



(51) International Patent Classification:

G06F 1/26 (2006.01)

(21) International Application Number:

PCT/US2012/040141

(22) International Filing Date:

31 May 2012 (31.05.2012)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/491,377 31 May 2011 (31.05.2011) US

13/483,873 30 May 2012 (30.05.2012) US

(71) Applicant (for all designated States except US): CISCO

TECHNOLOGY, INC. [US/US]; 170 West Tasman Drive, San Jose, CA 95134 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): TAFT, Jeffrey, D.

[US/US]; 1852 Henderson Avenue, Washington, PA 15301 (US).

(74) Agents: BEHMKE, James, M. et al.; Edwards Wildman

Palmer LLP, P.O. Box 55874, Boston, MA 02205 (US).

(81) Designated States (unless otherwise indicated, for every

kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every

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

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: DISTRIBUTED INTELLIGENCE ARCHITECTURE WITH DYNAMIC REVERSE/FORWARD CLOUDING

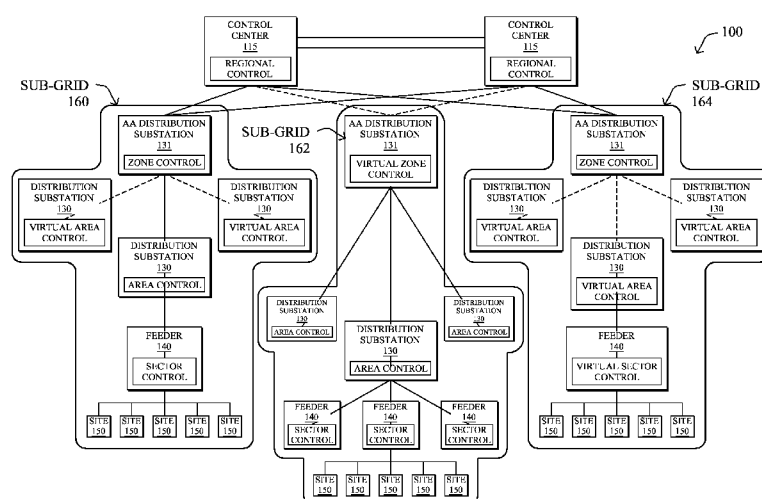


FIG. 15

(57) **Abstract:** In one embodiment, a grid application function is hosted at a primary grid device in a utility grid, which may determine whether or not to distribute at least a portion of the grid application function to one or more distributed secondary grid devices in the utility grid. In general, the one or more distributed secondary grid devices will be associated with a subset of the utility grid that is associated with the primary grid device (e.g., a sub-grid). Once a determination has been made, the primary grid device may dynamically distribute the grid application function, or a portion thereof, to the distributed secondary devices according to the determination. In addition, in another embodiment, the grid application function may also be withdrawn, such as from a particular secondary grid device or else from the primary grid device to an originating grid device.

DISTRIBUTED INTELLIGENCE ARCHITECTURE WITH DYNAMIC REVERSE/FORWARD CLOUDING

RELATED APPLICATIONS

The present application claims priority to U.S. Provisional Application No. 61/491,377, filed May 31, 2011, entitled VARIABLE TOPOLOGY DISTRIBUTED INTELLIGENCE FOR SMART GRIDS, by Jeffrey D. Taft, and U.S. Application No.: 13/483,873, filed May 30, 2012, entitled DISTRIBUTED INTELLIGENCE ARCHITECTURE WITH DYANAMIC REVERSE/FORWARD CLOUDING, by Jeffrey D. Taft, the contents of which are hereby incorporated by reference.

TECHNICAL FIELD

The present disclosure relates generally to utility control systems, e.g., to “smart grid” technologies.

BACKGROUND

Utility control systems and data processing systems have largely been centralized in nature. Energy Management Systems (EMS's), Distribution Management Systems (DMS's), and Supervisory Control and Data Acquisition (SCADA) systems reside in control or operations centers and rely upon what have generally been low complexity communications to field devices and systems. There are a few distributed control systems for utility applications, including a wireless mesh system for performing fault isolation using peer-to-peer communications among devices on feeder circuits outside of the substations. In addition, certain protection schemes involve substation-to-substation communication and local processing. In general however, centralized systems are the primary control architecture for electric grids.

Moreover, grid control operations have typically been carried out by grid control application functions implemented by a primary grid control device, which is inefficient.

BRIEF DESCRIPTION OF THE DRAWINGS

The embodiments herein may be better understood by referring to the following description in conjunction with the accompanying drawings in which like reference numerals indicate identically or functionally similar elements, of which:

- 5 FIG. 1 illustrates an example simplified utility grid hierarchy;
- FIG. 2 illustrates an example simplified communication network based on a utility grid (e.g., a “smart grid” network);
- FIG. 3 illustrates an example simplified device/node;
- FIG. 4 illustrates an example table showing challenges associated with complexity for
10 smart grids at scale;
- FIG. 5 illustrates an example of a smart grid core functions stack;
- FIG. 6 illustrates an example of various feedback arrangements;
- FIG. 7 illustrates an example chart showing a latency hierarchy;
- FIG. 8 illustrates an example table of data lifespan classes;
- 15 FIG. 9 illustrates an example of an analytics architecture;
- FIG. 10 illustrates an example of types of distributed analytic elements;
- FIG. 11 illustrates an example data store architecture;
- FIGS. 12A-12E illustrate an example layered services architecture model (“stack”);
- FIG. 13 illustrates an example logical stack for a distributed intelligence platform
- 20 FIGS. 14A-14D illustrate an example of a layered services platform;
- FIG. 15 illustrates an example of variable structure virtualization with forward and reverse clouding;
- FIG. 16 illustrates an example of a distributed platform operational environment; and
- FIG. 17 illustrates an example simplified procedure for a distributed intelligence
25 architecture with dynamic reverse and forward clouding.

DESCRIPTION OF EXAMPLE EMBODIMENTS

Overview

According to one or more embodiments of the disclosure, a grid application function is hosted at a primary grid device in a utility grid, which may determine whether or not to distribute at least a portion of the grid application function to one or more distributed secondary grid devices in the utility grid. In general, the one or more distributed secondary grid devices will be associated with a subset of the utility grid that is associated with the primary grid device (e.g., a sub-grid). Once a determination has been made, the primary grid device may dynamically distribute the grid application function, or a portion thereof, to the distributed secondary devices according to the determination. In addition, in one or more embodiments, the grid application function may also be withdrawn, such as from a particular secondary grid device or else from the primary grid device to an originating grid device.

Description

Electric power is generally transmitted from generation plants to end users (industries, corporations, homeowners, etc.) via a transmission and distribution grid consisting of a network of interconnected power stations, transmission circuits, distribution circuits, and substations. Once at the end users, electricity can be used to power any number of devices. Generally, various capabilities are needed to operate power grids at the transmission and distribution levels, such as protection, control (flow control, regulation, stabilization, synchronization), usage metering, asset monitoring and optimization, system performance and management, etc.

FIG. 1 illustrates an example simplified utility grid and an example physical hierarchy of electric power distribution. In particular, energy may be generated at one or more generation facilities 110 (e.g., coal plants, nuclear plants, hydro-electric plants, wind farms, etc.) and transmitted to one or more transmission substations 120. From the transmission substations 120, the energy is next propagated to distribution substations 130 to be distributed to various feeder circuits (e.g., transformers) 140. The feeders 140 may thus “feed” a variety of end-point “sites” 150, such as homes, buildings, factories, etc. over corresponding power-lines.

Note that the illustrative structure of the utility grid is shown as a highly simplified hierarchy, e.g., a hierarchy with generation at the top, transmission substations as the next tier, distribution substation as the next, etc. However, those skilled in the art will appreciate that FIG. 1 is merely an illustration for the sake of discussion, and actual utility grids may operate in a vastly more complicated manner (e.g., even in a vertically integrated utility). That is, FIG. 1 illustrates an example of power-based hierarchy (i.e., power starts at the generation level, and eventually reaches the end-sites), and not a logical control-based hierarchy. In particular, in conventional environments, transmission and primary distribution substations are at the same logical level, while generation is often its own tier and is really controlled via automatic generation control (AGC) by a Balancing Authority or other Qualified Scheduling Entity, whereas transmission lines and substations are under the control of a transmission operator Energy Management System (EMS). Primary distribution substations may be controlled by a transmission EMS in some cases and are controlled by a distribution control center, such as when distribution is via a Distribution System Operator (DSO). (Generally, distribution feeders do logically belong to primary distribution substations as shown.)

In the case of distributed control, that is, in terms of control-based hierarchy, substations may be grouped so that some are logically higher level than others. In this manner, the need to put fully duplicated capabilities into each substation may be avoided by allocating capabilities so as to impose a logical control hierarchy onto an otherwise flat architecture, such as according to the techniques described herein. In such cases, transmission substations may be grouped and layered, while primary distribution substations may be separately grouped and layered, but notably it is not necessary (or even possible) that distribution substations be logically grouped under transmission substations.

In general, utility companies can benefit from having accurate distribution feeder (medium voltage / low voltage or “MV/LV” circuit) connectivity information in their software applications and data stores. This is especially useful for outage management and for convenient application to planning, construction, operations, and maintenance. It is, however, very challenging to try to construct or approximate the circuit model within a geographic information systems (GIS) environment due to the complexity of modeling the dynamic nature of an electrical network. That is, while the utility may have an “as-built” database, it may differ from

the actual grid for various reasons, including inaccurate or incomplete data capture on grid construction, changes to circuits that are not reflected in updates to the database, and structural damage to the grid. In addition, circuit topology may change dynamically as feeder switches are operated in the course of either normal or emergency operations. Such changes result in an “as-operated” topology that is dynamic and is not reflected in the “as-built” database.

To assist in control of the utility grid, various measurement and control devices may be used at different locations within the grid 100. Such devices may comprise various energy-directing devices, such as reclosers, power switches, circuit breakers, etc. In addition, other types of devices, such as sensors (voltage sensors, current sensors, temperature sensors, etc.) or computational devices, may also be used. Electric utilities use alternating-current (AC) power systems extensively in generation, transmission, and distribution. Most of the systems and devices at the high and medium voltage levels operate on three-phase power, where voltages and currents are grouped in threes, with the waveforms staggered evenly. The basic mathematical object that describes an AC power system waveform (current or voltage) is the “phasor” (phase angle vector). Computational devices known as Phasor Measurement Units (PMUs) have thus been commercialized by several companies to calculate phasors from power waveforms. Because phase angle is a relative quantity, it is necessary when combining phasors taken from different parts of a power grid to align the phase angle elements to a common phase reference; this has been typically done in PMUs through the use of GPS timing signals. Such phasors are known as synchrophasors.

FIG. 2 is a schematic block diagram of a communication network 200 that may illustratively be considered as an example utility grid communication network. The network 200 illustratively comprises nodes/devices interconnected by various methods of communication, such as wired links or shared media (e.g., wireless links, Power-line communication (PLC) links, etc.), where certain devices, such as, e.g., routers, sensors, computers, etc., may be in communication with other devices, e.g., based on distance, signal strength, current operational status, location, etc. Those skilled in the art will understand that any number of nodes, devices, links, etc. may be used in the computer network, and that the view shown herein is for simplicity. Data packets may be exchanged among the nodes/devices of the computer network 200 using predefined network communication protocols such as certain known wired protocols, wireless

protocols (e.g., IEEE Std. 802.15.4, WiFi, Bluetooth®, DNP3 (distributed network protocol), Modbus, IEC 61850, etc.), PLC protocols, or other protocols where appropriate. In this context, a protocol consists of a set of rules defining how the nodes interact with each other.

Illustratively, a control center 210 (and backup control center 210a) may comprise various control system processes 215 and databases 217 interconnected via a network switch 219 to a system control network 205. Additionally, one or more substations 220 may be connected to the control network 205 via switches 229, and may support various services/process, such as a distributed data service 222, grid state service (e.g., “parstate”, a determination of part of the whole grid state) 223, control applications 225, etc. The substations 220 may also have a GPS clock 221 to provide timing, which may be distributed to the FARs 250 (below) using IEEE Std. 1588. Note that a monitoring center 230 may also be in communication with the network 205 via a switch 239, and may comprise various analytics systems 235 and databases 237. The substations 220 may communicate with various other substations (e.g., from transmission substations to distribution substations, as mentioned above) through various methods of communication. For instance, a hierarchy of wireless LAN controllers (WLCs) 240 and field area routers (FARs) 250 may provide for specific locality-based communication between various portions of the underlying utility grid 100 in FIG. 1. WLCs 240 (which may also be considered as a type of higher grid level FAR) may comprise various services, such as data collection 245, control applications 246, etc. Generally, grid devices on shared feeder sections (e.g., FAR 250-X) may communicate with both involved substations (e.g., both WLCs 240, as shown). Further, FARs 250 may also comprise data collection services 255 themselves, and may collect data from (or distribute data to) one or more end-point communication devices 260, such as sensors and/or actuators (e.g., home energy controllers, grid controllers, etc.).

Specific details of the operation of the smart grid devices are described below. Note that while there is a general correlation between the communication network 200 and underlying utility grid 100 (e.g., control centers, substations, end-points, etc.), such a correlation may only be generally assumed, and is not a necessity. For instance, FARs 250 may be associated with feeder circuits 140, or may be more granular such as, e.g., “pole-top” routers. In other words, the hierarchies shown in FIG. 1 and FIG. 2 are not meant to be specifically correlated, and are merely examples of hierarchies for illustration.

FIG. 3 is a schematic block diagram of an example node/device 300 that may be used with one or more embodiments described herein, e.g., as any capable “smart grid” node shown in FIG. 2 above. In particular, the device 300 is a generic and simplified device, and may comprise one or more network interfaces 310 (e.g., wired, wireless, PLC, etc.), at least one processor 320, and a memory 340 interconnected by a system bus 350, as well as a power supply 360 (e.g., battery, plug-in, etc.).

The network interface(s) 310 contain the mechanical, electrical, and signaling circuitry for communicating data over links coupled to the network 200. The network interfaces may be configured to transmit and/or receive data using a variety of different communication protocols. Note, further, that the nodes may have two different types of network connections 310, e.g., wireless and wired/physical connections, and that the view herein is merely for illustration. Also, while the network interface 310 is shown separately from power supply 360, for PLC the network interface 310 may communicate through the power supply 360, or may be an integral component of the power supply. In some specific configurations the PLC signal may be coupled to the power line feeding into the power supply.

The memory 340 of the generic device 300 comprises a plurality of storage locations that are addressable by the processor 320 and the network interfaces 310 for storing software programs and data structures associated with the embodiments described herein. Note that certain devices may have limited memory or no memory (e.g., no memory for storage other than for programs/processes operating on the device and associated caches). The processor 320 may comprise necessary elements or logic adapted to execute the software programs and manipulate the data structures 345. An operating system 342, portions of which are typically resident in memory 340 and executed by the processor, functionally organizes the device by, *inter alia*, invoking operations in support of software processes and/or services executing on the device. These software processes and/or services may comprise one or more grid-specific application processes 348, as described herein. Note that while the grid-specific application process 348 is shown in centralized memory 340, alternative embodiments provide for the process to be specifically operated within the network elements or network-integrated computing elements 310.

It will be apparent to those skilled in the art that other processor and memory types, including various computer-readable media, may be used to store and execute program instructions pertaining to the techniques described herein. Also, while the description illustrates various processes, it is expressly contemplated that various processes may be embodied as modules configured to operate in accordance with the techniques herein (e.g., according to the functionality of a similar process). Further, while the processes have been shown separately, those skilled in the art will appreciate that processes may be routines or modules within other processes.

As noted above, utility control systems and data processing systems have largely been centralized in nature. Energy Management Systems (EMS's), Distribution Management Systems (DMS's), and Supervisory Control and Data Acquisition (SCADA) systems reside in control or operations centers and rely upon what have generally been low complexity communications to field devices and systems. Both utilities and makers of various grid control systems have recognized the value of distributed intelligence, especially at the distribution level.

Generally, distributed intelligence is defined as the embedding of digital processing and communications ability in a physically dispersed, multi-element environment (specifically the power grid infrastructure, but also physical networks in general). In the area of sensing, measurement and data acquisition, key issues are:

- Sensing and measurement – determination of quantities to be sensed, type and location of sensors, and resulting signal characteristics;
- Data acquisition – collection of sensor data, sensor data transport;
- System state and observability – key concepts that can be used to guide the design of sensor systems for physical systems with topological structure and system dynamics; and
- Sensor network architecture – elements, structure, and external properties of sensor networks.

Key elements of distributed intelligence comprise:

- Distributed data collection and persistence – measurement of electrical grid state, power quality, asset stress and utilization factors, environmental data, real-time grid topology, and device operating states, as opposed to central SCADA;
- Distributed data transformation and analytics – processing of measured data and event messages generated by smart grid devices and systems to extract useful information, prepare data for use by applications, or to correlate and filter data and events for aggregation purposes, as opposed to data center processing; and
- Distributed control – execution of actual control algorithms, with control commands being sent directly to grid control actuators for relatively local controllers, as opposed to central control.

By establishing the network as a platform (NaaP) to support distributed applications, and understanding the key issues around sensing and measurement for dynamic physical network systems, key capabilities of smart communication networks may be defined (e.g., as described below) that support current and future grid applications. In particular, as ICT (Information Communication Technology) networks converge with physical power grids and as “smart” functions penetrate the grid, centralized architectures for measurement and control become increasingly inadequate. Distribution of intelligence beyond the control center to locations in the power grid provides the opportunity to improve performance and increase robustness of the data management and control systems by addressing the need for low latency data paths and supporting various features, such as data aggregation and control federation and disaggregation.

In particular, there are a number of compelling arguments for using distributed intelligence in smart power grids, and in large scale systems in general, such as:

- Low Latency Response – A distributed intelligence architecture can provide the ability to process data and provide it to the end device without a round trip back to a control center;
- Low Sample Time Skew – Multiple data collection agents can easily minimize first-to-last sample time skew for better system state snapshots;
- Scalability – No single choke point for data acquisition or processing; analytics at the lower levels of a hierarchical distributed system can be processed and passed on to

higher levels in the hierarchy. Such an arrangement can keep the data volumes at each level roughly constant by transforming large volumes of low level data into smaller volumes of data containing the relevant information. This also helps with managing the bursty asynchronous event message data that smart grids can generate (example: last gasp messages from meters during a feeder momentary outage or sag). The scalability issue is not simply one of communication bottlenecking however – it is also (and perhaps more importantly) an issue of data persistence management, and a matter of processing capacity. Systems that use a central SCADA for data collection become both memory-bound and CPU-bound in a full scale smart grid environment, as do other data collection engines; and

- Robustness – Local autonomous operation, continued operation in the presence of fragmentation of the network, graceful system performance and functional degradation in the face of failures, etc.

Standard approaches to distributed processing suffer from shortcomings relative to the electric grid environment. These shortcomings include inability to handle incremental rollout, variable distribution of intelligence, and applications not designed for a distributed (or scalable) environment. Further, existing approaches do not reflect the structure inherent in power grids and do not provide integration across the entire set of places in the grid where intelligence is located, or across heterogeneous computing platforms. Current systems also suffer from inability to work with legacy software, thus requiring massive software development efforts at the application level to make applications fit the platform, and also lack zero-touch deployment capability and requisite security measures.

For instance, one major obstacle in the adoption of distributed intelligence, now that IP communications and embedded processing capabilities are becoming available in forms that utilities can use, is that utilities cannot make large equipment and system changes in large discrete steps. Rather they must go through transitions that can take years to complete. This is due to the nature of their mission and the financial realities utilities must deal with. In practice, utilities must be able to transition from centralized to distributed intelligence, and must be able to operate in a complicated hybrid mode for long periods of time, perhaps permanently. This means that the utility must be able to roll out distributed intelligence incrementally while

maintain full operations over the entire service area, and must be able to modify the distributed architecture appropriately over time and geography. Simply having a distributed architecture implementation is not sufficient; it must be easily and continually mutable in terms of what functionality is distributed to which processing locations in the grid and must be capable of coexisting with legacy control systems where they remain in place. Therefore, there exist various kinds of variable topology for effective distributed intelligence:

- Transition Variability – Rollout of distributed intelligence functions will be uneven both geographically (topologically) and over time, and there is no one-size-fits-all solution, even for a single utility;
- End State Variability – Not every distributed intelligence function will be pushed to every end node of the same class, and distributed intelligence functions and distributions will have to change over the life of the system;
- Operational Variability – Users must be able to change locations of functions to deal with failures and maintenance, etc.

Additionally, design and implementation of smart grids at scale poses a number of challenging architecture issues. Many of these issues are not apparent or do not show significant effects at pilot scale, but can become crucial at full scale. Note that generally herein, “at full scale” means one or more of:

- Endpoint scale – the number of intelligent endpoints is in the millions per distribution grid;
- Functional complexity scale – the number and type of functions or applications that exhibit hidden layer coupling through the grid is three or more; or the number of control systems (excluding protection relays) acting on the same feeder section or transmission line is three or more; and
- Geospatial complexity – the geographical/geospatial complexity of the smart grid infrastructure passes beyond a handful of substation service areas or a simple metro area deployment to large area deployments, perhaps with interpenetrated service areas for different utilities, or infrastructure that cuts across or is shared across multiple utilities and related organizations.

In the table 400 shown in FIG. 4, some of the challenges arising from these levels of complexity for smart grids at scale are illustrated. For instance, ultra large scale (ULS) characteristics of smart grids at scale are usually associated with decentralized control, inherently conflicting diverse requirements, continuous evolution and deployment, heterogeneous, inconsistent, and changing elements, and various normal failure conditions. Also, hidden couplings via the grid exist, since systems and controls are inherently coupled through grid electrical physics and therefore interact in ways that may be unaccounted for in system designs. The grid may further be viewed at scale as a multi-objective, multi-control system, where multiple controls affecting the same grid portions, and where some of the controls actually lie outside of the utility and/or are operating on multiple time scales. Moreover, bulk or aggregate control commands, especially as regards secondary load control and stabilization, may not consider the specific localities within the grid, and are not broken down to the feeder or even section level, taking into account grid state at the level. Lastly, smart grid-generated data must be used on any of a number of latency scales, some of which are quite short, thus precluding purely centralized processing and control approaches. Note that there are additional issues affecting architecture for smart grids at scale than those that are shown in FIG. 4, but these are representative of some of the key challenges.

The smart grid has certain key attributes that lead to the concept of core function classes supported by the smart grid. These key attributes include:

- A geographically distributed analog infrastructure;
- A digital superstructure consisting of digital processing layered on top of the analog superstructure, along with ubiquitous IP-based digital connectivity; and
- Embedded processors and more general smart devices connected to the edges of the smart grid digital superstructure and the analog infrastructure; these include both measurement (sensor) and control (actuator) devices.

Given this environment, and given our present understanding of the nature of the desired behavior of the power grid, we may identify a number of key function classes; functional groups that arise inherently from the combination of desired smart grid behavior, grid structure, and the nature of the digital superstructure applied to the grid. An understanding of these core function

groups is key to developing a view toward a layered network services architecture for smart grids. A model is presented herein in which smart grid applications of any type are built upon a logical platform of core function classes that arise from the grid itself.

FIG. 5 illustrates the concept and the function classes themselves. For instance, to support distributed intelligence for electric power grids or any physical network, the concept of network services may be extended to become a stack of service groups, where the services become increasingly domain-oriented as one moves up the stack. This means that the lower layer contains ordinary network services. The next layer contains services that support distributed intelligence. The third layer provides services that support domain specific core functions. The top layer provides services that support application integration for real-time systems.

Specifically, as shown in the model of FIG 5, the function classes are divided into four tiers.

1) The base tier 510 is:

- Power Delivery Chain Unification: use of digital communications to manage secure data flows and to integrate virtualized information services at low latency throughout the smart grid; enable N-way (not just two-way) flow of smart grid information; provision of integration through advanced networking protocols, converged networking, and service insertion. Note that this layer is based on advanced networking and communication, and in general may be thought of as system unification. In this model, networking plays a foundational role; this is a direct consequence of the distributed nature of smart grid assets.

2) The second tier 520 is:

- Automatic Low Level Control 521 – digital protection inside and outside the substation, remote sectionalizing and automatic reclosure, feeder level flow control, local automatic voltage/VAr regulation, stabilization, and synchronization; and
- Remote Measurement 522 – monitoring and measurement of grid parameters and physical variables, including direct power variables, derived element such as power

quality measures, usage (metering), asset condition, as-operated topology, and all data necessary to support higher level function classes and applications.

3) The third tier 530 is:

- Control Disaggregation 531 – control commands that are calculated at high levels must be broken down into multiple commands that align with the conditions and requirements at each level in the power delivery chain; the process to accomplish this is the logical inverse of data aggregation moving up the power delivery chain, and must use knowledge of grid topology and grid conditions to accomplish the disaggregation; and
- Grid State Determination 532 – electrical measurement, power state estimation, and visualization, voltage and current phasors, bus and generator phase angles, stability margin, real and reactive power flows, grid device positions/conditions, DR/DSM available capacity and actual response measurement, storage device charge levels, circuit connectivity and device parametrics.

4) The fourth tier 540 is:

- Fault Intelligence 541 - detection of short or open circuits and device failures; fault and failure classification, characterization (fault parameters), fault location determination, support for outage intelligence, support for adaptive protection and fault isolation, fault prediction, fault information notification and logging;
- Operational Intelligence 542 – all aspects of information related to grid operations, including system performance and operational effectiveness, as well as states of processes such as outage management or fault isolation;
- Outage Intelligence 543 – detection of service point loss of voltage, inside/outside trouble determination, filtering and logging of momentaries, extent mapping and outage verification, root cause determination, restoration tracking and verification, nested root cause discovery, outage state and process visualization, crew dispatch support;
- Asset Intelligence 544 – this has two parts:

- asset utilization intelligence - asset loading vs. rating, peak load measurement (amplitude, frequency), actual demand curve measurement, load/power flow balance measurement, dynamic (real-time) de-rating/re-rating, real-time asset profitability/loss calculation; and
- 5 ○ asset health/accumulated stress intelligence - device health condition determination, online device and system failure diagnostics, device failure or imminent failure notification, asset accumulated stress measurement, Loss of Life (LoL) calculation, Estimated Time to Failure (ETTF) prediction, Asset Failure System Risk (AFSR) calculation; and
- 10 - Control Federation 545 – grid control increasingly involves multiple control objectives, possible implemented via separate control systems. It is evolving into a multi-controller, multi-objective system where many of the control systems want to operate the same actuators. A core function of the smart grid is to *federate* these control systems that include Demand Response and DSM, voltage regulation,
- 15 capacitor control, power flow control, Conservation Voltage Reduction (CVR), Electric Vehicle Charging Control, Line Loss Control, Load Balance Control, DSTATCOM and DER inverter VAr control, reliability event control, Virtual Power Plant (VPP) control, and meter connect/disconnect and usage restriction control.

These function classes may support one or more smart grid applications 550. In general, therefore, smart grid networks, that is, the combination of a utility grid with a communication network, along with distributed intelligent devices, may thus consist of various type of control, data acquisition (e.g., sensing and measurement), and distributed analytics, and may be interconnected through a system of distributed data persistence. Examples may include, among others, distributed SCADA data collection and aggregation, grid state determination and

25 promulgation, implementation of distributed analytics on grid data, control command delivery and operational verification, control function federation (merging of multiple objective/multiple control systems so that common control elements are used in non-conflicting ways), processing of events streams from grid devices to filter, prevent flooding, and to detect and classify events for low latency responses, and providing virtualization of legacy grid devices so that they are

30 compatible with modern approaches to device operation and network security.

In particular, there may be a number of types of control, such as sequence control (e.g., both stateless and stateful, typified by switching systems of various kinds), stabilizers (e.g., which moderate dynamic system behavior, typically through output or state feedback so that the system tends to return to equilibrium after a disturbance), and regulators (e.g., in which a system is made to follow the dynamics of a reference input, which may be dynamic or static set points). Quite often, all three of these are present in the same control system. In terms of electric power grids, flow control is sequence control, whereas model power oscillation damping and volt/VAr control represent stabilization and regulatory control, respectively.

For most control systems, feedback is a crucial component. FIG. 6 illustrates output feedback 610 and state feedback 620, both of which are quite common. FIG. 6 also illustrates a slightly more complex feedback arrangement 630 intended to be used when a system exhibits two very different sets of dynamics, one fast and one slow. There are a great many extensions of the basic control loop and the volume of mathematics, theory, and practice is enormous and widely used.

Regarding data acquisition, sensing and measurement support multiple purposes in the smart grid environment, which applies equally as well to many other systems characterized by either geographic dispersal, or large numbers of ends points, especially when some form of control is required. Consequently, the sensing system design can be quite complex, involving issues physical parameter selection, sensor mix and placement optimization, measurement type and sample rate, data conversion, sensor calibration, and compensation for non-ideal sensor characteristics.

Additionally, collection of the data in large scale systems such as smart grids presents issues of cycle time, data bursting, and sample skew. There are multiple modes of data collection for large scale systems and each presents complexities, especially when the system model involves transporting the data to a central location. In the typical round-robin scanning approach taken by many standard SCADA systems, the time skew between first and last samples represents an issue for control systems that is insignificant when the scan cycle time is short compared to system dynamics, but as dynamics increase in bandwidth with advanced regulation and stabilization, and as the number of sensing points increases, the sample time skew problem becomes significant.

Data is consumed in a variety of ways and places in a power grid; most of these are not located at the enterprise data center and much grid data does not enter the data center. Some of it does not even enter the control/operations center, as it must be consumed “on the fly” in grid devices and systems. Consequently it is important to classify data according to the latency requirements of the devices, systems, or applications that use it and appropriate persistence (or lack thereof) must also be defined. Note that much grid data has multiple uses; in fact, it is an element of synergy that has significant impact on smart grid economics and system design (networking, data architecture, analytics) to ensure that data is used to support as many outcomes as possible.

FIG. 7 is a chart 700 that illustrates the issue of latency, as latency hierarchy is a key concept in the design of both data management and analytics applications for physical networks with control systems or other real-time applications. In particular, in the example (and non-limiting) chart 700, grid sensors and devices are associated with a very low latency, where high-speed/low-latency real-time analytics may require millisecond to sub-second latency to provide results through a machine-to-machine (M2M) interface for various protection and control systems. The latency hierarchy continues toward higher latency associations as shown and described in chart 700, until reaching a very high latency at the business data repository level, where data within days to months may be used for business intelligence processing, and transmitted via a human-machine interface (HMI) for various reporting, dashboards, key performance indicators (KPI's), etc. Note that the chart does not illustrate that a given data element may in fact have multiple latency requirements, depending on the various ways it may be used, meaning that any particular datum may have multiple destinations.

The latency hierarchy issue is directly connected to the issue of lifespan classes, meaning that depending on how the data is to be used, there are various classes of storage that may have to be applied. This typically results in hierarchical data storage architecture, with different types of storage being applied at different points in the grid that correspond to the data sources and sinks, coupled with latency requirements.

FIG. 8 illustrates a table 800 listing some types of data lifespan classes that are relevant to smart grid devices and systems. In particular, transit data exists for only the time necessary to travel from source to sink and be used; it persists only momentarily in the network and the data

sink and is then discarded. Examples are an event message used by protection relays, and sensor data used in closed loop controls; persistence time may be microseconds. On the other hand, burst/flow data, which is data that is produced or processed in bursts, may exist temporarily in FIFO (first in first out) queues or circular buffers until it is consumed or overwritten. Examples of burst/flow data include telemetry data and asynchronous event messages (assuming they are not logged), and often the storage for these types of data are incorporated directly into applications, e.g., CEP engine event buffers. Operational data comprises data that may be used from moment to moment but is continually updated with refreshed values so that old values are overwritten since only present (fresh) values are needed. Examples of operational data comprise grid (power) state data such as SCADA data that may be updated every few seconds. Transactional data exists for an extended but not indefinite time, and is typically used in transaction processing and business intelligence applications. Storage of transactional data may be in databases incorporated into applications or in data warehouses, datamarts or business data repositories. Lastly, archival data is data that must be saved for very long (even indefinite) time periods, and typically includes meter usage data (e.g., seven years), PMU data at ISO/RTO's (several years), log files, etc. Note that some data may be retained in multiple copies; for example, ISO's must retain PMU data in quadruplicate. Just as with latency hierarchy, grid data may progress through various lifetime classes as it is used in different ways. This implies that some data will migrate from one type of data storage to another as its lifetime class changes, based on how it is used.

Distributed analytics may be implemented in a fully centralized manner, such as usually done with Business Intelligence tools, which operate on a very large business data repository. However, for real-time systems, a more distributed approach may be useful in avoiding the inevitable bottlenecking. A tool that is particularly suited to processing two classes of smart grid data (streaming telemetry and asynchronous event messages) is Complex Event Processing (CEP) which has lately also been called streaming database processing. CEP and its single stream predecessor Event Stream Processing (ESP) can be arranged into a hierarchical distributed processing architecture that efficiently reduces data volumes while preserving essential information embodied in multiple data streams.

FIG. 9 shows an example of such analytics architecture. In this case, the analytics process line sensor data and meter events for fault and outage intelligence. In particular, various line sensors 905 may transmit their data via ESPs 910, and may be collected by a feeder CEP 915 at a substation 920. Substation CEPs 925 aggregate the feeder CEP data, as well as any data from substation devices 930, and this data may be relayed to a control center CEP 935 within a control center 940. Along with meter events from meter DCE 945 and other data from database 950, the control center CEP 935 may thus perform a higher level of analytics than any of the below levels of CEPs, accordingly.

In general, distributed analytics can be decomposed into a limited set of analytic computing elements (“DA” elements), with logical connections to other such elements. Full distributed analytics can be constructed by composing or interconnecting basic analytic elements as needed. Five basic types of distributed analytic elements are defined herein, and illustrated in FIG. 10:

1. Local loop 1010 – an analytic element operates on data reports its final result to a consuming application such as a low latency control;
2. Upload 1020 – an analytic element operates on data and then reports out its final result;
3. Hierarchical 1030 – two or more analytic elements operate on data to produce partial analytics results which are then fused by a higher level analytics element, which reports the result;
4. Peer to peer 1040 – two or more analytics elements operate on data to create partial results; they then exchange partial results to compute final result and each one reports its unique final analytic; and
5. Database access 1050 – an analytic element retrieves data from a data store in addition to local data; it operates on both to produce a result which can be stored in the data store or reported to an application or another analytic element

A sixth type, “generic DA node” 1060, may thus be constructed to represent each of the five basic types above.

Given the above-described concept of distributed analytics, including the database access element 1050 shown in FIG. 10, it becomes useful to consider distributed data persistence as an architectural element. Low level and low latency analytics for smart grids (mostly related to control) require state information and while local state components are generally always needed, it is often the case that elements of global state are also necessary. Operational data (essentially extended system state) may be persisted in a distributed operational data store. The reason for considering a true distributed data store is for scalability and robustness in the face of potential network fragmentation. In power systems, it is already common practice to implement distributed time series (historian) databases at the control center and primary substation levels. The techniques described herein may incorporate this and the distributed operational data store into an integrated data architecture by employing data federation in conjunction with various data stores.

FIG. 11 illustrates a data store architecture 1100 that federates distributed and centralized elements in order to support a wide range of analytics, controls, and decision support for business processes. In particular, a control center 1110 may comprise various centralized repositories or databases, such as a waveform repository 1112, an operational (Ops) data database 1114, and a time series database 1116. For instance, common interface model (CIM) services 1118 within the control center 1110 may operate based on such underlying data, as may be appreciated in the art. The data itself may be federated (e.g., by data federation process 1119) from various transmission substation databases 1120, primary distribution substation databases 1130, secondary substation databases 1140, distribution feeder (or other distributed intelligence point) database 1150. Typically, edge devices (end-points, sites, etc.) need not have further database or storage capabilities, but may depending upon various factors and considerations of a given implementation.

Notably, the architecture herein may build upon the core function groups concept above to extend grid capabilities to the control center and enterprise data center levels, using the layer model to unify elements and approaches that have typically been designed and operated as if they were separate and unrelated. This model may also be extended to provide services related to application integration, as well as distributed processing. This yields a four tier model, wherein each tier is composed of multiple services layers. The four tiers are as follows (from the

bottom of the stack upward), where each of the layers and tiers is intended to build upon those below them:

1. Network services;
2. Distributed Intelligence services;
- 5 3. Smart Grid Core Function services; and
4. Application Integration services.

FIGS. 12A-12E illustrates the Layered Services Architecture model (“stack”) 1200. In particular, FIG. 12A shows a full stack model for the layered services. Application Integration Services 1210 comprises services that facilitate the connection of applications to data sources and each other. Note that at this top layer the stack splits into two parallel parts as shown in FIG. 12B: one for enterprise level integration 1212 and one for integration at the real-time operations level 1214. For the enterprise level, there are many available solutions, and the use of enterprise service buses and related middleware in a Service Oriented Architecture (SOA) environment is common. For the real-time operations side, the architecture herein relies less on such
15 middleware tools and much more on network services. This is for two reasons: network-based application integration can perform with much lower latencies than middleware methods, and the use of middleware in a control center environment introduces a layer of cost and support complexity that is not desirable, given that the nature of integration at the real-time operations level does not require the more general file transfer and service composition capabilities of the
20 enterprise SOA environment. The enterprise side of the application integration layer is not actually part of the distributed intelligence (DI) platform; it is shown for completeness and to recognize that interface to this form of integration environment may be needed as part of a fully integrated computing platform framework.

Additionally, the Smart Grid Core Function Services layer 1220 (detailed in FIG. 12C)
25 generally comprises the components listed above in FIG. 5, namely services that derive from or are required by the capabilities of the smart grid superstructure. Moreover, the Distributed Intelligence Services layer 1230 (FIG. 12D) comprises support for data processing and data management over multiple, geographically dispersed, networked processors, some of which are embedded. Lastly, Network Services layer 1240 (FIG. 12E) comprises IP-based data transport

services for grid devices, processing systems, and applications. Note that CEP is illustratively included here because it is fundamental to network management in the core grid architecture model.

Another way of approaching the layered services stack as shown in FIGS. 12A-12E above is from the perspective of the devices themselves, particularly as a logical stack. For instance, a logical stack 1300 for the distributed intelligence platform is illustrated in FIG. 13. Note that not all parts of this stack 1300 are intended to be present in every processing node in a system. FIG. 13 is correlated with the layered services stack 1200 of FIGS. 12A-12E, but the logical stack 1300 also shows placement of two types of data stores (historian 1365 to store a time series of data, thus maintaining a collection of (e.g., all of) the past values and database 1336 to store generally only the most recent (e.g., periodically refreshed) values of a set of operational variables), as well as an API layer 1340 to expose certain capabilities of the platform to the applications and to upper levels of the platform stack. Generally, at the base of the stack 1300 is the known IPv4/v6 protocol stack 1310, above which are grid protocols 1320 and peer-to-peer (P2P) messaging protocols 1325. Further up the stack 1300 are standard network services 1332, embedded CEP engines 1334, and the distributed database 1336. Through the API layer 1340, the stack 1300 reaches distributed intelligence services 1350 and unified computing / hypervisor(s) 1355, upon which rest grid-specific network services 1360 and historians 1365. Application integration services/tools 1370 tops the stack 1300, allowing for one or more applications 1380 to communicate with the grid devices, accordingly.

Based on the description above, a layered services platform may be created, which is a distributed architecture upon which the layered services and smart grid applications may run. The distributed application architecture makes use of various locations in the grid, such as, e.g., field area network routers and secondary substation routers, primary substations, control centers and monitoring centers, and enterprise data centers. Note that this architecture can be extended to edge devices, including devices that are not part of the utility infrastructure, such as building and home energy management platforms, electric vehicles and chargers, etc.

FIGS. 14A-14D illustrate an example of the layered services platform described above. For instance, as detailed in FIGS. 14A-14D, enterprise data centers 1410 may comprise various business intelligence (BI) tools, applications (enterprise resource planning or “ERP,” customer

information systems or “CIS,” etc.), and repositories based on a unified computing system (UCS). Other systems, such as meter data management systems (MDMS) may also be present. Via a utility tier network 1420, the enterprise data centers 1410 may be in communicative relationship with one or more utility control centers 1430, which comprise head-end control and other systems, in addition to various visualization tools, control interfaces, applications, databases, etc. Illustratively, a services-ready engine (SRE), application extension platform (AXP), or UCS may structurally organize the utility control centers 1420. Through a system control tier network 1440, one or more primary substations 1450 may be reached by the control centers 1430, where a grid connected router (GCR) interconnects various services (apps, databases, etc.) through local device interfaces. Utility FANs (field area networks) 1460 (or neighborhood area networks (NAN’s)) may then bridge the gap to pole top FARs 1470, or else (e.g., in Europe) secondary distribution substations 1480 to reach various prosumer (professional consumer) assets 1490, accordingly.

Distributed Intelligence Architecture

As noted above, grid control operations have typically been carried out by grid control application functions implemented by a primary grid control device. As shown in FIG.1, in an exemplary utility grid 100 grid control applications are typically managed by a central device (e.g., control center 115), which controls the various transmission substations 120, distribution substations 130, feeder stations 140, etc., present in the associated utility grid 100, or sub-grids. This traditional approach is inefficient, inflexible, and non-adaptive. The architecture disclosed herein moves away from such centralization to use “forward clouding” and “reverse clouding” as part of a dynamic, distributed intelligence architecture for grid control applications. For example, in conventional virtualization (i.e., “forward cloud” or “forward clouding”), applications are run on various distributed desktops, and their data/output are collected in a centralized location called the “cloud” (e.g., a remote data center), which provides the virtualization environment to run the applications remotely. However, in the case of a distributed intelligence architecture for utility grids, this virtualization process is inverted to create a reverse cloud approach: a centralized grid application is parsed out into the utility grid, or sub-grid, and run on a distributed set of computing resources (e.g., primary, secondary, tertiary, etc. grid devices, but typically not desktop computers). Reverse clouding, and its

dynamic allocation between reverse and forward clouding, may be particularly referenced to grid operation, but it should be noted that this is not a necessity.

FIG. 15 shows an example of such a distributed intelligence architecture in a reverse and forward cloud environment, as applied to a simplified control model for a utility grid 1500.

5 Utility grid 1500 may comprise multiple control levels that may be logically organized in a hierarchy, which is maintained in utility grid 1500 as a whole, as well as throughout sub-grids such as, for example, sub-grid 1560, sub-grid 1562, and sub-grid 1564. Typically, a control center(s) 1515 with regional control would occupy a high level in the utility grid, or the highest level in a sub-grid, and multiple grid control centers 1515 may reside within utility grid 1500.

10 Grid control centers 1515 oversee the various levels of grid devices below them such as, for example, “AA” distribution substations 131 (as will be understood in the art) with zone control, distribution substations 130 with area control, feeder stations 140 with sector control, and so on to the site level 150 (e.g., the consumer/prosumer use level). (Note that though the illustration in FIG. 15 relates generally to distribution components, the techniques herein are not so limited,

15 and may apply to any grid components that may be logically structured in a general hierarchy.)

Specifically, according to one or more embodiments of the disclosure as described in detail below, the techniques herein treat grid control operations as grid application functions that may be dynamically distributed throughout a utility grid according to need (or more generally, any determination), effectively allowing the presence of one or more grid application functions

20 to expand and contract throughout a utility grid, or sub-grid, as desired. For example, as shown in FIG. 15, a grid application function may be hosted at a primary grid device in a utility grid, which may determine whether or not to distribute at least a portion of the grid application function to one or more distributed secondary grid devices in the utility grid. Typically, the primary grid device may be a control center 1515, however, any capable grid device may also

25 serve this function (e.g., an AA distribution substation 131, a distribution substation 130, a feeder station 140, etc.). Typically, such distributed secondary grid devices will be associated with a subset of the utility grid that is associated with the primary grid device (e.g., a sub-grid 1560, a sub-grid 1562, or a sub-grid 1564). Once a determination has been made, the primary grid device may dynamically distribute the grid application function, or a portion thereof, to the

distributed secondary devices according to the determination by reverse cloud, or alternatively, pull the distributed grid application function back to the primary grid device by forward cloud.

Illustratively, FIG. 15 depicts this dynamic distribution process as solid and dashed lines connecting grid devices at various levels of the grid hierarchy. A solid line indicates reverse clouding, meaning that the application has been pushed from a centralized location (i.e., a primary grid device), or a higher tier or grid level, to a decentralized lower tier or grid level. In contrast, a dashed line indicates forward clouding, meaning that an application has been pulled back (i.e., forward cloud virtualized back) to a higher tier or grid level from a lower level. This approach is highly adaptive. For example, control centers 1515 may push grid application functions down to AA distribution substations 131 in sub-grids 1560 and 1564, while simultaneously pulling distributed grid application functions back from AA distribution substation 131 in sub-grid 1562. It should be noted that this approach allows the possibility of combinatorial control. For example, AA distribution substation 131 in sub-grid 1560 may also function as a primary grid device in the context of sub-grid 1560. Accordingly, in addition to “receiving” grid application functions from control center 1515 by reverse cloud, it may also push the same, or different, grid application functions to lower tiered devices in sub-grid 1560 such as, for example, a distribution sub-station 130. According to an exemplary embodiment, the tier or grid level at which the application is virtualized may be controlled by a central management tool.

This method is not limited to the station or sub-station level within a utility grid. Quite the contrary, it is contemplated that this method may be extended to platforms beyond the sub-station level, such as, for example, Field Area Network devices and non-utility grid-connected devices. It is to be understood that the principle of forward and reverse cloud in application virtualization is general, and is not limited to use with electric power grids. This approach increases grid control efficiency and flexibility by enabling grid application functions to be distributed to sub-grid regions on an as-needed basis.

Illustratively, with reference again to FIG. 3, the techniques described herein may be performed by hardware, software, and/or firmware, such as in accordance with the grid-specific application process 348, which may contain computer executable instructions executed by the processor 320 to perform functions relating to the techniques described herein. For example, the

techniques herein may be treated as a “dynamic forward/reverse grid application distribution process,” and may be located across a distributed set of participating devices 300, such as grid devices (e.g., primary, secondary, tertiary, and the like), grid service devices, and grid service controller devices, as described herein, with functionality of the process 348 specifically tailored to the particular device’s role within the techniques of the various embodiments detailed below.

Operationally, the techniques herein allow normally centralized grid control applications to be dynamically distributed to primary grid control devices in the utility grid, or sub-grid, and run on those distributed computing resources (e.g., primary, secondary, tertiary, etc. grid devices). This adapts the concept of virtualization into a cloud in a fashion referred to as “reverse cloud.” As mentioned above, in normal “forward cloud” virtualization, applications may be run on various distributed computing resources (e.g., desktop computers), and their input/output is collected into a centralized location called the cloud, which provides the virtualization environment to run the distributed resources remotely. In the case of distributed processing for electric utility grids, this process is reversed, and a centralized application is dynamically distributed and run on a distributed set of computing resources (e.g., primary or secondary grid devices).

As indicated above, the disclosed distributed intelligence architecture is not unidirectional. To the contrary, this architecture takes advantage of both reverse and forward cloud capability, due to the need to provide both non-uniform and dynamic distributions of grid control application computing. The techniques described herein thus allow for dynamic allocation of reverse/forward clouding. For example, in the case of an application that is hosted in a utility control center and that communicates with a large number of distribution grid data sources and control devices, it may be desirable to distribute the application to multiple primary substations, or primary grid control devices, so that each copy of the application may perform the application function for a subset of the measurement and control devices that are associated with the substation service area. Reverse cloud is used to distribute the application to the target substations. The substations may be arranged into a hierarchy as well, so that when the application is distributed to a primary substation, it may be further distributed (i.e., re-distributed) to second tier substations. Advantageously, this removes the correspondence between a grid control application execution environment and a specific piece of physical

hardware (e.g., a primary grid device, a server, etc.). In other words, forward/reverse clouding as part of a distributed intelligence architecture effectively deploys the grid control application execution environment(s) closer to the utility grid, or sub-grid, to be acted on by the grid control application, which reduces latency and increases scalability for grid control applications.

5 According to one or more of the embodiments herein, “application virtualization” provides a mechanism to control how applications are assigned to processors and may be used in “forward cloud” and “reverse cloud” modes in order to obtain the necessary flexibility to run smart grid functions dynamically across a utility grid.

10 Consider the following scenario: a grid control application (e.g., a demand response or “DR” management application) is hosted in a utility control center, and communicates with a large number of distribution grid data sources and control devices. It may be beneficial to distribute the grid control application to multiple primary substations, or grid control devices, so that each copy of the application may perform the grid control application function for a subset of the measurement and control devices that are associated with the substation service area (e.g.,
15 DR management of a corresponding subset/sub-grid). In this instance, reverse clouding may be used to distribute the application to the target substations, or grid control devices. It may also be beneficial to arrange the substations, or grid control devices, into a hierarchy as well, so that when the grid control application is distributed to a primary substation, or grid control device, it may then be re-distributed to second tier substations, or secondary grid control devices. This
20 will be further exemplified with respect to two situations.

For example, consider the case in which an AA distribution substation 131 platform in sub-grid 1562 requires repair, or an upgrade, and must be unavailable for a certain period of time. As shown in FIG. 15, the grid control applications associated with the AA distribution substation 131 platform in sub-grid 1562 may be virtualized back up to the next higher tier in the
25 utility grid capable of hosting the grid control application (i.e., forward clouding), and the grid control application is operated at the higher tier as though it were operating at the lower tier. In other words, AA distribution substation 131 in sub-grid 1562 now has “virtual” zone control, since the zone control grid application is actually being performed elsewhere; in this case, by the regional control grid application of control center(s) 1515. Similarly, if distribution substations
30 130 in sub-grid 1564 do not have the necessary computing platforms to support a particular grid

control application, forward clouding may be used to move the application back to the next higher tier (e.g., AA distribution substation 131) of the utility grid and be operated as though it were operating at the lower tier. In other words, distribution substations 130 in sub-grid 1562 now have virtual area control.

5 In application virtualization, forward clouding may be used as the mechanism to pull services and applications back from a lower processing tier when conditions require it, or when the application/service is no longer required by the lower tier. As such services/conditions may be temporary, it is important for the grid architecture (e.g., a control system or engineer) to be able to control the manner in which applications such as, for example, grid control applications,
10 are virtualized.

FIG. 16 illustrates an approach to the management and operation of a distributed intelligence architecture. According to an example embodiment, the distributed intelligence architecture 1600 is based on a centralized network management system 1610, that has the ability to push (i.e., reverse cloud) grid application(s), or grid application functions, from a
15 secure grid application code database 1640 to other grid devices (e.g., control center 1515, AA distribution substation 131, distribution substation 130, feeder 140, cloud server 1675, Field Area Network device 1680, grid attached device 1685, and the like) to operate autonomously as needed. The network management system 1610 pushes grid applications, or grid application functions, via a modified server virtualization hypervisor 1650 that includes an inverse cloud
20 manager 1655 and a virtual server relocater 1670.

Network management system 1610 determines, based on grid structural information provided by grid CIM device 1620 and/or grid topology information provided by network topology device 1630, whether there is a requirement (or other desire or trigger) for a particular grid application function in the utility grid. If such a determination is made, network
25 management system 1610 also determines which grid devices should receive the grid application function, and then pushes the appropriate grid application functions to the appropriate devices via the modified server virtualization hypervisor 1650. According to an example embodiment, network management system 1610 comprises a user interface 1615 by which a user can manually determine when and where to distribute a grid application.

Network management system 1610 also monitors, via information received from grid CIM device 1620 and network topology device 1630, the continued use of (e.g., need for) grid application functions that have been reverse clouded (i.e., pushed) to other grid devices. If network management system 1610 determines that such a need/use no longer exists, it pulls
5 (withdraws) the relevant grid application functions back via the modified server virtualization hypervisor 1650.

According to an exemplary embodiment, network management system 1610 may be integrally associated with a primary grid device. Alternatively, network management system 1610 may be provided as a stand alone management system service that is in communication
10 with one or more primary grid devices.

FIG. 17 illustrates an example simplified procedure 1700 for a distributed intelligence architecture with dynamic reverse and forward clouding in accordance with one or more embodiments described herein. The procedure 1700 may start at step 1705, and continues to step 1710, where, as described in greater detail above, a network management system 1610 associated
15 with a primary grid device (e.g., control center 115) hosts a grid application function.

The procedure continues to step 1715, where the network management system 1610 determines, based on grid structure and topology information provided by grid CIM device 1620 and network topology device 1630, whether to distribute at least a portion of a grid application function from grid application code database 1640 to one or more distributed secondary grid
20 devices (e.g., AA distribution substation 131, distribution substation 130, etc.).

Upon determining that at least a portion of a grid application function should be distributed in step 1720, the procedure continues to step 1725 in which the network management system 1610 distributes, via inverse cloud manager 1655 and virtual server relocater 1670, at least a portion of the grid application function to the one or more distributed secondary grid
25 devices.

The network management system 1610 monitors grid structure and topology information provided by grid CIM device 1620 and network topology device 1630, and in step 1730 determines whether the grid application function is still required at the one or more secondary grid devices.

Upon determining that the grid application function is no longer needed in step 1735, network management system 1610 withdraws the grid application function to the primary grid device in step 1740.

The procedure 1700 may then illustratively end in step 1745, though notably with the option to return to any appropriate step described above based on the dynamicity of the forward and reverse clouding as detailed within the disclosure above.

It should be noted that while certain steps within procedure 1700 may be optional as described above, the steps shown in FIG 17 are merely examples for illustration, and certain other steps may be included or excluded as desired. Further, while a particular order of the steps is shown, this ordering is merely illustrative, and any suitable arrangement of the steps may be utilized without departing from the scope of the embodiments herein.

The techniques described herein, therefore, provide for a distributed intelligence grid control architecture with both dynamic reverse and forward clouding. In particular, the techniques herein may dynamically distribute an application to multiple primary substations so that each copy of the application may perform the application function for a subset of the measurement and control devices that are associated with the substation service area. The actual functionality may be pushed downward or upward dynamically based on real-time needs and/or specific functionality.

Notably, a layered network services architecture approach addresses complexity management for smart grids at scale, one of the most challenging smart grid design issues. Short term adoption of a layered network services architecture allows for efficient transition to new control systems that are hybrids of distributed elements with centralized management. Later, as smart grid implementations approach full scale (in any given dimension), complexity management and the other smart grid architecture issues will benefit from a layered network services architecture.

Said differently, now that communications and embedded processing capabilities are becoming available in forms that utility companies can use, a major obstacle in the adoption of distributed intelligence is that utility companies cannot make large changes in their systems in discrete steps. Rather they must go through transitions that can take years to complete. This is

due to the nature of their mission and the financial realities utility companies face. In practice, utilities need to transition from centralized to distributed intelligence, and to operate in a complicated hybrid mode for long periods of time, perhaps permanently. This means that the utility service provider needs to be able to roll out distributed intelligence incrementally while maintaining full operations over the entire service area, and be able to modify the distributed architecture appropriately over time and geography. Simply having a distributed architecture implementation is not sufficient; it needs to be easily and continually mutable in terms of what functionality is distributed to which processing locations in the grid and be capable of coexisting with legacy control systems where they remain in place.

The present disclosure thus presents one or more specific features of a distributed intelligence platform that supports variable topology over both time and geography. The platform provides the mechanisms to locate, execute, and re-locate applications and network services onto available computing platforms that may exist in control and operations centers, substations, field network devices, field edge devices, data centers, monitoring centers, customer premises devices, mobile devices, and servers that may be located in power delivery chain entities external to the Transmission and Distribution utility. These techniques use a communication network as a future-proofed platform to incrementally and variably implement distributed intelligence and thereby achieve the associated benefits without being forced to make an untenable massive switchover or to use a single fixed architecture everywhere in its service area.

While there have been shown and described illustrative embodiments that provide for auto-discovery and provisioning of utility grid control operation services, it is to be understood that various other adaptations and modifications may be made within the spirit and scope of the embodiments herein. For example, the embodiments have been shown and described herein with relation to electric grids. However, the embodiments in their broader sense are not as limited, and may, in fact, be used with other types of utility grids, such as gas, water, etc., or specific types of “smart” networks where appropriate. For example, in addition to utility grids, recent trends indicate that the future will progress towards sensor-actuator based automation in various sectors including buildings, communities/cities, transportation, energy, etc. Experts predict that in the coming decades there will be a fabric of trillions of sensor-actuator devices embedded into

our surroundings. This fabric will bring about integrated automation that will greatly improve the efficiency of the environment/resources as well as the quality of living for the human and living being within the environment. Moreover, while certain protocols are shown, other suitable protocols may be used, accordingly.

5 Illustratively, the techniques herein can span the entire power delivery chain out to and including networks outside of the utility but connected to it. In addition, the techniques herein apply to all of the other agencies, such as:

- Rail systems – electric rail power control and monitoring , all rail and car condition monitoring, route control, accident detection/prevention, mobile WiFi, control
10 centers;
- Roadways/highways – hazard detection (fog/ice/flooding/earthquake damage), bridge/overpass structural condition, congestion monitoring, emergency response support, transit control facilities;
- Rivers and canals – locks and dams, flooding detection/extent measurement, dikes
15 and levees, flow/depth, traffic flow;
- Sewage/wastewater/storm drain systems – treatment plants, flow/blockage monitoring, leak/spill detection;
- Etc.

The foregoing description has been directed to specific embodiments. It will be apparent,
20 however, that other variations and modifications may be made to the described embodiments, with the attainment of some or all of their advantages. For instance, it is expressly contemplated that the components and/or elements described herein can be implemented as software being stored on a tangible (non-transitory) computer-readable medium (e.g., disks/CDs/RAM/EEPROM/etc.) having program instructions executing on a computer,
25 hardware, firmware, or a combination thereof. Accordingly this description is to be taken only by way of example and not to otherwise limit the scope of the embodiments herein. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the true spirit and scope of the embodiments herein.

CLAIMS

What is claimed is:

- 1 1. A method, comprising:
 - 2 hosting a grid application function at a primary grid device in a utility grid;
 - 3 determining whether to distribute at least a portion of the grid application function to one
 - 4 or more distributed secondary grid devices in the utility grid, the one or more distributed
 - 5 secondary grid devices associated with a subset of the utility grid associated with the primary
 - 6 grid device; and
 - 7 dynamically distributing the grid application function according to the determining.
- 1 2. The method as in claim 1, wherein the grid application function at the primary device is
- 2 distributed from an originating grid device associated with a superset of the utility grid
- 3 associated with the primary grid device.
- 1 3. The method as in claim 2, further comprising:
 - 2 determining that the grid application function is no longer to be used at the primary grid
 - 3 device; and, in response,
 - 4 withdrawing the grid application function from the primary grid device back to the
 - 5 originating grid device.
- 1 4. The method as in claim 1, wherein determining is based on analysis of utility grid parameters
- 2 selected from a group consisting of: a processing capability of the originating, primary, or
- 3 secondary grid device; a processing load on the originating, primary, or secondary grid device; a
- 4 latency requirement for the grid application function; a state of the utility grid; a state of the
- 5 subset of the utility grid; a size of the utility grid associated with the primary device; a size of the

utility grid associated with the one or more distributed secondary devices; and any combination thereof.

5. The method as in claim 1, wherein dynamically distributing further comprises:

determining that the grid application function is no longer to be used at the secondary grid device; and, in response,

withdrawing the grid application function from the secondary grid device back to the primary grid device.

6. The method as in claim 1, wherein dynamically distributing further comprises:

distributing responsibility for the grid application function.

7. The method as in claim 1, wherein dynamically distributing further comprises:

distributing an application code for the grid application function.

8. The method as in claim 1, wherein determining comprises:

receiving user input directing the determining.

9. The method as in claim 1, wherein the at least a portion of the grid application function comprises all functions related to the subset of a corresponding secondary grid device or a portion of all the functions.

10. The method as in claim 1, wherein the at least a portion of the grid application function corresponds to the utility grid associated with the primary grid device.

11. The method as in claim 1, wherein determining further comprises:

making an individual determination as to the processing capacity of each of the one or more distributed secondary grid devices in the utility grid; and

dynamically distributing the grid application function according to the individual determination.

12. The method as in claim 1, wherein the originating, primary, and/or secondary grid devices comprise at least one processing element selected from the group consisting of: enterprise servers; cloud virtualized servers; cloud servers; operations/control center virtualized servers; substation computing platforms; field area routers; embedded grid devices; balance/interchange servers; premises devices; and grid-attached embedded devices/systems.

13. An apparatus, comprising:

one or more network interfaces to communicate with a utility grid or sub-grid computer network;

a processor coupled to the network interfaces and adapted to execute one or more processes; and

a memory configured to store a process executable by the processor, the process when executed operable to:

host a grid application function as a primary grid device in the utility grid;

determine whether to distribute at least a portion of the grid application function to one or more distributed secondary grid devices in the utility grid, the one or more distributed secondary grid devices associated with a subset of the utility grid associated with the primary grid device; and

dynamically distribute the grid application function according to the determination.

1 14. The apparatus as in claim 13, wherein the grid application function at the primary device is
2 distributed from an originating grid device associated with a superset of the utility grid
3 associated with the primary grid device.

1 15. The apparatus as in claim 14, wherein the process when executed is further operable to:
2 determine that the grid application function is no longer to be used at the primary grid
3 device; and, in response,
4 withdraw the grid application function from the primary grid device back to the
5 originating grid device.

1 16. The apparatus as in claim 13, wherein determine is based on analysis of utility grid
2 parameters selected from the group consisting of a processing capability of the originating,
3 primary, or secondary grid device; a processing load on the originating, primary, or secondary
4 grid device, a latency requirement for the grid application function, a state of the utility grid, a
5 state of the subset of the utility grid, a size of the utility grid associated with the primary device,
6 a size of the utility grid associated with the one or more distributed secondary devices, and any
7 combination thereof.

1 17. The apparatus as in claim 13, wherein the process when executed is further operable to:
2 determine that the grid application function is no longer to be used at the secondary grid
3 device; and, in response,
4 withdraw the grid application function from the secondary grid device back to the
5 primary grid device.

1 18. The apparatus as in claim 13, wherein the process, when executed to dynamically distribute,
2 is further operable to:

3 distribute at least one of either responsibility or application code for the grid application
4 function.

1 19. A tangible, non-transitory, computer-readable media having software encoded thereon, the
2 software, when executed by a processor at a primary grid device in a utility grid, operable to:

3 host a grid application function;

4 determine whether to distribute at least a portion of the grid application function to one or
5 more distributed secondary grid devices in the utility grid, the one or more distributed secondary
6 grid devices associated with a subset of the utility grid associated with the primary grid device;
7 and

8 dynamically distribute the grid application function according to the determining.

1 20. The computer-readable media of claim 19, wherein the software when executed is further
2 operable to:

3 determine that the grid application function is no longer to be used at the secondary grid
4 device; and, in response,

5 withdraw the grid application function from the secondary grid device back to the
6 primary grid device.

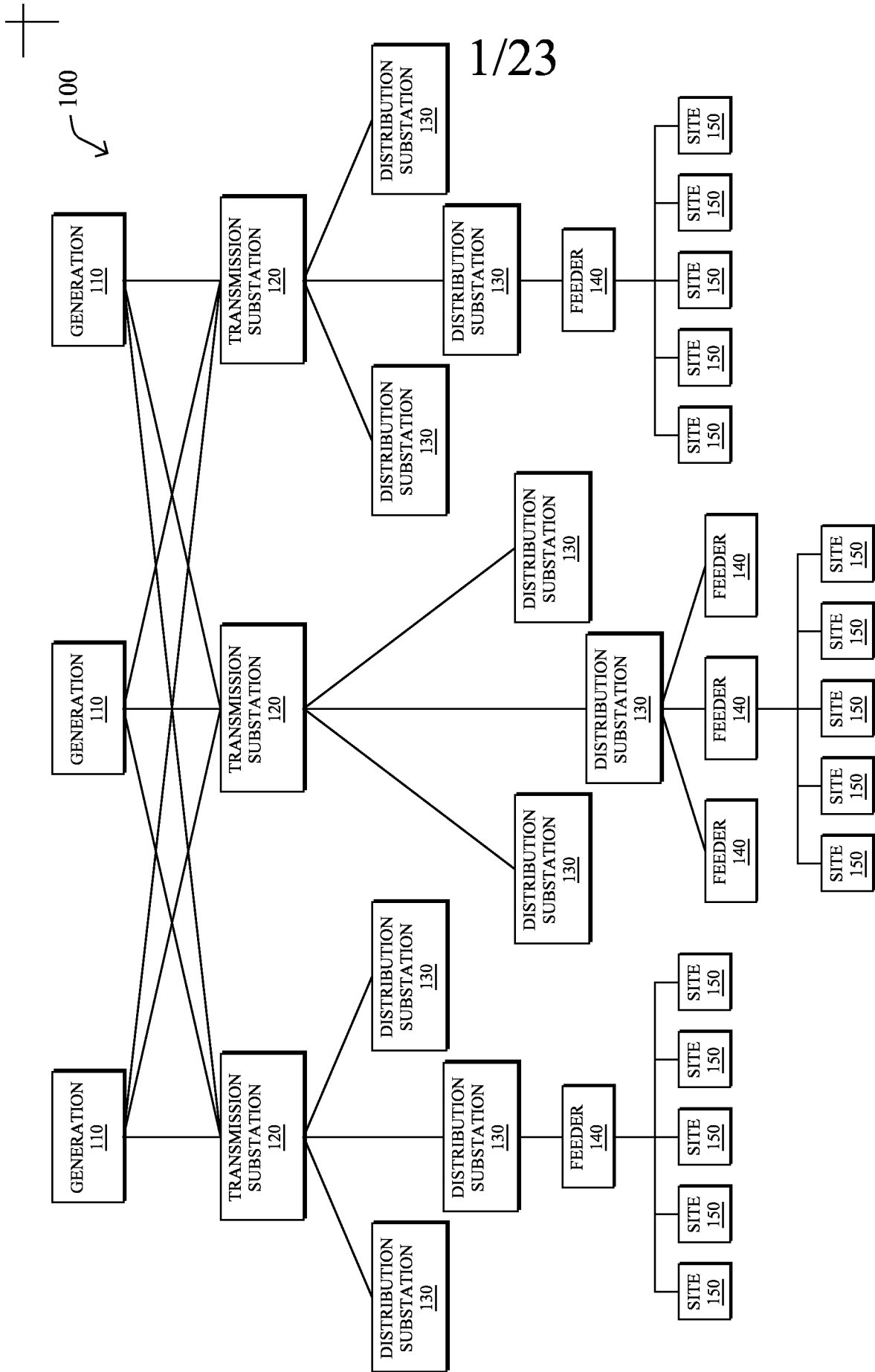


FIG. 1

2/23

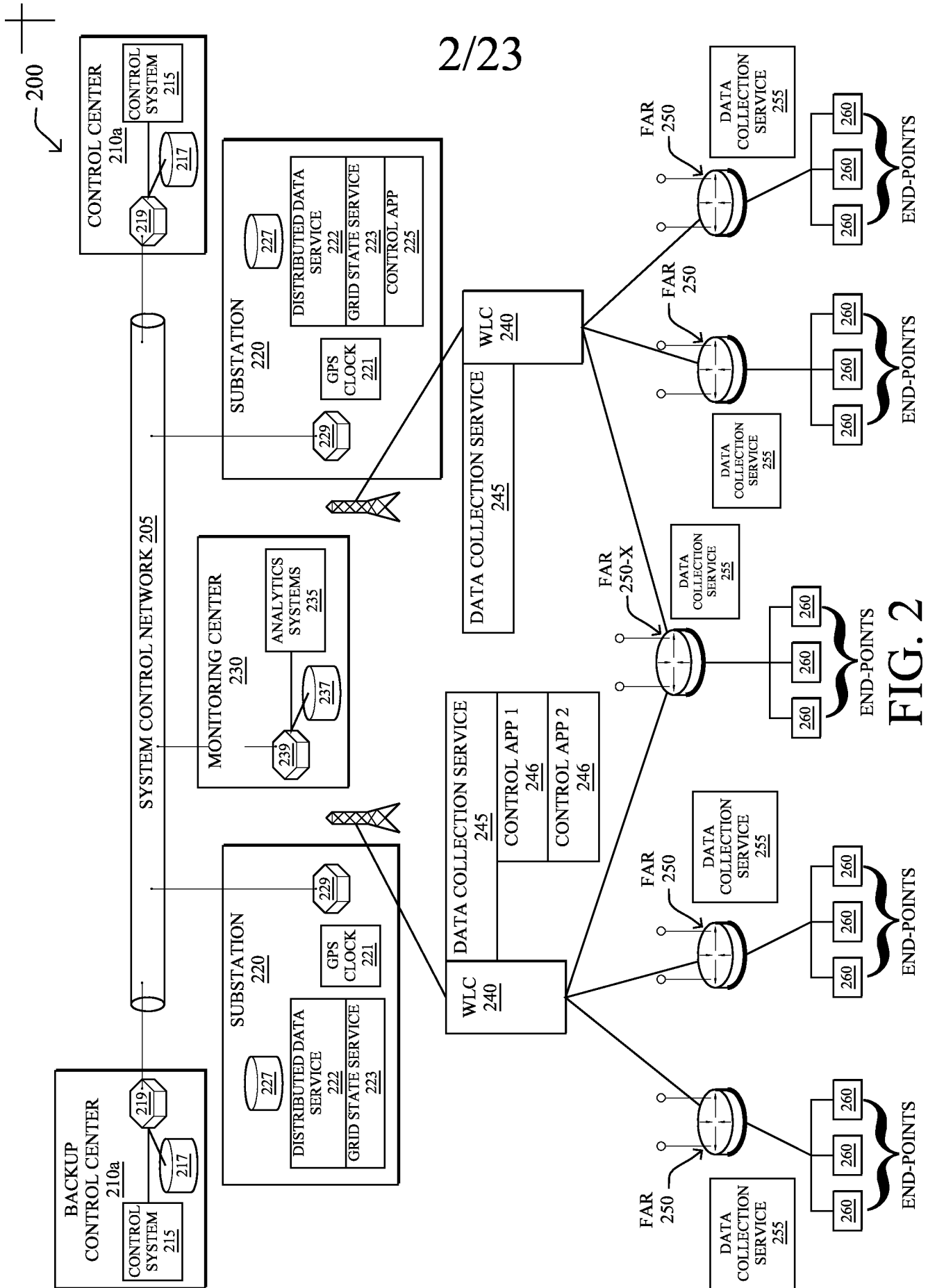


FIG. 2

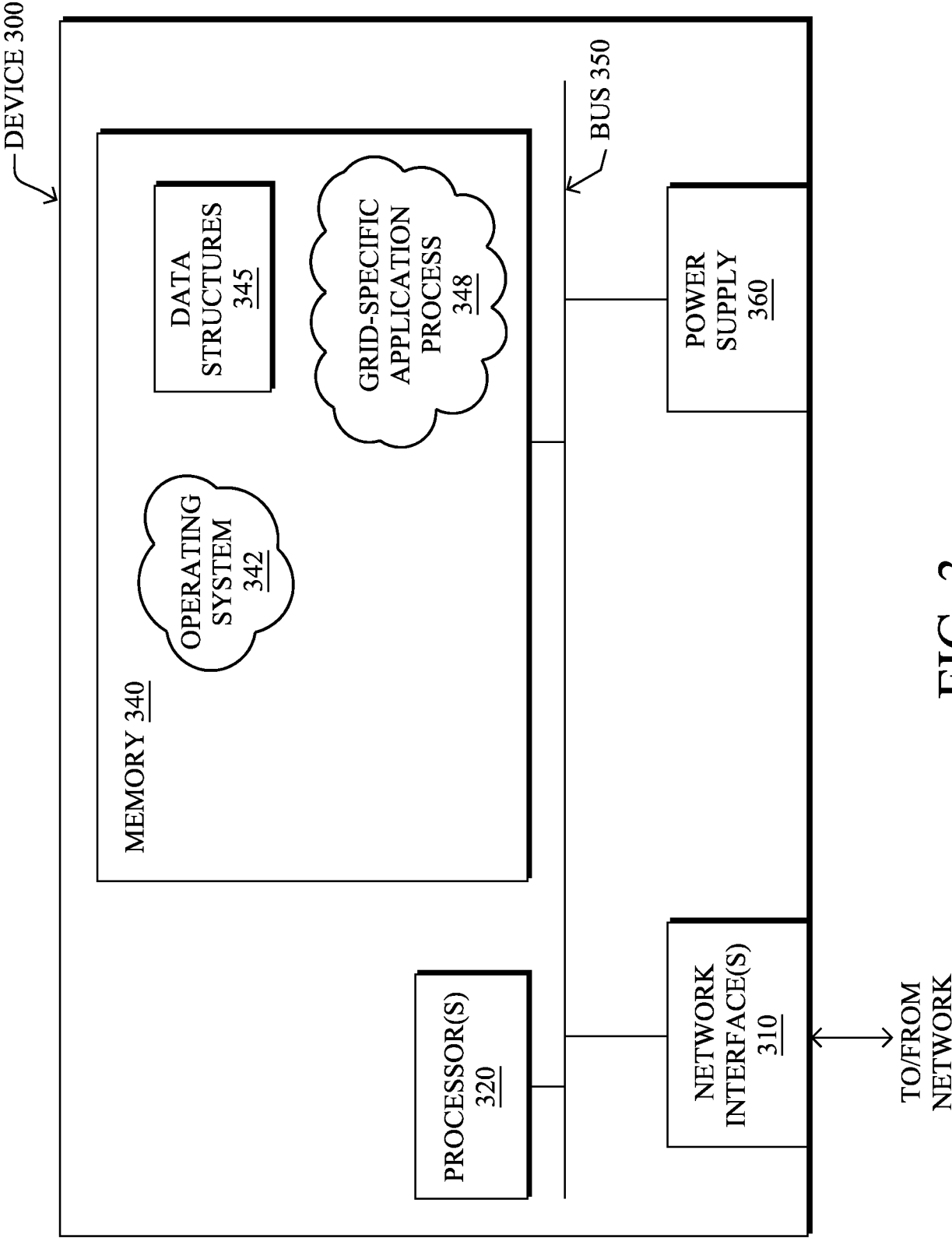


FIG. 3

TABLE 400: ISSUES FOR SMART GRID AT SCALE

ISSUE	COMMENT
ULS CHARACTERISTICS OF SMART GRIDS AT SCALE	• DECENTRALIZED CONTROL • INHERENTLY CONFLICTING DIVERSE REQUIREMENTS • CONTINUOUS EVOLUTION AND DEPLOYMENT • HETEROGENEOUS, INCONSISTENT, AND CHANGING ELEMENTS • NORMAL FAILURES
HIDDEN COUPLING VIA THE GRID	SYSTEMS AND CONTROLS ARE INHERENTLY COUPLED THROUGH GRID ELECTRICAL PHYSICS AND THEREFORE INTERACT IN WAYS THAT MAY BE UNACCOUNTED FOR IN EXISTING SYSTEM DESIGNS
GRID AS A MULTI-OBJECTIVE, MULTI-CONTROL SYSTEM	MULTIPLE CONTROLS AFFECTING THE SAME GRID PORTIONS; SOME OF THE CONTROLS LIE OUTSIDE OF THE UTILITY; CONTROLS OPERATING ON MULTIPLE TIME SCALES (SEE ALSO LATENCY HIERARCHY BELOW)
LOCALITY OF CONTROL	BULK OR AGGREGATE CONTROL COMMANDS, ESPECIALLY AS REGARDS SECONDARY LOAD CONTROL AND STABILIZATION, ARE NOT CURRENTLY BROKEN DOWN TO THE FEEDER OR EVEN SECTION LEVEL, TAKING INTO ACCOUNT GRID STATE AT THE RESPECTIVE LEVEL
LATENCY HIERARCHY	SMART GRID-GENERATED DATA MUST BE USED ON ANY OF A NUMBER OF LATENCY SCALES, SOME OF WHICH ARE QUITE SHORT, THUS PRECLUDING PURELY CENTRALIZED PROCESSING AND CONTROL APPROACHES

FIG. 4

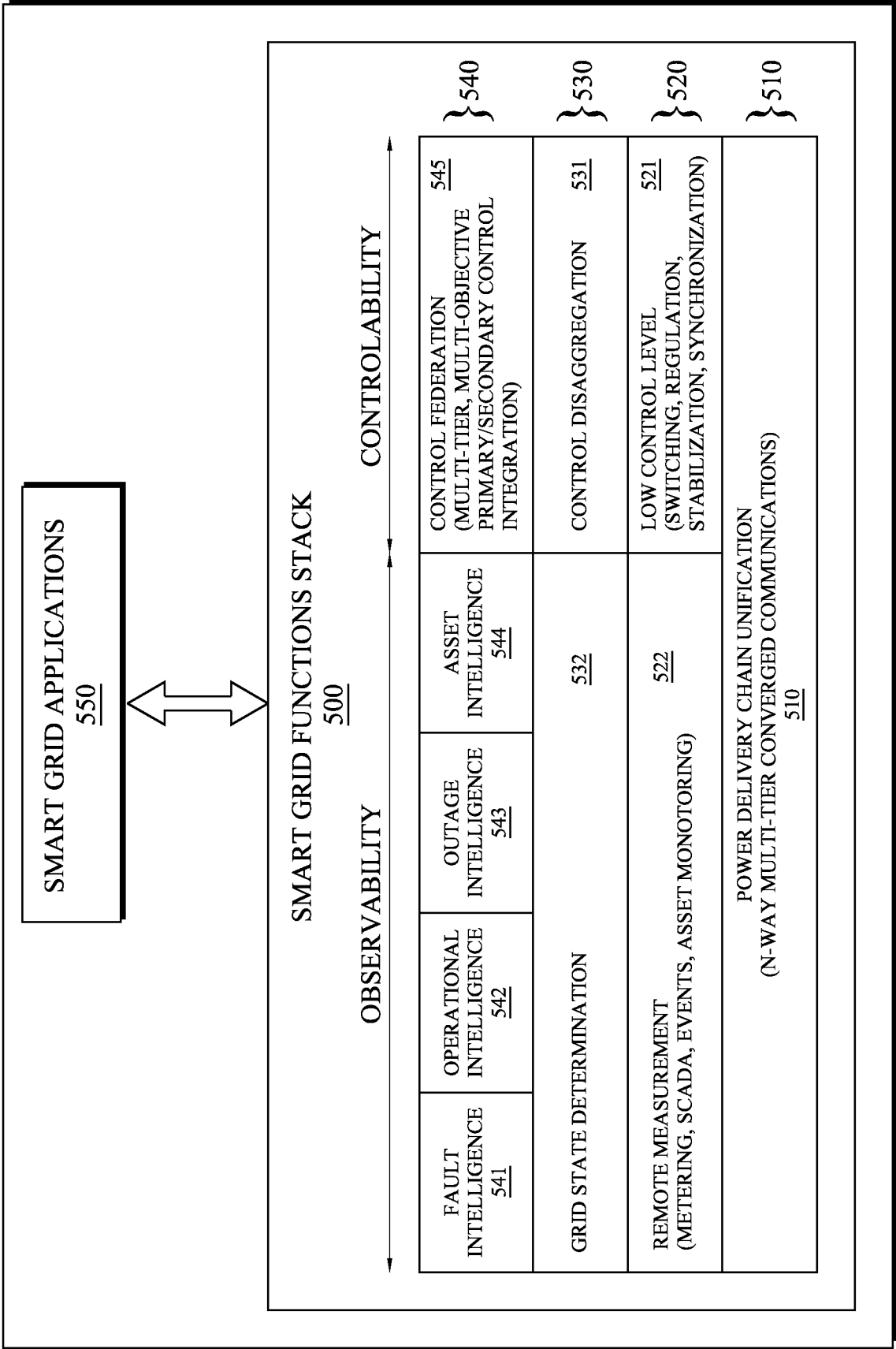


FIG. 5

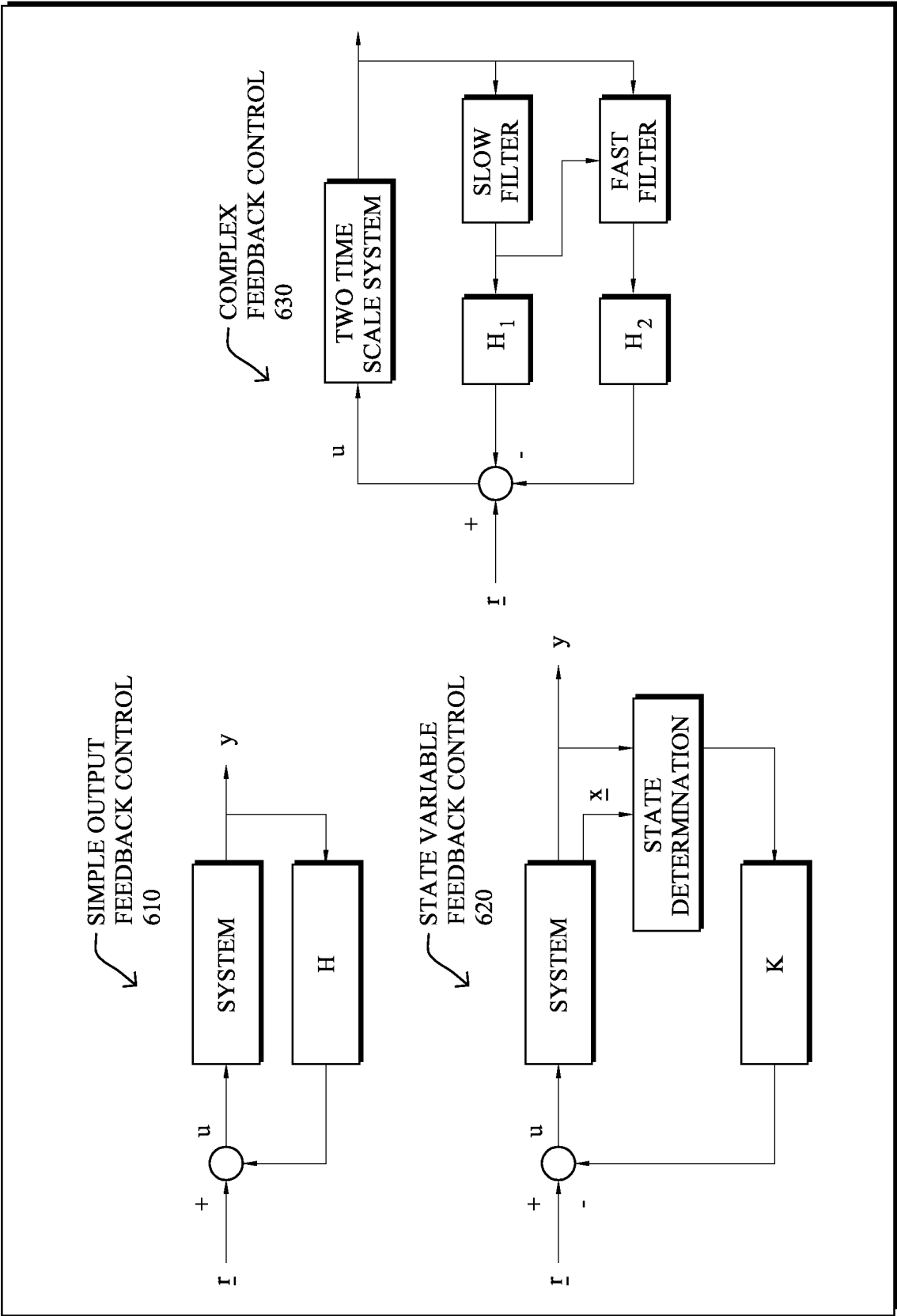


FIG. 6

700

GRID OBJECT	LATENCY	PROCESS	OUTPUT/USAGE
BUSINESS DATA REPOSITORY	DAYS TO MONTHS	BUSINESS INTELLIGENCE	KPI'S, DASHBOARDS, REPORTS (HMI)
HISTORICAL DATA	MINUTES TO DAYS	TRANSACTIONAL ANALYSIS	VIRTUALIZATION REPORTING (HMI)
			VIRTUALIZATION REPORTING (HMI)
OP NON-OP DATA	SECONDS TO SUB-MINUTE	MEDIUM SPEED/ MEDIUM LATENCY REAL TIME ANALYSIS	VISUALIZATION (HMI)
			SYSTEMS TO PROCESSES (M2M)
GRID SENSORS AND DEVICES	MILLISECONDS TO SUB-SECOND	HIGH SPEED/ LOW LATENCY REAL TIME ANALYSIS	PROTECTION AND CONSTROL SYSTEMS (M2M)

VERY HIGH LATENCY

VERY LOW LATENCY

FIG. 7

DATA LIFESPAN CLASS	CHARACTERISTICS
TRANSIT	DATA EXISTS FOR ONLY THE TIME NECESSARY TO TRAVEL FROM SOURCE TO SINK AND BE USED; IT PERSISTS ONLY MOMENTARILY IN THE NETWORK AND THE DATA SINK AND IS THEN DISCARDED; EXAMPLES ARE AN EVENT MESSAGE USED BY PROTECTION RELAYS, AND SENSOR DATA USED IN CLOSED LOOP CONTROLS; PERSISTENCE TIME MAY BE MICROSECONDS
BURST/FLOW	DATA THAT IS PRODUCED IN BURSTS OR IS PROCESSED IN BURSTS MAY EXIST TEMPORARILY IN FIFO QUEUES OR CIRCULAR BUFFERS UNTIL IT IS CONSUMED OR OVERWRITTEN; EXAMPLES INCLUDE TELEMETRY DATA AND ASYNCHRONOUS EVENT MESSAGES (ASSUMING THEY ARE NOT LOGGED) - OFTEN THE STORAGE FOR THESE DATA ARE INCORPORATED DIRECTLY INTO APPLICATIONS, E.G. CEP ENGINE EVENT BUFFERS
OPERATIONAL	DATA THAT MAY BE USED FROM MOMENT TO MOMENT BUT IS CONTINUALLY UPDATED WITH REFRESHED VALUES SO THAT OLD VALUES ARE OVERWRITTEN SINCE ONLY PRESENT (FRESH) VALUES ARE NEEDED; EXAMPLE: GRID (POWER) STATE DATA SUCH AS SCADA DATA THAT MAY BE UPDATED EVERY FEW SECONDS
TRANSACTIONAL	DATA THAT EXISTS FOR AN EXTENDED BUT NOT INDEFINITE TIME; TYPICALLY USED IN TRANSACTION PROCESSING AND BUSINESS INTELLIGENCE APPLICATIONS; STORAGE MAY BE IN DATABASES INCORPORATED INTO APPLICATIONS OR IN DATA WAREHOUSES, DATAMARTS OR BUSINESS DATA REPOSITORIES
ARCHIVAL	DATA THAT MUST BE SAVED FOR VERY LONG (EVEN INDEFINITE) TIME PERIODS; INCLUDES METER USAGE DATA (E.G. SEVEN YEARS), PMU DATA AT ISO/RTOS (SEVERAL YEARS); LOG FILES. NOTE THAT SOME DATA MAY BE RETAINED IN MULTIPLE COPIES; FOR EXAMPLE, ISO/S MUST RETAIN PMU DATA IN QUADRUPLICATE.

FIG. 8

9/23

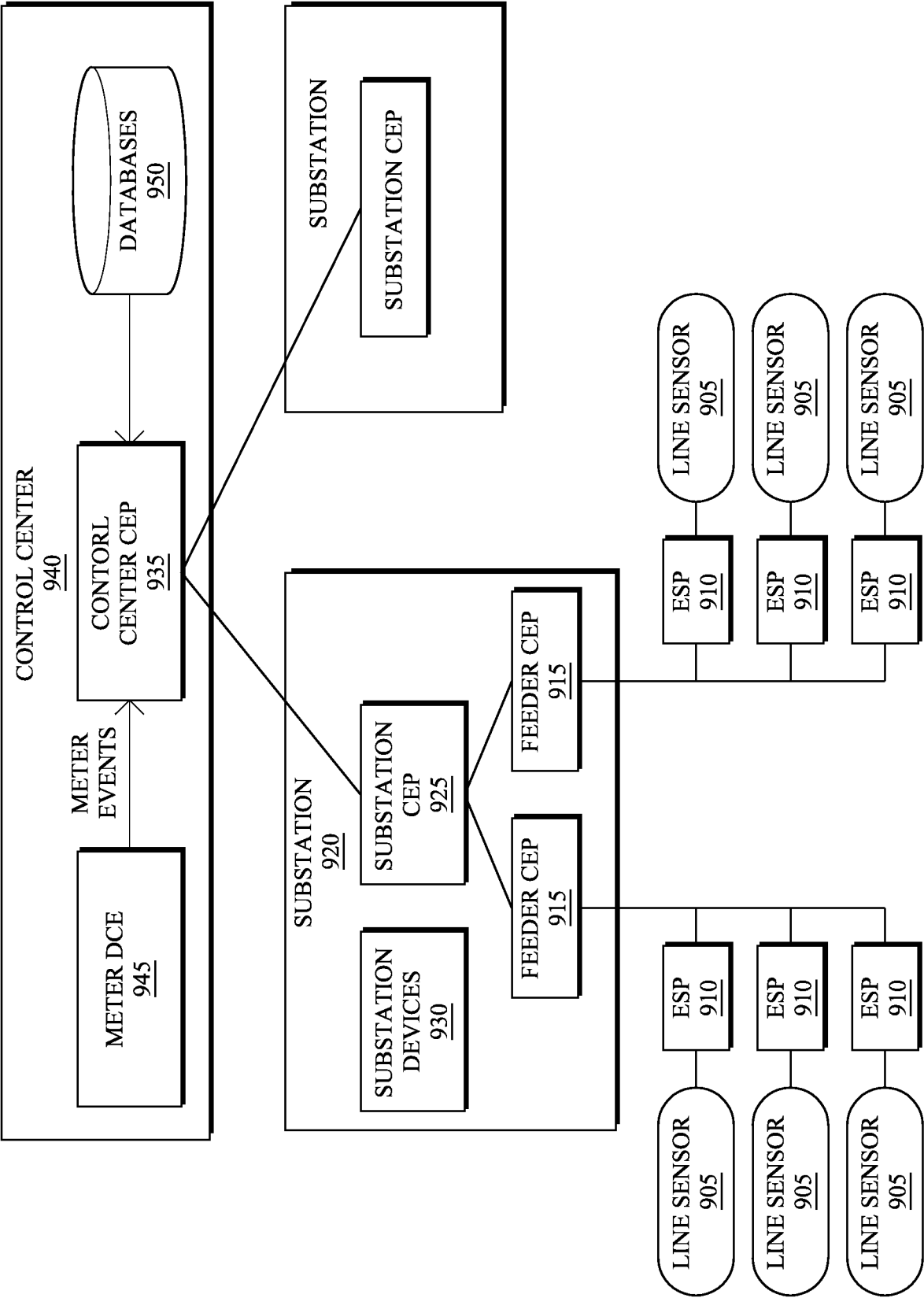


FIG. 9

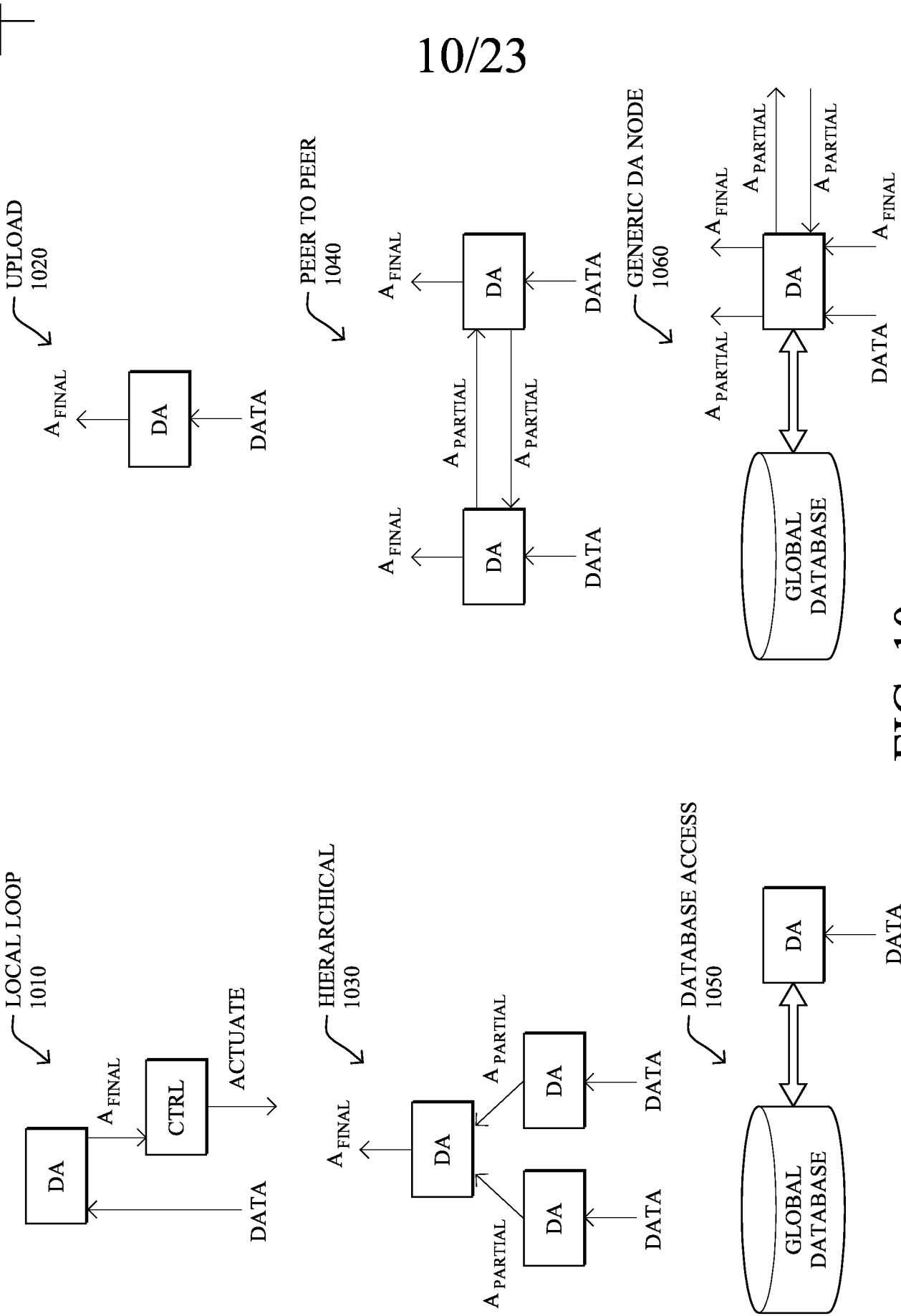


FIG. 10

11/23

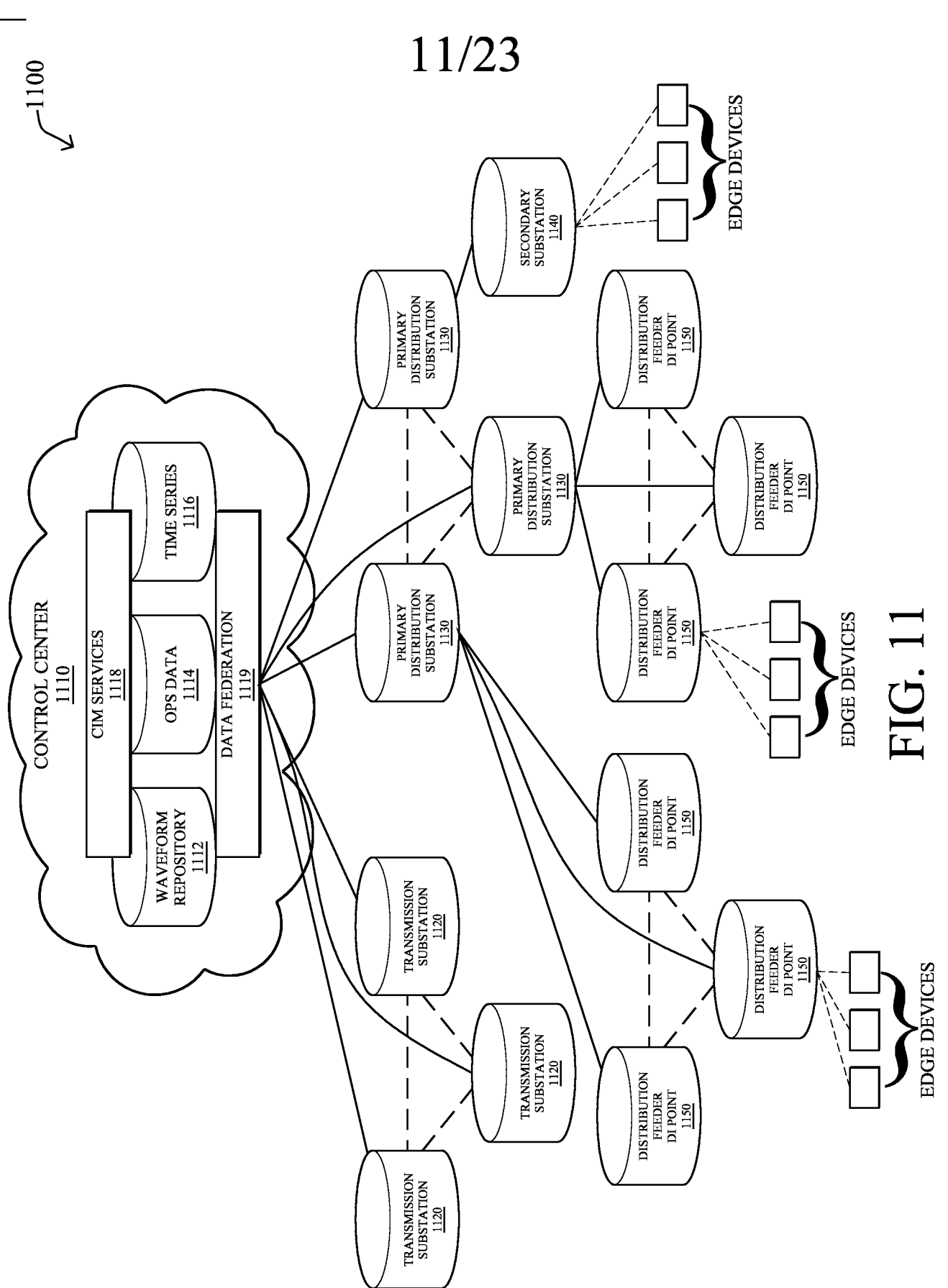


FIG. 11

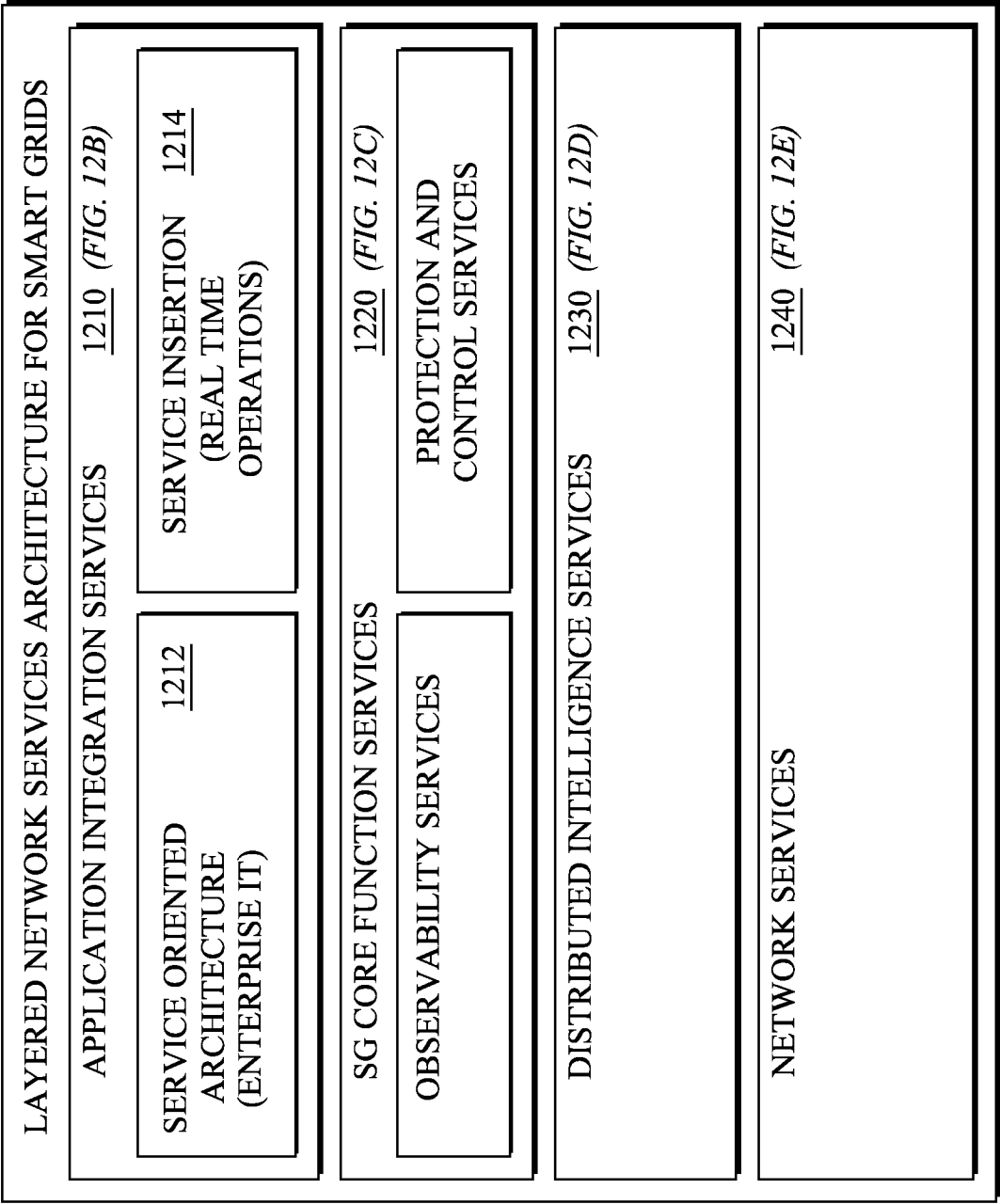


FIG. 12A

+

1200

✓

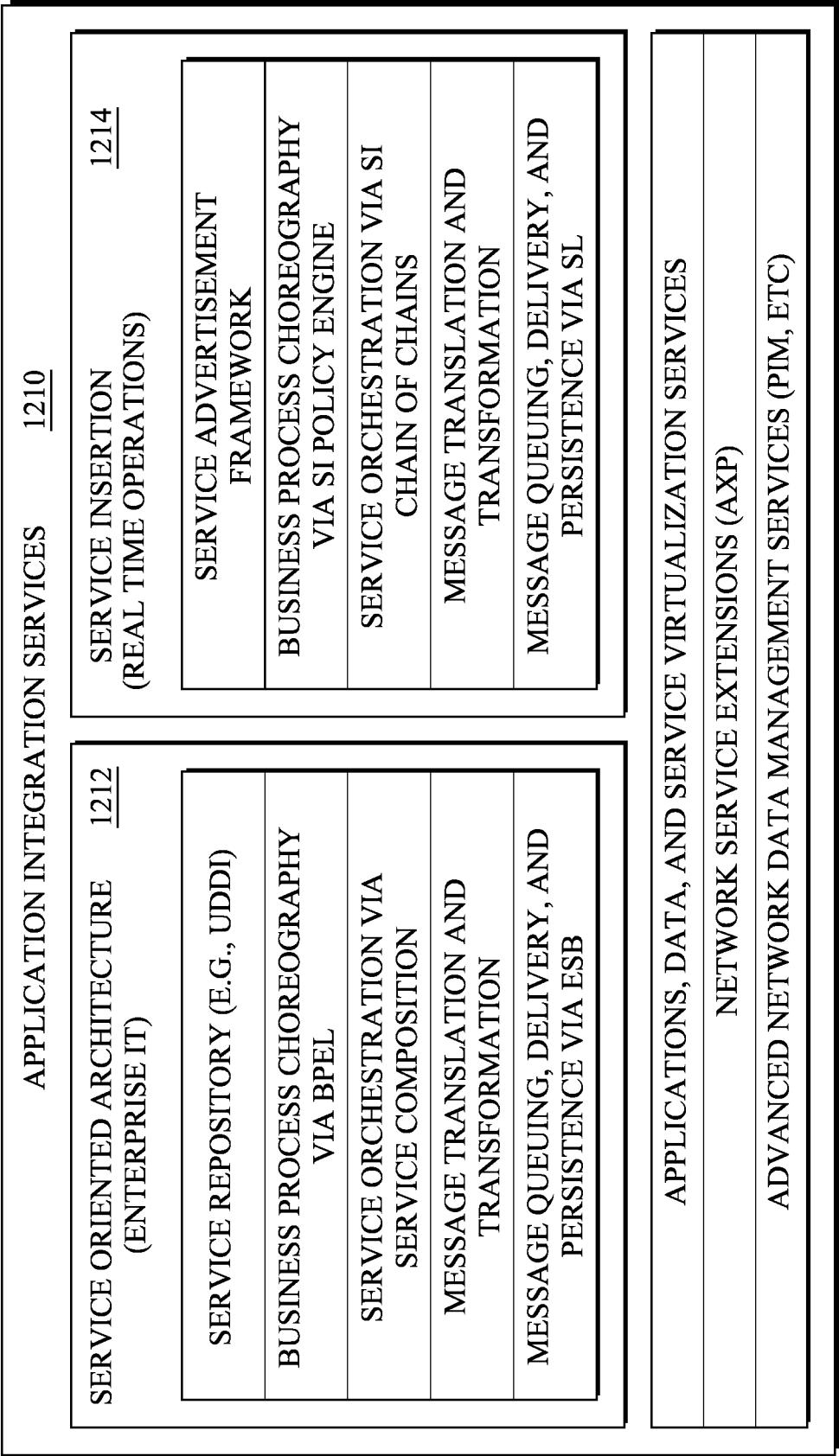


FIG. 12B

+

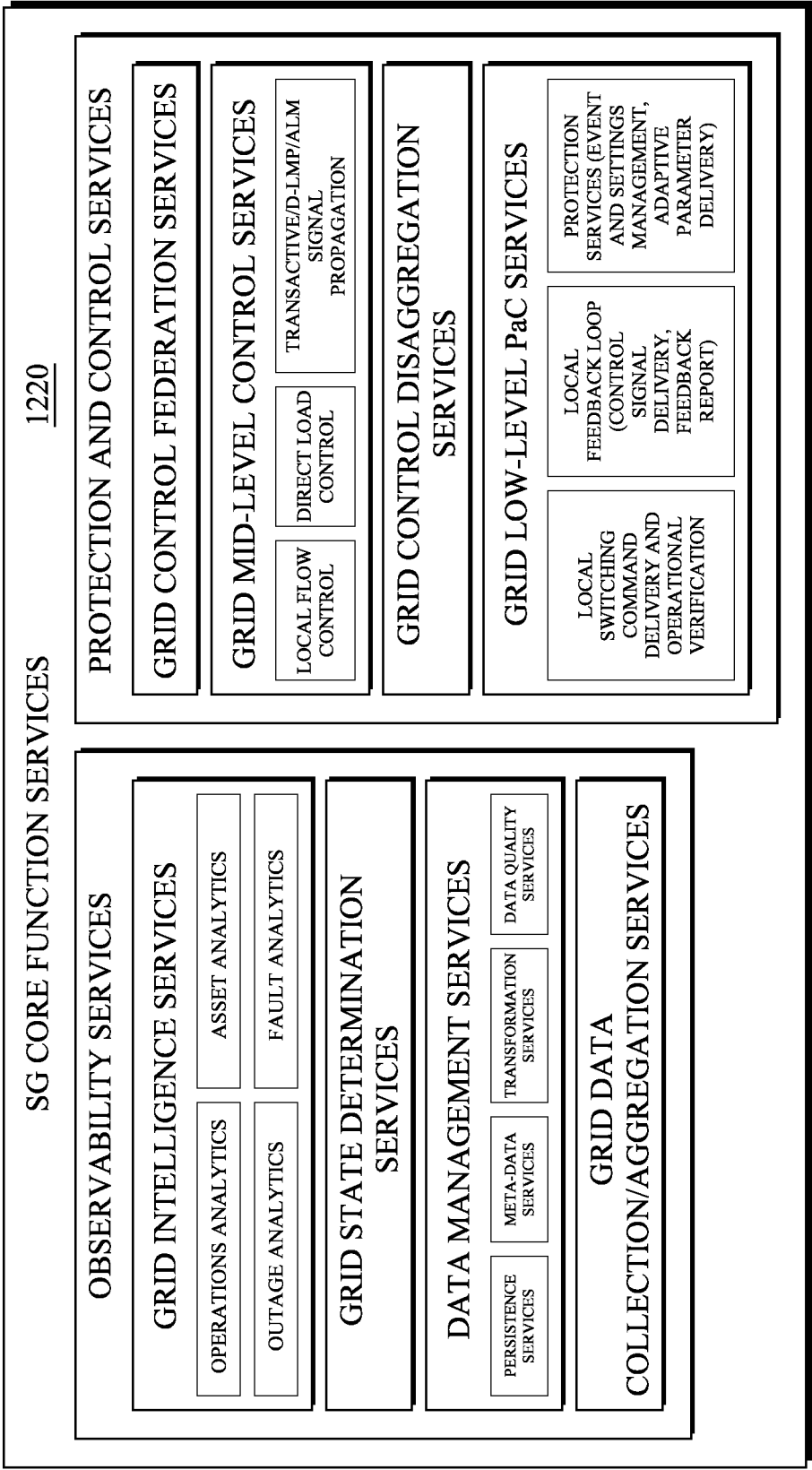
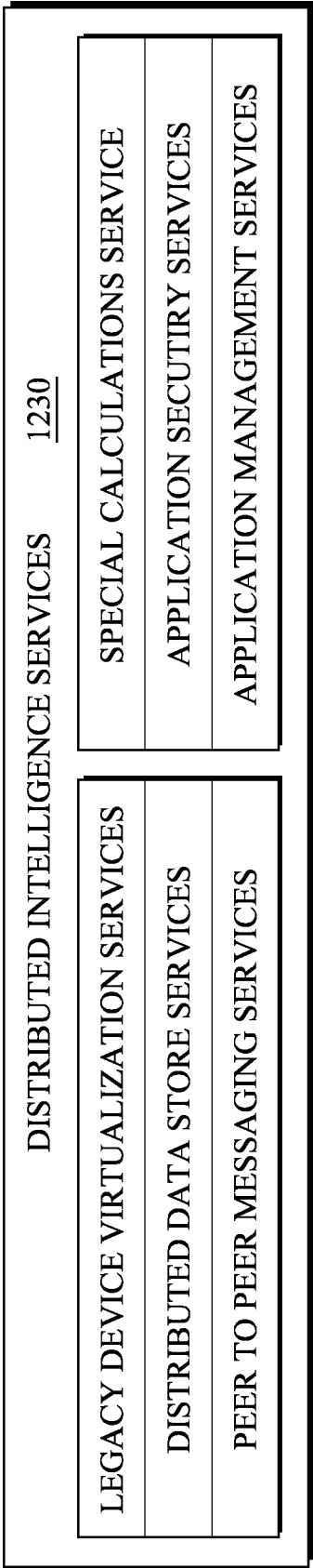


FIG. 12C



1200 ↘



15/23

FIG. 12D

1200 ↘

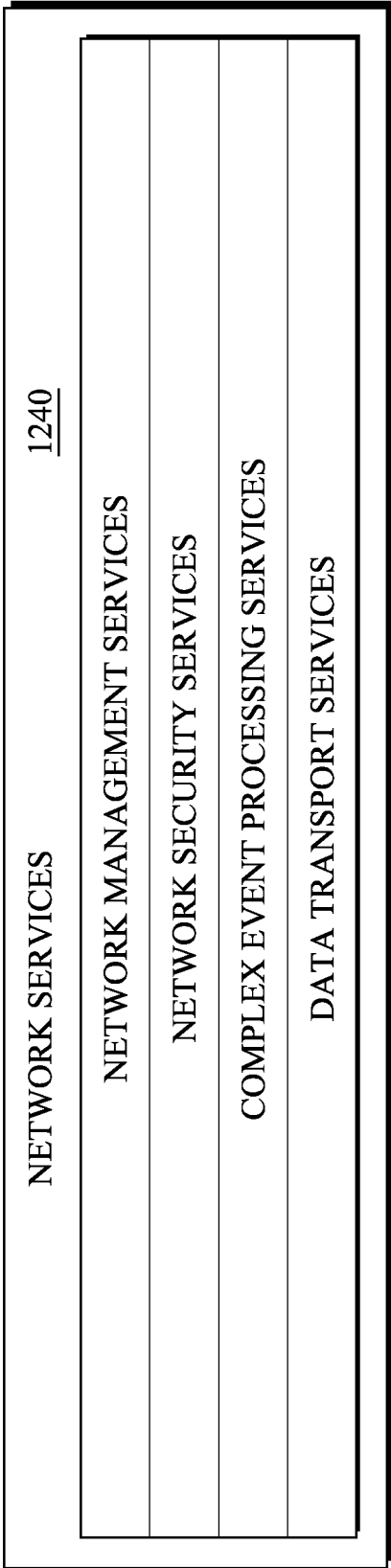


FIG. 12E



+

1300

16/23

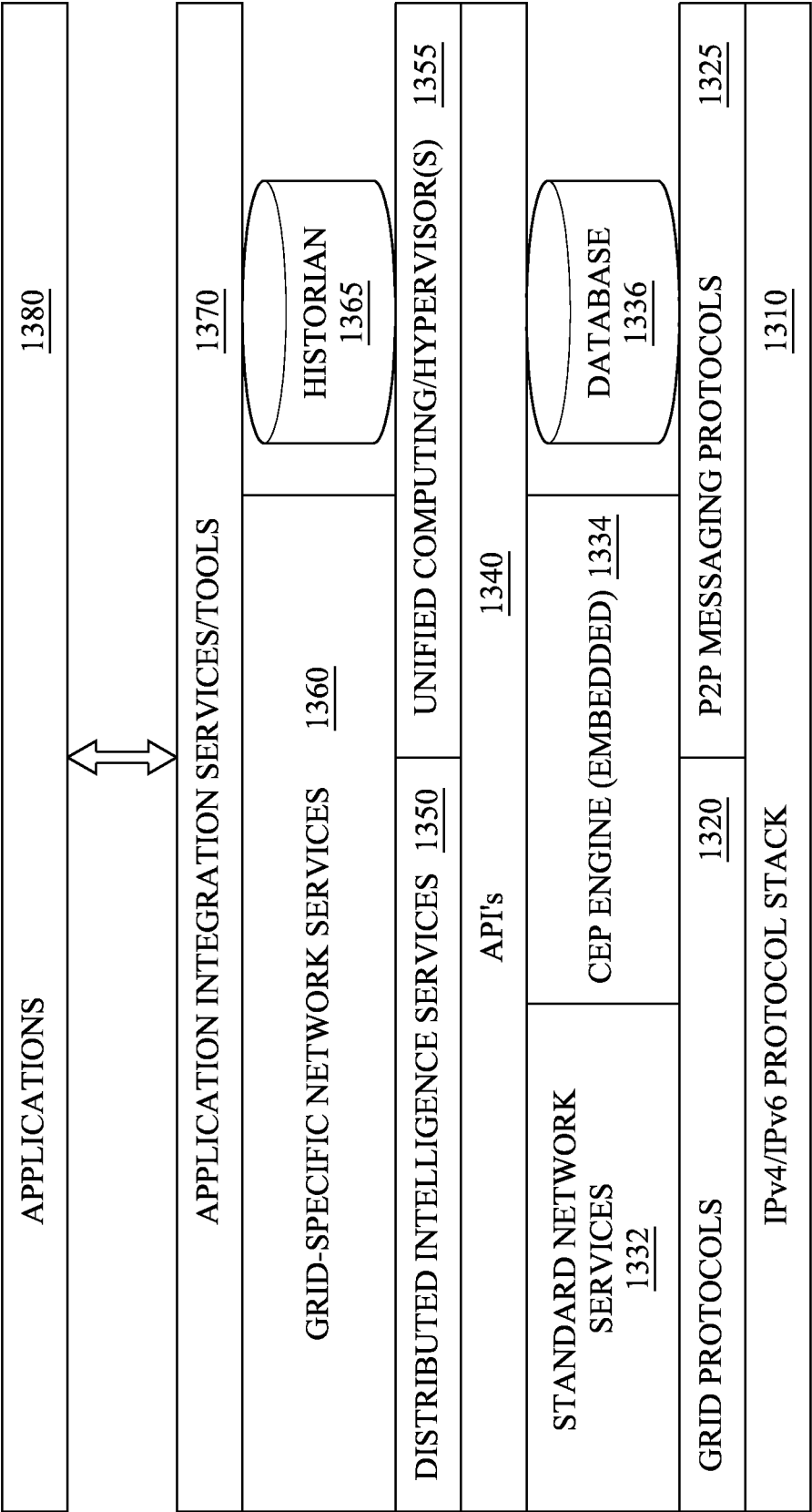


FIG. 13

+

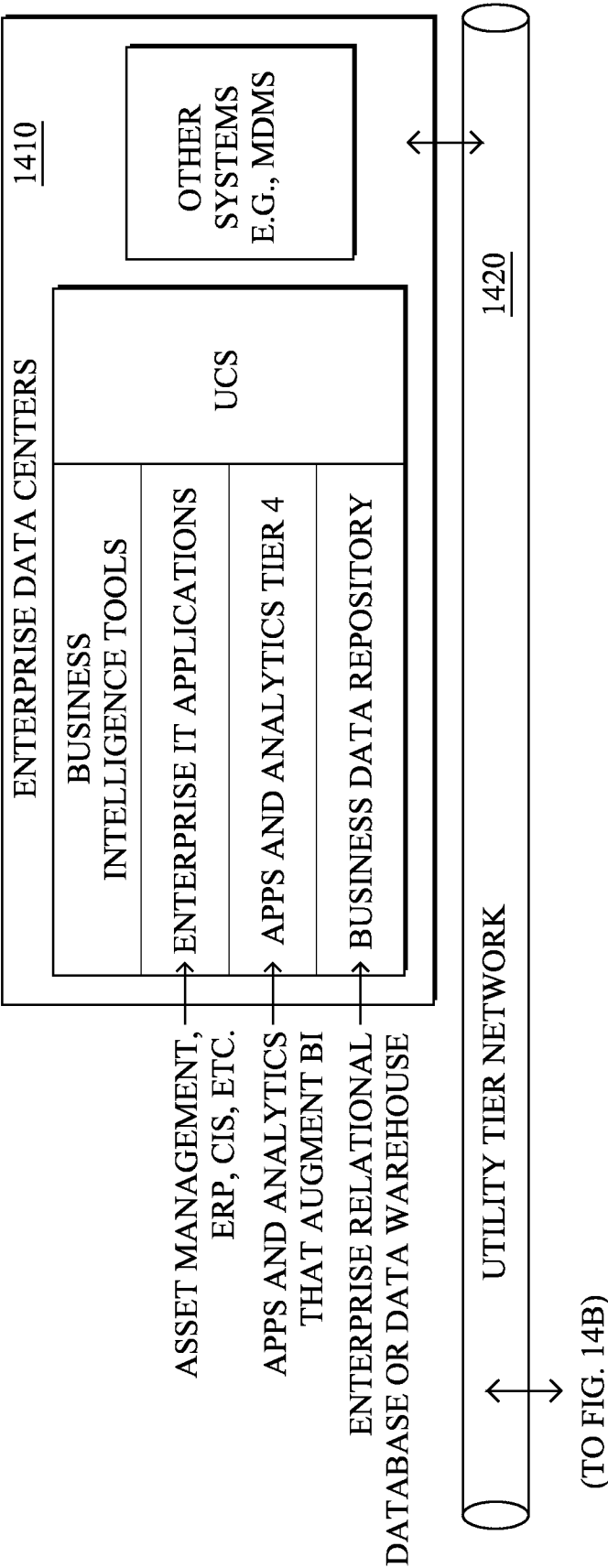
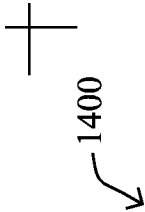


FIG. 14A



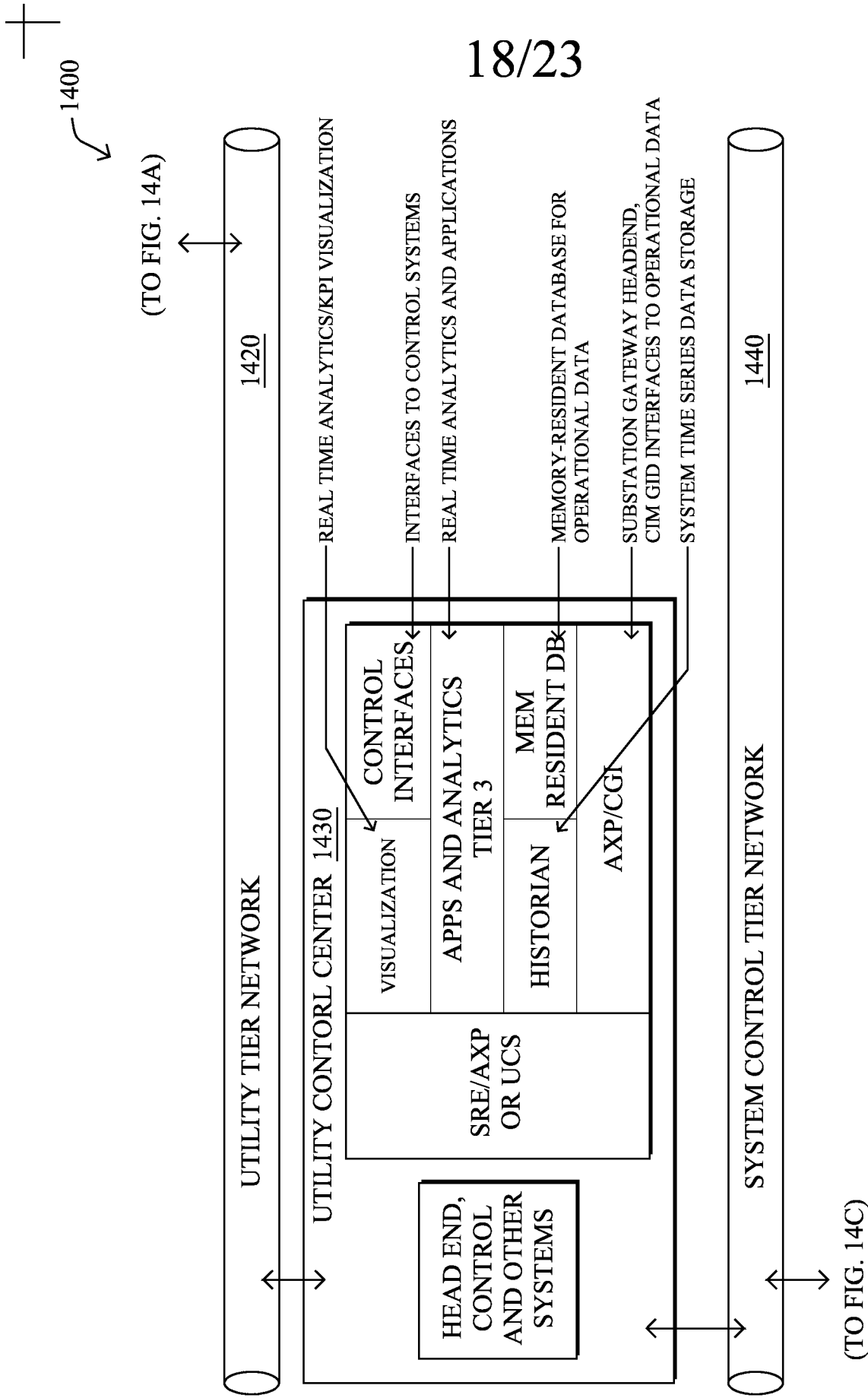


FIG. 14B

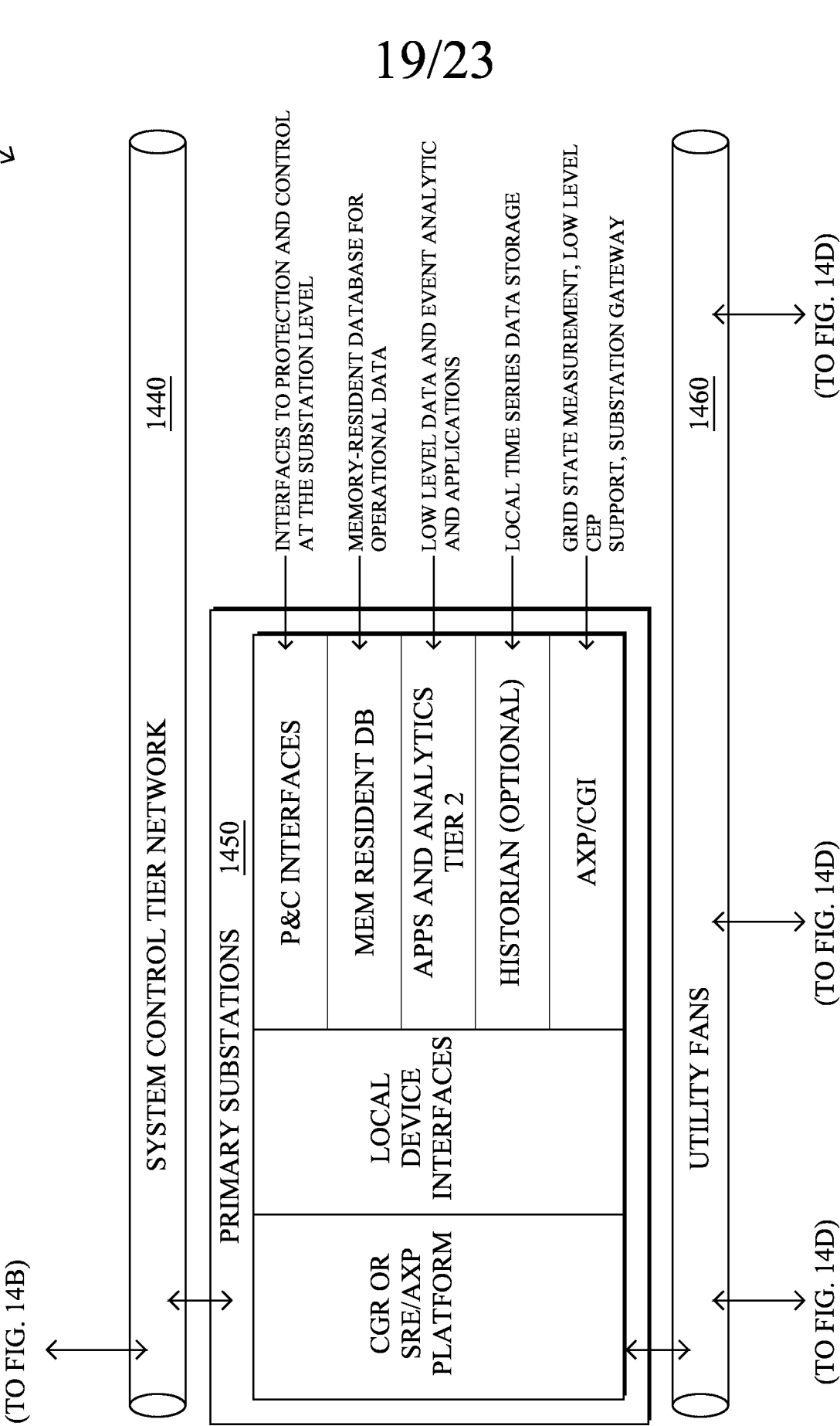


FIG. 14C

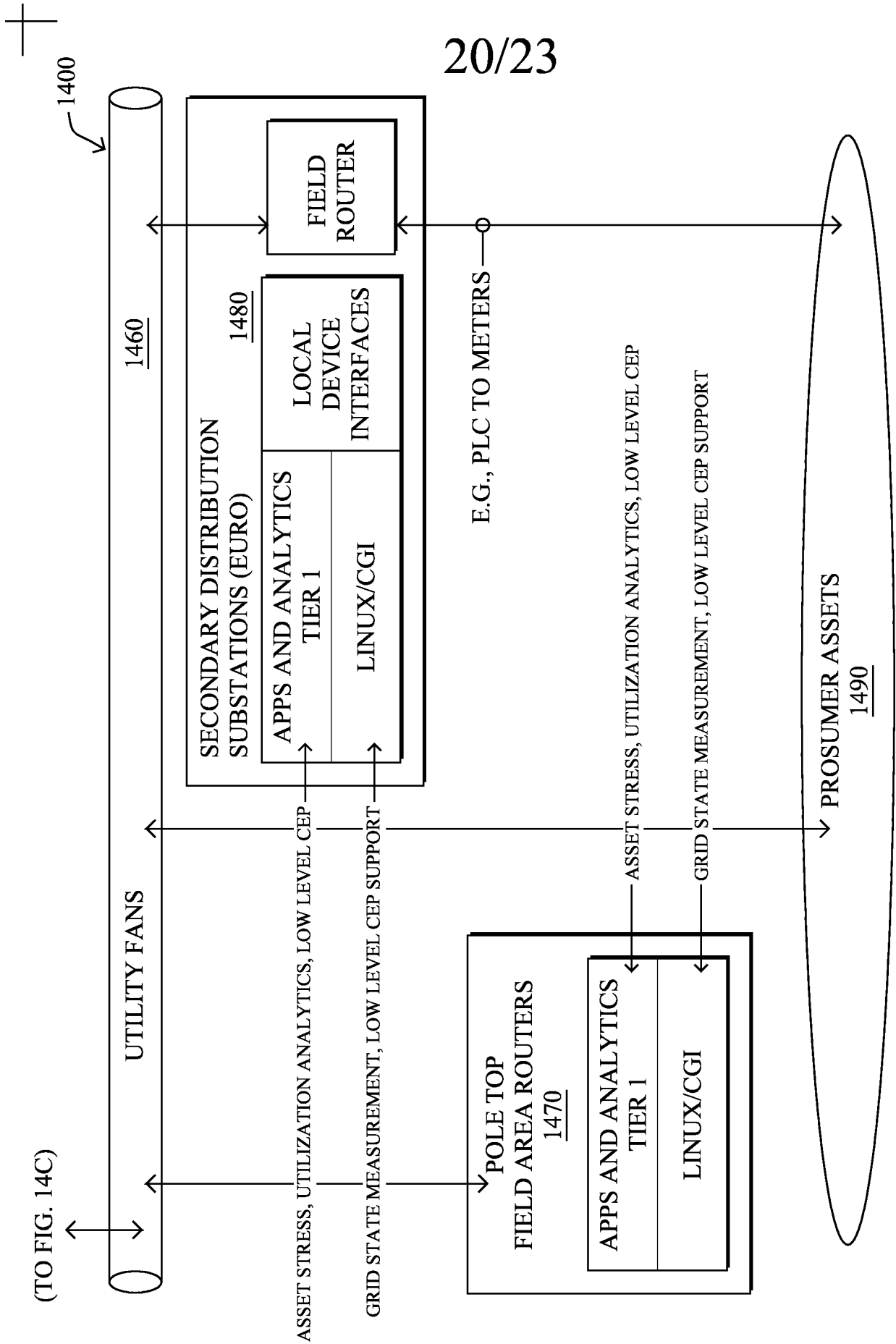
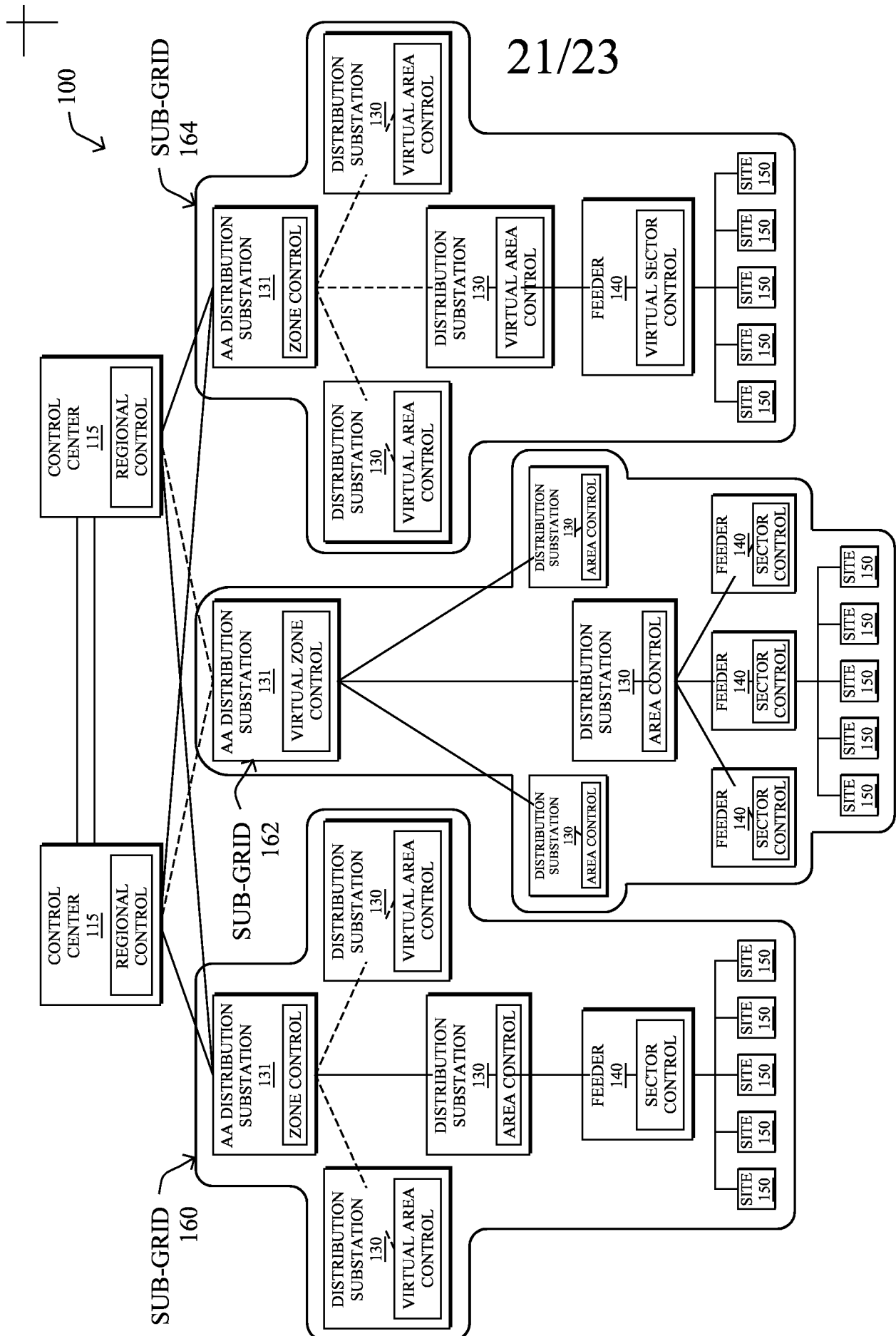


FIG. 14D



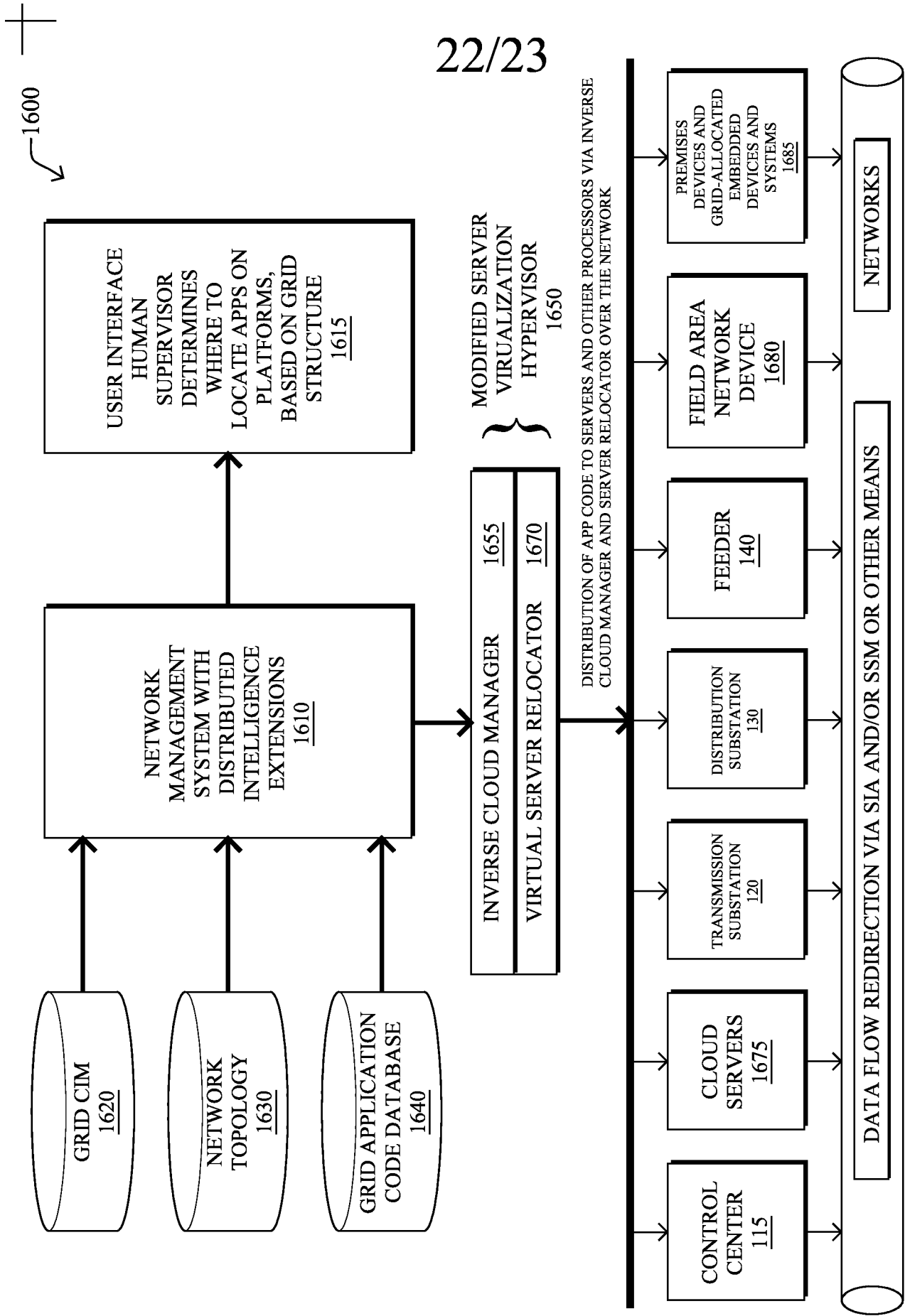


FIG. 16

23/23

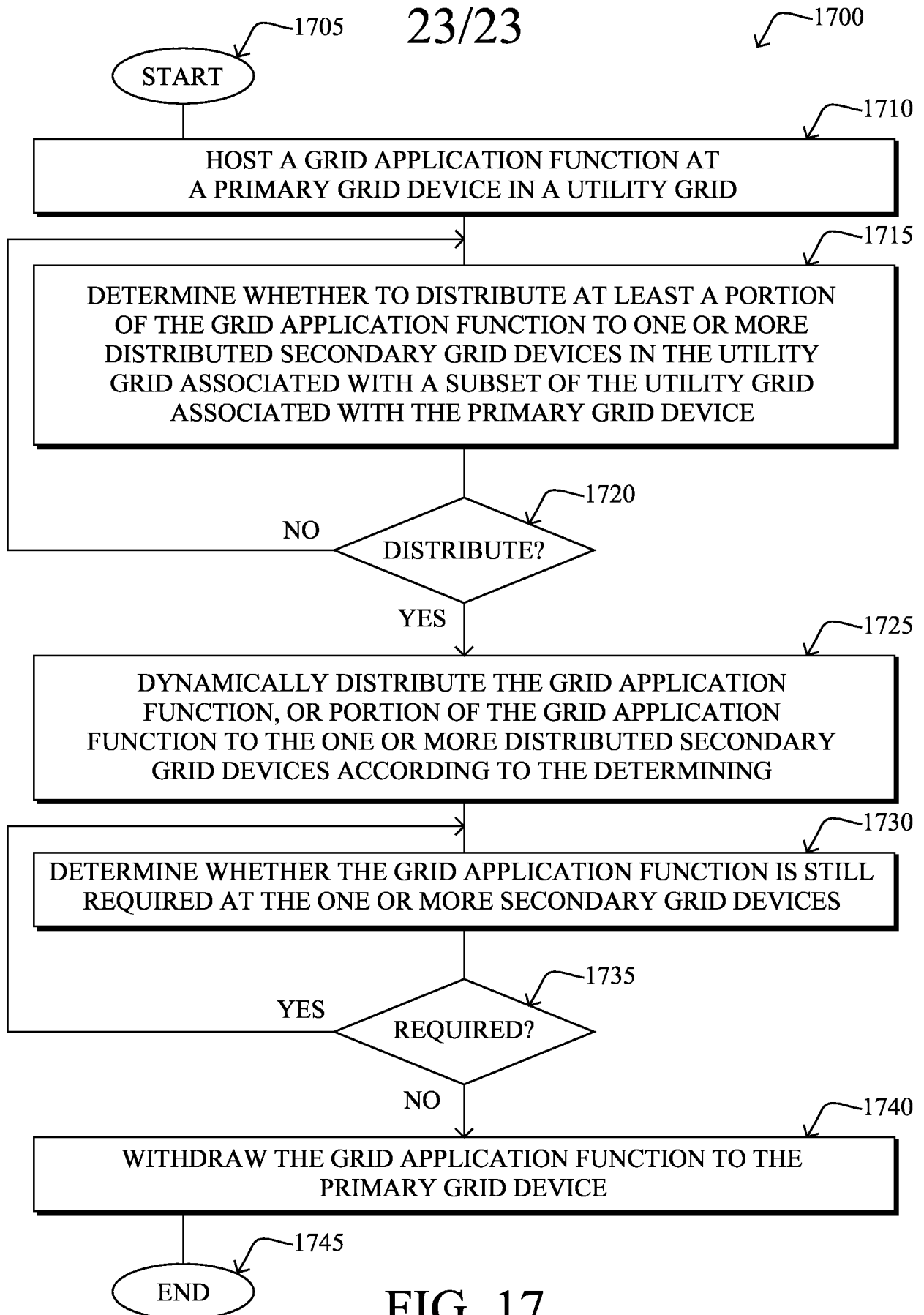


FIG. 17