(12) UK Patent (19) GB (11) 2582421 (13) B

(45) Date of B Publication 26.07.2023

(54) Title of the Invention: System for secure metering from systems of untrusted data derived from common sources

(51) INT CL: ***G05B 19/418*** (2006.01)

(21) Application No: **2000573.2**

(22) Date of Filing: **15.01.2020**

(30) Priority Data:
(31) **16248355** (32) **15.01.2019** (33) **US**

(43) Date of A Publication **23.09.2020**

(56) Documents Cited:
WO 2018/222066 A1 **WO 2018/059854 A1**
CN 109164780 A

(58) Field of Search:
As for published application 2582421 A viz:
INT CL **G05B, G06F**
Other: **WPI, EPODOC, Patent Fulltext, INTERNET**
updated as appropriate

Additional Fields
INT CL **H04L**
Other: **None**

(72) Inventor(s):
Mark John Nixon
Anthony Amaro Jr
Gang Wang

(73) Proprietor(s):
Fisher-Rosemount Systems, Inc
Bldg. 1, 1100 W. Louis Henna Blvd, Round Rock, Texas 78681, United States of America

(74) Agent and/or Address for Service:
Forresters IP LLP
Rutland House, 148 Edmund Street, BIRMINGHAM, B3 2JA, United Kingdom

GB 2582421 B

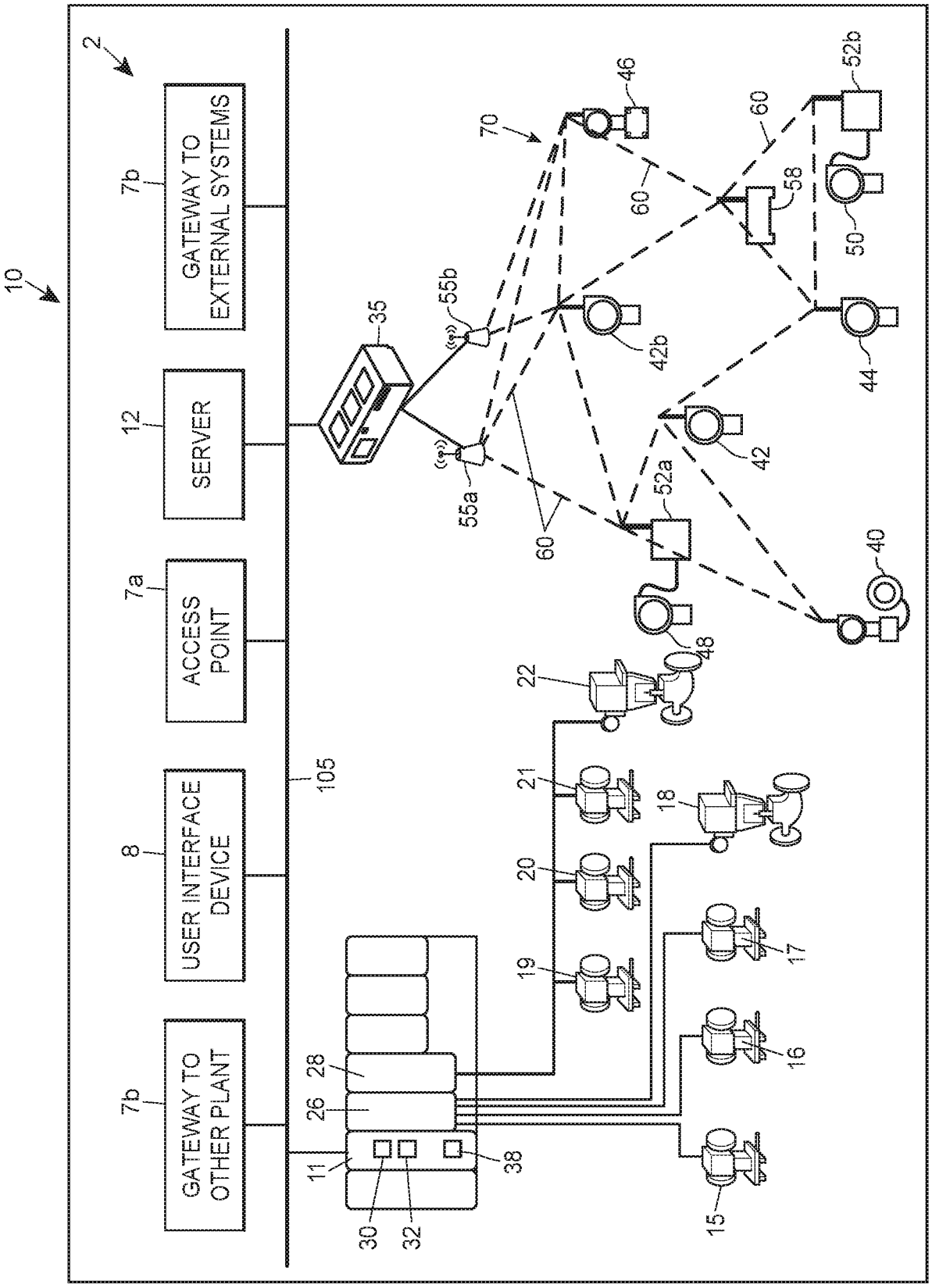
10
2
GATEWAY TO EXTERNAL SYSTEMS
7b
SERVER
12
ACCESS POINT
7a
USER INTERFACE DEVICE
8
GATEWAY TO OTHER PLANT
7b
105
35
55b
55a
70
46
60
58
52b
50
42b
44
42
52a
40
48
22
21
18
20
19
17
16
15
28
26
11
38
30
32

FIG. 1

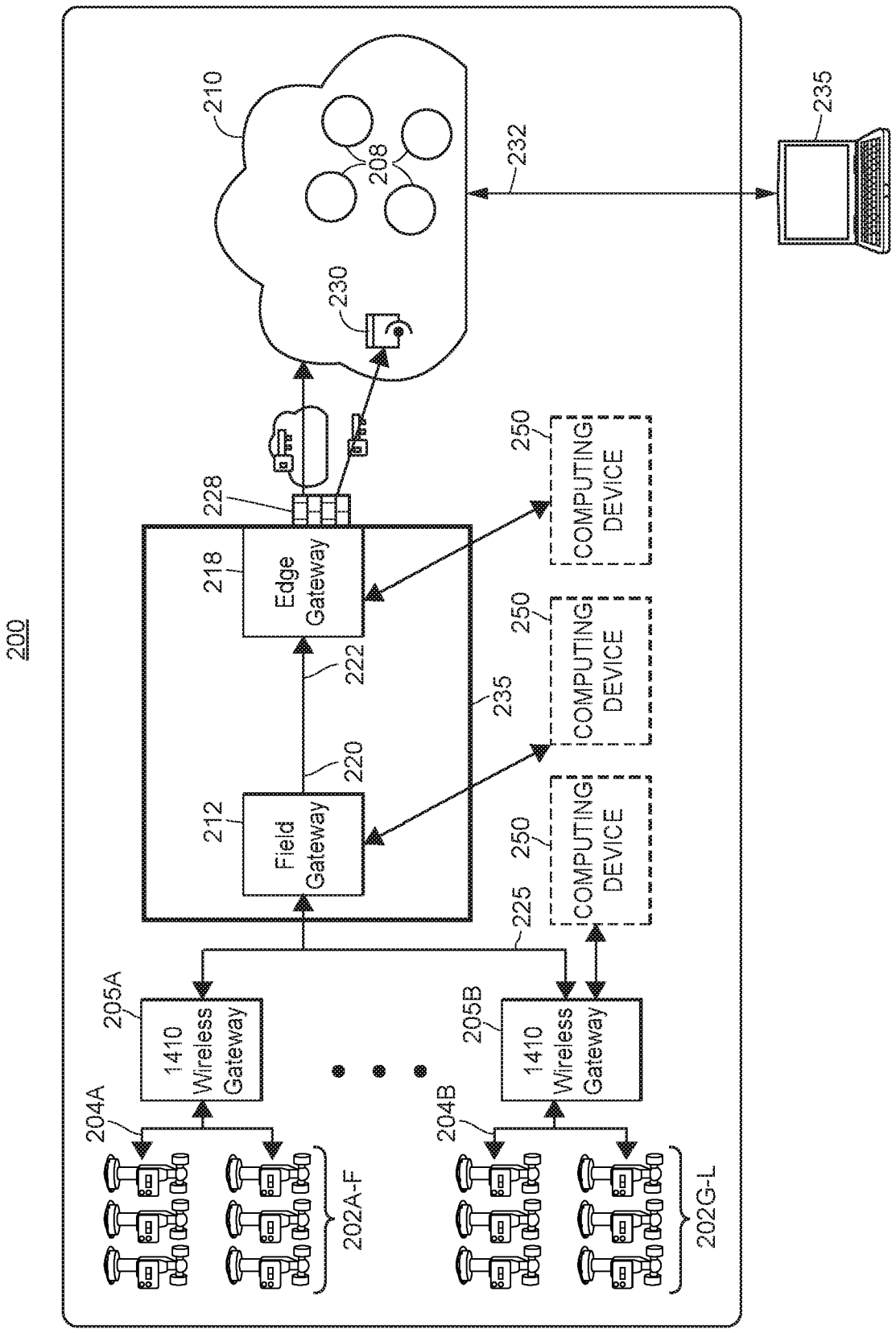
200
210
208
232
235
230
228
218
Edge Gateway
222
220
212
Field Gateway
235
250
COMPUTING DEVICE
250
COMPUTING DEVICE
250
COMPUTING DEVICE
225
205A
1410 Wireless Gateway
205B
1410 Wireless Gateway
204A
204B
202A-F
202G-L

FIG. 2

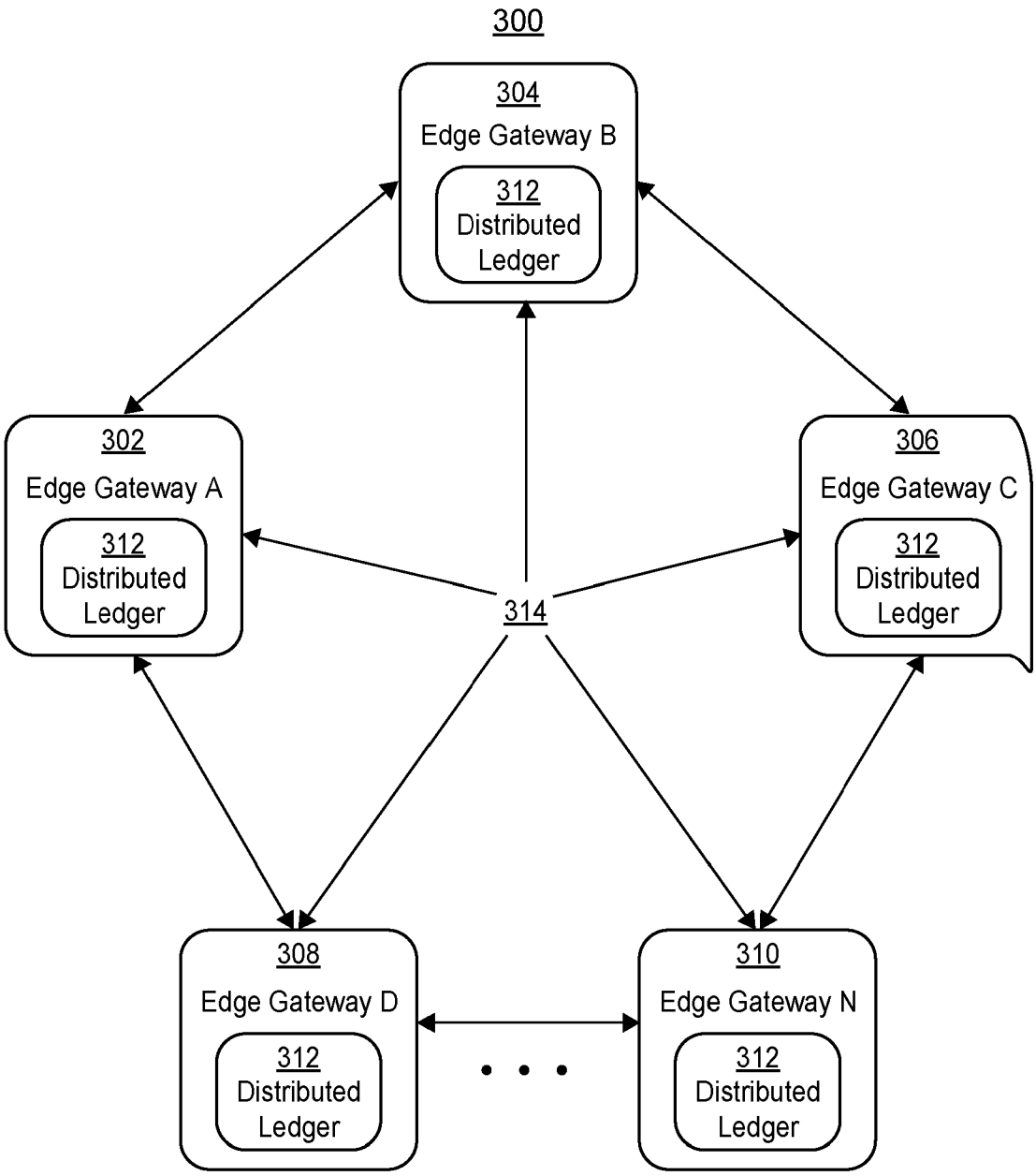
300
304
Edge Gateway B
312
Distributed
Ledger
302
Edge Gateway A
312
Distributed
Ledger
306
Edge Gateway C
312
Distributed
Ledger
314
308
Edge Gateway D
312
Distributed
Ledger
310
Edge Gateway N
312
Distributed
Ledger

FIG. 3

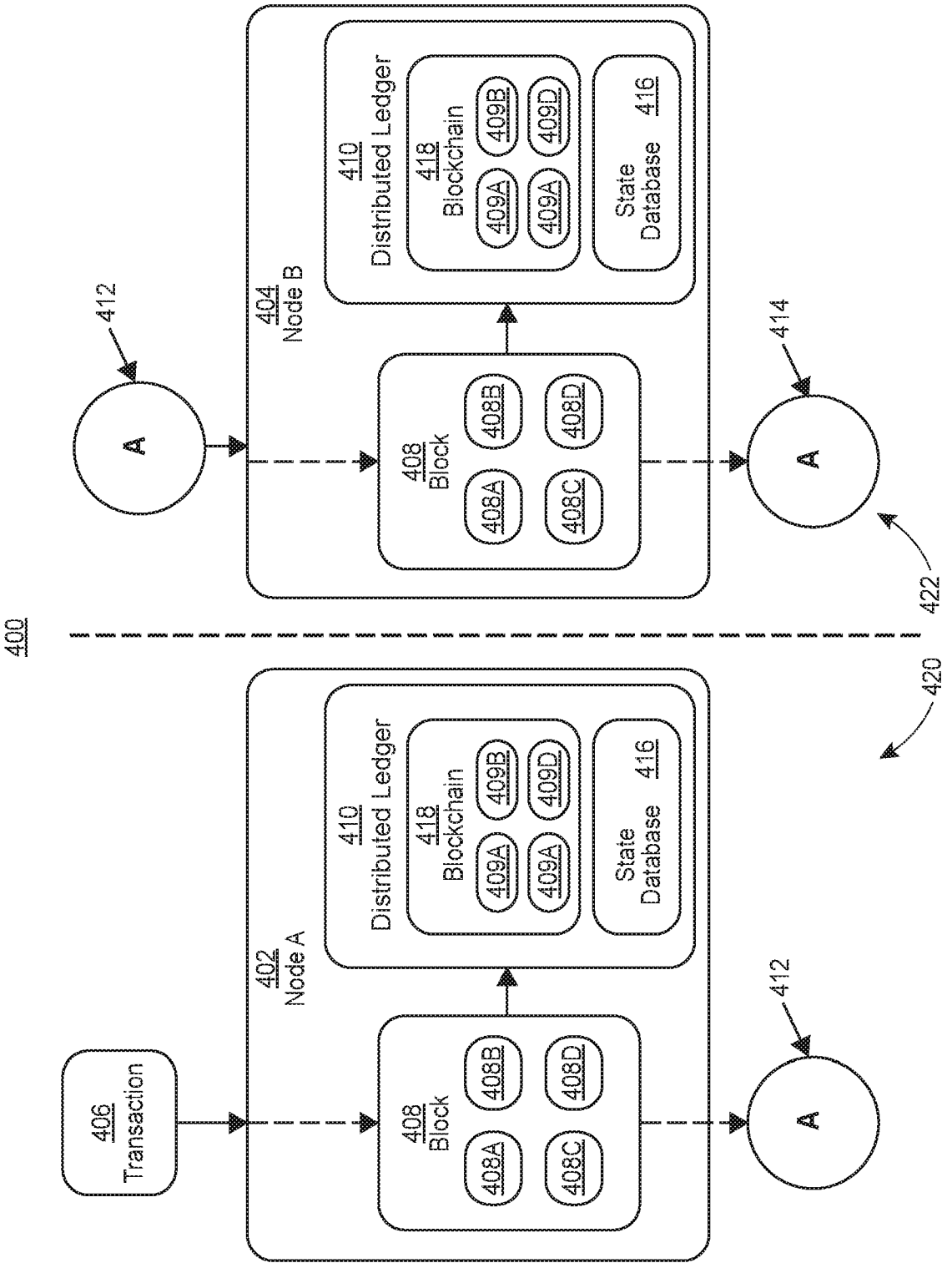
400
406
Transaction
402
Node A
408
Block
408A
408B
408C
408D
410
Distributed Ledger
418
Blockchain
409A
409A
409B
409D
State
Database
416
412
A
404
Node B
408
Block
408A
408B
408C
408D
410
Distributed Ledger
418
Blockchain
409A
409A
409B
409D
State
Database
416
412
A
414
A
420
422

FIG. 4

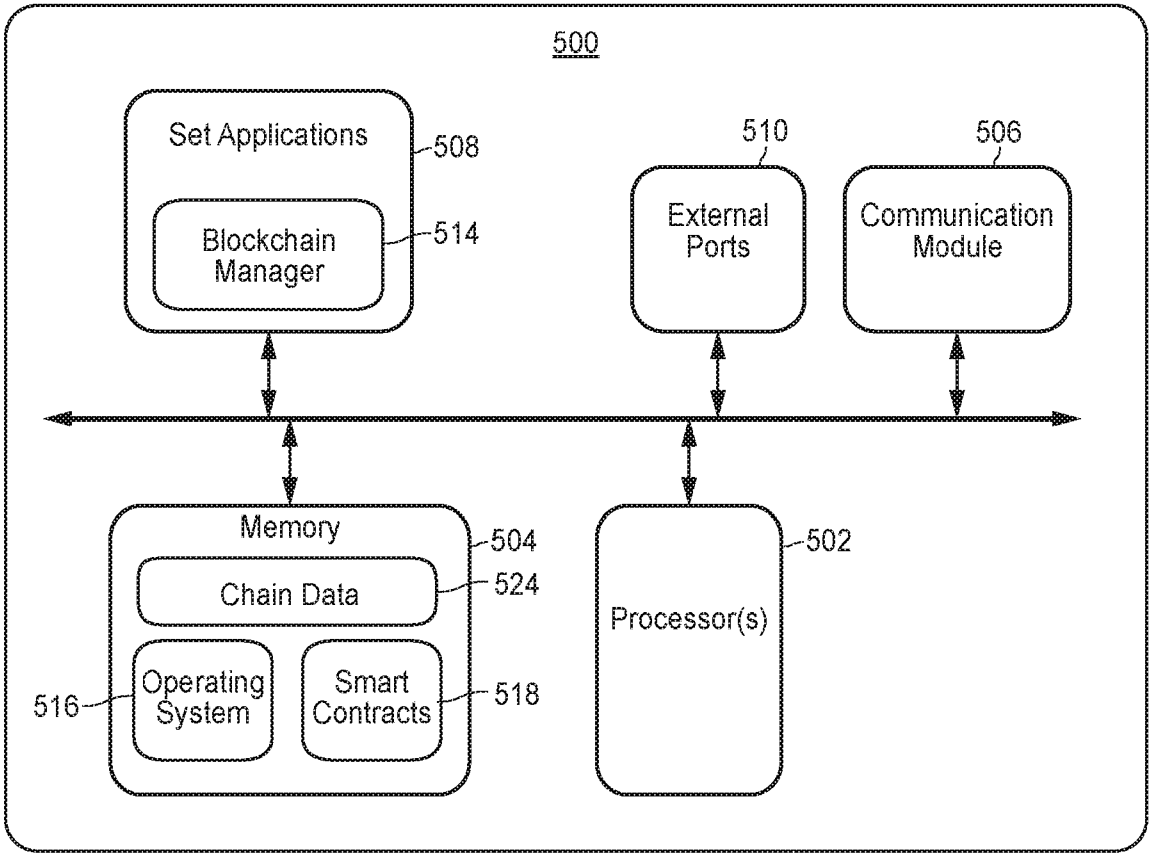
500
Set Applications
508
Blockchain Manager
514
510
External Ports
506
Communication Module
Memory
504
Chain Data
524
516
Operating System
Smart Contracts
518
Processor(s)
502

FIG. 5

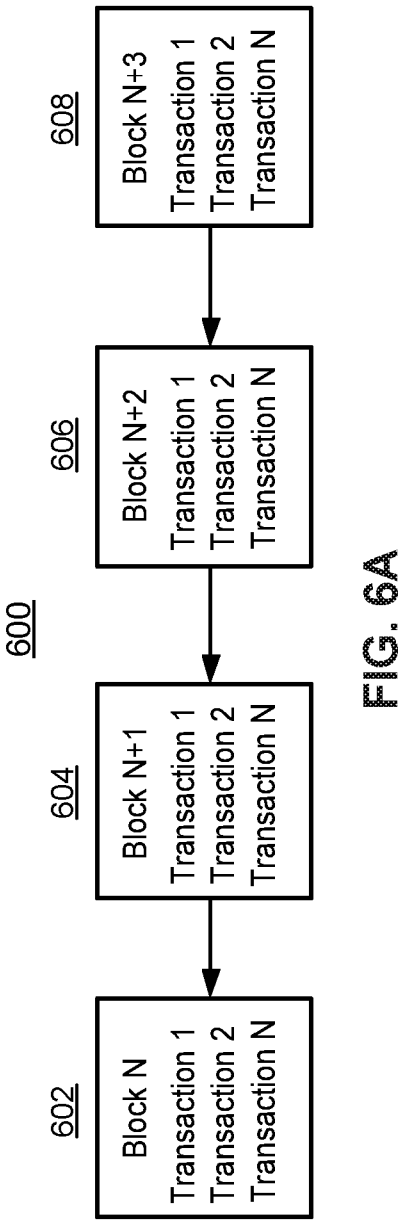
600
602
Block N
Transaction 1
Transaction 2
Transaction N
604
Block N+1
Transaction 1
Transaction 2
Transaction N
606
Block N+2
Transaction 1
Transaction 2
Transaction N
608
Block N+3
Transaction 1
Transaction 2
Transaction N

FIG. 6A

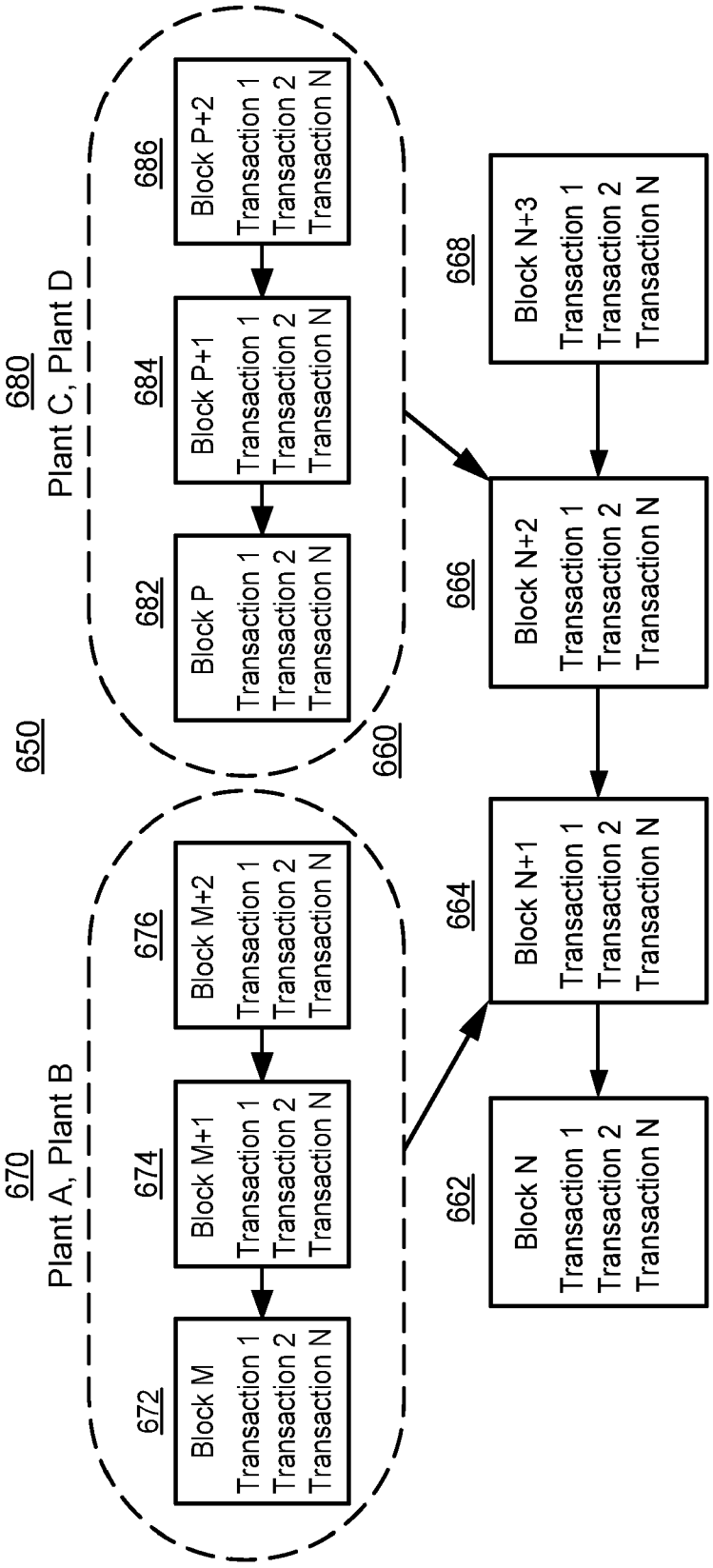
650
670
Plant A, Plant B
672
Block M
Transaction 1
Transaction 2
Transaction N
674
Block M+1
Transaction 1
Transaction 2
Transaction N
676
Block M+2
Transaction 1
Transaction 2
Transaction N
660
680
Plant C, Plant D
682
Block P
Transaction 1
Transaction 2
Transaction N
684
Block P+1
Transaction 1
Transaction 2
Transaction N
686
Block P+2
Transaction 1
Transaction 2
Transaction N
662
Block N
Transaction 1
Transaction 2
Transaction N
664
Block N+1
Transaction 1
Transaction 2
Transaction N
666
Block N+2
Transaction 1
Transaction 2
Transaction N
668
Block N+3
Transaction 1
Transaction 2
Transaction N

FIG. 6B

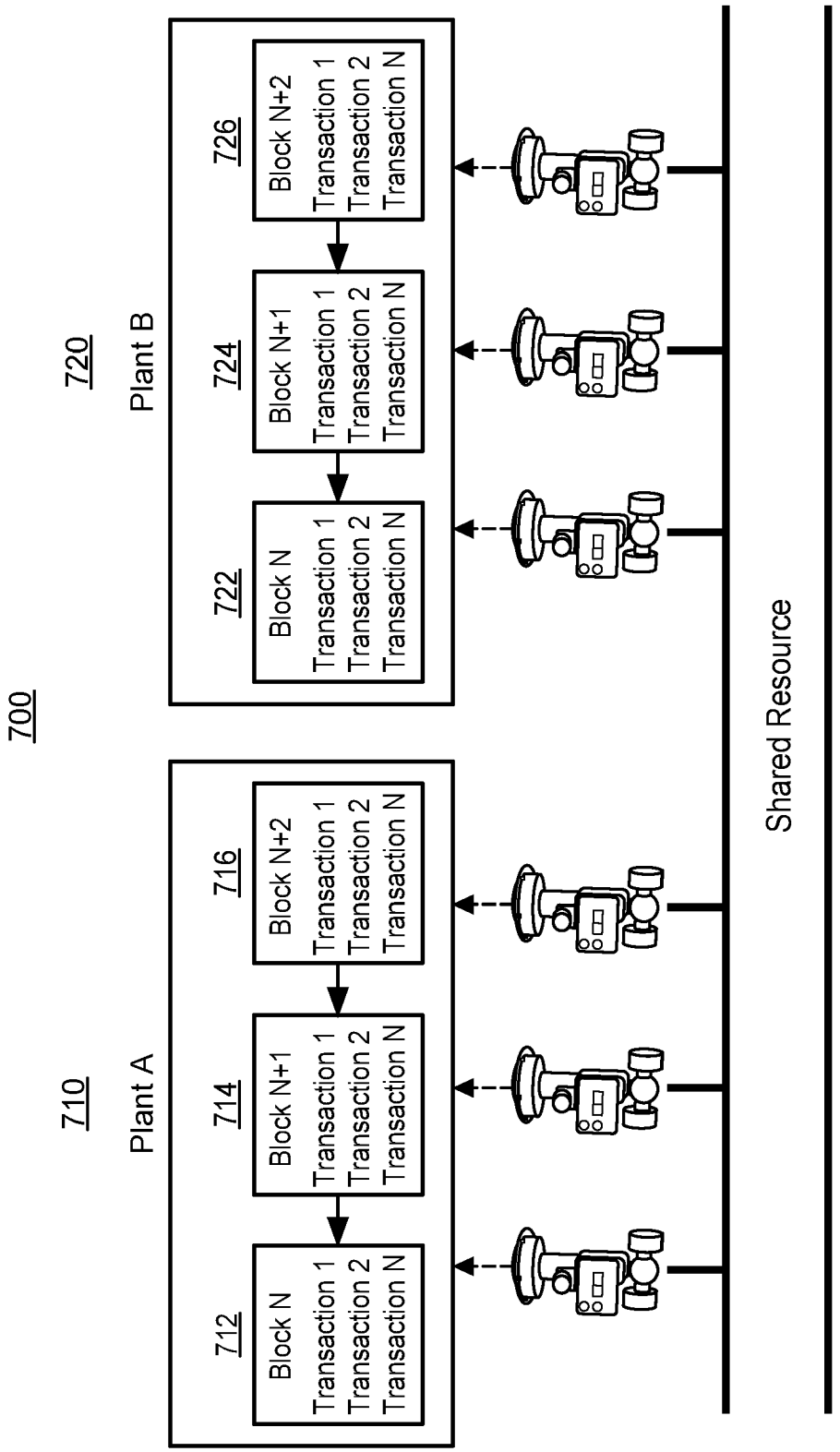
700
710
Plant A
712
Block N
Transaction 1
Transaction 2
Transaction N
714
Block N+1
Transaction 1
Transaction 2
Transaction N
716
Block N+2
Transaction 1
Transaction 2
Transaction N
720
Plant B
722
Block N
Transaction 1
Transaction 2
Transaction N
724
Block N+1
Transaction 1
Transaction 2
Transaction N
726
Block N+2
Transaction 1
Transaction 2
Transaction N
Shared Resource

FIG. 7A

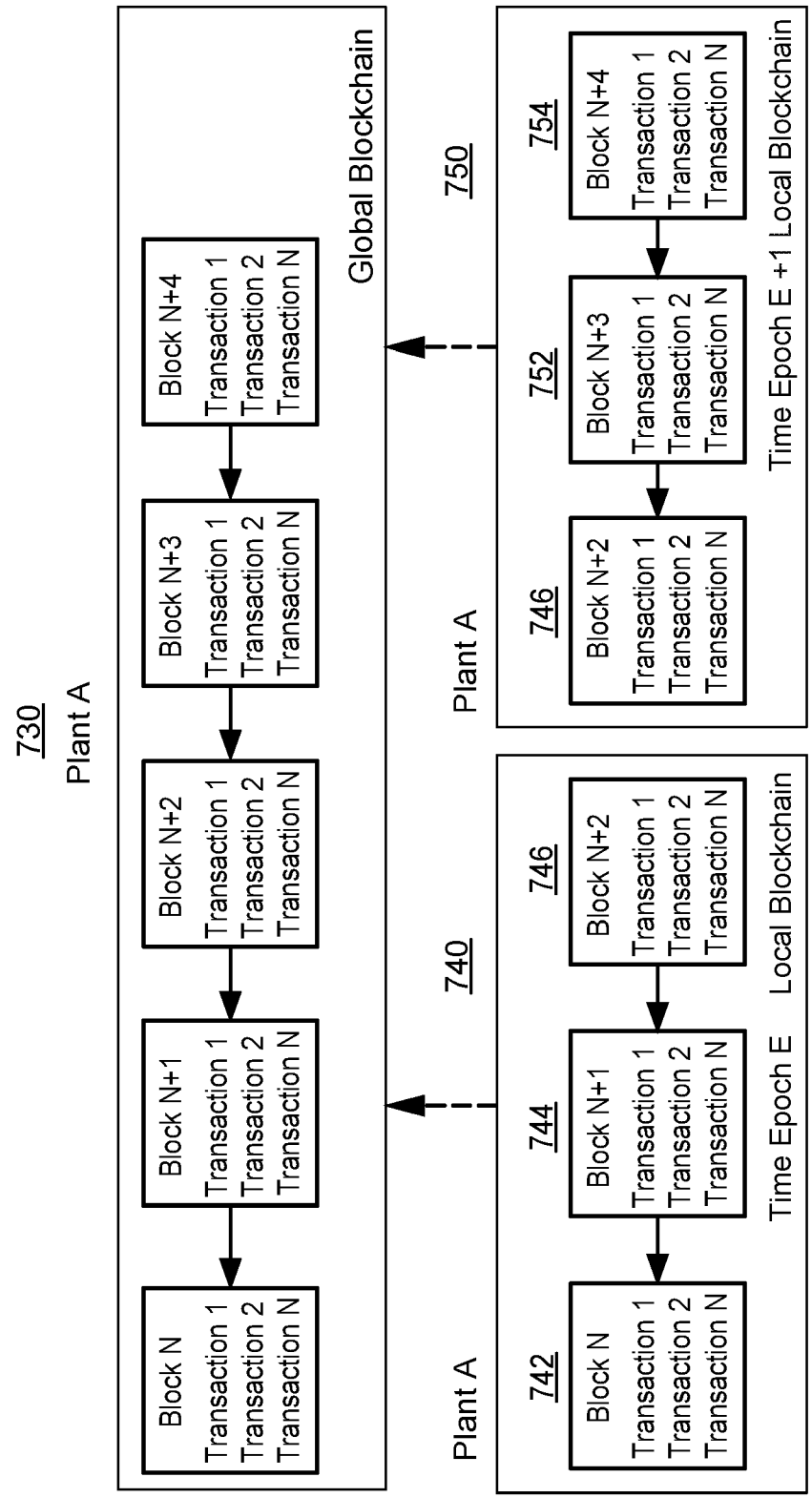
730
Plant A
Global Blockchain
Block N
Transaction 1
Transaction 2
Transaction N
Block N+1
Transaction 1
Transaction 2
Transaction N
Block N+2
Transaction 1
Transaction 2
Transaction N
Block N+3
Transaction 1
Transaction 2
Transaction N
Block N+4
Transaction 1
Transaction 2
Transaction N
740
Plant A
Time Epoch E Local Blockchain
742
744
746
750
Plant A
Time Epoch E +1 Local Blockchain
746
752
754

FIG. 7B

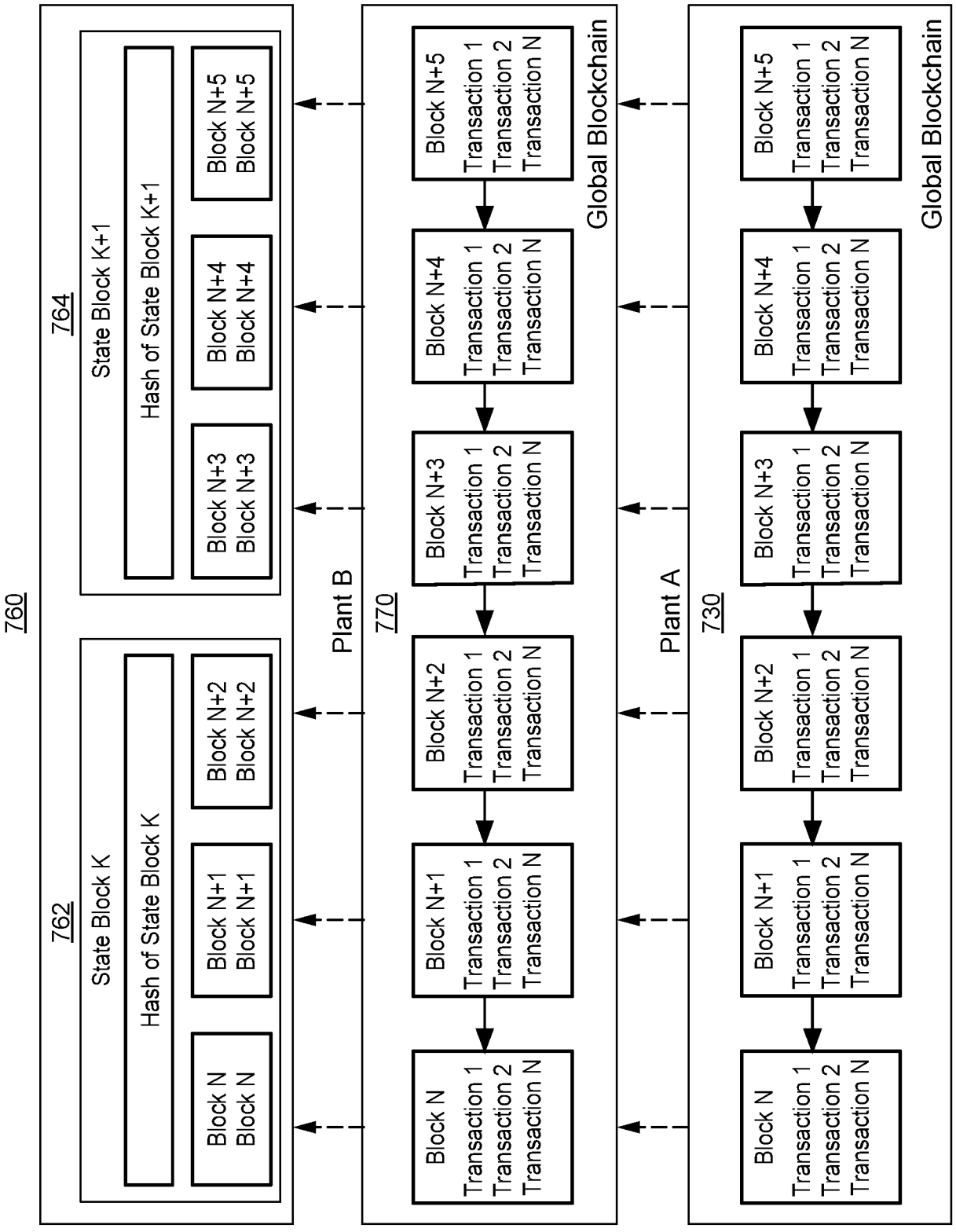
760
762
State Block K
Hash of State Block K
Block N
Block N
Block N+1
Block N+1
Block N+2
Block N+2
764
State Block K+1
Hash of State Block K+1
Block N+3
Block N+3
Block N+4
Block N+4
Block N+5
Block N+5
Plant B
770
Global Blockchain
Block N
Transaction 1
Transaction 2
Transaction N
Block N+1
Transaction 1
Transaction 2
Transaction N
Block N+2
Transaction 1
Transaction 2
Transaction N
Block N+3
Transaction 1
Transaction 2
Transaction N
Block N+4
Transaction 1
Transaction 2
Transaction N
Block N+5
Transaction 1
Transaction 2
Transaction N
Plant A
730
Global Blockchain
Block N
Transaction 1
Transaction 2
Transaction N
Block N+1
Transaction 1
Transaction 2
Transaction N
Block N+2
Transaction 1
Transaction 2
Transaction N
Block N+3
Transaction 1
Transaction 2
Transaction N
Block N+4
Transaction 1
Transaction 2
Transaction N
Block N+5
Transaction 1
Transaction 2
Transaction N

FIG. 7C

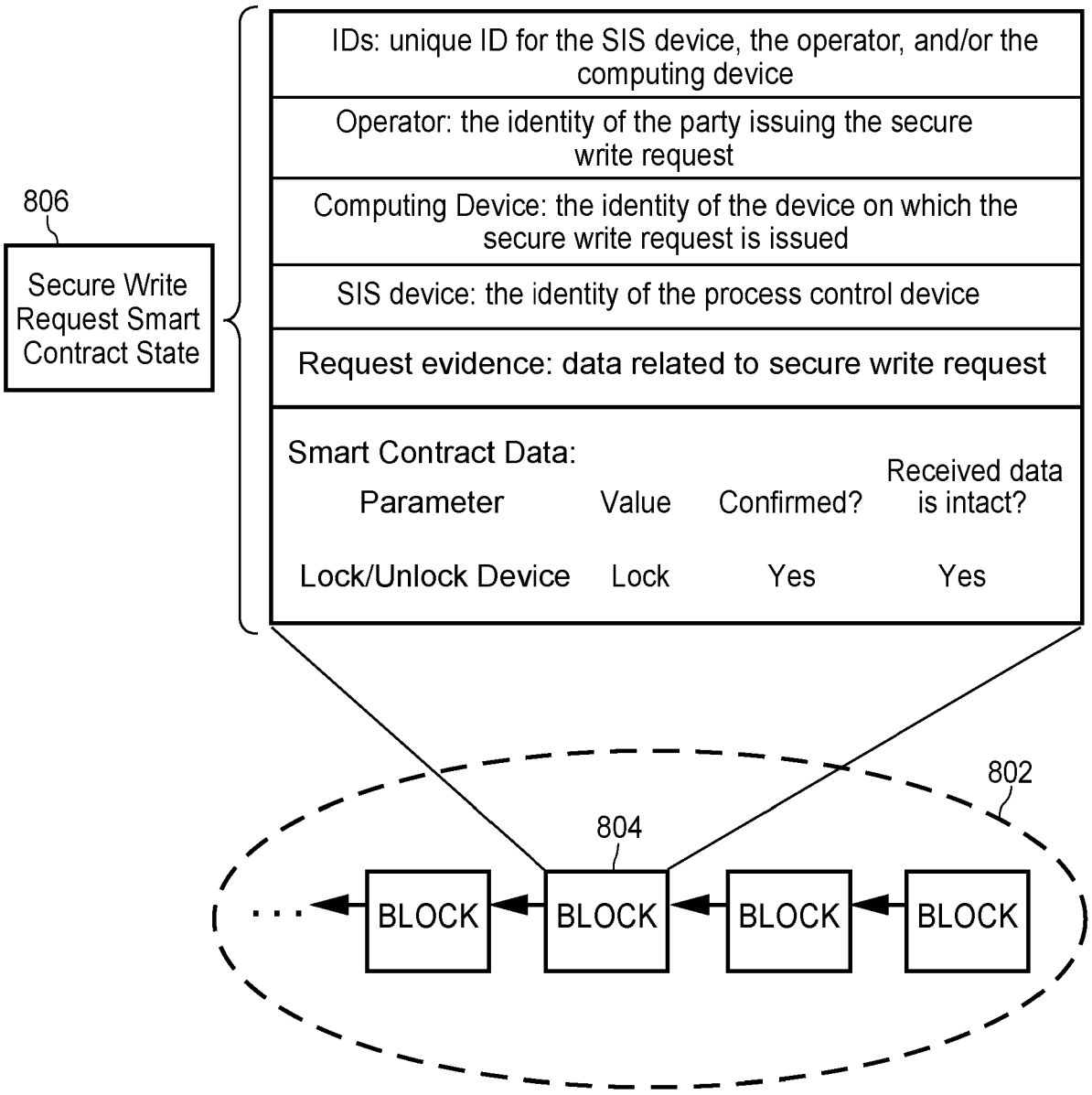
806
Secure Write Request Smart Contract State
IDs: unique ID for the SIS device, the operator, and/or the computing device
Operator: the identity of the party issuing the secure write request
Computing Device: the identity of the device on which the secure write request is issued
SIS device: the identity of the process control device
Request evidence: data related to secure write request
Smart Contract Data:
Parameter
Value
Confirmed?
Received data is intact?
Lock/Unlock Device
Lock
Yes
Yes
802
804
BLOCK
BLOCK
BLOCK
BLOCK

FIG. 8

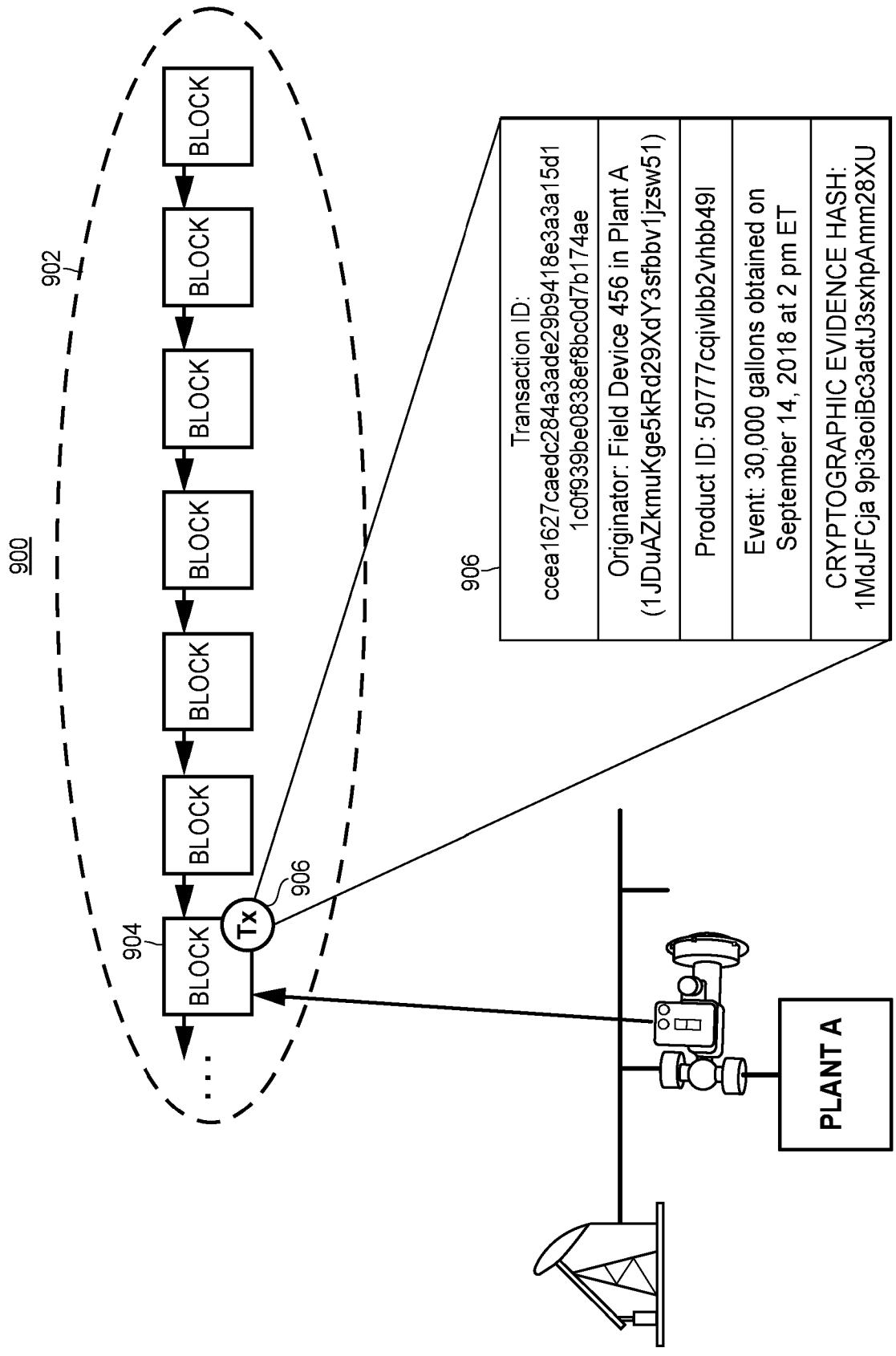
900
902
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
904
Tx
906
906
Transaction ID:
ccea1627caedc284a3ade29b9418e3a3a15d1
1c0f939be0838ef8bc0d7b174ae
Originator: Field Device 456 in Plant A
(1JDuAZkmuKge5kRd29XdY3sfbbv1jzsw51)
Product ID: 50777cqivlbb2vhbb49l
Event: 30,000 gallons obtained on
September 14, 2018 at 2 pm ET
CRYPTOGRAPHIC EVIDENCE HASH:
1MdJFCja 9pi3eoiBc3adtJ3sxhpAmm28XU
PLANT A

FIG. 9

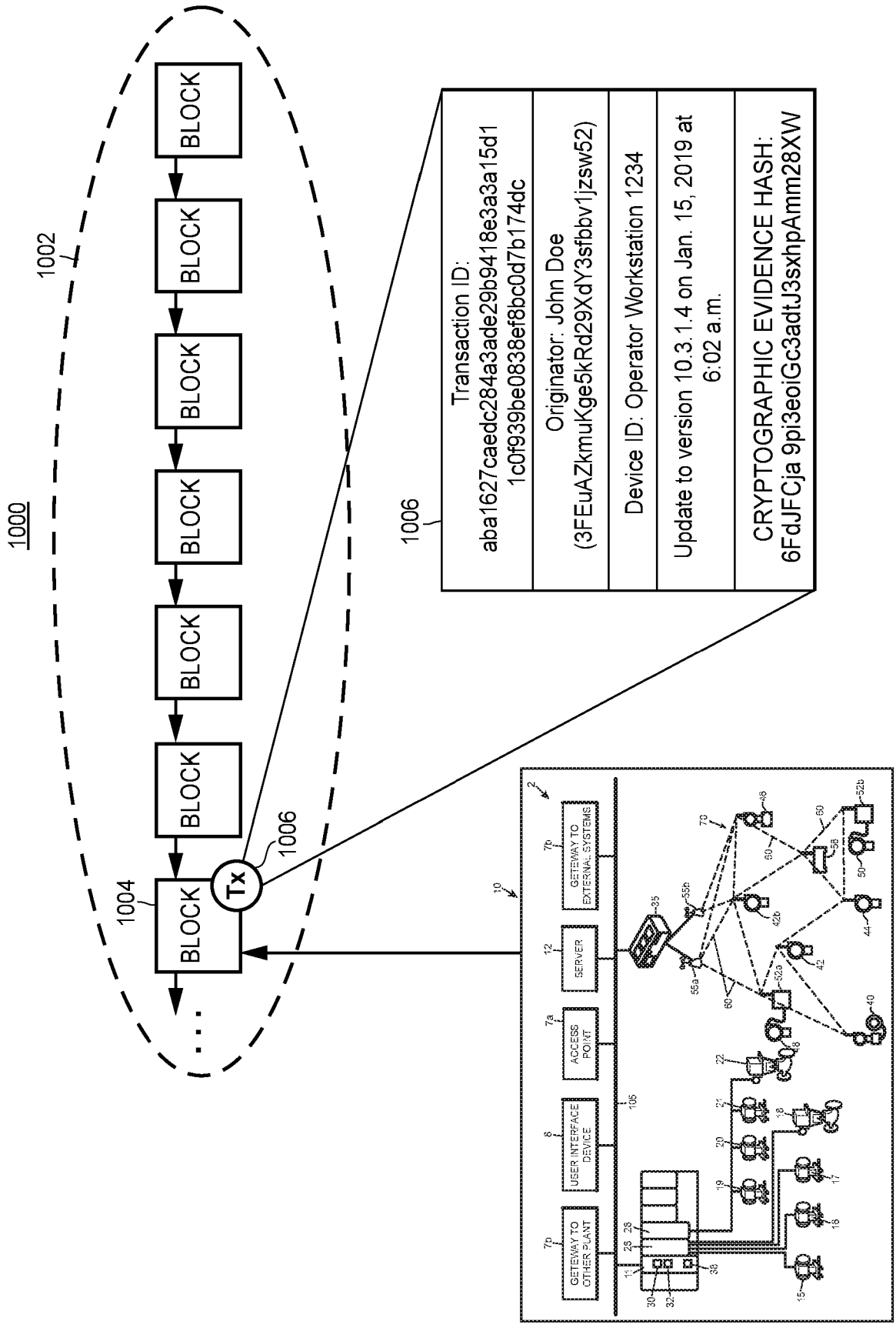
1000
1002
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
1004
Tx
1006
1006
Transaction ID:
aba1627caedc284a3ade29b9418e3a3a15d1
1c0f939be0838ef8bc0d7b174dc
Originator: John Doe
(3FEuAZkmuKge5kRd29XdY3sfbbv1jzsw52)
Device ID: Operator Workstation 1234
Update to version 10.3.1.4 on Jan. 15, 2019 at
6:02 a.m.
CRYPTOGRAPHIC EVIDENCE HASH:
6FdJFCja 9pi3eoiGc3adtJ3sxhpAmm28XW
GETEWAY TO OTHER PLANT
USER INTERFACE DEVICE
ACCESS POINT
SERVER
GETEWAY TO EXTERNAL SYSTEMS

FIG. 10

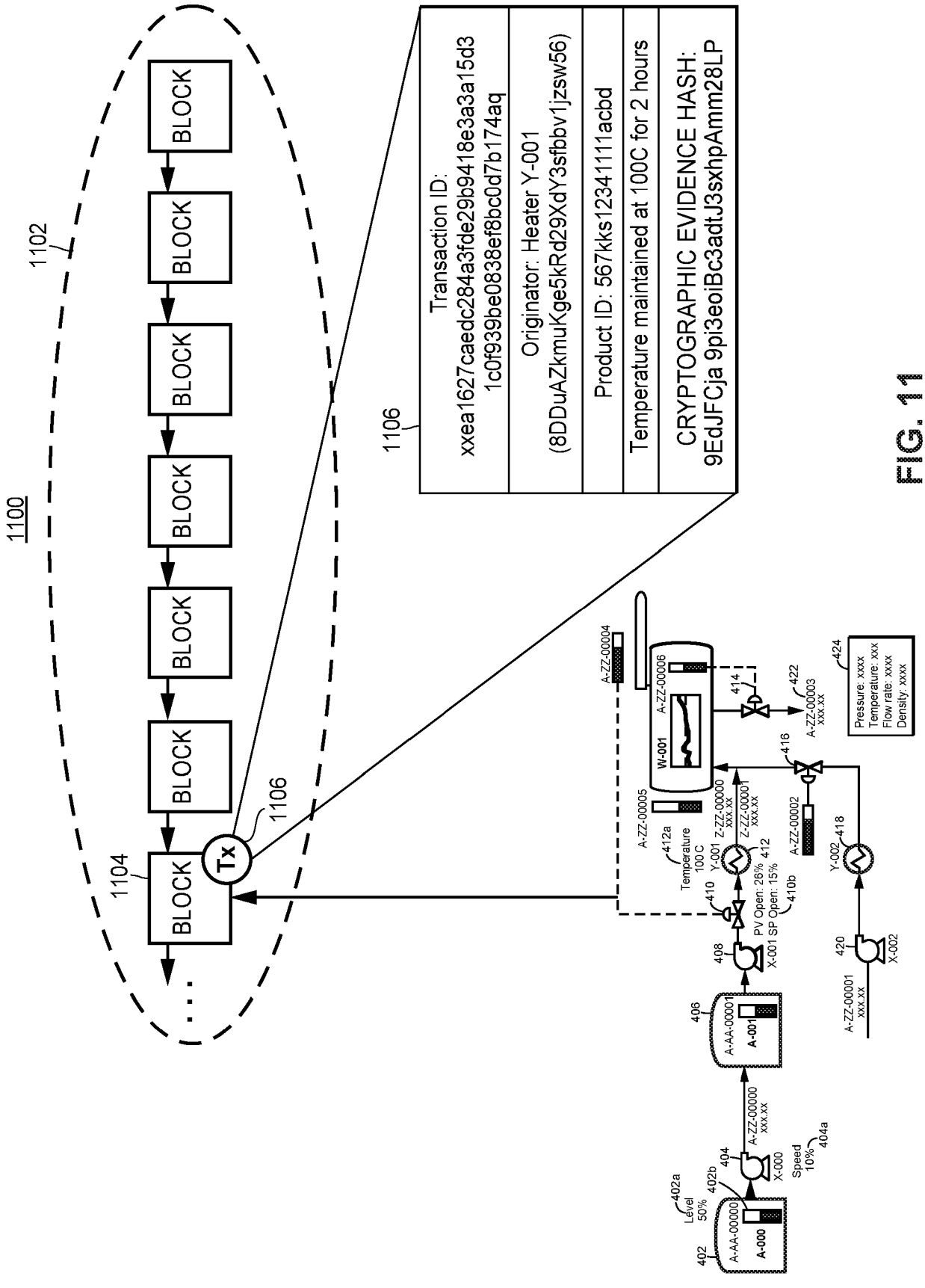
1100
1102
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
BLOCK
1104
Tx
1106
1106
Transaction ID:
xxea1627caedc284a3fde29b9418e3a3a15d3
1c0f939be0838ef8bc0d7b174aq
Originator: Heater Y-001
(8DDuAZkmuKge5kRd29XdY3sfbbv1jzsw56)
Product ID: 567kks12341111acbd
Temperature maintained at 100C for 2 hours
CRYPTOGRAPHIC EVIDENCE HASH:
9EdJFCja 9pi3eoiBc3adtJ3sxhpAmm28LP
402
A-AA-00000
A-000
402a
Level
50%
402b
404
X-000
A-ZZ-00000
xxx.xx
404a
Speed
10%
406
A-AA-0000
A-001
408
X-001
PV Open: 26%
SP Open: 15%
410
410b
412
Y-001
Z-ZZ-00000
xxx.xx
Z-ZZ-00001
xxx.xx
412a
Temperature
100 C
A-ZZ-00004
A-ZZ-00005
W-001
A-ZZ-00006
414
416
A-ZZ-00002
418
Y-002
420
X-002
A-ZZ-00001
xxx.xx
422
A-ZZ-00003
xxx.xx
424
Pressure: xxxx
Temperature: xxx
Flow rate: xxxx
Density: xxxx

FIG. 11

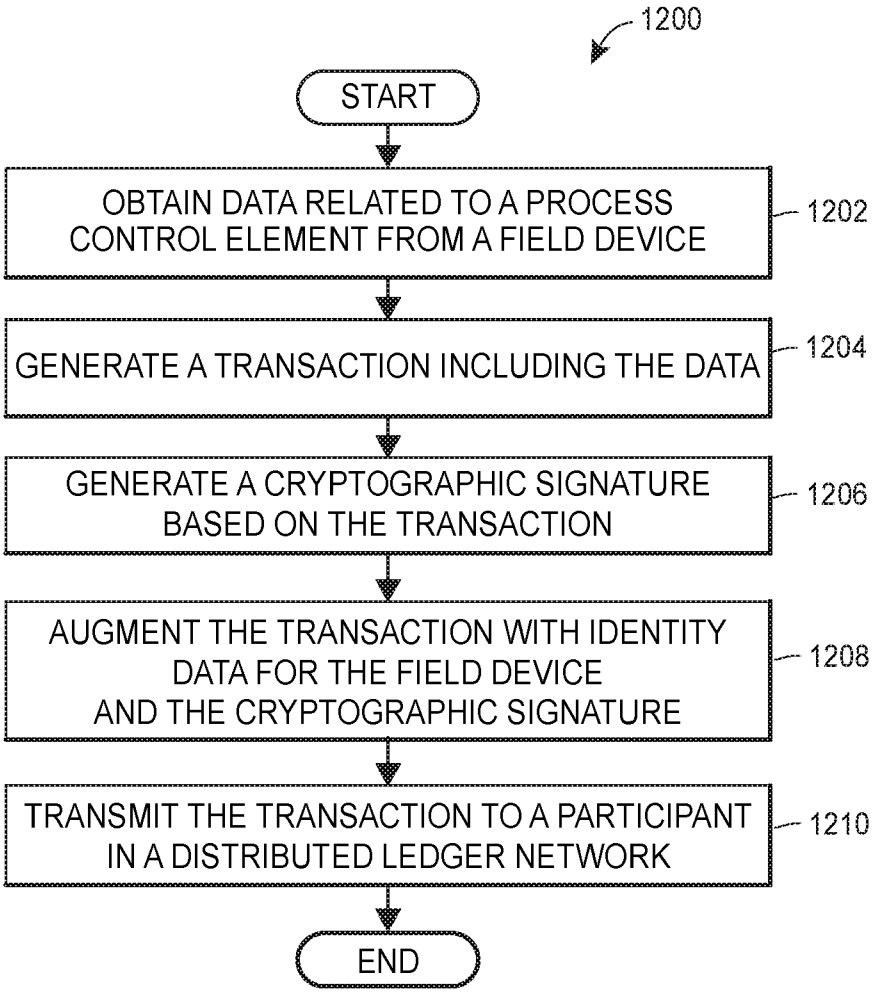
1200
START
OBTAIN DATA RELATED TO A PROCESS CONTROL ELEMENT FROM A FIELD DEVICE
1202
GENERATE A TRANSACTION INCLUDING THE DATA
1204
GENERATE A CRYPTOGRAPHIC SIGNATURE BASED ON THE TRANSACTION
1206
AUGMENT THE TRANSACTION WITH IDENTITY DATA FOR THE FIELD DEVICE AND THE CRYPTOGRAPHIC SIGNATURE
1208
TRANSMIT THE TRANSACTION TO A PARTICIPANT IN A DISTRIBUTED LEDGER NETWORK
1210
END

FIG. 12

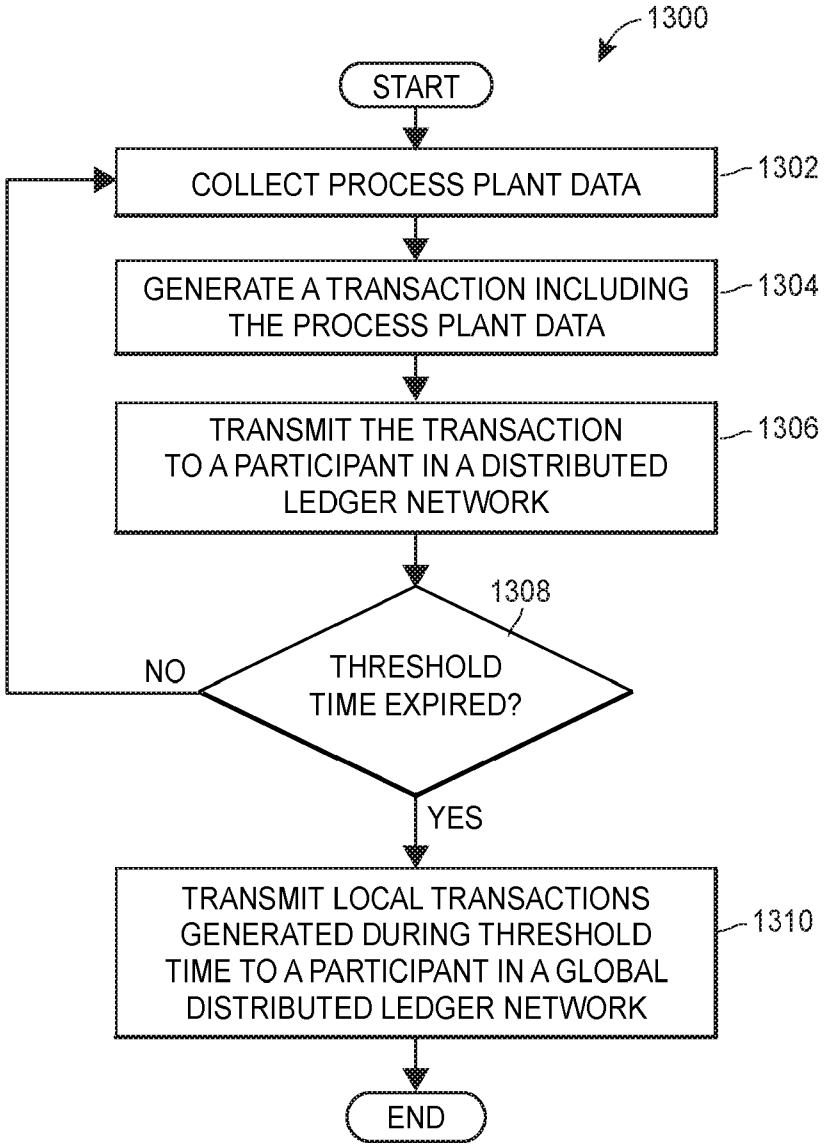
1300
START
COLLECT PROCESS PLANT DATA
1302
GENERATE A TRANSACTION INCLUDING THE PROCESS PLANT DATA
1304
TRANSMIT THE TRANSACTION TO A PARTICIPANT IN A DISTRIBUTED LEDGER NETWORK
1306
1308
THRESHOLD TIME EXPIRED?
NO
YES
TRANSMIT LOCAL TRANSACTIONS GENERATED DURING THRESHOLD TIME TO A PARTICIPANT IN A GLOBAL DISTRIBUTED LEDGER NETWORK
1310
END

FIG. 13

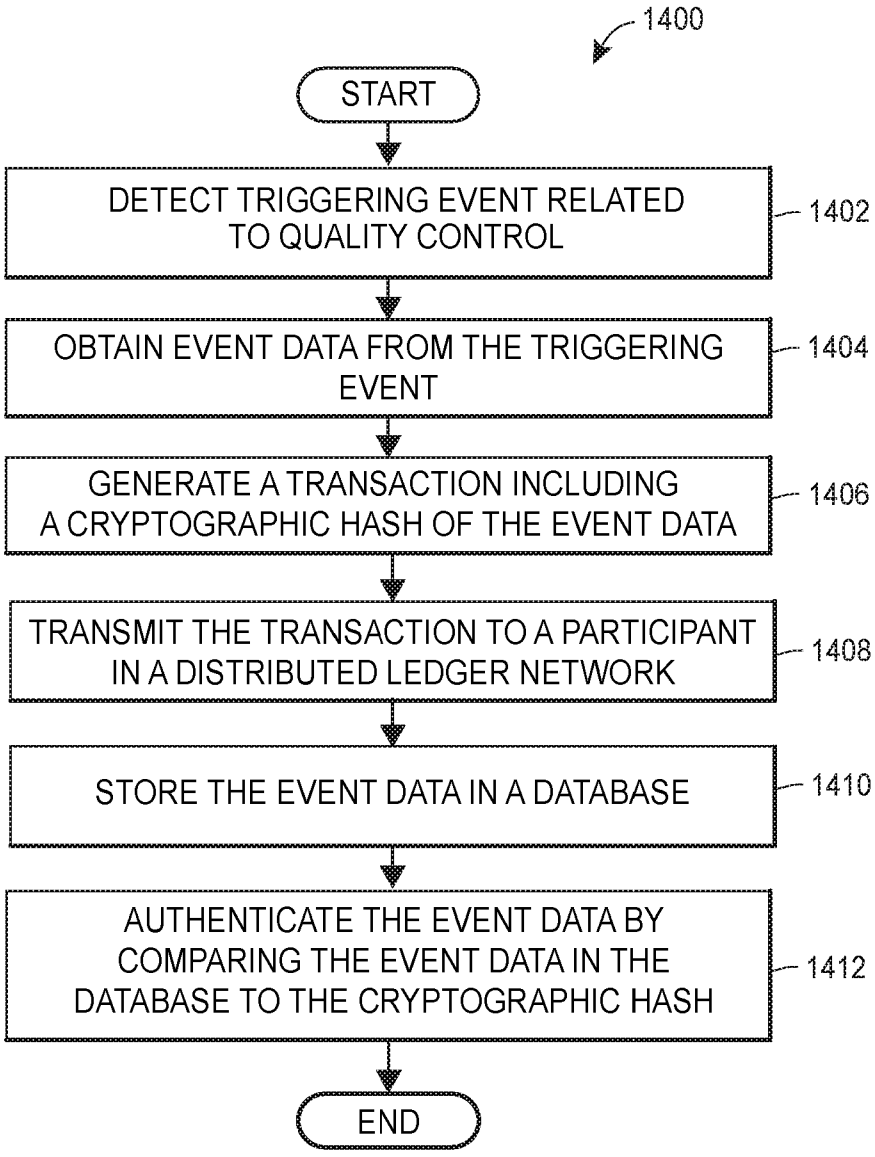
1400
START
DETECT TRIGGERING EVENT RELATED TO QUALITY CONTROL
1402
OBTAIN EVENT DATA FROM THE TRIGGERING EVENT
1404
GENERATE A TRANSACTION INCLUDING A CRYPTOGRAPHIC HASH OF THE EVENT DATA
1406
TRANSMIT THE TRANSACTION TO A PARTICIPANT IN A DISTRIBUTED LEDGER NETWORK
1408
STORE THE EVENT DATA IN A DATABASE
1410
AUTHENTICATE THE EVENT DATA BY COMPARING THE EVENT DATA IN THE DATABASE TO THE CRYPTOGRAPHIC HASH
1412
END

FIG. 14

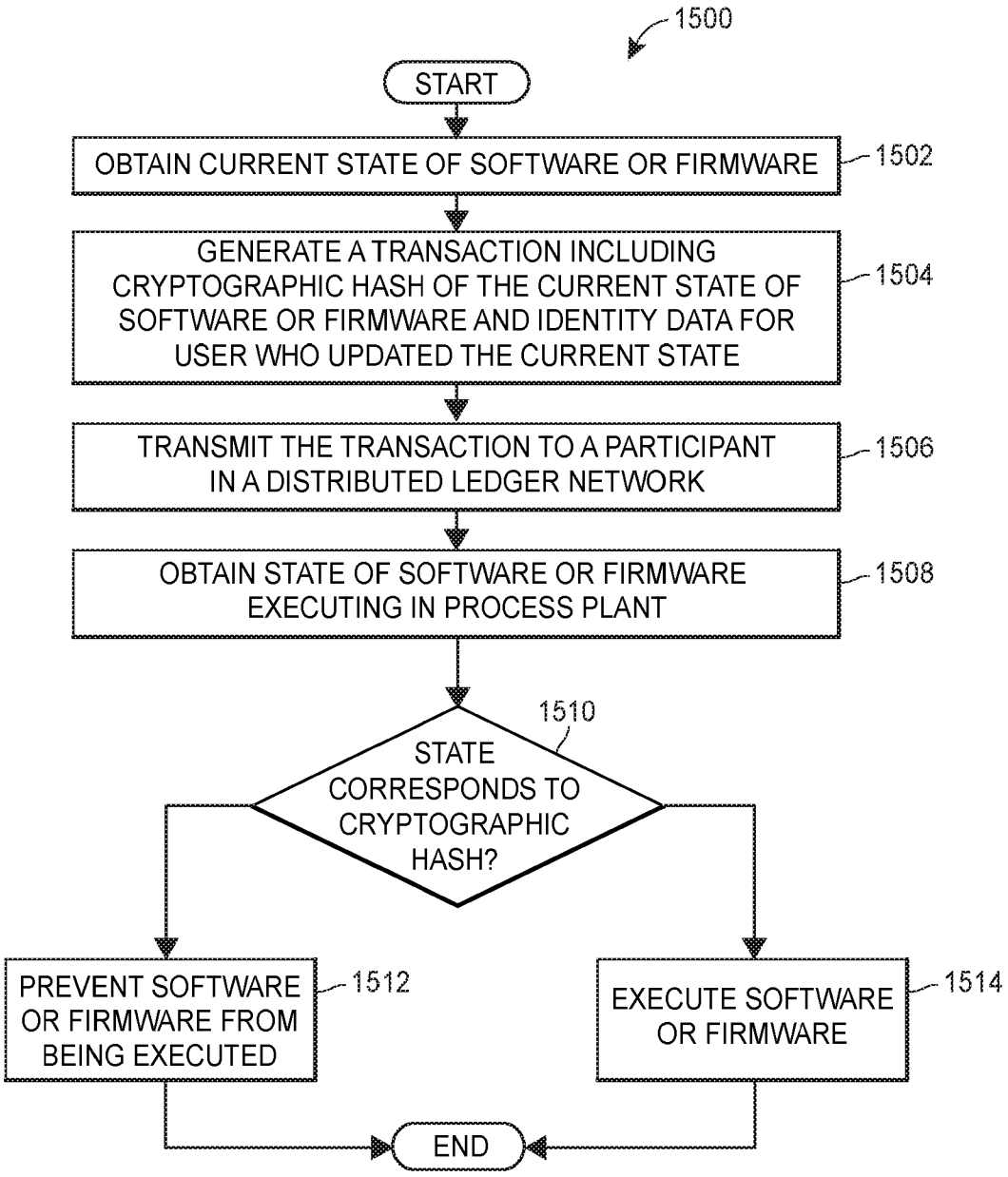
1500
START
OBTAIN CURRENT STATE OF SOFTWARE OR FIRMWARE
1502
GENERATE A TRANSACTION INCLUDING CRYPTOGRAPHIC HASH OF THE CURRENT STATE OF SOFTWARE OR FIRMWARE AND IDENTITY DATA FOR USER WHO UPDATED THE CURRENT STATE
1504
TRANSMIT THE TRANSACTION TO A PARTICIPANT IN A DISTRIBUTED LEDGER NETWORK
1506
OBTAIN STATE OF SOFTWARE OR FIRMWARE EXECUTING IN PROCESS PLANT
1508
1510
STATE CORRESPONDS TO CRYPTOGRAPHIC HASH?
PREVENT SOFTWARE OR FIRMWARE FROM BEING EXECUTED
1512
EXECUTE SOFTWARE OR FIRMWARE
1514
END

FIG. 15

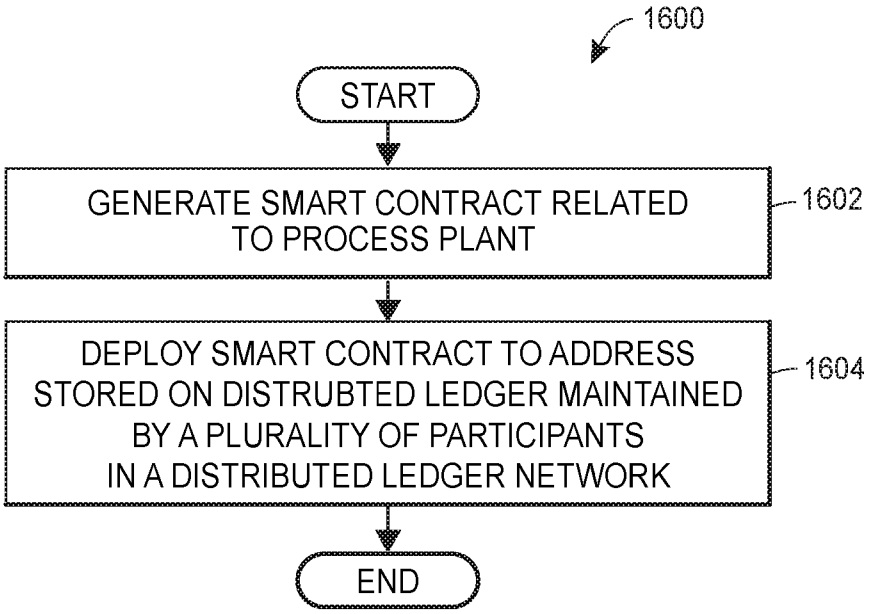
1600
START
GENERATE SMART CONTRACT RELATED TO PROCESS PLANT
1602
DEPLOY SMART CONTRACT TO ADDRESS STORED ON DISTRUBTED LEDGER MAINTAINED BY A PLURALITY OF PARTICIPANTS IN A DISTRIBUTED LEDGER NETWORK
1604
END

FIG. 16

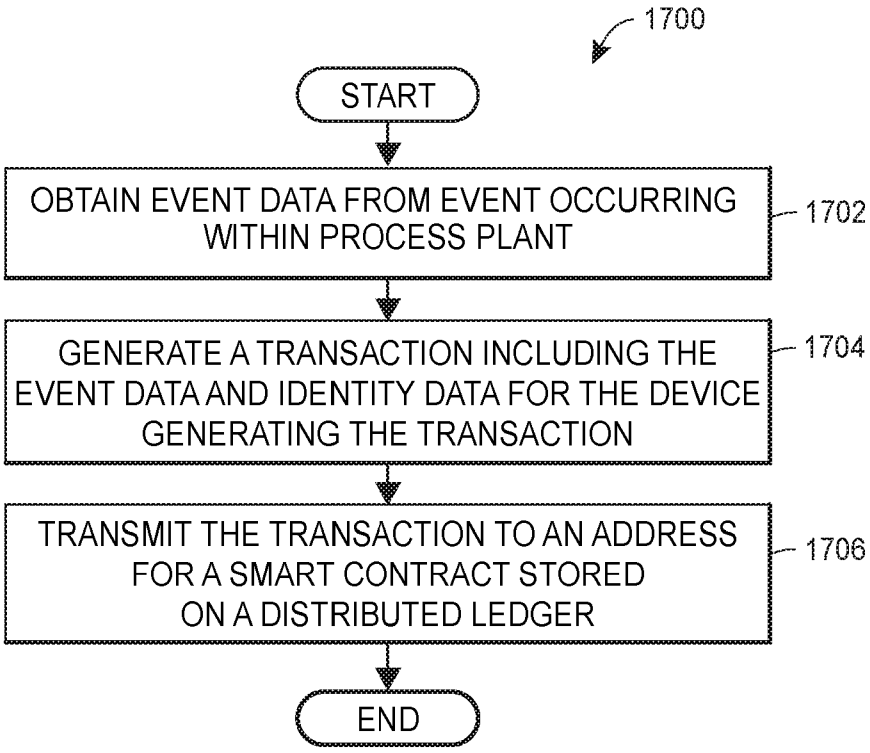
1700
START
OBTAIN EVENT DATA FROM EVENT OCCURRING WITHIN PROCESS PLANT
1702
GENERATE A TRANSACTION INCLUDING THE EVENT DATA AND IDENTITY DATA FOR THE DEVICE GENERATING THE TRANSACTION
1704
TRANSMIT THE TRANSACTION TO AN ADDRESS FOR A SMART CONTRACT STORED ON A DISTRIBUTED LEDGER
1706
END

FIG. 17

SYSTEM FOR SECURE METERING FROM SYSTEMS OF UNTRUSTED DATA SOURCES DERIVED FROM COMMON SOURCES

[0001] The present disclosure relates generally to process plants and to process control systems, and more particularly, to the use of distributed ledgers in process control systems to record data and events.

[0002] Distributed process control systems, like those used in chemical, petroleum or other process plants, typically include one or more process controllers communicatively coupled to one or more field devices via analog, digital or combined analog/digital buses, or via a wireless communication link or network. The field devices, which may be, for example, valves, valve positioners, switches and transmitters (e.g., temperature, pressure, level and flow rate sensors), are located within the process environment and generally perform physical or process control functions such as opening or closing valves, measuring process parameters such as pressure, temperature, etc., and the like to control one or more process executing within the process plant or system. Smart field devices, such as the field devices conforming to the well-known Fieldbus protocol, may also perform control calculations, alarming functions, and other control functions commonly implemented within the controller. The process controllers, which are also typically located within the plant environment, receive signals indicative of process measurements made by the field devices and/or other information pertaining to the field devices and execute a controller application that runs, for example, different control modules which make process control decisions, generate control signals based on the received information and coordinate with the control modules or blocks being performed in the field devices, such as HART (RTM), WirelessHART (RTM), and FOUNDATION (RTM) Fieldbus field devices. The control modules in the controller send the control signals over the communication lines or links to the field devices to thereby control the operation of at least a portion of the process plant or system. As utilized herein, field devices and controllers are generally referred to as "process control devices."

[0003] Information from the field devices and the controller is usually made available over a data highway to one or more other hardware devices, such as operator workstations, personal computers or computing devices, data historians, report generators, centralized databases, or other centralized administrative computing devices that are typically placed in control rooms or other locations away from the harsher plant environment. Each of these hardware devices typically is centralized across the process plant or across a portion of the process plant. These hardware devices run applications that may, for example, enable an operator to perform functions with respect to controlling a process and/or operating the process plant, such as changing settings of the process control routine, modifying the operation of the control modules within the controllers or the field devices, viewing the current state of the process, viewing alarms generated by field devices and controllers, simulating the operation of the process for the purpose of training personnel or testing the process control software, keeping and updating a configuration database, etc. The data highway utilized by the hardware devices, controllers and field devices may include a wired communication path, a wireless communication path, or a combination of wired and wireless communication paths.

[0004] As an example, the DeltaV^{TM} control system, sold by Emerson Process Management, includes multiple applications stored within and executed by different devices located at diverse places within a process plant. A configuration application, which resides in one or more workstations or computing devices, enables

users to create or change process control modules and download these process control modules via a data highway to dedicated distributed controllers. Typically, these control modules are made up of communicatively interconnected function blocks, which are objects in an object oriented programming protocol that perform functions within the control scheme based on inputs thereto and that provide outputs to other function blocks within the control scheme. The configuration application may also allow a configuration designer to create or change operator interfaces which are used by a viewing application to display data to an operator and to enable the operator to change settings, such as set points, within the process control routines. Each dedicated controller and, in some cases, one or more field devices, stores and executes a respective controller application that runs the control modules assigned and downloaded thereto to implement actual process control functionality. The viewing applications, which may be executed on one or more operator workstations (or on one or more remote computing devices in communicative connection with the operator workstations and the data highway), receive data from the controller application via the data highway and display this data to process control system designers, operators, or users using the user interfaces, and may provide any of a number of different views, such as an operator's view, an engineer's view, a technician's view, etc. A data historian application is typically stored in and executed by a data historian device that collects and stores some or all of the data provided across the data highway while a configuration database application may run in a still further computer attached to the data highway to store the current process control routine configuration and data associated therewith. Alternatively, the configuration database may be located in the same workstation as the configuration application.

[0005] Generally speaking, a process control system of a process plant includes field devices, controllers, workstations, and other devices that are interconnected by a set of layered networks and buses. The process control system may, in turn, be connected with various business and external networks, e.g., to reduce manufacturing and operational costs, enhance productivity and efficiencies, provide timely access to process control and/or process plant information, etc. On the other hand, the interconnection of process plants and/or process control systems to enterprise and/or external networks and systems increases the risk of cyber intrusions and/or malicious cyber attacks that may arise from expected vulnerabilities in commercial systems and applications, such as those used in enterprise and/or external networks. Cyber intrusions and malicious cyber attacks of process plants, networks, and/or control systems may negatively affect the confidentiality, integrity, and/or availability of information assets, which, generally speaking, are vulnerabilities similar to those of general purpose computing networks. However, unlike general purpose computer networks, cyber intrusions of process plants, networks, and/or control systems may also lead to damage, destruction, and/or loss of not only plant equipment, product, and other physical assets, but also to the loss of human life. For example, a cyber intrusion may cause a process to become uncontrolled, and thereby produce explosions, fires, floods, exposure to hazardous materials, etc. Thus, securing communications related to process control plants and systems is of paramount importance.

SUMMARY

[0006] Techniques, systems, apparatuses, components, devices, and methods are disclosed for utilizing a distributed ledger, or blockchain, in process control systems. Said techniques, systems, apparatuses,

components, devices, and methods may apply to industrial process control systems, environments, and/or plants, which are interchangeably referred to herein as "industrial control," "process control," or "process" systems, environments, and/or plants. Typically, such systems and plants provide control, in a distributed manner, of one or more processes that operate to manufacture, refine, transform, generate, or produce physical materials or products.

[0007] For example, in a process control system a distributed ledger may be maintained by nodes referred to herein as "edge gateways." The nodes receive transactions broadcasted to a distributed ledger network from field devices, controllers, operator workstations, or other devices operating within the process plant. In some scenarios, the transactions include process parameter values for process parameters corresponding to a process plant entity. A process plant entity may include devices within a process plant for use in a portion of the process which contain, transform, generate, or transfer physical materials, such as a valve, a tank, a mixer, a pump, a heat exchanger, etc. The transactions may also include product parameter values such as properties of a physical material or product produced by the process plant, including a temperature of the product, a volume of the product, a mass of the product, a density of the product, a pressure of the product, etc.

[0008] The recorded process parameter values and product parameter values may then be retrieved to verify the quality of a product. For example, a first process plant may manufacture, refine, transform, generate, or produce a product which is then shipped to a second process plant. The second process plant may determine that the product meets certain quality standards by retrieving the recorded process parameter values and product parameter values from the distributed ledger. Additionally, regulatory data may be recorded in the distributed ledger. For example, in response to a triggering event such as an alarm, an error, a leak, a repair event, a process milestone, a corrective action, etc., process control elements such as field devices or controllers may generate transactions including data from the triggering event, such as the time in which the event occurred, the duration of the event, process parameter values for process plant entities involved in the event, product parameter values for products involved in the event, etc. The regulatory data is then recorded in the distributed ledger, so that regulatory agencies can review the data.

[0009] Still further, distributed ledgers may be utilized to execute smart contracts, described in more detail below. Process control systems can deploy smart contracts to the distributed ledger to exchange value, for example upon receiving a product in good condition. Smart contracts may also be deployed to the distributed ledger to allow machines such as field devices to transact by themselves without human intervention. For example, according to the terms of a smart contract, a computing device in a first process plant may automatically provide a predetermined token amount to a computing device in a second process plant upon receiving indications from one or more field devices in the first process plant that a product has been delivered from the second process plant and the product meets certain quality standards. Smart contracts may also be utilized in process plants for a multitude of other applications, described in more detail below.

[0010] By utilizing distributed ledgers and in some scenarios, smart contracts in process plants, each process plant or a network of process plants may provide a trusted, secure, and immutable record of transactions within the process plant. The secure, immutable, and trustless nature of distributed ledgers is particularly important in process control systems where cyber intrusions may lead to damage, destruction, and/or loss of not only plant

equipment, product, and other physical assets, but also to the loss of human life. Additionally, distributed ledgers allow process plants to track product lineage from raw materials to finished products and to further track the products after they have been manufactured. Moreover, when competing entities utilize or transfer a common resource, distributed ledgers can be used to determine the amount of the resource utilized by one of the entities and fair compensation to the competing entity for the use of the resource. For example, an oil refinery may produce oil that is provided via an oil pipeline to several entities or process plants. Each process plant is responsible for compensating the oil refinery for the amount of oil in which the process plant received from the oil pipeline. Distributed ledgers can be used to record the amount of oil each process plant received from devices measuring the amount of oil at the time the oil is provided. Due to the difficulty of changing the recorded data in the distributed ledgers, competing entities do not have to trust that the data is reliable.

[0011] An aspect provides a method for secure metering of untrusted data in process control systems using a distributed ledger maintained by a plurality of participants, the method comprising: collecting, by a field device performing a physical function to control an industrial process in a process plant, a measurement of a parameter within the process plant; obtaining, by a computing device, the measurement of the parameter; generating a transaction including the measurement; and transmitting the transaction to at least one other participant in a local distributed ledger network of participants maintaining a local distributed ledger; after a threshold time period, transmitting a plurality of transactions generated during the threshold time period to at least one participant in a global distributed ledger network of participants maintaining a global distributed ledger, wherein the plurality of transactions includes the transaction which includes the measurement.

[0012] The method may further comprise: adding the transaction to a local block of transactions; solving a cryptographic puzzle based on the local block of transactions; adding the solution to the cryptographic puzzle to the local block of transactions; and transmitting the local block of transactions to at least one other participant in the local distributed ledger network.

[0013] The method may further comprise: after the threshold time period, transmitting one or more local blocks of transactions generated during the threshold time period to at least one participant in the global distributed ledger network.

[0014] The method may further comprise: after the threshold time period, pruning at least some of the plurality of transaction generated during the threshold time period from the local distributed ledger network.

[0015] The global distributed ledger may be a permissioned blockchain viewable by a plurality of entities operating a plurality of process plants.

[0016] The parameter may be related to a shared resource between the plurality of entities operating the plurality of process plants.

[0017] The global distributed ledger may include a plurality of global distributed ledgers corresponding to the plurality of entities, each global distributed ledger including transactions stored in the local distributed ledger for a same respective entity as the global distributed ledger.

[0018] The method may further comprise: for transactions generated during the threshold time period, adding the transaction from each of the plurality of global distributed ledgers to a state block of transactions; solving a cryptographic puzzle based on the state block of transactions; adding the solution to the cryptographic puzzle to the state block of transactions; and transmitting the state block of transactions to at least one other participant in a super blockchain network of participants maintaining a super blockchain.

[0019] The local distributed ledger may be a private blockchain viewable by an entity operating the process plant.

[0020] Generating a transaction including the measurement may include generating the transaction including a cryptographic hash value corresponding to the measurement.

[0021] The shared resource between the plurality of entities operating the plurality of process plants may be a fluid in a fluid pipeline, and the parameter measurement may be an amount of fluid obtained by one of the plurality of entities from the fluid pipeline.

[0022] Another aspect provides a system for secure metering of untrusted data in process control systems using a distributed ledger maintained by a plurality of participants comprising: one or more field devices disposed in a process plant each performing a physical function to control an industrial process, the one or more field devices configured to collect measurements of parameters within the process plant and provide the parameter measurements to one or more edge gateway devices; and the one or more edge gateway devices executing in the process plant each including: one or more processors; a communication unit; and a non-transitory computer-readable medium coupled to the one or more processors and the communication unit and storing instructions thereon, that when executed by the one or more processors, causes the edge gateway device to: obtain at least one of the parameter measurements; generate a transaction including the measurement; and transmit the transaction to at least one other edge gateway in a local distributed ledger network of edge gateways maintaining a local distributed ledger; and after a threshold time period, transmit a plurality of transactions generated during the threshold time period to at least one participant in a global distributed ledger network of participants maintaining a global distributed ledger, wherein the plurality of transactions includes the transaction which includes the measurement.

[0023] The instructions may further cause the edge gateway to: add the transaction to a local block of transactions; solve a cryptographic puzzle based on the local block of transactions; add the solution to the cryptographic puzzle to the local block of transactions; and transmit the local block of transactions to at least one other edge gateway in the local distributed ledger network.

[0024] The instructions may further cause the edge gateway to: after the threshold time period, transmit one or more local blocks of transactions generated during the threshold time period to at least one participant in the global distributed ledger network.

[0025] The instructions may further cause the edge gateway to: after the threshold time period, prune at least some of the plurality of transactions generated during the threshold time period from the local distributed ledger network.

[0026] The global distributed ledger may be a permissioned blockchain viewable by a plurality of entities operating a plurality of process plants.

[0027] The parameter may be related to a shared resource between the plurality of entities operating the plurality of process plants.

[0028] The global distributed ledger may include a plurality of global distributed ledgers corresponding to the plurality of entities, each global distributed ledger including transactions stored in the local distributed ledger for a same respective entity as the global distributed ledger.

[0029] The system may further comprise: a computing device in a global distributed ledger network maintaining a global distributed ledger including: one or more processors; a communication unit; and a non-transitory computer-readable medium coupled to the one or more processors and the communication unit and storing instructions thereon, that when executed by the one or more processors, causes the computing device to: for transactions generated during the threshold time period, add the transaction from each of the plurality of global distributed ledgers to a state block of transactions; solve a cryptographic puzzle based on the state block of transactions; add the solution to the cryptographic puzzle to the state block of transactions; and transmit the state block of transactions to at least one other participant in a super blockchain network of participants maintaining a super blockchain.

[0030] The local distributed ledger may be a private blockchain viewable by an entity operating the process plant.

[0031] The transaction may include a cryptographic hash value corresponding to the measurement.

[0032] The shared resource between the plurality of entities operating the plurality of process plants may be a fluid in a fluid pipeline, and the parameter measurement may be an amount of fluid obtained by one of the plurality of entities from the fluid pipeline.

[0033] Another aspect provides a validating network node in a process plant on a local distributed ledger network comprising: a transceiver configured to (i) communicate with one or more field devices each performing a physical function to control an industrial process in the process plant and collecting measurements of parameters within the process plant, and to (ii) exchange local distributed ledger data with peer network nodes, the local distributed ledger data including transactions having parameter measurements; a storage media configured to store a copy of the local distributed ledger; and a process data validator configured to apply a set of consensus rules to the distributed ledger data received from the peer network nodes, the process data validator being further configured to append the distributed ledger data received from the peer network nodes to the copy of the distributed ledger if the distributed ledger data satisfies the consensus rules, wherein after a threshold time period, the transceiver is configured to transmit a plurality of transactions generated during the threshold time period to at least one participant in a global distributed ledger network of participants maintaining a global distributed ledger, wherein the plurality of transactions includes the transactions having parameter measurements.

[0034] After the threshold time period, the validating network node may be configured to prune at least some of the plurality of transactions generated during the threshold time period from the copy of the local distributed ledger.

[0035] The global distributed ledger may be a permissioned blockchain viewable by a plurality of entities operating a plurality of process plants.

[0036] At least one of the parameters may be related to a shared resource between the plurality of entities operating the plurality of process plants.

[0037] The global distributed ledger may include a plurality of global distributed ledgers corresponding to the plurality of entities, each global distributed ledger including transactions stored in the local distributed ledger for a same respective entity as the global distributed ledger.

[0038] The local distributed ledger may be a private blockchain viewable by an entity operating the process plant.

[0039] A transaction may include a cryptographic hash value corresponding to a parameter measurement.

[0040] Embodiments are described, by way of example only, with reference to the accompanying drawings, in which:

[0041] FIG. 1 is a block diagram of an example process plant or process control system, that illustrates, *inter alia*, interconnections between various example components of the process control system, the process control system itself, and other example systems and/or networks;

[0042] FIG. 2 is a block diagram of an example security architecture for a process plant or process control system;

[0043] FIG. 3 is an exemplary distributed ledger system for recording transactions and executing smart contracts in a process control system;

[0044] FIG. 4 illustrates exemplary validating network nodes and an exemplary transaction flow on a distributed ledger network in a process control system;

[0045] FIG. 5 illustrates exemplary components of a network node on a distributed ledger network in a process control system;

[0046] FIG. 6A illustrates an example distributed ledger including a blockchain having blocks of transactions in a process control system;

[0047] FIG. 6B illustrates another example distributed ledger including multiple side blockchains or side chains maintained by different process plants and a main blockchain maintained by several process plants that incorporates transaction data from the side chains;

[0048] FIG. 7A illustrates yet another example distributed ledger including multiple local blockchains each maintained by a different process plant;

[0049] FIG. 7B illustrates a global blockchain for a process plant that is maintained by several process plants and that incorporates blocks from the local blockchain;

[0050] FIG. 7C illustrates a super blockchain maintained by several process plants that incorporates blocks from each of the global blockhains for each process plant;

[0051] FIG. 8 illustrates an exemplary smart contract state in a distributed ledger network for performing secure write operations in a process plant to write a process parameter to a safety instrumented system (SIS) device;

[0052] FIG. 9 illustrates an exemplary transaction representing an evidence transaction generated by an evidence oracle which is a field device reporting the amount of oil received from an oil pipeline;

[0053] FIG. 10 illustrates an exemplary transaction representing an evidence transaction generated by an evidence oracle which is a computing device reporting a software or firmware update;

[0054] FIG. 11 illustrates an exemplary transaction representing an evidence transaction generated by an evidence oracle which is a process plant entity reporting process parameter or product parameter data;

[0055] FIG. 12 illustrates a flow diagram representing an exemplary method for recording data in a process control system using a distributed ledger;

[0056] FIG. 13 illustrates a flow diagram representing an exemplary method for secure metering of untrusted data in process control systems using a distributed ledger;

[0057] FIG. 14 illustrates a flow diagram representing an exemplary method for recording quality control, production, or regulatory data in a process control system using a distributed ledger;

[0058] FIG. 15 illustrates a flow diagram representing an exemplary method for recording states of software or firmware in a process control system and connected instrumentation using a distributed ledger;

[0059] FIG. 16 illustrates a flow diagram representing an exemplary method for creating smart contracts in a process control system using a distributed ledger; and

[0060] FIG. 17 illustrates a flow diagram representing an exemplary method for interacting with a smart contract in a process control system using a distributed ledger.

[0061] A distributed ledger is a storage mechanism for data, events, transactions, etc. that is maintained by several participants. More specifically, a distributed ledger is a way of achieving a distributed consensus on the validity or invalidity of information recorded in the distributed ledger. In other words, the distributed ledger provides a decentralized trust to participants and observers. As opposed to relying on a central authority, a distributed ledger is a decentralized database in which a transactional record of changes to the ledger is maintained and validated by each node of a peer-to-peer network. One type of distributed ledger, a blockchain, is comprised of groupings of transactions organized together into a "block," and ordered sequentially (thus the term "blockchain"). While the distributed ledgers discussed herein are referred to in the context of a blockchain, this is merely one example of a distributed ledger. Distributed ledgers may also include a tangle, a block lattice, or other directed acyclic graph (DAG). In any event, nodes may join and leave the blockchain network over time

and may obtain blocks from peer nodes that were propagated while the node was gone. Nodes may maintain addresses of other nodes and exchange addresses of known nodes with one another to facilitate the propagation of new information across the network in a decentralized, peer-to-peer manner.

[0062] The nodes that share the ledger form what is referred to herein as the distributed ledger network. The nodes in the distributed ledger network validate changes to the blockchain (e.g., when a new transaction and/or block is created) according to a set of consensus rules. The consensus rules depend on the information being tracked by the blockchain and may include rules regarding the chain itself. For example, a consensus rule may include that the originator of a change supply a proof-of-identity such that only approved entities may originate changes to the chain. A consensus rule may require that blocks and transactions adhere to format requirements and supply certain meta information regarding the change (e.g., blocks must be below a size limit, transactions must include a number of fields, etc.). Consensus rules may include a mechanism to determine the order in which new blocks are added to the chain (e.g., through a proof-of-work system, proof-of-stake, etc.).

[0063] Additions to the blockchain that satisfy the consensus rules are propagated from nodes that have validated the addition to other nodes that the validating node is aware of. If all of the nodes that receive a change to the blockchain validate the new block, then the distributed ledger reflects the new change as stored on all nodes, and it may be said that distributed consensus has been reached with respect to the new block and the information contained therein. Any change that does not satisfy the consensus rule is disregarded by validating nodes that receive the change and the change is not propagated to other nodes. Accordingly, unlike a traditional system which uses a central authority, a single party cannot unilaterally alter the distributed ledger unless the single party can do so in a way that satisfies the consensus rules. The inability to modify past transactions leads to blockchains being generally described as trusted, secure, and immutable.

[0064] The validation activities of nodes applying consensus rules on a blockchain network may take various forms. In one implementation, the blockchain may be viewed as a shared spreadsheet that tracks data such as the ownership of assets. In another implementation, the validating nodes execute code contained in "smart contracts" and distributed consensus is expressed as the network nodes agreeing on the output of the executed code.

[0065] A smart contract is a computer protocol that enables the automatic execution and/or enforcement of an agreement between different parties. In particular, the smart contract may be computer code that is located at a particular address on the blockchain. In some cases the smart contract may run automatically in response to a participant in the blockchain sending funds (e.g., a cryptocurrency such as bitcoin, ether, or other digital/virtual currency) to the address where the smart contract is stored. Additionally, smart contracts may maintain a balance of the amount of funds that are stored at their address. In some scenarios when this balance reaches zero the smart contract may no longer be operational.

[0066] The smart contract may include one or more trigger conditions, that, when satisfied, correspond to one or more actions. For some smart contracts, the action(s) performed may be determined based upon one or more decision conditions. In some instances, data streams may be routed to the smart contract so that the smart contract may detect that a trigger condition has occurred and/or analyze a decision condition.

[0067] Blockchains may be deployed in a public, decentralized, and permissionless manner meaning that any party may view the distributed ledger, submit new information to be added to the ledger, or join the network as a validating node. Other blockchains are private (e.g., permissioned ledgers) that keep chain data private among a group of entities authorized to participate in the blockchain network. Other blockchain implementations may be both permissioned and permissionless whereby participants may need to be validated, but only the information that participants in the network wish to be public is made public.

[0068] In some implementations, a distributed ledger includes multiple blockchains such as a main blockchain and several side chains operating independently of the main blockchain. The side chains then interact with the main blockchain to provide some of the transaction data from the side chains to the main blockchain. In this manner, the side chains can be private while the main blockchain is public or available to a larger number of entities than the side chains. Non-sensitive information from the side chains may be shared on the main blockchain. Also in some implementations, a distributed ledger includes multiple layers or separate blockchains executing in parallel that are maintained by the same validating nodes. Some of the transaction data from the blockchain for the first layer may be provided to the blockchain for the second layer or vice versa.

[0069] In one example, a distributed ledger in a process control system may be maintained by validating nodes referred to herein as "edge gateways" which transmit data to remote systems such as other process plants using one or more public and/or private networks, such as a private enterprise network, the Internet, a cellular router, a backhaul Internet or other type backhaul connection. The edge gateways receive transactions broadcasted to the distributed ledger network by for example, process control devices such as field devices or controllers operating in the process plant. Other computing devices such as operator workstations, server devices, or other user interface devices in the process plant may also broadcast transactions to the distributed ledger network. The edge gateways then validate the broadcasted transactions.

[0070] In another example, the edge gateways execute code contained in "smart contracts" and field devices act as "evidence oracles" which provide evidence related to quality control, compliance with regulations, delivery or receipt of a product and the quantity delivered/received, etc. to the blockchain.

[0071] FIG. 1 is a block diagram of an example process plant 10 which may be utilize any one or more of the novel distributed ledger techniques described herein. The process plant 10 (which is also interchangeably referred to herein as a process control system 10 or process control environment 10) includes one or more process controllers that receive signals indicative of process measurements made by field devices, process this information to implement a control routine, and generate control signals that are sent over wired or wireless process control communication links or networks to other field devices to control the operation of a process in the plant 10. Typically, at least one field device performs a physical function (e.g., opening or closing a valve, increasing or decreasing a temperature, taking a measurement, sensing a condition, etc.) to control the operation of a process. Some types of field devices communicate with controllers by using I/O devices. Process controllers, field devices, and I/O devices may be wired or wireless, and any number and combination of wired and wireless process controllers, field devices and I/O devices may be included in the process plant environment or system 10.

[0072] For example, FIG. 1 illustrates a process controller 11 that is communicatively connected to wired field devices 15-22 via input/output (I/O) cards 26 and 28, and that is communicatively connected to wireless field devices 40-46 via a wireless gateway 35 and a process control data highway or backbone 105. The process control data highway 105 may include one or more wired and/or wireless communication links, and may be implemented using any desired or suitable or communication protocol such as, for example, an Ethernet protocol. In some configurations (not shown), the controller 11 may be communicatively connected to the wireless gateway 35 using one or more communications networks other than the backbone 105, such as by using any number of other wired or wireless communication links that support one or more communication protocols, e.g., Wi-Fi (RTM) or other IEEE 802.11 compliant wireless local area network protocol, mobile communication protocol (e.g., WiMAX (RTM), LTE, or other ITU-R compatible protocol), Bluetooth (RTM), HART (RTM), WirelessHART (RTM), Profibus, FOUNDATION (RTM) Fieldbus, etc.

[0073] The controller 11, which may be, by way of example, the DeltaV™ controller sold by Emerson Process Management, may operate to implement a batch process or a continuous process using at least some of the field devices 15-22 and 40-46. In an embodiment, in addition to being communicatively connected to the process control data highway 105, the controller 11 is also communicatively connected to at least some of the field devices 15-22 and 40-46 using any desired hardware and software associated with, for example, standard 4-20 mA devices, I/O cards 26, 28, and/or any smart communication protocol such as the FOUNDATION (RTM) Fieldbus protocol, the HART (RTM) protocol, the WirelessHART (RTM) protocol, etc. In FIG. 1, the controller 11, the field devices 15-22 and the I/O cards 26, 28 are wired devices, and the field devices 40-46 are wireless field devices. Of course, the wired field devices 15-22 and wireless field devices 40-46 could conform to any other desired standard(s) or protocols, such as any wired or wireless protocols, including any standards or protocols developed in the future.

[0074] The process controller 11 of FIG. 1 includes a processor 30 that implements or oversees one or more process control routines 38 (e.g., that are stored in a memory 32). The processor 30 is configured to communicate with the field devices 15-22 and 40-46 and with other nodes communicatively connected to the controller 11. It should be noted that any control routines or modules described herein may have parts thereof implemented or executed by different controllers or other devices if so desired. Likewise, the control routines or modules 38 described herein which are to be implemented within the process control system 10 may take any form, including software, firmware, hardware, etc. Control routines may be implemented in any desired software format, such as using object oriented programming, ladder logic, sequential function charts, function block diagrams, or using any other software programming language or design paradigm. The control routines 38 may be stored in any desired type of memory 32, such as random access memory (RAM), or read only memory (ROM). Likewise, the control routines 38 may be hard-coded into, for example, one or more EPROMs, EEPROMs, application specific integrated circuits (ASICs), or any other hardware or firmware elements. Thus, the controller 11 may be configured to implement a control strategy or control routine in any desired manner.

[0075] The controller 11 implements a control strategy using what are commonly referred to as function blocks, where each function block is an object or other part (e.g., a subroutine) of an overall control routine and operates in conjunction with other function blocks (via communications called links) to implement process control

loops within the process control system 10. Control based function blocks typically perform one of an input function, such as that associated with a transmitter, a sensor or other process parameter measurement device, a control function, such as that associated with a control routine that performs PID, fuzzy logic, etc. control, or an output function which controls the operation of some device, such as a valve, to perform some physical function within the process control system 10. Of course, hybrid and other types of function blocks exist. Function blocks may be stored in and executed by the controller 11, which is typically the case when these function blocks are used for, or are associated with standard 4-20 mA devices and some types of smart field devices such as HART (RTM) devices, or may be stored in and implemented by the field devices themselves, which can be the case with FOUNDATION (RTM) Fieldbus devices. The controller 11 may include one or more control routines 38 that may implement one or more control loops which are performed by executing one or more of the function blocks.

[0076] The wired field devices 15-22 may be any types of devices, such as sensors, valves, transmitters, positioners, etc., while the I/O cards 26 and 28 may be any types of I/O devices conforming to any desired communication or controller protocol. In FIG. 1, the field devices 15-18 are standard 4-20 mA devices or HART (RTM) devices that communicate over analog lines or combined analog and digital lines to the I/O card 26, while the field devices 19-22 are smart devices, such as FOUNDATION (RTM) Fieldbus field devices, that communicate over a digital bus to the I/O card 28 using a FOUNDATION (RTM) Fieldbus communications protocol. In some embodiments, though, at least some of the wired field devices 15, 16 and 18-21 and/or at least some of the I/O cards 26, 28 additionally or alternatively communicate with the controller 11 using the process control data highway 105 and/or by using other suitable control system protocols (e.g., Profibus, DeviceNet, Foundation (RTM) Fieldbus, ControlNet, Modbus, HART (RTM), etc.).

[0077] In FIG. 1, the wireless field devices 40-46 communicate via a wireless process control communication network 70 using a wireless protocol, such as the WirelessHART (RTM) protocol. Such wireless field devices 40-46 may directly communicate with one or more other devices or nodes of the wireless network 70 that are also configured to communicate wirelessly (using the wireless protocol or another wireless protocol, for example). To communicate with one or more other nodes that are not configured to communicate wirelessly, the wireless field devices 40-46 may utilize a wireless gateway 35 connected to the process control data highway 105 or to another process control communications network. The wireless gateway 35 provides access to various wireless devices 40-58 of the wireless communications network 70. In particular, the wireless gateway 35 provides communicative coupling between the wireless devices 40-58, the wired devices 15-28, and/or other nodes or devices of the process control plant 10. For example, the wireless gateway 35 may provide communicative coupling by using the process control data highway 105 and/or by using one or more other communications networks of the process plant 10.

[0078] Similar to the wired field devices 15-22, the wireless field devices 40-16 of the wireless network 70 perform physical control functions within the process plant 10, e.g., opening or closing valves, or taking measurements of process parameters. The wireless field devices 40-46, however, are configured to communicate using the wireless protocol of the network 70. As such, the wireless field devices 40-46, the wireless gateway 35, and other wireless nodes 52-58 of the wireless network 70 are producers and consumers of wireless communication packets.

[0079] In some configurations of the process plant 10, the wireless network 70 includes non-wireless devices. For example, in FIG. 1, a field device 48 of FIG. 1 is a legacy 4-20 mA device and a field device 50 is a wired HART (RTM) device. To communicate within the network 70, the field devices 48 and 50 are connected to the wireless communications network 70 via a wireless adaptor 52A, 52B. The wireless adaptors 52A, 52B support a wireless protocol, such as WirelessHART (RTM), and may also support one or more other communication protocols such as Foundation (RTM) Fieldbus, PROFIBUS, DeviceNet, etc. Additionally, in some configurations, the wireless network 70 includes one or more network access points 55A, 55B, which may be separate physical devices in wired communication with the wireless gateway 35 or may be provided with the wireless gateway 35 as an integral device. The wireless network 70 may also include one or more routers 58 to forward packets from one wireless device to another wireless device within the wireless communications network 70. In FIG. 1, the wireless devices 40-46 and 52-58 communicate with each other and with the wireless gateway 35 over wireless links 60 of the wireless communications network 70, and/or via the process control data highway 105.

[0080] In FIG. 1, the process control system 10 includes one or more operator workstations or user interface devices 8 that are communicatively connected to the data highway 105. Via the operator workstations 8, operators may view and monitor run-time operations of the process plant 10, as well as take any diagnostic, corrective, maintenance, and/or other actions that may be required. At least some of the operator workstations 8 may be located at various, protected areas in or near the plant 10, and in some situations, at least some of the operator workstations 8 may be remotely located, but nonetheless in communicative connection with the plant 10. Operator workstations 8 may be wired or wireless computing devices.

[0081] The example process control system 10 may further include a configuration application (not shown) and configuration database (not shown), each of which is also communicatively connected to the data highway 105. As discussed above, various instances of the configuration application (not shown) may execute on one or more user interface devices 8 to enable users to create or change process control modules and download these modules via the data highway 105 to the controllers 11, as well as enable users to create or change operator interfaces via which an operator is able to view data and change data settings within process control routines. The configuration database (not shown) stores the created (e.g., configured) modules and/or operator interfaces.

[0082] In some configurations, the process control system 10 includes one or more other wireless access points 7a that communicate with other devices using other wireless protocols, such as Wi-Fi (RTM) or other IEEE 802.11 compliant wireless local area network protocols, mobile communication protocols such as WiMAX (RTM) (Worldwide Interoperability for Microwave Access), LTE (Long Term Evolution) or other ITU-R (International Telecommunication Union Radiocommunication Sector) compatible protocols, short-wavelength radio communications such as near field communications (NFC) and Bluetooth (RTM), or other wireless communication protocols. Typically, such wireless access points 7a allow handheld or other portable computing devices to communicate over a respective wireless process control communication network that is different from the wireless network 70 and that supports a different wireless protocol than the wireless network 70. For example, a wireless or portable user interface device 8 may be a mobile workstation or diagnostic test equipment that is utilized by an operator within the process plant 10. In some scenarios, in addition to portable

computing devices, one or more process control devices (e.g., controller 11, field devices 15-22, or wireless devices 35, 40-58) also communicate using the wireless protocol supported by the access points 7a.

[0083] In some configurations, the process control system 10 includes one or more gateways 7b, 7c to systems that are external to the immediate process control system 10 (also referred to herein as "edge gateway" and described in more detail below). Typically, such systems are customers or suppliers of information generated or operated on by the process control system 10. For example, the process control plant 10 may include a gateway node 7b to communicatively connect the immediate process plant 10 with another process plant. Additionally or alternatively, the process control plant 10 may include a gateway node 7c to communicatively connect the immediate process plant 10 with an external public or private system, such as a laboratory system (e.g., Laboratory Information Management System or LIMS), an operator rounds database, a materials handling system, a maintenance management system, a product inventory control system, a production scheduling system, a weather data system, a shipping and handling system, a packaging system, the Internet, another provider's process control system, or other external systems.

[0084] It is noted that although FIG. 1 only illustrates a single controller 11 with a finite number of field devices 15-22 and 40-46, wireless gateways 35, wireless adaptors 52, access points 55, routers 58, and wireless process control communications networks 70 included in the example process plant 10, this is only an illustrative and non-limiting embodiment. Any number of controllers 11 may be included in the process control plant or system 10, and any of the controllers 11 may communicate with any number of wired or wireless devices and networks 15-22, 40-46, 35, 52, 55, 58 and 70 to control a process in the plant 10.

[0085] Further, it is noted that the process plant or control system 10 of FIG. 1 includes a field environment (e.g., "the process plant floor") and a back-end environment (e.g., servers 12) which are communicatively connected by the data highway 105. As shown in FIG. 1, the field environment includes physical components (e.g., process control devices, networks, network elements, etc.) that are disposed, installed, and interconnected therein to operate to control the process during run-time. For example, the controller 11, the I/O cards 26, 28, the field devices 15-22, and other devices and network components 40-46, 35, 52, 55, 58 and 70 are located, disposed, or otherwise included in the field environment of the process plant 10. Generally speaking, in the field environment of the process plant 10, raw materials are received and processed using the physical components disposed therein to generate one or more products.

[0086] The back-end environment of the process plant 10 includes various components such as server computing devices 12, operator workstations 8, databases or databanks, etc. that are shielded and/or protected from the harsh conditions and materials of the field environment. Referring to FIG. 1, the back-end environment includes, for example, the operator workstations 8, server computing devices 12, and/or functionality that support the run-time operations of the process plant 10. In some configurations, various computing devices, databases, and other components and equipment included in the back-end environment of the process plant 10 may be physically located at different physical locations, some of which may be local to the process plant 10, and some of which may be remote.

[0087] FIG. 2 includes a block diagram of an example security architecture 200 for the process plant 10. As shown in FIG. 2, one or more devices 202 are communicatively connected to one or more wireless gateways 205A, 205B which, for example, may be instances of the wireless gateway 35 of FIG. 1. The communicative connections between the gateways 205A, 205B and the devices 202 are denoted by the references 204A, 204B.

[0088] The set of devices 202 is depicted as comprising a finite number of wireless field devices. However, it is understood that the concepts and features described herein with respect to the devices 202 may be easily applied to any number of field devices of the process plant 10, as well as to any types of field devices. For example, the field devices 202 may include one or more wired field devices 15-22 that are communicatively connected to the wireless gateways 205A, 205B via one or more wired communication networks of the process plant 10, and/or the field devices 202 may include wired field devices 48, 50 that are coupled to wireless adaptors 52A, 52B.

[0089] Further, it is understood that the set of devices 202 is not limited to only field devices, but may additionally or alternatively include any device or component within the process plant 10 that generates data as a result of the process plant 10 controlling the on-line process. For example, the set of devices 202 may include a diagnostic device or component that generates diagnostic data, a network routing device or component that transmits information between various components of the process plant 10, and the like. Indeed, any of the components shown in FIG. 1 (e.g., components 7a-7c, 8, 11, 12, 15-22, 26, 28, 35, 40-46, 52, 55, 58, 60, and 70) and other components that are not shown may be a device that generates data for delivery to the remote system 210. As such, the set of devices 202 is referred to interchangeably herein as "data sources 202" or "data source devices 202."

[0090] FIG. 2 further illustrates a set of remote applications or services 208 that may be utilized for the process plant 10 and/or that the process plant 10 utilizes. The set of remote applications or services 208 may execute or be hosted at one or more remote systems 210. At least some of the applications or services 208 operate in real-time on real-time data as the real-time data is generated by the process plant 10 and received by the applications or services 208. Other applications or services 208 may operate or execute on process plant-generated data with less stringent timing requirements. Examples of applications/services 208 that may execute or be hosted at the remote system 210 and that are consumers of data generated by the process plant 10 include applications that monitor and/or sense conditions and/or events occurring at the process plant 10, and applications or services that monitor at least a portion of the on-line process itself as it is executing at the process plant 10. Other examples of applications/services 208 include descriptive and/or prescriptive analytics, which may operate on data generated by the process plant 10 and, in some cases, may operate on knowledge gleaned or discovered from analyzing the process plant-generated data, as well as on data generated by and received from other process plants. Still other examples of applications/services 208 include one or more routines that implement prescriptive functions and/or changes that are to be implemented back into the process plant 10, e.g., as a result of another service or application. Other examples of applications and services 208 operate on knowledge gleaned from analyzing historical data generated by the process plant and/or other process plants or from comparing data for a process plant entity to data process plant entities of a same or similar type.

[0091] The one or more remote systems 210 may be implemented in any desired manner, such as by a remote bank of networked servers, one or more cloud computing systems, one or more networks, etc. For ease of discussion, the one or more remote systems 210 are referred to herein using the singular tense, i.e., "remote system 210," although it is understood that said term may refer to one system, more than one system, or any number of systems. In some scenarios, the computing device 250 which analyzes process plant data may be included within the remote system 210.

[0092] Generally speaking, the security architecture 200 provides end-to-end security from the field environment of the process plant 10 in which devices 202 are installed and operate, to the remote system 210 providing applications and/or services 208 that consume and operate on the data generated by the process plant 10. As such, data that is generated by the devices 202 and other components of the process plant 10 is able to be securely transported to the remote system 210 for use by the remote applications/services 208 while protecting the plant 10 from cyber attacks, intrusions, and/or other malicious events. In particular, the security architecture 200 includes a field gateway 212, and an edge gateway 218 disposed between the process plant 10 (e.g., between the wireless gateways 205A, 205B of the process plant 10) and the remote system 210.

[0093] Data that is egressed from the process plant 10 and transmitted from the input port 220 to the output port 222 may be further secured by encryption. In an example, the field gateway 212 encrypts data and delivers encrypted data to the input port 220. The data traffic that is encrypted and transported may be UDP (User Datagram Protocol) data traffic, in an example, and may be JSON data traffic or some other general purpose communication format, in another example.

[0094] The field gateway 212 communicatively connects to the process control plant 10. As shown in FIG. 2, the field gateway 212 is communicatively connected to the wireless gateways 205A, 205B that are disposed within the field environment of the process plant 10, and that are communicatively connected to one or more devices or data sources 202. As previously discussed, the devices or data sources 202 and the wireless gateways 205A, 205B may communicate using the WirelessHART (RTM) industrial protocol or other suitable wireless protocol that is structured to provide secured communications via one or more security mechanisms. For instance, the WirelessHART (RTM) industrial protocol provides 128-bit AES encryption, and the communication paths 204A, 204B may be secured accordingly.

[0095] Additionally, the communicative connection 225 between the wireless gateways 205A, 205B and the field gateway 212 is respectively secured using the same or a different security mechanism as utilized for the communicative connections 204A, 204B. In an example, the communicative connection 225 is secured by a TLS (Transport Layer Security) wrapper. For instance, the wireless gateways 205A, 205B generate packets in the HART-IP format which are secured by a TLS wrapper for transit to the field gateway 212.

[0096] Thus, as described above, in an embodiment, data or packets generated by the devices 202 may be secured for transit 204A, 204B to the wireless gateways 205A, 205B using a first security mechanism, and subsequently secured for transit 225 from the wireless gateways 205A, 205B to the field gateway 212 using a second security mechanism, and still subsequently secured for transit to the edge gateway 218 using a third

security mechanism. Additionally or alternatively, and as depicted in FIG. 2, the edge gateway 218 may be protected by a firewall 228.

[0097] Data transiting from the edge gateway 218 to the remote system 210 may be delivered using one or more public and/or private networks, such as a private enterprise network, the Internet, a cellular router, a backhaul Internet or other type backhaul connection. Significantly, the data transiting from the edge gateway 218 to the remote system 210 is secured by using a fourth security mechanism or by using one of security mechanisms previously discussed above. FIG. 2 depicts the data traffic delivered from the edge gateway 218 to the remote system 210 as being secured via an SAS (Shared Access Signature) Token, which may be managed through a token service 230 provided at the remote system 210. The edge gateway 218 authenticates to the token service 230 and requests an SAS token, which may be valid for only a finite period of time, e.g., two minutes, five minutes, thirty minutes, no more than an hour, etc. The edge gateway 218 receives and uses the SAS token to secure and authenticate an AMQP (Advanced Message Queuing Protocol) connection to the remote system 210 via which content data is transmitted from the edge gateway 218 to the remote system 210.

[0098] At the remote system 210, security is provided via a domain authentication service 232. As such, only user interface devices 235 that are authenticated and authorized via the domain authentication service 232 are able gain access to at least some of the data that is available at the remote system 210, which includes, *inter alia*, the data generated by the devices 202.

[0099] Thus, as described above, the security architecture 200 provides end-to-end security for data generated by devices or data sources 202 while operating in the process plant 10 to control a process, e.g., from the data's inception by the data sources 202 through its transmission to the remote system 210 to be operated on by one or more remote applications or services 208. Importantly, the security architecture 200 provides this end-to-end security while preventing malicious attacks from being incurred on the process plant 10.

[00100] It is noted that although FIG. 2 depicts wireless gateways 205A, 205B as communicatively connecting the devices or data sources 202 to the field gateway 212, in some arrangements one or more of the wireless gateways 205A, 205B are omitted and source data is transmitted from the data sources 202 directly to the field gateway 212. For example, the data sources 202 may transmit source data directly to the field gateway 212 via a big data network of the process plant 10. Generally speaking, a big data network of the process plant 10 is not the backbone plant network 105, nor is the big data network an industrial protocol network used to transmit control signals between devices using an industrial communication protocol (e.g., Profibus, DeviceNet, Foundation (RTM) Fieldbus, ControlNet, Modbus, HART (RTM), etc.). Rather, a big data network of the process plant 10 may be an overlay network implemented for the process plant 10 that streams data between nodes for data processing and analytics purposes, for example. The nodes of a big data network may include, for example, the data sources 202, the wireless gateways 205A, 205B, and the field gateway 212, as well as any one or more of the components 7a-7c, 8, 11, 12, 15-22, 26, 28, 35, 40-46, 52, 55, 58, 60, and 70 shown in FIG. 1 and other components. Accordingly, for many nodes of a process plant data network include, respectively, a designated interface for process plant operations that typically utilizes an industrial communication protocol, and another designated interface for data processing/analytics operations that may utilize a streaming protocol, for instance.

[00101] It is further noted with respect to FIG. 2 that in some embodiments, a wired gateway (not shown) may be utilized in lieu of one of the wireless gateways 205A, 205B. Still further, the field gateway 212 and the edge gateway 218 may be physically co-located, such as indicated by the box 235 shown in FIG. 2, or the components 212 and 218 may be physically located across multiple locations. For example, one or more of the field gateway 212 or the edge gateway 218 may be disposed at the process plant 10. Additionally or alternatively, one or more of the field gateway 212 or the edge gateway 218 may be disposed remotely from the process plant 10.

[00102] The process plant 10 may be serviced by more than one field gateway 212, if desired, and any number of field gateways 210 may be serviced by a single edge gateway 218. In some embodiments, the remote system 210 is serviced by more than one edge gateway 218, if desired.

[00103] While the example above refers to the computing device 250 for analyzing process plant data as a component of the remote system 210, the computing device 250 may receive process plant data by communicating with any suitable communication component in a secure manner. For example, the computing device 250 may be communicatively connected to the wireless gateways 205A, 205B, the field gateway 212, or the edge gateway 218. The communication paths may be secured from the devices 202 to the computing device 250 via encryption techniques, firewalls, a data diode, or with any other suitable security mechanism.

05 09 22

[00104] Once the process plant data is received at the computing device 250, the computing device analyzes the process plant data to identify conditions in corresponding process plant entities. Indications of the conditions are then transmitted to the user interface device 235 via a domain authentication service, for example. In this manner, an operator may view the conditions occurring at various process plant entities within the process plant. The operator may then take the appropriate actions to resolve issues created by these conditions.

Distributed Ledger Architecture in a Process Control System

[00105] While the process plant 10 is illustrated in FIG. 2 as including a single edge gateway 218, the process plant 10 may include several edge gateways each acting as a validating node in a distributed ledger network. FIG. 3 depicts an exemplary distributed ledger system 300 for recording process plant data. Process plant data may include process parameter data, product parameter data, configuration data, user interaction data, maintenance data, commissioning data, plant network data, product tracking data, event data related to events in the process plant 10 such as alarms, leaks, failures, errors, etc., or any other suitable data generated in or related to one or several process plants.

[00106] The system 300 includes a distributed ledger 312 and plurality of nodes 302, 304, 306, 308, and 310, which may be edge gateways in the process plant 10, such as the edge gateway 218, may be field devices, or may be any suitable computing devices operating in the process plant 10 or other process plants. Each node maintains a copy of the distributed ledger 312. As changes are made to the distributed ledger 312, each node receives the change via the network 314 and updates its respective copy of the distributed ledger 312. A consensus mechanism may be used by the nodes 302-310 in the distributed ledger system 300 to decide whether it is appropriate to make received changes to the distributed ledger 312.

[00107] Each node in the system therefore has its own copy of the distributed ledger 312, which is identical to every other copy of the distributed ledger 312 stored by the other nodes. The distributed ledger system 300 may be more robust than a central authority database system because of the distributed ledger's decentralized nature. As such, there is no single point of failure on the distributed ledger system 300 as there would be in a centralized system.

[00108] FIG. 4 depicts exemplary validating network nodes and an exemplary transaction flow 400 on a distributed ledger network for resolving transactions. FIG. 4 includes two time frames 420 and 422 represented by the left and right sides of the dotted line, respectively, Node A 402 and Node B 404 (which may be two edge gateways in a process plant 10, may be two edge gateways in two different process plants, may be two field devices in the same or different process plants, etc.), a set of transactions 408A-408D, a set of blocks of transactions 409A-409D, a distributed ledger 410, and a blockchain 418.

[00109] The block propagation flow 400 may begin with Node A 402 receiving transaction 406 at time 420. When Node A 402 confirms that transaction 406 is valid, Node A 402 may add the transaction to a newly generated block 408. As part of adding the transaction 406 to block 408, Node A 402 may solve a cryptographic puzzle and include the solution in the newly generated block 408 as proof of the work done to generate the block 408. Alternatively, a proof of stake algorithm may be used to generate the block 408, whereby Node A 402 "stakes" an amount of a digital token used on the network, however, the network itself determines the node that will mint the new block. In other embodiments, the transaction 406 may be added to a pool of transactions until a sufficient number of transactions in the pool exist to form a block. Node A 402 may transmit the newly created block 408 to the network at time 412. Before or after propagating the block 408, Node A 402 may add the block 408 to its copy of the blockchain 418.

[00110] While proof of work and proof of stake are described herein as consensus algorithms for selecting a node to mint a new block, these are merely a few example consensus algorithms and are not intended to be limiting. Additional consensus algorithms may be utilized, such as delegated proof of stake where nodes elect a subset of nodes referred to as delegates to perform validation, and the delegates take turns minting new blocks. Consensus algorithms may also include proof of authority, proof of weight, Byzantine fault tolerance, tangle consensus algorithms, block lattice consensus algorithms, etc.

[00111] In any event, the transactions 409A-409D may include updates to a state database 416. The state database 416 may contain current values of variables created by smart contracts deployed on the blockchain 418. Validated blocks, such as block 408, may include transactions effecting state variables in state database 416. At time 422, Node B 404 may receive the newly created block 408 via the network at 412. Node B 404 may verify that the block of transactions 408 is valid by checking the solution to the cryptographic puzzle provided in the block 408. If the solution is accurate, then Node B 404 may add the block 408 to its blockchain 418 and make any updates to the state database 416 as rejected by the transactions in block 408. Node B 404 may then transmit the block 408 to the rest of the network at time 314.

[00112] FIG. 5 depicts exemplary components of a validating network node 500 on a distributed ledger network for recording process plant data. Node 500 may include at least one processor 502, memory 504, a

communication module 506, a set of applications 508, external ports 510, a blockchain manager 514, smart contracts 516, and an operating system 518. In some embodiments, the node 500 may generate a new block of transactions, or may broadcast transactions to other network nodes by using the blockchain manager 514. Similarly, the node 500 may use the blockchain manager 514 in conjunction with the smart contracts 516 stored in the memory 504 to execute the functionality disclosed herein. The memory 504 may further include chain data 524 including, for example, a state database of the blockchain for storing states of smart contracts deployed thereon.

[00113] In other embodiments, the smart contracts 516 operate independent of the blockchain manager 514 or other applications. In some embodiments, the node 500 does not have a blockchain manager 514, or smart contracts 516 stored at the node. In some embodiments, the node 500 may have additional or fewer components than described. The components of the node 500 are described in more detail below.

[00114] The node 500, as part of a decentralized ledger system 300, or another decentralized or centralized network, may be used as part of systems that interact with and/or manipulate transactions associated with data or events occurring in one or several process plants.

[00115] FIG. 6A depicts an exemplary distributed ledger 600 including a blockchain having blocks 602-608 of transactions in a process control system. In some embodiments, the blockchain 600 includes several blocks 602-608 connected together to form a chain of blocks 602-608 of transactions. To cryptographically link blocks and transactions together, each block in the blockchain 600 organizes its transactions into a Merkle Tree. In a Merkle Tree each transaction is hashed according to a cryptographic hashing algorithm (*e.g.,* SHA-256) and the resulting output hash is then combined with the hash of another transaction. Then the combined result is also hashed according to the cryptographic hashing algorithm. This output is then combined with the hash of two other transactions and this process is repeated until all of the transactions in the block are combined and hashed to generate a Merkle root that is used in the header for a block 602-608. If any single transaction in the block is tampered with, a different Merkle root would be generated since the Merkle root is a combination of the hashes of all of the transactions in the block.

[00116] In other words, the transactions may be hashed using a cryptographic hash algorithm, such as the algorithms discussed above, and the hash of each transaction may be stored in the tree. As the tree is constructed the hash of each adjacent node at the same level may be hashed together to create a new node that exists at a higher level in the tree. Therefore, the node at the top of the tree or Merkle root, is dependent upon the hash of each transaction stored below in the tree. Each transaction may include a set of data. The set of data may include identifying data for the transaction, and transaction data identifying the nature of the transaction and what the transaction entails (e.g., input and output addresses, a transaction value, a document hash value, a timestamp, a transaction fee value, etc.).

[00117] To verify that a block is valid, a node may compare the Merkle root of the block to the Merkle root for the same block included in other nodes' copies of the blockchain. Thus, the Merkle root can be used as proof of the transactions included in the block and as proof that the contents of the block have not been tampered with if the Merkle root is the same in each node's copy of the block.

[00118] In one implementation, documents stored "on" a blockchain are documents that have been hashed according to a cryptographic hashing algorithm (e.g., SHA-256) and the resulting output hash has been included in a transaction in a block that has been accepted by the network nodes as satisfying the consensus rules of the blockchain. As such, the documents may be later verified or validated by comparing the hash of the documents to the hash stored on the blockchain. For example, if a set of documents results in a SHA-256 hash that was recorded on a blockchain on a certain date, then the blockchain provides cryptographic proof that the documents existed as of that date.

[00119] One way of storing a document on a blockchain is to broadcast a transaction including a hash of the document to the network, which will be included in a block if the transaction satisfies all of the consensus rules of the network. In some implementations, the blockchain is a permissioned ledger, meaning only authorized network participants may broadcast transactions. In other implementations, only some authorized network participants may make certain transactions. For example, product parameter data indicating properties of a product generated in a process plant 10 may be uploaded by a field device to the blockchain 600 as the field device determines the product's properties (e.g., a temperature of the product, a volume of the product, a mass of the product, a density of the product, a pressure of the product, etc.). Only a cryptographic hash of the data may be included in the blockchain 600, such that the data may be verified using the blockchain even if it is obtained by a party off-chain.

[00120] Validating network nodes may verify that the signed transaction or signed message was signed by the private cryptographic key corresponding to the published public cryptographic key owned by the field device collecting the measurements. In at least one implementation, a valid proof-of-identity may be applied as a consensus rule by the blockchain network. As such, any transaction attempting to add new product parameter data without a cryptographic proof-of-identity matching an identity authorized to add new product parameter data is rejected by the network as non-compliant with the consensus rule. Each field device in a process plant 10 may be assigned a public key/private key pair which is identified in the blockchain network as corresponding to the field device. Additionally, each field device may be authorized to collect certain types of measurements. For example, a first field device may be authorized to collect temperature measurements for a product while a second field device may be authorized to collect volume measurements indicating the volume of the product manufactured. If the validating network nodes receive a transaction regarding product parameter data that is not from an authorized field device or includes a type of measurement that the field device is not authorized to collect, the validating network nodes reject the transaction.

[00121] FIG. 6B depicts another exemplary distributed ledger 650 including a different architecture from the architecture described in FIG. 6A. The distributed ledger 650 in FIG. 6B includes a blockchain 660 having blocks 662-668 of transactions in a process control system, similar to the distributed ledger 600 in FIG. 6A. The blockchain 660 may be referred as the main blockchain in the distributed ledger 650. In addition to the main blockchain 660, the distributed ledger 650 includes multiple side blockchains 670, 680 or side chains maintained by different process plants having blocks 672-676, 682-686 of transactions. For example, the side chain 670 may be maintained by two process plants: Plant A and Plant B to record transactions related to events occurring within or between the two process plants. These transactions may include Plant B sending payment in the form

of a token value to Plant A when Plant A ships a product to Plant B. The side chain 680 may also be maintained by two process plants: Plant C and Plant D to record transactions related to events occurring within or between the Plant C and Plant D. These transactions may include Plant D recording the amount of oil received from Plant C within a particular time period.

[00122] In some embodiments, the main blockchain 660 is maintained by several process plants including Plants A-D along with several other process plants. Also in some embodiments, the side chains 670, 680 interact with the main blockchain 660 to provide at least some of the transactions in their respective blocks 672-676, 682-686 to the main blockchain 660. In this manner, the side chains 670, 680 may include data from transactions related to the process plants maintaining them. The main blockchain 660 may include data from transactions related to each of the process plants. Additionally, the side chains 670, 680 may include private or sensitive data that is not meant to be shared outside of the process plants maintaining a particular side chain. Data from the side chain 670 which is not private or sensitive may be provided to the main blockchain 660, while the private or sensitive data is not provided to the main blockchain 660. For example, side chain 670 may execute a smart contract between Plant A and Plant B that transfers a token value from Plant A to Plant B when Plant A receives a product from Plant B that meets certain quality standards. Plants A and B may not want to disclose all of the terms of the smart contract to the public or to a large group of process plants by deploying the smart contract on the main blockchain 660, or may not want each measurement of the product's properties to be provided to the public or a large group of process plants. Additionally, the memory storage requirements for the main blockchain 660 increase as more transactions are added to the main blockchain 660. Accordingly, it may reduce memory requirements for validating nodes in the distributed ledger network to store some transactions off the main blockchain 660. In any event, when the smart contract determines that Plant A has received a product from Plant B that meets the requisite quality standards, the transaction transferring the token value from Plant A to Plant B may be provided to the main blockchain 660.

[00123] In some embodiments, the main blockchain 660 is a public blockchain meaning that any party may view the distributed ledger, submit new information to be added to the ledger, or join the network as a validating node. The side chains 670, 680 are private or permissioned blockchains that keep chain data private among a group of entities authorized to participate in the side blockchain network (e.g., the side chain 670 may be private between Plant A and Plant B). In other embodiments, the main blockchain 660 is also a permissioned blockchain but the main blockchain has a larger number of entities authorized to participate in the blockchain network than the side chains 670, 680. For example, the main blockchain 660 may be private between a large number of process plants including Plants A-D and several other process plants, whereas the side chain 670 is private between Plant A and Plant B.

[00124] In addition or as an alternative to side chains, the distributed ledger 650 may include other forms of transactions which occur off-chain that are not a part of the main blockchain 660. For example, two parties such as Plant A and Plant B may open up a payment channel, where an initial transaction exchanging a threshold amount of a token between Plant A and Plant B is provided to the main blockchain 660. Then Plant A and Plant B may transact with each other without recording anything on the main blockchain 660 as long as they are sending portions of the threshold amount back and forth to each other, and none of the transactions result in one

of the process plants having more than the threshold amount. When the two process plants are done transacting with each other, they may close the payment channel and provide the final token amounts for each process plant in the main blockchain 660. For example, Plant A and Plant B may open up a payment channel when Plant A sends two tokens to Plant B. Plant B can then send one token back to Plant A so that each process plant has one token, Plant B can send 0.5 tokens back to Plant A and so forth, so long as neither process plant has more than two tokens. In other embodiments, the distributed ledger 650 may include multiple blockchain layers including separate blockchains operating independently of each other. For example, a first blockchain layer may record transactions related to the supply chain, while a second blockchain layer may record transactions related to the exchange of tokens. The first blockchain layer may be public while the second blockchain layer is private or vice versa.

[00125] In addition to protecting privacy via side chains or off-chain transactions, in some embodiments, privacy may be preserved on a public blockchain, such as the blockchain 600 as shown in FIG. 6A. For example, the transactions in the blockchain 600 may obfuscate the identities of the parties to the transaction and the transaction amounts through various encryption techniques.

[00126] FIGS. 7A-7C depict another exemplary distributed ledger 700 including a different architecture from the architecture described in FIG. 6A. The distributed ledger 700 in FIGS. 7A-7C includes multiple local blockchains 710, 720, where each local blockchain 710, 720 is maintained by a different party or process plant. Each local blockchain 710, 720 includes a block of transactions 712-716, 722-726 in a process control system. For example, multiple process plants may share a resource, such as oil from an oil pipeline, electricity from an electric power generation system, a product over rail, automotive, marine, or airborne transportation, a product through a liquid, gas, steam, fuel, or materials pipeline, or water from a water distribution system. Field devices in Plant A may collect measurements regarding the shared resource such as an amount of oil obtained from the pipeline, and broadcast the measurement data in transactions to the local blockchain for Plant A. Similarly, field devices in Plant B may collect measurements regarding the shared resource, and broadcast the measurement data in transactions to the local blockchain for Plant B.

[00127] As shown in FIG. 7B, transactions from each local blockchain 710, 720 are provided to a global blockchain 730 for the respective party or process plant, where the global blockchain 730 is maintained by several process plants and/or via cloud services having several cloud computing systems. For example, blocks from the local blockchain 710 for Plant A are provided to the global blockchain 730 for Plant A, blocks from the local blockchain 720 for Plant B are provided to the global blockchain for Plant B, etc. The blocks of transactions may be provided from local blockchains to corresponding global blockchains after a threshold time period or epoch. In this manner, validating nodes within a particular process plant maintaining each local blockchain may remove or prune blocks from the local blockchain that have been provided to the global blockchain other than the most recent block to reduce storage requirements.

[00128] As shown in FIG. 7B, block N (ref. no. 742), block N+1 (ref. no. 746), and block N+2 (ref. no. 748) are added to the local blockchain 710 for Plant A during time epoch E (ref. no. 740). After the threshold time period for time epoch E has expired, the validating nodes maintaining the local blockchain 710 for Plant A provide blocks N – N+2 (ref. no. 742-746) to the global blockchain 730 for Plant A. Then the validating nodes

maintaining the local blockchain 710 for Plant A remove or prune block N (ref. no. 742) and block N+1 (ref. no. 744) from the local blockchain 710 to reduce storage requirements. The local blockchain 710 at this time only includes the most recent block, block N+2 (ref. no. 746). Then during time epoch E+1 (ref. no. 750), block N+3 (ref. no. 752) and block N+4 (ref. no. 754) are added to the local blockchain 710. After the threshold time period for time epoch E+1 has expired, the validating nodes maintaining the local blockchain 710 for Plant A provide blocks N+3 – N+4 (ref. nos. 752-754) to the global blockchain 730 for Plant A. Then the validating nodes maintaining the local blockchain 710 for Plant A remove or prune blocks N+2 – N+3 (ref. nos. 746, 752) from the local blockchain 710. The local blockchain 710 at this time only includes the most recent block, block N+4 (ref. no. 754).

[00129] As shown in FIG. 7C, the validating nodes maintaining the global blockchains, such as the global blockchain for Plant A 730 and the global blockchain for Plant B 770, combine the global blockchains 730, 770 to create a super blockchain 760 having state blocks 762, 764. Each state block 762, 764 includes each of the blocks from the global blockchains 730, 770 for a particular time period. For example, state block K (ref. no. 762) includes the respective block N, block N+1, and block N+2 from each global blockchain 730, 770. State block K+1 (ref. no. 764) includes the respective block N+3, block N+4, and block N+5 from each global blockchain 730, 770.

[00130] To cryptographically link blocks and transactions together, each state block 762, 764 in the super blockchain 760 organizes its transactions into a Merkle Tree. If any single transaction in the state block block is tampered with, a different Merkle root would be generated since the Merkle root is a combination of the hashes of all of the transactions in the block. The Merkle root for each state block 762, 764 is included in the header for the state block 762, 764.

[00131] The distributed ledger architecture 700 described in FIGS. 7A-7C having local blockchains, global blockchains, and a super blockchain allows for competing entities to verify the accuracy of measurement data. For example, if Plant A reports to Plant B that Plant A retrieved 30,000 gallons of oil from an oil pipeline shared between the two entities, Plant B may retrieve measurement data from the super blockchain to verify the accuracy of this measurement. The measurement data may also be cryptographically verified within the super blockchain 760 by calculating an expected Merkle root for the header of the state block that includes the measurement data and comparing the actual Merkle root in the header of the state block to the expected Merkle root. This allows competing entities analyzing the super blockchain 760 to validate that the state blocks 762, 764 in the super blockchain 760 have not been tampered with.

Smart Contracts in a Process Control System

[00132] As described above, process control systems can deploy smart contracts to the distributed ledger to exchange value, for example upon receiving a product in good condition. Smart contracts may also be deployed to the distributed ledger to allow machines such as field devices to transact by themselves without human intervention.

[00133] FIG. 8 depicts an exemplary smart contract state 806 in a distributed ledger network within a process control system. FIG. 8 includes a blockchain 802, a block of transactions 804, and a secure write request smart

contract state 806. A smart contract may be deployed by any participant in the distributed ledger network or blockchain network (e.g., plant operators, configuration engineers, process control system designers, etc.) to establish a contract state 806 for a secure write request, for example. The deployed smart contract may expose methods and data to other participants in the blockchain network. Some of the data in the smart contract state may be private data that may only be altered by calling a method of the smart contract, or only altered by authorized blockchain participants. One way of altering the smart contract state is to broadcast a transaction to the distributed ledger network. If the broadcasted transaction satisfies consensus rules, network validators may include the transaction in a block. Inclusion in the blockchain of a transaction sending data to the smart contract may cause validating nodes to update a state database for the smart contract, thus allowing network participants access to a rich state mechanism to manage the secure write request, and ultimately to write parameter data to a safety instrumented system (SIS) device.

[00134] The secure write request smart contract state 806 may include pieces of data to identify the operator submitting the secure write request, the computing device that the operator uses to submit the secure write request, and/or the SIS device that is the target of the secure write request. In some embodiments, the operator may be identified by cryptographic public keys assigned to the operator's electronic wallet. The operator's computing device may be identified by the same cryptographic public keys as the operator if the operator's electronic wallet operates on the operator's computing device. In other embodiments, the operator's computing device may be identified by other cryptographic public keys known to belong to the operator's computing device by the other network participants.

[00135] In some embodiments, a contract owner may select a unique ID for the SIS device such that subsequent transactions and data sent to the smart contract can identify the SIS device by ID number. For example, each SIS device may have a different unique identifier in the smart contract. The contract owner may also specify identifiers of operators and/or computing devices authorized to perform secure writes. Subsequent data sent to the smart contract may include a message signed by private keys corresponding to the public keys identifying the operator and/or computing device in the smart contract, thus providing cryptographic proof that the transaction was originated by an authorized operator and/or an authorized computing device. The private and public keys may be managed solely by the operators/computing devices to minimize the attack surface for any attackers that might attempt to forge a transaction (e.g., the operators/computing devices generate public/private cryptographic key pairs offline, and only provide the public key to other network participants). An operator's and/or computing device's private keys may be generated according to a securely stored seed value (e.g., on a piece of physical paper or multiple copies of a piece of paper) such that the private keys may be recovered in the case of a data loss.

[00136] To write parameter data to an SIS device, the secure write request smart contract state 806 may obtain evidence of the secure write request. The evidence for the secure write request may include the name of the parameter to be changed in the SIS device and/or path information for the parameter. The evidence may also include a new parameter value, and in some embodiments, the evidence may include a cyclical redundancy check (CRC) value or other error checking value along with the new parameter value to ensure the parameter information is intact has not been corrupted. In some embodiments, in response to receiving the parameter

information, the smart contract may provide a confirmation dialog to the operator's computing device that includes the name of the SIS device, the name and/or path for the parameter to be changed in the SIS device, the new parameter value, and a confirmation button for the operator to confirm the secure write request. In this scenario, the evidence may include an indication of whether the operator selected the confirmation button.

[00137] The operator and/or the operator's computing device may broadcast transactions to the blockchain 802 that includes the evidence. The evidence may be cryptographically signed to provide cryptographic proof-of-identity that the evidence came from an operator and/or operator's computing device authorized to perform a secure write request. Accordingly, the smart contract may compare the provided identity to a list of operators and/or computing devices authorized to perform secure write requests. In some embodiments, the smart contract may compare the provided identity to a list of operators and/or computing devices authorized to perform secure write requests for the particular SIS device that is the target of the secure write request.

[00138] Another aspect of the secure write request smart contract state 806 is the smart contract data. Smart contract data may be thought of like the private and public data in an object created according to an object-oriented programming paradigm in that the smart contract data may be directly updated from outside the object, or the smart contract data may be updated only in limited ways, such as by calling a method of the smart contract. The smart contract data may include the name and/or path for the parameter to be changed in the SIS device, and the new parameter value. In some embodiments, the smart contract data may include an indication of whether the parameter information has been received intact. For example, the transaction including the parameter to be changed and parameter information may also include a CRC value or other error checking value. The smart contract may generate an expected CRC value based on the parameter to be changed and parameter information and compare the expected CRC value to the received CRC value. If the expected CRC value matches the received CRC value, the smart contract may determine that the parameter information has been received intact. Also in some embodiments, the smart contract data may include an indication of whether the secure write request was confirmed. For example, if the smart contract receives a transaction by the operator and/or operator's computing device indicating that the operator selected the confirmation button, the smart contract may determine that the secure write request was confirmed.

[00139] For example, as illustrated in FIG. 8, the smart contract data may include a parameter of lock/unlock the SIS device, a parameter value of '1' or 'lock' indicating to set the parameter to lock the SIS device, a confirmed value of '1,' 'yes,' or 'true' indicating that the secure write request was confirmed, and a received data intact value of '1,' 'yes,' or 'true' indicating that the parameter information has not been corrupted. Accordingly, the smart contract may determine that the new parameter value should be provided to the SIS device. Then the smart contract may provide the parameter information to the SIS device or to a controller communicatively coupled to the SIS device to perform the secure data write.

[00140] In some embodiments, the secure write request smart contract may provide parameter information to a target SIS device or to a controller communicatively coupled to the target SIS device when the operator and/or computing device sending the secure write request is authorized to perform secure data writes for the target SIS device, the parameter information has not been corrupted, and the secure write request is confirmed. In other embodiments, the secure write request smart contract does not determine whether the parameter information is

received intact. Instead, the secure write request smart contract provides a first instance of the parameter information including the parameter name and/or parameter path, the new parameter value, and the CRC value to the target SIS device or controller in response to receiving the secure write request. The secure write request smart contract also provides a second instance of the parameter information to the target SIS device or controller in response to receiving confirmation of the secure write request. The controller or target SIS device then determines whether the parameter information in both instances is the same and whether the parameter information has been received intact. When the parameter information in both instances is the same and the parameter information has been received intact, the controller or the target SIS device writes the new parameter value for the parameter to the target SIS device.

[00141] While FIG. 8 illustrates a smart contract state 806 for a secure write request, this is merely one example smart contract for ease of illustration only. Participants in the distributed ledger network (e.g., plant operators, configuration engineers, process control system designers, etc.) may deploy any suitable smart contracts related to process control.

[00142] In another example, a smart contract may be deployed that obtains device information for a device within the process plant 10 that experiences a failure, and provides the device information to a device supplier in response to receiving a request to share the device information. More specifically, when a device within the process plant 10 experiences a failure, such as a process plant entity, the device may transmit a transaction to an address for the smart contract stored on the distributed ledger. The transaction may be cryptographically signed to provide cryptographic proof-of-identity that the transaction came from the device. In other embodiments, the process plant entity may transmit an indication of the failure to a controller, field device, or other process control device which acts as an evidence oracle and generates the transaction. In any event, the transaction may include device information for the device, such as identification information for the device, the make, model, and year of the device, maintenance history for the device, the type of failure, damaged parts within the device, etc.

[00143] In some embodiments, the smart contract transmits the device information to a computing device of maintenance personnel within the process plant 10 for the maintenance personnel to review the device information. Upon reviewing the device information, the maintenance personnel may determine that the device information needs to be reviewed by the device supplier for further investigation into the failure and/or for providing a replacement device or replacement parts. Accordingly, the maintenance personnel's computing device may generate a transaction requesting the smart contract provide the device information to the device supplier. The transaction may be cryptographically signed to provide cryptographic proof-of-identity that the transaction came from the maintenance personnel. In response to determining that the request to provide the device information to the device supplier came from authorized maintenance personnel, the smart contract may provide the device information to a computing device of the device supplier.

[00144] Another example smart contract is a smart contract that obtains a token value from a first process plant, determines that a product meeting certain quality standards transferred from a second process plant to the first process plant, and provides the token value to the second process plant. In some embodiments, the smart contract may receive an indication that the product has been received at the first process plant from an evidence

oracle such as a field device in the first process plant. The field device may also provide parameter data related to the product which the smart contract compares to a set of quality metrics to determine whether the products meets the quality standards. If the product meets the quality standards, the smart contracts provides the token value to the second process plant. Otherwise, the smart contract may return the token value to the first process plant.

Types of Transactions Recorded in Distributed Ledgers in a Process Control System

[00145] The process control system distributed ledgers may include many different types of transactions related to process control. These transactions may include 1) transactions related to delivery or receipt of a product at a process plant 10 and the quantity delivered/received; 2) transactions related to software or firmware upgrades at devices within the process plant 10, such as operator workstations, server devices, controllers, I/O devices, network devices, field devices, etc.; 3) transactions related to quality control, production, or regulatory reporting in the process plant 10; 4) transactions recording process plant data; and 5) transactions recording chain of custody via product tracking data.

[00146] In some scenarios, the transactions are provided to smart contracts to alter a smart contract state, for example. In other scenarios, the transactions are not provided to smart contracts and are merely recorded in the distributed ledger as a secure, immutable, and trustless record of information related to one or several process plants.

Transactions Related to Delivery or Receipt of a Product and the Quantity Delivered/Received

[00147] FIG. 9 depicts an exemplary transaction 906 representing an evidence transaction reporting the amount of oil received at a process plant 10 from an oil pipeline. While the exemplary transaction 906 in FIG. 9 reports the amount of oil from an oil pipeline, this is merely one example for ease of illustration only. Other materials or products from other sources may also be reported such as electricity from an electric power generation system, a product over rail, automotive, marine, or airborne transportation, a product through a liquid, gas, steam, fuel, or materials pipeline, or water from a water distribution system. In any event, the transaction 906 may be generated by a field device acting as an evidence oracle. When the field device detects oil flowing through a valve, the field device broadcasts a transaction 906 to blockchain 902 to be included in a block, such as block 904.

[00148] The transaction 906 may include a transaction ID and an originator such as field device 456 in Plant A (identified by a cryptographic proof-of-identity). The transaction 906 may also include identification information related to the product, the provider of the product (e.g., an oil producer) and information regarding the quantity of the product received. For example, the field device may be a flow rate sensor that determines the volume of oil obtained at Plant A over a particular time period (e.g., an hour, a day, etc.) and includes the volume in the transaction. In other embodiments, the field device may include several flow rates at various time periods in a series of transactions, and the flow rates as a function of time may be used to determine the amount of oil received at Plant A. Furthermore, the transaction 906 may include a cryptographic hash of the information regarding the event, the product identifier, and the product provider identifier. In another implementation, the

information regarding the event, the product identifier, and the product provider identifier is not stored as a cryptographic hash, but is directly accessible in block 904 by an observer or other network participant.

[00149] While in this example, the field device for the process plant 10 that receives the product generates a transaction, a field device for a process plant 10 or other entity that provides the product may generate a transaction. This transaction may be generated in addition or as an alternative to the transaction by the field device for the process plant 10 that receives the product.

Transactions Related to Software or Firmware Upgrades at Devices within the Process Plant

[00150] To prevent unauthorized software or firmware from being introduced into a process plant 10, software and firmware upgrades to devices within the process plant 10 may be digitally recorded in a distributed ledger, such as the distributed ledgers described above. The distributed ledger may maintain a record of each software and firmware upgrade to a device within the process plant 10 including the time and date of the upgrade, the identity of the user performing the upgrade (via a cryptographic proof-of-identity), changes to the previous version of the software, and/or the new version of the software. A server device 12 or other computing device within the process plant 10 may continuously or periodically (e.g., once per second, once per minute, once per hour, once per day, etc.) obtain current versions of software and firmware running in devices in the process plant 10. The server device 12 may also retrieve the transactions from the distributed ledger and compare the current software or firmware in a device to the latest version of the software or firmware recorded in the distributed ledger. In some embodiments, the distributed ledger stores a cryptographic hash of the new version of the software or firmware, and compares the current software or firmware executing in the device to the cryptographic hash value to verify the software or firmware has not been tampered with.

[00151] If the current software or firmware in the device does not match with the latest version of the software or firmware recorded in the distributed ledger, the server device 12 may prevent the device from executing the current software or firmware. In some embodiments, the server device 12 may cause the software or firmware in the device to revert back to a previous version, for example by downloading the previous version to the device. In this manner, unauthorized users are unable to tamper with the software or firmware executing in the process plant 10.

[00152] FIG. 10 depicts an exemplary transaction 1006 representing an evidence transaction reporting a software or firmware update in a device within a process plant 10. The transaction 1006 may be generated by the device receiving the upgrades, such as an operator workstation, another user interface device 8, a server device 12, a controller 11, an I/O device 26, 28, a network device, a field device 15-22, 40-46, etc. A network device in the process plant 10 may include for example, a wireless gateway 35, a router 58, a wireless access point 7a, 55, an edge gateway, a wireless adaptor 52, etc.

[00153] The transaction 1006 may include a transaction ID and an originator modifying the software or firmware such as John Doe (identified by a cryptographic proof-of-identity). The transaction 1006 may also include identification information (Operator Workstation 1234) for the device executing the software or firmware (identified by a cryptographic proof-of-identity), a description including a version number and time and date of the upgrade ("Update to version 10.3.1.4 on January 15, 2019 at 6:02 a.m."). Furthermore, the transaction 1006

may include a cryptographic hash of the software instructions for the new version of the software. In another implementation, the new version of the software is not stored as a cryptographic hash, but is directly accessible in block 1004 by an observer or other network participant. In some embodiments, the consensus rules indicate that only authorized users may record software or firmware updates on the distributed ledger. Accordingly, when the transaction 1006 is broadcasted to the distributed ledger, the validating nodes validate the transaction 1006 if the originator is an authorized user. If the originator is not an authorized user, the transaction 1006 is not included in the distributed ledger and the update to the software will not match with the latest version of the software recorded in the distributed ledger.

[00154] In an exemplary scenario, at 6:03 a.m. on January 15, 2019, a server device 12 in the process plant 10 obtains the state of the software executing in Operator Workstation 1234 and compares the software to the cryptographic hash of the software instructions for the new version of the software in the distributed ledger by for example, performing a cryptographic hash of the software instructions executing in Operator Workstation 1234. If the cryptographic hashes are the same, the server device 12 determines the software has not been tampered with. If, on the other hand, the cryptographic hashes differ, the server device determines the software has been tampered with and prevents Operator Workstation 1234 from executing the software in its current state. The server device 12 then downloads the previous state of the software to the Operator Workstation 1234, and the Operator Workstation 1234 resumes executing the software in its previous state.

05 09 22

Transactions Related to Quality Control, Production, or Regulatory Reporting in the Process Plant

[00155] Process plants have reporting and recordkeeping requirements to comply with regulatory agencies, such as the Environmental Protection Agency (EPA). For example, the EPA promulgated Leak Detection and Repair (LDAR) regulations to minimize the emission of fugitive volatile organic compounds and hazardous air pollutants from for example, leaking equipment such as valve, pumps, and connectors in process plants. To comply with the regulations and provide a secure, immutable, and trustless record, regulatory data may be recorded in a distributed ledger. For example, in response to a triggering event such as an alarm, an error, a leak, a repair event, a process milestone, a corrective action, etc., process control elements such as field devices, controllers, or process plant entities may generate transactions including data from the triggering event, such as the time in which the event occurred, the duration of the event, process parameter values for process plant entities involved in the event, product parameter values for products involved in the event, etc. The regulatory data is then recorded in the distributed ledger, so that regulatory agencies can review the data.

[00156] In some embodiments, when a triggering event occurs, the triggering event is detected by one of the process control elements. The process control element then notifies other process control elements of the triggering event and assigns a unique identifier to the triggering event. In this manner, each of the process control elements may collect measurements related to the triggering event and broadcast transactions to the distributed ledger, where each transaction includes the same unique identifier for the triggering event.

[00157] In some embodiments, regulatory data is recorded in a public blockchain so that anyone can view the regulatory data from a process plant 10. In other embodiments, the regulatory data is recorded in a private or permissioned blockchain accessible to the process plant 10 and the regulatory agency. In yet other

embodiments, the regulatory data is recorded in a private or permissioned blockchain accessible to several process plants in a process plant network along with the regulatory agency.

[00158] FIG. 11 depicts an exemplary transaction 1106 representing an evidence transaction reporting process parameter or product parameter data. The transaction 1106 may be generated by a process plant entity which may be a device within a process plant 10 for use in a portion of the process which contains, transforms, generates, or transfers physical materials, such as a valve, a tank, a mixer, a pump, a heater, etc.

[00159] The transaction 1106 may include a transaction ID and an originator (Heater Y-001) collecting the product or process parameter measurement (identified by a cryptographic proof-of-identity). The transaction 1106 may also include identification information related to the product, product parameter data (e.g., the product's temperature has been maintained at 100°C for 2 hours), and process parameter data (e.g., the temperature in Heater Y-001 is 120°C). When the transaction 1106 is generated in response to a triggering event, the transaction 1106 may also include identification information for the triggering event and event data from the triggering event, such as a time of the triggering event, a duration of the triggering event, and/or a description of the triggering event. In some scenarios, multiple process plant entities generate transactions in response to the same triggering event and communicate with each other to assign a unique identifier to the triggering event. In this manner, a party such as a regulatory agency reviewing the distributed ledger may view each of the transactions associated with the same triggering event.

[00160] Furthermore, the transaction 1106 may include a cryptographic hash of the product and/or process parameter data along with data related to a triggering event. In another implementation, the product parameter data, process parameter data, and other data related to a triggering event is not stored as a cryptographic hash, but is directly accessible in block 1104 by an observer or other network participant.

[00161] As described above, triggering events may include alarms, errors, leaks, repair events, corrective actions, etc. In an example scenario, the triggering event may be a leak in the process plant 10 caused by the opening of a relief valve. The relief valve may open when the pressure in the process control system exceeds a threshold amount of pressure, or the relief valve may open in proportion to the amount of pressure detected at the valve. When the relief valve opens, the relief valve or one or several other field devices may detect the time of the opening, the duration of the opening, the size of the opening, the pressure in the relief valve when it opened, the flow rate of fluid leaking out of the relief valve, and/or properties of the fluid such as the temperature of the fluid, the type of fluid, etc. In some embodiments, the amount of fluid leaking out of the relief valve may also be determined based on the flow rate, the size of the opening, and the duration of the opening of the relief valve. Then the relief valve and/or one or several other field devices may generate transactions, similar to the transaction 1106 including the same unique identifier for the triggering event, and/or the same description for the triggering event of a leak caused by the opening of the relief valve. Each of the transactions may also include process parameter data, such as the time of the opening, the size of the opening, the pressure in the relief valve, the flow rate of the fluid leaking out of the relief valve, etc. The transactions may also include product parameter data, such as the properties of the fluid. The devices generating the transactions then broadcast the transactions to the distributed ledger network for validating nodes, such as edge gateways to confirm the transactions are valid and include the transactions in the distributed ledger.

[00162] A regulatory agency reviewing the incident may request and obtain event data from the distributed ledger that is included in transactions having the triggering event identifier. The regulatory agency's computing device, such as the computing device 235 as shown in FIG. 2, may then present the event data on a user interface. In other embodiments, the distributed ledger includes cryptographic hashes of the event data which are provided to the regulatory agency's computing device 235 in response to a request to authenticate the event data. The event data is obtained from other data sources such as a database communicatively coupled to a server device 12 in the process plant 10. The regulatory agency's computing device 235 then computes a cryptographic hash of the obtained event data and compares the cryptographic hash of the obtained event data to the cryptographic hash of the event data from the distributed ledger. If the cryptographic hashes are the same, the regulatory agency's computing device 235 determines that the event data from the database has not been tampered with. Otherwise, the regulatory agency's computing device 235 determines that the event data from the database is unreliable.

Transactions Recording Process Plant Data

[00163] In addition to recording process parameter data and product parameter data in transactions related to a triggering event, process and product parameter data may be included in transactions unrelated to a triggering event for example, for maintaining accurate records of the operations of a process plant 10. Other types of process plant data may also be included in transactions, such as configuration data, user interaction data, maintenance data, commissioning data, plant network data, product tracking data, or any other suitable data generated in or related to one or several process plants. User interaction data may include operations performed by an operator or a configuration engineer for example, at an operator workstation. The operator may adjust set points, respond to alarms, etc., via user controls at the operator workstation which may be included in transactions as user interaction data. In this manner, when a competing entity calls into question the quality of a product manufactured in the process plant 10, the process plant 10 may retrieve process plant data from the distributed ledger that is related to the product. The process plant 10 may then review records of each of the process plant entities involved in manufacturing the product, parameter values for the process plant entities as the product was manufactured, parameter values for the product at various stages in the manufacturing process, triggering events that occurred during the manufacturing of the product, etc. Accordingly, the process plant 10 may determine whether the product was manufactured properly to meet certain quality standards or whether an abnormality occurred during production causing the product to fail to meet the quality standards.

[00164] The process plant data may also be used to perform a root-cause analysis on products. For example, products may have predicted shelf-lives such as gasoline which has a half-life that may be less than a month. In some embodiments, a computing device may predict the shelf-life of a product based on the characteristics of the product including process parameter data and product parameter data recorded in the distributed ledger while the product was manufactured. The computing device may also predict the shelf-life of the product based on historical data for similar products having similar components and/or process parameter data and product parameter data during manufacturing. More specifically, the computing device may predict the shelf-life of a product based on the average shelf-life of the same type of product (e.g., gasoline).

[00165] The computing device may then increase or decrease the predicted shelf-life from the average shelf-life based on the quality of components in the product. For example, components may be categorized as above average, average, or below average. Indications of components may be stored in a database with associated rankings or quality scores. Components having a quality score below a first threshold score or a ranking below a first threshold ranking may be categorized as below average. Components having a quality score above a first threshold score and below a second threshold score or ranking above the first threshold ranking and below a second threshold ranking may be categorized as average. Components having a quality score above the second threshold score or ranking above the second threshold ranking may be categorized as above average.

[00166] The computing device may further increase or decrease the predicted shelf-life depending on the properties of the product, such as a temperature of the product, a volume of the product, a mass of the product, a density of the product, a pressure of the product, a viscosity of the product, a chemical composition of the product, etc. For example, the computing device may assign a quality score to each property and adjust the predicted shelf-life based on each of the quality scores.

[00167] In some embodiments, the computing device may generate a machine learning model to predict the shelf-life of a product based on the actual shelf-lives of previous products, the components in the previous products, and the properties of the previous products.

[00168] Furthermore, when the actual shelf-life of a product differs from the predicted shelf-life, the computing device may retrieve process plant data related to the product from the distributed ledger to identify the cause. For example, the actual shelf-life may be lower than the predicted shelf-life due to poor quality components in the product. In another example, the actual shelf-life may be lower than the predicted shelf-life due to a heater in the process plant 10 heating the product to an undesirable temperature.

Transactions Recording Chain of Custody via Product Tracking Data

[00169] To provide accurate records of the chain of custody of products in a supply chain, transactions may be generated that include identification information for the source or supplier of a product, and the entities that handled the product, such as manufacturers, distributers, distribution facilities, retailers, and the customer who purchases the product. More specifically, the transactions may include product tracking data having identification information for the product, identification information for the supplier/manufacturer of the product, identification information for manufacturers/providers of each of the components of the product, identification information for entities in the supply chain which receive and handle the product, identification information for retailers that sell the product, and/or identification information for customers that purchase the product. When the product is delivered from one entity (e.g., a process plant) to another entity (e.g., a warehouse), the delivering entity may generate a transaction that includes identification information for the delivering entity, identification information for the receiving entity, and an indication that the product is being transferred to the receiving entity.

[00170] Accordingly, a user such as a customer, via a user interface device, may retrieve each of the transactions involving a particular product from the distributed ledger using the identification information for the product. The user interface device may then display, via a user interface, indications of the supplier or source of

the product, and the entities that handled the product, such as manufacturers, distributers, distribution facilities, retailers, and the customer who purchases the product. The user interface device may also display indications of the components of the product via the user interface. The user may then retrieve each of the transactions involving a particular component of the product from the distributed ledger using identification information for the component. Then the user interface device may display, via the user interface, indications of the supplier or source of the component, and the entities that handled the component, such as manufacturers, distributers, distribution facilities, etc.

[00171] In some embodiments, the product packaging may include a product identifier, such as a barcode or radio frequency identification (RFID) tag that when scanned, provides data from the distributed ledger for the product. For example, a user may scan the barcode or RFID tag via a mobile device which then presents indications of the supplier or source of the product, and the entities that handled the product on the mobile device.

[00172] FIG. 12 illustrates a flow diagram representing an exemplary method 1200 for recording data in a process control system using a distributed ledger. The method 1200 may be executed by a field device 15-22, 40-46 within the process plant 10, a controller 11 in the process plant 10, or another computing device within the process plant 10 such an operator workstation, server device 12, user interface device 8, I/O device 26, 28, network device 35, etc.

[00173] At block 1202, data related to a process control element is obtained from a field device. The process control element may be a field device, a controller, or a process plant entity such as a valve, a tank, a mixer, a pump, a heat exchanger, etc. The data may include process plant data, such as process parameter data for parameters of the process control element (e.g., a tank fill level, a pump speed, the temperature in a heat exchanger), and product parameter data for a product entering, exiting, within, and/or controlled by the process control element (e.g., the temperature of a fluid in a tank, the flow rate of fluid exiting a valve). Then at block 1204, a transaction is generated that includes the process plant data related to the process control element. The entity generating the transaction (e.g., a field device) signs the transaction with a cryptographic signature unique to the entity (block 1206) and augments the transaction with identity data for the entity such as a public cryptographic key owned by the entity (block 1208). For example, the transaction may be signed by a private cryptographic key corresponding to the public cryptographic key owned by the entity.

[00174] At block 1210, the transaction is transmitted to a participant in the distributed ledger network. For example, a field device may broadcast the transaction to the distributed ledger network. A validating node such as an edge gateway may then confirm the transaction is valid, add the transaction to a block of transactions, solve a cryptographic puzzle, and include the solution in the newly generated block as proof of the work done to generate the block. The validating node may then provide the newly generated block to each of the other validating nodes in the distributed ledger network to include the newly generated block in their respective copies of the distributed ledger.

[00175] In some embodiments, the validating node confirms the transaction against a set of consensus rules and adds the transaction to a block when the transaction satisfies each of the consensus rules. For example, a

consensus rule may include that the originator of a transaction supply a proof-of-identity such that only approved entities may originate transactions to the distributed ledger. A consensus rule may require that blocks and transactions adhere to format requirements and supply certain meta information regarding the transaction (e.g., blocks must be below a size limit, transactions must include a number of fields, etc.). Any transaction that does not satisfy the consensus rule is disregarded by validating nodes that receive the transaction and the transaction is not propagated to other nodes.

[00176] The validating node includes a transceiver to communicate with field devices, controllers, or other computing devices in the process plant 10 that broadcast transactions having distributed ledger data, such as process plant data. Additionally, the validating node may include a memory for storing a copy of the distributed ledger including a state database for storing states of smart contracts deployed on the distributed ledger. Furthermore, the validating node may include applications, such as a process data validator that applies a set of consensus rules to the distributed ledger data and appends the distributed ledger data to the validating node's copy of the distributed ledger if the distributed ledger data satisfies the consensus rules.

[00177] FIG. 13 illustrates a flow diagram representing an exemplary method 1300 for secure metering of untrusted data in a process control system using a distributed ledger. The method 1300 may be executed by a field device 15-22, 40-46 within the process plant 10, a controller 11 in the process plant 10, or another computing device within the process plant 10 such an operator workstation, server device 12, user interface device 8, I/O device 26, 28, network device 35, etc. The method 1300 may also be executed by a validating node such as edge gateway or a combination of a field device and a validating node.

[00178] At block 1302, data related to a process control element is obtained from a field device. The process control element may be a field device, a controller, or a process plant entity such as a valve, a tank, a mixer, a pump, a heat exchanger, etc. The data may include process plant data, such as process parameter data for parameters of the process control element (e.g., a tank fill level, a pump speed, the temperature in a heat exchanger), and product parameter data for a product entering, exiting, within, and/or controlled by the process control element (e.g., the temperature of a fluid in a tank, the flow rate of fluid exiting a valve). Then at block 1304, a transaction is generated that includes the process plant data related to the process control element. The entity generating the transaction (e.g., a field device) signs the transaction with a cryptographic signature unique to the entity and augments the transaction with identity data for the entity such as a public cryptographic key owned by the entity. For example, the transaction may be signed by a private cryptographic key corresponding to the public cryptographic key owned by the entity.

[00179] At block 1306, the transaction is transmitted to a participant in a local distributed ledger network. There may be several local distributed ledgers, where each local distributed ledger is maintained by a different party or process plant. For example, a local distributed ledger network for Plant A may be made up of edge gateways within Plant A. The edge gateways may record transactions that include process plant data related to events and devices within Plant A. Transactions are then added to the local distributed ledger network for a threshold time period or epoch. After the threshold time period has expired (block 1308), the validating nodes maintaining the local distributed ledger provide the transactions or blocks of transactions generated during the threshold time period to a global distributed ledger network (block 1310). The global distributed ledger network

may include validating nodes across multiple process plants, such as a cloud service having several cloud computing systems. The validating nodes may maintain a global distributed ledger (e.g., a global blockchain) for each process plant. Then the validating nodes in the local distributed ledger network may remove or prune blocks from the local distributed ledger that have been provided to the global distributed ledger other than the most recent block. The validating nodes for the local distributed ledger may continue to generate blocks, broadcast the blocks to the global distributed ledger network after each time epoch expires, and remove local copies of the blocks when the blocks have been added to the global blockchain.

[00180] Also in some embodiments, each of the global blockchains for the respective entities or process plants are combined to create a super blockchain having state blocks. Each state block includes each of the blocks from the global blockchains corresponding to a particular time period or epoch.

[00181] FIG. 14 illustrates a flow diagram representing an exemplary method 1400 for recording quality control, production, or regulatory data in a process control system using a distributed ledger. The method 1400 may be executed by a field device 15-22, 40-46 within the process plant 10, a controller 11 in the process plant 10, or another computing device within the process plant 10 such an operator workstation, server device 12, user interface device 8, I/O device 26, 28, network device 35, etc.

[00182] At block 1402, a triggering event related to quality control is detected by a process control element. The triggering event may be an alarm, an error, a leak, a repair event, a process milestone, a corrective action, etc. In some embodiments, an indication of the triggering event is provided to a field device, controller, or other computing device within the process plant 10. In other embodiments, the field device, controller, or other computing device detects the triggering event.

[00183] In any event, at block 1404, event data is obtained for the triggering event. The event data may include a unique identifier for the triggering event, a time of the triggering event, a duration of the triggering event, a description of the triggering event, identification information for the process control elements involved in the triggering event, identification information for a product being manufactured by the process control elements during the triggered event, etc. Then at block 1406, a transaction is generated that includes the event data and/or a cryptographic hash of the event data for the triggering event. The transaction may also include identification information for the originator of the transaction, product parameter data for the product as the triggering event occurred, process parameter data for process control elements during the triggering event, or any other suitable information. In some embodiments, several field devices, controllers, or other computing devices within the process plant 10 may generate transactions related to the triggering event. For example, a first field device may generate a transaction that includes the temperature in a heater at the time of the triggering event, while a second field device may generate a transaction that includes the speed of a pump at the time of the triggering event.

[00184] At block 1408, the transaction is transmitted to a participant in the distributed ledger network. For example, a field device may broadcast the transaction to the distributed ledger network. A validating node such as an edge gateway may then confirm the transaction is valid, add the transaction to a block of transactions, solve a cryptographic puzzle, and include the solution in the newly generated block as proof of the work done to

generate the block. The validating node may then provide the newly generated block to each of the other validating nodes in the distributed ledger network to include the newly generated block in their respective copies of the distributed ledger.

[00185] As described above, the transaction may include a cryptographic hash of the event data for the triggering event and/or a combination of the event data for the triggering event and other process plant data related to the triggering event. In addition to generating the transaction, the field device may provide the event data or other process plant data related to the triggering event to a server device 12 to be stored in a database, for example (block 1410).

[00186] Then to authenticate the event data, the event data stored in the database is compared to the cryptographic hash included in the distributed ledger (block 1412). If there is a match, the event data has not been tampered with. For example, a regulatory agency reviewing an incident may request and obtain cryptographic hashes of event data from the distributed ledger that is included in transactions having the triggering event identifier. The event data is obtained from other data sources such as a database communicatively coupled to a server device 12 in the process plant 10. The regulatory agency's computing device then computes a cryptographic hash of the obtained event data and compares the cryptographic hash of the obtained event data to the cryptographic hash of the event data from the distributed ledger. If the cryptographic hashes are the same, the regulatory agency's computing device determines that the event data from the database has not been tampered with. Otherwise, the regulatory agency's computing device determines that the event data from the database is unreliable. In other embodiments, a computing device within the process plant 10 retrieves the event data stored in the database and the cryptographic hash of the event data from the distributed ledger and compares the event data to the cryptographic hash to authenticate the event data.

[00187] FIG. 15 illustrates a flow diagram representing an exemplary method 1500 for recording states of software or firmware in a process control system and connected instrumentation using a distributed ledger. The method 1500 may be executed by a field device 15-22, 40-46 within the process plant 10, a controller 11 in the process plant 10, or another computing device within the process plant 10 such an operator workstation, server device 12, user interface device 8, I/O device 26, 28, network device 35, etc.

[00188] At block 1502, a current state of software or firmware executing on a device in the process plant 10 is obtained. For example, a device within the process plant 10 that receives a software or firmware upgrade may obtain the new version of the software or firmware. The device may be an operator workstation, another user interface device 8, a server device 12, a controller 11, an I/O device 26, 28, a network device 35, a field device 15-22, 40-46, etc. Then at block 1504, the device may generate a transaction that includes an indication of the current state of the software or firmware. For example, the indication may be a cryptographic hash of the software instructions for the new version of the software. The transaction may also include an originator modifying the software or firmware identified by a cryptographic proof-of-identity, identification information for the device executing the software or firmware, a description of the upgrade, a time and date of the upgrade, etc.

[00189] At block 1506, the transaction is transmitted to a participant in the distributed ledger network. For example, a computing device may broadcast the transaction to the distributed ledger network. A validating node such as an edge gateway may then confirm the transaction is valid, add the transaction to a block of transactions, solve a cryptographic puzzle, and include the solution in the newly generated block as proof of the work done to generate the block. The validating node may then provide the newly generated block to each of the other validating nodes in the distributed ledger network to include the newly generated block in their respective copies of the distributed ledger.

[00190] In some embodiments, the validating node confirms the transaction against a set of consensus rules and adds the transaction to a block when the transaction satisfies each of the consensus rules. Also in some embodiments, the consensus rules indicate that only authorized users may record software or firmware updates on the distributed ledger. Accordingly, when the transaction is broadcasted to the distributed ledger, the validating nodes validate the transaction if the originator is an authorized user. If the originator is not an authorized user, the transaction is not included in the distributed ledger and the update to the software will not match with the latest version of the software recorded in the distributed ledger.

[00191] In any event, at block 1508, a state of software or firmware executing on the device in the process plant 10 is obtained. For example, a server device 12 or other computing device within the process plant 10 may continuously or periodically (e.g., once per second, once per minute, once per hour, once per day, etc.) obtain current versions of software and firmware running in devices in the process plant 10. The state of the software or firmware obtained at the server device 12 is then compared to the cryptographic hash value for the software or firmware stored in the distributed ledger to verify the software or firmware has not been tampered with (block 1510). If the state of the software or firmware matches the cryptographic hash value for the software or firmware stored in the distributed ledger, the software or firmware continues executing on the device (block 1514). Otherwise, the server device 12 determines the software has been tampered with and prevents the device from executing the software in its current state (block 1512). In some embodiments, the server device 12 then downloads the previous state of the software to the device, and the device resumes executing the software in its previous state.

[00192] FIG. 16 illustrates a flow diagram representing an exemplary method 1600 for creating smart contracts in a process control system using a distributed ledger. The method 1600 may be executed by a field device 15-22, 40-46 within the process plant 10, a controller 11 in the process plant 10, or another computing device within the process plant 10 such an operator workstation, server device 12, user interface device 8, I/O device 26, 28, network device 35, etc.

[00193] At block 1602, a smart contract is generated that is related to one or several process plants. For example, the smart contract may transfer a token value from Plant A to Plant B when Plant A receives a product from Plant B that meets certain quality standards. Another example smart contract in a process control system may include a secure write request smart contract that allows plant personnel to write parameter data to an SIS device in the process plant 10. Yet another example smart contract in a process control system may include a device information smart contract that obtains device information from a device experiencing a failure, and

provides the device information to a device supplier in response to receiving a request to share the device information.

[00194] At block 1604, the smart contract is deployed to an address stored on the distributed ledger. The deployed smart contract may expose methods and data to other participants in the distributed ledger network. Some of the data in the smart contract state may be private data that may only be altered by calling a method of the smart contract, or only altered by authorized distributed ledger participants. One way of altering the smart contract state is to broadcast a transaction to the distributed ledger network. If the broadcasted transaction satisfies consensus rules, network validators may include the transaction in the distributed ledger.

[00195] In some embodiments, validating nodes such as edge gateways execute the code contained in the smart contract and field devices act as evidence oracles and provide evidence transactions which alter the smart contract state.

[00196] FIG. 17 illustrates a flow diagram representing an exemplary method 1700 for interacting with a smart contract in a process control system using a distributed ledger. The method 1700 may be executed by a field device 15-22, 40-46 within the process plant 10, a controller 11 in the process plant 10, or another computing device within the process plant 10 such an operator workstation, server device 12, user interface device 8, I/O device 26, 28, network device 35, etc.

[00197] At block 1702, event data is obtained from an event occurring within the process plant 10. An event may be a product being delivered by or received at a process plant 10, the completion of a product being manufactured at the process plant 10, a change in the properties of a product, a change in a process parameter value, a triggering event such as an alarm, an error, a leak, a repair event, a corrective action, a user interaction such as a request to write to an SIS device, a request to provide device information to a device supplier, or a request to transfer a token value when a particular product is received, or any other suitable event occurring within the process plant 10. The event data may include process parameter data, product parameter data, configuration data, user interaction data, maintenance data, commissioning data, plant network data, product tracking data, or any other suitable data related to the event, such as the date and time of the event, the duration of the event, a description of the event, etc.

[00198] Then at block 1704, a transaction is generated that includes the event data and identification information for the entity generating the transaction, such as a cryptographic public key assigned to the entity. The transaction may be cryptographically signed to provide cryptographic proof-of-identity of the entity generating the transaction. At block 1706, the transaction is transmitted to the address on the distributed ledger where the smart contract is deployed. In this manner, validating nodes such as edge gateways alter the smart contract state according to the event data included in the transaction.

[00199] For example, a smart contract may transfer a token value from Plant A to Plant B when Plant A receives a product from Plant B that meets certain quality standards. A field device in Plant A may generate a transaction including event data that is related to the quality of the product, such as an identification information for Plant A, identification information for the product, an indication that the product was received from Plant B, and product parameter data which describes properties of the product (e.g., the temperature of the product, the

volume of the product, the density of the product, the viscosity of the product, or the chemical composition of the product). The field device may provide the transaction to the address for the smart contract, and the validating nodes may alter the smart contract state to include the product parameter data. In some embodiments, the smart contract compares the properties of the product included in the product parameter data to a set of minimum threshold requirements for the product to satisfy the appropriate quality standards. If the product satisfies the quality standards, the smart contract may transfer the token value to Plant B. In some embodiments, a field device in Plant B may generate a transaction including event data that is related to the quality of the product, such as process parameter data which describes parameter values for process plant entities in Plant B involved in manufacturing the product, where the parameter values are collected as the product is being manufactured.

[00200] When implemented in software, any of the applications, services, and engines described herein may be stored in any tangible, non-transitory computer readable memory such as on a magnetic disk, a laser disk, solid state memory device, molecular memory storage device, or other storage medium, in a RAM or ROM of a computer or processor, etc. Although the example systems disclosed herein are disclosed as including, among other components, software and/or firmware executed on hardware, it should be noted that such systems are merely illustrative and should not be considered as limiting. For example, it is contemplated that any or all of these hardware, software, and firmware components could be embodied exclusively in hardware, exclusively in software, or in any combination of hardware and software. Accordingly, while the example systems described herein are described as being implemented in software executed on a processor of one or more computer devices, persons of ordinary skill in the art will readily appreciate that the examples provided are not the only way to implement such systems.

[00201] Thus, while the present invention has been described with reference to specific examples, which are intended to be illustrative only and not to be limiting of the invention, it will be apparent to those of ordinary skill in the art that changes, additions or deletions may be made to the disclosed embodiments without departing from the spirit and scope of the invention.

[00202] When used in this specification and claims, the terms "comprises" and "comprising" and variations thereof mean that the specified features, steps or integers are included. The terms are not to be interpreted to exclude the presence of other features, steps or components.

[00203] The features disclosed in the foregoing description, or the following claims, or the accompanying drawings, expressed in their specific forms or in terms of a means for performing the disclosed function, or a method or process for attaining the disclosed result, as appropriate, may, separately, or in any combination of such features, be utilised for realising the invention in diverse forms thereof.

CLAIMS:

1. A method for secure metering of untrusted data in process control systems using a distributed ledger maintained by a plurality of participants, the method comprising:

collecting, by a field device performing a physical function to control an industrial process in a process plant, a measurement of a parameter within the process plant;

obtaining, by a computing device, the measurement of the parameter;

generating a transaction including the measurement; and

transmitting the transaction to at least one other participant in a local distributed ledger network of participants maintaining a local distributed ledger;

after a threshold time period, transmitting a plurality of transactions generated during the threshold time period to at least one participant in a global distributed ledger network of participants maintaining a global distributed ledger, wherein the plurality of transactions includes the transaction which includes the measurement.

2. The method of claim 1, further comprising:

adding the transaction to a local block of transactions;

solving a cryptographic puzzle based on the local block of transactions;

adding the solution to the cryptographic puzzle to the local block of transactions; and

transmitting the local block of transactions to at least one other participant in the local distributed ledger network.

3. The method of claim 2, further comprising:

after the threshold time period, transmitting one or more local blocks of transactions generated during the threshold time period to at least one participant in the global distributed ledger network.

4. The method of any preceding claim, further comprising:

after the threshold time period, pruning at least some of the plurality of transaction generated during the threshold time period from the local distributed ledger network.

5. The method of any preceding claim, wherein the global distributed ledger is a permissioned blockchain viewable by a plurality of entities operating a plurality of process plants.

6. The method of claim 5, wherein the parameter is related to a shared resource between the plurality of entities operating the plurality of process plants.

7. The method of claim 5 or 6, wherein the global distributed ledger includes a plurality of global distributed ledgers corresponding to the plurality of entities, each global distributed ledger including transactions stored in the local distributed ledger for a same respective entity as the global distributed ledger.

8. The method of claim 7, further comprising:

for transactions generated during the threshold time period, adding the transaction from each of the plurality of global distributed ledgers to a state block of transactions;

solving a cryptographic puzzle based on the state block of transactions;

adding the solution to the cryptographic puzzle to the state block of transactions; and

transmitting the state block of transactions to at least one other participant in a super blockchain network of participants maintaining a super blockchain.

9. The method of any preceding claim, wherein the local distributed ledger is a private blockchain viewable by an entity operating the process plant.

10. The method of any preceding claim, wherein generating a transaction including the measurement includes generating the transaction including a cryptographic hash value corresponding to the measurement.

11. The method of any of claims 6 to 10, wherein the shared resource between the plurality of entities operating the plurality of process plants is a fluid in a fluid pipeline, and the parameter measurement is an amount of fluid obtained by one of the plurality of entities from the fluid pipeline.

12. A system for secure metering of untrusted data in process control systems using a distributed ledger maintained by a plurality of participants comprising:

one or more field devices disposed in a process plant each performing a physical function to control an industrial process, the one or more field devices configured to collect measurements of parameters within the process plant and provide the parameter measurements to one or more edge gateway devices; and

the one or more edge gateway devices executing in the process plant each including:

one or more processors;

a communication unit; and

a non-transitory computer-readable medium coupled to the one or more processors and the communication unit and storing instructions thereon, that when executed by the one or more processors, causes the edge gateway device to:

obtain at least one of the parameter measurements;

generate a transaction including the measurement; and

transmit the transaction to at least one other edge gateway in a local distributed ledger network of edge gateways maintaining a local distributed ledger; and

after a threshold time period, transmit a plurality of transactions generated during the threshold time period to at least one participant in a global distributed ledger network of participants maintaining a global distributed ledger, wherein the plurality of transactions includes the transaction which includes the measurement.

13. The system of claim 12, wherein the instructions further cause the edge gateway to:

add the transaction to a local block of transactions;

solve a cryptographic puzzle based on the local block of transactions;

add the solution to the cryptographic puzzle to the local block of transactions; and

transmit the local block of transactions to at least one other edge gateway in the local distributed ledger network.

14. The system of claim 13, wherein the instructions further cause the edge gateway to:

after the threshold time period, transmit one or more local blocks of transactions generated during the threshold time period to at least one participant in the global distributed ledger network.

15. The system of any of claims 12-14, wherein the instructions further cause the edge gateway to:

after the threshold time period, prune at least some of the plurality of transactions generated during the threshold time period from the local distributed ledger network.

16. The system of any of claims 12-15, wherein the global distributed ledger is a permissioned blockchain viewable by a plurality of entities operating a plurality of process plants.

17. The system of claim 16, wherein the parameter is related to a shared resource between the plurality of entities operating the plurality of process plants.

18. The system of claim 16 or 17, wherein the global distributed ledger includes a plurality of global distributed ledgers corresponding to the plurality of entities, each global distributed ledger including transactions stored in the local distributed ledger for a same respective entity as the global distributed ledger.

19. The system of claim 18, further comprising:

a computing device in a global distributed ledger network maintaining a global distributed ledger including:

one or more processors;

a communication unit; and

a non-transitory computer-readable medium coupled to the one or more processors and the communication unit and storing instructions thereon, that when executed by the one or more processors, causes the computing device to:

for transactions generated during the threshold time period, add the transaction from each of the plurality of global distributed ledgers to a state block of transactions;

solve a cryptographic puzzle based on the state block of transactions;

add the solution to the cryptographic puzzle to the state block of transactions; and

transmit the state block of transactions to at least one other participant in a super blockchain network of participants maintaining a super blockchain.

20. The system of any of claims 12-19, wherein the local distributed ledger is a private blockchain viewable by an entity operating the process plant.

21. The system of any of claims 12-20, wherein the transaction includes a cryptographic hash value corresponding to the measurement.

22. The system of any of claims 17-18, wherein the shared resource between the plurality of entities operating the plurality of process plants is a fluid in a fluid pipeline, and the parameter measurement is an amount of fluid obtained by one of the plurality of entities from the fluid pipeline.

23. A validating network node in a process plant on a local distributed ledger network comprising:

a transceiver configured to (i) communicate with one or more field devices each performing a physical function to control an industrial process in the process plant and collecting measurements of parameters within the process plant, and to (ii) exchange local distributed ledger data with peer network nodes, the local distributed ledger data including transactions having parameter measurements;

a storage media configured to store a copy of the local distributed ledger; and

a process data validator configured to apply a set of consensus rules to the distributed ledger data received from the peer network nodes, the process data validator being further configured to append the distributed ledger data received from the peer network nodes to the copy of the distributed ledger if the distributed ledger data satisfies the consensus rules,

wherein after a threshold time period, the transceiver is configured to transmit a plurality of transactions generated during the threshold time period to at least one participant in a global distributed ledger network of participants maintaining a global distributed ledger, wherein the plurality of transactions includes the transactions having parameter measurements.

24. The validating network node of claim 23, wherein after the threshold time period, the validating network node is configured to prune at least some of the plurality of transactions generated during the threshold time period from the copy of the local distributed ledger.

25. The validating network node of claim 23 or 24, wherein the global distributed ledger is a permissioned blockchain viewable by a plurality of entities operating a plurality of process plants.