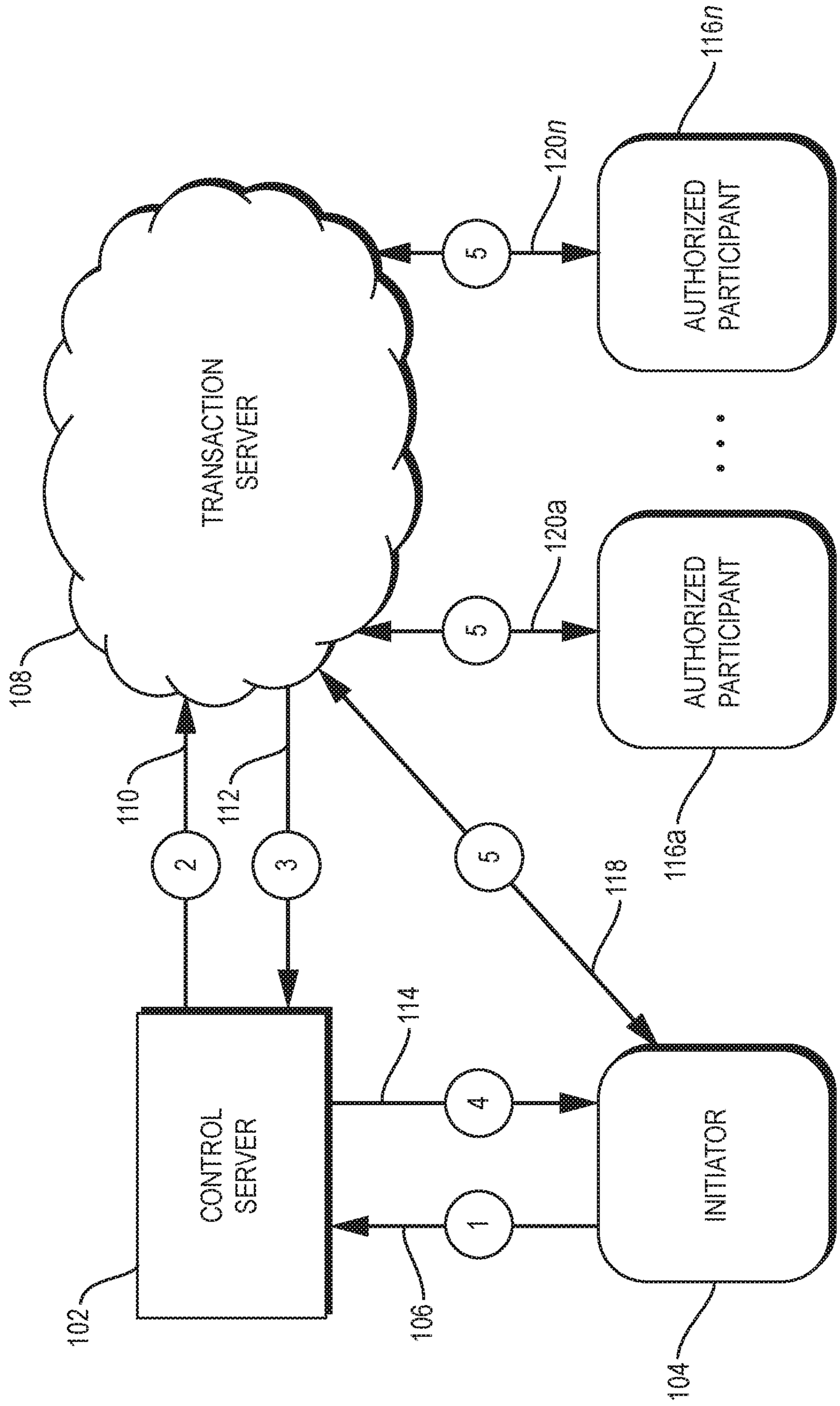
100
TRANSACTION SERVER
108
120n
AUTHORIZED PARTICIPANT
116n
120a
AUTHORIZED PARTICIPANT
116a
110
112
118
CONTROL SERVER
114
INITIATOR
102
106
104
1
2
3
4
5

FIG. 1

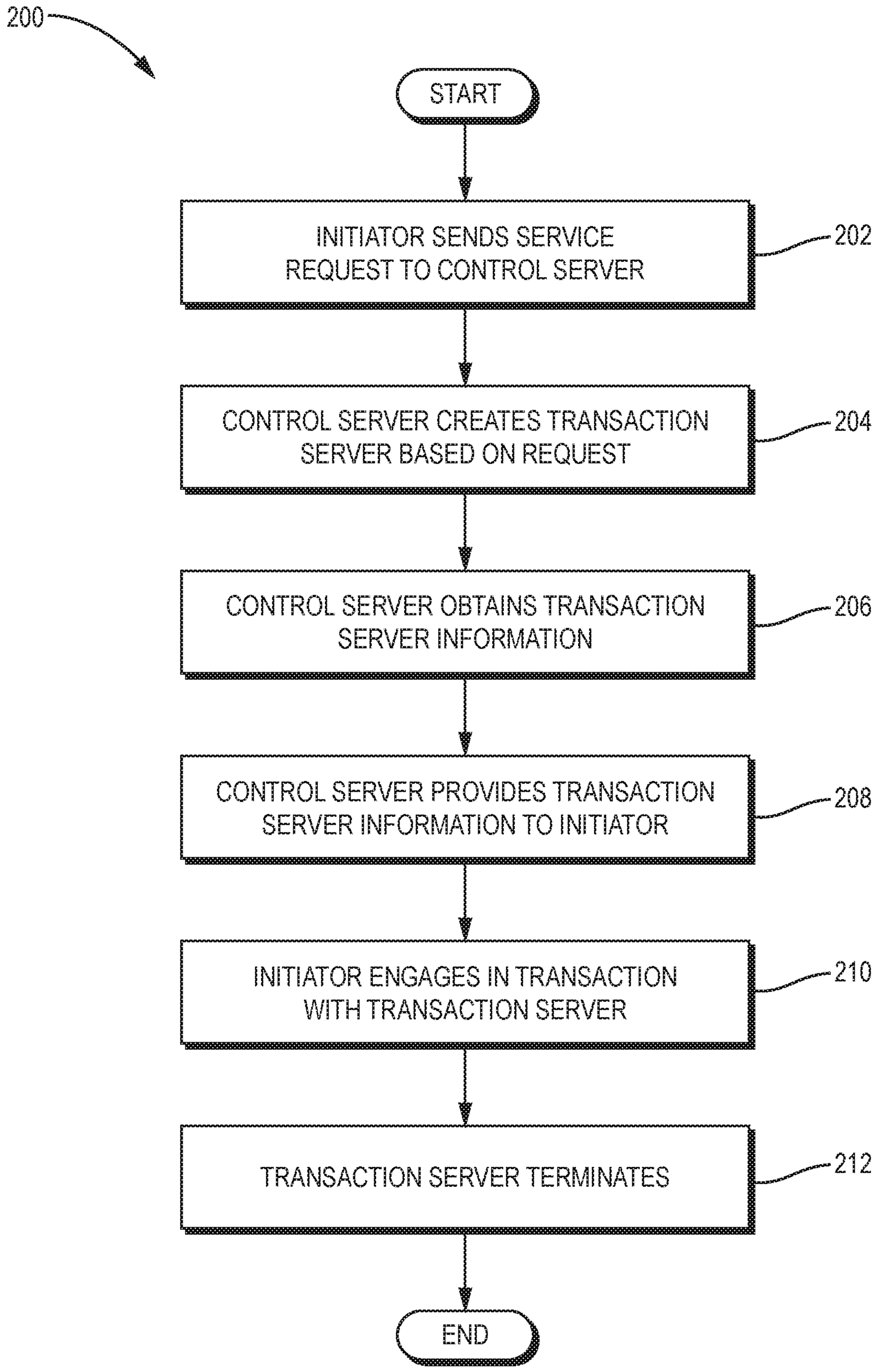
200
START
INITIATOR SENDS SERVICE REQUEST TO CONTROL SERVER
202
CONTROL SERVER CREATES TRANSACTION SERVER BASED ON REQUEST
204
CONTROL SERVER OBTAINS TRANSACTION SERVER INFORMATION
206
CONTROL SERVER PROVIDES TRANSACTION SERVER INFORMATION TO INITIATOR
208
INITIATOR ENGAGES IN TRANSACTION WITH TRANSACTION SERVER
210
TRANSACTION SERVER TERMINATES
212
END

FIG. 2

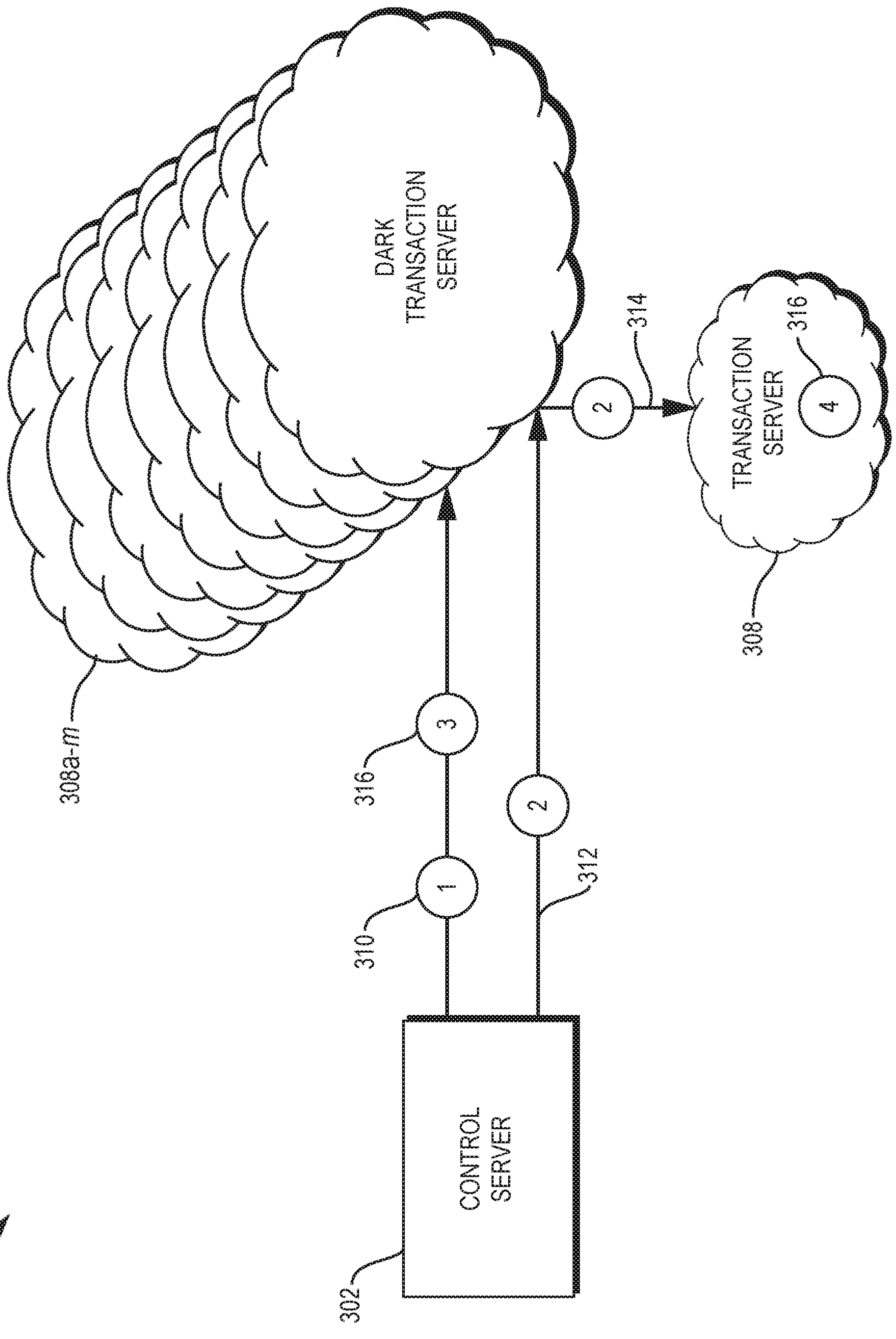
DARK
TRANSACTION
SERVER
2
314
TRANSACTION
SERVER
316
4
308
308a-m
316
3
2
310
1
312
CONTROL
SERVER
302
300

FIG. 3

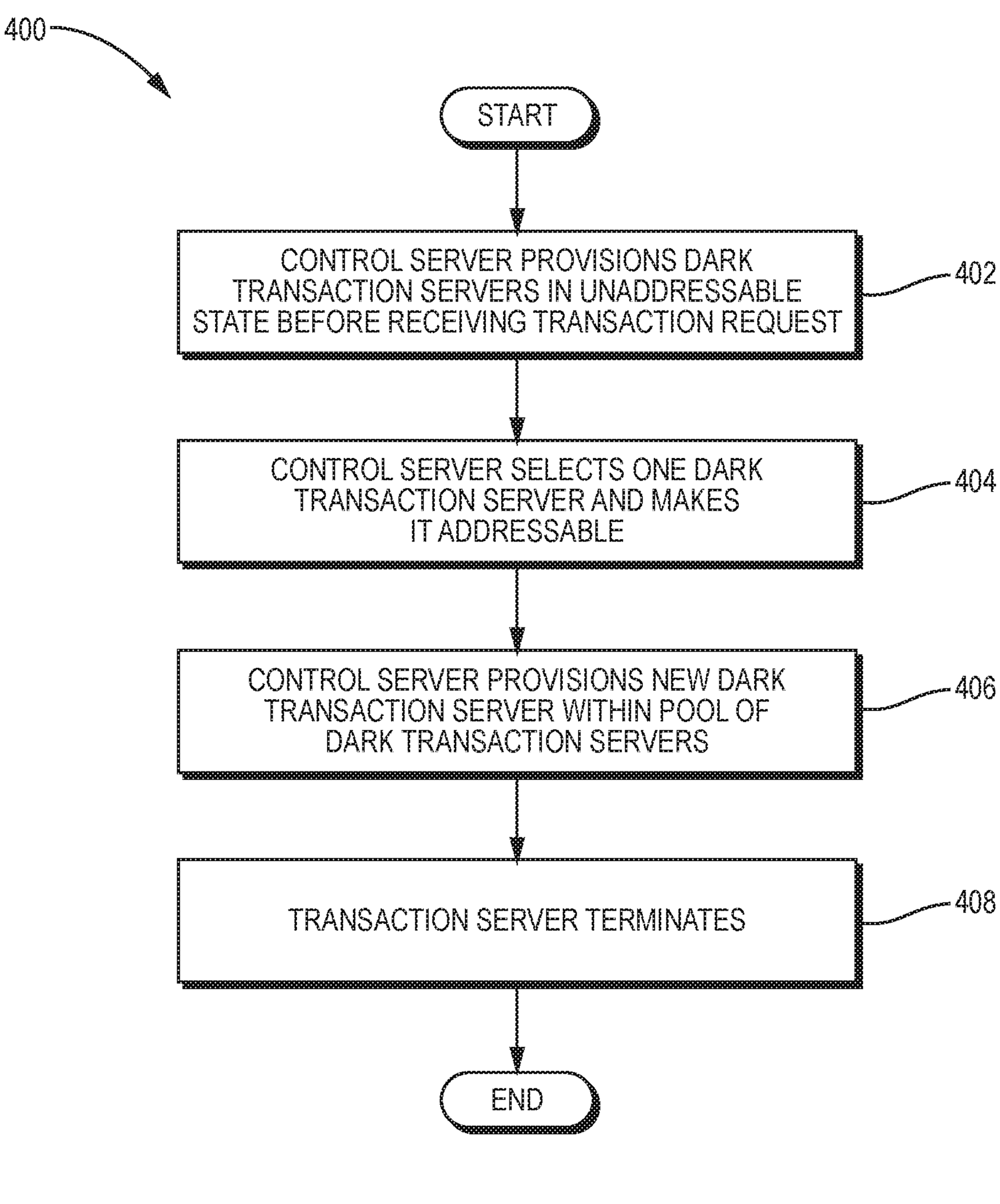
400
START
CONTROL SERVER PROVISIONS DARK TRANSACTION SERVERS IN UNADDRESSABLE STATE BEFORE RECEIVING TRANSACTION REQUEST
402
CONTROL SERVER SELECTS ONE DARK TRANSACTION SERVER AND MAKES IT ADDRESSABLE
404
CONTROL SERVER PROVISIONS NEW DARK TRANSACTION SERVER WITHIN POOL OF DARK TRANSACTION SERVERS
406
TRANSACTION SERVER TERMINATES
408
END

FIG. 4

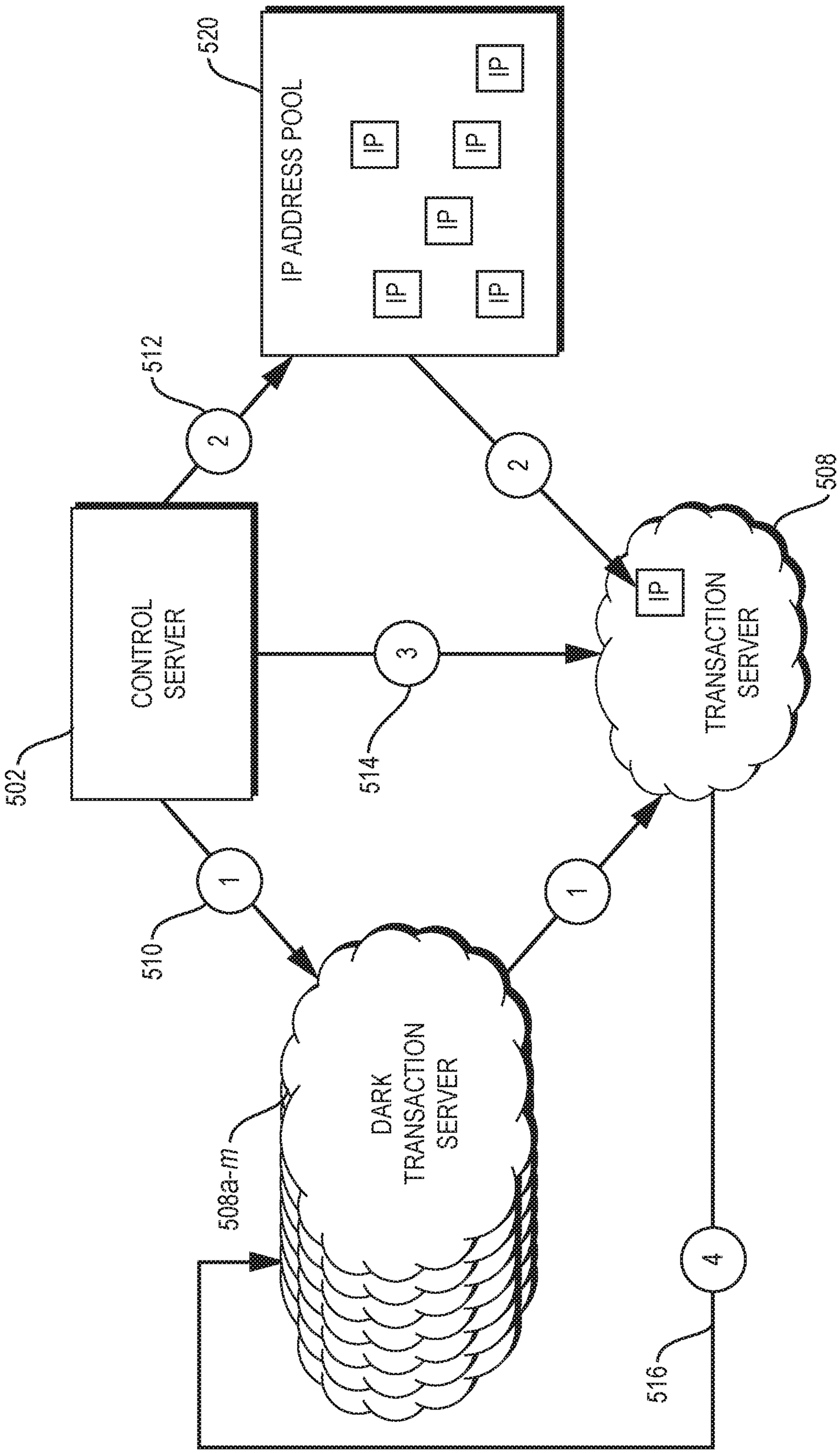
500
502
CONTROL SERVER
504
IP ADDRESS POOL
IP
506
DARK TRANSACTION SERVER
508
TRANSACTION SERVER
508a-m
510
512
514
516
1
2
3
4

FIG. 5

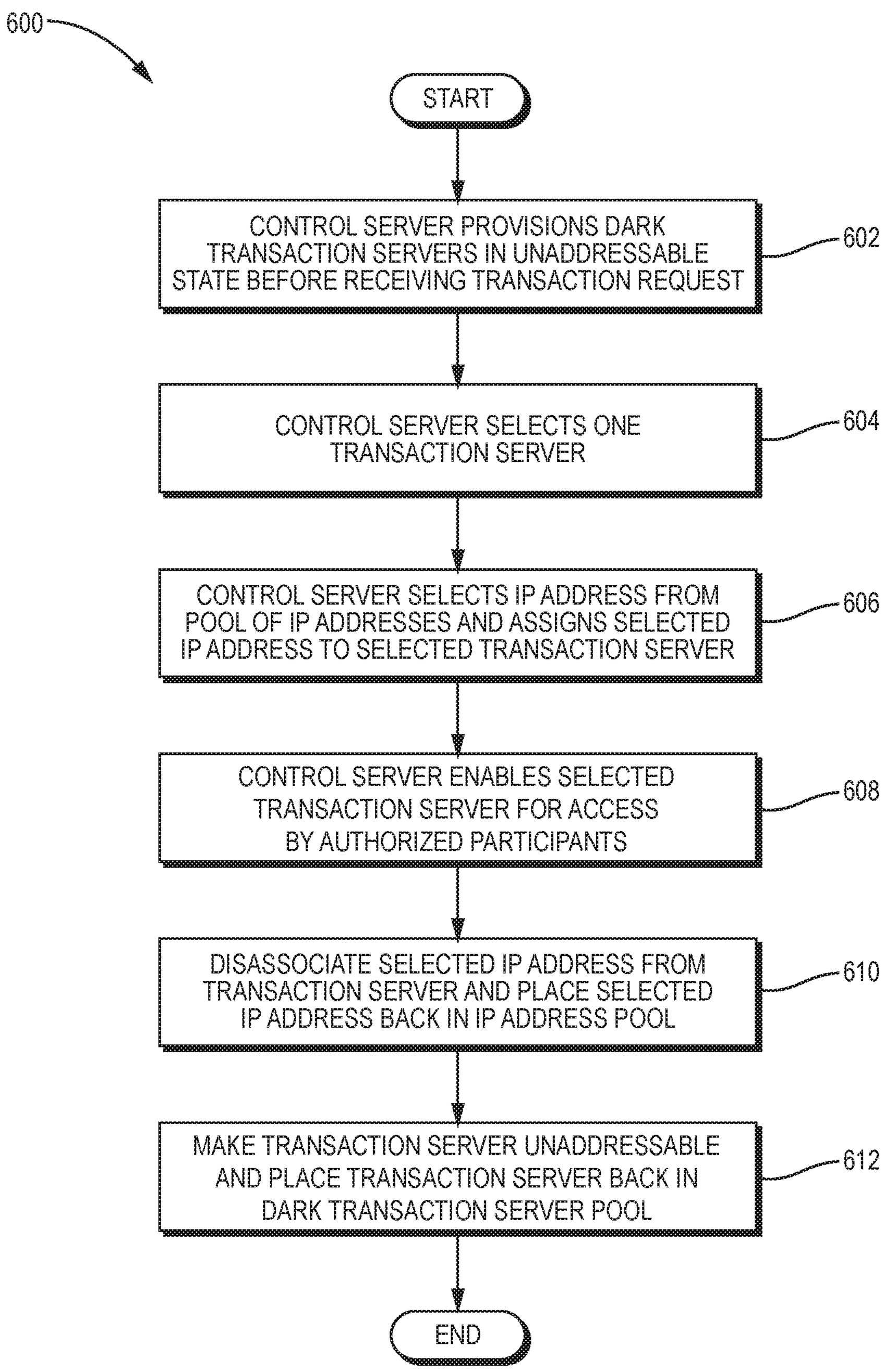
600
START
CONTROL SERVER PROVISIONS DARK TRANSACTION SERVERS IN UNADDRESSABLE STATE BEFORE RECEIVING TRANSACTION REQUEST
602
CONTROL SERVER SELECTS ONE TRANSACTION SERVER
604
CONTROL SERVER SELECTS IP ADDRESS FROM POOL OF IP ADDRESSES AND ASSIGNS SELECTED IP ADDRESS TO SELECTED TRANSACTION SERVER
606
CONTROL SERVER ENABLES SELECTED TRANSACTION SERVER FOR ACCESS BY AUTHORIZED PARTICIPANTS
608
DISASSOCIATE SELECTED IP ADDRESS FROM TRANSACTION SERVER AND PLACE SELECTED IP ADDRESS BACK IN IP ADDRESS POOL
610
MAKE TRANSACTION SERVER UNADDRESSABLE AND PLACE TRANSACTION SERVER BACK IN DARK TRANSACTION SERVER POOL
612
END

FIG. 6

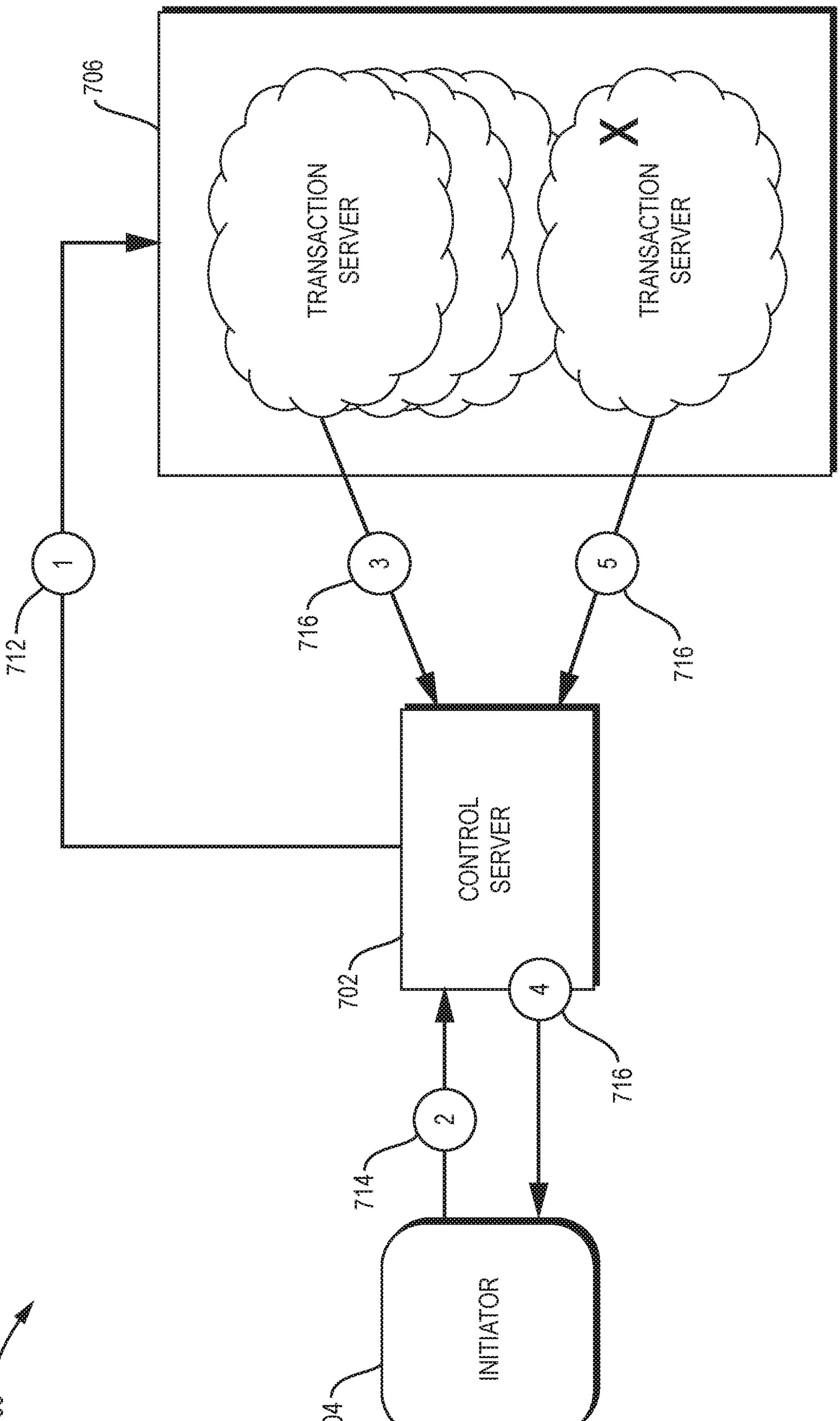
700
706
TRANSACTION SERVER
TRANSACTION SERVER
712
716
716
CONTROL SERVER
702
716
714
INITIATOR
704
1
2
3
4
5

FIG. 7

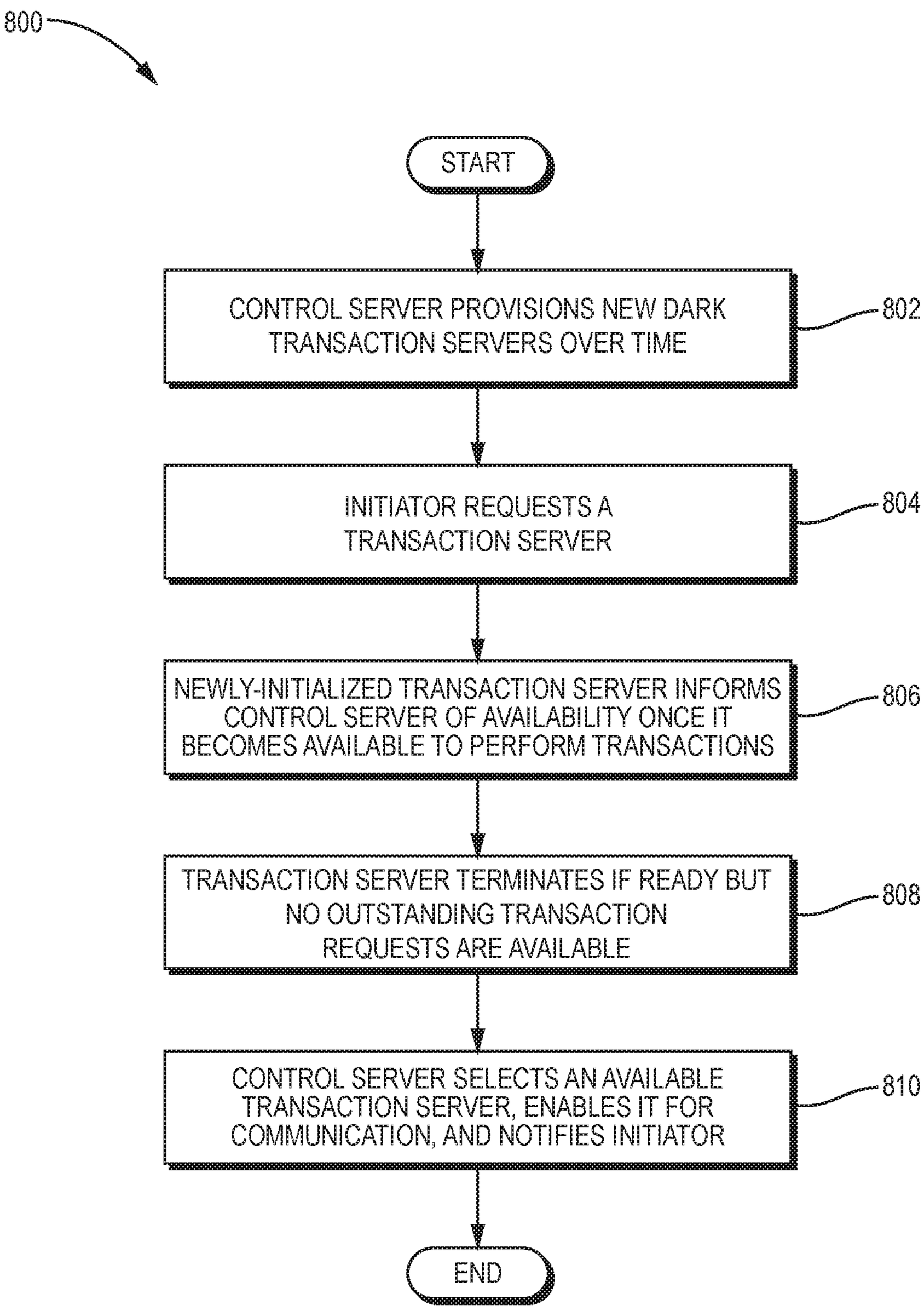
800
START
CONTROL SERVER PROVISIONS NEW DARK TRANSACTION SERVERS OVER TIME
802
INITIATOR REQUESTS A TRANSACTION SERVER
804
NEWLY-INITIALIZED TRANSACTION SERVER INFORMS CONTROL SERVER OF AVAILABILITY ONCE IT BECOMES AVAILABLE TO PERFORM TRANSACTIONS
806
TRANSACTION SERVER TERMINATES IF READY BUT NO OUTSTANDING TRANSACTION REQUESTS ARE AVAILABLE
808
CONTROL SERVER SELECTS AN AVAILABLE TRANSACTION SERVER, ENABLES IT FOR COMMUNICATION, AND NOTIFIES INITIATOR
810
END

FIG. 8

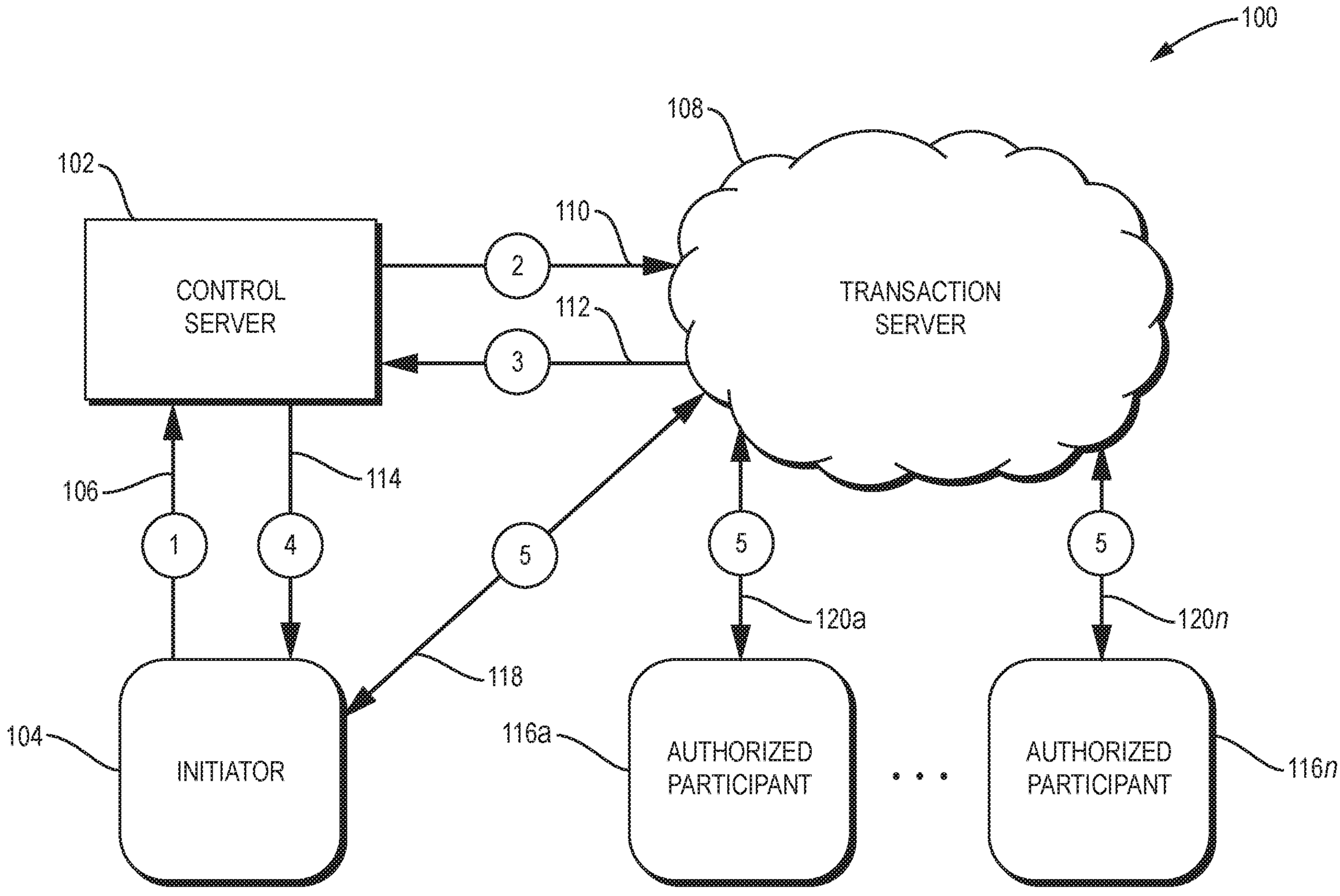
100
102
CONTROL SERVER
108
TRANSACTION SERVER
110
112
2
3
106
114
1
4
5
5
5
118
120a
120n
104
INITIATOR
116a
AUTHORIZED PARTICIPANT
AUTHORIZED PARTICIPANT
116n

FIG. 1