



(51) International Patent Classification:

G05B 13/02 (2006.01) G06F 3/01 (2006.01)
G06K 9/20 (2006.01) G06F 17/30 (2006.01)
G06T 7/20 (2017.01) G06N 5/04 (2006.01)
B23Q 35/12 (2006.01)

(21) International Application Number:

PCT/US2017/018495

(22) International Filing Date:

17 February 2017 (17.02.2017)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/297,621 19 February 2016 (19.02.2016) US
15/261,136 9 September 2016 (09.09.2016) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:

US 15/261,136 (CON)
Filed on 9 September 2016 (09.09.2016)

(71) Applicant: GENERAL ELECTRIC COMPANY
[US/US]; 1 River Road, Schenectady, New York 12345 (US).

(72) Inventors: WRAGG, Martin; 2623 Camino Ramon, San Ramon, California 94583 (US). HERNANDEZ, Carlos; 2623 Camino Ramon, San Ramon, California 94583 (US).

(74) Agents: SCHEER, Bradley W. et al.; Schwegman, Lundberg & Woessner, P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

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

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

Published:

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

(54) Title: MANAGING AND PRESENTING DATA FOR INDUSTRIAL ASSETS

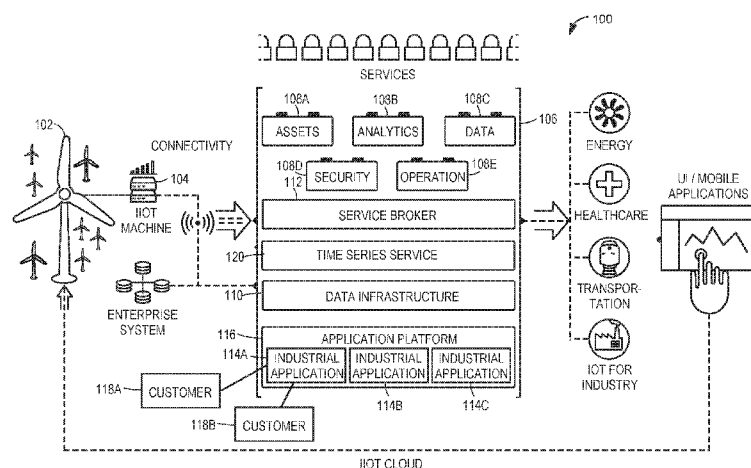


FIG. 1

(57) **Abstract:** In example embodiments, a method of presenting time series data in a user interface is disclosed. A time series data component is embedded in a user interface of an application executing on a device. Time series data corresponding to an asset in an industrial internet of things (IIoT) is presented by the time series data component based on a context of a user of the device. Customizations pertaining to the presenting of the time series data are received. The customizations include annotations relating to the time series data. A snapshot of the time series data and the customizations is shared. Upon an accessing of the shared snapshot from an additional time series data component, the snapshot is presented by the additional time series data component based on a combination the context of the user of the device and a context of a user of an additional device.

MANAGING AND PRESENTING DATA FOR INDUSTRIAL ASSETS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority to U.S. Application No. 15/261,136, filed September 9, 2016, and to U.S. Provisional Application No. 62/297,621, filed February 19, 2016, each of which is incorporated by reference herein in its entirety.

TECHNICAL FIELD

[0002] This application relates generally to graphical user interfaces and, in one specific example, to a graphical user interface component for presenting time series data pertaining to assets in an Industrial Internet of Things (IIoT) and managing metadata corresponding to the time series data.

BACKGROUND

[0003] The traditional Internet of Things (IoT) involves the connection of various consumer devices, such as coffee pots and alarm clocks, to the Internet to allow for various levels of control and automation of those devices. The Industrial Internet of Things (IIoT), on the other hand, involves connecting industrial assets as opposed to consumer devices. There are technical challenges involved in interconnecting diverse industrial assets, such as wind turbines, jet engines, and locomotives, that do not exist in the realm of consumer devices.

BRIEF DESCRIPTION OF DRAWINGS

[0004] The present disclosure is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0005] FIG. 1 is a block diagram illustrating a system, in accordance with an example embodiment, implementing an IIoT.

[0006] FIG. 2 is a block diagram illustrating different edge connectivity options that an IIoT machine provides, in accordance with an example embodiment.

[0007] FIG. 3 is an example method of managing metadata pertaining to time series data using a time series component;

[0008] FIG. 4 depicts an example embodiment of a user interface of a time series component.

[0009] FIG. 5 depicts another example embodiment of a user interface of a time series component.

[0010] FIG. 6 depicts another example embodiment of a user interface of a time series component.

[0011] FIG. 7 depicts another example embodiment of a user interface of a time series component.

[0012] FIG. 8 depicts another example embodiment of a user interface of a time series component.

[0013] FIG. 9 depicts another example embodiment of a user interface of a time series component.

[0014] FIG. 10 depicts another example embodiment of a user interface of a time series component.

[0015] FIG. 11 is a block diagram illustrating a representative software architecture which may be used in conjunction with various hardware architectures herein described.

[0016] FIG. 12 is a block diagram illustrating components of a machine, according to some example embodiments, able to read instructions from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein.

DETAILED DESCRIPTION

[0017] The description that follows includes illustrative systems, methods, techniques, instruction sequences, and machine-readable media (e.g., computing machine program products) that embody illustrative embodiments. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide an understanding of various embodiments of the inventive subject matter. It will be evident, however, to those skilled in the art that embodiments of the inventive subject matter may be practiced without these specific details. In general, well-known instruction instances, protocols, structures, and techniques have not been shown in detail.

[0018] Some of the technical challenges involved in an IIoT include items such as predictive maintenance, where industrial assets can be serviced prior to problems developing to reduce unplanned downtimes. As such, one such

technical challenge involves prediction of when industrial assets or parts thereon will fail. In an example embodiment, an IIoT may be designed that monitors data collected from sensors and, using physics-based analytics, detects potential error conditions based on an asset model. The asset in question can then be gracefully shut down for maintenance at the appropriate time. In addition to these types of edge applications (applications involving the industrial assets directly), the IIoT may also pass the sensor data to a cloud environment where operational data for all similar machines under management can be stored and analyzed. Over time, data scientists can discover new patterns and create new and improved physics-based analytical models. The new analytical model can then be pushed back to all of the assets, effectively improving the performance of all assets simultaneously.

[0019] In example embodiments, a method of presenting time series data in a user interface is disclosed. A time series data component (or widget) is embedded in a user interface of an application executing on a device. Time series data corresponding to an asset in an industrial internet of things (IIOT) is presented by the time series data component based on a context of a user of the device. Customizations pertaining to the presenting of the time series data are received. The customizations include annotations relating to the time series data. A snapshot of the time series data and the customizations is shared. Upon an accessing of the shared snapshot from an additional time series data component embedded in an additional user interface of an additional application executing on an additional device, the snapshot is presented in the additional user interface based on a combination of the context of the user of the device and the context of the user of the additional device.

[0020] FIG. 1 is a block diagram illustrating a system 100, in accordance with an example embodiment, implementing an IIoT. An industrial asset 102, such as a wind turbine as depicted here, may be directly connected to an IIoT machine 104. The IIoT machine 104 is a software stack that can be embedded into hardware devices such as industrial control systems or network gateways. The software stack may include its own software development kit (SDK). The SDK includes functions that enable developers to leverage the core features described below.

[0021] One responsibility of the IIoT machine 104 is to provide secure, bi-directional cloud connectivity to, and management of, industrial assets, while also enabling applications (analytical and operational services) at the edge of the IIoT. The latter permits the delivery of near-real-time processing in controlled environments. Thus, the IIoT machine 104 connects to an IIoT cloud 106, which includes various modules, including asset module 108A, analytics module 108B, data module 108C, security module 108D, and operations module 108E, as well as data infrastructure 110. This allows other computing devices, such as client computers running user interfaces/mobile applications, to perform various analyses of either the individual industrial asset 102 or assets of the same type.

[0022] The IIoT machine 104 also provides security, authentication, and governance services for endpoint devices. This allows security profiles to be audited and managed centrally across devices, ensuring that assets are connected, controlled, and managed in a safe and secure manner, and that critical data is protected.

[0023] In example embodiments, a service broker 112 may deliver various services, including asset services 108A, analytics services 108B, data services 108C, security services 108D, operational services 108E, connectivity services, platform services, infrastructure services, and so on. The connectivity services may provide a managed, secure, end-to-end connectivity solution from the edge of a customer's network to the IIoT cloud. Such connectivity services may include providing physical connectivity globally (e.g., via cellular, fixed, or satellite networks), a secure virtual private network (VPN) between edge assets and the IIoT cloud to help ensure data privacy and asset protection, an ability to manage and control edge assets by providing remote access (e.g., via VNC, RDP, SSH, and HTTP), end-to-end monitoring and notifications about the connectivity between the IIoT cloud and edge assets, one-stop-shop billing and reporting for all connectivity and IP services, and a self-management portal.

[0024] An application platform 116 supports the building of responsive web, mobile, and embedded applications that scale gracefully from smart phone to desktop. The user interface system provides developers and designers with simple, modular, cohesive solutions for theming, layout, and UI components

with tailored integration points into the rest of the platform stack. Applications, such as industrial applications 114A-114C, are not only context-aware, but also context-adaptive, meaning they will change according to the context, so users can visualize and interface with the application in a way that is relevant to them. This paradigm removes the need for multiple applications and context switching by users.

[0025] In order to meet requirements for industrial connectivity, the IIoT machine 104 can support gateway solutions that connect multiple edge components via various industry standard protocols. FIG. 2 is a block diagram illustrating different edge connectivity options that an IIoT machine 104 provides, in accordance with an example embodiment. There are generally three types of edge connectivity options that an IIoT machine 104 provides: machine gateway (M2M) 202, cloud gateway (M2DC) 204, and mobile gateway (M2H) 206.

[0026] Many assets may already support connectivity through industrial protocols such as Open Platform Communication (OPC)-UA or ModBus. A machine gateway component 208 may provide an extensible plug-in framework that enables connectivity to assets via M2M 202 based on these common industrial protocols.

[0027] A cloud gateway component 210 connects an IIoT machine 104 to an IIoT cloud 106 via M2DC 204.

[0028] A mobile gateway component 212 enables people to bypass the IIoT cloud 106 and establish a direct connection to an asset 102. This may be especially important for maintenance scenarios. When service technicians are deployed to maintain or repair machines, they can connect directly from their machine to understand the asset's operating conditions and perform troubleshooting. In certain industrial environments, where connectivity can be challenging, the ability to bypass the cloud and create this direct connection to the asset may be important.

[0029] As described above, there are a series of core capabilities provided by the IIoT system 100. Industrial scale data, which can be massive and is often

generated continuously, cannot always be efficiently transferred to the cloud for processing, unlike data from consumer devices. Edge analytics provide a way to preprocess the data so that only the pertinent information is sent to the cloud. Various core capabilities provided include file and data transfer, store and forward, local data store and access, sensor data aggregation, edge analytics, certificate management, device provisioning, device decommissioning, and configuration management.

[0030] As described above, the IIoT machine 104 can be deployed in various different ways. These include on a gateway, on controllers, or on sensor nodes. The gateway acts as a smart conduit between the IIoT cloud 106 and the asset(s) 102. The IIoT machine 104 may be deployed on the gateway device to provide connectivity to asset(s) 102 via a variety of protocols.

[0031] The IIoT machine 104 can be deployed directly onto machine controller units. This decouples the machine software from the machine hardware, allowing connectivity, upgradability, cross-compatibility, remote access, and remote control. It also enables industrial and commercial assets that have traditionally operated standalone or in very isolated networks to be connected directly to the IIoT cloud 106 for data collection and live analytics.

[0032] The IIoT machine 104 can be deployed on sensor nodes. In this scenario, the intelligence lives in the IIoT cloud 106 and simple, low-cost sensors can be deployed on or near the asset(s) 102. The sensors collect machine and environmental data and then backhaul this data to the IIoT cloud 106 (directly or through an IIoT gateway), where it is stored, analyzed, and visualized.

[0033] Customers or other users may create applications to operate in the IIoT cloud 106. While the applications reside in the IIoT cloud 106, they may rely partially on the local IIoT machines 104 to provide the capabilities to gather sensor data, process it locally, and then push it to the IIoT cloud 106.

[0034] The IIoT cloud 106 enables the IIoT by providing a scalable cloud infrastructure that serves as a basis for platform-as-a-service (PaaS), which is what developers use to create Industrial Internet applications for use in the IIoT cloud 106.

[0035] Referring back to FIG. 1, services provided by the IIoT cloud 106 and generally available to applications designed by developers include asset services from asset module 108A, analytics services from analytics module 108B, data services from data module 108C, application security services from security module 108D, and operational services from operations module 108E.

[0036] Asset services include services to create, import, and organize asset models and their associated business rules. Data services include services to ingest, clean, merge, and ultimately store data in the appropriate storage technology so that it can be made available to applications in the manner most suitable to their use case.

[0037] Analytics services include services to create, catalog, and orchestrate analytics that will serve as the basis for applications to create insights about industrial assets. Application security services include services to meet end-to-end security requirements, including those related to authentication and authorization.

[0038] Operational services enable application developers to manage the lifecycle and commercialization of their applications. Operational services may include development operational services, which are services to develop and deploy Industrial Internet applications in the cloud, as well as business operational services, which are services that enable transparency into the usage of Industrial Internet applications so that developers can ensure profitability.

[0039] The asset model may be the centerpiece of many, if not all, Industrial Internet applications. While assets are the instantiations of asset types (types of industrial equipment, such as turbines), the asset model is a digital representation of the asset's structure. In an example embodiment, the asset service provides Application Program Interfaces (APIs), such as Representational State Transfer (REST) APIs that enable application developers to create and store asset models that define asset properties, as well as relationships between assets and other modeling elements. Application developers can then leverage the service to store asset-instance data. For example, an application developer can create an asset model that describes the logical component structure of all turbines in a wind farm and then create instances of that model to represent each individual

turbine. Developers can also create custom modeling objects to meet their own unique domain needs.

[0040] In an example embodiment, the asset module 108A may include an API layer, a query engine, and a graph database. The API layer acts to translate data for storage and query in the graph database. The query engine enables developers to use a standardized language, such as Graph Expression Language (GEL), to retrieve data about any object or property of any object in the asset service data store. The graph database stores the data.

[0041] An asset model represents the information that application developers store about assets, how assets are organized, and how they are related. Application developers can use the asset module 108A APIs to define a consistent asset model and a hierarchical structure for the data. Each piece of physical equipment may then be represented by an asset instance. Assets can be organized by classification and by any number of custom modeling objects. For example, an organization can use a location object to store data about where its pumps are manufactured, and then use a manufacturer object to store data about specific pump suppliers. It can also use several classifications of pumps to define pump types, assign multiple attributes, such as Brass or Steel, to each classification, and associate multiple meters, such as Flow or Pressure, to a classification.

[0042] Data services from the data module 108C enable Industrial Internet application developers to bring data into the IIoT system 100 and make it available for their applications. This data may be ingested via an ingestion pipeline that allows for the data to be cleansed, merged with data from other data sources, and stored in the appropriate type of data store, whether it be a time series data store for sensor data, a Binary Large Object (BLOB) store for medical images, or a relational database management system (RDBMS).

[0043] Since many of the assets are industrial in nature, much of the data that will commonly be brought into the IIoT system 100 for analysis is sensor data from industrial assets. In an example embodiment, a time series service 120 may provide a query-efficient columnar storage format optimized for time series data. As the continuous stream of information flows from sensors and needs to be

analyzed based on the time aspect, the arrival time of each stream can be maintained and indexed in this storage format for faster queries. The time series service 120 also may provide the ability to efficiently ingest massive amounts of data based on extensible data models. The time series service 120 capabilities address operational challenges posed by the volume, velocity, and variety of IIoT data, such as efficient storage of time series data, indexing of data for quick retrieval, high availability, horizontal scalability, and data point precision.

[0044] The application security services provided by the security module 108D include user account and authentication (UAA) and access control. The UAA service provides a mechanism for applications to authenticate users by setting up a UAA zone. An application developer can bind the application to the UAA service and then use services such as basic login and logout support for the application, without needing to recode these services for each application. Access control may be provided as a policy-drive authorization service that enables applications to create access restrictions to resources based on a number of criteria.

[0045] Thus, a situation arises where application developers wishing to create industrial applications for use in the IIoT may wish to use common services that many such industrial applications may use, such as a log-in page, time series management, data storage, and the like. The way a developer can utilize such services is by instantiating instances of the services and then having their applications consume those instances. Typically, many services may be so instantiated.

[0046] A service may communicate (e.g., via a specific API) with a user interface component. For example, a component may be created with a specific API that talks to a service. The component may have its own generic API and generic data format. In this case, an adapter or transformer may be used by the UI component to transform the service data format and protocol and API into something that the component can communicate with. These adapters can themselves be components or attached reusable behaviors. The services and component may be included in or provided as part of a user interface (UI) platform, such as General Electric (GE) Digital's Predix platform. The UI

platform may be framework agnostic. The platform provides functions, capabilities, and a default set of components that may be used with other frameworks, without a framework (or just the DOM framework itself). The UI platform may also be used to construct frameworks. The UI platform may be analogized to a bunch of Legos and building blocks that are used to build bigger and bigger components. For example, in user interface (e.g., web) application and system design, the building blocks are pieces that may be assembled together into pages, applications, multiple applications, and eventually an entire system.

[0047] A time series component may be one of a set of higher-order components that has been built from different components. In various embodiments, the time series component may present or visualize time series data in a graphical user interface. The time series component may allow users to interact with time series data in various ways, such as by zooming in or out, overlaying sets of time series data on top of each other, customizing the presentation of the time series data (e.g., size, colors, format, etc.) for a particular application, and so on. Using various such tools provided by the time series component, a user of the component may be able to identify patterns or outliers in different sets of time series data.

[0048] In various embodiments, a higher-order application component in which the user interface component is included may be configured (e.g., programmatically) to compare different sets of time series data presented in multiple time series components (or multiple instances of the component) to identify anomalies based on analytics, including overlays of raw time series data or calculations pertaining to the raw time series data, to identify anomalies (e.g., based on a statistical variation exceeding a threshold value).

[0049] In various embodiments, the component is customizable programmatically by higher-order components and by a user of the component. For example, the component may be enabled by a developer to allow the user to associate annotations with the time series data being presented in the component. In various embodiments, all aspects of the component, such as colors used, fonts used, thickness of lines used, and so on, are configurable by the user. In various

embodiments, the aspects of the component are automatically adjusted programmatically based on a context of the user, as described in more detail below. For example, if the user moves into a darker environment, the component may be programmed to automatically become brighter.

[0050] A time series component may be included in a card. In various embodiments, a card is another higher-order component that represents a self-contained unit of work within an application or workflow. In various embodiments, a user may interact with cards, which together form a user interface experience, and there is a purpose associated with each card. For example, in an email application, an inbox user interface may be implemented on a first card and a details section corresponding to an individual selected email may be implemented on another card. The cards are tied together in a workflow that may be saved by the user, capturing a snapshot in time. The saved state may be shared with other users, but presented in a different manner to each user based on each user's context. Users may be alerted that some data cannot be presented to them based on their security level. In other cases, the user may not be notified that some data is not being presented to the user, such that the end user does not even know that the other data is there. The time series component itself may be saved as well, either individually or as part of a saving of a card on which it is included. In various embodiments, when the time series component is saved, the customizable aspects pertaining to the presentation of the time series data within the component are saved along with metadata pertaining to the time series data. The metadata may include annotations entered by the user. The metadata may also include a context of the user. The context of the user may include a security level of the user, a role of the user, a type of machine used by the user, or any other data pertaining to the user's context, as described herein. The snapshot of the data may then be shared such that another user may view the snapshot. Depending on various configuration options, the time series data may present the data in the other (e.g., viewing) user's context, as explained in more detail below. In various embodiments, depending on the other user's context, some data may be hidden from the user or presented in a different way within the user interface component accessed by the other user. For example, if the other user does not have security permissions that are equal or greater to the

user who saved the snapshot, the other user may not have access to the same data or be able to view the data in the same way.

[0051] In various embodiments, a context may include information pertaining to an asset or set of assets that are currently active (e.g., in a user interface) to a user as well as data pertaining to the user, including a profile of the user, an environment of the user, and the way in which the assets are being represented to the user in the user interface component. The information may include security access levels, roles, permissions, system state, device modality, interaction types, and other things that make up the context in which the user is using the component.

[0052] A time series component may be one of a set of time series components on a card. The set may have a primary time series component. Each component may have time series data belonging to an asset, an asset type, or an asset sub-type (e.g., asset classification or asset tag). The master component may control the other components. In various embodiments, the control of the master component over the other components allows the user to make a change to the way in which data is viewed in the master component, which, in turn, automatically affects the way data is displayed in the other components. As an example, a user could select a portion of time in a master component and have the data in that portion automatically compared with data in the other components at another specified interval of time. Moving controls associated with the master component may automatically move controls associated with other components. For example, setting date ranges within multiple components may cause them to move synchronously or do analytics between components (e.g., additive, multiplicative, or subtractive analytics). In this way, the process of identifying anomalies (e.g., in the startup or operation of an asset or set of assets) with a certain amount of deviation may be automated or simplified. The component may handle smoothing and interpolation in the presentation of the time series data to help with diagnostics and prognostics pertaining to a corresponding asset or a set of assets.

[0053] The component is configured to be deployed (e.g., as part of a user interface of an application) on any of a number of types of devices. Examples

include a mobile client device, such as a mobile phone, a tablet, a PC, or a set of machines in a control center having multiple monitors. In the control center environment, multiple components or instances of a component may be configured such that each monitor shows different types of information for different analyses, such as a single asset at multiple different points in time or multiple assets at a current (e.g., real) time.

[0054] In various embodiments, raw time series data (e.g., including data received from sensors associated with an asset) may go through a pre-processing step before being ingested by the component. For example, a developer of an application may include multiple other components on a card, along with the time series component, to handle tasks such as translating or transforming or adapting the raw time series data for each of a number of different assets into a common time series data format or calculating various parameters (e.g., using formulas pertaining to the raw time series data) for providing as additional time series data in addition to the raw time series data. The pre-processing logic may be included in these other components (e.g., for translation or other pre-processing) included in the modular design or included in the logic of a higher-order component containing the component, such as a card, deck (i.e., a set of cards), or application component. Thus, for example, the card in which the component is included may perform pre-processing of raw time series sensor data to provide normalization, deltas, differences, or other analytics on the raw data. Decks may have a common user interface theme, logic, or be part of a larger workflow module. In various embodiments, decks are stored as a higher-order component. A deck may have many cards; a card may belong to many decks; hence, decks and cards may have a many-to-many relationship. Depending on the context, a user may not be able to see every card or visualize all cards in the same manner as other users. Thus, a deck may be context-aware, context-sensitive, and context-adaptive. In various embodiments, multiple decks can be grouped together to create views. And many views can be grouped together to create a page.

[0055] Each of the components may register with the card and receive information from the card concerning other components. The components may communicate various ways in which the presentation of their data has changed to

the other components, and the other components may be notified of and respond to the changes. In various embodiments, the component may be a dummy consumer of the data, based on a specification, and have no responsibility to communicate changes to the data, leaving a separate component to handle any communications. For example, a communication component may be dropped onto a card containing the time series component and a time series data format adapter. A main communication component may be integrated with an adapter component in the card or deck level to provide the communication separate from the component. The components may leverage the web DOM/browser as the UI framework, thus keeping the framework agnostic of any specific implementation, such as JavaScript frameworks like Angular, Knockout, and Ember. As an example, each DOM component in HTML, such as TABLE or a DIV element, may have an associated Polymer element.

[0056] In various embodiments, a set of assets is accessible through the IIoT. The sensor data is received at a high-order level, such as a deck, which then communicates to children, such as cards or components on the cards. Adapters incorporated into the cards may then be connected to the time series data sources. There may be data stored in different formats by different assets. The format may be transformed by adapters into a common data format by modular components, added as needed by a developer of an application using the UI platform. In other words, an adapter specific to an asset may be included on a card as needed.

[0057] The time series data component may be context aware and context adaptive. For example, the component may automatically adapt to a user situation in the field, on the factory floor, or at home. The time series data component may be configured to show each user the relevant data in the right interaction mode (e.g., based on the context of the user). The differences may be as subtle as the lines needing to be thicker or bolder for display in a large monitor in a factory environment than they are when they are displayed on a mobile device to a field engineer. The field user may be presented with a set of actions relevant to the field user, such as touch-screen buttons for logging the current status of an asset being monitored in the field. The control center user may be alerted of any anomalies. In various embodiments, an analyst may

identify anomalies and save them in a file for attaching to a case report as a summary of an incident. The saved state may be shared with other users, but presented in a different manner to each user based on each user's context. Users may be alerted that some data cannot be presented to them based on their security level. In other cases, the user may not be notified that some data is not being presented to the user, such that the end user does not even know that the other data is there.

[0058] As an example, an analyst may look at multiple (e.g., 10 sets) of time series data overlaid in various ways. An anomaly may be detected based on the analytics performed. The analyst (e.g., working in an "analyst" or "reporting" context) may then annotate the sets of time series data and save a snapshot for inclusion as an attachment to a case (e.g., for reporting purposes). The viewer of the report (e.g., working in a "viewer" or "decision-making" context) may not be interested in data other than the particular overlay showing the anomaly. Thus, when presented with the snapshot saved by the analyst, the time series data component may not present all of the multiple overlays used by the analyst to discover the anomaly. Instead, if the viewer is a decision maker, for example, the specific overlay(s) used to discover the anomaly may be highlighted or a smoothing of an aggregation of the multiple sets of time series data may be presented. Or, if the viewer is a field engineer (e.g., working in a "field" context), data pertaining to monitoring of the asset (e.g., time and place of likely reoccurrence) may be presented. In example embodiments, when a user requests an asset, they are also requesting the context of that asset (which in turn gets combined with the user's context to form a global context for an application). This global context is used by the platform and end application in an adaptive and responsive way. The application will "transform" according to the currently selected context or contexts. As an example, a user browses for Asset A with Context AC. The user has a general user context UC. The application responds to this by showing only cards that available for Asset A and Context AC-intersected-with-UC. A card may have visualization components inside of it that allow User U to work with Asset A and analyze its data. The visualization components may present different ways of visualizing an asset hierarchy, including Miller columns or graph-like visualizations (e.g., for multi-relationship

asset-visualization that does not neatly fall into a hierarchy). In example embodiments, visual cues (e.g., icons) are used to classify assets (e.g., by type), making it easier for a user to quickly scan the hierarchy for relevant information. Given user context UC, the user may only have access to certain data stream or certain parts of the asset tree, and will therefore only see the parts that they have access to.

[0059] As an example, on startup of a wind turbine, there is an expected startup curve (e.g., with respect to various parameters, such as temperature or vibration). An analysis of the time series data may include taking overlays of recent (e.g., the most recent 20) startups of a single asset or of multiple similar assets, thus placing more value on data from more recent startups. If an analysis of the time series data shows there is a statistical deviation (e.g., with respect to vibration) for a particular asset, the time series data component may then present a notification of a detection, based on threshold settings, of an anomaly pertaining to an asset. The alert may be presented (or not) in the user interface depending on developer customizable settings. A presentation a snapshot of the anomaly in the time series component included in a user interface of an application executing on a device of a field engineer may provide specific information pertaining to the asset and a marker in the time series data showing the anomaly (e.g., a spike in vibration). The field engineer would then have helpful information to monitor and troubleshoot the asset. For example, the field engineer could restart the asset and troubleshoot the asset, focusing at the identified point in time of the anomaly. The data may be viewed in real-time by the field engineer and a control center analyst at the same time, each working in different contexts and thus being presented with different, context-optimized views of the same data. An annotation stream from the field engineer may be synchronized with event data and other metadata for viewing along with the corresponding time series data at the control center. In example embodiments, a set of industrial assets within a geographical range from the field engineer may be populated in an industrial asset browser component (e.g., corresponding to a time series data component) such that the field engineer may easily navigate to information pertaining to one or more industrial assets within a user interface presented on a device. In example embodiments, the industrial asset browser

component may allow a user to rank or filter results using a context of the user, such as the GPS location of the user relative to the industrial assets or information pertaining to one or more work tasks that the user is assigned pertaining to the industrial assets (e.g., startup, shutdown, maintenance, reporting, and so on), as a starting point for navigation of a tree (or hierarchy) of available industrial assets.

[0060] In various embodiments, a slider control allows a user to move back and forth in the startup times for several assets, whereas an additional component would show a history of startup times for multiple assets. Thus, time series data over multiple time series may be grouped, as explained in more detail below.

[0061] In various embodiments, a time series API provides a developer with programmatic access and control over a time series component. The API may define various elements, such as JavaScript Object Notation (JSON) elements, for language-independent data interchange of objects comprising attribute-value pairs. For example, such elements may include the following example elements:

[0062] start. start is the start time for a query window. It can either be expressed as a timestamp integer value in milliseconds, or it can be a relative time (<value><time-unit>-ago). i.e. "start" : 1427463525000 or "start" : "12h-ago" Supported Time-Units may include: ms – Milliseconds, s – Seconds, mi – Minutes, h – Hours, d - Days (24 hours), w - Weeks (7 days), mm - Months (30 days), y - Years (365 days).

[0063] end. end is the end time for query window. It can either be expressed as a timestamp integer value in milliseconds, or it can be a relative time (<value><time-unit>-ago). i.e. "end" : 1427463525000 or "end" : "12h-ago". If this is not defined it will default to the current time. Supported Time-Units, as mentioned in "start" element.

[0064] tags. tags in the object with all the query results and query stats for that particular tag name.

[0065] name. name is the tag name for the following results array.

[0066] `limit`. limits the sample size of the query. This limit is pre-aggregation and is after the quality filtering. See `filter:quality` for details on how quality is filtered.

[0067] `order`. order defines the order of data points in each result. The options are either "desc" (descending order by time stamp) or "asc" (ascending order by timestamp). If this element is not defined, it defaults to ascending order.

[0068] `aggregations`. aggregations are the calculations and samplings that can be performed on the data within the time window. A list of available aggregations can be found at [/v1/aggregations](#). By default aggregations occur on only good data unless `filter:qualities` is defined differently.

[0069] `aggregations:type`. aggregation type is the name of the aggregation that will be performed on the data points in the chosen time window.

[0070] `aggregations:interval/aggregations:count`. aggregation interval is the bucketing of time intervals that the aggregation is performed on. i.e. "interval"."1h" when the type is interpolation, will give one point per hour (starting at the start time) for the timestamp at the end of each time interval. aggregation count returns the chosen aggregation with the number of data points defined in count, i.e., if "count":14, the results set will contain 14 points, evenly dispersed over the time window.

[0071] `filters`. filters is an array of filters for the results. The possible filters are attributes, measurements, and qualities. All three of these filters can be applied in a single query.

[0072] `filters:attributes`. The filters attributes element filters the results by arrays of the chosen attribute host, attribute metricName and attribute queryIndex depending on what the available attributes are.

[0073] `filters:measurements`. filters measurements filters the results conditionally on the measurement/value of each data point in the result.

[0074] `filters:measurements:condition`. filters measurements condition is the conditional filtering for the measurement filter. The possible conditions are

"lt","gt","eq","le","ge","ne". These conditions are applied to the filters measurements values.

[0075] filters:measurements:values. filters measurements values are the value for the conditional filter, i.e., "condition":"le", "value":[36] would filter the results for only data points whose measurement/value is less than 36.

[0076] filters:qualities. filters qualities filters the data points in the result by their quality. Quality is the third parameter in the data point. Zero is bad and 3 is good data. If this filter is not defined, it is assumed that only the good data should be returned.

[0077] filters:qualities:value. filters qualities values is the values of quality the query will provide, i.e., "values":"0" will only return data points with qualities of 0.

[0078] groups. groups is used to group data points into separate results elements. The possibilities for groupings include attributes and quality.

[0079] groups:name. groups name is the name the attribute result is grouped by.

[0080] groups:values. groups values are the values the results will group by for the given name parameter, i.e., "name":"attribute", "value":["host"] will group each unique host name with its set of data points into its own result for the tag.

[0081] stats:rawCount. stats rawCount is the number of data points in the results prior to aggregation. This total is calculated after the quality filtering, i.e., if there is no filtering set for quality the rawCount will return the number of good data points in the time window specified.

[0082] stats:count. stats count is the total number of data points returned in the result.

[0083] datapoints. datapoints is an array of all the datapoints that matched in the query. Each datapoint is of format [<timestamp in epoch milliseconds>,<measurement>,<quality>].

[0084] FIG. 3 is an example method 300 of managing metadata pertaining to time series data using a time series component. In various embodiments, the time series data component is included in a user interface of an application executing on a device of a user and interacts with the time series service 120.

[0085] At operation 302, one or more sets of time series data corresponding to one or more IIOT assets are connected to one or more user interface components. In various embodiments, the connection may be established programmatically by a developer. In various embodiments, the user of the component may establish the connection using a user interface of the component. In various embodiments, the establishment of the connection to the time series data may be governed by user access controls. For example, to establish certain connection between the component and certain sets of time series data, the user may need to log into an account or otherwise establish necessary security credentials.

[0086] At operation 304, a presentation of the time series data in the time series component is customized. In various embodiments, a context of the user is identified. For example, it is determined whether the user is a field engineer, a data analyst, a control center operator, or a decision-maker. For example, the context may include a profile of the user, information pertaining to the environment in which the user is working, including a location of the user, a device of the user (e.g., including device type, information pertaining to displays connected to the device), and so on. The context may include information gathered from a sensor of a device of the user, such as audio and visual conditions, such as levels of ambient noise, lighting conditions, whether the user is in motion, and so on. The context may include a purpose of the user in viewing the information. For example, if the user is a field engineer, the purpose of the user may be to monitor an asset in the field with respect to an anomaly that was detected in the control center. Based on the context, the way in which the time series data is presented and the ways in which the user may interact with the data are customized.

[0087] At operation 306, annotations corresponding to the time series data are received. For example, a field engineer may annotate time series data for an

asset while monitoring the asset in the field. As another example, an analyst may annotate time series data based on a detection of a possible anomaly pertaining to an asset.

[0088] At operation 308, a state of the time series component is saved as a snapshot. The state includes a combination of user context data and annotations corresponding to the time series data being presented in the component. It also includes settings pertaining to the presentation of the time series data, such as the colors, overlays, and so on that were included in customizations made by the user.

[0089] At operation 310, the snapshot of the state of the time series component is shared for access by other users. In various embodiments, the other users may be specified by the user who saved the snapshot.

[0090] At operation 312, the snapshot of the state of the time series component is presented to another user. For example, the other user may open the snapshot in a time series component executing on the device of the other user. The presentation of the snapshot may be presented to the user in the context of the user who shared the snapshot or in a context of the user who is viewing the snapshot. In various embodiments, the user opening the saved snapshot may be able to control whether the presentation is based on the sharer's context or the sharee's context.

[0091] FIG. 4 depicts an example embodiment of a user interface 400 of the time series component. In various embodiments, the user interface 400 is presented by the time series component of an application executing on a device of the user. Reference numeral 4.1 corresponds to an asset context. The asset context may be a node in an asset hierarchy to which the time series data is linked (e.g., an asset that is the source of the time series data). A user may use the user interface to select a different asset, thus linking the component to a different data source. Reference numeral 4.2 corresponds to the legend, which indicates the meaning of the color-coded lines in the time series data visualization. Here, the inlet pressure and the inlet temperature of Turbine 1 and Turbine 2 are being compared in the graph. Reference numeral 4.3 corresponds to the Y-axis, which shows the scale of the parameters being plotted. Reference

numeral 4.4 corresponds to the X-axis, which shows the scale of the time scope. Reference numeral 4.5 corresponds to the time scope, which specifies a date and time range for the time series data. Reference numeral 4.6 corresponds to a Zoom level for the time series data. Reference numeral 4.7 corresponds to a submission button for changing the range of the time series data. Reference numeral 4.8 corresponds to a refresh button for reloading the time series data. Reference numeral 4.9 corresponds to an export button (e.g., for saving a snapshot of the time series data, including annotations and customizations). Reference numeral 4.10 corresponds to a settings button for configuring the time series component, including data sources, colors, parameters graphed, and so on.

[0092] FIG. 5 depicts an example embodiment of a user interface 500 of the time series component. In various embodiments, the user interface 500 is presented by the time series component of an application executing on a device of the user. Reference numeral 5.1 corresponds to a navigator. The navigator may provide a small preview of a larger time frame, and the ability to click and drag the entire time window or one edge to change what is displayed in the main plot or graph. Reference numeral 5.2 corresponds to a navigator scrubber. This feature allows the user to change the time range of the viewable graph area on the fly. They can “scrub” with the left and right handles independently or together.

[0093] FIG. 6 depicts an example embodiment of a user interface 600 of the time series component. In various embodiments, the user interface 600 is presented by the time series component of an application executing on a device of the user. Reference numeral 6.1 corresponds to click-and-drag-to-zoom feature. Users may zoom by clicking and dragging on a portion of the main plot. In various embodiments, users may zoom in on only the Y or X axis by performing a particular action, such as double-clicking on the selected region. In various embodiments, users may pan the plot by performing another action, such as shift+click+drag. Reference numeral 6.2 corresponds to the navigator scrubber, described above with respect to FIG. 5. The navigator scrubber may be updated to reflect any customization of the main plot.

[0094] FIG. 7 depicts an example embodiment of a user interface 700 of the time series component. In various embodiments, the user interface 700 is presented by the time series component of an application executing on a device of the user. Reference numeral 7.1 corresponds to the time scope. For example, when the user changes a time range with the scrubber or clicks and drags, the new range of time may be reflected in the time scope fields. Reference numeral 7.2 corresponds to a data point. For example, when the user drills down past a certain threshold on the graph, the appropriate data points may be represented on their correlating graph lines. Reference numeral 7.3 corresponds to a zoom feature. For example, when the user changes the time range of the chart in the Time Scope fields, on click and drag, or in the navigator, the zoom time range buttons may deselect. Reference numeral 7.4 corresponds to the X-axis. The X-axis shows the scale of the time scope. As the user changes the time range of the graph, the change may be reflected in the time scale on the X-axis. Reference numeral 7.5 corresponds to the navigator scrubber described above. As the user changes the time range of the graph, the change may be reflected in the scrubber.

[0095] FIG. 8 depicts an example embodiment of a user interface 800 of the time series component. In various embodiments, the user interface 800 is presented by the time series component of an application executing on a device of the user. Reference numeral 8.1 corresponds to a Popover Tool tip. When the user selects (or hovers or pauses) over a data point or multiple data points, information pertaining to the data points is presented in a popover window. Here, the selected data point includes two data items, particularly the inlet pressure of Turbine 1 and the inlet pressure of Turbine 2. The popover window associates the data items appropriately with the legend.

[0096] FIG. 9 depicts an example embodiment of a user interface 900 of the time series component. In various embodiments, the user interface 900 is presented by the time series component of an application executing on a device of the user. Reference numeral 9.1 corresponds to an export button. Clicking on the export button, for example, may bring up a list of multiple shareable file formats as well as sharing security options for saving a snapshot, as described above.

[0097] FIG. 10 depicts an example embodiment of a user interface 1000 of the time series component. In various embodiments, the user interface 1000 is presented by the time series component of an application executing on a device of the user. The user interface 1000 includes options for configuring the time series component. Reference numeral 10.1 corresponds to the title of the modal overlay for entering the configuration data. Reference numeral 10.2 corresponds to a button for closing the overlay. Reference numeral 10.3 corresponds to a component type. The user may choose from various types, including a time series plot. Reference numeral 10.4 corresponds to a data source. The user may specify the data source to which the time series component is connected (e.g., in this case, Turbine 1 Historian). Reference numeral 10.5 corresponds to a parameter from the data that is to be plotted in the component (e.g., in this case, temperature). In various embodiments, although not depicted, multiple parameters and multiple data sources may be selected. Reference numeral 10.6 corresponds to a width setting for controlling the width of the component with respect to a page (e.g., or card in which the component is included). Reference numeral 10.7 corresponds to data settings for the component, including a time frame, sampling mode, and sampling rate. Reference numeral 10.8 corresponds to component settings, including plot type and Y-axis settings. As described above, any such settings may be saved as metadata in a snapshot for sharing with other users.

[0098] The following examples are non-limiting examples of example embodiments of the subject matter disclosed herein.

Example 1

[0099] In example embodiments, a system is disclosed that comprises an instance of a time series data component. The instance of the time series data component configures one or more processors of a first device to perform operations comprising, at least presenting time series data corresponding to an asset in an industrial internet of things (IIoT) in a user interface of the first device based on a first context, receiving customizations pertaining to the presenting of the time series data, the customizations including annotations relating to the time series data, and sharing a snapshot of the time series data and

the customizations. In example embodiments, the system additionally comprises an additional instance of the time series data component. The additional instance of the time series data component configures one or more processors of a second device to, at least, upon an accessing of the shared snapshot, present the snapshot in a user interface of the second device based on a second context.

Example 2

[00100] In example embodiments, the system of Example 1 further comprises an instance of an adapter component. The instance of the adapter component configures one or more processors of the first device to connect with one or more time series data services to ingest the time series data.

Example 3

[00101] In example embodiments, in the system of Example 1 or 2, the second context includes information pertaining to security permissions of a user and the presenting of the snapshot in the user interface of the second device includes hiding some of the annotations or time series data based on the security permissions.

Example 4

[00102] In example embodiments, in the system of Example 1, 2, or 3, the second context pertains to attributes of an environment in which the presenting of the snapshot is to occur and the presenting of the snapshot includes adapting the presenting based on the environment.

Example 5

[00103] In example embodiments, in the system of Example 4, at least one of the attributes is an amount of light in the environment and the adapting of the presenting includes adjusting a brightness of the snapshot.

Example 6

[00104] In example embodiments, in the system of Example 1, 2, 3, 4, or 5, the second context corresponds to a field context and the presenting of the

snapshot includes incorporating information pertaining to an industrial asset that is within a geographical range of the second device.

Example 7

[00105] In example embodiments, in the system of Example 6, the information pertaining to the industrial asset and the snapshot assist a user of the second device with troubleshooting of the industrial asset.

[00106] Certain embodiments are described herein as including logic or a number of components, modules, or mechanisms. Modules may constitute either software modules (e.g., code embodied on a machine-readable medium) or hardware modules. A “hardware module” is a tangible unit capable of performing certain operations and may be configured or arranged in a certain physical manner. In various example embodiments, one or more computer systems (e.g., a standalone computer system, a client computer system, or a server computer system) or one or more hardware modules of a computer system (e.g., a processor or a group of processors) may be configured by software (e.g., an application or application portion) as a hardware module that operates to perform certain operations as described herein.

[00107] In some embodiments, a hardware module may be implemented mechanically, electronically, or any suitable combination thereof. For example, a hardware module may include dedicated circuitry or logic that is permanently configured to perform certain operations. For example, a hardware module may be a special-purpose processor, such as a field-programmable gate array (FPGA) or an application specific integrated circuit (ASIC). A hardware module may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations. For example, a hardware module may include software executed by a general-purpose processor or other programmable processor. Once configured by such software, hardware modules become specific machines (or specific components of a machine) uniquely tailored to perform the configured functions and are no longer general-purpose processors. It will be appreciated that the decision to implement a hardware module mechanically, in dedicated and permanently configured circuitry, or in temporarily configured circuitry (e.g., configured by software) may be driven by cost and time considerations.

[00108] Accordingly, the phrase “hardware module” should be understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein. As used herein, “hardware-implemented module” refers to a hardware module. Considering embodiments in which hardware modules are temporarily configured (e.g., programmed), each of the hardware modules need not be configured or instantiated at any one instance in time. For example, where a hardware module comprises a general-purpose processor configured by software to become a special-purpose processor, the general-purpose processor may be configured as respectively different special-purpose processors (e.g., comprising different hardware modules) at different times. Software accordingly configures a particular processor or processors, for example, to constitute a particular hardware module at one instance of time and to constitute a different hardware module at a different instance of time.

[00109] Hardware modules can provide information to, and receive information from, other hardware modules. Accordingly, the described hardware modules may be regarded as being communicatively coupled. Where multiple hardware modules exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses) between or among two or more of the hardware modules. In embodiments in which multiple hardware modules are configured or instantiated at different times, communications between such hardware modules may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware modules have access. For example, one hardware module may perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further hardware module may then, at a later time, access the memory device to retrieve and process the stored output. Hardware modules may also initiate communications with input or output devices, and can operate on a resource (e.g., a collection of information).

[00110] The various operations of example methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant

operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented modules that operate to perform one or more operations or functions described herein. As used herein, “processor-implemented module” refers to a hardware module implemented using one or more processors.

[00111] Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented modules. Moreover, the one or more processors may also operate to support performance of the relevant operations in a “cloud computing” environment or as a “software as a service” (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines including processors), with these operations being accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., an API).

[00112] The performance of certain of the operations may be distributed among the processors, not only residing within a single machine, but deployed across a number of machines. In some example embodiments, the processors or processor-implemented modules may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors or processor-implemented modules may be distributed across a number of geographic locations.

[00113] The modules, methods, applications, and so forth described herein are implemented, in some embodiments, in the context of a machine and an associated software architecture. The sections below describe representative software architecture(s) and machine (e.g., hardware) architecture(s) that are suitable for use with the disclosed embodiments.

[00114] Software architectures are used in conjunction with hardware architectures to create devices and machines tailored to particular purposes. For example, a particular hardware architecture coupled with a particular software architecture will create a mobile device, such as a mobile phone, tablet device, or so forth. A slightly different hardware and software architecture may yield a smart device for use in the “internet of things,” while yet another combination produces a server computer for use within a cloud computing architecture. Not

all combinations of such software and hardware architectures are presented here, as those of skill in the art can readily understand how to implement the inventive subject matter in different contexts from the disclosure contained herein.

[00115] FIG. 11 is a block diagram 1100 illustrating a representative software architecture 802, which may be used in conjunction with various hardware architectures herein described. FIG. 11 is merely a non-limiting example of a software architecture 802, and it will be appreciated that many other architectures may be implemented to facilitate the functionality described herein. The software architecture 802 may be executing on hardware such as a machine 1200 of FIG. 12 that includes, among other things, processors 910, memory/storage 930, and I/O components 950. A representative hardware layer 804 is illustrated and can represent, for example, the machine 1200 of FIG. 12. The representative hardware layer 804 comprises one or more processing units 806 having associated executable instructions 808. The executable instructions 808 represent the executable instructions of the software architecture 802, including implementation of the methods, modules, and so forth herein. The hardware layer 804 also includes memory and/or storage modules 810, which also have the executable instructions 808. The hardware layer 804 may also comprise other hardware 812, which represents any other hardware of the hardware layer 804, such as the other hardware illustrated as part of the machine 1200.

[00116] In the example architecture of FIG. 11, the software architecture 802 may be conceptualized as a stack of layers where each layer provides particular functionality. For example, the software architecture 802 may include layers such as an operating system 814, libraries 816, frameworks/middleware 818, applications 820, and a presentation layer 844. Operationally, the applications 820 and/or other components within the layers may invoke API calls 824 through the software stack and receive a response, returned values, and so forth illustrated as messages 826 in response to the API calls 824. The layers illustrated are representative in nature, and not all software architectures have all layers. For example, some mobile or special purpose operating systems may not provide a frameworks/middleware 818, while others may provide such a layer. Other software architectures may include additional or different layers.

[00117] The operating system 814 may manage hardware resources and provide common services. The operating system 814 may include, for example, a kernel 828, services 830, and drivers 832. The kernel 828 may act as an abstraction layer between the hardware and the other software layers. For example, the kernel 828 may be responsible for memory management, processor management (e.g., scheduling), component management, networking, security settings, and so on. The services 830 may provide other common services for the other software layers. The drivers 832 may be responsible for controlling or interfacing with the underlying hardware. For instance, the drivers 832 may include display drivers, camera drivers, Bluetooth® drivers, flash memory drivers, serial communication drivers (e.g., Universal Serial Bus (USB) drivers), Wi-Fi® drivers, audio drivers, power management drivers, and so forth, depending on the hardware configuration.

[00118] The libraries 816 may provide a common infrastructure that may be utilized by the applications 820 and/or other components and/or layers. The libraries 816 typically provide functionality that allows other software modules to perform tasks in an easier fashion than to interface directly with the underlying operating system 814 functionality (e.g., kernel 828, services 830, and/or drivers 832). The libraries 816 may include system libraries 834 (e.g., C standard library) that may provide functions such as memory allocation functions, string manipulation functions, mathematic functions, and the like. In addition, the libraries 816 may include API libraries 836 such as media libraries (e.g., libraries to support presentation and manipulation of various media formats such as MPEG4, H.264, MP3, AAC, AMR, JPG, PNG), graphics libraries (e.g., an OpenGL framework that may be used to render 2D and 3D in a graphic context on a display), database libraries (e.g., SQLite that may provide various relational database functions), web libraries (e.g., WebKit that may provide web browsing functionality), and the like. The libraries 816 may also include a wide variety of other libraries 838 to provide many other APIs to the applications 820 and other software components/modules.

[00119] The frameworks/middleware 818 may provide a higher-level common infrastructure that may be utilized by the applications 820 and/or other software components/modules. For example, the frameworks/middleware 818 may provide various graphic user interface (GUI) functions, high-level resource

management, high-level location services, and so forth. The frameworks/middleware 818 may provide a broad spectrum of other APIs that may be utilized by the applications 820 and/or other software components/modules, some of which may be specific to a particular operating system or platform.

[00120] The applications 820 include built-in applications 840 and/or third-party applications 842. Examples of representative built-in applications 840 may include, but are not limited to, a contacts application, a browser application, a book reader application, a location application, a media application, a messaging application, and/or a game application. Third-party applications 842 may include any of the built-in applications 840 as well as a broad assortment of other applications. In a specific example, the third-party application 842 (e.g., an application developed using the Android™ or iOS™ software development kit (SDK) by an entity other than the vendor of the particular platform) may be mobile software running on a mobile operating system such as iOS™, Android™, Windows® Phone, or other mobile operating systems. In this example, the third-party application 842 may invoke the API calls 824 provided by the mobile operating system such as the operating system 814 to facilitate functionality described herein.

[00121] The applications 820 may utilize built-in operating system functions (e.g., kernel 828, services 830, and/or drivers 832), libraries (e.g., system libraries 834, API libraries 836, and other libraries 838), and frameworks/middleware 818 to create user interfaces to interact with users of the system. Alternatively, or additionally, in some systems, interactions with a user may occur through a presentation layer, such as the presentation layer 844. In these systems, the application/module “logic” can be separated from the aspects of the application/module that interact with a user.

[00122] Some software architectures utilize virtual machines. In the example of FIG. 11, this is illustrated by a virtual machine 848. A virtual machine creates a software environment where applications/modules can execute as if they were executing on a hardware machine (such as the machine 1200 of FIG. 12, for example). The virtual machine 848 is hosted by a host operating system (operating system 814 in FIG. 11) and typically, although not always, has a virtual machine monitor 846, which manages the operation of the virtual

machine 848 as well as the interface with the host operating system (i.e., operating system 814). A software architecture executes within the virtual machine 848, such as an operating system 850, libraries 852, frameworks/middleware 854, applications 856, and/or a presentation layer 858. These layers of software architecture executing within the virtual machine 848 can be the same as corresponding layers previously described or may be different.

[00123] FIG. 12 is a block diagram illustrating components of a machine 1200, according to some example embodiments, able to read instructions 916 from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein. Specifically, FIG. 12 shows a diagrammatic representation of the machine 1200 in the example form of a computer system, within which the instructions 916 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 1200 to perform any one or more of the methodologies discussed herein may be executed. For example, the instructions 916 may cause the machine 1200 to execute the flow diagram of FIG. 3. Additionally, or alternatively, the instructions 916 may implement modules of FIG. 1, and so forth. The instructions 916 transform the general, non-programmed machine 1200 into a particular machine programmed to carry out the described and illustrated functions in the manner described. In alternative embodiments, the machine 1200 operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine 1200 may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 1200 may comprise, but not be limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (PDA), an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the instructions 916, sequentially or otherwise, that specify actions to be taken by the machine 1200. Further, while only a single machine

1200 is illustrated, the term “machine” shall also be taken to include a collection of machines 1200 that individually or jointly execute the instructions 916 to perform any one or more of the methodologies discussed herein.

[00124] The machine 1200 may include processors 910, memory/storage 930, and I/O components 950, which may be configured to communicate with each other such as via a bus 902. In an example embodiment, the processors 910 (e.g., a central processing unit (CPU), a reduced instruction set computing (RISC) processor, a complex instruction set computing (CISC) processor, a graphics processing unit (GPU), a digital signal processor (DSP), an ASIC, a radio-frequency integrated circuit (RFIC), another processor, or any suitable combination thereof) may include, for example, a processor 912 and a processor 914 that may execute the instructions 916. The term “processor” is intended to include a multi-core processor 912, 914 that may comprise two or more independent processors 912, 914 (sometimes referred to as “cores”) that may execute the instructions 916 contemporaneously. Although FIG. 12 shows multiple processors 910, the machine 1200 may include a single processor 912, 914 with a single core, a single processor 912, 914 with multiple cores (e.g., a multi-core processor 912, 914), multiple processors 912, 914 with a single core, multiple processors 912, 914 with multiples cores, or any combination thereof.

[00125] The memory/storage 930 may include a memory 932, such as a main memory, or other memory storage, and a storage unit 936, both accessible to the processors 910 such as via the bus 902. The storage unit 936 and memory 932 store the instructions 916 embodying any one or more of the methodologies or functions described herein. The instructions 916 may also reside, completely or partially, within the memory 932, within the storage unit 936, within at least one of the processors 910 (e.g., within the cache memory of processor 912, 914), or any suitable combination thereof, during execution thereof by the machine 1200. Accordingly, the memory 932, the storage unit 936, and the memory of the processors 910 are examples of machine-readable media.

[00126] As used herein, “machine-readable medium” means a device able to store the instructions 916 and data temporarily or permanently and may include, but not be limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, optical media, magnetic media, cache memory, other types of storage (e.g., erasable programmable read-only

memory (EEPROM)), and/or any suitable combination thereof. The term “machine-readable medium” should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, or associated caches and servers) able to store the instructions 916. The term “machine-readable medium” shall also be taken to include any medium, or combination of multiple media, that is capable of storing instructions (e.g., instructions 916) for execution by a machine (e.g., machine 1200), such that the instructions 916, when executed by one or more processors of the machine 1200 (e.g., processors 910), cause the machine 1200 to perform any one or more of the methodologies described herein. Accordingly, a “machine-readable medium” refers to a single storage apparatus or device, as well as “cloud-based” storage systems or storage networks that include multiple storage apparatus or devices. The term “machine-readable medium” excludes signals per se.

[00127] The I/O components 950 may include a wide variety of components to receive input, provide output, produce output, transmit information, exchange information, capture measurements, and so on. The specific I/O components 950 that are included in a particular machine 1200 will depend on the type of machine 1200. For example, portable machines such as mobile phones will likely include a touch input device or other such input mechanisms, while a headless server machine will likely not include such a touch input device. It will be appreciated that the I/O components 950 may include many other components that are not shown in FIG. 12. The I/O components 950 are grouped according to functionality merely for simplifying the following discussion, and the grouping is in no way limiting. In various example embodiments, the I/O components 950 may include output components 952 and input components 954. The output components 952 may include visual components (e.g., a display such as a plasma display panel (PDP), a light emitting diode (LED) display, a liquid crystal display (LCD), a projector, or a cathode ray tube (CRT)), acoustic components (e.g., speakers), haptic components (e.g., a vibratory motor, resistance mechanisms), other signal generators, and so forth. The input components 954 may include alphanumeric input components (e.g., a keyboard, a touch screen configured to receive alphanumeric input, a photo-optical keyboard, or other alphanumeric input components), point based input components (e.g., a mouse, a touchpad, a

trackball, a joystick, a motion sensor, or other pointing instruments), tactile input components (e.g., a physical button, a touch screen that provides location and/or force of touches or touch gestures, or other tactile input components), audio input components (e.g., a microphone), and the like.

[00128] In further example embodiments, the I/O components 950 may include biometric components 956, motion components 958, environmental components 960, or position components 962, among a wide array of other components. For example, the biometric components 956 may include components to detect expressions (e.g., hand expressions, facial expressions, vocal expressions, body gestures, or eye tracking), measure biosignals (e.g., blood pressure, heart rate, body temperature, perspiration, or brain waves), identify a person (e.g., voice identification, retinal identification, facial identification, fingerprint identification, or electroencephalogram based identification), and the like. The motion components 958 may include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope), and so forth. The environmental components 960 may include, for example, illumination sensor components (e.g., photometer), temperature sensor components (e.g., one or more thermometers that detect ambient temperature), humidity sensor components, pressure sensor components (e.g., barometer), acoustic sensor components (e.g., one or more microphones that detect background noise), proximity sensor components (e.g., infrared sensors that detect nearby objects), gas sensors (e.g., gas detection sensors to detect concentrations of hazardous gases for safety or to measure pollutants in the atmosphere), or other components that may provide indications, measurements, or signals corresponding to a surrounding physical environment. The position components 962 may include location sensor components (e.g., a Global Position System (GPS) receiver component), altitude sensor components (e.g., altimeters or barometers that detect air pressure from which altitude may be derived), orientation sensor components (e.g., magnetometers), and the like.

[00129] Communication may be implemented using a wide variety of technologies. The I/O components 950 may include communication components 964 operable to couple the machine 1200 to a network 980 or devices 970 via a coupling 982 and a coupling 972 respectively. For example, the communication

components 964 may include a network interface component or other suitable device to interface with the network 980. In further examples, the communication components 964 may include wired communication components, wireless communication components, cellular communication components, near field communication (NFC) components, Bluetooth® components (e.g., Bluetooth® Low Energy), Wi-Fi® components, and other communication components to provide communication via other modalities. The devices 970 may be another machine or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a USB).

[00130] Moreover, the communication components 964 may detect identifiers or include components operable to detect identifiers. For example, the communication components 964 may include radio frequency identification (RFID) tag reader components, NFC smart tag detection components, optical reader components (e.g., an optical sensor to detect one-dimensional bar codes such as Universal Product Code (UPC) bar code, multi-dimensional bar codes such as Quick Response (QR) code, Aztec code, Data Matrix, Dataglyph, MaxiCode, PDF417, Ultra Code, UCC RSS-2D bar code, and other optical codes), or acoustic detection components (e.g., microphones to identify tagged audio signals). In addition, a variety of information may be derived via the communication components 964, such as location via Internet Protocol (IP) geolocation, location via Wi-Fi® signal triangulation, location via detecting an NFC beacon signal that may indicate a particular location, and so forth.

[00131] In various example embodiments, one or more portions of the network 980 may be an ad hoc network, an intranet, an extranet, a virtual private network (VPN), a local area network (LAN), a wireless LAN (WLAN), a wide area network (WAN), a wireless WAN (WWAN), a metropolitan area network (MAN), the Internet, a portion of the Internet, a portion of the public switched telephone network (PSTN), a plain old telephone service (POTS) network, a cellular telephone network, a wireless network, a Wi-Fi® network, another type of network, or a combination of two or more such networks. For example, the network 980 or a portion of the network 980 may include a wireless or cellular network and the coupling 982 may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or another type of cellular or wireless coupling. In this example, the

coupling 982 may implement any of a variety of types of data transfer technology, such as Single Carrier Radio Transmission Technology (1xRTT), Evolution-Data Optimized (EVDO) technology, General Packet Radio Service (GPRS) technology, Enhanced Data rates for GSM Evolution (EDGE) technology, third Generation Partnership Project (3GPP) including 3G, fourth generation wireless (4G) networks, Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX), Long Term Evolution (LTE) standard, others defined by various standard-setting organizations, other long range protocols, or other data transfer technology.

[00132] The instructions 916 may be transmitted or received over the network 980 using a transmission medium via a network interface device (e.g., a network interface component included in the communication components 964) and utilizing any one of a number of well-known transfer protocols (e.g., hypertext transfer protocol (HTTP)). Similarly, the instructions 916 may be transmitted or received using a transmission medium via the coupling 972 (e.g., a peer-to-peer coupling) to the devices 970. The term “transmission medium” shall be taken to include any intangible medium that is capable of storing, encoding, or carrying the instructions 916 for execution by the machine 1200, and includes digital or analog communications signals or other intangible media to facilitate communication of such software.

[00133] Throughout this specification, plural instances may implement components, operations, or structures described as a single instance. Although individual operations of one or more methods are illustrated and described as separate operations, one or more of the individual operations may be performed concurrently, and nothing requires that the operations be performed in the order illustrated. Structures and functionality presented as separate components in example configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the subject matter herein.

[00134] Although an overview of the inventive subject matter has been described with reference to specific example embodiments, various

modifications and changes may be made to these embodiments without departing from the broader scope of embodiments of the present disclosure. Such embodiments of the inventive subject matter may be referred to herein, individually or collectively, by the term “invention” merely for convenience and without intending to voluntarily limit the scope of this application to any single disclosure or inventive concept if more than one is, in fact, disclosed.

[00135] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[00136] As used herein, the term “or” may be construed in either an inclusive or exclusive sense. Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, modules, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present disclosure. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

1. A system comprising:
an instance of a time series data component, the instance of the time series data component configuring one or more processors of a first device to perform operations comprising, at least:
presenting time series data corresponding to an asset in an industrial internet of things (IIoT) in a user interface of the first device based on a first context;
receiving customizations pertaining to the presenting of the time series data, the customizations including annotations relating to the time series data; and
sharing a snapshot of the time series data and the customizations; and
an additional instance of the time series data component, the additional instance of the time series data component configuring one or more processors of a second device to, at least, upon an accessing of the shared snapshot, present the snapshot in a user interface of the second device based on a second context.
2. The system of claim 1, further comprising an instance of an adapter component, the instance of the adapter component configuring one or more processors of the first device to connect with one or more time series data services to ingest the time series data.
3. The system of claim 1, wherein the second context includes information pertaining to security permissions of a user and the presenting of the snapshot in the user interface of the second device includes hiding some of the annotations or time series data based on the security permissions.
4. The system of claim 1, wherein the second context pertains to attributes of an environment in which the presenting of the snapshot is to occur and the presenting of the snapshot includes adapting the presenting based on the environment.

5. The system of claim 4, wherein at least one of the attributes is an amount of light in the environment and the adapting of the presenting includes adjusting a brightness of the snapshot.

6. The system of claim 1, wherein the second context corresponds to a field context and the presenting of the snapshot includes incorporating information pertaining to an industrial asset that is within a geographical range of the second device.

7. The system of claim 6, wherein the information pertaining to the industrial asset and the snapshot assist a user of the second device with troubleshooting of the industrial asset.

8. A method comprising:
at a first device configured to access a data store:
 presenting time series data corresponding to an asset in an industrial internet of things (IIoT) in a user interface of the first device based on a first context;
 receiving customizations pertaining to the presenting of the time series data, the customizations including annotations relating to the time series data; and
 sharing a snapshot of the time series data and the customizations in the data store; and
at a second device configured to access the data store:
 upon an accessing of the shared snapshot in the data store, presenting the snapshot in a user interface of the second device based on a second context.

9. The method of claim 8, further comprising, at the first device, communicating with one or more time series data services to ingest the time series data.

10. The method of claim 8, wherein the second context includes information pertaining to security permissions of a user and the presenting of the snapshot in the user interface of the second device includes hiding some of the annotations or time series data based on the security permissions.

11. The method of claim 8, wherein the second context pertains to attributes of an environment in which the presenting of the snapshot is to occur and the presenting of the snapshot includes adapting the presenting based on the environment.

12. The method of claim 11, wherein at least one of the attributes is an amount of light in the environment and the adapting of the presenting includes adjusting a brightness of the snapshot.

13. The method of claim 8, wherein the second context corresponds to a field context and the presenting of the snapshot includes incorporating information pertaining to an industrial asset that is within a geographical range from the second device.

14. The method of claim 13, wherein the information pertaining to the industrial asset and the snapshot assist a user of the second device with troubleshooting of the industrial asset.

15. A non-transitory machine-readable storage medium storing a set of instructions that, when executed by at least one processor, causes the at least one processor to perform operations, the operations comprising:
using a first instance of a time series data component for presenting time series data corresponding to an asset in an industrial internet of things (IIoT) in a user interface of a first device based on a first context;
receiving customizations pertaining to the presenting of the time series data, the customizations including annotations relating to the time series data;
sharing a snapshot of the time series data and the customizations; and

using a second instance of the time series data component for, upon an accessing of the shared snapshot by a second device, presenting the snapshot in a user interface of the second device based on a second context.

16. The non-transitory machine-readable storage medium of claim 15, using an adapter component for connecting with one or more time series data services to ingest the time series data.

17. The non-transitory machine-readable storage medium of claim 15, wherein the second context includes information pertaining to security permissions of a user and the presenting of the snapshot in the user interface of the second device includes hiding some of the annotations or time series data based on the security permissions.

18. The non-transitory machine-readable storage medium of claim 15, wherein the second context pertains to attributes of an environment in which the presenting of the snapshot is to occur and the presenting of the snapshot includes adapting the presenting based on the environment.

19. The non-transitory machine-readable storage medium of claim 18, wherein at least one of the attributes is an amount of light in the environment and the adapting of the presenting includes adjusting a brightness of the snapshot.

20. The non-transitory machine-readable storage medium of claim 15, wherein the second context corresponds to a field context and the presenting of the snapshot includes incorporating information pertaining to an industrial asset that is within a geographical range from the second device.

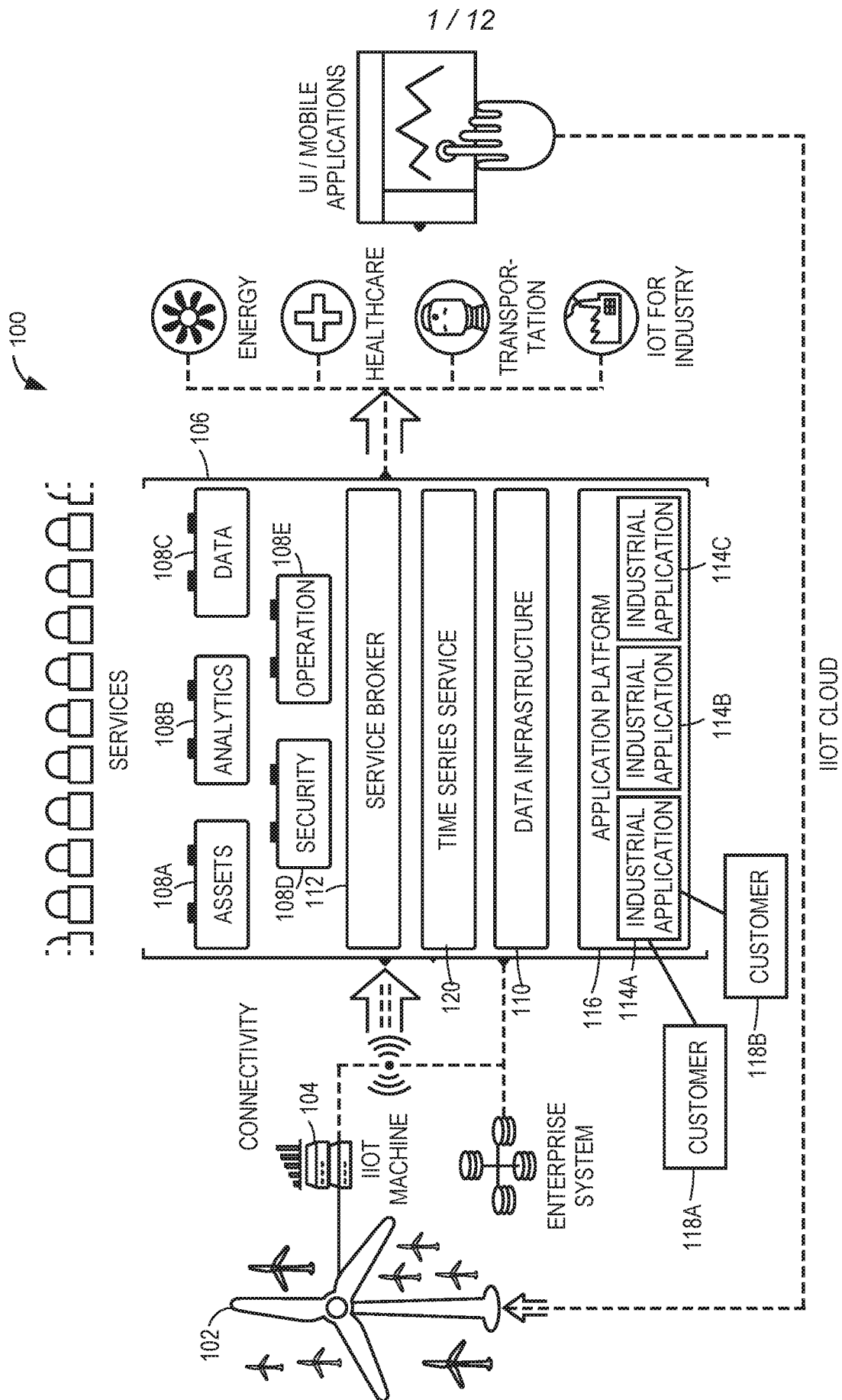


FIG. 1

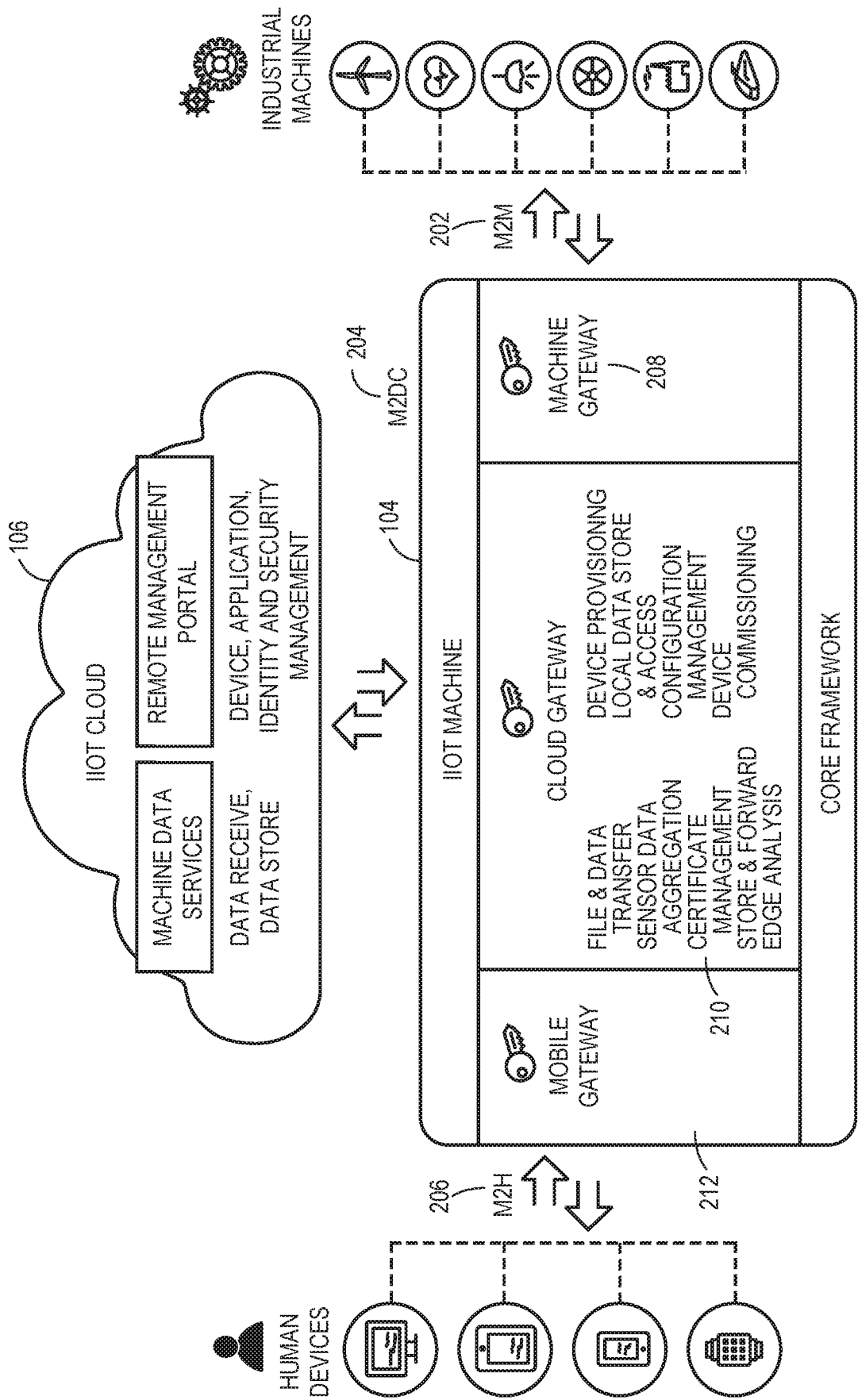


FIG. 2

3 / 12

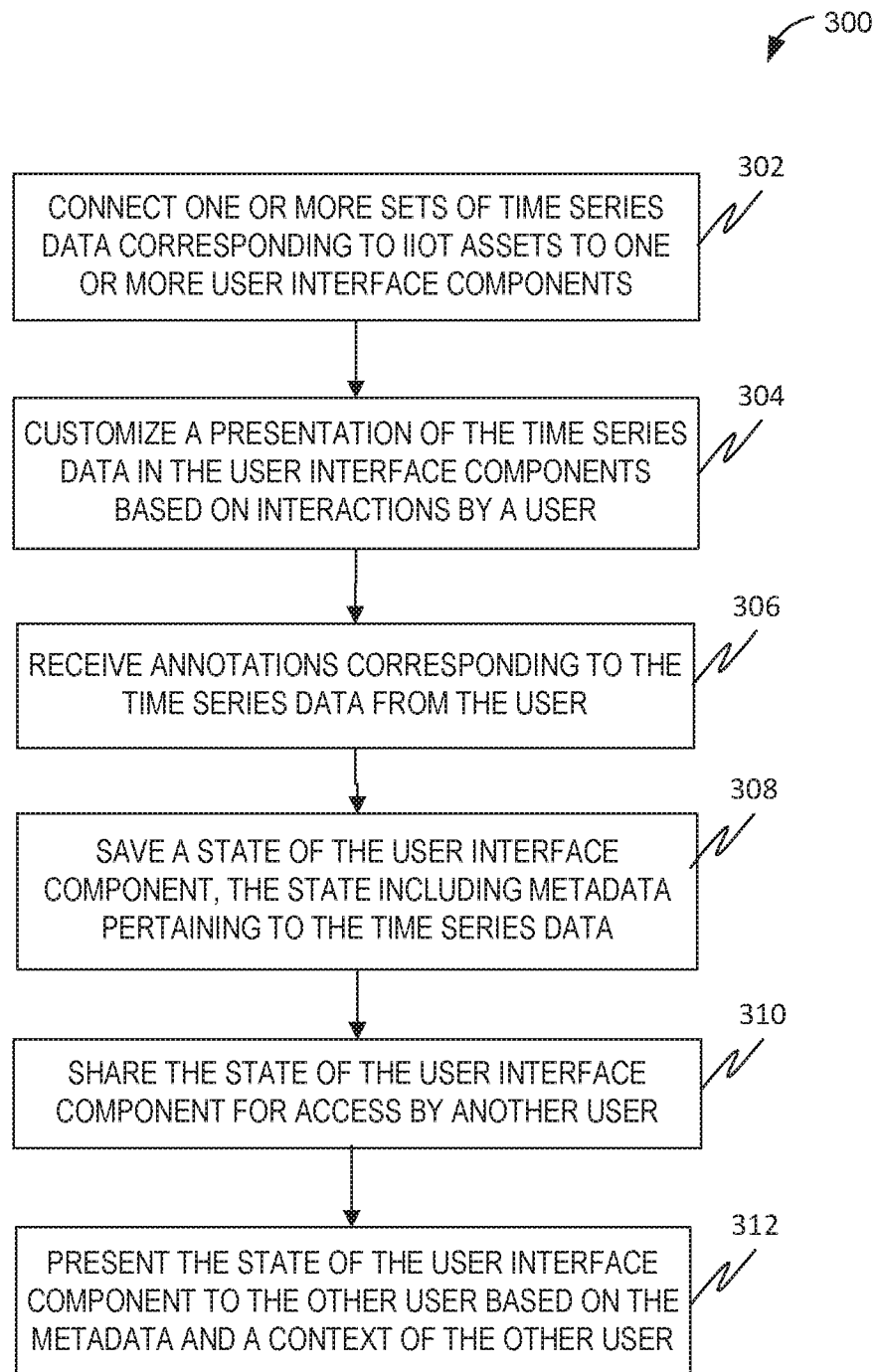


FIG. 3

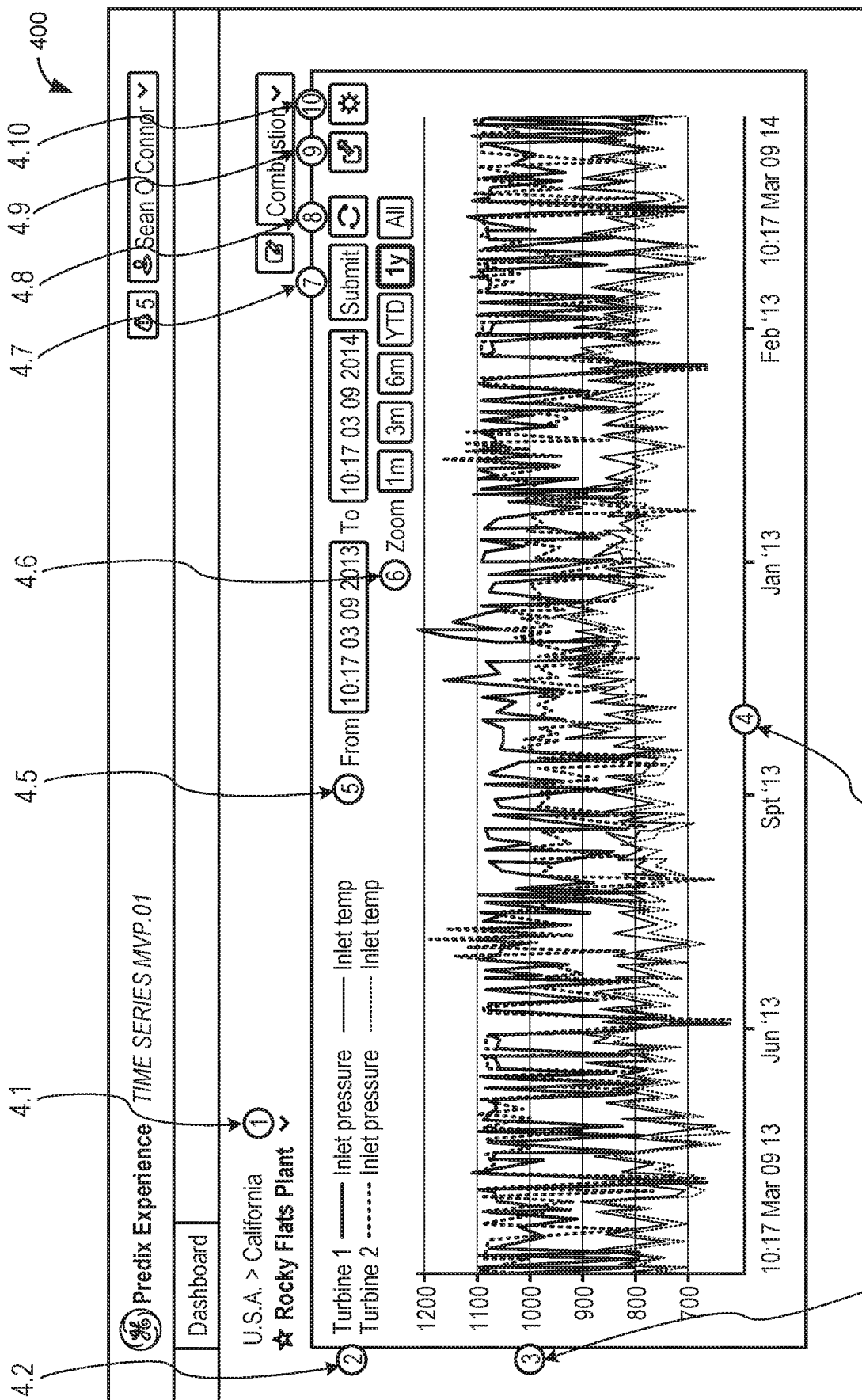


FIG. 4

5 / 12

500

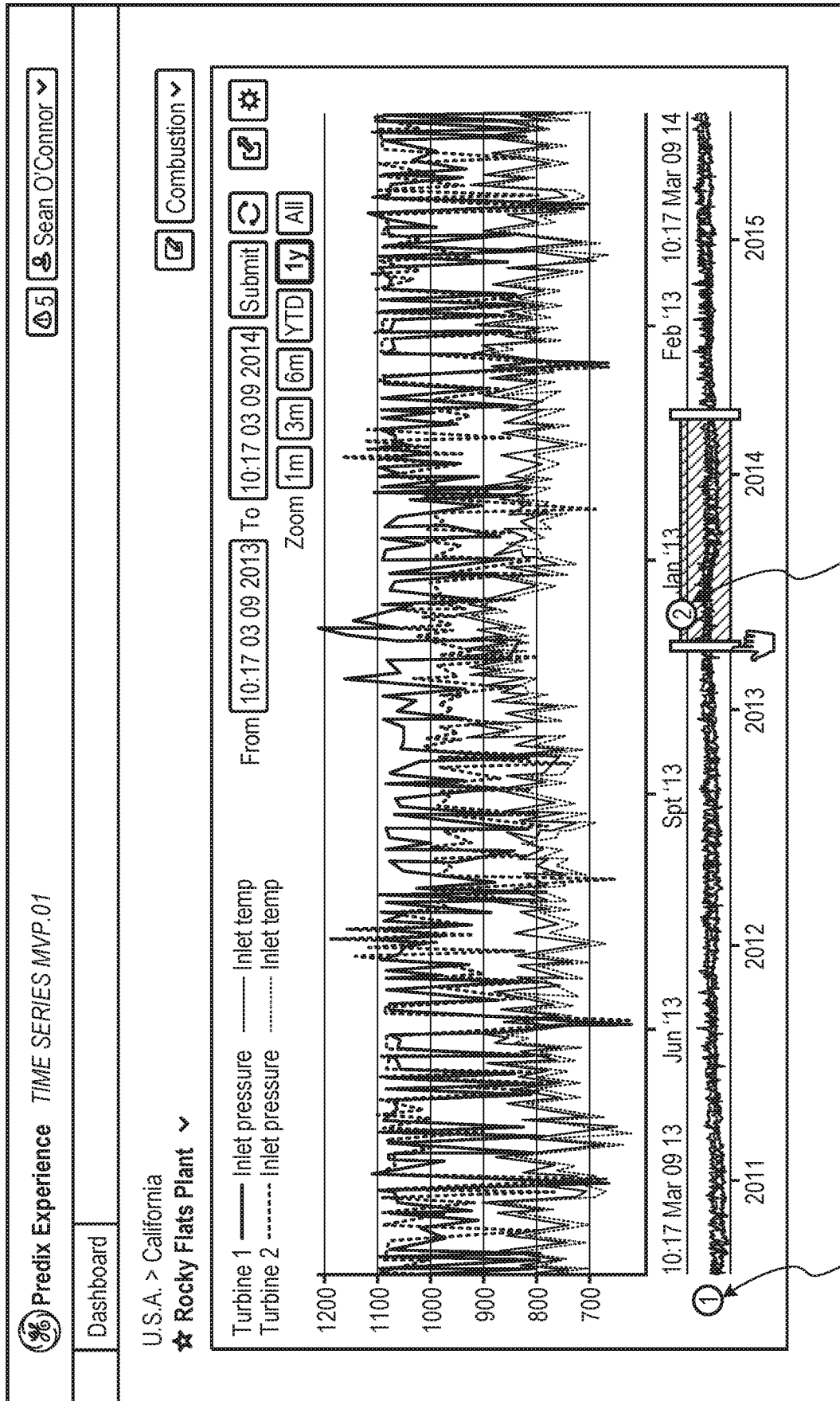


FIG. 5

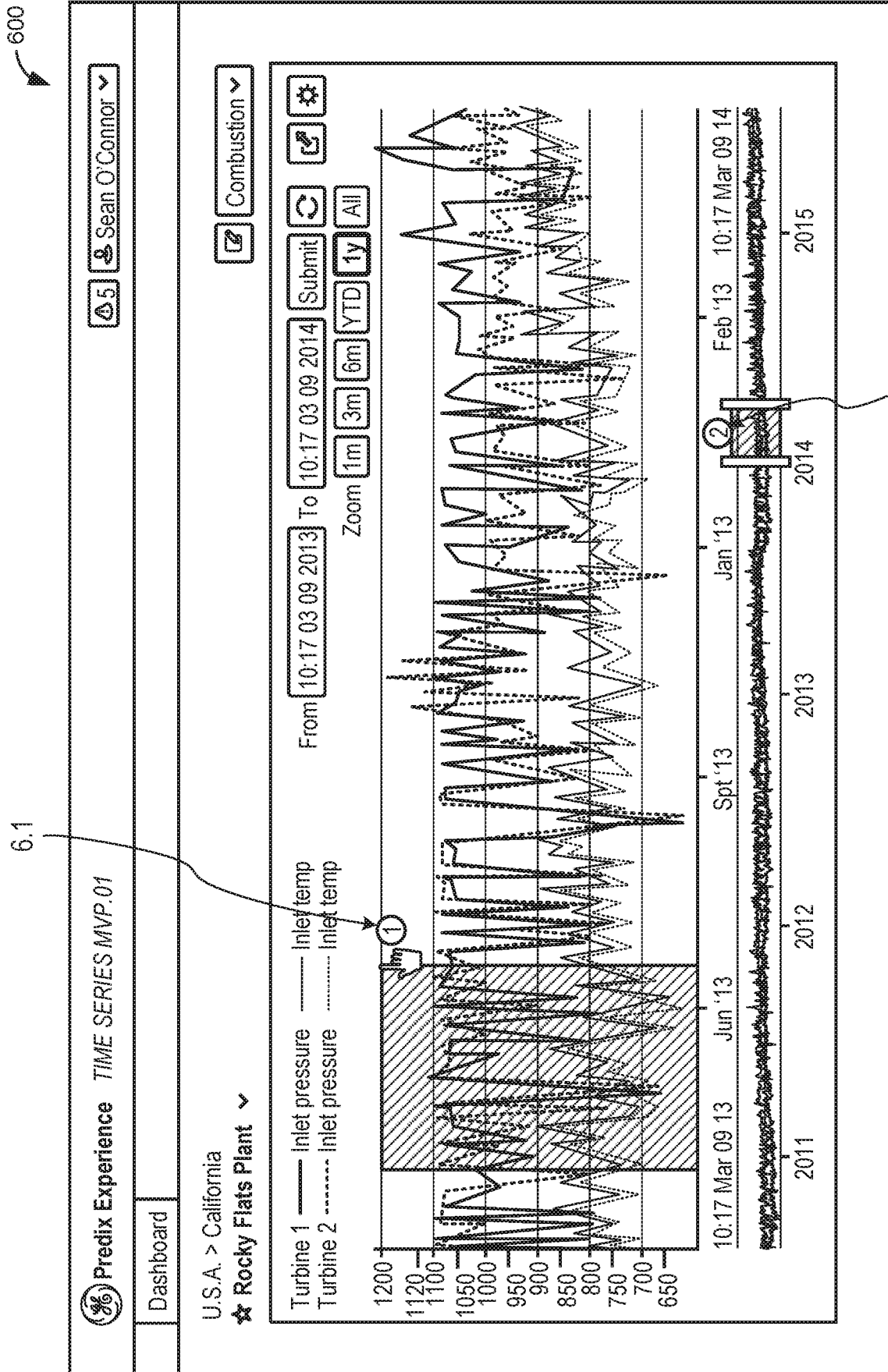
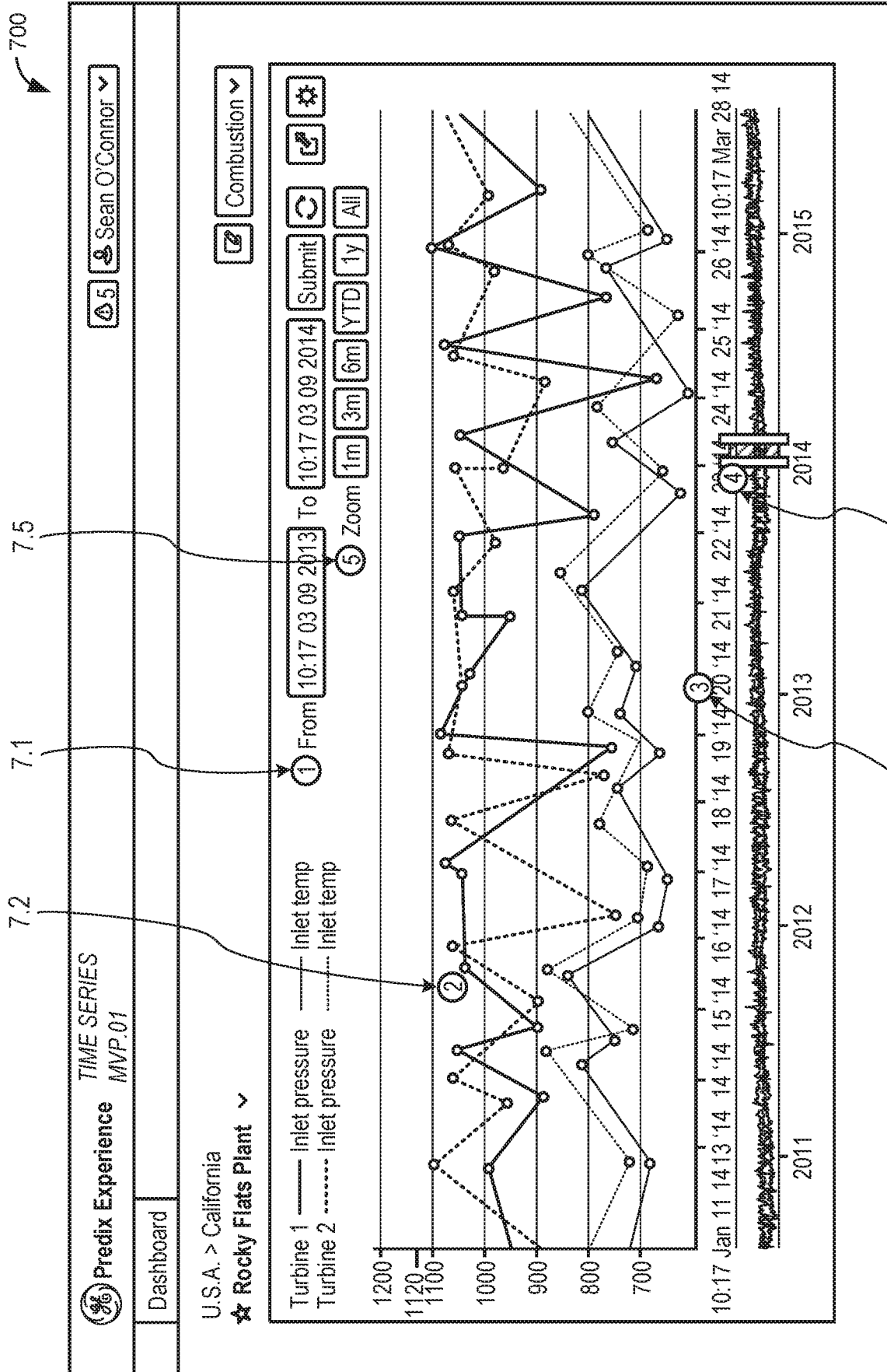


FIG. 6

7 / 12



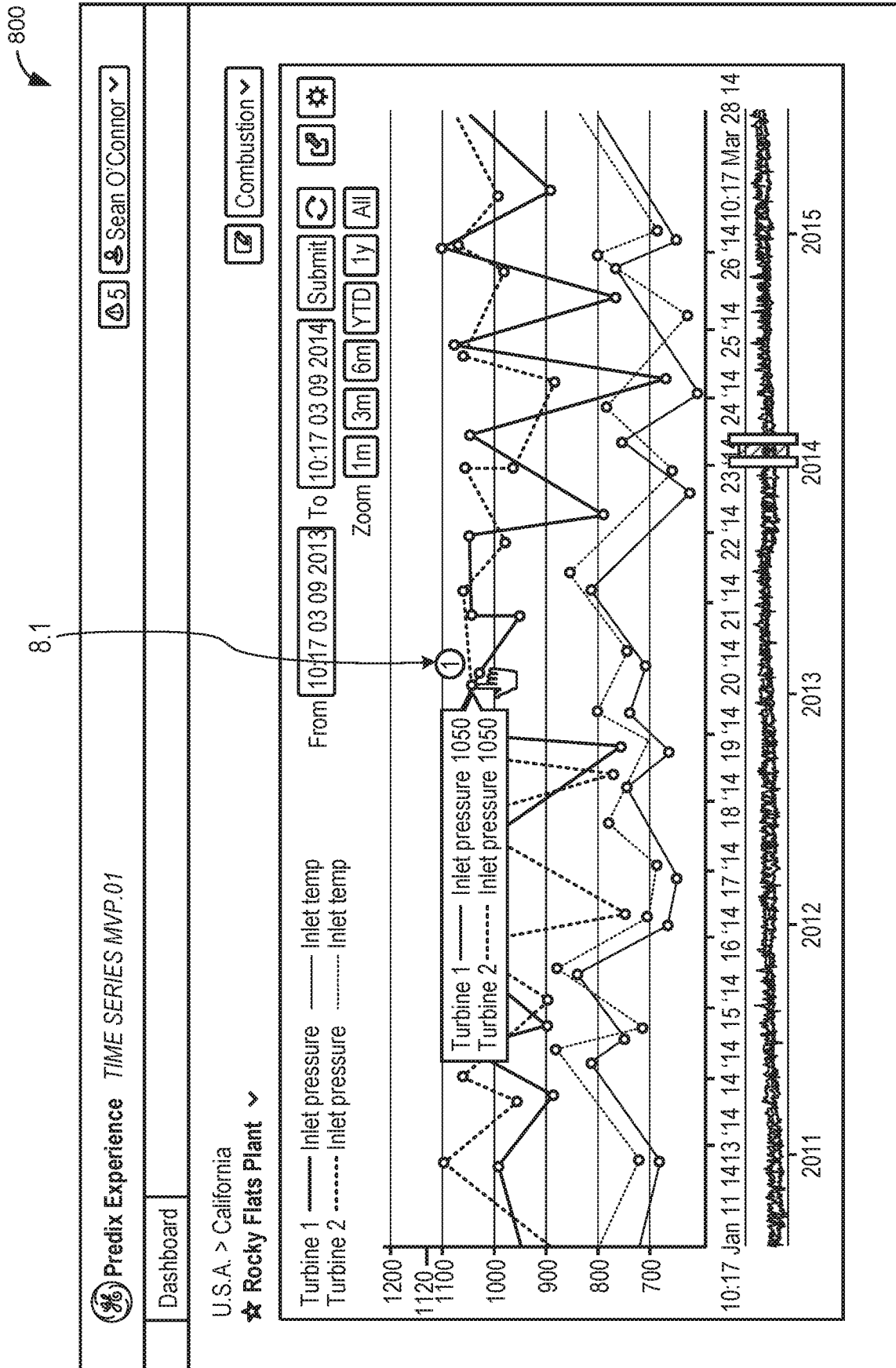


FIG. 8

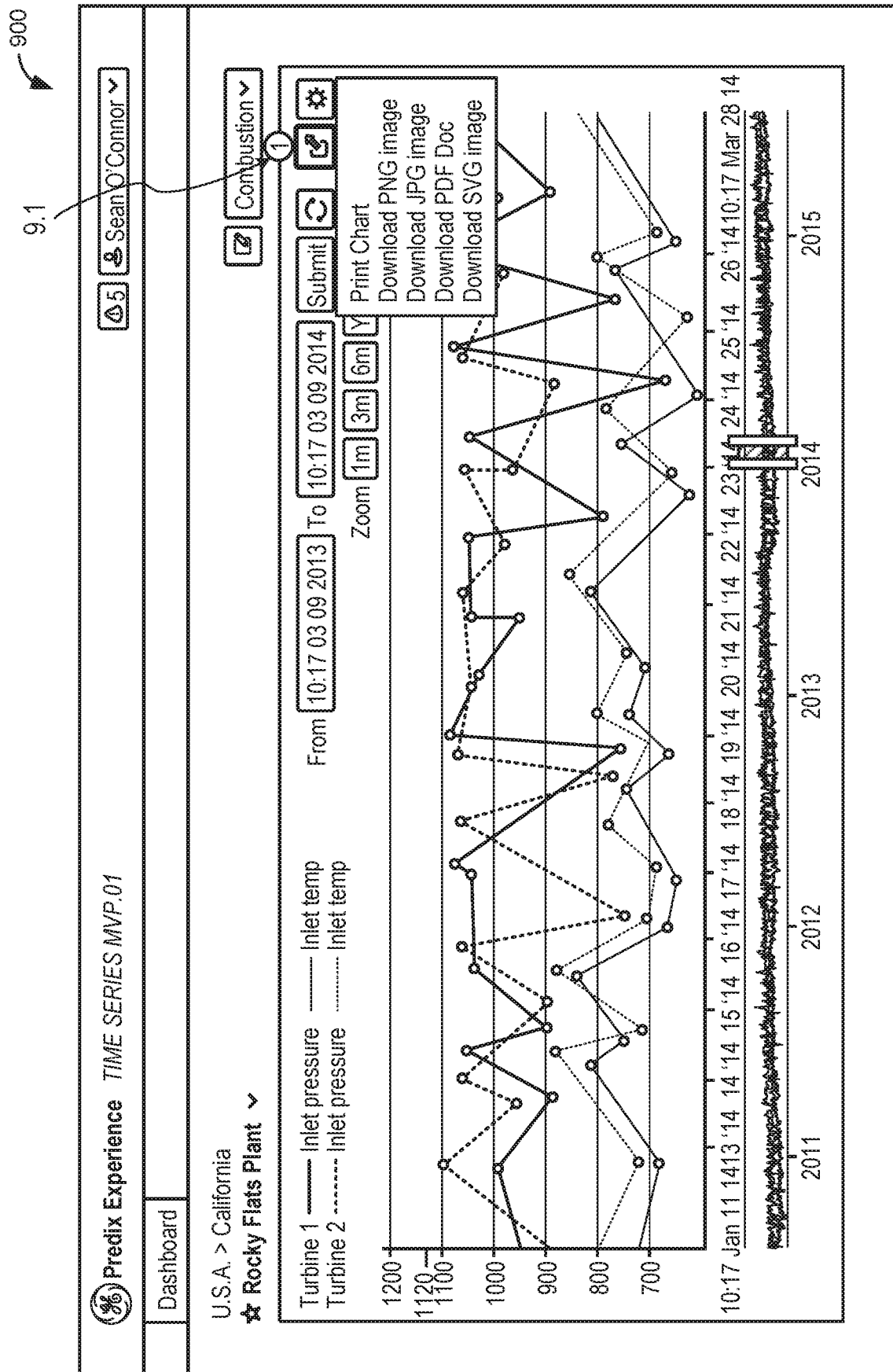


FIG. 9

10 / 12

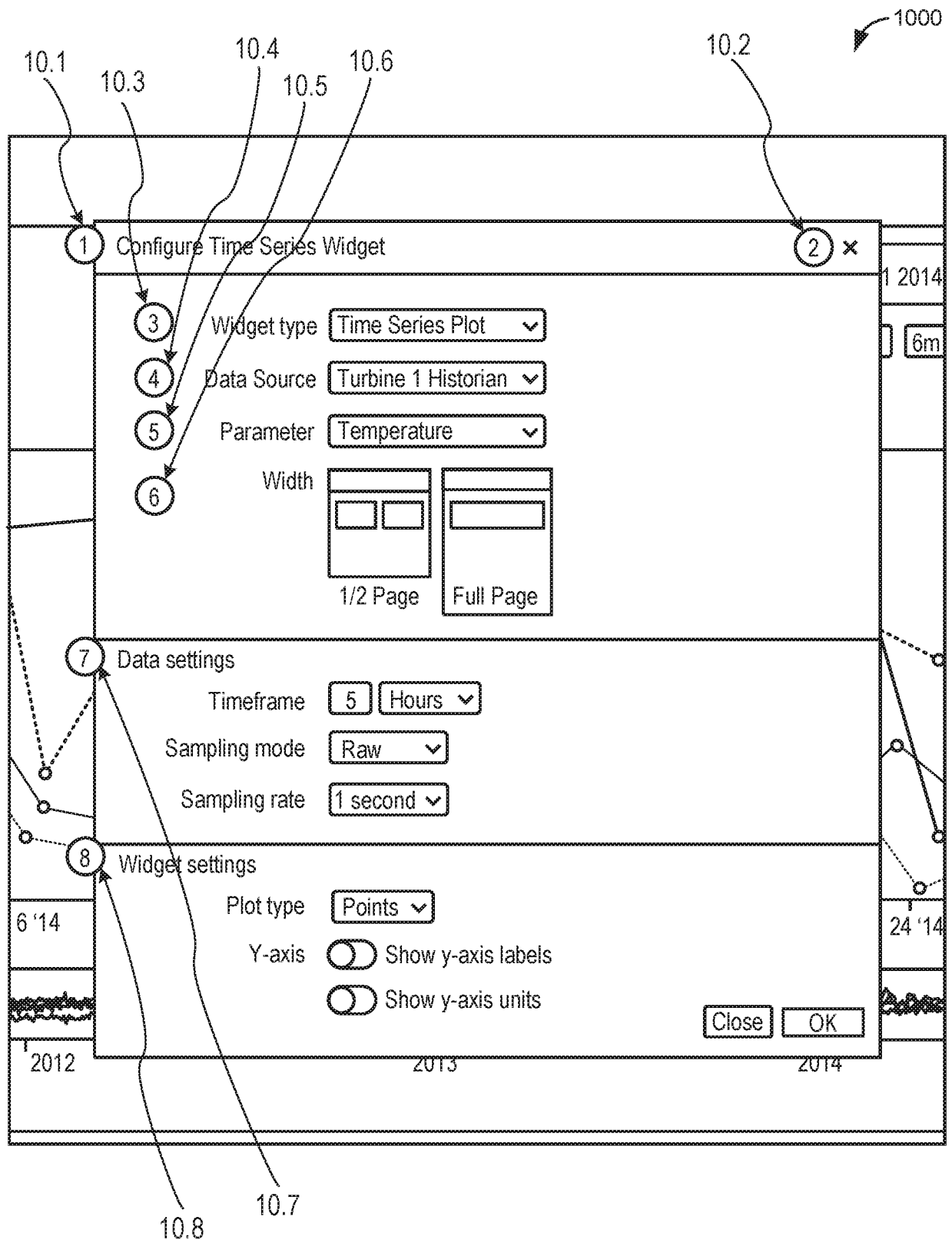


FIG. 10

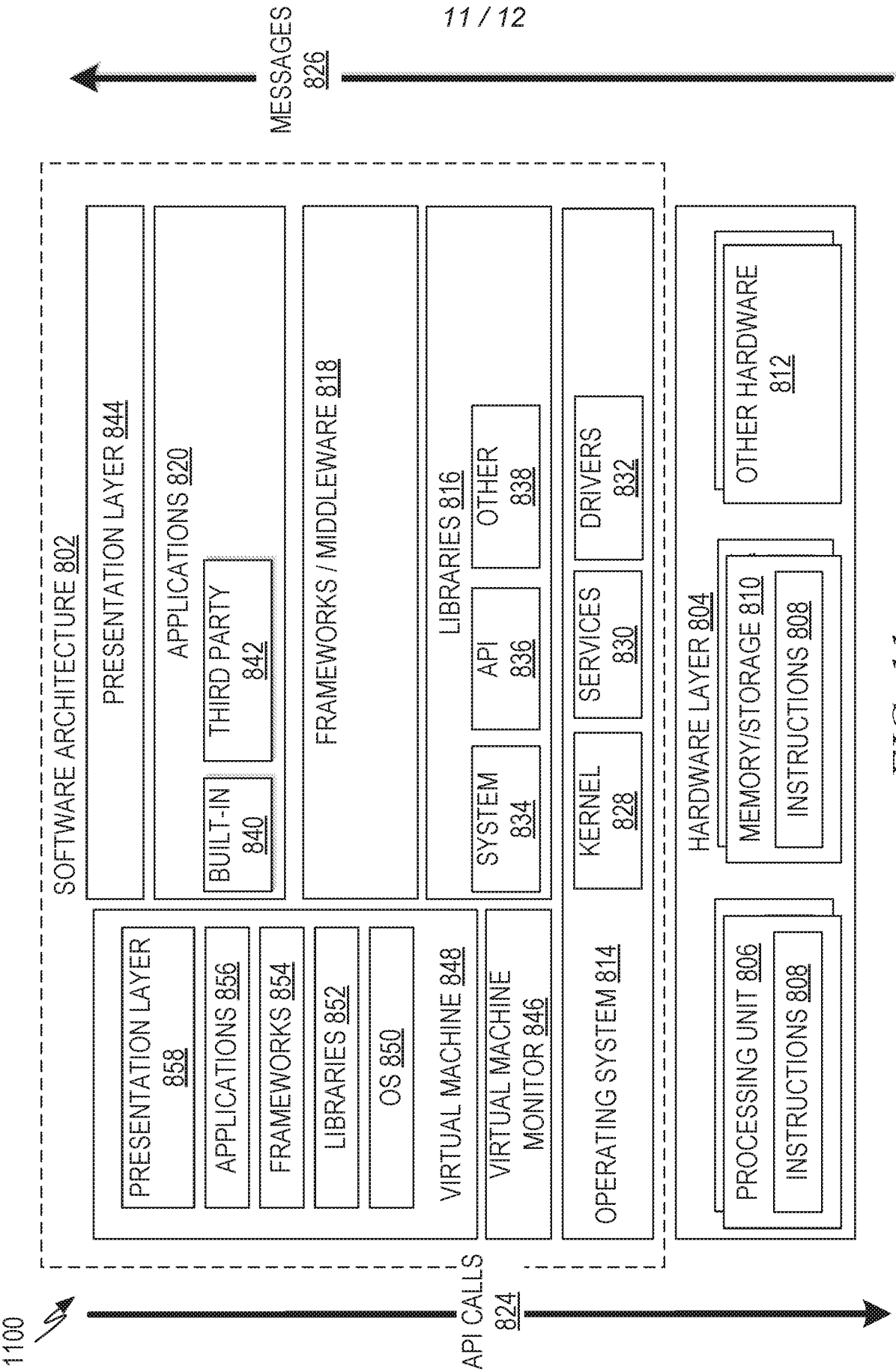


FIG. 11

1200

12 / 12

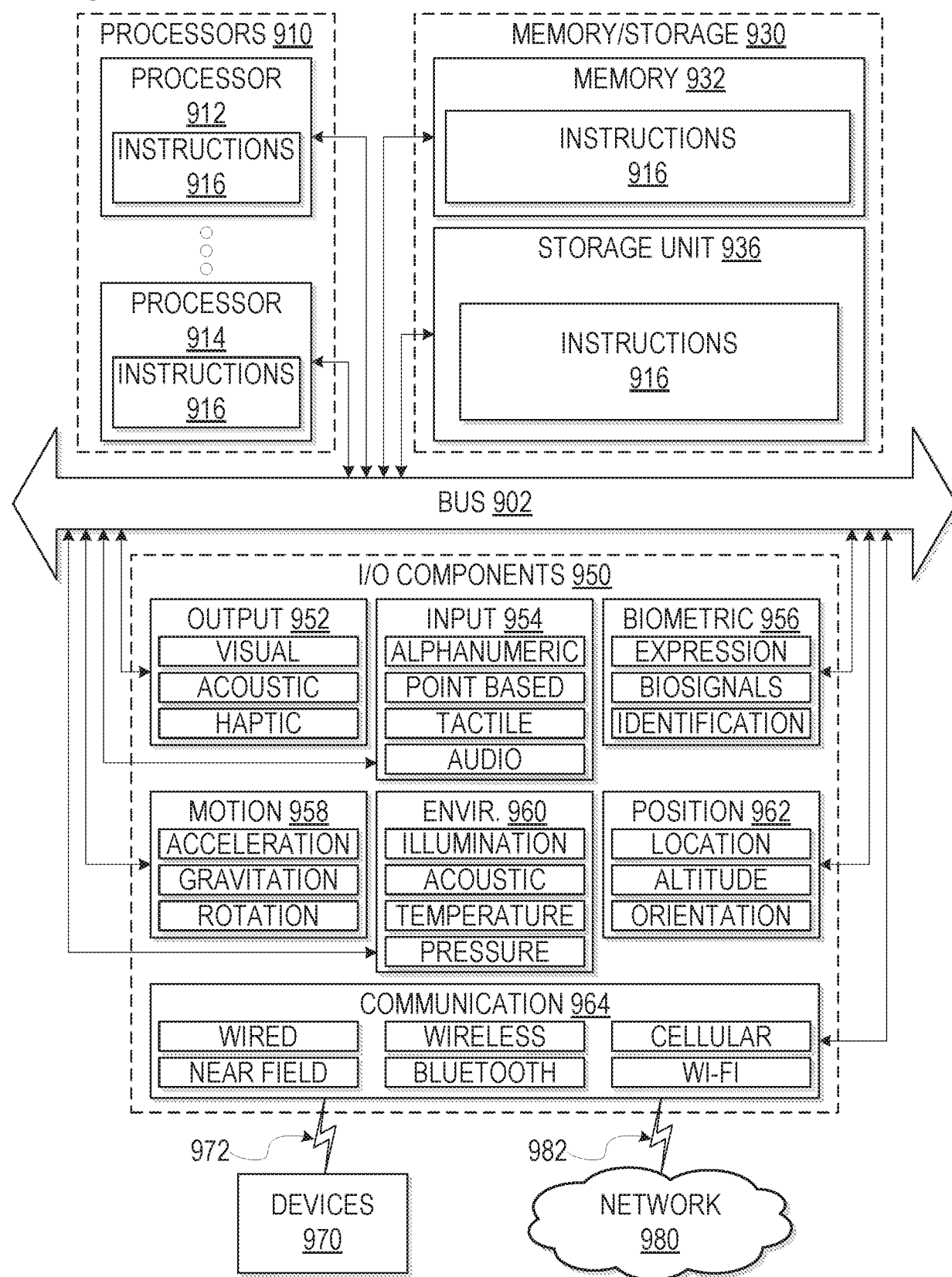


FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US17/18495

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G05B 13/02; G06K 9/20; G06T 7/20; B23Q 35/12; G06F 3/01, 17/30; G06N 5/04 (2017.01)
 CPC - G05B 13/0285, 19/02, 19/18, 19/4183, 19/41875; G06K 9/00771; G06T 19/006, 7/20, 7/246;
 B23Q 35/12; G06F 3/011, 3/017; G06N 5/048; G06Q 10/06, 10/087, 10/10, 30/02; H04N 7/18

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

See Search History document

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

See Search History document

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

See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012/0242648 A1 (BAIER, J et al.) 27 September 2012; abstract; figures 1, 11, 22, 37-39, 60, 63; paragraphs [0007], [0053]-[0055], [0060], [0066], [0068], [0069], [0072], [0077], [0092]-[0094], [0111], [0122], [0154]-[0155], [0176], [0180], [0193]-[0194].	1-20
Y	US 2015/0381737 A1 (DAVRA NETWORKS LIMITED) 31 December 2015; abstract; figure 1; paragraphs [0005], [0076]-[0082], [0087].	1-20
A	US 2014/0222813 A1 (EMC CORPORATION) 07 August 2014; entire document.	1-20
A	US 2003/0158795 A1 (MARKHAM, C et al.) 21 August 2003; entire document.	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

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

Date of the actual completion of the international search

07 April 2017 (07.04.2017)

Date of mailing of the international search report

10 MAY 2017

Name and mailing address of the ISA/

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
 P.O. Box 1450, Alexandria, Virginia 22313-1450
 Facsimile No. 571-273-8300

Authorized officer

Shane Thomas

PCT Helpdesk: 571-272-4300
 PCT OSP: 571-272-7774