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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 January 2022 (13.01.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/010339 A1

(51) International Patent Classification:
H04L 12/24 (2006.01)

(21) International Application Number:
PCT/MY2020/050187

(22) International Filing Date:
02 December 2020 (02.12.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI 2020003512 06 July 2020 (06.07.2020) MY

(71) Applicant: **MIMOS BERHAD** [MY/MY]; Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(72) Inventors: **ISMAIL, Ahmad Zuhairi**; MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **ANBARASAN, Durairaj**; MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **ARIFFIN, Nizamuddin**; MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **HAMZAH, Siti Hajar**; MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **GOON, Wooi Kin**; MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(74) Agent: **AWANG, Muhammad Irfan Mustaqim**; Patentsworth International Sdn Bhd, A-SG-11A, Sunway GEO Avenue, Jalan Lagoon Selatan, Sunway South Quay, Sunway City, Subang Jaya, 47500 (MY).

(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, IT, JO, JP, KE, KG, KH, KN,

(54) Title: SYSTEM AND METHOD FOR SEAMLESS PROVISION, CONFIGURATION, AND DEPLOYMENT OF ENTERPRISE-GRADE PRIVATE BLOCKCHAIN NETWORK

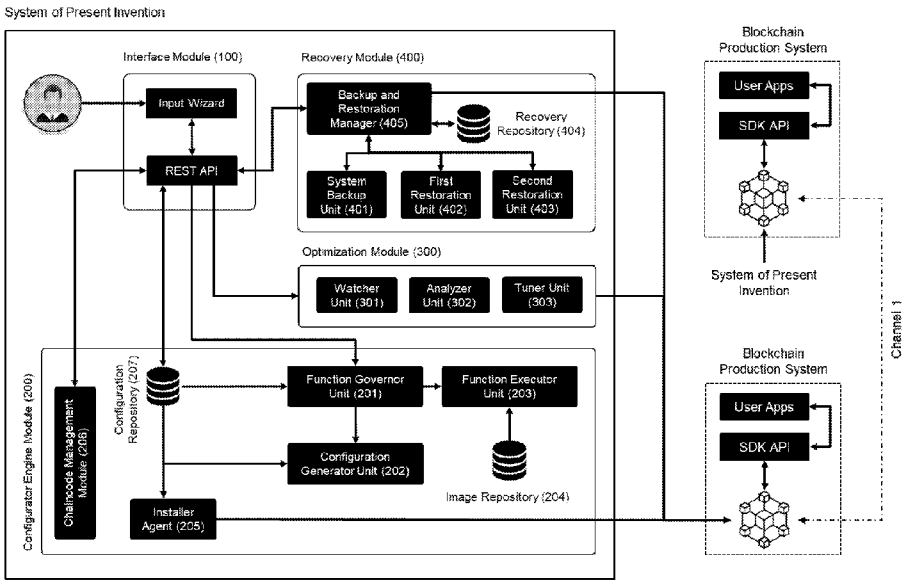


FIGURE 2

(57) Abstract: The present invention discloses a system and a method for seamlessly provisioning, configuring, and deploying an enterprise-grade private blockchain network. The system comprises an interface module, a configurator engine module, an optimization module, and a recovery module. The configurator engine module comprising a function governor unit, a configuration generator unit, and a function executor unit processes a user-specified requirement to generate a full network configuration for the private blockchain network. The optimization module comprising a watcher unit, an analyzer unit, and a tuner unit dynamically optimizes the private blockchain network by way of monitoring the full network configuration generated thereof. The recovery module comprising a system backup unit, a first restoration unit, and a second restoration unit performs a network recovery operation of the said private blockchain

[Continued on next page]

WO 2022/010339 A1

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

SYSTEM AND METHOD FOR SEAMLESS PROVISION, CONFIGURATION, AND DEPLOYMENT OF ENTERPRISE-GRADE PRIVATE BLOCKCHAIN NETWORK

FIELD OF THE INVENTION

The present invention generally relates to private blockchain networks. More particularly, the present invention relates to a system for seamlessly provisioning, configuring, and deploying an enterprise-grade private blockchain network.

BACKGROUND OF THE INVENTION

Blockchain is the underlying technology that powers cryptocurrency. Because blockchain is decentralized, it manages to be both transparent and secure. The two most common types of blockchains are private blockchains and public blockchains. The primary difference between public and private blockchains is the level of access participants are granted. Unlike public blockchains that are entirely open, private blockchains are a restrictive, permissioned blockchain that only operates within a closed network. It is the private property of an organization or an individual. It comprises specific constituent nodes on the network that are given the rights to create, view, and authorize transactions. A blockchain "network" operates as a decentralized database that consists of a network of computers or devices that come to an agreement on a single version of a shared ledger. In a permissioned chain like private blockchain network, only a few trusted nodes would operate a network between organizations.

Enterprise-grade blockchain is a blockchain network that can be integrated and utilized for enterprise-level purposes. It is specially equipped to deal with enterprise-level organizational demands. The current implementation of enterprise blockchain, however, requires a robust blockchain network setup with redundancy infrastructure. For an organization to utilize this blockchain, they would need at least one paid staff with advanced blockchain knowledge and experience, or they would have to invest in public blockchain cloud with the commitment of monthly or annual subscription fees. Apart from being time-consuming and complicated, the existing blockchain network setup still suffers from the problems of human error associated with manual configuration, complexity and massive efforts for multi-hosts, the

requirement of Internet connection for performing installation, and complexity of data recovery related to the private blockchain network.

By way of background, International PCT Patent Application Publication No. WO 2019/010064 A1 discloses an event interface system that facilitates the creation and deployment of a first blockchain object and a second blockchain object on a first blockchain and a second blockchain, respectively. The system, according to the '064 publication, also provides an interface between the first blockchain object and the second blockchain object via the event hub. Additionally, the system can allow interaction between blockchain objects on a private blockchain and a participant on the system.

It would, therefore, be advantageous to provide a solution that would overcome the deficiencies and shortcomings of prior art by way of providing a system and a method for seamlessly, efficiently provisioning, configuring, and deploying the enterprise-grade private blockchain network. Although there are systems and methods for the same in the prior art, for many practical purposes, there is still considerable room for improvement.

SUMMARY OF THE INVENTION

The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

Accordingly, the present invention provides a system for seamlessly provisioning, configuring, and deploying an enterprise-grade private blockchain network.

The system of the present invention may be characterized by an interface module configured for receiving a user-specified requirement having a system specification of a private blockchain network; a configurator engine module connected to the interface module configured for processing the user-specified requirement to generate a full network configuration for the said private blockchain network, comprising a function governor unit for generating a list of execution

functions in a sequential order, configuration files and filenames derivable based on the said system specification, a configuration generator unit for executing the configuration files, and a function executor unit connected to an image repository for executing the list of execution functions in the said sequential order; an optimization module connected to the interface module and the configurator engine module configured for dynamically optimizing the private blockchain network by way of monitoring the full network configuration generated thereof, comprising a watcher unit for monitoring a list of blockchain hosts and services associated with the full network configuration, an analyzer unit for identifying an irregularity in the list of blockchain hosts and services thereof, and a tuner unit for correcting the irregularity detected thereof; and a recovery module connected to the interface module, the configurator engine module and the optimization module configured for performing a network recovery operation of the said private blockchain network, comprising a system backup unit for providing a snapshot backup having network components of the said private blockchain network, a first restoration unit for providing a list of available backup sets retrievable from a recovery repository based on metadata associated thereof, and a second restoration unit for providing a restoration of a single network component selected from the said network components.

Preferably, the configurator engine module further comprises an installer agent deployed on a remote node for performing installation of the private blockchain network in a different network domain and a chaincode management module.

Preferably, the configurator engine module further comprises a configuration repository for storing the user-specified requirement, the full network configuration comprising the list of execution functions, the configuration files, and the filenames, and the list of blockchain hosts and services thereof.

Preferably, the recovery module comprises a backup and restoration manager to manage the system backup unit, the first restoration unit, the second restoration unit, and the recovery repository thereof.

Preferably, the user-specified requirement includes a quick-start module for a system default private blockchain network and a customized module for a customized private blockchain network.

Preferably, the system specification comprises network attributes selected from a group comprising network name, number of virtual machines, number of hosts, username, password, port number, number of organizations, domain name, and organization name.

Preferably, the irregularity includes any error in the list of blockchain hosts and services monitored thereof.

Preferably, the list of blockchain hosts and services comprises node availability, host availability, utilization parameters, service health (up and down), number of ports, and service logs (errors and warnings).

Preferably, the network components include peer nodes, orderer nodes, certificate authority nodes, and middleware nodes.

In accordance with another aspect of the present invention, there is provided a method for seamlessly provisioning, configuring, and deploying an enterprise-grade private blockchain network.

The method of the present invention may be characterized by the steps of receiving a user-specified requirement including a system specification of a private blockchain network; processing the user-specified requirement to generate a full network configuration for the said private blockchain network, including generating a list of execution functions in a sequential order, configuration files and filenames derivable based on the said system specification; executing the configuration files; and executing the list of execution functions in the said sequential order; dynamically optimizing the private blockchain network by way of monitoring the full network configuration generated thereof, including monitoring a list of blockchain hosts and services associated with the full network configuration; identifying an irregularity in the list of blockchain hosts and services thereof; and correcting the irregularity detected thereof; and performing a network recovery operation of the said private blockchain network, including providing a snapshot backup having network components of the said private blockchain network; providing a list of available backup sets retrievable from a recovery repository based on metadata associated thereof; and providing a restoration of a single network component selected from the said network components.

Advantageously, the present invention enables an organization to deploy its own enterprise grade private blockchain network in a highly specific, cost-effective, quick and simple manner, without the necessity of highly-knowledge and trained personnel and the use of complicated and sophisticated parts. The present invention deploys the said private blockchain network in a private server and may remain effective and live without the commitment of paying subscription fees monthly or annually.

The system and the method of the present invention also, advantageously, permit data recovery of the private blockchain network in various ways depending on the user's choice. The present invention further monitors, analyzes and tunes the said private blockchain network automatically and dynamically, without disrupting the ongoing network and requiring extensive logistic efforts. The present invention can reduce cycle time and human errors and is capable of scalability to any user requirement. It is a further advantage of the present invention that the installation of private blockchain network works in an offline mode, i.e. without the Internet connection.

The foregoing and other objects, features, aspects and advantages of the present invention will become better understood from a careful reading of a detailed description provided herein below with appropriate reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the invention and many of the attendant advantages thereof will be readily as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

Figure 1 is a schematic diagram of a system for seamlessly provisioning, configuring and deploying an enterprise grade private blockchain network according to one embodiment of the present invention;

Figure 2 is a schematic diagram depicting a high level architecture of the system of Figure 1 according to one embodiment of the present invention;

Figure 3 is a flow diagram of a method for seamlessly provisioning, configuring and deploying an enterprise grade private blockchain network according to one embodiment of the present invention;

Figure 4 is a flow diagram of a step of processing the user-specified requirement to generate a full network configuration for the said private blockchain network as stated in the method of Figure 3 according to one embodiment of the present invention;

Figure 5 is a flow diagram of a step of dynamically optimizing the private blockchain network by way of monitoring the full network configuration generated thereof as stated in the method of Figure 3 according to one embodiment of the present invention;

Figure 6 is a flow diagram of a step of performing a network recovery operation of the said private blockchain network as stated in the method of Figure 3 according to one embodiment of the present invention;

Figure 7 is a flow diagram representing a process flow involved in an interface module which is related to a step of receiving a user-specified requirement including a system specification of a private blockchain network according to one embodiment of the present invention;

Figure 8 is a flow diagram representing a process flow involved in a configurator engine module which is related to the step of Figure 4 according to one embodiment of the present invention;

Figure 9 is a flow diagram representing a process flow involved in a function governor unit according to one embodiment of the present invention;

Figure 10 is a flow diagram representing a process flow involved in a configurator generator unit according to one embodiment of the present invention;

Figure 11 is a flow diagram representing a process flow involved in a function executor unit according to one embodiment of the present invention;

Figure 12 is a flow diagram representing a process flow involved in an optimization module which is related to the step of Figure 5 according to one embodiment of the present invention; and

Figure 13 is a flow diagram representing a process flow involved in a recovery module which is related to the step of Figure 6 according to one embodiment of the present invention.

It is noted that the drawings may not be to scale. The drawings are intended to depict only typical aspects of the invention, and therefore should not be considered as limiting the scope of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Essentially, the present invention discloses a system and a method for seamlessly and efficiently provisioning, configuring, and deploying an enterprise-grade private blockchain network. The system of the present invention preferably connects an application ("app") owner, user, developer, operator, and the like to a blockchain production system. The blockchain production system comprises a software development kit (SDK) with an application programming interface (API). SDK is a set of tools, guidelines, and programs for developing an app for a private blockchain network. SDK is connected to the API, which is a computer interface that defines interactions between software intermediaries. Figure 1 schematically shows the system of the present invention and its connection with the app owner and the blockchain production system.

According to one preferred embodiment of the present invention, with reference to Figure 2, the system comprises an interface module **100**, a configurator engine module **200**, an optimization module **300,** and a recovery module **400**.

The interface module **100** is preferably directly connected to the blockchain production system. It is adapted for enabling communication with at least one computing device at the said blockchain production system and for setting up a communication link therefor. The interface module **100** may be configured for receiving a user-specified requirement having a system specification related to a proposed private blockchain.

In one embodiment, the user-specified requirement includes, but is not limited to, a quick-start module and a customized module. The quick-start module is preferably chosen for setting up a system default private blockchain network. The customized module is preferably chosen for setting up a customized private blockchain network. The system specification of the said user-specified requirement preferably comprises network attributes that may be selected from a group comprising network name, number of virtual machines, number of hosts, username, password, port number, number of organizations, domain name and organization name.

It is preferred that the interface module **100** collects inputs, including the user-specified requirement thereof on the proposed private blockchain network. For example, an app owner or user may choose the quick-start module as its user-specified requirement. In this regard, the quick-start module comprising system default values shall be used to create the corresponding system default private blockchain network. The quick-start module does not require any other inputs from the app owner or user. In another example, the app owner or user may choose the customized module as its user-specified requirement. The customized module allows the app owner or user to specify or select various combinations of the attributes for establishing the desired private blockchain network to suit the needs.

The interface module **100** preferably comprises a graphical user interface (GUI) with another API. Representational state transfer (REST) which is a software architectural style for distributed hypermedia systems may be adopted for use with the said API to enable a hypertext representation. The REST API may be used for bridging the inputs from the app owner or user such as the user-specified requirement and its system specification received from the GUI thereof. Among the inputs bridged therebetween are number of virtual machines, host numbers, username, password, port number (optional), number of organizations, domain name, and organization name.

The configurator engine module **200** is preferably connected to the interface module **100**. The configurator engine module **200** may be configured for automatically processing the user-specified requirement transmitted from the said interface module **100**. Upon processing, the configurator engine module **200** generates a full network configuration for the private blockchain network proposed thereof.

The configurator engine module **200** comprises a function governor unit **201**, a configuration generator unit **202**, a function executor **203**, an image repository **204**, an installer agent **205**, a chaincode management module **206**, and a configuration repository **207** which are exclusively linked with each other.

The function governor unit **201** is preferably configured for generating a list of execution functions in a sequential order, configuration files and filenames derivable based on the said system specification. In one embodiment, the function governor unit **201** governs about 18 functions for use in the configuration generator unit **202** and the function executor **203**. It receives the input from the interface module **100** and triggers the associated functions at the said configuration generator unit **202** and the said function executor **203**.

The configuration generator unit **202** is preferably configured for executing the configuration files. In one embodiment, the configuration generator unit **202** reads the user-specification requirement and the system specification that are stored in the configuration repository **207**. The configuration files include crypto-yaml files, Configtx files, and Compose files.

The function executor unit **203** is connected to the image repository **204**. The function executor unit **203** is preferably configured for executing the list of execution functions in the said sequential order. It calls the functions and executes them accordingly. In certain circumstances, the function executor unit **203** requires data stored in the said image repository **204** to execute some of the said functions. The execution functions include, but not limited to, those associated with test IP address, transfer prerequisites, install prerequisites, bring up orderer nodes, bring up certificate authority (CA), bring up peers, and CouchDBs, create channel, join channel and update anchor peers.

The installer agent **205** is deployed on a remote node. It is preferably configured for performing the installation of the private blockchain network in a different network domain. In one embodiment, the installer agent **205** is installed at the remote node when the installation is performed for different network domains. It is crucial to ensure that each different site or organization would still have their own authority control over the installation.

The chaincode management module **206** is preferably configured to manage a plurality of chaincodes employed for the said private blockchain network. A chaincode is a piece of code that is written in one of supported languages, e.g. Java, that implements a prescribed interface, i.e. the interface module **100**. The chaincode can be installed and instantiated through the API at the interface module **100** onto the private blockchain network thereof, enabling interaction therebetween. The chaincode includes, but is not limited to, an upload chaincode, an install chaincode, an instantiate chaincode, an upgrade chaincode, an invoke chaincode, and a query chaincode.

The configuration repository **207** is preferably configured for storing the said user-specified requirement, the said full network configuration comprising the list of execution functions, the configuration files and the filenames, and the said list of blockchain hosts and services. It can be made accessible, with a proper authorization, to the interface module **100**, the function governor unit **201**, the configuration generator unit **202,** and the installer agent **205**. In a subsequent run, any previous date will be considered as default, and the app owner or user would be still able to change or amend with other attributes.

The optimization module **300** is connected to the interface module **100** and the configurator engine module **200**. It can be configured for dynamically optimizing the private blockchain network by way of monitoring the full network configuration generated thereof, which is retrievable from its dedicated storage in the configuration repository **207**.

The optimization module **300** comprises a watcher unit **301**, an analyzer unit **302,** and a tuner unit **303**.

The watcher unit **301** is preferably configured for preparing and monitoring a list of blockchain hosts and services associated with the full network configuration. According to one embodiment, the list of blockchain hosts and services comprises node availability, host availability, utilization parameters, service health (up and down), number of ports, and service logs (errors and warnings). Peer nodes, orderer nodes and CA nodes, command line interface (CLI), REST API, and blockchain API may also be included into the said list of blockchain hosts and services. During its operation, the watcher unit **301** will read the said full network configuration from the configuration repository **207**. Upon completion, the watcher

unit **301** will automatically set up and monitor all components of the said private blockchain network provided in the list of blockchain hosts and services thereof.

The analyzer unit **302** is preferably configured for identifying an irregularity in the list of blockchain hosts and services thereof. In one embodiment, the irregularity includes any error in the list of blockchain hosts and services monitored thereof. The irregularity may be, for example, error from logs, performance patterns, and dead services or nodes.

The tuner unit **303** is preferably configured for correcting the irregularity detected thereof. It is preferred that the tuner unit **303** dynamically optimize the entire private blockchain network. In one embodiment, the tuner unit **303**, for example, restarts dead services, tunes service performance, regulates the number of virtual machines, domain name, organization name, and host address.

The recovery module **400** is connected to the interface module **100**, the configurator engine module **200,** and the optimization module **300**. It is preferably configured for performing a network recovery operation of the said private blockchain network.

The recovery module **400** comprises a system backup unit **401**, a first restoration unit **402**, a second restoration unit **403**, a recovery repository **404**, and a backup and restoration manager **405**.

The system backup unit **401** is preferably configured for providing a snapshot backup having network components of the said private blockchain network. The network components include, but not limited to, peer nodes, orderer nodes, CA nodes, and middleware nodes. It is preferred that the system backup unit **401** performs the snapshot backup to the said private blockchain network which includes node configuration, ledger data, CouchDB data, and certificates. The system backup unit **401** preferably reads the live private blockchain network and write the same into the recovery repository **404** thereof. The inputs received from the app owner or user, i.e. the user-specified requirement and its system specification, will be captured and stored in the same recovery repository **404** too. In a subsequent run, any previous date will be considered as default, and the app owner or user would be still able to change or amend with other attributes.

The first restoration unit **402** is preferably configured for providing a list of available backup sets retrievable from the recovery repository **404** based on metadata associated thereof. Based on the choice made by the app owner or user on the list of available backup sets, the first restoration unit **402** will restore a system configuration and data into the said private blockchain network which would be able to bring them to a state as if nothing happened since a specific point in time from the chosen backup set thereof.

The second restoration unit **403** is preferably configured for providing a restoration of a single network component selected from the said network components. In one embodiment, the second restoration unit **403** allows the app owner or user to choose which one of the network components is desired for restoration. Instead of total system restoration, the app owner or user is given an option to restore the selected one network component only, for instance, either peer node, orderer node, CA node, or middleware node. However, all peer nodes need to be restored in all or none mode.

The recovery repository **404** is configured to store the backup data and provide metadata associated thereof.

The backup and restoration manager **405** is preferably configured for managing the system backup unit **401**, the first restoration unit **402**, the second restoration unit **403,** and the recovery repository **404** thereof. It is preferred that the backup and restoration manager **405** manages the backup activities thereof, call backup thereof, and restore the network components thereof.

Concerning the method of the present invention, Figure 3 provides a summarized flow diagram depicting the steps involved, as explained in the preceding paragraphs in connection with the system thereof. For purposes of clarity in explanation and understanding, the method of the present invention is described in the following section.

The method preferably begins with step **500** of receiving, by the interface module **100**, a user-specified requirement, including a system specification of a proposed private blockchain network from the app owner or user. It is subsequently followed by step **501** of processing, by the configurator engine module **200**, the user-specified requirement to generate a full network configuration for the said

private blockchain network. Following that, the method initiates the step **502** of dynamically optimizing, by the optimization module **300**, the private blockchain network by way of monitoring the full network configuration generated thereof. The method further provides the step **503** of performing, by the recovery module **400**, a network recovery operation of the said private blockchain network.

Figures 4, 5, and 6 provide the flow diagrams of step **501**, step **502,** and step **503**, respectively.

According to Figure 4, step **501** preferably includes the step **501a** of generating a list of execution functions in a sequential order, configuration files and filenames derivable based on the said system specification, the step **501b** of executing the configuration files, and the step **501c** of executing the list of execution functions in the said sequential order.

According to Figure 5, step **502** preferably includes the step **502a** of monitoring a list of blockchain hosts and services associated with the full network configuration, the step **502b** of identifying an irregularity in the list of blockchain hosts and services thereof, and the step **502c** of correcting the irregularity detected thereof.

According to Figure 6, step **503** preferably includes the step **503a** of providing a snapshot backup having network components of the said private blockchain network, the step **503b** of providing a list of available backup sets retrievable from a recovery repository based on metadata associated thereof, and the step **503c** of providing a restoration of a single network component selected from the said network components.

Figures 7, 8, 12 and 13 may provide greater insight to those of ordinary skill in the art as to the operation of the interface module **100** (see step **500**), the configurator engine module **200** (see step **501**), the optimization module **300** (see step **502**) and the recovery module **400** (see step **503**), respectively.

Figure 7 related to the operation of the interface module **100** having its flow described by step **500** begins with a user login session. In this regard, the user is logged in to the system of the present invention using a password through the GUI. If the user selects the quick-start module being the user-specified requirement, then

the interface module **100** will proceed to call the configurator engine module **200**. The configurator engine module **200** shall utilize the system default values to create the corresponding system default private blockchain network. The quick-start module does not require any other inputs from the app owner or user. If the user selects the customized module being the user-specified requirement, then the interface module **100** will proceed to receive a system specification from the user to set up a customized private blockchain network. The system specification may include network attributes that may be selected from a group comprising network name, number of virtual machines, number of hosts, username, password, port number, number of organizations, domain name and organization name. Once the hosts and passwords are verified, the interface module **100** will store the system specification provided thereof in the configuration repository **207**. Subsequently, the interface module **100** will call the configurator engine module **200** to create the said customized private blockchain network.

According to Figure 8, with reference to step **501**, the configurator engine module **200** begins by way of receiving and parsing a command request transmitted from the interface module **100**. It will proceed to retrieve the user-specified requirement and associated data including the user specification from the configuration repository **207**. The configurator engine module **200** subsequently executes the configuration files and the list of execution functions in the predetermined sequential order until all functions are executed. The configuration files include crypto-yaml files, Configtx files, and Compose files. The execution functions include, but not limited to, those associated with test IP address, transfer prerequisites, install prerequisites, bring up orderer nodes, bring up CA, bring up peers, and CouchDBs, create channel, join channel and update anchor peers. Following that, the configurator engine module **200** generates a full network configuration for the private blockchain network proposed thereof.

Meanwhile, Figures 9, 10, and 11 may respectively provide greater insight into the operation of the function governor unit **201** (see step **501a**), the configuration generator unit **202** (see step **501b**), and the function executor unit **203** (see step **501c**) of the said configurator engine module **200**.

In Figure 9, the function governor unit **201,** with reference to step **501a**, accesses the configuration repository **207** to read data therefrom such as virtual machine IP addresses, usernames and passwords associated thereof. Following

that, the function governor unit **201** generates a master configuration. The master configuration may include a prerequisite list, nodes running on virtual machines, and port numbers to be used. For instance,

{"VM": "10,1,25,107",
"Config": {"Nodes": ["peer1", "peer2", "orderer1", ..] },
"VM":"10.1.25.108",
"Config": {"Nodes": ["peer3", "peer4", "orderer2", ..] },

Subsequently, the function governor unit **201** reads the execution function list and dependencies (templates). It also will check the function dependencies and execution mode. An example of the execution function list, for each virtual machine, is as follows:

{"VM": "10.1.25.107",
"Function": [GenerateCrypto, GenerateConfigtx, GenerateCompose, IP, transferPrerequisite, installPrerequisite..]}

The function governor unit **201** will next proceed to generate function parameters, exemplarily shown as follows:

{"Function": { "Name": "GenerateConfigtx, "Parameters":{ "Channel_Name": "mimoschannel"}}}

Finally, function governor unit **201** generates a test function list, as follows:

{"Functions_List": ["GeneretaCrypto", "GenereateConfigtx"]}

In Figure 10, the configuration generator unit **202**, with reference to step **501b**, receives and parses the received parameters based on the configuration name. Next, it will check validity of those parameters to ensure the parameter data is valid. For instance, username=dj.

The configuration generator unit **202** will read the configuration files and their template files to generate configuration lines depending on certain criteria, for example, different numbers of virtual machines generate different port numbers. Next, it will check the generation mode whether it is required to insert the

configuration files or to insert the configuration lines while preserving other modifications. Finally, the configuration generator unit **202** writes to a destination file and outputs the configuration files.

In Figure 11, the function executor unit **203**, with reference to step **501c**, parses the function name and parameters thereof, prior to checking the validity of the function. The function executor unit **203** next will check the function execution mode, for instance,

Sync, Async, SSH CLI Call, Web service Call

If the function execution mode is synchronized (or sync) execution, then the function executor unit **203** will check for an exit status and return value. Upon completion, the function executor unit **203** will pass the said exit status and the return value to the calling module, e.g. the interface module **100**. If the function execution mode is an asynchronized (or async) execution, then the function executor unit **203** will check for a process status. A process status checker will run in new threads, and shall be checked at a predefined interval, e.g. one minute. Upon completion, the function executor unit **203** will pass it back to the calling module thereof.

Figure 12 related to the operation of the optimization module **300** having its flow described by step **502** begins with the step of reading the configuration repository to build a list of blockchain hosts and services to be monitored. Following that, the optimization module **300** will execute the watcher unit **301** for monitoring the said list of blockchain hosts and services, including node availability, utilization parameters and service health by way of checking the availability of port number and service logs (for errors and warnings). The collected parameters of attributes shall be forwarded to the analyzer unit **302**. The analyzer unit **302** will analyze the collected parameters for identifying any irregularities and requirement for attention or action. It parses the collected parameters to determine host availability, port status, service status, utilization of central processing unit (CPU), memory and storage, and trend and projection of CPU, memory and storage thereof. The process shall continue or iterating until an exit command is received. The analyzer unit **302** will, based on the analysis, determine if an action for tuning is required. If so, then analyzer unit **302** will generate a tuning command based on the action

required and executed to the tuner unit **303**. A notification will be sent to an administrator to indicate that the tuning command has been executed.

Figure 13 related to the operation of the recovery module **400** having its flow described by step **503** begins with the step of choosing an operation from a selection comprising a system backup executable by the system backup unit **401**, a system restore executable by the first restoration unit **402** and a component restore executable by the second restoration unit **403**. If the system backup is chosen by the user, then the system backup unit **401** will perform a snapshot backup having network components of the said private blockchain network. If the system restore is chosen by the user, the first restoration unit **402** will provide the list of available backup sets. Based on the choice made by the app owner or user on the list of available backup sets, the first restoration unit **402** will restore a system configuration and data into the said private blockchain network which would be able to bring them to a state as if nothing happened since a specific point in time from the chosen backup set thereof. If the component restore is chosen by the user, then the second restoration unit **403** will proceed with restoration of a single network component selected from the said network components. The second restoration unit **403** will allow the app owner or user to choose which one of the network components is desired for restoration. Instead of total system restoration, the app owner or user is given an option to restore the selected one network component only, for instance, either peer node, orderer node, CA node, or middleware node. However, all peer nodes need to be restored in all or none mode. Finally, the recovery module 400 will update the backup data and catalogue accordingly.

Although an overview of the inventive subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments without departing from the broader scope of embodiments of the present disclosure. Such embodiments of the inventive subject matter may be referred to herein, individually or collectively, by the term "invention" merely for convenience and without intending to voluntarily limit the scope of this application to any single disclosure or inventive concept if more than one is, in fact, disclosed.

The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical

substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

As used herein, the term "or" may be construed in either an inclusive or exclusive sense. Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, modules, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present disclosure. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

The foregoing description, for the purpose of explanation, has been described with reference to specific example embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the possible example embodiments to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The example embodiments were chosen and described in order to best explain the principles involved and their practical applications, to thereby enable others skilled in the art to best utilize the various example embodiments with various modifications as are suited to the particular use contemplated.

It will also be understood that, although the terms "first," "second," and so forth may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first contact could be termed a second contact, and, similarly, a second contact could be termed a first contact, without departing from the scope of the present example embodiments. The first contact and the second contact are both contacts, but they are not the same contact.

The terminology used in the description of the example embodiments herein is for the purpose of describing particular example embodiments only and is not intended to be limiting. As used in the description of the example embodiments and the appended examples, the singular forms "a," "an," and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term "and/or" as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

As used herein, the term "if" may be construed to mean "when" or "upon" or "in response to determining" or "in response to detecting," depending on the context. Similarly, the phrase "if it is determined" or "if [a stated condition or event] is detected" may be construed to mean "upon determining" or "in response to determining" or "upon detecting [the stated condition or event]" or "in response to detecting [the stated condition or event]," depending on the context.

CLAIMS

1. A system for seamlessly provisioning, configuring, and deploying an enterprise-grade private blockchain network, **characterized in that**, the system comprising:

an interface module (100) configured for receiving a user-specified requirement having a system specification of a private blockchain network;

a configurator engine module (200) connected to the interface module (100) configured for processing the user-specified requirement to generate a full network configuration for the said private blockchain network, comprising:

a function governor unit (201) for generating a list of execution functions in sequential order, configuration files and filenames derivable based on the said system specification;

a configuration generator unit (202) for executing the configuration files; and

a function executor unit (203) connected to an image repository (204) for executing the list of execution functions in the said sequential order;

an optimization module (300) connected to the interface module (100) and the configurator engine module (200) configured for dynamically optimizing the private blockchain network by way of monitoring the full network configuration generated thereof, comprising:

a watcher unit (301) for monitoring a list of blockchain hosts and services associated with the full network configuration;

an analyzer unit (302) for identifying an irregularity in the list of blockchain hosts and services thereof; and

a tuner unit (303) for correcting the irregularity detected thereof; and

a recovery module (400) connected to the interface module (100), the configurator engine module (200) and the optimization module (300) configured for performing a network recovery operation of the said private blockchain network, comprising:

a system backup unit (401) for providing a snapshot backup having network components of the said private blockchain network;

a first restoration unit (402) for providing a list of available backup sets retrievable from a recovery repository (404) based on metadata associated thereof; and

a second restoration unit (403) for providing a restoration of a single network component selected from the said network components.

2. The system according to Claim 1, wherein the configurator engine module (200) further comprises:

an installer agent (205) deployed on a remote node for performing installation of the private blockchain network in a different network domain; and

a chaincode management module (206).

3. The system according to Claim 1, wherein the configurator engine module (200) further comprises a configuration repository (207) for storing the user-specified requirement, the full network configuration comprising the list of execution functions, the configuration files and the filenames, and the list of blockchain hosts and services thereof.

4. The system according to Claim 1, wherein the recovery module (400) comprises a backup and restoration manager (405) to manage the system backup unit (401), the first restoration unit (402), the second restoration unit (403) and the recovery repository (404) thereof.

5. The system according to Claim 1, wherein the user-specified requirement includes a quick-start module for a system default private blockchain network and a customized module for a customized private blockchain network.

6. The system according to Claim 1, wherein the system specification comprises network attributes selected from a group comprising network name, number of virtual machines, number of hosts, username, password, port number, number of organizations, domain name and organization name.

7. The system according to Claim 1, wherein the irregularity includes any error in the list of blockchain hosts and services monitored thereof.

8. The system according to Claim 1, wherein the list of blockchain hosts and services comprises node availability, host availability, utilization parameters, service health (up and down), number of ports, and service logs (errors and warnings).

9. The system according to Claim 1, wherein the network components include peer nodes, orderer nodes, certificate authority nodes, and middleware nodes.

10. A method for seamlessly provisioning, configuring, and deploying an enterprise-grade private blockchain network, **characterized in that**, the method comprising the steps of:

receiving a user-specified requirement including a system specification of a private blockchain network (500);

processing the user-specified requirement to generate a full network configuration for the said private blockchain network (501), including:

generating a list of execution functions in a sequential order, configuration files and filenames derivable based on the said system specification (501a);

executing the configuration files (501b); and

executing the list of execution functions in the said sequential order (501c);

dynamically optimizing the private blockchain network by way of monitoring the full network configuration generated thereof (502), including:

monitoring a list of blockchain hosts and services associated with the full network configuration (502a);

identifying an irregularity in the list of blockchain hosts and services thereof (502b); and

correcting the irregularity detected thereof (502c); and

performing a network recovery operation of the said private blockchain network (503), including:

providing a snapshot backup having network components of the said private blockchain network (503a);

providing a list of available backup sets retrievable from a recovery repository based on metadata associated thereof (503b); and

providing a restoration of a single network component selected from the said network components (503c).

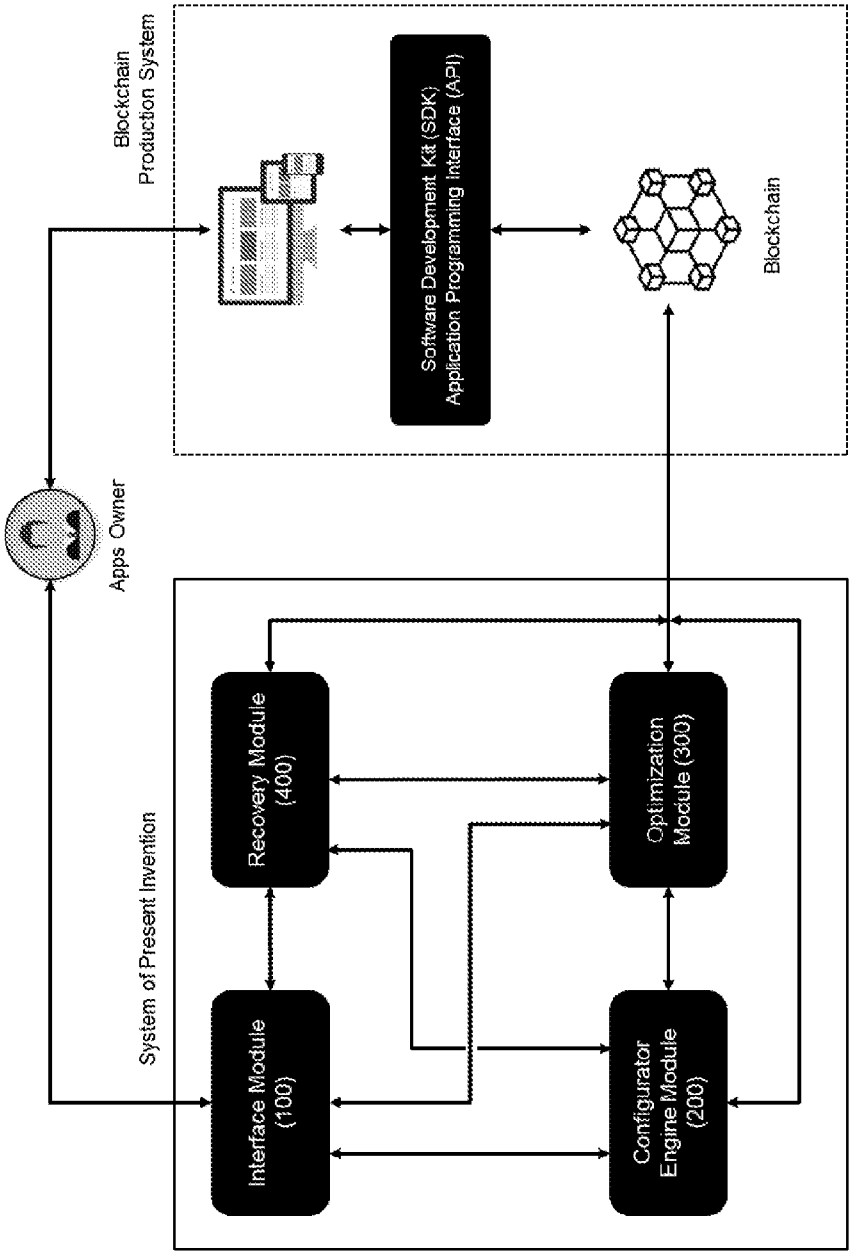

Blockchain Production System
Software Development Kit (SDK) Application Programming Interface (API)
Blockchain
Apps Owner
System of Present Invention
Recovery Module (400)
Optimization Module (300)
Interface Module (100)
Configurator Engine Module (200)

FIGURE 1

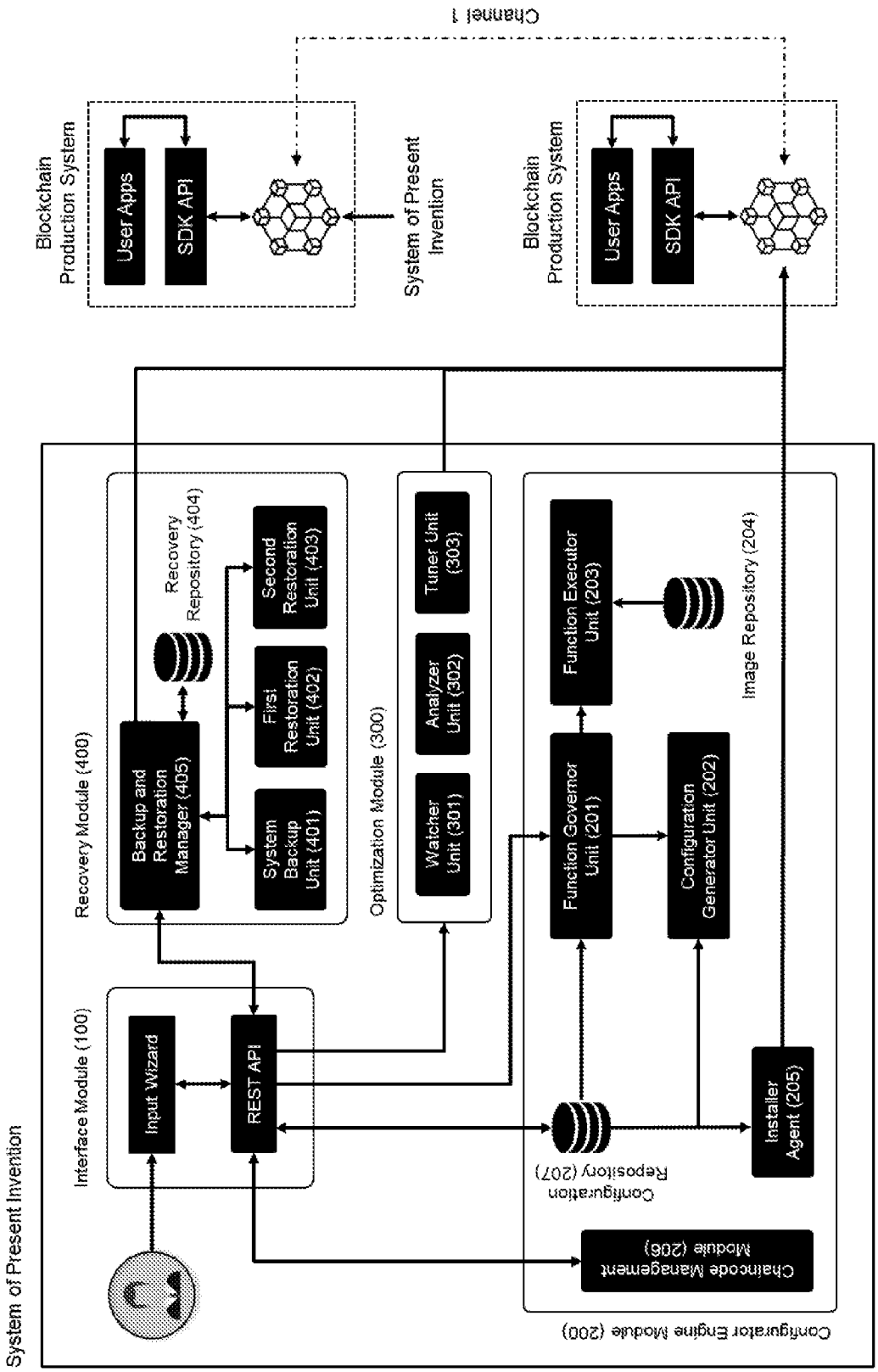

Channel 1
Blockchain Production System
User Apps
SDK API
System of Present Invention
Blockchain Production System
User Apps
SDK API
System of Present Invention
Interface Module (100)
Input Wizard
REST API
Recovery Module (400)
Backup and Restoration Manager (405)
Recovery Repository (404)
System Backup Unit (401)
First Restoration Unit (402)
Second Restoration Unit (403)
Optimization Module (300)
Watcher Unit (301)
Analyzer Unit (302)
Tuner Unit (303)
Configurator Engine Module (200)
Function Governor Unit (201)
Configuration Generator Unit (202)
Function Executor Unit (203)
Image Repository (204)
Installer Agent (205)
Chaincode Management Module (206)
Configuration Repository (207)

FIGURE 2

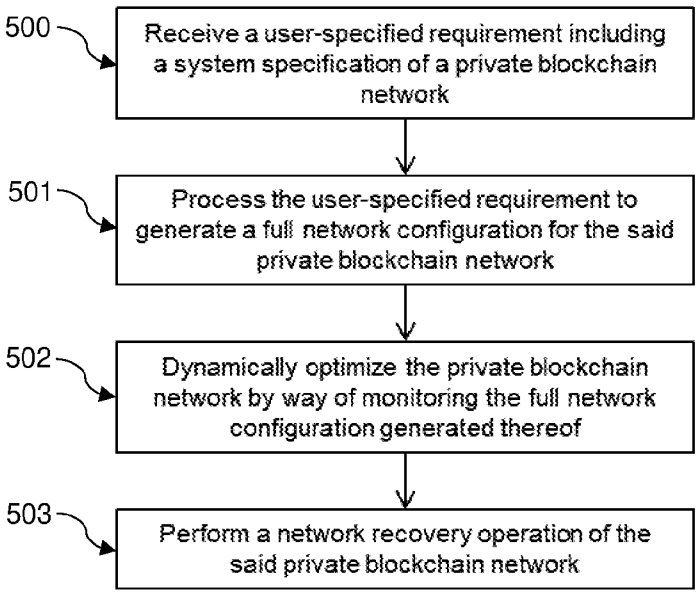


FIGURE 3

501

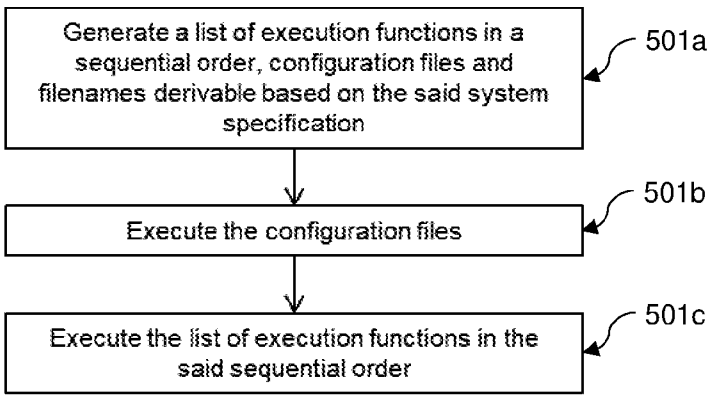


FIGURE 4

502

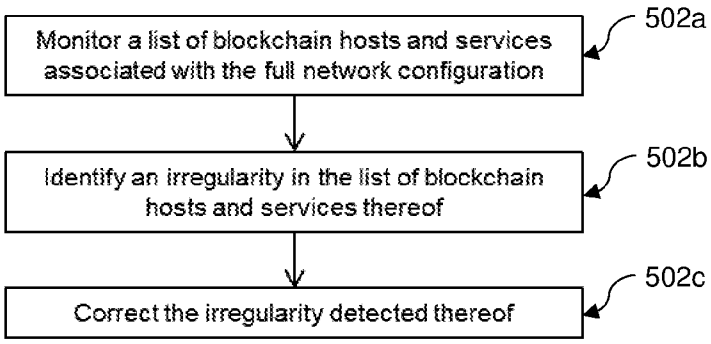


FIGURE 5

503

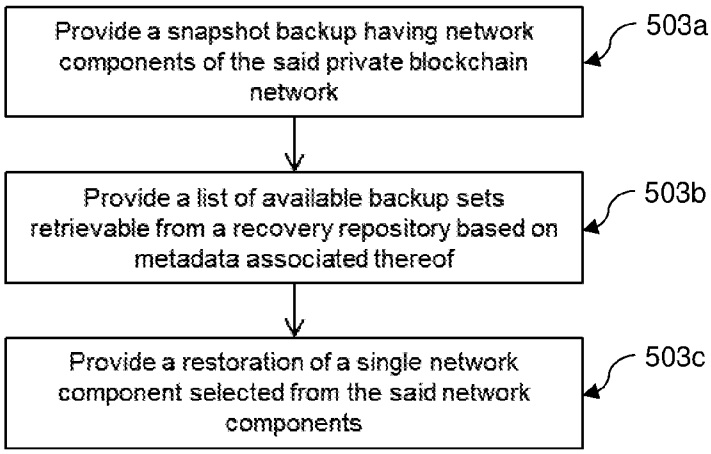


FIGURE 6

Interface Module (100)

Step 500

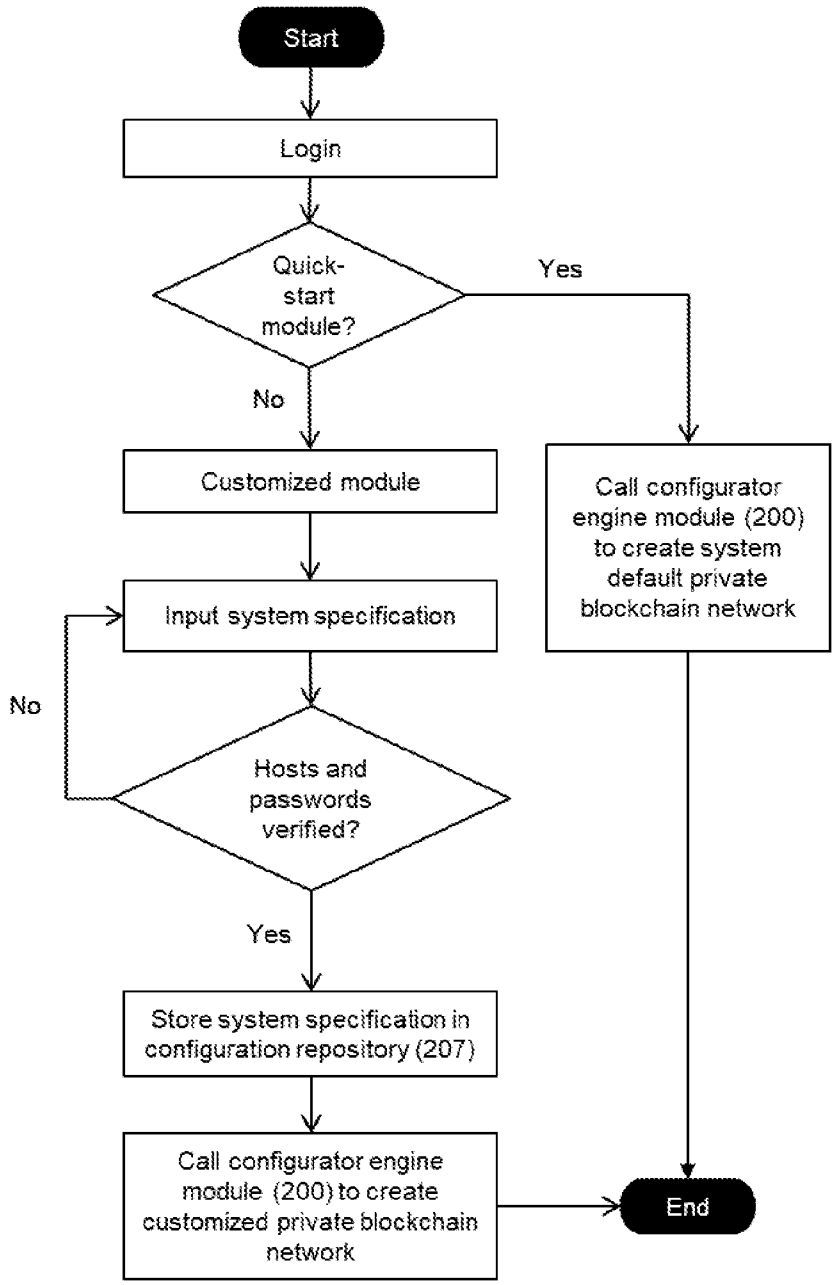


FIGURE 7

Configurator Engine Module (200)

Step 501

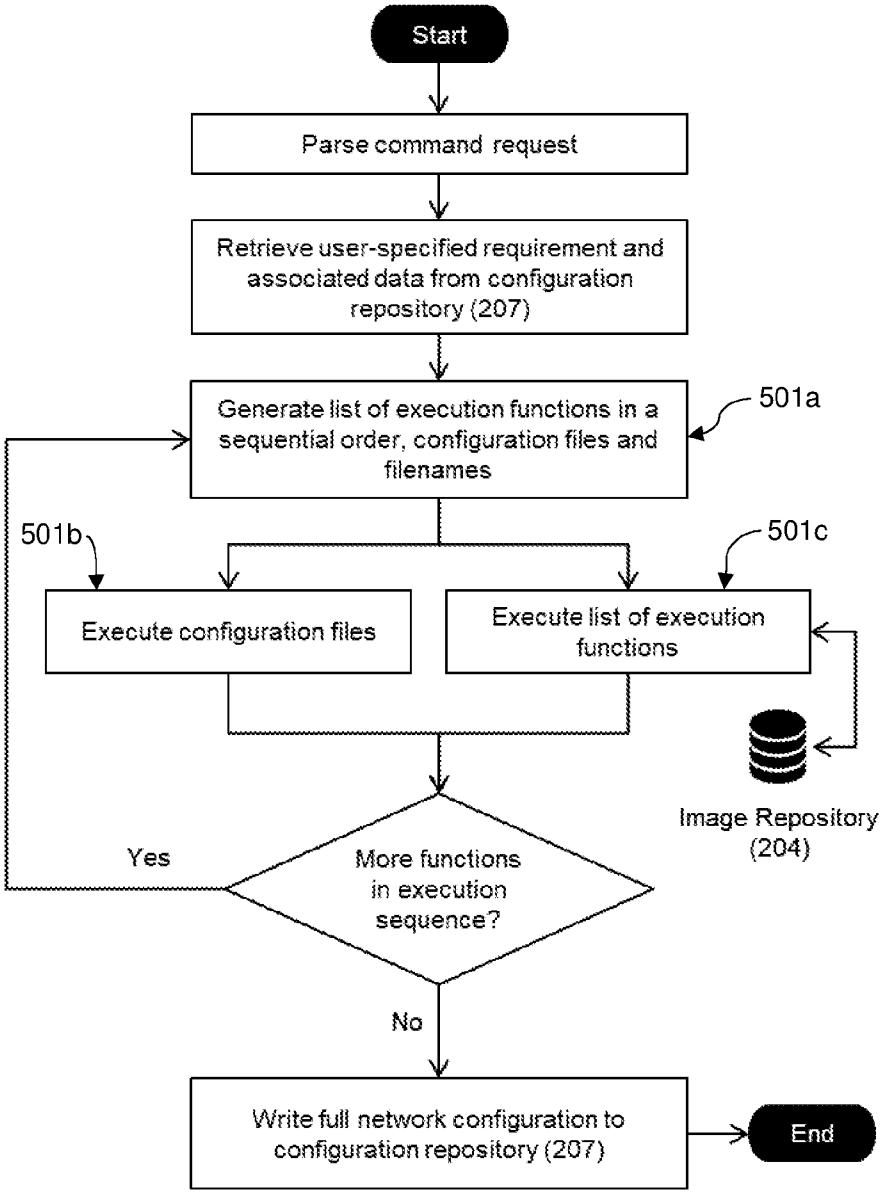


FIGURE 8

Function Governor Unit (201)

Step 501a

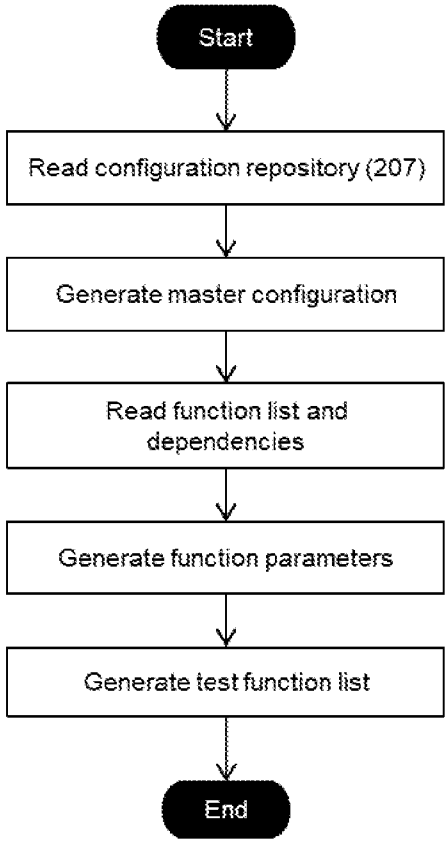


FIGURE 9

Configuration Generator Unit (202)

Step 501b

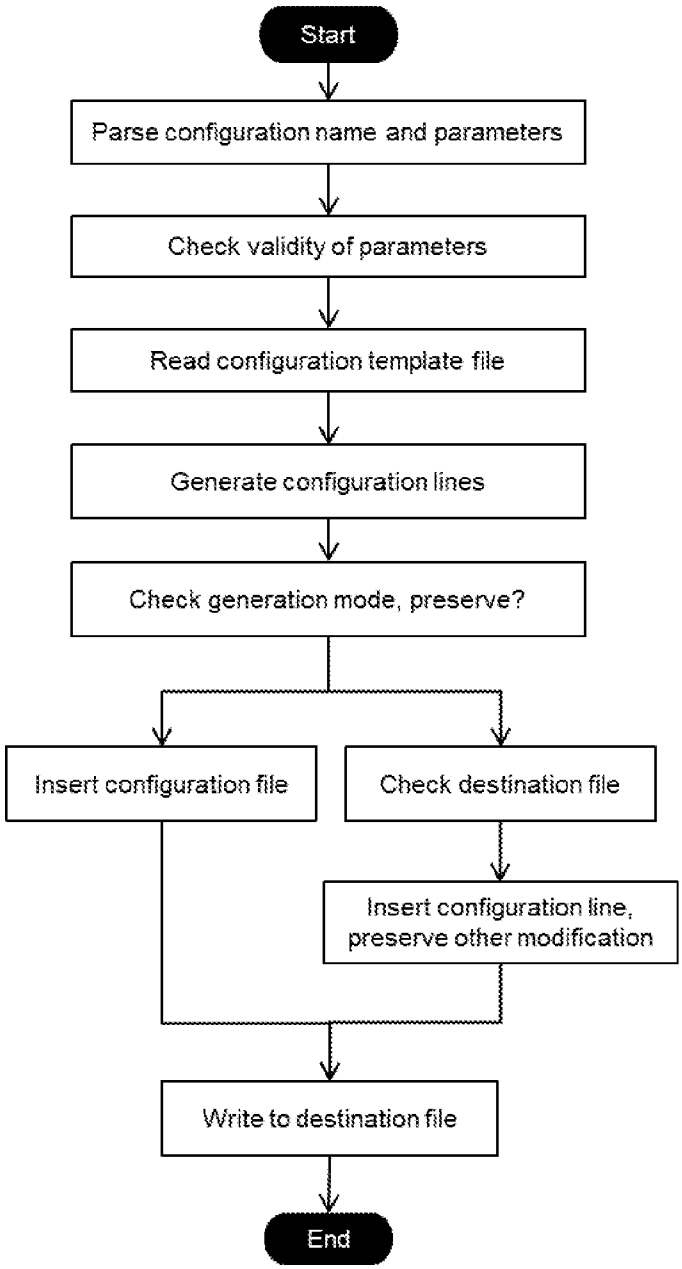


FIGURE 10

Function Executor Unit (203)

Step 501c

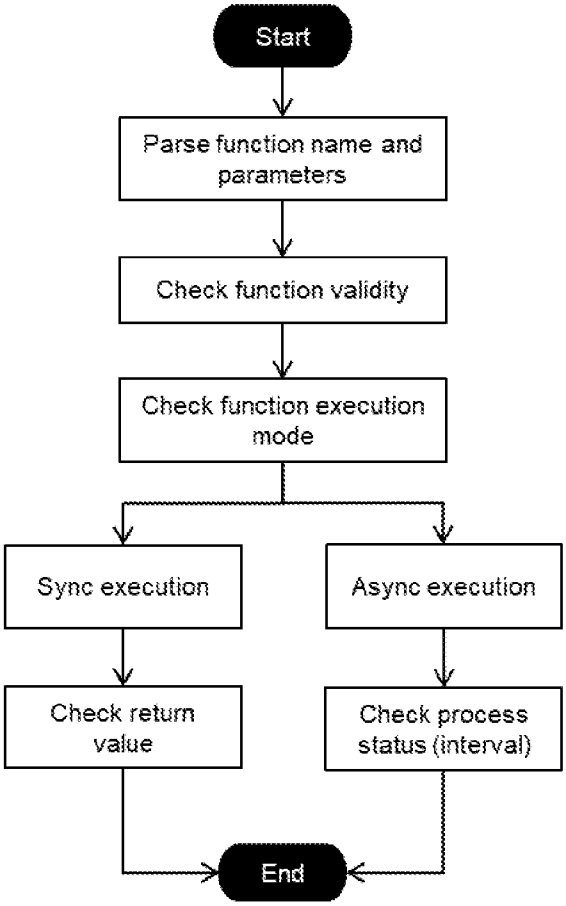


FIGURE 11

Optimization Module (300)

Step 502

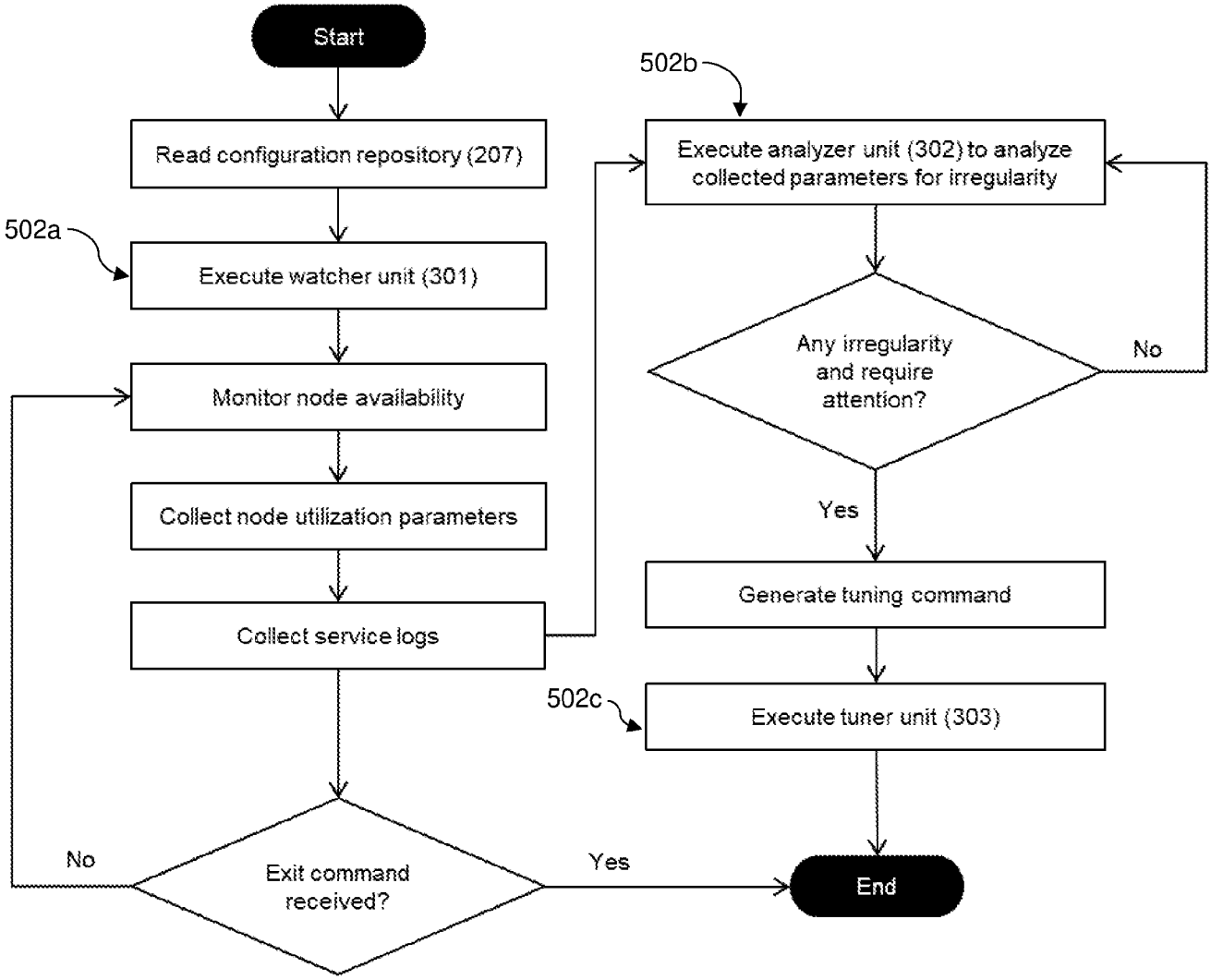


FIGURE 12

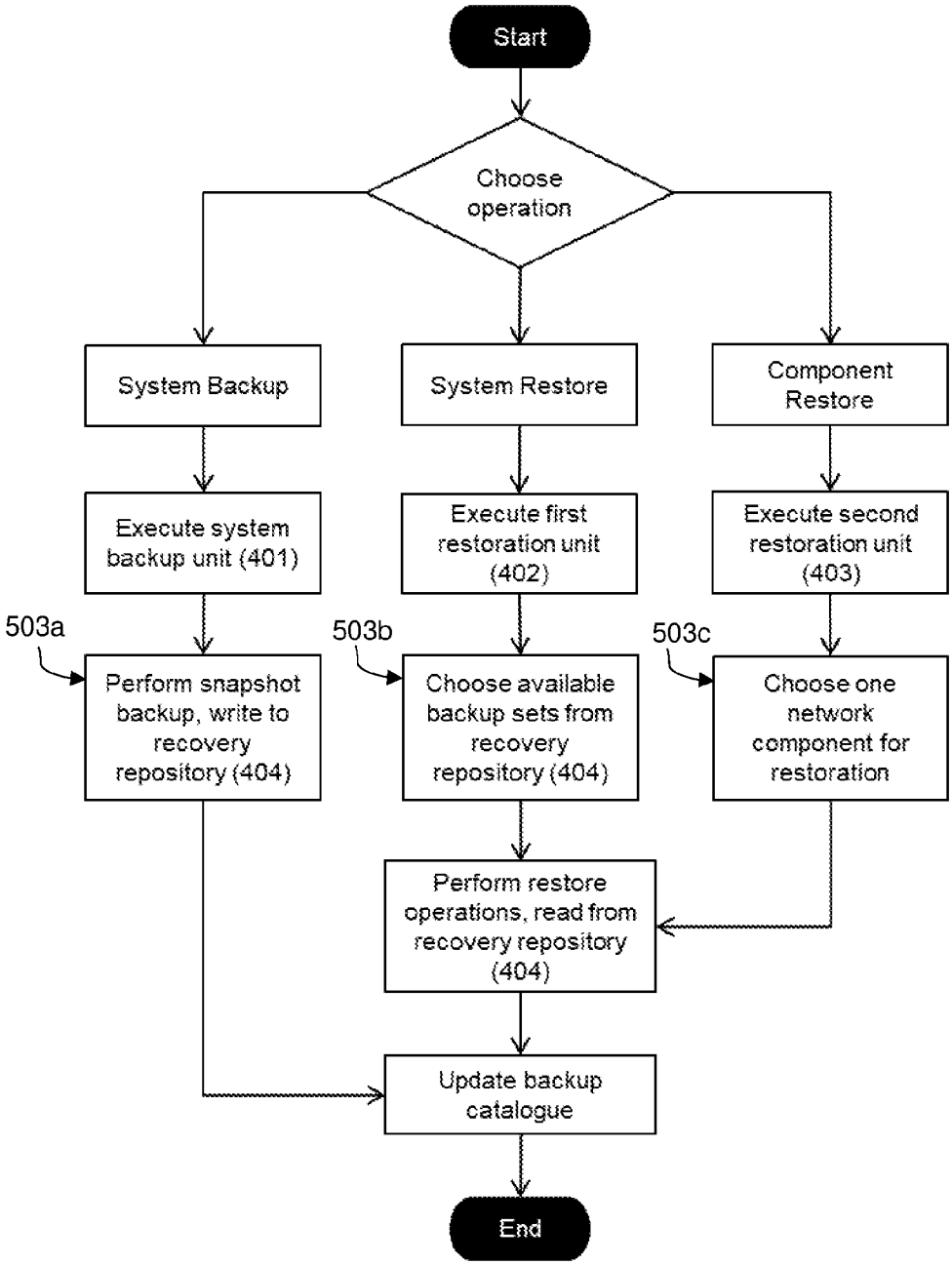

Recovery Module (400)
503
Start
Choose operation
System Backup
System Restore
Component Restore
Execute system backup unit (401)
Execute first restoration unit (402)
Execute second restoration unit (403)
503a
503b
503c
Perform snapshot backup, write to recovery repository (404)
Choose available backup sets from recovery repository (404)
Choose one network component for restoration
Perform restore operations, read from recovery repository (404)
Update backup catalogue
End

FIGURE 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/MY2020/050187

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/24(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L 12/24(2006.01); G06F 16/18(2019.01); G06F 17/27(2006.01); G06Q 20/06(2012.01); H04L 29/06(2006.01); H04L 9/06(2006.01); H04L 9/08(2006.01)

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

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & keywords: enterprise, private blockchain network, full network configuration, file, sequential order, optimizing, irregularity, recovery

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2019-0013933 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC) 10 January 2019 (2019-01-10) paragraphs [0048]-[0136]; claim 1; and figure 3	1-10
A	WO 2020-140015 A1 (MOX-SPEEDCHAIN, LLC) 02 July 2020 (2020-07-02) claims 1-24	1-10
A	WO 2019-210321 A1 (OPTHERIUM LABS OU) 31 October 2019 (2019-10-31) paragraph [0054]; and figure 6	1-10
A	US 2020-0112430 A1 (JAMES TAYLOR et al.) 09 April 2020 (2020-04-09) paragraphs [0072]-[0075]; and figure 4	1-10
A	US 2020-0195495 A1 (INTEL CORPORATION) 18 June 2020 (2020-06-18) claims 9-17	1-10

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

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 April 2021	**05 April 2021**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea**	**YANG, JEONG ROK**
Facsimile No. **+82-42-481-8578**	Telephone No. **+82-42-481-5709**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/MY2020/050187

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2019-0013933 A1	10 January 2019	EP 3649570 A1	13 May 2020
		US 10938548 B2	02 March 2021
		US 2019-0012249 A1	10 January 2019
		US 2019-0013932 A1	10 January 2019
		US 2019-0013934 A1	10 January 2019
		US 2019-0013948 A1	10 January 2019
		WO 2019-010063 A1	10 January 2019
		WO 2019-010064 A1	10 January 2019
		WO 2019-010065 A1	10 January 2019
		WO 2019-010067 A1	10 January 2019
		WO 2019-010068 A1	10 January 2019
WO 2020-140015 A1	02 July 2020	US 2020-0210380 A1	02 July 2020
		US 2020-0213110 A1	02 July 2020
		US 2020-0213291 A1	02 July 2020
		US 2020-0213292 A1	02 July 2020
WO 2019-210321 A1	31 October 2019	AU 2019-260778 A1	26 November 2020
		CA 3097905 A1	31 October 2019
		KR 10-2021-0018226 A	17 February 2021
		US 2019-0330453 A1	31 October 2019
		WO 2019-210244 A1	31 October 2019
US 2020-0112430 A1	09 April 2020	US 2018-0131511 A1	10 May 2018
US 2020-0195495 A1	18 June 2020	None	