

(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2006201516 B2**

(54) Title
Service delivery platform

(51) International Patent Classification(s)
G06Q 50/00 (2006.01)

(21) Application No: **2006201516** (22) Date of Filing: **2006.04.11**

(30) Priority Data

(31) Number	(32) Date	(33) Country
60/670,114	2005.04.11	US

(43) Publication Date: **2006.10.26**

(43) Publication Journal Date: **2006.10.26**

(44) Accepted Journal Date: **2012.05.03**

(71) Applicant(s)
Accenture Global Services Limited

(72) Inventor(s)
Hwang, Hans;lkiz, Cenk

(74) Agent / Attorney
Lesicar Murray Trento, PO Box 2545, Kent Town, SA, 5071

(56) Related Art
US 2004/0111506
US 2003/0005034

ABSTRACT OF THE DISCLOSURE

A service delivery platform is disclosed, including method and apparatus for managing the delivery of a variety of services to customers or subscribers. Exemplary services that may be managed using the service delivery platform include telecommunication services such as cable, wire line and wireless services. The service delivery platform creates, hosts, and manages services over different channels and end user devices. The implementation of the service delivery platform will dramatically simplify service deployment and management, and allow an enterprise to develop new services more rapidly with the reduced development efforts.

AUSTRALIA

Patents Act

COMPLETE SPECIFICATION
(ORIGINAL)

Class Int. Class

Application Number:
Lodged:

Complete Specification Lodged:
Accepted:
Published:

Priority

Related Art:

Name of Applicant:

Accenture Global Services GmbH

Actual Inventor(s):

Hans Hwang, Cenk Ikiz

Address for Service and Correspondence:

PHILLIPS ORMONDE & FITZPATRICK
Patent and Trade Mark Attorneys
367 Collins Street
Melbourne 3000 AUSTRALIA

Invention Title:


SERVICE DELIVERY PLATFORM

Our Ref : 770133
POF Code: 1058/474312

The following statement is a full description of this invention, including the best method of performing it known to applicant(s):

2006201516 11 Apr 2006

2006201516 11 Apr 2006

SERVICE DELIVERY PLATFORM

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/670,114, filed April 11, 2005.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND

[0003] The present application relates generally to provision of services such as telecommunication services. More particularly, the present application relates to a service delivery platform and method to improve the delivery of telecommunications services.

[0004] Service providers provide services to their customers in response to customer orders, change requests and other processes. One particular class of service providers is telecommunications service providers, which provide telecommunication services to their customers, referred to as subscribers. Telecommunications services currently include both wire line and wireless technologies. Examples of wire line telecommunication services include telephone service and related services such as voice mail, call forwarding, three way calling and caller identification, or cable television service and associated cable-provided services, such as internet access. Examples of wireless telecommunication services include cellular telephone service and associated services such as voice mail and three way calling, wireless electronic mail and paging.

[0005] Provision of such services includes many aspects. The service must be initially provisioned, or established. This can include establishing an account for

charging and billing the subscriber. This can include setting up service provision channels, such as designating a telephone number or other unique resource for the subscriber, providing dedicated hardware and software for the subscriber, such as installing a telephone line. After provisioning, if the service is provided on a per-unit basis, the service must be metered to measure the quantity of the service consumed. Bills must be generated and payments credited. If additional services are subscribed, the additional services must be provisioned and the account adjusted accordingly. Each change requires updating the equipment and software involved in service provision.

[0006] Recently, there has been a significant interest in the Service Provider (SP) industry to converge or consolidate different media such as voice, data, and video into single enterprise architecture. This would enable the enterprise to deliver value-added services such as unified communications, video and media rich solutions, and collaboration and knowledge sharing.

[0007] Previous systems for provisioning new services and updating existing services have required a new service setup for each service added. Thus, if a telecommunication service provider has in place telephone voice service to a number of subscribers and decides to add a new voice mail service, the telecommunication service provider must develop new hardware and software for implementing the system, a new billing process and other related features of the new service. If further services are developed for offer to subscribers, the process must be repeated for each new service or modification of a service. Repeating the process is expensive, time consuming and can delay the introduction of new services or features.

[0008] Moreover, a technical problem is created for service providers who seek to expand their offerings to different services or to extend existing services to add new features. Legacy systems may be implemented using hardware and software which are fully functional but no longer supported or no longer the best choice. Such legacy systems may be technologically incompatible with new equipment for providing services which are to be packaged with a legacy service. New implementations of services or features are preferably built on advanced

equipment and software to leverage current technologies, efficiencies and standardized tools. Getting legacy systems to work well with newly installed systems may be difficult to accomplish technically and may require substantial time and cost. As one example, many current telecommunications services are provided using the standardized Advanced Intelligent Network. Newer equipment makes use of the infrastructure of the internet, which employs a data transfer protocol such as transmission control protocol/internet protocol (TCP/IP). Even a technical interworking of new and legacy systems is possible, the resulting system may not include all desired functionality and may not be optimized for the service provider's needs. The final product or suite or products may be limited by the limitations of the incorporated legacy system.

[0009] Accordingly, there is a need for an improved system and method for delivering services to existing subscribers and new subscribers. There is a technical need for a system and method which allow a variety of technically different service provision systems to operate smoothly and efficiently.

BRIEF SUMMARY

[0010] According to a first aspect of the invention, there is provided a system for managing delivery of products, features and services, the system including:

a service catalogue including a set of products, features and services for provisioning and delivery to service subscribers; and

an order management module in data communication with the service catalogue, wherein the order management module enables provision and delivery of selected products, features, and services requested by a service subscriber, wherein the order management module decomposes the selected products, features, and services into atomic work orders workable by the service providers and releases the work orders to the service providers associated with the selected products, features and services, wherein the order management module maintains relationships across the work orders to determine dependencies for work order

release to the service providers associated with the selected products, features and services.

[0010a] According to a second aspect of the invention, there is provided a service delivery platform (SDP) for managing delivery of a plurality of unrelated services by one or more service providers to service subscribers, the SDP including:

a service orchestration and integration environment which manages work flow for services, manages business processes for the service providers and integrates constituent applications, the service orchestration and integration environment operable for providing and delivering selected services requested by a service subscriber by decomposing the selected services into atomic work orders workable by the service providers and releasing the work orders to the service providers associated with the selected services, wherein relationships are maintained across the work orders to determine dependencies for work order release to the service providers associated with the selected products, features and services;

a service creation environment for creation and modification of services, including a service provider interface to receive provider service definition data about provisioning and delivery of a first selected service from a service provider;

a service subscriber interface to receive subscriber service definition data about provisioning and delivery of a second selected service from the service subscriber;

a service deployment environment to automatically deploy the selected services;

a service assurance environment to identify faults in services; and

a unified directory accessible by the service orchestration and integration environment, the service creation environment, the service deployment environment, and the service assurance environment, the unified directory including a standardised structure that allows services to be dynamically configured based on the provider service definition data and the subscriber service definition data.

[0010b] According to a third aspect of the invention, there is provided a service delivery method including:

at a server, receiving a request from a service provider for creating a new service that is not in a catalogue of services, wherein the catalogue of services has a standard service catalogue structure for a plurality of telecommunications services to be offered, the standard service catalogue structure defining elements to be included in new service definitions;

in response, at the server, requesting product definitions from the catalogue of services;

at the server, receiving the product definitions and providing the product definitions to the service provider, wherein the product definitions are provided in work orders and relationships are maintained across the work orders to determine dependencies for releasing the work orders to the service providers associated with the selected products, features, and services;

at the server, receiving a request for service definitions associated with a selected product definition;

at the server, receiving from the service provider service information for the new service based on the service definitions and accessing a stored common data model of a service delivery platform including the catalogue of services, the received service information including the elements defined by the standard service catalogue structure;

at the server, automatically creating facilities for providing the new service, including extending the common data model with information about the new service, the provider of the new service and subscribers to the new service, including the received elements defined by the standard service catalogue structure; and

at the server, receiving a request from a service subscriber to initiate provision of the new service to the service subscriber.

[0010c] According to a third aspect of the invention, there is provided a service delivery platform for telecommunications services, the service delivery platform including:

a service creation environment accessible by service providers and service subscribers to create the telecommunication services;

a service deployment environment for managing the telecommunication services in one or more networks;

a service enablement environment to execute the telecommunication services;

a service orchestration and integration environment operable to decompose selected telecommunication services requested by a service subscriber into atomic work orders workable by the service providers, to release the work orders to the service providers associated with the selected telecommunication services, and to track and monitor the provision of the telecommunication services, wherein relationships are maintained across work orders to determine dependencies for work order release to the service providers associated with the selected products, features, and services; and

a common data model accessible by each respective environment for creating and modifying services and accounts of the service delivery platform, the common data model arranged according to a standardized structure for services for specification of elements of a telecommunication service.

[0010d] The invention will now be described in further detail with reference to particular embodiments. It is to be understood that the particularity of the embodiments does not supersede the generality of the foregoing description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 is a block diagram of a service provider system;

[0012] FIG. 2 is a functional block diagram of a service delivery system;

- [0013] FIG. 3 is a functional block diagram of a service delivery platform for use in the service delivery system of FIG. 1;
- [0014] FIG. 4 is a physical architecture diagram of a service delivery platform;
- [0015] FIG. 5 is an interaction diagram illustrating a process of changing features of an existing service using a service delivery platform;
- [0016] FIG. 6 is an interaction diagram illustrating a process of creating a new offer using a service delivery platform;
- [0017] FIG. 7 is an interaction diagram illustrating a process of creating a billing feed using a service delivery platform;
- [0018] FIG. 8 is an interaction diagram illustrating a process of providing a value-added reseller with a customer view using a service delivery platform;
- [0019] FIG. 9, including FIG. 9a and FIG. 9b, is an interaction diagram illustrating a process of creating a new account in an enterprise environment using a service delivery platform;
- [0020] FIG. 10 is an alternative interaction diagram illustrating a process of creating an account with two new services using a service delivery platform;
- [0021] FIG. 11 is an interaction diagram illustrating a process of authenticating and authorizing an end user of services using a service delivery platform;
- [0022] FIG. 12 is an interaction diagram illustrating a process of handling a provisioning exception when provisioning services using a service delivery platform;
- [0023] FIG. 13 is an interaction diagram illustrating a process of changing offer properties of an offer of services by a service provider, using a service delivery platform; and.
- [0024] FIG. 14, including FIG. 14a, FIG. 14b, FIG. 14c and FIG. 14d, illustrates an object model for use in conjunction with a system and method like the present embodiments.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

[0025] The present invention provides a technology architecture that enables efficient and rapid creation and support of applications by service providers who provide or seek to provide services to subscribers. The architecture standardizes the interfaces linking applications to operational and back office support systems and network capabilities. This reduces the time required for the service creation process and gets products to market faster. It also reduces the cost of service creation and support. The architecture provides a common, reusable set of development tools. It allows reuse of discrete and modular functionality across the entire platform. It has the capability to define and apply common control policies to applications. And it has the capabilities to define and manage integrated profiling of users, services and resources. These all provide greater flexibility in bundling and packaging features and applications to create new services.

[0026] In one embodiment, the present invention provides a service delivery platform which is modular, flexible and designed to support the delivery of value-added services. The service delivery platform includes a suite of plug-and-play solutions built on a standardized platform, such as the Microsoft .NET platform, available from Microsoft Corp., Redmond, Washington. This platform is optimized for delivery to SMBs as managed services. The service delivery platform further provides a modular, end-to-end service delivery platform that substantially automates sales, order management, provisioning, billing and operations management. The service delivery platform relies on a reusable, component-based model that maximizes reuse and flexible service bundling.

[0027] The service delivery platform can be extended to the widest variety of services, service providers and customers. Examples of telecommunications services that can beneficially be implemented in accordance with the service delivery platform include hosted Microsoft Exchange messaging services, including mobile electronic mail (email), facsimile and short message service and email archiving; hosted Microsoft Sharepoint, which is a system allowing

network-based collaboration within an organization and with customers and business partners; Unified Messaging/Unified Communications (UM/UC); HiPath Openscape, a multimodal communications application from Siemens Enterprise Networks, San Jose, California; web hosting; media on demand; secure instant messaging; web conferencing and live meetings (LM); and Voice Over Internet Protocol (VoIP); Microsoft Customer Relationship Manager (CRM); Microsoft Great Plains, which provides capabilities for financial management, distribution, manufacturing, project accounting, human resource management, field service management, and business analytics; and television over Internet Protocol (IP TV), and others applications. This list is intended to be illustrative and not exhaustive.

[0028] A system in accordance with the present embodiments reduces the amount of work required to implement new services by incorporating the functions common to all managed services into the service delivery platform itself. For example, for developing core service functionality, common routines such as logging, metering, tracking, etc., are well-defined and available for use by the service. For developing portal screens for the user interface (UI) for a service, pre-existing WebParts, from Microsoft Sharepoint, reduce the UI development time. For adding a new service description to a catalog of services, the service delivery platform provides a standardized service catalog structure which defines elements to be included, which ensures seamless integration with the rest of the service. Further, for creating provisioning logic and workflows, a rules engine based on Microsoft BizTalk simplifies the effort required to implement workflow. Existing workflows provide a starting point for implementing new workflows. For creating new interfaces that may be required by the new or expanded service, the system in accordance with the present embodiments isolates interface complexities from developers and allows reuse of existing interfaces. For providing interface for a service to a billing platform, the system features an interface to a billing platform which is architected to minimize changes with additional services. For creation of service specific reports, existing reports can be used as a starting point. Well-defined data structures reduce development time. Further, for establishing

operations processes for a new service, the system allows the additional service to suggest modifications to existing operations support processes, rather than establishing new processes. These features substantially improve the efficiency of establishing a new service or expanding a service.

[0029] Referring now to the drawing, FIG. 1 is a block diagram of a service provider system 100. The system 100 includes customer-side components 102 and system-side components 104. The components communicate over an enterprise messaging backbone 106.

[0030] The customer-side components 102 include a customer self-service function 108, a customer management system 110, an order manager 112, a usage manager 114, and a customer database manager 116. The customer side components 102 provide order entry, billing and customer care functionalities. The self-service function 108 allows a customer or service subscriber to view or modify an account without assistance and intervention by an agent of the service provider. This could be through World Wide Web access by the subscriber or by actuation of a telephone menu system. The order manager 112 manages the provisioning of new or changed services for subscribers. The usage manager 114 monitors usage of the service, for example, to ensure the customer only accesses services in accordance with his subscription. The customer database manager 116 manages the content of and access to the customer database. Other components may be included or substituted depending on the particular services provided or the particular service provider.

[0031] The system-side components 104 include a network 120, a third party supplier gateway 122, a billing and invoicing system 124, an enterprise manager 126, external interfaces 128, and a service delivery platform 130. The network 120 includes the service delivery infrastructure of data processing equipment and associated software. The third party supplier gateway 122 is a portal permitting service providers other than the service provider associated with the system 100 to provide services to customers. The billing and invoicing system 124 receives information about the nature and amount of services used, terms of service between the service provider and the subscriber and generates a bill for the

subscriber. The enterprise manager 126 provides overall system control. The external interfaces 128 allow the system 100 to communicate data and information with other sources.

[0032] The service delivery platform 130 creates, hosts, manages, controls, and deploys the value-added services provided by the system 100. The service delivery platform 130 operates as a system within the larger service provider system 100. The service delivery platform 130 interacts with the entire system 100 through the enterprise messaging backbone 106. The enterprise messaging backbone provides interfaces and process management over the entire network, bridging system and information in order to provide complete and consistent operations and management. The service delivery platform 130 will be described in greater detail below.

[0033] FIG. 2 is a functional block diagram of a service delivery system 200 including the service delivery platform 130 of FIG. 1. The service delivery system 200 includes the service delivery platform 130, a service activation function 202, a service integration and management function 204, an operations support system 206, business support systems 208 and service platforms 210. The SDP 130 is a core engine that encapsulates support services for value added services that might be offered by the service provider. The SDP 130 includes service creation, service deployment, service enablement and service assurance capabilities as will be discussed below.

[0034] The service activation function 202 permits initiation of service provision to one or more customers. It presents a user interface or other access portal to a customer of the service provider or an agent or operator working on behalf of the service provider to activate the service. This function 202 also allows subsequent modification of the service, such as adding or deleting capabilities. The customer may access this function 202 himself, such as over a World Wide Web interface or through a telephone menu system. Alternatively, the customer may engage the operator in a phone call or live online session, or by any other means, to specify the service requirements.

[0035] The service integration and management (SIM) function 204 includes a customer relationship manager (CRM) front end 212 and a portal 214. The SIM function 204 allows users, both internal users such as operators and customer relations managers and external users such as customers the ability to access and use SDP services. The CRM front end 212 provides the hardware and software for interacting with a customer or operator. Any suitable customer relations management device or service may be used. Examples include call center software and hardware which allows customers to speak by telephone with operators and allows the operators, in turn, to activate and update service provisioning using the service delivery platform 130. To this end, the CRM front end 212 interacts with a CRM interface web service 216 to access data and other information of the SDP 130. The SDP 130 has a counterpart CRM integration adapter 218 which communicates with the CRM interface web service 216, preferably using internet protocol (IP) or other standard data communication.

[0036] The portal 214 allows outside users access to the SDP 130 and includes a portal interface web service 220. The SDP 130 has a counterpart portal integration adapter 222. Again, the portal web service 220 and the portal integration adapter 222 preferably use internet protocol (IP) or another standard. Customers may utilize external customer relation management (CRM) systems to request service activation. The customers will purchase product services or packages offered by service catalog for the external systems, not for the SDP. The portal 214 allows communication between the SDP 130 and these external systems.

[0037] The operations support system (OSS) 206 provides inventory control, monitoring, etc., and is integrated with the SDP 130. The OSS 206 includes an inventory/resources function 224 and a manager of managers (MOM) 226. The OSS 206 permits integration between the SDP 130 and new or pre-existing OSS systems for the enterprise. In one embodiment, the OSS 206 uses Microsoft's EAI middleware, BizTalk, to offer coded business events, a fundamental architecture backbone, workflow management, data requirements, and data transformation

rules required to integrate the service provider's Customer Relationship Management, Billing, and Provisioning Services.

[0038] The inventory/resources function 224 operates with an inventory/resources web service 228 which communicates with a counterpart inventory/resources adapter 230 of the SDP. Similarly, the MOM 226 operates in conjunction with a MOM interface web service 232 which communicates with a counterpart MOM integration adapter 234 of the SDP 130.

[0039] The business support systems 208 provide billing, customer relation management functions, etc., and are integrated with the SDP 130. The business support systems (BSS) 208 include a billing function 236, an invoicing function 238, a customer insight function 240 and a customer relation manager 242.

[0040] The BSS 208 receives billing records and other information about the provision of services to the customer by the service provider. Billable data records as created by a metering service are submitted to the back-end billing system 236 for billing and real-time charging purposes. Billing integration with BSS allows operators to effectively monetize their services to customers based on their service usage and to prevent revenue leaks due to missing service events.

[0041] The billable data records submitted to the operator's business support systems are also be used by the invoicing function 238, the customer insight function 240 and the customer relation manager 242. The invoicing function 238 uses the billing data records and other user account information to prepare and send invoices to customers. The customer insight function 240 extracts information about customer behavior, purchasing patterns and so forth. For example, this function 240 enables service providers to analyze service usage activities of subscribers based on access history in order to generate abstract usage patterns and supply such valuable insight to value-added services for fine-tuning personalization engines. The customer relation manager 242 monitors controls the way in which interaction with a customer is tailored to each individual customer, based on other information about the customer, such as services purchased, payment patterns, etc.

[0042] The billing function 236 interacts with the SDP 130 using a billing interface web service 244. The SDP 130 has a counterpart billing integration adapter 246. Similarly, the invoicing function 238 interacts with the SDP 130 using an invoice interface web interface 248, which communicates with an invoicing integration adapter 250 of the SDP 130. The customer insight function 240 interacts with the SDP 130 using an insight interface web service 252, which has a counterpart insight integration adapter 254 of the SDP 130. Lastly, the CRM 242 interacts with the SDP 130 using a CRM interface web service 256, which in turn has a counterpart CRM integration adapter 258.

[0043] FIG. 3 is a functional block diagram of a Service Control / Order Management Component Architecture 300 for use in the service delivery system of FIG. 1. The component architecture 300 is divided up in the four Service Control / Order Management Component modules that make up a larger infrastructure interoperating with the Enterprise Messaging Backbone.

[0044] The Service Control / Order Management Component provides integrated subscriber and service management. It includes common logic that allows resources and services to be dynamically configured based on user profile, service definition, policy information, current resource allocation, and resource state.

[0045] This environment shall also consist of the control services that provide identity management for access control, metering service for creating billable data records, profile and personalization management, notification engine to support push-based alerts to customers, and content provider services that facilitate the integration of SDP with content providers.

[0046] These modules include a Service Deployment Environment 302, a Service Creation Environment 304, a Service Enablement Environment 306 and a Service Assurance Environment 308. In addition, in the illustrated embodiment, the component architecture 300 includes a Service Interaction and Management Environment 310. This environment 310 may be omitted from some embodiments of the SDP architecture. However, when included, it is tightly integrated to the

SDP architecture. The environment 310 provides the interfaces through which users interact with services.

[0047] The Service Deployment Environment 302 provides deployment tools and mechanisms and resource management to help system administrators to manage deployed services in a distributed network. This module provides resource management that helps system administrators to manage deployed services in a distributed network. As an example, it supports the integration with content distribution networks to distribute services to operators' edge networks on a global scale to enable customers to access services hosted by the nearest servers.

[0048] The Service Creation Environment 304 consists of several subsidiary components that allow for the creation of new services to be introduced to the Service Control Environment. Further, the Service Creation (SC) Environment 304 provides a development environment for software developers to create new services and to extend existing ones. The Service Creation Environment 304 simplifies the creation and reduces the time-to-market of new services. This module further provides the service catalogue that manages the services, an integrated development environment, and platform software development kits (SDKs) to facilitate the creation of new services. It also helps content providers create new content applications that leverage the services from the SDP.

[0049] The SC Environment 304 provides a development environment for software developers to create new services and to extend existing ones. It simplifies the creation and reduces the time-to-market of new services. Essentially, this environment creates necessary capabilities to host new services and deploy them to the SCM environment so that the customers can sign-up, configure and receive any information services that are hosted and managed by the SDP.

[0050] The SC Environment 304 is comprised of the following modules. These modules create and port-in any necessary capabilities to deliver new or updated data services to the customers;

Avanade Connected Architecture .NET 3.0 (ACA.NET)

Service Configuration Tool

Web Services based Service Directory
Integrated Development Environment (IDE)
Product-specific Software Development Kits (SDKs)

[0051] Integrated Development Environment (IDE) and development toolkit should be utilized to facilitate building of new features and services that utilize component oriented, service building blocks that to build new services. ACA.NET provides APIs to third party developers and includes deployment infrastructure. Service Configuration Tool facilitates building of Execution and Control and Management functions capabilities. In addition to the above modules, a testing environment can be set up to check for any errors before the new services are launched to the SCM environment.

[0052] ACA.NET is a set of application blocks that are generally built on the Microsoft.NET platform. ACA.NET complements and extends the .NET framework with common architecture and services. Additionally, the ACA.NET framework is built on the new development paradigm of Service Oriented Architecture (SOA). Additionally, ACA.NET is configurable based upon Aspect Oriented Architecture (AOA) which allows the behavior of an application to be configured via XML configuration files and attribute declarations instead of programmatically. ACA.NET is intended to abstract the infrastructure details of messaging between objects away from developers allowing the developers to concentrate on enabling functionality while ACA.NET generates the necessary plumbing.

[0053] ACA.NET is composed of a Service, Framework, and Aspect architectures. The Service architecture consists of the business logic and the communication infrastructure. The Service architecture enables messaging between objects via transports. Messages have a type and wire format specific to the system being built. Messages can be serialized and marshaled. The transports include web services, .NET Remoting, and in-process function calls.

[0054] The Framework architecture consists of “mainline” core functionality such as data access and manipulation and “sideline” functionality such as caching

and remoting. The Aspect architecture allows behavior declaration to instance objects that enable the Service and Framework architectures. Using ACA.NET, developers can quickly create and enable applications that can be easily deployed on the SDP platform. ACA.NET also has wizards to transform existing classes and components into Web Services without writing any new code. There are no API calls into the framework so misuse of the framework and bugs are eliminated or reduced. Also, services can be deployed over multiple transports without affecting service code. Lastly, application logic is protected from changes in communication technologies. Overall, ACA.NET 3.0 allows developers to create loosely coupled applications that communicate to compose large, complex systems.

[0055] For any application that a vendor or service provider intends to create, ACA.NET automatically generates the code for the sender and the receiver and the underlying transports that are used to communicate messages with the service layer, which is the core functionality that is developed by the vendor. Using this model, vendors can rapidly develop applications that can be seamlessly deployed within the SDP environment.

[0056] ACA.NET is based on Aspect Oriented architecture. Aspects are a declarative way to control the behavior of an application's functionality. Aspect attributes are set through XML configurations rather than programmatically through code. This reduces the amount of custom code and increases consistency throughout an application. An example of Aspect Oriented Architecture is Authorization. Virtually all applications must use some form of authorization, regardless of how it may be implemented. The basic functionality for authorization remains the same for all applications. Thus, using ACA.NET vendors can quickly configure global attributes for authorization then using minimal code can implement it as desired while certain core functionality remains centrally controlled.

[0057] The Service Enablement Environment 306 provides support for the execution of SDP and value added services. This includes service control, service orchestration and management and service execution. The Service Enablement

Environment 306 includes a Service Brokering and Integration Environment 312, a Service Control module 314 and Service Execution module 316.

[0058] Although the central core functionality of the SDP platform is the BizTalk integration, several areas of functionality must remain consistent throughout the solution. In order to meet immediate requirements and provide a scalable system that allows future enhancement, a standard platform must exist. This standard platform ensures common methods for data access, exception handling, security, and logging. The use of ACA.NET as a foundation ensures that development will be consistent and reliable throughout the application lifecycle. Additionally, configuration and maintenance will be reduced as well as shortening the development time.

[0059] ACA.NET can also help vendors extend their applications into other areas of the SDP platform by enabling them to quickly and efficiently write to Microsoft Exchange Server, Microsoft Active Directory, or Microsoft Operations Manager (MOM). For example, the ACA.NET Framework writes significant events to the event log, providing a way to debug the application during development and after deployment. It also helps administrators detect and diagnose problems when they occur. The critical events in ACA.NET are not only logged to the event log but also published as WMI events. The message in any of the event logs contains detailed information and is human readable. Each event ID is defined distinctly within each framework and therefore can be used as an event filter by monitoring tools such as MOM.

[0060] ACA.NET reports significant events within the frameworks through the publishing of WMI events. These events cover all the events that the ACA.NET Framework writes to the NT event log. The mechanism by which Windows collects performance data on various system resources is called the performance counter. Performance counters can provide useful information on how the system is running. They not only provide statistics on how much stress a system or application is under, but also allow the application throughput measuring. ACA.NET takes advantage of performance counters to expose the inner health and

stress of the framework to monitoring tools. Applications built with ACA.NET have critical performance counters built-in just by using the framework.

[0061] The events generated by the ACA.NET framework (for example, a Data Service configuration failure) are written to the event log by default. With the ACA.NET Logging Service you need the configuration to use the EventLogSink, run the Log Distributor, and then call WriteToLog in your code to write to the event log. The ACA.NET Framework instrumentation event log uses the name of the framework that the event originated from as the event source of the logged message. Using ACA.NET operators can quickly enable existing applications as web services and extend them to use server other server and system components much more efficiently than would be possible without ACA.NET. Additionally any new development can be developed more quickly and with more consistency, allowing developers to concentrate on the core services rather than communications within the application or within an integrated system.

[0062] The Service Brokering and Integration Environment 312 consists of service orchestration that provides rule-based process scheduling and integration services that integrate the service delivery platform with content management, BSS, OSS, and device management systems to provide end-to-end services to customers.

[0063] The Service Control module 314 provides integrated profile and service management. The Service Control module 314 includes common logic that allows resources and services to be dynamically configured based on user profile, service definition, policy information, current resource allocation, and resource state. Accordingly, the Service Control module 314 includes an Identity Management and Administration function 318, a Profile Management function 320, a Service Catalog 322, a Security Policy Management module 324 and a Reporting / Auditing function 326.

[0064] The Identity Management and Administration function 318 allows management and control of identity information for service subscribers. Operators need an identity management solution to prevent unauthorized access to their services and protect the privacy of customer data. By providing effective security

management for identity data, the identity management system also helps to prevent the revenue leakage problem that results from identity theft and malicious attacks, which has plagued operators on a global scale.

[0065] Accordingly, in one embodiment, the Identity Management and Administration function 318 includes an Access Management function, a Confidentiality function, an Identity Integration function, a Unified Directory and provision for Single Sign-on. Identity Administration applies to identity lifecycle management; that is, the provisioning and maintenance of digital identities for customers and content providers. The Access Management function is the service associated with establishing, enforcing, and managing the authentication, authorization, and auditing policies for customers and content providers. The Confidentiality function applies to protecting the privacy of connections between customers and services and securing identity data managed by SDP against unauthorized access. Identity data is often distributed in multiple systems within an operator's environment, or across different Internet communities. The Identity Integration function represents a holistic approach to the management of distributed but interrelated identities in separate repositories by providing a consistent, accurate representation of identities across heterogeneous systems while recognizing the identity ownership roles of individual systems. The Unified Directory is a consolidated or federated database for managing subscriber, service, and device information. Single Sign-on is a mechanism to verify a user across multiple applications through a single authentication challenge.

[0066] The Profile Management function 320 allows management of subscriber and other profiles. Information about subscribers, networks, and services are managed in profiles. A profile is a container of properties for a particular subject. Profile management function 320 manages the following profiles.

[0067] **Network Profile.** This profile contains properties about the operator's CDN that the SDP deployment environment supports for deploying services onto servers on edge networks. It also contains profiles for network elements in the operator's network including SMS, MMS, HLS, and others.

[0068] Subscriber Profile Each profile contains identity data about a subscriber, including the subscriber's registered account name and access credential, address, and other identity data. It also contains preferences, subscriptions, summarized usage patterns, and device configurations for the subscriber.

[0069] Content Provider Profile Each profile contains the identity data about a content provider, which is used by the content provider service to manage the access to the operator's SDP system from content providers.

[0070] Service Profile Each service deployed by the operator needs to be managed in profiles in order to facilitate operations management. The service profiles also facilitate access to each service from device applications.

[0071] The Service Catalog 322 includes a business level set of service and products that are used to manage packages and offers that are sold/resold to customers and resellers. The Service Catalog in one embodiment is organized as follows:

[0072] Product A Product is hardware or software used to provide a consumable unit of service. Each product is based on an application and/or hardware that enables services and service variants.

[0073] Service A Service is an atomic derivative of a Product (i.e., the minimum possible) that offers a consumable unit of service. For a given product, SDP can support one or many services.

[0074] Service Variant A Service Variant is a catalog unit derived from Service that contains configured details and attributes that define a set of unit of service. The system supports creating different variants of every service, so that the service is offered to different types of target users.

[0075] Offer An Offer is set of Service Variants for presentation to customers for purchase. Service offers consist of one or many Service Variants. It can also include other offers.

[0076] Service Group A Service Group is a container used to allow customers to group end-users and link the accounts to a service template. In this context, a customer is a service provider and an end user is a service subscriber.

[0077] Solution A solution is a set of service variants or offers that represents a customer's right for subscription. When a customer subscribes to an offer or service variant, that selection becomes a solution for that customer. The customer's end users or subscribers will be eligible to utilize the corresponding services that are included in the solution.

[0078] The Service Catalog provides all the necessary information about value-added services that can be hosted on the Accenture's SDP. It is a reference data source that will describe the product and feature information, network location, configuration, and access mechanisms of each service. SDP supports two different ways in which the service activation request can be submitted to the SDP:

The service activation requests can be created by bundling service catalog information from the SDP's portal and submitted directly to the SDP.

The customers can request service activation by purchasing products and packages from external CRM systems such as Siebel platform and send the requests to the SDP for processing.

[0079] The Service Catalog helps the customers who are requesting service activation or configuration directly from the SDP's portal by presenting all the selectable options and features for desired value-added service. The Service Catalog will be retrieved and configured into the SDP's portal so that the customers can "browse" through the catalog to select the products and features for the service activation.

[0080] The catalog elements can be selected by the customers on the portal and submitted to the SDP as part of a service activation request order. The order management module is responsible for pre-processing the service activation requests, and the Provision Engine picks up all the selected catalog items and provisions them as service products and features. During the service activation process, the workflow manager may trigger billing events based on the catalog

items selected by the customers to properly bill the customers for the products and features.

[0081] For the instances where the customers utilize external CRM systems to request service activation, the customers purchase product services or packages offered by the service catalog for the external systems, not for the SDP.

Therefore, the external catalog only provides product and feature information for service activation. The SDP's service catalog then complements this by providing other technical information about the services such as service configuration, network resources, and access mechanisms to deliver the services that are being activated. The SDP's service catalog should be comprehensive to accommodate the service activation requests sent by external systems.

[0082] The identity management solution for SDP provides four major functions:

Identity Administration

This applies to identity lifecycle management; that is, the provisioning and maintenance of digital identities for customers and content providers.

Access Management

This is the service associated with establishing, enforcing, and managing the authentication, authorization, and auditing policies for customers and content providers.

Confidentiality

This applies to protecting the privacy of connections between customers and services and securing identity data managed by SDP against unauthorized access.

Identity Integration

Identity data is often distributed in multiple systems within an operator's environment, or across different Internet communities. Identity integration represents a holistic approach to the management of distributed but interrelated identities in separate repositories by providing a consistent, accurate representation of identities across heterogeneous systems while recognizing the identity ownership roles of individual systems.

[0083] The Service Control module 314 further includes the Security Policy Management module 324. Within this the SDP, the Security Policy Management module 324 provides a single authoritative, integrated and federation directory, provided via Active Directory and interoperating with an SDP Access Manager to ensure that capabilities and information with the platform are relevant and private.

[0084] The Security Policy directory may be characterized by a Role and a Privilege. A Role is a business classification assigned to a customer or account that is defined by a set of characteristics or privileges. Usage of the Role in SDP allows for specialized distinction of abilities to represent a business function, such as a value added reseller (VAR) as a role, where the VAR role represents each of the abilities that a VAR or accounts of a VAR is able to perform. A Privilege is a business characteristic or right that is represented in SDP. An example of this may be the right to create an account or over-ride service pricing. The grouping of privileges defines the Role and within SDP, the usage of Role and Privilege is extremely flexible and is not defined via inline code. What this does is allow SDP to define new roles and new privileges as extended capabilities and new business agreements are formed.

[0085] The Reporting / Auditing function 326 provides standardized logging functions to support reporting and audit functions. In one embodiment, this is characterized by the following:

[0086] **Activity / Availability:** SDP activity and availability reporting. This may include service uptime, transactions/sec, active users, time based activity, etc.

[0087] **QoS / GoS:** Quality of Service and Grade of Service reporting. This can include reporting over platform and over value added service (VAS).

[0088] **Service Usage:** VAS usage reporting, including either VAS specific or platform general information. Platform specific reporting includes information such as subscription to service, most common used services and most common used features.

[0089] **Events / Auditing:** SDP events and auditing logs. This includes, for example, a number of failed login attempts, login history, suspension history, etc.

- [0090] **Performance:** reporting on platform health. This includes metrics-based reporting on counters such as average provision time, number of expired processes, number of retry attempts, active accounts, etc.
- [0091] **Resource:** Reporting over SDP physical and logical resources that cover each Service. This may include quota levels, resources exceeding critical level, resources over utilized, resource over time, etc.
- [0092] **End-User:** End-User reporting, number of services, subscribed service health.
- [0093] **Customer:** Customer reporting, number of services, number of accounts, service usage, resource usage, etc.
- [0094] Other types of reporting and reporting content may be included or substituted as well.
- [0095] The Service Execution module 316 forms a standardized framework and architecture to execute services for a subscriber or an application. The Service Execution module 316 encourages reuse of existing components such as a VXML media server. The Service Execution module 316 leverages a set of common services such as authentication, authorization, presence, and personal information management functions. The Service Execution Module 316 includes in one embodiment Service Gateways 328, Service Building Blocks 330 and Infrastructure 332.
- [0096] The Service Gateways 328 are responsible for managing solutions for customers and value-added resellers (VARs). A Solution is a container for the Service Variants and Offers selected by a Customer to be reserved for their Users. In the illustrated embodiment, there are two types of service gateways. The first type is Third Party Service Access Gateways, which extend the service delivery platform to support third party retail and wholesale service providers such as content providers, gateway providers and retail channels. The second type is Network & Service Gateways which provide interfaces to network and service elements to allow higher layer service logic to access them in a standardized and protected way.

[0097] The Service Building Blocks 330 provide a set of common services and resources that are generally common to all SDP value added services. The Service Building Blocks 330 provide synergy across products and simplify handling of special customer requests. Several exemplary building blocks are described below. Many others may be included as well.

[0098] A Session / Signaling building block provides real-time management of a service session for data, voice, and video applications. This building block also signals the appropriate network and service elements to set up one or more sessions in order to deliver service. Further, the Session / Signaling building block maintains the session as required to support the service. Examples include a call agent using session initiation protocol (SIP), H.323 data communication protocol, Media Gateway Control Protocol, etc., SIP proxies, a Video on Demand (VoD) server, SS7 to PSTN and mobile networks, etc.

[0099] A Content Management building block provides content provisioning and management capabilities including Digital Rights Management (DRM). Further, this building block may be shared across services such as internet protocol television (IPTV), VoD) and it provides provisioning and management application programming interfaces (APIs). Also, the Content Management building block provides a mechanism for real-time interaction with content by working with security access authentication modules such as “authenticate user,” “check parental controls” and other access, providing uniform resource locators (URL) to a content delivery network (CDN) and correctly encoded content, and providing a DRM key to access content.

[00100] A Notification Engine building block provides a high-performance platform for scalable notification services that generate messages customized to meet target subscribers' specific needs. The notification messages, or alerts, can originate from a variety of sources such as external data feeds and data updates by value-added services, and they can be delivered to various devices including pocket PCs, PCs or mobile telephones through multiple communication channels. Typical notification messages include those for appointment reminders, flight delays, and availability of new service or content. The Notification Engine in one

embodiment is an orchestrated process that supports event collection, event generation, event distribution and subscription.

[00101] A Presence / Location building block provides common service for managing state and context information for users, applications, and devices, such as IP address, bit rate, network location, device, preferences, etc. Further, the Presence / Location building block keeps track of current user and service profiles including user preferences and subscribed service profiles. Also, the Presence / Location building block forms a centralized or federated repository for managing user and service profiles including features and parameters needed to execute services. Still further, the Presence / Location building block provides protocols and interfaces for accessing and updating presence/state information. Examples of such protocols include Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE), Extensible Messaging and Presence Protocol (XMPP) and Java Message Service (JMS). Examples include a Class 5 Switch subscriber/service database, a Remote Authentication Dial In User Service (RADIUS) server, a Wireless Home Location Registrar (HLR) and the Third Generation Partnership Project IP Multimedia System (3GPP-IMS) Home Subscriber Server (HSS).

[00102] A Security Services building block provides common authentication and authorization services that can be shared across applications and services platform.

[00103] A Presentation / Rendering building block provides real-time data transformation and formatting to support different types of devices such as extensible markup language (XML), hypertext markup language (HTML), and wireless markup language (WML). The Presentation / Rendering building block also provides format conversion functions on behalf of servers and devices.

[00104] A Metering Service building block acts as a mediation system between the service providers and billing systems. This building block is an orchestrated process that supports; event collection, aggregation, transformation, tracking and submission. This building block receives or captures events in call detailed record (CDR) format from other services for tracking and billing

purposes. The building block further integrates with the operator's billing system by converting CDR data to billable data records and submitting them to the operator's billing system for real-time charging and billing purposes. Further, the Metering Service building block supports multiple units of measurement by event type, volume (in bytes), time, locality, and content type, based on business requirements. Preferably, the Metering Service building block supports the use of Internet Protocol Detailed Record (IPDR) as the standard data format for billable data records and supports Charging Control which allows integration with billing engines to support pre-paid environments such as real-time signaling networking and post-paid environments, and combinations.

[00105] The Metering engine receives or captures events in call detailed record (CDR) format from other services for tracking and billing purposes. It integrates with the operator's or service provider's billing system by converting CDR data to billable data records and submitting them to the operator's billing system for real-time charging and billing purposes. It supports multiple units of measurement by event type, volume (in bytes), time, locality, and content type, based on business requirements.

[00106] The metering engine is an orchestrated process that involves the following tasks:

Event Collection

Receives and captures events from other services and SDP. Product catalog pricing and subscription level information will be captured and input into CDR in a consistent and routine manner. Collection data will include detailed information of bundle, product, pricing, discounting, and even description.

Aggregation

May aggregate multiple CDRs or consolidate one CDR with other data into one billable data record. CDRs will be grouped during each billing cycle linked by shared account number that the billing system will also use or reference.

Transformation

Transforms CDR data into billable data record formats accepted by specific billing systems. This transformation also includes the creation of summary level billing items, such as discounting totals, grand totals, usage & tiered based pricing.

Tracking

Records each CDR for tracking and auditing purposes. Each process in the billing cycle event are tagged with a tracking code or status. These tracking codes are used to properly handle the work process in the processing of all events and billable accounts in the SDP solution.

Submission

Submits billable data records to billing systems through the BSS Integration Service. The SDP solution submits in a configurable schedule and contain run-time business logic to determine the integration source for generated bill records.

[00107] To provide operators maximum flexibility in integrating diverse billing systems, the metering service supports the use of Internet Protocol Detailed Record (IPDR) as the standard data format for billable data records. IPDR is a standard defined by IPDR.Org, an open consortium of leading service providers, equipment vendors, system integrators, and billing and mediation vendors collaborating to facilitate the exchange of usage and control data between network and hosting elements and business support systems. IPDR defines an XML-based schema to represent CDR details for billing purposes, and it is widely supported by many leading billing systems vendors.

[00108] A Media Services building block provides media access and delivery services and provides a request/response mechanism for accessing multi-media such as audio, video and data. Further, the Media Services building block supports protocols for delivering media to devices, applications, and users, and provides media transformation capabilities and encoding and decoding services.

Examples include a VoD server, CDN, Broadcast TV head-end, a Web server using HTTP/HTTPS, an audio server and a VoiceXML Media Server.

[00109] A Media Gateways building block provides a translation function between media streams for different networks and clients. Examples include a voice over internet protocol (VoIP) to time division multiplexing (TDM) media gateway and other vocoding devices.

[00110] A Bridging building block bridges media such as voice, video and data for multi-user collaboration. The building block preferably ideally supports reformatting to different devices such as Media Gateway functions. Examples include voice conference bridges, data conferencing such as the facilities provided by WebEx, PlaceWare and MeetingPlace, and video conferencing.

[00111] A Naming / Routing building block provides a name resolution mechanism for identifying services and resources in networked environment. Further this building block provides routing service to determine the optimal path and allocates networked resources. Still further, this building block dynamically assigns resources to devices and servers.

[00112] A Messaging building block provides various forms of messaging between addressable users, devices, and applications and provides protocols, routing, and data stores to deliver messages. Further, the building block provides channel and routing mechanisms to support notification services. Examples include electronic mail (SMTP, POP3, IMAP), instant messaging, voice mail, fax and short message service (SMS).

[00113] A Synchronization Services building block manages and synchronizes information between applications and servers that need to use similar information. Examples include a Personal Information Manager (PIM) calendar and contacts across devices, such as Microsoft ActiveSync.

[00114] The Infrastructure 332 of the Service Execution Module 316 allows for a set of Next Generation Network (NGN) services to facilitate support of emerging and future value added services. The Infrastructure 332 includes and is not limited to in an all-IP network approach to the support for Data, Voice, Wireless, Rich Media Content, and Unified Messaging Services.

[00115] The Service Brokering and Integration Environment 312 provides support for the execution of Service Control / Order Management and value added services. This environment 312 includes Service Control, Service Orchestration and Management and Service Delivery. The environment 312 includes several modules, including a Process Management module 334, an Order Management module 336, a Provisioning Management module 338, a Content Management module 340, a Network Integration module 342, a Device Integration module 344, and a OSS/BSS Integration module 346.

[00116] The Service Brokering component in SDP is a set of core service control elements designed to control network resources and coordination of services. Resource control will allow fundamental control to the Service Execution infrastructure for management of ports, server connections, and other when customers access service resources. The service resources will work to monitor and control quality of service (QoS) to allow for efficient usage of service resources. This will involve reallocation, redirection and denial of services in order to maximize the availability of the service resources.

[00117] The Process Management module 334 coordinates information flow within SDP, acting as a message processing backbone. Further, the Process Management module 334 enables integration of all internal (enterprise application integration, or EAI) and external (business-to-business, or B2B) applications and services. The Process Management module 334 is further responsible for orchestrating and executing workflow. In one embodiment, it is implemented with Microsoft BizTalk Server 2004, but other implementations are possible.

[00118] The Order Management module 336 interacts with one provisioning system or trading partner at a time, accepting complex orders to be “decomposed” into more atomic “work orders” to be workable by provisioning entities. This module creates work orders based on decomposition criteria, dynamically releases work orders and validates work orders.

[00119] The Provisioning Management module 338 receives, processes, and fulfils the provision requests for authorized users. The Provisioning Management

module 338 in one embodiment consists of capabilities for service provision management and directory provision management.

[00120] The Content Management module 340 implements an offer syndication process that includes the acquisition, aggregation, transformation, staging, rating, and publishing of a variety of content from content providers and aggregators.

[00121] The Network Integration module 342 leverages the network service elements, such as those for MMS, SMS, HLS, SIP, and so on, to provide end-to-end solutions for customers. The Network Integration 342 leverages the network service elements, such as those for MMS, SMS, HLS, SIP, and so on, to provide end-to-end solutions for customers.

[00122] The Content Integration 344 operates to simplify integration tasks, connect through an adapter to retrieve content and provide the content to the value-added services which provision them to customers' devices.

[00123] The Device Integration module 346 provides device management and device adaptation for value-added services. This includes fault management, or the process of identifying and fixing device-related system or application problems and faults and configuration, or providing device configurations. The also includes accounting, or managing profiles and service usage of all devices, and performance, or identifying network congestion or system performance issues and providing fixes, and security, including ensuring access to specific services.

[00124] The OSS/BSS Integration module 348 provides a common framework for integration with OSS, including device services; BSS, including billing system; and network service elements, including LBS, SIP, SMS/MMS, and HLS.

[00125] The Service Creation Environment 304 is a development environment for software developers to create new services and to extend existing ones. The Service Creation Environment 304 simplifies the creation and reduces the time-to-market of new services. The Service Creation Environment 304 includes a Service Directory 350, an IDE and .NET Framework 352, a Service

Configurator 354, Software Development Kits (SDKs) 356, and user interface (UI) Widgets and Wizards 358.

[00126] The Service Directory 350 enables developers to discover, select, and reuse components from each service in creating new services and extending existing ones. Since its delivery of services relies on web services, standards such as UDDI and WSDL are used.

[00127] With respect to the IDE and .NET Framework 352, the SDP in one embodiment is built on Microsoft's .NET Framework using Visual Studio .NET as the integrated development environment (IDE). The Visual Studio .NET provides a Rapid Application Development environment for service creation. Developers can develop new services with a variety of .NET compliant programming languages.

[00128] The Service Configurator 354 is used to dynamically generate ASPX pages, the controls and the properties associated with these controls, thus enabling customization of web pages by each client. The SDKs 356 are resources for software developers. In addition to the IDE, developers can take advantage of product-based Software Development Kits (SDKs) in creating new services, or expanding the existing ones. Finally, the UI Widgets and Wizards 358 are a set of libraries to facilitate the creation of new service control and management pages. The power of this is insuring the modularity of user interface (UI) code, consistency with style sheets and reduced programming time for reusable widgets

[00129] The Service Deployment Environment 302 provides deployment tools and mechanisms and resource management to help system administrators to manage deployed services in a distributed network. The Service Deployment Environment 302 includes a Configuration Management module 360, a Patch Management module 362, an Image Management module 364, a Capacity Management module 366, an Inventory Management module 368 and a Resource Management module 370.

[00130] While it is important to enable developers to quickly develop new services under the SDP, it is equally important to enable operators to manage the

deployment of new services in the core networks or content deployment networks.

The SD environment shall provide the following functions:

Deployment Tools

Help to automate the deployment process in order to reduce complexity and eliminate downtime.

Support for Content Deployment Network

Content deployment network helps operators to deploy services on edge servers in the operator's content deployment network or content deployment network-like environment, in order to improve accessibility, reduce latency, and customize data based on the customer's locale. The Service Deployment environment of SDP supports deployment with third-party content deployment network systems.

Resource Management

Administrators must be able to manage information about the deployed services in terms of version numbers, operating system, system configuration and hardware configuration. The Resource Management (RM) module stores and manages resources needed for activation of services. Key strengths are:

The RM module will support at least the following kind of resources:

- Sequence-based resources (such as UIDs)
- Limited resources (such as dedicated servers)
- Round-robin resources
- Manually managed resources
- IP blocks (for managing IP addresses)

The RM module should have a web-based tool in order to manage *resources*, *thresholds*, *constrains* and *alarms*

[00131] The Configuration Management module 360 is responsible for managing the attributes and values that make up and define a service. This model provides the flexibility for SDP to implement and deploy Services in the future

that could not initially be foreseen. Using an adaptable and configurable model ensures that within SDP there is a standard understanding of what constitutes a Service so that other components are able to access and operate effectively.

[00132] The Patch Management module 362 is responsible for ensuring that the latest security updates and patches are installed for Services. SDP includes a Patch Management module 362 that manages hardware updates, application updates, operating system updates, scheduling management, inventory detection and roll back automation.

[00133] The Image Management 364 is designed to rapidly deploy new OS and application images to physical or virtualized hardware for the service provider enterprise to deploy new Services and scale out for future demand. This reduces deployment time and defects by using hardened and repeatable deployment images

[00134] The Capacity Management module 366 captures and manages the capacity levels and sources and contains a library of functions used to perform interaction with asset management and resource threshold. From the SDP side, capacity tracking is performed via tools like Operations Manager (MOM), System Management Server (SMS) and other tools. From the services side, performance thresholds that feed into capacity tracking are performed.

[00135] The Inventory Management module 368 is also known as Asset Management. This module considers the entire asset life cycle, from procurement through disposal. It enables the service provider to track and control both hardware and software. It accounts for and empowers the service provider to administer all financial and contractual aspects of the service provider's information technology (ITA) environment. This module also allows the service provider to take control of hardware and software inventory, track corporate contracts, fixed assets, and determine TCO of all of SP's assets

[00136] The Resource Management module 370 is a comprehensive set of capabilities that allow for: resource partitioning, assignment of resources by sequence, manual, round-robin, limited resource, physical resource limit checks or reserve/release resource. The module 370 operates at a logic level, working with

resource pools and quotas, and at the physical level where it is able to check and notify when resources are over allocated or nearing limit.

[00137] The Service Assurance Environment 308 provides monitoring and management functions for the SDP infrastructure and services. The Service Assurance Environment 308 includes a Business Activity Monitoring module 372, a Fault & Availability Monitoring module 374, Performance Monitoring module 376, and an Event Monitoring module 378. This module provides management functions to manage the deployment and execution of services in the operator's network, which include security, service, and service level agreement (SLA) management.

[00138] The Service Assurance environment provides management functions that span all the layers in the SDP framework. At a high level, it provides the following functions for the SDP and the value-added services:

Fault Management

This service quickly identifies problems and faults in the network or services with real-time service monitoring and alerts systems administrators with accurate problem descriptions. This service also provides automatic fail-over services in the event of unexpected system failures. Fault management is the ability to locate faults, determine the cause, and make corrections. It also includes implementing fault-tolerant hardware systems and fault-tolerant procedures. Fault management involves the following

Configuration

This service manages configuration, such as settings, policies, version controls for all the services. This service further manages the change control for all the software components.

Accounting

This service tracks detailed activities and changes for all the services and provides historical logs for auditing purposes.

Accounting is essential to measuring all aspects of the system to ensure SLA compliance.

Performance

This service monitors the performance of each service and the whole system, and identifies and reports performance-related problems to system administrators for trouble shooting and problem solving. This service further provides summarized reports on system performance for decision makers. One of the main reasons of an application or system failures are a condition of extremely degraded performance. Performance management allows high levels of application performance in a server based computing environment.

Security

This service manages the security implementation for all the services and the operator's network to detect and prevent unauthorized intrusions, denial of service attacks, and any suspicious activities.

[00139] The Business Activity Monitoring module 372 performs the real-time monitoring and access to key business and service performance indicators of up to date, live and relevant information for the purpose of improvement of the management and operations of the platform and its services.

[00140] The Fault & Availability Monitoring module 374 provides the ability to locate faults, determine the cause, and make corrections. This module also implements fault-tolerant hardware systems and fault-tolerant procedures. Fault management involves the continuous monitoring and the collection of statistics on servers, traffic conditions, and usage so potential faults can be forecast and avoided as well as setting threshold conditions, alarms, and the ability to remotely control servers and other devices to perform some or all of the preceding tasks from a single management location.

[00141] The Performance Monitoring module 376 provides improvement of application response times, management of server processor and memory resources, prevention of server lockups caused by rogue application, the ability to clamp individual applications to a fixed percentage of processor resource, the ability to identify, in real-time, any number of threads that are contributing to

excessive processor load, and then to reduce the load for this set of threads appropriately.

[00142] The Event Monitoring module 378 is an underlying operations platform that captures a wide variety of system and application events from servers that are distributed throughout a hosting environment and collects them in a central database. This module supports event consolidation, and the ability to intelligently and automatically determine which items are of critical importance to an administrator.

[00143] The various components of the Service Control / Order Management Component Architecture 300 rely on certain core technologies 380 for communication of data. The service delivery platform uses a layered, service-oriented architecture based on open standards enabling interoperability, flexibility, and adaptability. In one embodiment, the core technologies 300 include an Integrated Services Network (ISN). ISN is a run-time component that provides configurable routing based on web services description language (WSDL) documents. ISN maps any WSDL-defined service to another service on any available transport channel. ISN is usually deployed at the firewall and has access to internal services. The ISN provides the following features: Intelligent Routing; Export Routing, Import Services; Security and Management; Transformation; Universal Description, Discovery, and Integration (UDDI) Publication and Lookup.

[00144] As noted, a variety of readily available technologies may be used to implement the core technologies 380. Certain core products are used in combination in one embodiment to facilitate the integration of systems and the automation of business processes and the presentation of services in an integrated "dashboard" experience. These products include Microsoft .NET, Microsoft .NET / ACA.NET; Windows Server 2003; BizTalk Server 2004; SQL Server 2000; Active Directory; and Microsoft Provisioning Framework. Other commercially available technologies may be included as well, or substituted, and custom software may replace any of these commercially available applications.

[00145] SDP .NET is at the core of SDP as illustrated in the embodiment of FIG. 3. The application is built using Windows Server 2003, IIS 6.0 web server, Visual Studio & .NET Frameworks in C#.NET. The core of SDP is built using Microsoft .NET technologies creating a distributed enterprise application. At the core of the technology include background servicing, scheduling, real-time exception handling, audit logging, Simple Object Access Protocol (SOAP) headers, custom performance counters, caching, remoting, and a Service Configurator. The .NET core includes components written in C# .NET for business functions like, Server Management, Profile Management, Resource Management and Metering

[00146] The backbone of SDP business processing is contained in a SDP Middleware component, part of the foundation and core services. The SDP Middleware component is built using BizTalk Server 2004 to leverage the technologies capabilities for state management, Hydration/Dehydration, data transformation, EAI / Adapters / XML interface, dynamic rules processing, status management, automated and human workflow, and integrated visual studio EDI.

[00147] In the SDP Middleware component, a Web Service layer simplifies integration with Service Provider infrastructure. BizTalk coordinates actions required to complete requested business processes, such as order entry submissions from a customer relations manager or customer server representative or account management actions by a service subscriber. Custom adapters allow integration with multiple products and platforms.

[00148] SDP uses a layered, service-oriented architecture based on open standards enabling interoperability, flexibility, and adaptability. Avanade Connected Architecture (ACA).NET is a set of application blocks built entirely on the Microsoft.NET platform. It is intended to abstract infrastructure details of messaging between objects away from developers and it ensures that development will be consistent and reliable throughout the application lifecycle. Microsoft Connected Service Framework (CSF) simplifies service control, is scalable, reliable and secure, supports a broad range of devices and is based on proven

products such as BizTalk, Windows, SQL, Visual Studio.NET, and open standards such as XML, HTTP, and Web Services

[00149] The SDP core technologies 380 further includes an SDP database. The SDP database is the central store for Products/Services, Customer profiles, subscriptions, orders, resources, and provision information. The SDP database is interrelated to the core platform and the Unified Directory. Built using enhanced Telecom Operations Map (eTOM/SID) as a basis and implemented using Microsoft SQL Server 2000 on Windows Server 2003. In order to handle the SDP functional modules and capabilities of the application, a Master Data Model (MDM) supports a shared service oriented approach and operates as the data store for all master SDP data that may include services, accounts, events, exceptions, packages, resource metering, billing event data, value-add services, value-add resellers, packages & bundles.

[00150] A Unified Directory provides integrated profile and service management. The Unified Directory includes common logic that allows resources and services to be dynamically configured based on user profile, service definition, policy information, current resource allocation, and resource state. The Unified Directory is implemented in one embodiment with Microsoft Active Directory and is responsible for providing a multi-tenant organization hierarchy. All resellers, customers, end-users, service provider, and other forms of organizations and users, will all be contained in an active directory under a service oriented security architecture. What this does is allow everyone to "live" in the same directory while having access only to the little "island" that one is responsible for. This results in a customer only being able to view its own end-users and no one else's.

[00151] The architecture 300 of FIG. 3 further includes an Application / Services Gateway 382.: The Application / Services Gateway 382 provides an interface to value added services in the SDP architecture via open standards like Parlay/OSA and proprietary software API's. Opening and controlling access to services allows third party vendors, resellers, developers and external systems to interface with SDP value added services.

[00152] FIG. 4 illustrates a physical architecture of a service delivery platform (SDP) 400. The SDP 400 includes a Service Deployment Environment 402, a Service Creation Environment 404, a Service Enablement Environment 406 and a Service Assurance Environment 408. The SDP 400 is operative to control and provide access to one or more value added services, including value added service (VAS) 410. To that end, the SDP 400 has access to the Internet 412 and communicate using data protocols appropriate to the Internet, such as TCP/IP. For implementing the service, the SDP 400 includes a Service Execution Environment 416.

[00153] The VAS 410 implements any of a variety of services, including for example telecommunications services. These services may include wire line telecommunication service such as telephone and facsimile service, wireless telecommunication service such as cellular telephone and paging, and internet service, either wire line, wireless or a combination of the two. In the illustrated example, the VAS 410 includes servers and associated hardware and software to provide streaming media, electronic mail service, provisioning, web hosting and real time communications. For particular services, the VAS 410 may include alternative or additional components. The VAS 410 includes a switch 414 in data communication with a switch 418 of the Service Execution Environment 416.

[00154] The Service Deployment Environment 402 provides deployment tools and mechanisms and resource management to help system administrators to manage deployed services in a distributed network. The Service Deployment Environment 402 implements in hardware and software the functions associated with the Service Deployment Environment 302 of FIG. 3. The Service Deployment Environment 402 in the illustrated embodiment includes servers and other equipment for Code Migration, Server/Image Deployment, Capacity Planning, Inventory Management, IP Management, Patch Management, Configuration Management, Catalog Importing and Bulk Account Importing. Other devices may be provided as well. The Service Deployment Environment 402 is connected through a switch to the Service Execution Environment 416 and the Service Enablement Environment 406.

[00155] The Service Creation Environment 404 includes several components that allow for the creation of new services to be introduced to the Service Enablement Environment 406. Further, the Service Creation Environment 404 provides a development environment for software developers to create new services and to extend existing ones. The Service Creation Environment 404 simplifies the creation and reduces the time-to-market of new services. The Service Creation Environment 404 implements in hardware and software the functions associated with the Service Creation Environment 304 of FIG. 3.

[00156] To that end, the Service Creation Environment 404 includes servers and other equipment to implement Version Control, an ACA.NET framework, the SDP framework, a Service Configuration Tool, a Build Directory, a Build Database and a Build Server. These components and optionally others are accessed through a Development Station. The Service Creation Environment 404 communicates through a switch with the Service Deployment Environment 402, the Service Execution Environment 416 and the Service Enablement Environment 406.

[00157] The Service Enablement Environment 406 in the embodiment of FIG. 4 includes a Service Interaction and Management Environment 420, a Service Orchestration and Integration Environment 422, a Service Control and Management Environment 424 and Integrated Systems and Applications 426.

[00158] The Service Interaction and Management Environment 420 provides a front end or user interaction space for service users and providers. The Service Interaction and Management Environment 420 accordingly includes several servers providing user interface (UI) functionality and databases. The user interfaces include a Self Service UI portal, an Operator UI portal, a Partner UI portal and an Order Entry portal. The Service Interaction and Management Environment 420 further includes a UI Portal database and an Order Entry database accessible by the UI portals. The Service Interaction and Management Environment 420 is connected through a switch with the Service Deployment Environment 402, the Service Execution Environment 416 and the Service Orchestration and Integration Environment 422.

[00159] The Service Orchestration and Integration Environment 422 includes an Orchestration Front End Zone 428 and an Orchestration Back End Zone 430. The Service Orchestration and Integration Environment 422 orchestrates various operations required for provisioning, providing and billing for services. The Orchestration Front End Zone 428 includes an SDP Work Flow Orchestration server, a Message Queue Server, an External SDP Web Services server, an SDP Integration Server, and an Integrated Order Manager server. The Orchestration Back End Zone 430 includes a Presence Server, an SDP Work Flow Database, a Message Queue Database and an Order Database. The Service Orchestration and Integration Environment 422 communicates through switches with the Service Interaction and Management Environment 420, the Service Creation Environment 402, the Service Execution Environment 416 and the Service Control and Management Environment 424.

[00160] The Service Execution environment provides a run-time environment for controlled services such as identity management for access control, service management, subscriber management, and content management. None of these services works in isolation. In order for the Service Execution environment to execute the necessary business processes to fulfill the requests made by the users of the SDP (via Web Portal), or by the external systems (for example, via OSS and BSS), collaboration among different services in the environment is required.

[00161] The Orchestration and Integration component is responsible for the collaboration among different services by providing following capabilities:

- Work Flow Management
- Business Process Management
- Application Integration

[00162] As may be seen from above list, the Orchestration and Integration component is not only be responsible for being the information “broker” among all interacting entities, but also be responsible for “orchestrating” the execution of

business process based on the logical order of actions and the corresponding flow of messages for given process.

[00163] In SDP, the work flow should usually be initiated by the users requesting information services in real-time from the web front-end. However, on some occasions, the flow will be initiated by internal batch applications such as Notification Engine or even by external application like OSS and BSS via receiver adaptors. Regardless of the origin of the flow, the request/message will be first processed by the Orchestration and Integration module upon arrival if the request needs to be work-flowed in order to be fulfilled. Then the Work Flow Management module should make the following decisions in order to forward the request/message to the appropriate processing elements/modules:

Should the request be forwarded to different module or processed within the Orchestration Scheduler.

To where the message should be forwarded.

How incoming message be forwarded based on the system or request status.

Once the Work Flow Management module determines where to forward messages, the messages can be simply placed onto the adapter for the destination module/application.

[00164] The routing rules can be stored into a database table or into a XML document as a look-up file. This maintains the logic in the Work Flow Management being transparent from the specific business rules. This design approach promotes better maintainability and scalability for the Work Flow Management module. Another design option is to utilize BizTalk Server's Business Rules Engine.

[00165] However, not all transactions or requests that are submitted into the SDP will be processed via work flow manager. There are a number of requests that can be submitted to the SDP which do not require any kind of work flow due to the simple nature of the requests. For example, the requests to get user account and product lists, and to save profile information about a user should not invoke

any work flow to manage and process the requests. These requests, for example, can be fulfilled by calling appropriate API's via interfaces such as web services from the web portal.

[00166] One example where work flow management would be required is for the request that interacts with the systems external to the core SDP (for example, CRM or Billing systems) where the requests and responses are not usually synchronized and the whole transaction can be "long-running". Another example would be where a single request into the system spawns multiple invocations to the different modules in the system asynchronously.

[00167] The work flow manager should be utilized or invoked on an as-needed basis, not as a "Gateway" or an "Entry point" to the SDP. It should be transaction-type based. This approach will conserve the system resources for the SDP during run-time and improve overall performance of the SDP platform.

[00168] Microsoft BizTalk Server 2004 is implemented as an Orchestration and Integration module within the Service Control and Management and Service Execution environments. It enables integration of all internal (EAI) and external (B2B) applications and services to deliver and provision variety of information services by coordinating the flow of information. Again, in addition to the integration capability, it executes a pre-determined set of business processes according to the Orchestration Schedules that dictate the sequence of tasks to be performed based on the business requirements.

[00169] With respect to application integration, several modules within the SDP exchange information and collaborate in real-time to deliver information services and fulfill requests generated by external entities. These include Web Portal, Service Management, Identity Management, Profile and Personalization Management, Content Management, OSS and BSS.

[00170] The Orchestration and Integration module acts as an information broker that manages the flow of information for all of constituent applications. In order for any two modules to exchange information, the Orchestration and Integration acts as intermediary. No two modules can communicate each other without the intervention of Orchestration and Integration module. This design

option allows easier coordination of information flow and the SDP platform to be scalable for additional functional modules in the future.

[00171] The Service Control and Management Environment 424 includes a Service Control Front End Zone 432 and a Service Control Back End Zone 434. The Service Control Front End Zone 432 communicates with the Service Orchestration and Integration Environment 422. The Service Control Back End Zone 434 communicates with local storage 436. The Orchestration and Integration module acts as an information broker that manages the flow of information for all of the above modules. In order for any two modules to exchange information, the Orchestration and Integration acts as intermediary. No two modules can communicate each other without the intervention of Orchestration and Integration module. This design option allows easier coordination of information flow and the SDP platform to be scalable for additional functional modules in the future.

[00172] The following are different modes of information exchange between modules that can be implemented in the SDP:

Synchronous Transaction - in real-time mode

Asynchronous Transaction – in real-time mode

Batch Transaction

Design decisions are made in order to determine for each module in the Service Execution environment what transaction mode should be implemented to communicate with the Orchestration and Integration module.

[00173] Since the synchronous transaction waits until the response arrives, it is ideally suited for cases in which the immediate response of the transaction is required. For example, the transaction to authenticate a user via Identity Management, or the transaction to display personalized contents via Content Management are ideal candidates for this type of transaction. The XML documents are exchanged between modules as means of communication.

[00174] An asynchronous transaction is a bit more difficult to manage, but it allows multiple transactions to be executed concurrently and optimizes resource

usage. The Notification Engine within the SDP provides push-based alerts to customers. The requests may be made to the Notification Engine from the Orchestration and Integration module in the beginning of the user session, but the request may not be fulfilled instantaneously. The alerts can be useful to the user if they can be provided anytime during the session, or during allotted system time-out. Hence, this type of transaction can be implemented as asynchronous. The BizTalk Server provides a feature to “de-hydrate” the transaction to store it on the database table if it exceeds given time-outs.

[00175] A batch transaction should be implemented for those transactions that have much longer response time, in the order of hours and days. Therefore, the ideal candidates for this type of transaction are that of external organizations. The interaction of the SDP with OSS and BSS can be implemented as batch transactions. Additional complexities arise when interfacing with external organization, and therefore the following functionalities should be built around the adapters for interaction in the Orchestration and Integration module:

Validation capability to check all the fields in the incoming message based on the Interface Agreements: if the validation fails, the documents are returned to the sender without processing.

Encoding and decoding capability for encryption: this may be required for secure transmission.

XML document converter capability: the BizTalk Server can only process XML documents internally. For those organizations that use other messaging mechanisms such as EDI and MQ Series, the incoming messages should be converted into XML equivalents.

[00176] In addition to the above components to implement batch transaction, there must be a batch process that constantly run in the background to frequently check the message queues to see if any new messages to receive and to send.

[00177] The Service Control Front End Zone 432 includes an Access Control Manager server, a Directory Manager server, a Role and Policy Manager server, a Single Sign-on Manager server, a Profile Subscription Manager server, a

Service Catalog Manager server and an Internal SDP Web Services server. Further, the Service Control Front End Zone 432 includes a Provision Request Manager server, a Service Dispatcher manager, a Metering Engine, a Notification Engine, a Personalization Engine, SDP Core Architecture Components and a Resource Manager server.

[00178] The Service Control Back End Zone 434 includes the Unified Directory, a Role & Privilege Database, a Single Sign-on Database, a Profile Subscription Database, a Service Configuration Database and the Service Catalog. The Service Control Back End Zone 434 further includes a Provision Route Rule Database, and SDP Usage Patterns Database, a Metering Collection Database, a Notification Database, a Personalization Database, SDP Core Architecture Components Database and a Resources Database. In addition to the Service Orchestration and Integration Environment 422 and the local storage 436, the Service Control and Management Environment 424 communicates with the Service Assurance Environment 408.

[00179] The Service Assurance Environment 408 provides monitoring and management functions for the SDP infrastructure and services. The Service Assurance Environment 408 includes hardware and software for implementing the functions associated with the Service Assurance Environment 308 of FIG. 3.

[00180] The Service Assurance Environment 408 includes an electronic support portal server, a web console server, a management console server, a root cause analysis server, a Data Access Consolidator Agent Manager server, a Reporting Server, a Database, a Domain Controller and a domain name server. The Service Assurance Environment 408 thus includes facilities for monitoring fault conditions and performance of the SDP system 400.

[00181] FIGS. 5-12 illustrate use cases showing operation of the SDP system described above. These use cases illustrate interaction among the interested parties, including a customer, a Service Provider, and the SDP operator. Other active individuals include a Customer Service Representative, a Help Desk Operator and a Value Added Reseller. Upon initiation by these parties, several components of the SDP are active to implement the use case interaction, including

a Web Service Interface, a Customer Profile Management engine, a Process Manager, a Customer Profile Management engine, User Profile Management engine, an integrated order manager, a Provisioning system and a Metering engine. Other active components include a Service Catalog Management engine, a Service Creator, a Work Flow Management engine, a Credential Management engine, a Directory Management engine.

[00182] In the last above description, the integration capability of the Service Orchestration and Integration Environment has been described. The orchestration capability of the module allows tasks within a business process to be sequenced and executed in real-time.

[00183] One ultimate goal of the SDP is to be able to provision and deliver any type of information and telecom services for the users. This implies that the orders or requests that users send from web front-end to the back-end may consist of different combinations of services, products, and features. This also implies that the SDP needs to interact with a variety of trading partners to fulfill customers' requests by provisioning different features on the orders. For example, a local phone service order with voice mail feature maybe provisioned at two different trading partners.

[00184] The fulfillment of the "complex" or "bundled" provision orders/requests demands the need of defining a business process to perform order management in the SDP. This subsection will describe the necessary tasks that will be required to implement the Order Management Process in the SDP. The following two areas of functionalities are built within the Service Orchestration and Integration Environment:

Order Processing and Work Order Generation

Order Lifecycle Management

The following describes these two capabilities.

[00185] The end users experiences are not limited to a single, unidirectional web front-end page flow. The users should be able to add, remove, and update any services, products, and features they can order during a single user session by visiting different pages in the SDP's Web Portal in any way that the user desires.

[00186] The added flexibility to the user experience on the web front-end implies that the orders/requests that are submitted by end users are bundled with different combinations of services, products, and features. This type of order can be referred to as a “complex” order because it may take more than one provisioning system or trading partner to fulfill the order.

[00187] Since the Orchestration and Integration module can interact with one provisioning system or trading partner at a time, the complex orders have to be first “decomposed” into more atomic “work orders” to be workable by provisioning entities. The following functionalities should be built in the Orchestration and Integration module to create and process work orders by decomposing the complex orders:

[00188] Creation of work orders based on decomposition criteria:

The creation of work orders can be initiated by both end-user’s complex orders and the responses received from the provisioning system

The complex orders can be split into work orders according to set of predetermined business rules

[00189] Dynamic release of work orders

Relationships are established and maintained across work orders to determine the dependencies for work order release and required processing steps

[00190] Validating work orders

After the all the necessary work orders are created, the validity checks should be performed on all the work orders before submission.

[00191] The complex orders are decomposed into work orders in any way that a business rule dictates. They should be decomposed by different trading partners, services, or even by different features on the complex orders. One important design principle to note for this capability is that all the business rules that provide criteria for decomposition reside in database tables or in XML files.

This allows the Order Management module to be transparent from the specific business rules, and encapsulate pure order processing capability that can be integrated with other applications.

[00192] On some occasions, all of the work orders that are generated from a complex order cannot be released into the provisioning systems at the same time. The release of work orders sometimes has to be sequenced and scheduled based on business rules. For example, if a complex order consists of two different services, Local Phone Service and DSL Service, two work orders can be generated: one for Local Phone Service and other for DSL service. However, the Local Phone Service work order has to be released and provisioned first before the DSL work order is released. Again, the predetermined work order release schedules are maintained in database tables or in XML files. This enables the capability to be reusable and maintainable.

[00193] Once all the work orders are submitted to the provisioning systems, the Order Lifecycle Management capability starts to track the work orders until completion if successfully provisioned. Depending on the kind of service that the user needs to provision, the time to complete provisioning may vary. For example, the provisioning MS SharePoint or Exchange Services via MPS may complete instantaneously whereas the provisioning of telecom or data services may take hours or days.

[00194] All the requests/orders that should be provisioned by the SDP are tracked by the Order Management capability built within Orchestration and Integration module, and it notifies end-users whenever the order status changes. One of the benefits of implementing work order tracking capability is that especially for those work orders that require relatively lengthy provisioning time, if the work order "fallout" or exception occurs in the middle of the provisioning lifecycle, the order tracking capability should be able to determine where in the lifecycle the work order failed, and notify users immediately. If the exception is not fatal and can be resolved without the user's intervention, the Order Management capability dynamically generates and re-submits additional work orders so that the provisioning lifecycle can proceed.

[00195] In order to perform Order Lifecycle Management, the following capabilities are implemented in the Orchestration and Integration module:

- Transitioning of the order status in the provisioning lifecycle
- Handling of exceptions and fallouts during provision
- Dynamic generation of work orders based on incoming responses from external systems

[00196] After the work order is submitted to the provisioning system, the provision status of the work order can be tracked. This can be accomplished by updating the order status maintained in the database table based on the confirmation response the SDP receives whenever each phase of provision is completed from the provisioning system. Whenever there is a change in the order status, the user is notified of the change. Depending on the new status to which the work order moves, the components in the SDP may have to be notified as well. For example, if the order moves to the 'Complete' status, the message is sent to the Billing System to generate the bills for the customer, and the message is also sent to the Subscriber Management to update the customer's profile to grant access to the fully provisioned service.

[00197] However, during the usual provision lifecycle, the orders sometime "fallout" or fail to move on to the next phase. Then the provisioning system sends the message to the SDP indicating the exception had occurred. This triggers the order status to move to 'Exception' or 'Fail' status. The Orchestration and Integration module first assesses the response from the provision system, and if it is determined that it can be resolved by, for example, sending additional information from the database to the provisioning system, it dynamically generates and re-submit a work order. However, if it cannot be resolved in the module, the user is notified, and the errors in the work order should be manually fixed.

[00198] There are also "sunny days" work order provision scenarios that call for dynamic generation of work orders. During the provision lifecycle, if the provisioning system determines that it needs additional information from the SDP

to complete the provision, it may request the information. Then the SDP will submit new work order that contains the requested information.

[00199]

[00200] FIG. 5 is an interaction diagram illustrating a process of changing features of an existing service using a service delivery platform. FIG. 5 illustrates an operation in which a service variant manager chooses to modify features in an existing service variant. All users subscribed to the service will have their subscription updated.

[00201] In the use case illustrated in FIG. 5, the primary actor is a customer service representative of the service provider. A customer or service subscriber desires to have an existing service adapted to fit specific needs. The SDP operator desires to keep and maintain a record of the changed features. The Service Provider wants accurate information to provision feature changes for a user, and a Help Desk Operator wants to have access to records which tracks a user's request and the status of the request.

[00202] At step 502, the Customer Service Representative receives a request from the customer to change a feature on an existing service and The Customer Service Representative (CSR) requests the solution information for the customer, step 504. The solution information is stored as profile data in an SDP database. The Web Service Interface receives the request and initiates a request for the customer solution information to the Customer Profile Management engine, step 506. The Customer Profile Management engine returns the solutions for the customer to the Customer Service Representative, step 508.

[00203] The Customer Service Representative then selects the solution to update and requests to edit the details of a Solution Service Variant contained in the Solution. The CSR modifies an editable feature(s). The CSR selects how to apply these modifications to the underlying subscriptions. The CSR submits the changes, step 510. The Web Service Interface receives the request and initiates a request for modified feature(s) to the Process Manager, step 512. The Process Manager identifies the appropriate Orchestration to call. The Process Manager initiates a request to update the Solution Service Variant to the Customer Profile

Management engine, step 514. The Profile Management engine returns status to the Process Manager, step 616.

[00204] The Process Manager initiates a request to retrieve the Subscriptions based on the Solution Service Variant to the User Profile Management engine, step 518. The User Profile Management engine returns the affected subscriptions to the Process Manager. The Process Manager updates the Subscription Service Variants with the modified values based on the override settings and initiates a request to update the Subscription Service Variant to the User Profile Mgmt engine. The User Profile Management engine returns status to the Process Manager. The Process Manager creates integrated order manager (IOM) Requests for the Subscription Service Variants affected. Based on the override settings, the IOM Request to Update Subscription is either submitted to IOM, step 520, or saved to a Database for User Activation. The Integrated Order Management engine will determine if the work order requires bulk or on-demand scheduling and initiate a request to the Provisioning system to propagate the changes to the subscribed service(s), step 522. The Provisioning system will provision the feature change for the users and return status to the Integrated Order Management engine, step 524. The Integrated Order Management engine will return status to the Process Manager, step 526. The Process Manager identifies the next step in the Orchestration schedule; update the status on the subscription. The Process Manager identifies the next step in the Orchestration schedule, Create a Billing Event. The Process Manager initiates a request to the Metering engine to create a billing event. The Metering engine creates a billing event for the subscription to the Process Manager. The Process Manager identifies that there are no more steps in the Orchestration schedule.

[00205] FIG. 6 is an interaction diagram illustrating a process of creating a new offer using a service delivery platform. In this use case, the primary actor is the service variant creator. In implementing this process, an Offer Manager wants the ability to create new offers containing the new service variant. The SDP Operator wants a record of the service creation and wants the provisioning engine to be able to provision the new service variant. The SDP Operator wants the

ability to monitor the new service variant. The Service Variant Creator wants accurate, fast entry, with no errors to create the new service variant on the SDP and wants to make the service variant available to an Offer Manager.

[00206] In FIG. 6, the Service Variant Creator requests to view the available product definitions, step 602. The Web Service Interface receives the request and initiates a request for the product definitions to the Service Catalog Management engine, step 604. The Service Catalog Management engine returns the list of available product definitions to the Web Service Interface, step 606. The Web Service Interface returns the list of available product definitions to the Service Variant Creator, step 608.

[00207] The Service Variant Creator then selects a product definition and requests to create a new service, step 610. The Web Service Interface receives the request and initiates a request for the service definitions associated with the product definition to the Service Catalog Management engine, step 612. The Service Catalog Management engine returns the list of available service definitions to the Web Service Interface, step 614. The Service Creator enters the information necessary to create a service based on a selected service definition, step 616. The Web Service Interface receives the request and initiates a request to create the new service variant based on the selected service definition to the Service Catalog Management engine, step 618. The Service Catalog Management engine returns the new service variant creation status to the Web Service Interface, step 620.

[00208] FIG. 7 is an interaction diagram illustrating a process of creating a billing feed using a service delivery platform. In the process of FIG. 7, the primary actor is an external billing system. Among the stakeholders, an End User and customer both want the bill for services to be accurate, the SDP Operator wants to create an accurate billing feed for the billing system and wants a record of the billing feed creation. Finally, the Service Provider wants the billing events to be accurately captured in the SDP.

[00209] Prior to the process, the external billing system has requested billing records and the metering service receives and stores billing events from services.

The External Billing System requests a billing feed, step 702. At step 704, the Work Flow Management engine interprets the request and routes a request to create a billing feed to the Metering Service. The Metering Service aggregates the billing events to create billable data records. The Metering Service requests subscription information from the Profile Management engine to append to the billable data records, step 706. The Metering Service requests account information from the Directory Management engine to append to the billable data records, step 708. The Metering Service creates a billable feed file comprised of billable data records. The Metering Service returns the billable feed file to the Work Flow Management engine, step 710.

[00210] FIG. 8 is an interaction diagram illustrating a process of providing a value-added reseller with a customer view using a service delivery platform. In this use case, the primary actor is a Value Added Reseller (VAR). Among the stakeholders in this process, the Customer wants the Customer's information to be accurate, the VAR wants an accurate summary of Customers, and the Service Provider wants accurate and easily accessible information regarding their VARs.

[00211] In FIG. 8 at step 802, the Value Added Reseller requests to view all of their customers. The Web Service Interface receives the request and initiates a request to the Customer Profile Management engine to retrieve the customer solutions, step 804. The Customer Profile Management engine identifies all the customers for the given reseller. The Customer Profile Management engine retrieves the customer solutions containing the Service Variants and Offers provided by the VAR. The Customer Profile Management engine returns the customer solutions containing the Service Variants and Offers provided by the VAR to the Web Service Interface, step 806. The Web Service Interface returns the customer solution back to the value added reseller, step 808. The Value Added Reseller requests to view the detailed customer information for each customer containing the Service Variants provided by the VAR, step 810. The Web Service Interface receives the request and initiates a request to the Directory Mgmt engine to retrieve the customer information, step 812. The Directory Management engine returns the customer information for the given VAR from Unified Directory to the

Web Service Interface, step 814. The Web Service Interface returns the customer information back to the value added reseller, step 816.

[00212] FIG. 9 is an interaction diagram illustrating a process of creating a new account in an enterprise environment using a service delivery platform. FIG. 9 includes FIG. 9a and FIG. 9b. In this process, the primary actors are a Customer Administrator and the End User (of a Customer). Other stakeholders affected by the process include an End User, who wants accurate, fast entry, and no errors when ordering products, a Customer who wants a record of the order, the SDP Operator who wants a record of the order and the status of the order, and the Service Provider who wants to accurately provision the services for the new user.

[00213] The process begins at step 902 when the Customer Administrator (Admin) requests to create a new Solution, viewing the Offers/Service Variants within the Service Catalog. The Web Service Interface interprets the request and routes a request for the Offers/Service Variants available to the Customer to the Service Catalog Management engine, step 904. The Service Catalog Management engine returns the Offers/Service Variants that the Customer has a right to view to the Web Service Interface, step 906. The Web Service Interface returns the Offers/Service Variants to the End User, step 908, allowing him to select the items to add to their solution. The Customer selects 1 Offer containing 2 new services (1 asynchronous, 1 synchronous) to order, and submits the request to create a solution containing the Offer, step 910.

[00214] The Web Service Interface routes the request to the Customer Profile Mgmt engine to create the Solution, step 912. At step 914, the Customer Profile Mgmt engine returns the Solution ID to the Web Service Interface. The Customer selects to create a new Service Group, step 916, enters information for the New Service Group and selects to submit. The Web Service Interface routes the request to the Credential Mgmt engine to create the Service Group, step 918. The Credential Mgmt engine creates the new Service Group in the Unified Directory.

[00215] The Credential Management engine returns the Service Group ID to the Web Service Interface, step 920. The Customer selects to edit their Service

Groups, step 922. The Web Service Interface routes the request to the Credential Management engine, step 924, to retrieve the Service Groups belonging to the Customer. The Web Service Interface routes the request to the Customer Profile Management engine, step 926, to retrieve the Solutions belonging to the Customer.

[00216] The customer selects to associate the 2 services in the above offer with the new Service Group and submits it, step 928. The Web Service Interface routes the request to the Customer Profile Management engine to create the relationship between the Service Group and the Services, step 930. The Customer Profile Management engine returns status to the Web Service Interface, step 932.

[00217] At step 934, the Customer requests to create a new End User. The Web Service Interface submits a request for the Available Roles to the Credential Management engine, step 936. The Credential Management Engine returns a list of available Roles to the Web Service Interface, step 938. The Web Service Interface submits a request for the Available Service Groups to the Credential Management Engine, step 940. The Credential Management Engine returns a list of available Service Groups to the Web Service Interface, step 942.

[00218] At step 944, the Customer enters the End User Account Information, including selecting a Role and Service Group (using Service Group created in the earlier steps). The Customer submits a request to create the new account. The Web Service Interface submits a request to the Directory Management engine to create an Account, step 946. The Directory Management engine sends a request to Credential Management to assign the new User to a Role group. The Directory Management engine sends a request to Credential Management to assign the new User to a Service Group. The Directory Management engine returns the User ID to the Web Service Interface, step 948.

[00219] Next, the Customer Logs Off and the new End User Logs On. The End User selects to subscribe to services, step 950. The Web Service Interface submits a request to the Customer Profile Management engine to retrieve the available Solution Service Variants based on Service Group and Customer, step 952. The Customer Profile Management engine returns an array of Solution Service Variants to the Web Service Interface at step 954.

[00220] At step 956, the End User selects the two Services to which to subscribe. The Web Service Interface routes the request to the Process Mgmt engine to create Subscriptions, steps 958. The Process Management engine routes a request to the User Profile Management engine to create a Subscription in the SDP Database for Service 1, step 960. The User Profile Management engine creates the subscription and returns status to the Process Management engine, step 962. The Process Management engine then routes a request to the User Profile Management engine to create a Subscription in the SDP Database for Service 2, step 964. The User Profile Management engine creates the subscription and returns status to the Process Management engine at step 966.

[00221] The Process Management engine creates an IOM Request and submits it to the Integrated Order Management engine at step 968. The Integrated Order Management engine validates the services using the Service Catalog Management engine. The Service Catalog Management engine returns validation status to the Integrated Order Management engine. The Integrated Order Management engine generates work orders based on the order request. The Integrated Order Management engine releases the work order to provision the new service 1 to the Provision Request Management engine, step 970. The Integrated Order Management engine releases the work order to provision the new service 2 to the Provision Request Management engine, step 972.

[00222] The Provision Request Management engine uses the Provisioning engine to provision new services. The Provision Request Management engine notifies the Integrated Order Management engine that the service 1 has been successfully provisioned, step 974. The Provision Request Management engine notifies the Integrated Order Management engine that the service 2 has been successfully provisioned, step 976. The Integrated Order Management engine returns order status to the Process Mgmt engine, step 976. The Process Management engine updates the status of the subscriptions, including the configuration information, step 978.

[00223] FIG. 10 is an alternative interaction diagram illustrating a process of creating an account with two new services using a service delivery platform. This

process is a consumer version of the process of FIG. 9. In the process of FIG. 10, the primary actor is an End User, not from a customer. The relevant stakeholders in this process include the End User, who wants accurate, fast entry and no errors when ordering products, the SDP Operator who wants a record of the order and the status of the order, and the Service Provider who wants to accurately provision the services for the new user. Prior to execution of the process, there is anonymous access to a self-service portal of SDP.

[00224] In FIG.10, the process begins at step 1002 when the End User selects to subscribe to services and requests to view the Offers/Service Variants within the Consumer Solution. The Web Service Interface interprets the request and routes a request for the Offers/Service Variants available to the End User to the Customer Profile Management engine, step 1004. The Customer Profile Management engine returns an array of Solution Service Variants to the Web Service Interface, step 1006.

[00225] The End User selects the two services to which to subscribe, step 1008, enters account information and submits the order as a new User. The Web Service Interface routes the request to the Process Management engine to create Subscriptions for a new user, step 1010. At step 1012, the Process Management engine routes a request to the Directory Management engine to create an Account for the User. The Directory Management engine creates the Account under the Consumer OU and returns the User ID, step 1014. The Process Management engine routes a request to the User Profile Management engine to create a Subscription in the SDP Database for Service 1, step 1016. The User Profile Management engine creates the subscription and returns the Subscription to the Process Management engine, step 1018. The Process Management engine routes a request to the User Profile Management engine to create a Subscription in the SDP Database for Service 2, 1020. The User Profile Management engine creates the subscription and returns the Subscription to the Process Management engine, step 1022. The Process Management engine then creates a Create Subscriptions Order Request and routes the order request to the Integrated Order Management engine,

step 1024. The Integrated Order Management engine validates the services using the Service Catalog Management engine.

[00226] The Service Catalog Management engine returns validation status to the Integrated Order Management engine. The Integrated Order Management engine generates work orders based on the order request. The Integrated Order Management engine releases the work order to provision the new service 1 to the Provision Request Management engine, step 1026. The Integrated Order Management engine releases the work order to provision the new service 2 to the Provision Request Management engine, step 1028. The Provision Request Management engine uses the Provisioning engine to provision new services. The Provision Request Management engine notifies the Integrated Order Management engine that the service 1 has been successfully provisioned, step 1030. The Provision Request Management engine notifies the Integrated Order Management engine that the service 2 has been successfully provisioned, step 1032. The Integrated Order Management engine returns order status to the Process Management engine, step 1034. The Process Management engine updates the status of the subscriptions, including the configuration information, step 1036.

[00227] FIG. 11 is an interaction diagram illustrating a process of authenticating and authorizing an end user of services using a service delivery platform. In this process, the primary actor is the End User of a Customer. The End User wants access to purchase products, view their products, and manage their products/profile. Other stakeholders in the process include the Customer who wants access to purchase offers, to view their offers, and manage their offers/profile/end users, the Offer Manager wants access to manage offers, the SDP Operator, who wants a record of the users on the system and to provide a secure environment, and also wants to lock down privileges to reduce errors created by users on the system. Further, the SDP Operator wants access to monitor and maintain the SDP, the Service Provider wants access to manage the services and a Help Desk Operator: Wants access to system and user records to assist in resolving issues.

[00228] Referring to FIG. 11, at step 1102, the End User enters their credentials into the login screen and submits them. The credentials are encrypted. The Web Service Interface receives the request and initiates a request to authenticate the End User to the Access Control Management engine, step 1104. The Access Control Management engine decrypts the credentials. The Access Control Management engine validates the credentials against Unified Directory and returns account and membership information, step 1106. The Access Control Management engine requests validation of the subscription status for the End User from the Profile Management engine, step 1108. The Profile Management engine returns the subscription status for the End User to the Access Control Management engine, step 1110. The Access Control Management engine returns the account, membership, and subscription status information to the Web Service Interface at step 1112.

[00229] The Web Service Interface engine requests role information based on the membership information of the End User from the Role Management engine at step 1114. The Role Management engine returns the privileges for the End User to the Web Service Interface at step 1116. Finally, at step 1118, the Web Service Interface returns the account, membership, privileges information, and subscription status to the End User where it will be stored to the session.

[00230] FIG. 12 is an interaction diagram illustrating a process of handling a provisioning exception when provisioning services using a service delivery platform. In this process, the primary actor is the Provisioning Engine. Among the stakeholders in the process, the End User wants the system to handle the exception and be notified with a friendly message if the exception could not be resolved by the system; the Customer wants the system to handle the exception and be notified with a friendly message if the exception could not be resolved by the system; and the SDP Operator wants to be able to view the exceptions and the technical detail associated with the exception. Further, the Service Provider wants to be able to view the exceptions and the technical detail associated with the exception; and the Help Desk Operator wants to be able to view the exceptions and the technical detail associated with the exception.

[00231] Referring to FIG. 12, initially a provision request failure is logged as exception. The initial provision request is corrected and is reinstated for processing. At step 1202, the Provisioning engine responds with provisioning exception information to the Provision Request Management engine, step 1204. The Provision Request Management engine responds to the Integrated Order Management engine stating that the service could not be provisioned, step 1206.

[00232] The Integrated Order Management engine responds to the Work Flow Management engine stating that the service could not be provisioned, step 1208. The Work Flow Management engine requests to log the provision exception to the Exception Engine, 1210. The Exception Engine fails to correct the exception. The Exception Engine responds to the Work Flow Management engine with the provision exception details, 1212. The Work Flow Management engine sends the provision exception information to the Notification Engine at step 1214.

[00233] At step 1216, the Notification Engine sends the provision exception event to the SDP Operator. The SDP Operator views the provision exception event and resubmits the request to provision the service, steps 1218, 1220. The Work Flow Management engine routes the provision request to the Integrated Order Management engine at step 1222. The Integrated Order Management engine reconciles the provision request with the original work order for service provisioning and releases the work order to the Provision Request Management engine, step 1224. The Provision Request Management engine uses the Provisioning engine to provision the service at step 1226.

[00234] FIG. 13 is an interaction diagram illustrating a process of changing offer properties of an offer of services by a service provider, using a service delivery platform. The primary actor in this process is the Offer Manager, who wants to edit an existing offer property. Another interested stakeholder is the SDP Operator, who wants a record of the property change(s) for the offer.

[00235] In FIG. 13, at step 1302, the Offer Manager requests to view their existing offers. The Web Service Interface receives the request, step 1304, and initiates a request for the Offer Manager's offers to the Service Catalog

Management engine, step 1306. The Service Catalog Management engine returns at step 1308 the filtered offers that the Offer Manager has the right to see to the Web Service Interface, step 1308.

[00236] The Web Service Interface returns the filtered offers to the Offer Manager, step 1310, allowing him to select offers to edit. The Offer Manager requests to edit an offer in the Service Catalog, 1312, 1314. The Web Service Interface receives the request and initiates a request at block 1316 to the Service Catalog Management. The Service Catalog Management engine returns the offer information and related service details for the selected offer to the Web Service Interface at step 1318.

[00237] The Web Service Interface returns the offer information and related service details to the Offer Manager, step 1320, allowing him to change the properties of the offer. The Offer Manager edits the desired offer properties and requests to save the changes. The Web Service Interface interprets the request and routes a request to update the offer to the Service Catalog Management engine, step 1320, 1322. The Service Catalog Management engine updates the offer information to reflect the property changes requested by the offer manager and returns status to the Web Service Interface at step 1324.

[00238] In the illustrated embodiments, all service activation, deactivation and modification processes are based on a workflow. A workflow is designed by using an operator web-based tool. A workflow is composed of:

[00239] A set of Status - Each status will represent a phase in the life-cycle of a particular service;

[00240] A set of Events - An event is something that changes the status of a service. Workflow state transitions are triggered by events. In the context of this documentation, there are two major types of events; Business Event and System Event. Most of the events will fall under these two main categories.

[00241] A Business Event is a system-independent message that represents a detailed business action through the Enterprise Application Integration (EAI) application. EAI is the process of coordinating the operation of various

applications across an enterprise or service provider so that the applications can perform as an integrated enterprise-wide system.

[00242] A System Event is a system-dependent message that is sent by or sent to an application external to the EAI application. System Events correspond to the public interface exposed by a particular system, such as an XML document, an API call, or a database table. System Events are converted to or from Business Events and can have a one-to-one, one-to-many, or many-to-many relationship with Business Events.

[00243] A workflow is further composed of a set of Actions - An action is an operation that has to be performed when a state transition happens.

[00244] The services workflow rules are stored in a State-Transition Table (STT) kept on an SDP database. Each service type has its own STT.

[00245] In the illustrated embodiments, a common data mapping approach is implemented to create a solution that enables service providers to have a choice in selecting their billing, customer care, provisioning, and other applications and to enable business event definitions to be consistent across various application combinations. For each of the business events in scope a data-mapping matrix is created. Using this, commonalities are extracted to create event definitions. This common data model is used as an input to determine the data type conversion rules for the data transformations between applications.

[00246] These data mappings create the start of a master data model (MDM), which can be extended as additional applications are added to the solution. As new applications are added, the application data model will only need to be mapped to the master model rather than mapping to each of the other applications independently. Thus, as the SP solution grows, there is only the effort to maintain each application's specific format with its mappings to the SP (common) format for each business event.

[00247] In order to handle the SDP functional modules described above and the capabilities of the application, a master data model (MDM) is supported. The MDM supports a shared service oriented approach and operates as the data store

for all master SDP data that may include services, accounts, events, exceptions, packages, resource metering, and billing event data.

[00248] The MDM is designed to support a common universal set of information. The SDP solution supports a wide array of value-added services, value-added resellers, packages and bundles. The SDP MDM supports differing value-added services by encapsulating a wide set of universal attributes. This allows the SDP to support current and future products and services that service providers will offer. The MDM also provides this level of flexibility for other SDP capabilities such as product packaging and account hierarchy.

[00249] In order to handle the business event transformations within the application connection models in a consistent manner, a common transformation framework will be utilized across each of the connection interfaces within the SDP. BizTalk performs transformations using XSLT. This requires that the incoming and outgoing messages must be in XML format. This means that some applications that cannot provide messages in XML format will utilize BizTalk's ability to convert other file formats to XML.

[00250] Presence Management is utilized to dynamically detect a user's 'presence' on a range of devices and networks. These can include fixed and mobile telephony, paging, instant messaging, voicemail, and email. A combination of network and device information is used to build an intelligent picture of a user's situation and accessibility. In order to manage user experience, the SDP recognizes a user's presence and customizes their experience based on their profile.

[00251] The components of presence management are as follows:

[00252] Presence

User Access Control -- Receive and manage events related to user network access and perform user authentication and authorization.

User Access Policy Management -- Provide policies to authenticate users from different types of networks (Wired, Wi-Fi, Wireless, Mobile)

- User Access Policy Configuration -- Update user presence information on Unified Directory
- [00253]** Unified Directory
- Subscriber Information Management - Provide information regarding user status (authentication, localization, type of access)
- User Access Profile -- Provide services subscribed by the user and the service technical profile (ex. Bandwidth)
- User Device Information -- Provide details of user's device to access the services
- Third Party Management -- Provide a single point of access to retrieve or store information to Unified Directory
- [00254]** Presence information is made available within the SDP in one embodiment by utilizing Microsoft Live Communication Server (LCS) 2003. LCS will be configured to trigger server processes without the users' intervention.
- [00255]** FIG. 14 illustrates an object model for use in conjunction with a service delivery platform (SDP) system and method. FIG. 14 includes FIGS. 14a, 14b, 14c and 14d. FIG. 14 forms an entity-relationship diagram of an example of a data model 1400 that stores and links information that relates to delivering a variety of services by one or more service providers to a plurality of service subscribers.
- [00256]** Referring to the portion of FIG. 14 shown in FIG. 14a first, the data model 1400 generally includes an SDP_Offer entity 1402 that stores data relating to an offer. An offer is set of Service Variants for presentation to customers for purchase. Service offers consist of one or many Service Variants. An offer can also include other offers, referred to as sub-offers. In the example of FIG. 14, the SDP_Offer entity 1402 is designed to store offers according to offer name and offer description. The SDP_Offer entity 1402 includes an Offer ID field which serves as the primary key (PK) for the entity, thus uniquely defining each SDP offer data object. In addition, the SDP_Offer entity 1402 may include the following fields as foreign keys: Offer_Name, Offer_Description.

[00257] Additional fields may also be included in the SDP_Offer entity 1402. For example, as shown in FIG. 14, the SDP_Offer entity 1402 includes a Base_Offer_ID field for storing an identifier for a base offer; a Created_By_ID for storing the identity of the person or entity who created the offer; a Created_Date_ID, for storing the date of creation of the offer; and Updated_By_ID for storing the identification of the person or entity which last updated the offer; and an Updated_Date, for storing the date of last update of the offer. Generally all of the data storage entities of the data model 1400 include these last four fields.

[00258] The data model 1400 further includes an SDP_Offer_Relation entity 1404. The SDP_Offer_Relation entity 1404 defines relationships between an offer and a sub-offer. The SDP_Offer_Relation entity 1404 includes the Offer_ID and Sub_Offer_ID fields as foreign keys which are linked to the SDP-offer entity 1402.

[00259] The data model 1400 further includes an SDP_Offer_Regrade_Options entity 1406. The SDP_Offer_Regrade_Options entity 1406 stores information about options for regarding an offer. The SDP_Offer_Regrade_Options entity 1406 includes the Offer_ID and Regrade_Offer_ID fields as foreign keys which are linked to the SDP-offer entity 1402.

[00260] The data model 1400 further includes a LNK_Offer_Service_Variant 1408. The LNK_Offer_Service_Variant 1408 stores information linking the SDP_Offer entity 1402 to a SDP_Service_Variant entity 1410, discussed below. The LNK_Offer_Service_Variant 1408 includes the Offer_ID as a foreign key which is linked to the SDP_Offer entity 1402. The LNK_Offer_Service_Variant 1408 includes the Service_Variant_ID as a foreign key which is linked to the SDP_Service_Variant entity 1410.

[00261] The data model 1400 further includes an SDP_Event_Message entity 1412. The SDP_Event_Message entity 1412 stores data defining a message about an event occurring in the SDP system. The SDP_Event_Message entity 1412 includes an Event_Message_ID as a primary key. The entity 1412 also

includes several fields defining the event message, including an Event_Category_ID, an Event_Category_Name, and Event_Message_Type, and Event_Message_Name and an Event_Message_Description.

[00262] The data model 1400 further includes an SDP_Subscription_Type entity 1414. The SDP_Subscription_Type entity 1414 stores data about a subscription to a service provided by the SDP. The SDP_Subscription_Type entity 1414 includes a Subscription_Type_ID as a primary key and also includes a Subscription_Type_Description field and a Type_Value field defining the subscription type.

[00263] The data model 1400 also includes a UD_Service_Group entity 1416 defining a service group within the unified directory. The UD_Service_Group entity 1416 includes a Service_Group_ID as a primary key and also stores a Service_Group_Description field.

[00264] The data model 1400 also includes an SDP_Offer_Rule entity 1418. The SDP_Offer_Rule entity 1418 stores information defining rules about services which must be prescribed as a prerequisite to a particular service. The SDP_Offer_Rule entity 1418 includes an Offer_Rule_ID as a primary key and also stores a Service_ID field defining the service and a Prerequisite_Service_ID field defining the service which must be subscribed as a prerequisite.

[00265] The data model 1400 further includes an SDP_Pricing entity 1420. The SDP_Pricing entity 1420 stores information about service pricing and includes a Pricing_ID as a primary key. Further, the SDP_Pricing entity 1420 stores several field defining pricing of the service, including a Pricing_Value field, a Currency_ID field, a Payment_Model_ID and a Payment_Term_ID.

[00266] The offer model 1400 further includes an SDP_Role_Privilege entity 1422. The SDP_Role_Privilege entity 1422 includes a Privilege_ID that is a primary key and also functions as a foreign key and is linked to an SDP_Privilege entity 1424. The SDP_Role_Privilege entity 1422 further includes a Role_ID as a primary key and a field labeled Privilege.

[00267] The SDP_Privilege entity 1424 linked to the SDP_Role_Privilege entity 1422 stores information about privileges in the SDP system. The

SDP_Privilege entity 1424 includes a Privilege_ID field as a primary key and further includes a description field for storing information describing the privilege.

[00268] The data model 1400 further includes an SDP_Detail entity 1426. The SDP_Detail entity 1426 includes a Detail_ID or detail identifier as a primary key. The SDP_Detail entity 1426 also stores information in Detail_Value, Detail_Description and Reference_ID fields.

[00269] The data model 1400 further includes a REF_Override_Settings entity 1428 defining override settings for use in the SDP. The entity 1428 includes fields such as Override_Setting_ID, Override_Setting_Value, Soluton_Override_Flag, Catalog_Override_Flag and Default_Flag.

[00270] For records that require geographical information, the data model 1400 includes several geographical entities. These include a REF_City entity 1430, a REF_State entity 1432 and a REF_Country entity 1434. The REF_City entity 1434 defines the relevant city and includes a City_ID field as a primary key. The entity 1430 further includes a City_Name field and a Country_ID field as a foreign key linked to the REF_Country entity 1434 and a State_ID as a foreign key linked to the Ref_State entity 1432.

[00271] The REF_State entity 1432 includes a State_ID field as a primary key and also stores a State_Name field defining state-related geographic information. Further, the REF_State entity 1432 includes a Country_ID field as a foreign key linked to the REF_Country entity 1434

[00272] The REF_Country entity 1434 stores information defining the country relevant to the described service or offer. The REF_Country entity 1434 includes a Country_ID field as a primary key and also a Country_Name field for storing information about the country associated with the service.

[00273] The data model 1400 further includes an SDP_Term entity 1436. The SDP_Term entity 1436 includes an identification field Term_ID as a primary key and also stores a Term_Description field with description information and a Pricing_ID field.

[00274] The data model 1400 further includes a Communication_Method_ID entity 1438. The Communication_Method_ID entity

1438 stores information about a communication method. The entity 1438 includes a `Communication_Method_ID` field with identification information, which is used as a primary key. Further, the `Communication_Method_ID` entity 1438 includes fields defining a `Method_Description` and `Bill_Feed_ID`.

[00275] Referring to the portion of FIG. 14 shown in FIG. 14b, the data model 1400 includes an `SDP_Product` entity 1440 that stores data related to a product offered through the SDP. The `SDP_Product` entity 1440 includes a `Product_ID` field which serves as the primary key for the `SDP_Product` entity 1440. Additional fields are also included in the example of FIG. 14, including a `Product_Name` for storing a name or other identifier for the product, a `Vendor` field for storing identification of the vendor of the service, a `Version` for storing version identification information, a `Service_ID` field which identifies the service, a `Product_Description` including descriptive information about the product and an `Added_Date` field defining when the product was added to the SDP.

[00276] The data model 1400 also includes an `SDP_Product Category` entity 1442. This entity defines category information for the product. The `SDP_Product Category` entity 1442 includes a `Product_ID` field and a `Category_ID` field which function as both primary keys and foreign keys. As a foreign key, the `Product_ID` field is linked to the field in the `SDP_Product` entity 1440 with the same name. Similarly, as a foreign key, the `Category_ID` field is linked to the field in a `REF_Category` entity 1444 with the same name.

[00277] The `REF_Category` entity 1444 stores information about the product category. It includes a `Category_ID` field as a primary key. The entity 1444 further includes fields such as `Category_Name`, `Category_Description` and `Category_type_ID`. This last field may be used as a foreign key and is linked to a field with the same name in a `REF_Category_Type` entity 1446. The `REF_Category_Type` entity 1446 includes the field `Category_Type_ID` as a primary key. The `REF_Category_Type` entity 1446 further includes fields `Category_Type_Name` and `Category_Type_Description` for storing additional information about a category type for a product.

[00278] The data model 1400 further includes and SDP_Product_Configuration entity 1448. The SDP_Product_Configuration entity 1448 stores information about the configuration of a particular product. The SDP_Product_Configuration entity 1448 includes a field labeled Configuration_ID which serves as a primary key for the entity 1448. The SDP_Product_Configuration entity 1448 further includes fields storing configuration information labeled Product_ID, which also serves as a foreign key, Configuration_Name and Configuration_Description.

[00279] The data model 1400 further includes a REF_Billing_Schema entity 1450 which stores information about how a service is billed. The REF_Billing_Schema entity 1450 includes a primary key field labeled Billing_Schema_ID which stores an identity for the billing schema. The REF_Billing_Schema entity 1450 further includes a foreign key labeled Category_ID which is linked to the field of the same name in the REF_Category entity 1444. The REF_Billing_Schema entity 1450 further includes, in this embodiment, several other fields for storing data about the billing schema, including Billing_Schema_Name, Billing_Schema_Description, Billing_Schema_Version, Billing_Schema_XMLNS:XSI; Billing_Schema_XSI:SchemaLocation, Billing_Schema_IPDRREcorder and MXLNS.

[00280] The data model 1400 also includes an SDP_Service entity 1452. The SDP_Service entity 1452 stores information about a service offered as part of the SDP. The SDP_Service entity 1452 includes a primary key labeled Service_ID, which stores identification information for the service. The SDP_Service entity 1452 further stores information about the service in several fields, including Service_Name, which stores a name of the service, Service_Description which stores a description of the service, Product_ID and Category_ID which store identifiers of the product and category for the service. Further, the SDP_Service entity 1452 includes fields labeled Type, which stores a category type, Start_Date and End_Date, which store information about the timing of provision of the service, and a Status_ID field.

[00281] The data model 1400 further includes the SDP_Service_Variant entity 1410 referenced above. The SDP_Service_Variant entity 1410 stores information about a service variant. A Service Variant is a catalog unit derived from a Service that contains configured details and attributes that define a set of unit of service. The SDP system supports creating different variants of every service, so that the service is offered to different types of target users. The SDP_Service_Variant entity 1410 includes a field labeled Service_Variant_ID which stores identification information for the service variant. The Service_Variant_ID operates as a primary key for the SDP_Service_Variant entity 1410.

[00282] The SDP_Service_Variant entity 1410 further includes a field labeled Service_ID which operates as a foreign key for the SDP_Service_Variant entity 1410. The field Service_ID is linked to the field of the same name in the SDP_Service entity 1452. The SDP_Service_Variant entity 1410 further includes fields which store information defining the service variant, including Service_Variant_Name, which stores the name of the service variant, Service_Variant_Description, which stores a description of the service variant, Base_Service_Variant_ID and Reseller_ID.

[00283] The data model 1400 further includes a SDP_Service_Dependency entity 1454. The SDP_Service_Dependency entity 1454 defines information about related services and respective dependencies among related services. The SDP_Service_Dependency entity 1454 includes a field labeled Service_ID which operates as a first foreign key for the SDP_Service_Dependency entity 1454, and is linked to the field of the same name in the SDP_Service entity 1452. Further, the SDP_Service_Dependency entity 1454 includes a second foreign key labeled Related_Service_ID which is also linked to the SDP_Service entity 1452. In this embodiment, the SDP_Service_Dependency entity 1454 further includes fields which define the service dependency, labeled Dependency_Name and Dependency_Description.

[00284] The data model 1400 still further includes a LNK_Service_Payment_Term entity 1456. The LNK_Service_Payment_Term

entity 1456 includes two fields which can operate as primary keys and foreign keys. The first is labeled `Payment_Term_ID`, which is linked to a field with the same label in a `REF_Payment_Term` entity 1458 (FIG. 14c). The second is labeled `Service_ID` and is linked to the field with the same name in the `SDP_Service` entity 1452.

[00285] The data model 1400 further includes an `SDP_Profile_Pricing` entity 1460. The `SDP_Profile_Pricing` entity 1460 includes a primary key field labeled `Profile_Pricing_ID`. The `SDP_Profile_Pricing` entity 1460 further includes a first foreign key field labeled `Currency_ID` which is linked to a field with the same name in a `REF_Currency` entity 1470 (FIG. 14c). Further, the `SDP_Profile_Pricing` entity 1460 includes a second foreign key field labeled `Payment_Model_ID`, which is linked to the field having the same label in a `REF_Payment_Model` entity 1472 (FIG. 14c). The `SDP_Profile_Pricing` entity 1460 further includes a third foreign key field labeled `Payment_Term_ID` which is linked to the field with the same label in the `REF_Payment_Term` entity 1458 (FIG. 14c).

[00286] The data model 1400 further includes a `LNK_Service_Currency` entity 1462. The `LNK_Service_Currency` entity 1462 includes two fields that can function as both primary keys and foreign keys. The first is labeled `Service_ID` and as a foreign key is linked to the field of the same name in the `LNK_Service_Payment_Term` entity 1456. The second is labeled `Currency_ID` and is linked as a foreign key to the `Currency_ID` field of the `REF_Currency` entity 1470 (FIG. 14c).

[00287] The data model 1400 further includes a `LNK_Billing_Element` entity 1464. The `LNK_Billing_Element` entity 1466 includes a primary key field labeled `Billing_Element_ID`. The `LNK_Billing_Element` entity 1466 further includes fields which can function as foreign keys, includes `Billing_Schema_ID`, `Billing_Element_Name` and `Billing_Element_Description`.

[00288] The data mode 1400 further includes an `SDP_Billable_Data_Record_Detail` entity 1466. The `SDP_Billable_Data_Record_Detail` entity 1466 includes a primary key field

labeled `Billable_Data_Record_Detail_ID`. The `SDP_Billable_Data_Record_Detail` entity 1466 further includes fields which function as foreign keys, including a field labeled `Billable_Data_Record_ID`, which point to the field of the same name in a `SDP_Billable_Data_Record` entity 1468; a field labeled `Billing_Element_ID`, which is linked to the field of the same name in the `LNK_Billing_Element` entity 1646. The

`SDP_Billable_Data_Record_Detail` entity 1466 further includes a field labeled `Billing_Element_Value`.

[00289] The data model 1400 further includes a `SDP_Billable_Data_Record` entity 1468. The `SDP_Billable_Data_Record` entity 1468 includes a field labeled `Billable_Data_Record_ID`, which serves as a primary key for the entity 1468. The `SDP_Billable_Data_Record` entity 1468 further includes a field labeled `Billing_Schema_ID` which functions as a foreign key. It is linked to a field of the same name in the `REF_Billing_Schema` entity 1450. The `SDP_Billable_Data_Record` entity 1468 further includes a field labeled `Bill_Feed_ID` which functions as a foreign key. It is linked to a field with the same label in the a `SDP_Bill_Feed` entity 1474 (FIG. 14c).

[00290] Referring now to the portion of the data model 1400 shown in FIG. 14c, the data model 1400 further includes the `REF_Payment_Term` field entity 1458. The `REF_Payment_Term` field entity 1458 includes a primary key field labeled `Payment_Term_ID`. The `REF_Payment_Term` field entity 1458 further includes two additional fields configured to store information about the payment term. These are a `Payment_Term_Name` field and a `Payment_Term_Desc` field.

[00291] The data model 1400 further includes a `LNK_Service_Payment_Model` entity 1476. The `LNK_Service_Payment_Model` entity 1476 includes two fields that can function as either a primary key or a foreign key. First is a field labeled `Payment_Model_ID` which is linked to the field with the same name in the `REF_Payment_Model` entity 1472. Second is a field labeled `Service_ID` which is linked to the `Service_ID` field of the `SDP_Service` entity 1452 (FIG. 14b).

[00292] The REF_Payment_Model entity 1472 includes the field labeled Payment_Model_ID as a primary key. Further, the REF_Payment_Model entity 1472 includes two field configured to store data that is descriptive of the payment model. First is a field labeled Payment_Model_Name. Second is a field labeled Payment_Model_Desc.

[00293] The data model 1400 further includes the SDP_IOM_Request_Type entity 1478. SDP_IOM_Request_Type entity 1478 includes the field labeled IOM_Request_Type_ID as a primary key. The SDP_IOM_Request_Type entity 1478 further includes fields storing additional information defining the request to the integrated order manager (IOM), including a field labeled IOM_Request_Type_Name and a field labeled IOM_Request_Type_Description.

[00294] The data model 1400 further includes a SDP_Subscription_Service_Variant entity 1480. This entity stores information about a subscription service variant. This entity includes a primary key field labeled Subscription_Service_Variant_ID which identifies the subscription service variant. The SDP_Subscription_Service_Variant entity 1480 includes several fields with function as foreign keys. The first is labeled Currency_ID and is linked to the field of the same name in the REF_Currency entity 1470. The second field is labeled Payment_Model_ID and is linked to the same named field of the REF_Payment_Model entity 1472. The third field is labeled Payment_Term_ID and is linked to the same named field of the REF_Payment_Term entity 1458. The fourth field is labeled Status_ID and is linked to the field with the same name of a REF_Status entity 1486. The fifth field is labeled Solution_Offer_ID and is linked to a field with the same name of a SDP_Solution_Offer entity 1488.

[00295] The SDP_Subscription_Service_Variant entity 1480 further includes a sixth field labeled Solution_Service_Variant_ID which is linked to a field with the same label in a SDP_Solution_Service_Variant entity 1490. The SDP_Subscription_Service_Variant entity 1480 further includes a field labeled Subscription_ID which is linked to the field with the same name of the SDP_Subscription entity 1484. The SDP_Subscription_Service_Variant entity

1480 further includes fields labeled `Service_Variant` name to store the name of the service variant and `Service_Variant_Description` to store a description of the service variant. Finally, in this embodiment, the `SDP_Subscription_Service_Variant` entity 1480 includes a field labeled `Pricing_Value`.

[00296] The data model 1400 further includes an `SDP_IOM_Request` entity 1482. This entity 1482 stores data about a request made to the SDP integrated order manager. The `SDP_IOM_Request` entity 1482 includes a primary key field labeled `IOM_Request_ID` which stores identification information for the request to the IOM. The `SDP_IOM_Request` entity 1482 further includes a first foreign key field, labeled `Status_ID`. This field is linked to the field of the same name in the `REF_Status` entity 1486. The `SDP_IOM_Request` entity 1482 further includes a second foreign key field, labeled `IOM_Request_Type_ID`, which is linked to the field with that name of the `DSO_IOM_Request_Type` entity 1478. The `SDP_IOM_Request` entity 1482 includes other fields for storing data, including and `IOM_Request_Description` field, an `IOM_Request_Content` field and an `Affected_ID` field.

[00297] The data model 1400 further includes the `SDP_Subscription` entity 1484. This entity 1484 has a primary key field labeled `Subscription_ID`. Further, the `SDP_Subscription` entity 1484 includes fields labeled `Subscription_Name`, `Subscription_Description` and `User_ID` for storing information about the SDP Subscription.

[00298] The data model 1400 still further includes the `REF_Status` entity 1486, which has a primary key field labeled `Status_ID` and stores an identifier. The `REF_Status` entity 1486 further includes a `Status_Name` field for storing name information.

[00299] The data model 1400 further includes and `SDP_Solution_Offer` entity 1488. This entity 1488 has a primary key field labeled `Solution_Offer_ID` and a foreign key field labeled `Solution_ID`. The foreign key field is linked to the field with the same name of a `SDP_Solution` entity 1492 (FIG. 14d). The `SDP_Solution_Offer` entity 1488 includes additional fields for defining a solution

offer, including fields labeled Offer_Name, Offer_Description, Offer_Price, Offer_ID, Catalog_Offer_ID, Is_Catalog_Base and Status_ID.

[00300] Still further, the data model 1400 includes a SDP_VAR_Customer_Relations entity 1494. This entity has a primary key field VAR_Customer_Relations_ID. This entity in this example further has three foreign key fields, including a first field labeled Solution_ID which is linked to a field in the SDP_Solution entity 1492, a second field labeled Solution_Offer_ID which is linked to a field in the SDP_Solution_Offer entity 1488, and a third field labeled Solution_Service_Variant_ID, which is linked to a field of the same name in a SDP_Solution_Service_Variant entity 1490. The SDP_VAR_Customer_Relations entity 1494 further has fields labeled Reseller_ID and Customer_ID for storing identification information for some of the parties involved with this service and the SDP.

[00301] The SDP_Solution_Service_Variant entity 1490 includes a primary key labeled Solution_Service_Variant_ID which stores data defining the solution service variant. This entity 1490 further includes several fields that define the content of a solution service variant, including fields labeled Solution_ID, Service_Variant_ID, Service_Variant_Name,, Service_Variant_Description and Solution_Offer_ID. Other fields include Is_Catalog_Base, Catalog_Service_Variant_ID, Pricing_Value, Currency_ID,Payment_Model_ID, Payment_Term_ID and Status_ID.

[00302] The data model 1400 further includes a REF_Security_Group_Type entity 1496. This entity 1496 includes a primary key Security_Group_Type_ID which stores identification information. This entity also includes fields storing information about the security group, including a Security_Group_Type_ID field and a Security_Group_Type_Description field.

[00303] The date mode 1400 further includes a SDP_Subscription_Configuration entity 1498. This entity 1498 has a primary key field Subscription_Configuration_ID and in this embodiment, two foreign key fields. The first foreign key field is the Configuration_ID which points to a field in the SDP_Product_Configuration entity 1448 (FIG. 14b). The second foreign

key field is labeled `Subscription_ID` and points to the primary key field of the `SDP_Subscription` field 1484. The `SDP_Subscription_Configuration` entity 1498 includes other fields for storing information about the subscription configuration, including a `Configuration_Value` field.

[00304] Turning now to the portion of FIG. 14 shown in FIG. 14d, the data model 1400 includes a `LNK_Attribute_Detail_Attribute_Mask` entity 1502. This entity includes two foreign key fields. The first is labeled `Attribute_Detail_ID` and points to a like named field in an `SDP_Attribute_Detail` entity 1504. The second is labeled `Attribute_Mask_ID` and points to a field of an `SDP_Attribute_Mask` entity 1506.

[00305] The data model 1400 still further includes the `SDP_Solution` entity 1492. This entity 1492 has the primary key field labeled `Solution_ID` and several other fields for storing information. These fields include fields labeled `Customer_ID`, `Solution_Name`, `Solution_Description` and `Is_Reseller_Solution`.

[00306] The data model 1400 further includes a `LNK_Solution_Service_Group` entity 1508. This entity includes a field labeled `Solution_ID` which functions as both a primary key and a foreign key, a field labeled `Security_Group_ID` which functions as both a primary key and a foreign key and a field labeled `Solution_Service_Variant`, which also functions as both a primary key and a foreign key. Still further, the data model 1400 includes a `SDP_Security_Group` entity 1510 which has a primary key field labeled `Security_Group_ID`. This entity 1510 has a foreign key labeled `Security_Group_Type_ID` which points to a field of the same name in the `REF_Security_Group_Type` entity 1496 (FIG. 14c). Still further, this entity 1510 includes other fields to store information about the security group, including a `Security_Group_Name` field, a `Security_Group_Description` field, `Security_Group_Guide` field, a `Security_Group_Type_ID` field and a `Customer_ID` field.

[00307] The data model still further includes a `SDP_Workflow` entity 1512. This entity 1512 has a primary key labeled `Workflow_ID` and stores additional information in fields labeled `Orchestration_Name` and `Workflow_Type`.

[00308] The data model 1400 further includes the SDP_Attribute_Mask entity 1506, with a primary key field labeled Attribute_Mask_ID. This entity 1506 includes other fields for storing data about the attribute mask, including fields labeled Value_Type, Text_Value, Default_Value, Min_Value, Max_Value, Inc_Value, Begin_Date, End_Date and Value_Description.

[00309] The data model 1400 also includes an SDP_Attribute_Detail entity 1504. The SDP_Attribute_Detail entity 1504 has a primary key field labeled Attribute_Detail_ID and further includes two fields for further defining the SDP attribute, a first field named Attribute_Detail_Name and Attribute_Detail_Description.

[00310] The data model 1400 further includes a SDP_Attribute entity 1514. This entity 1514 includes an identifying primary key labeled Attribute_ID. It further includes a foreign key Attribute_Detail_ID which is linked to the same named field of the SDP_Attribute_Detail entity 1504. The SDP_Attribute entity 1514 further includes fields labeled Attribute_Table, Attribute_Table_Record_ID, Attribute_Value, Editable_Flag and Override_Flag.

[00311] The data model 1400 further includes an SDP_Request_Workflow entity 1516. This entity 1516 includes two fields which can serve both as primary keys and foreign keys. The first field is labeled Request_ID, which as a foreign key, points to a field with the same name in a SDP_Request entity 1518. The second field is labeled Workflow_ID and points to the field with the same name in the SDP_Workflow entity 1512. The SDP_Request_Workflow entity 1516 also includes fields named Schema_Name, Schema_ID and Status_ID containing other information about the request.

[00312] The data model 1400 further includes a REF_Action_Type entity 1520. This entity 1520 has a primary key labeled Action_Type_ID and two additional fields for storing information about an action type. The first field is labeled Action_Type and the second field is labeled Action_Table.

[00313] Further, the data model 1400 includes a UD_LNK_Customer_User entity 1522. This entity 1522 includes two fields that can function as both primary keys and foreign keys. The first is labeled Customer_ID and as a foreign key,

points to a field of the same name in a UD_Customer entity 1526. The second field is labeled User_ID and is linked to a UD_User entity 1524. The UD_User entity 1524 in turn includes the primary key SDP_UserID and has two additional fields for storing information about a user of the SDP system, labeled First_Name and Last_Name.

[00314] The data model 1400 further includes an SDP_Request entity 1518. This entity 1518 has as a primary key Request_ID. The entity 1518 further has as a foreign key a field labeled SDP_User_ID which is linked to the primary key of the UD_User entity 1524. The SDP_Request entity 1518 further includes fields labeled Schema_Name, Schema_ID and Status_ID for storing additional information about an SDP request.

[00315] The data model 1400 for SDP further includes an SDP_Action entity 1528. This entity 1528 includes a primary key labeled Action_ID and a foreign key labeled Action_TypeID which points to the primary key of the REF_Action_Type entity 1520. The SDP_Action entity 1528 further includes fields labeled Action_Value and Action_Identifier.

[00316] The data model 1400 further includes the UD_Customer entity 1530. This entity has a primary key labeled Customer_ID and an additional field labeled SDP_User_ID. Further, the data model 1400 includes Role_ID entity 1526. This entity 1526 has a primary key field labeled Role_ID and an additional field labeled Role_Description.

[00317] The data model 1400 includes a UN_LNK_User_Role entity 1526. This entity has two fields that can function as both primary keys and foreign keys. The first field is labeled Role_ID and as a foreign key, points to the primary key of the Role_ID entity 1526. The second field is labeled User_ID and points to the primary key of the UD_User entity 1524.

[00318] Further, the data model 1400 includes a UD_User_Address entity 1532. This entity has two fields that can function as both primary keys and foreign keys. The first field is labeled Address_ID and as a foreign key, points to the primary key of a SDP_Address entity 1534. The second field has a label of

SDP_User_ID and as a foreign key, points to the primary key of the UD_User entity 1524.

[00319] Finally, the data model 1400 includes the SDP_Address entity 1543. This entity 1534 has as a primary key a field labeled Address_ID. This entity 1543 includes several additional fields for storing address information, including Address_1, Address_2, City_ID, State_ID, Country_ID, and Postal_Code.

[00320] From the foregoing, it can be seen that the present application provides a service delivery platform, including method and apparatus for managing the delivery of a variety of services to customers or subscribers. The service delivery platform (SDP) is a platform architecture that fully exploits the convergence opportunities made available for the service providers (SPs) recently. The SDP creates, hosts, and manages services over different channels and end user devices. The implementation of the SDP will dramatically simplify service deployment and management, and allow the enterprise to develop new services more rapidly with the reduced development efforts

[00321] It is therefore intended that the foregoing detailed description be regarded as illustrative rather than limiting, and that it be understood that it is the following claims, including all equivalents, that are intended to define the spirit and scope of this invention.

2006201516 18 Apr 2012

- 80 -

The claims defining the invention are as follows:

1. A system for managing delivery of products, features and services, the system including:

a service catalogue including a set of products, features and services for provisioning and delivery to service subscribers; and

an order management module in data communication with the service catalogue, wherein the order management module enables provision and delivery of selected products, features, and services requested by a service subscriber, wherein the order management module decomposes the selected products, features, and services into atomic work orders workable by the service providers and releases the work orders to the service providers associated with the selected products, features and services, wherein the order management module maintains relationships across the work orders to determine dependencies for work order release to the service providers associated with the selected products, features and services.

2. A system according to claim 1 further including:

a third party portal for communication of data with a business partner of a service provider to fulfil the request from the service subscriber.

3. A system according to either claim 1 or claim 2 wherein the services include telecommunications services.

4. A service delivery platform (SDP) system including:

a service orchestration and integration environment which manages work flow for services, manages business processes for the service providers and integrates constituent applications, the service orchestration and integration environment operable for providing and delivering selected services requested by a service subscriber by decomposing the selected

2006201516 18 Apr 2012

- 81 -

services into atomic work orders workable by the service providers and releasing the work orders to the service providers associated with the selected services, wherein relationships are maintained across the work orders to determine dependencies for work order release to the service providers associated with the selected products, features and services;

a service creation environment for creation and modification of services, including a service provider interface to receive provider service definition data about provisioning and delivery of a first selected service from a service provider;

a service subscriber interface to receive subscriber service definition data about provisioning and delivery of a second selected service from the service subscriber;

a service deployment environment to automatically deploy the selected services;

a service assurance environment to identify faults in services; and

a unified directory accessible by the service orchestration and integration environment, the service creation environment, the service deployment environment, and the service assurance environment, the unified directory including a standardised structure that allows services to be dynamically configured based on the provider service definition data and the subscriber service definition data.

5. A SDP according to claim 4 wherein the unified directory is configured to store information regarding subscriber status, services subscribed by the user and details of a user's device to access the subscribed services.

6. A SDP according to claim 5 further including a database to store data for common access by the service orchestration and integration environment, the service creation environment, the service deployment environment and the service assurance environment.

2006201516 18 Apr 2012

- 82 -

7. A SDP according to claim 6 further including a data model defining relations among data stored in the database for subscribers, subscription accounts and related information.

8. A SDP according to claim 7 wherein the data model determines data type conversion rules for the data transformations between applications accessing the data stored in the database.

9. A service delivery method including:

at a server, receiving a request from a service provider for creating a new service that is not in a catalogue of services, wherein the catalogue of services has a standard service catalogue structure for a plurality of telecommunications services to be offered, the standard service catalogue structure defining elements to be included in new service definitions;

in response, at the server, requesting product definitions from the catalogue of services;

at the server, receiving the product definitions and providing the product definitions to the service provider, wherein the product definitions are provided in work orders and relationships are maintained across the work orders to determine dependencies for releasing the work orders to the service providers associated with the selected products, features, and services;

at the server, receiving a request for service definitions associated with a selected product definition;

at the server, receiving from the service provider service information for the new service based on the service definitions and accessing a stored common data model of a service delivery platform including the catalogue of services, the received service information including the elements defined by the standard service catalogue structure;

at the server, automatically creating facilities for providing the new service, including extending the common data model with information

2006201516 18 Apr 2012

- 83 -

about the new service, the provider of the new service and subscribers to the new service, including the received elements defined by the standard service catalogue structure; and

at the server, receiving a request from a service subscriber to initiate provision of the new service to the service subscriber.

10. A method according to claim 9 wherein storing data includes:

storing in a service entity information about a service;

storing in a product entity linked to the service entity, information about a product offered by a service provider; and

storing in a service variant entity linked to the service entity, information about a service variant.

11. A method according to claim 10 wherein storing data further includes:

storing in an offer entity information about a service offer; and

storing in an offer service variant entity a link between the offer entity information and the service entity.

12. A method according to claim 10 or claim 11 further including:

at the server, providing a web service for interaction with the service delivery platform by the service provider, the service subscriber and a business partner of the service provider;

at the server, at least one of:

receiving the request to provision the new service via the web service from one of the service subscriber and the business partner of the service provider; or

receiving a request to create a new service via the web service from the service subscriber; and

at the server, providing the product definitions via the web service.

2006201516 18 Apr 2012

- 84 -

13. A method according to any one of claims 9 to 12 wherein automatically creating facilities includes:

adding an application to provide the new service; and

mapping an application data model for the added application to the common data model without mapping to other existing applications.

14. A service delivery platform for telecommunications services, the service delivery platform including:

a service creation environment accessible by service providers and service subscribers to create the telecommunication services;

a service deployment environment for managing the telecommunication services in one or more networks;

a service enablement environment to execute the telecommunication services;

a service orchestration and integration environment operable to decompose selected telecommunication services requested by a service subscriber into atomic work orders workable by the service providers, to release the work orders to the service providers associated with the selected telecommunication services, and to track and monitor the provision of the telecommunication services, wherein relationships are maintained across work orders to determine dependencies for work order release to the service providers associated with the selected products, features, and services; and

a common data model accessible by each respective environment for creating and modifying services and accounts of the service delivery platform, the common data model arranged according to a standardized structure for services for specification of elements of a telecommunication service.

15. A service delivery platform according to claim 14 wherein the service deployment environment includes:

2006201516 18 Apr 2012

- 85 -

a configuration management module to manage attributes and values for a service.

16. A service delivery platform according to claim 14 or claim 15 wherein the service enablement environment includes:

a user interaction space for the service subscribers and the service providers.

17. A system according to any one of claims 1 to 3, further including:

a service provider interface to communicate data about the provisioning and delivery of the selected products, features, and services with the service providers;

a service subscriber interface to communicate data about the provisioning and delivery of the selected products, features, and services with the service subscriber; and

a service delivery database organized in part using a common data model to define a standardized structure for the service catalog to enable access to provisioning and delivery of the selected products, features, and services.

18. A system according to any one of claims 1 to 3, wherein the order management module tracks relationships across the work orders to determine a sequence for releasing the work orders to the service providers associated with the selected products, features, and services.

19. A system according to any one of claims 1-3, wherein the order management module tracks the work orders until completion, identifies which of the work orders is successfully provisioned, and provides notification when there is a change in status of a work order.

2006201516 18 Apr 2012

- 86 -

20. A system for managing delivery of a plurality of unrelated services according to claim 1, or a service delivery platform according to claim 4 or claim 14, or a service delivery method according to claim 9, substantially as hereinbefore described with reference to the accompanying drawings.

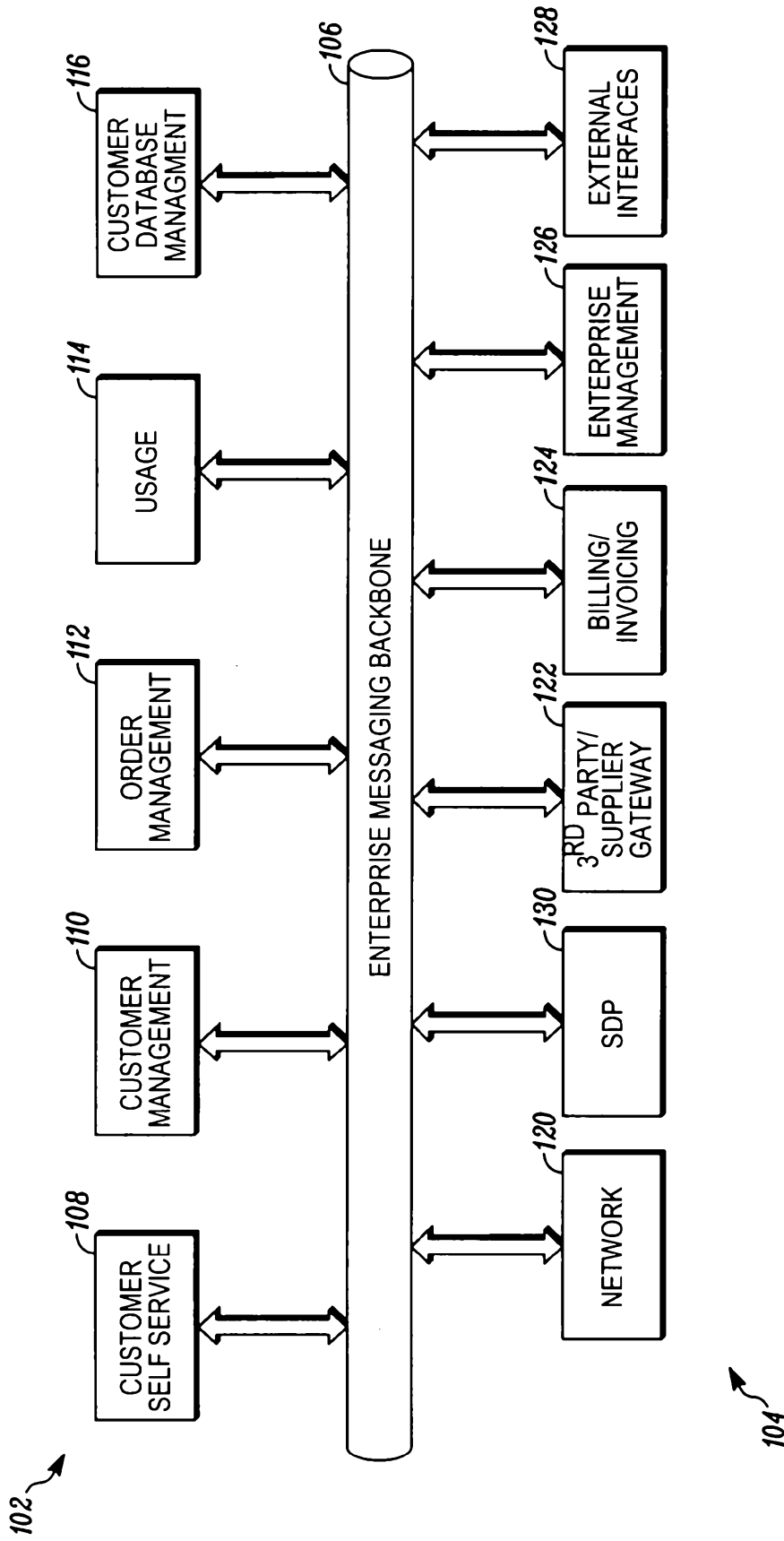


FIG. 1

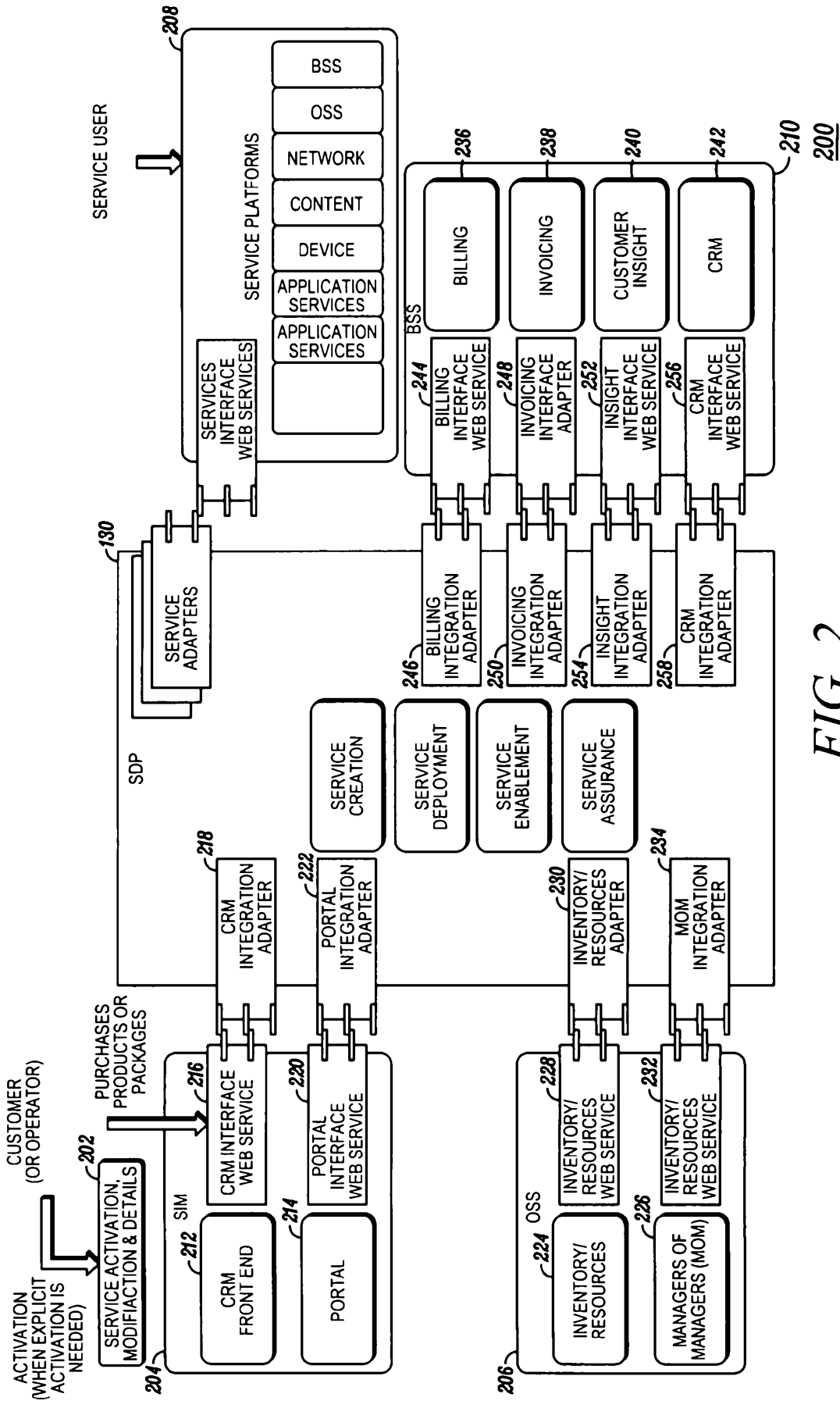
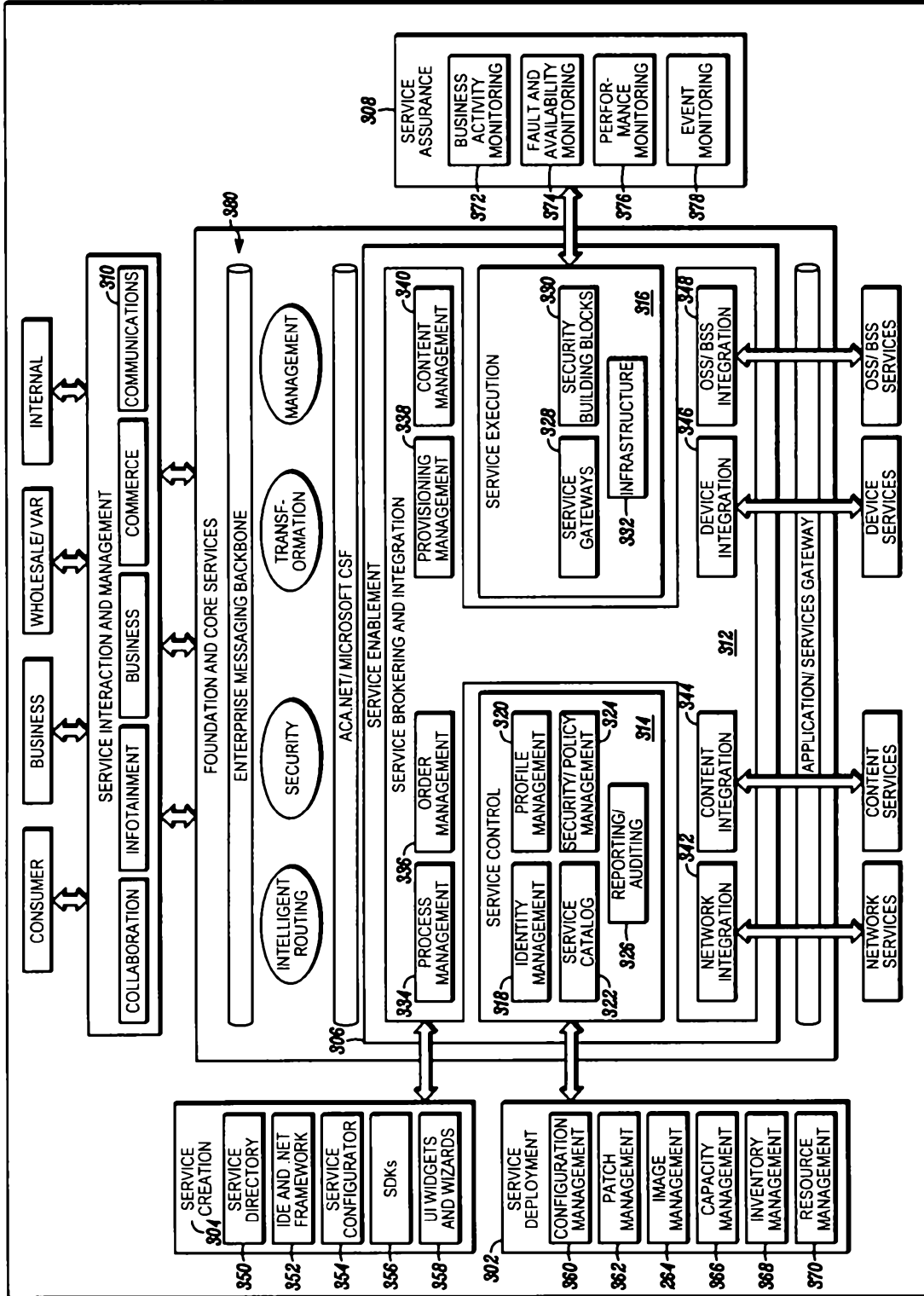


FIG. 2



300

FIG. 3

FIG. 4

FIG. 4A FIG. 4B FIG. 4C

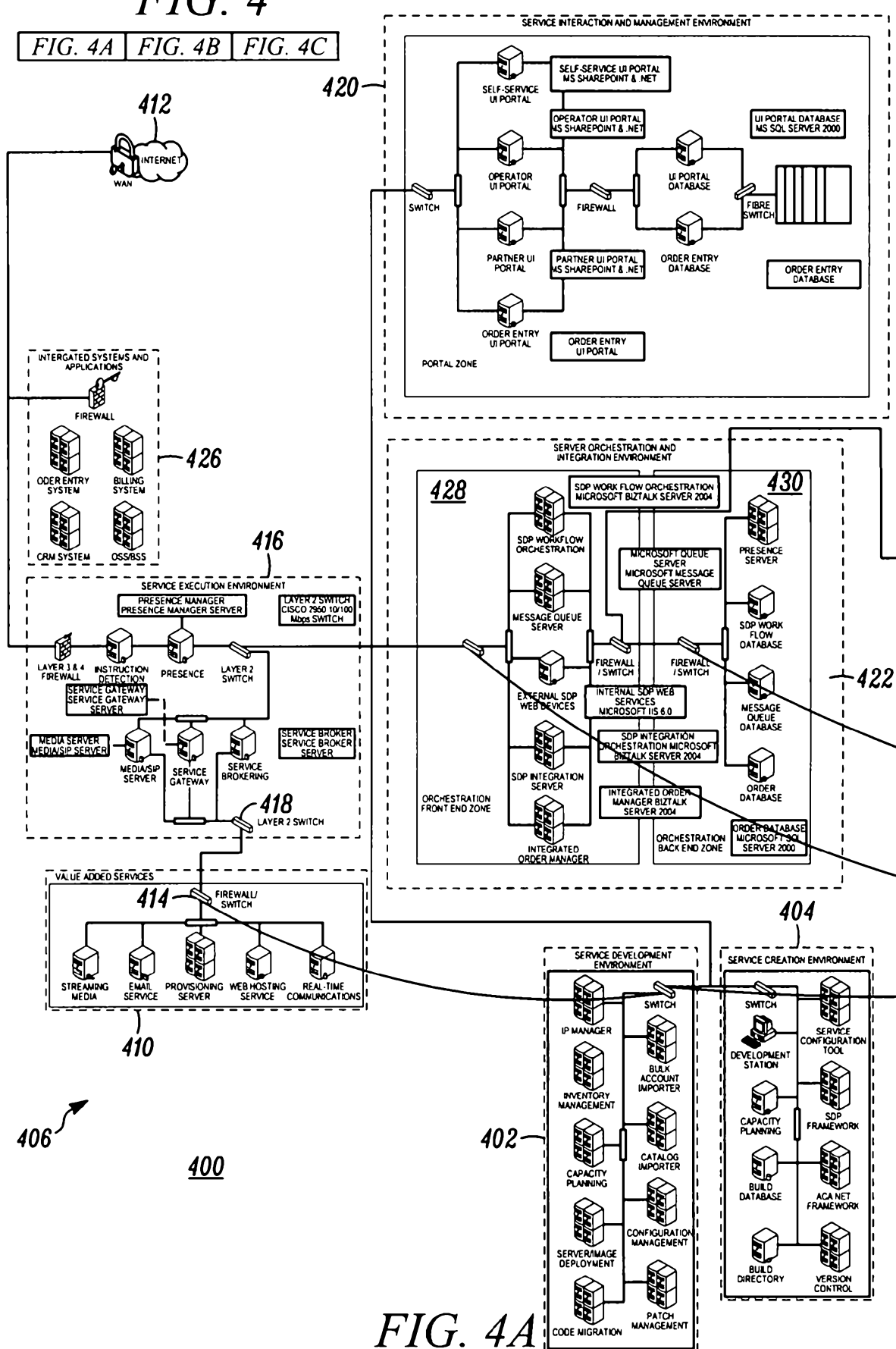


FIG. 4A

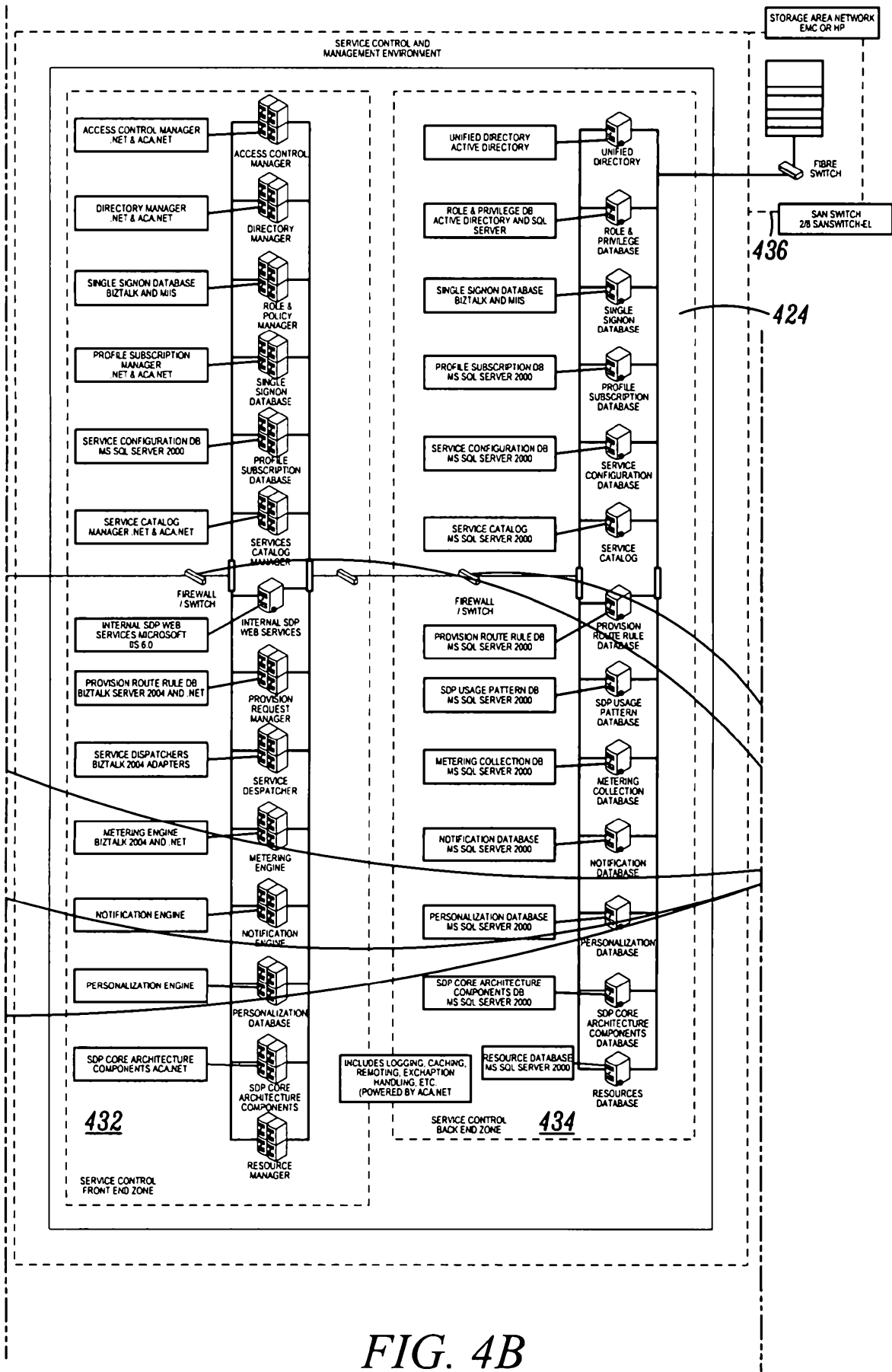
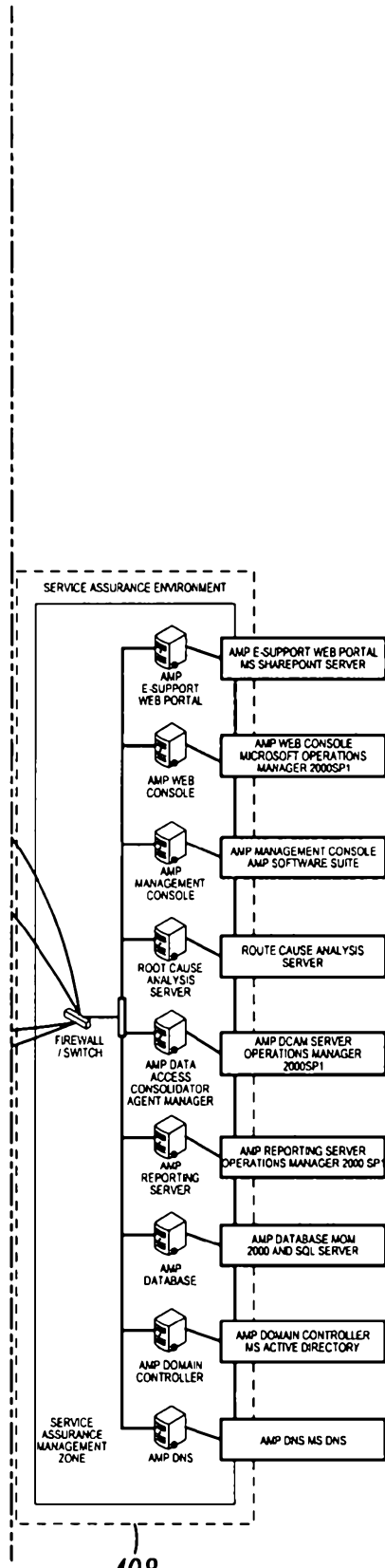
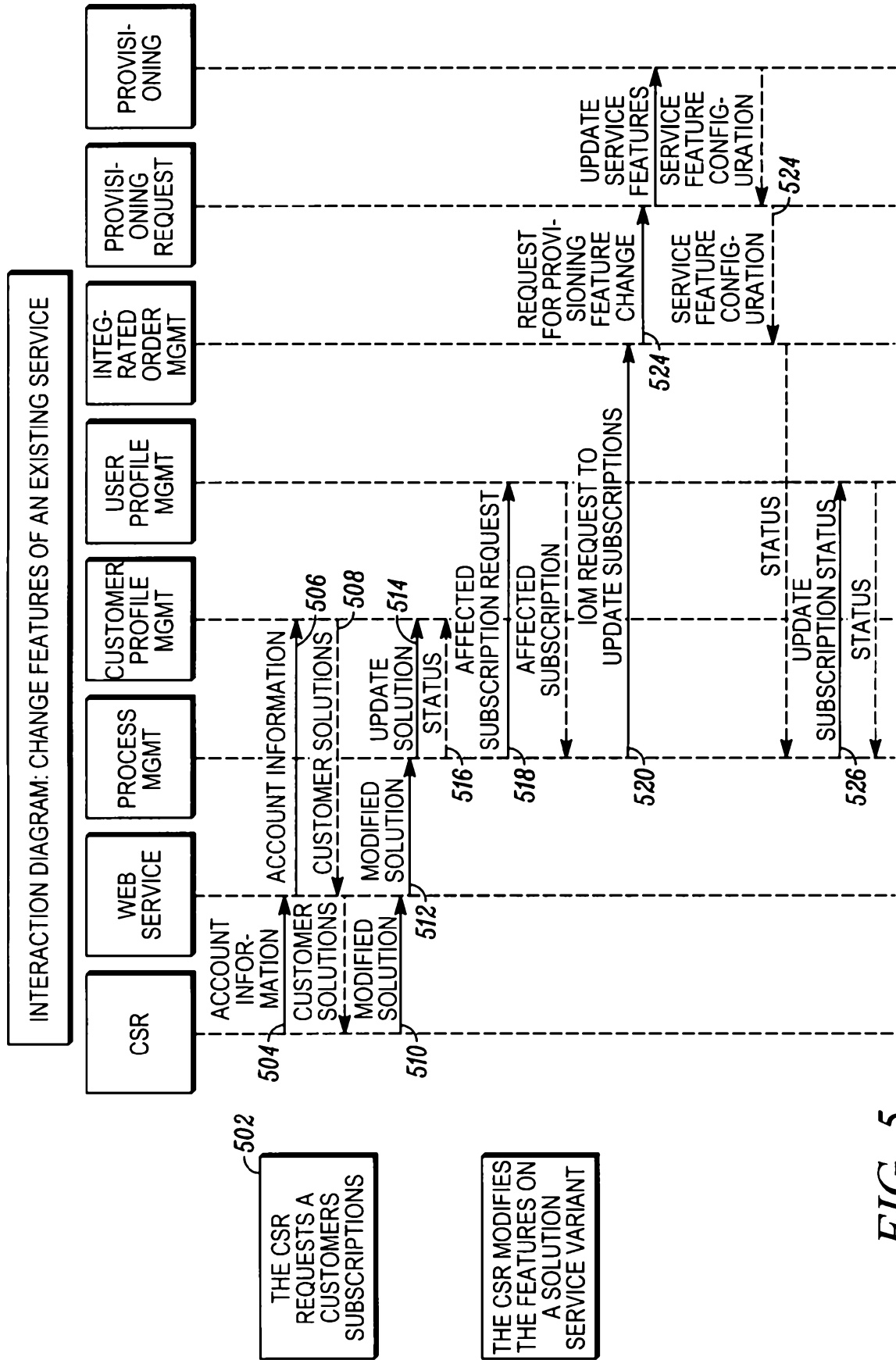


FIG. 4B



408
FIG. 4C



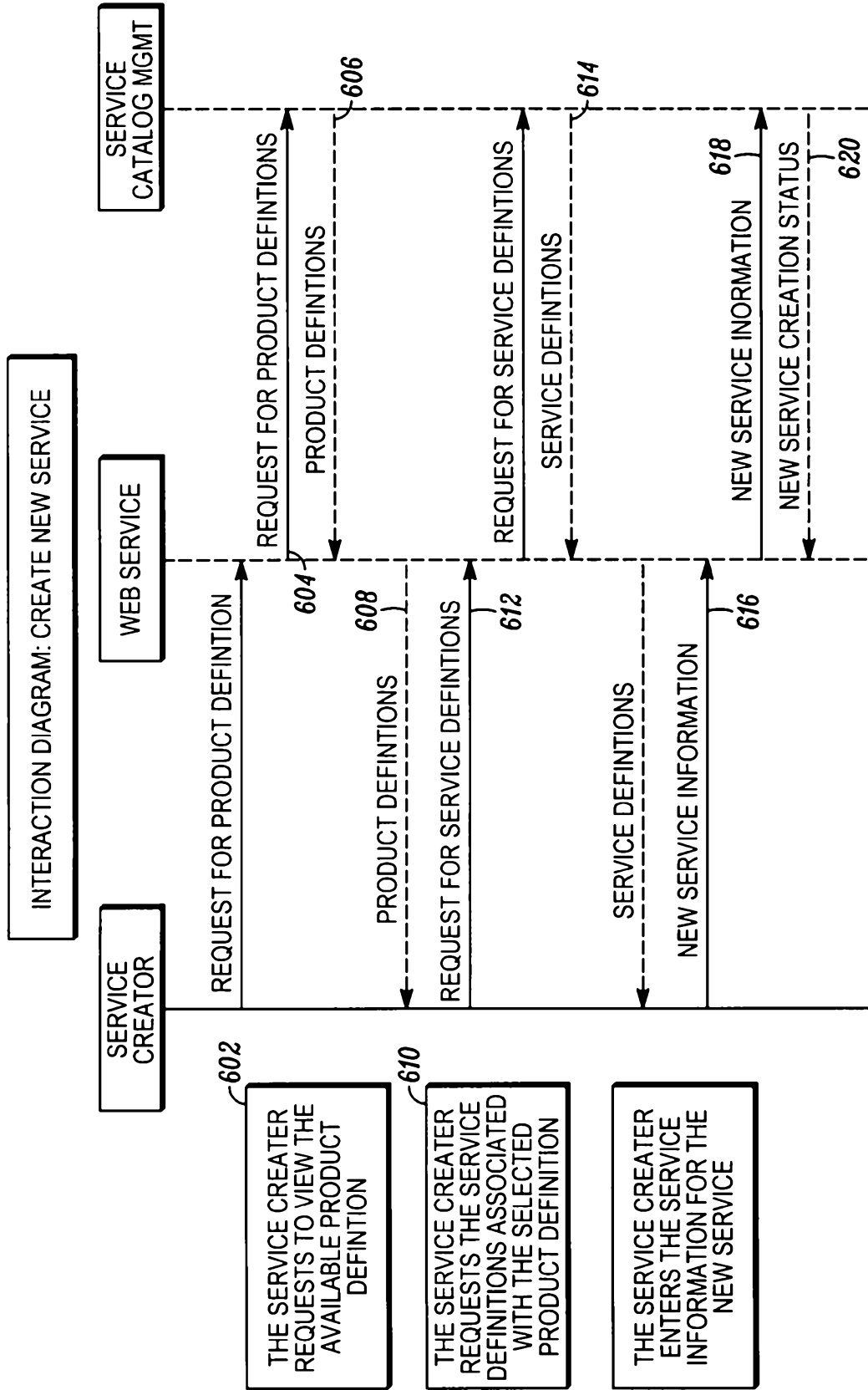


FIG. 6

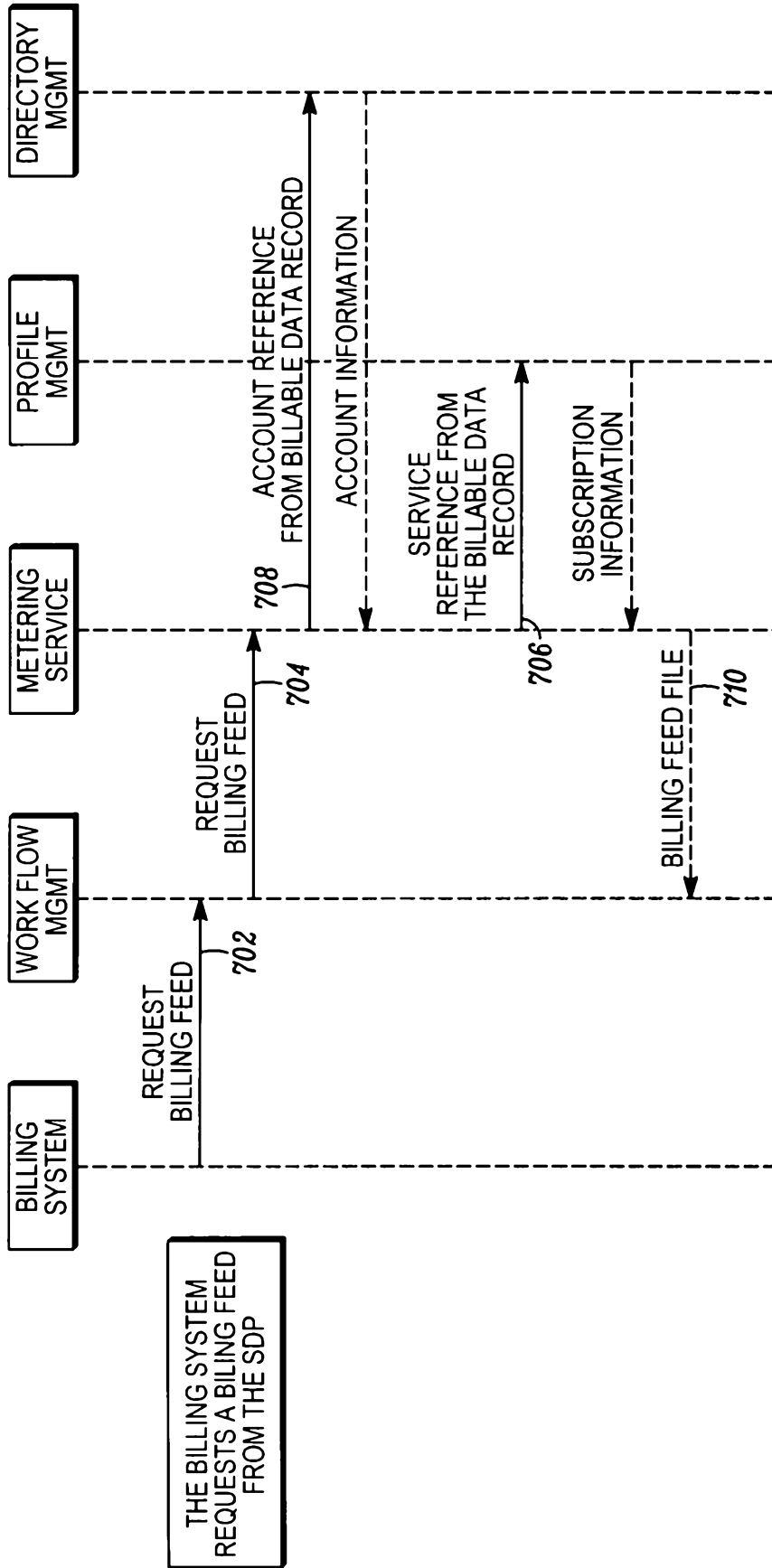


FIG. 7

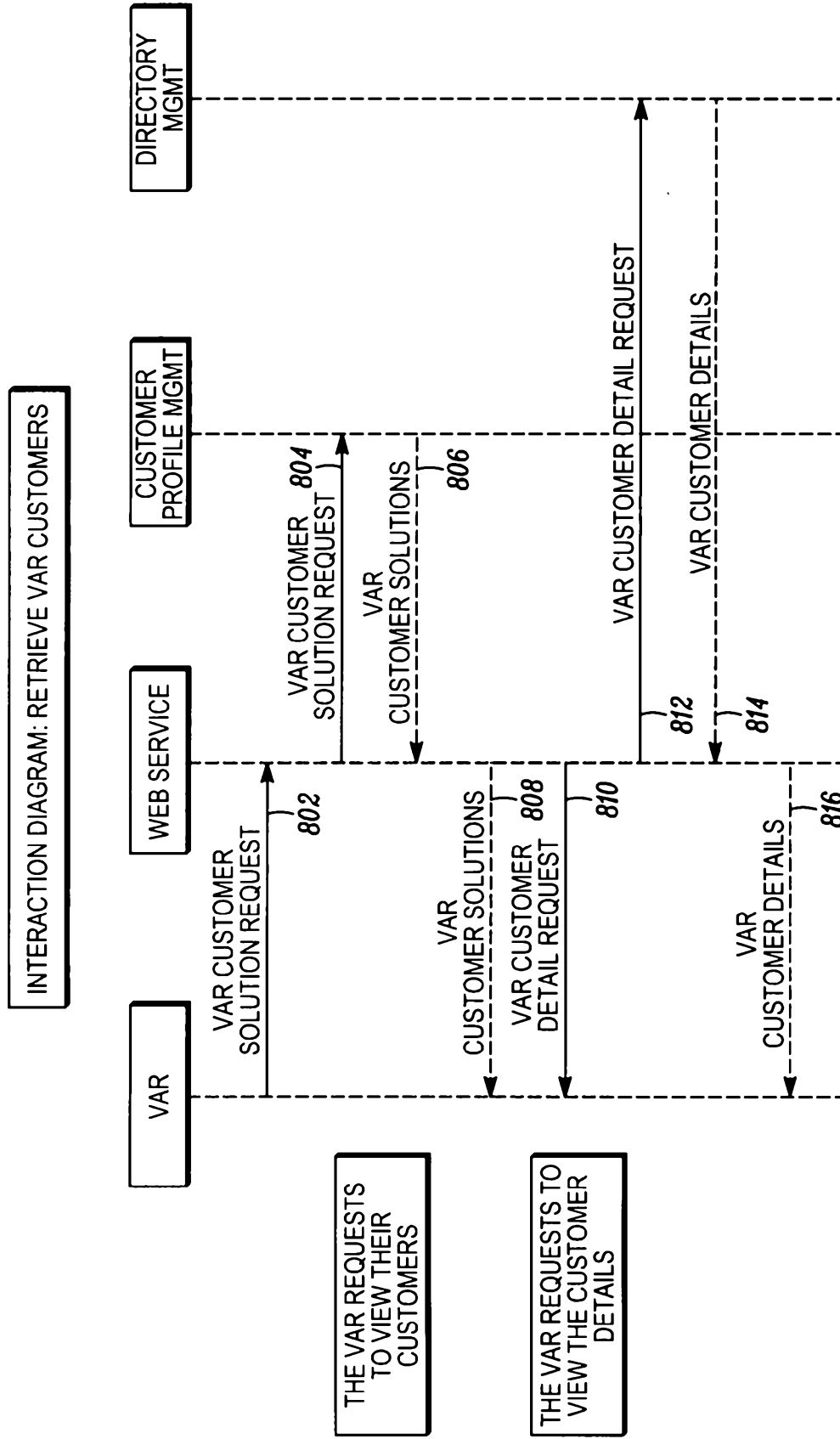


FIG. 8

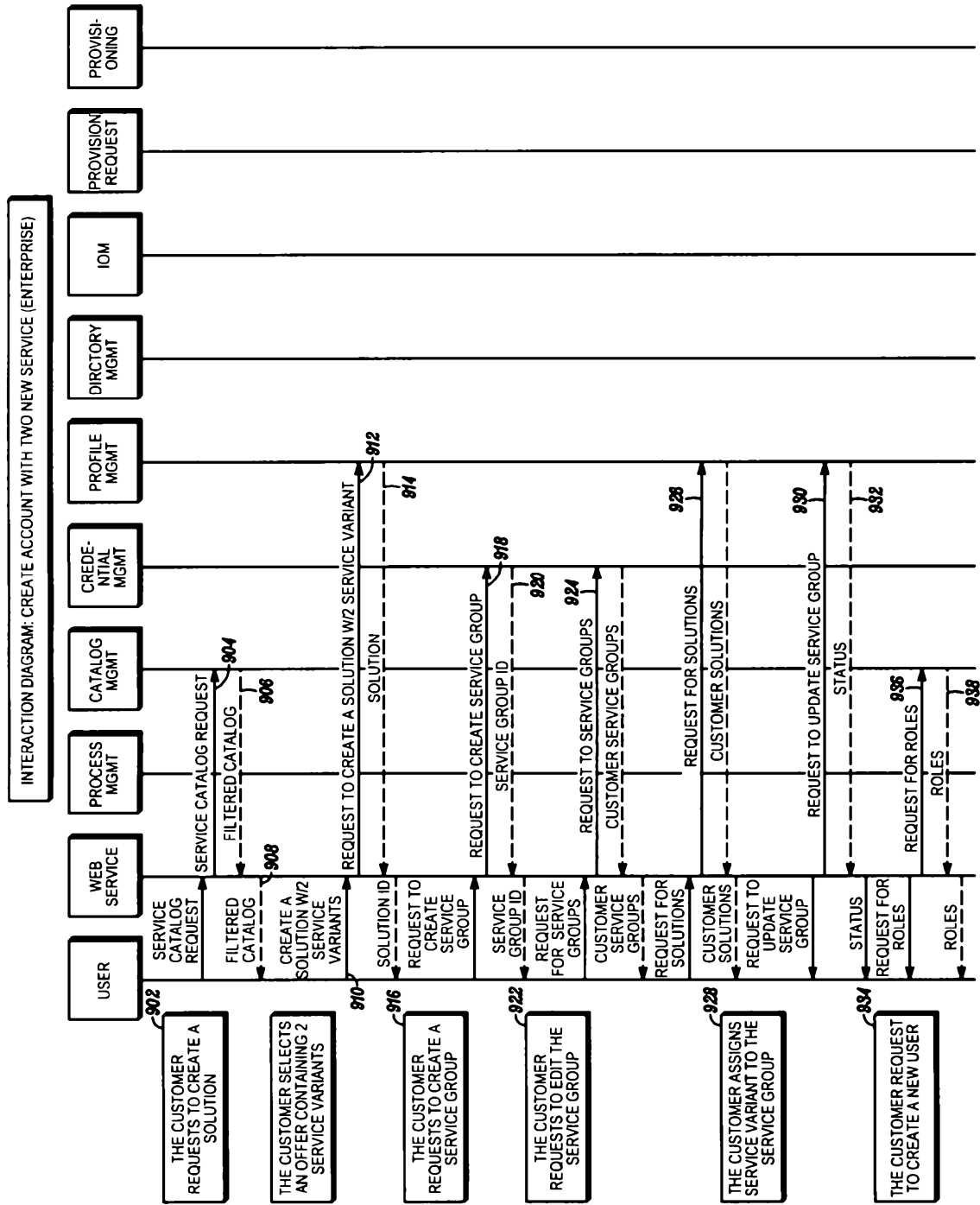


FIG. 9A

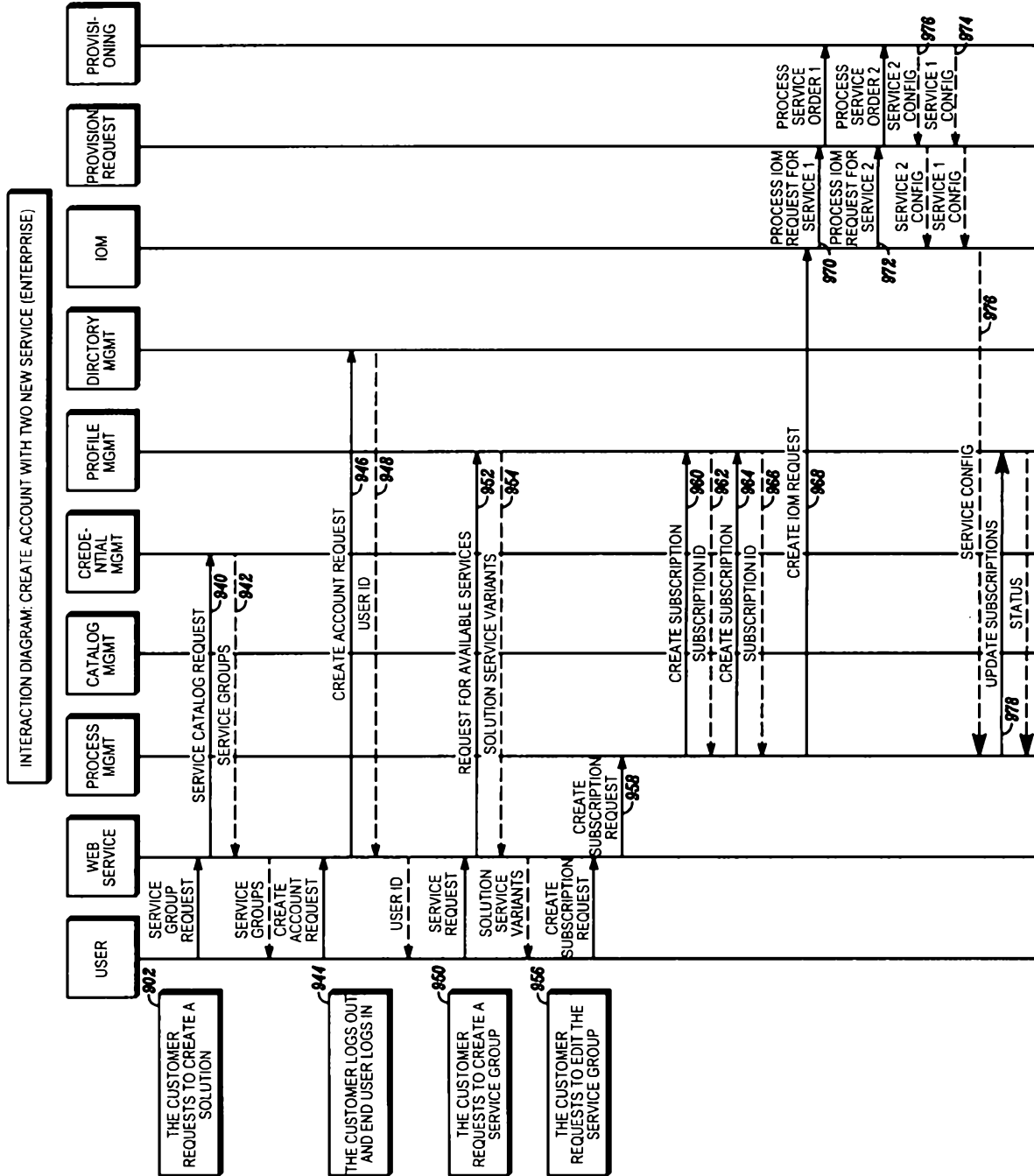


FIG. 9B

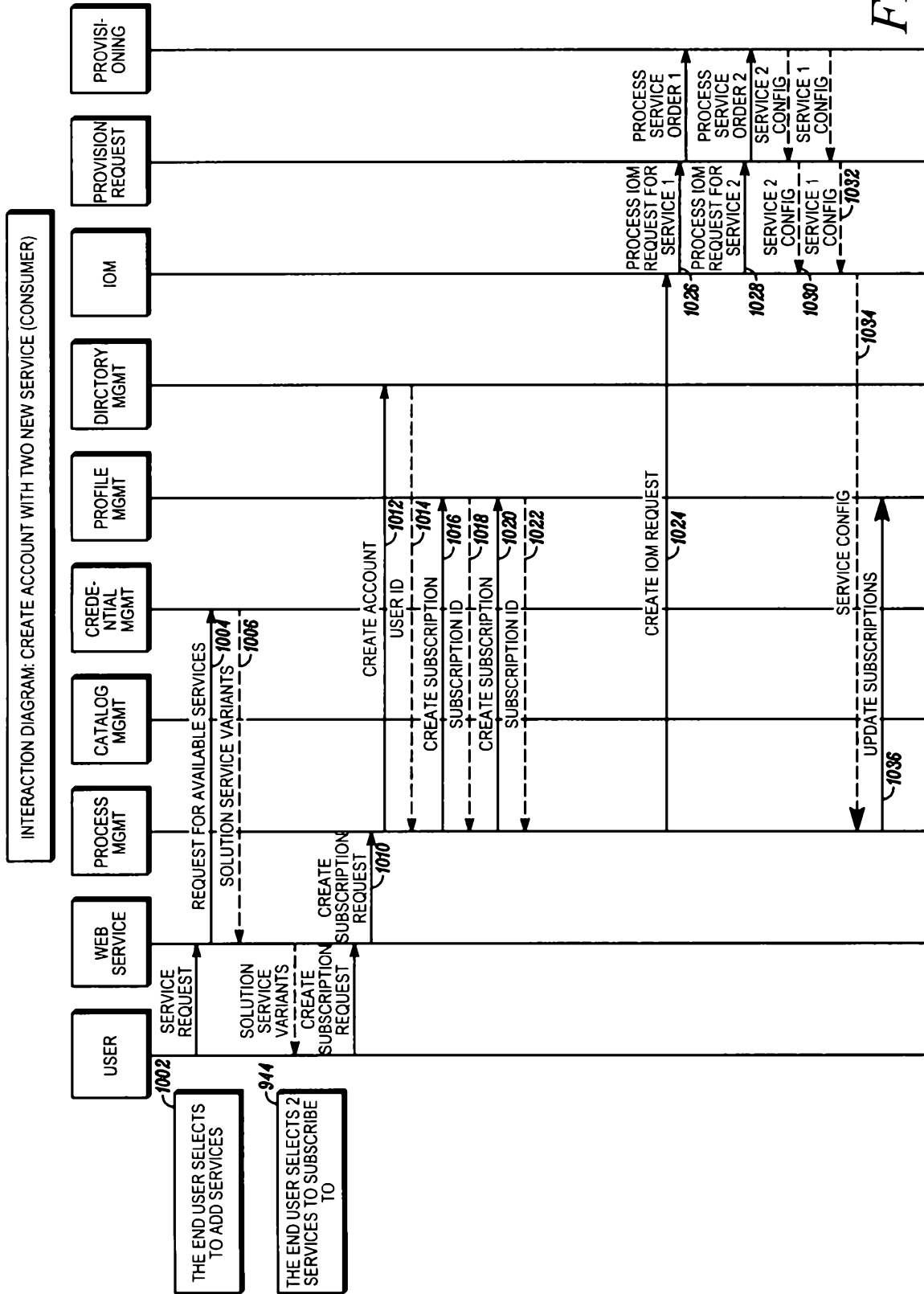


FIG. 10

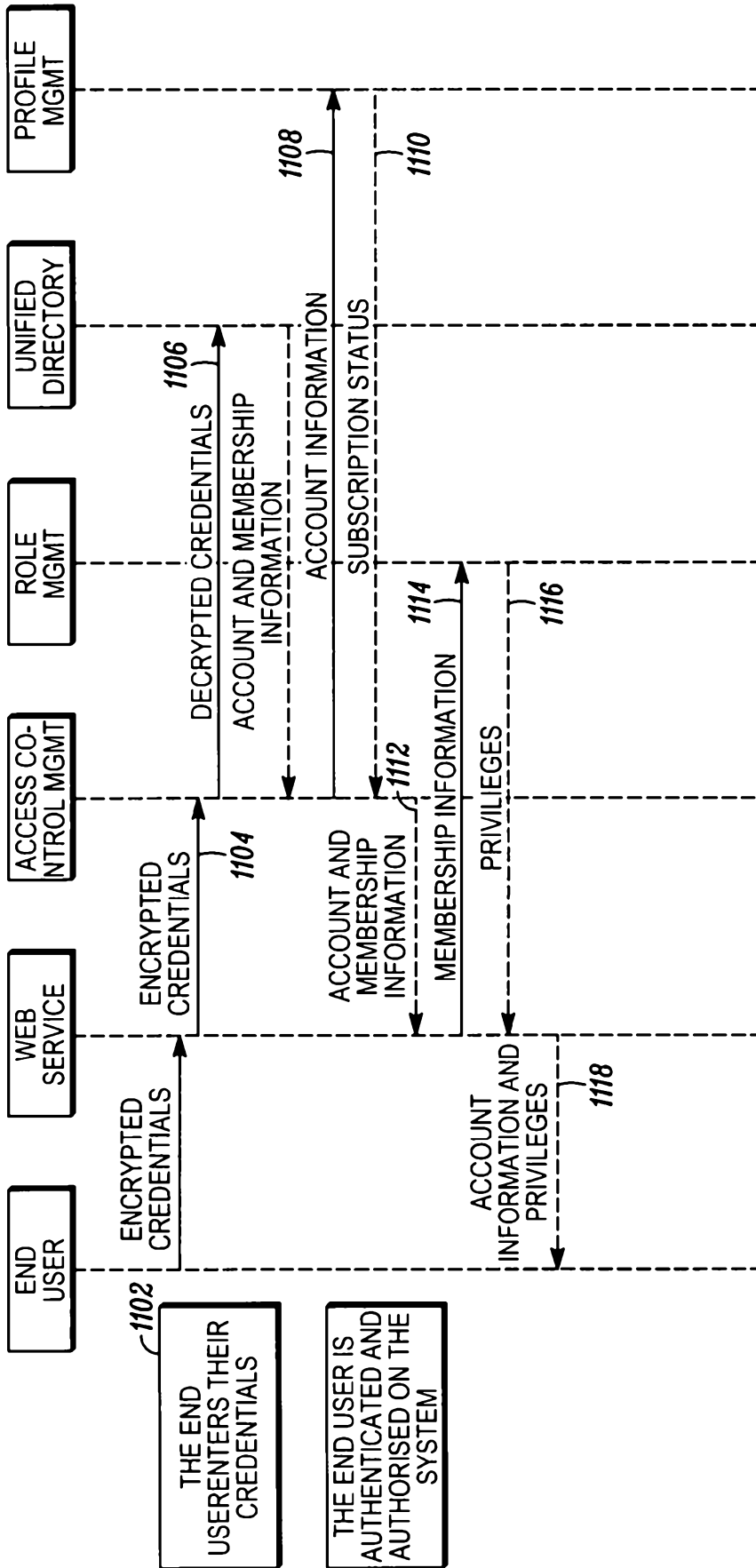


FIG. 11

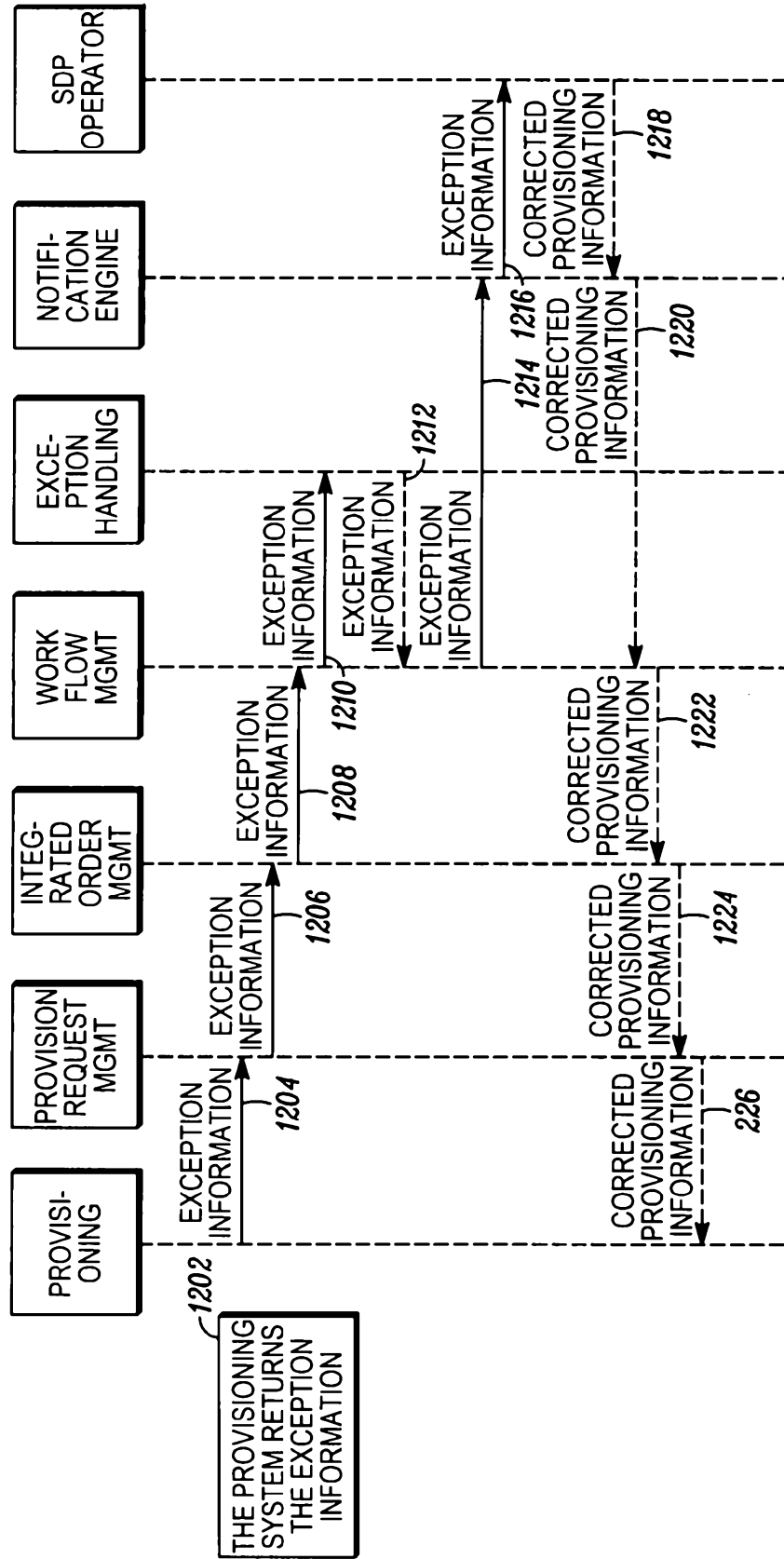


FIG. 12

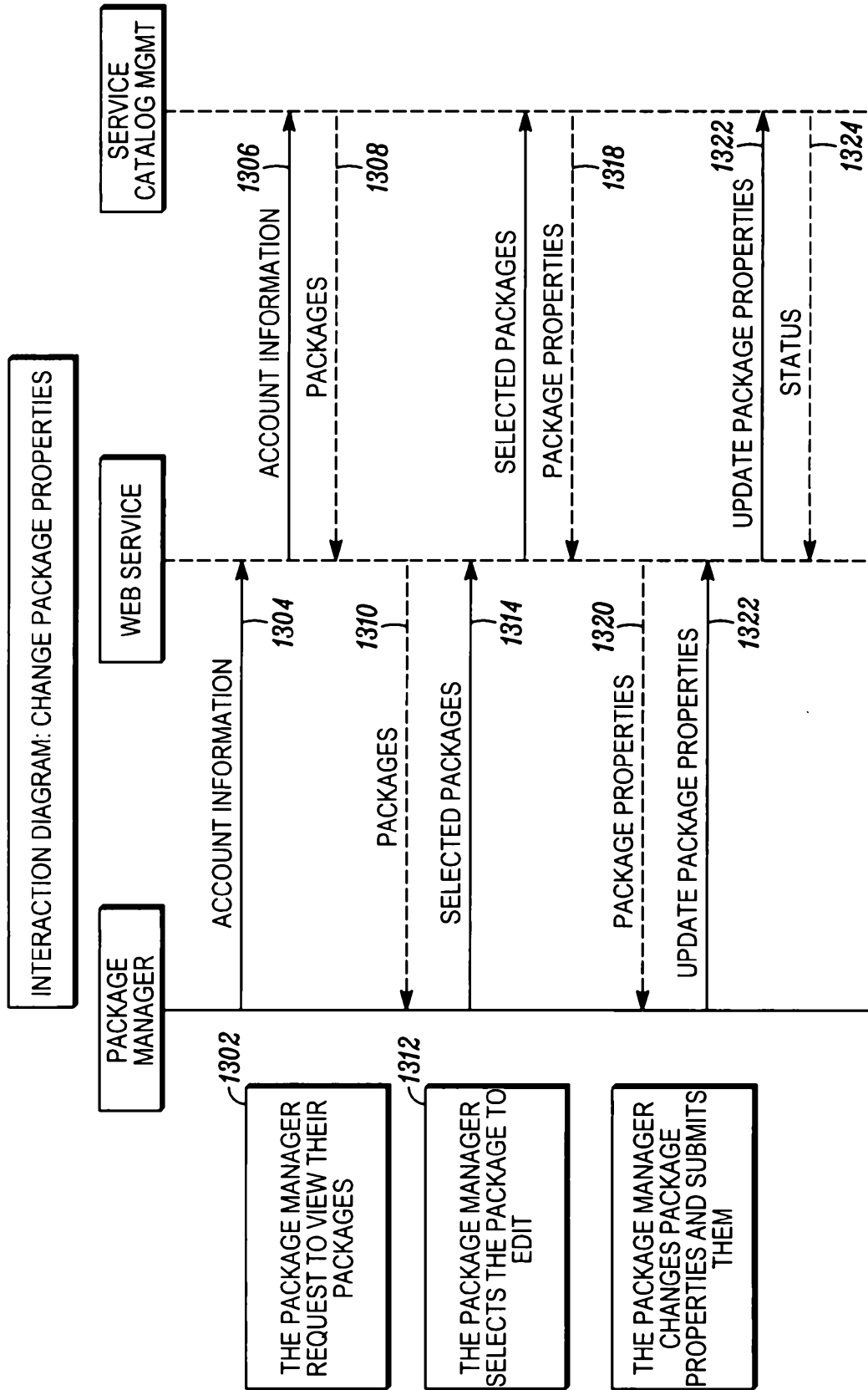


FIG. 13

FIG. 14

FIG. 14D FIG. 14C FIG. 14B FIG. 14A

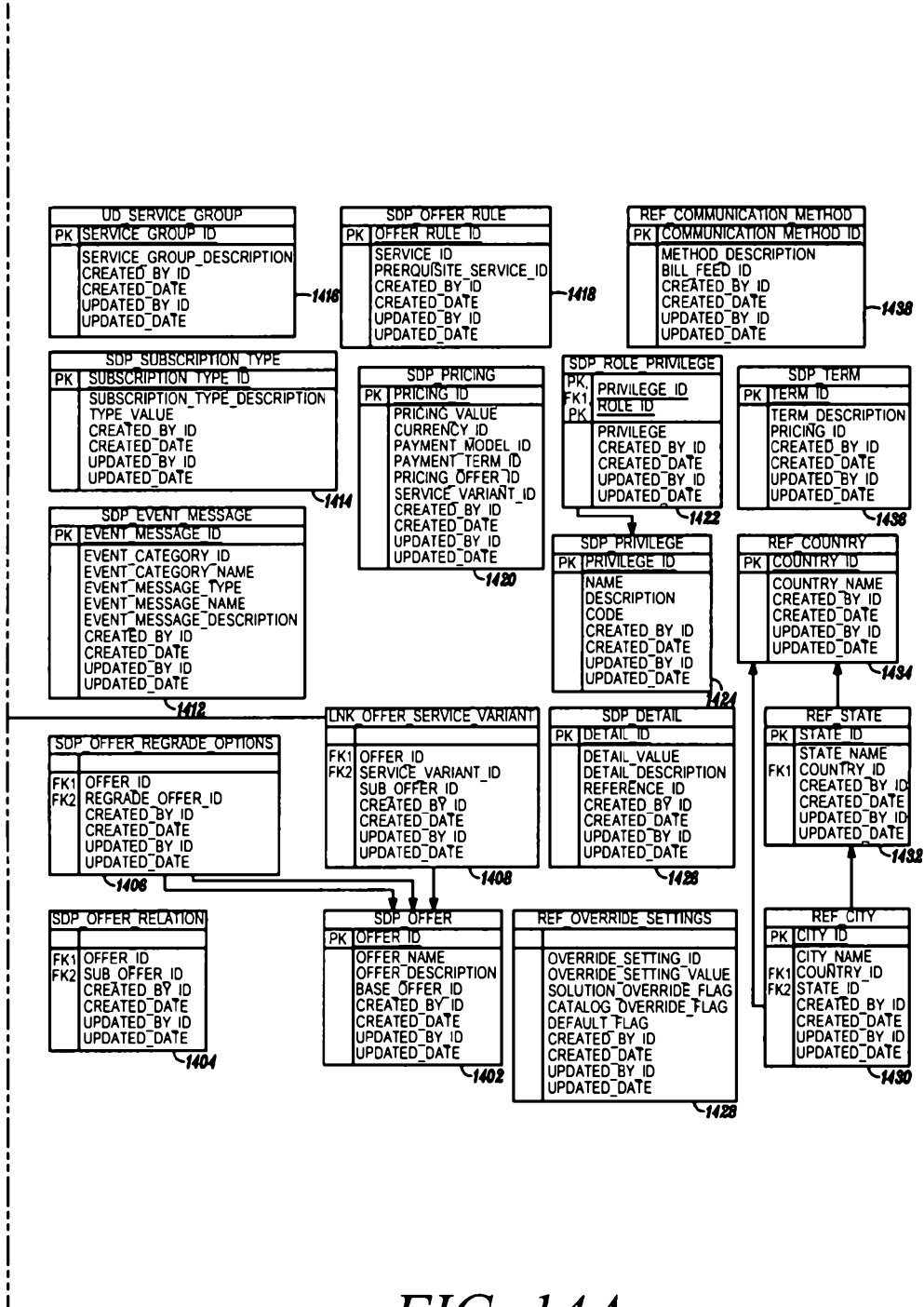


FIG. 14A

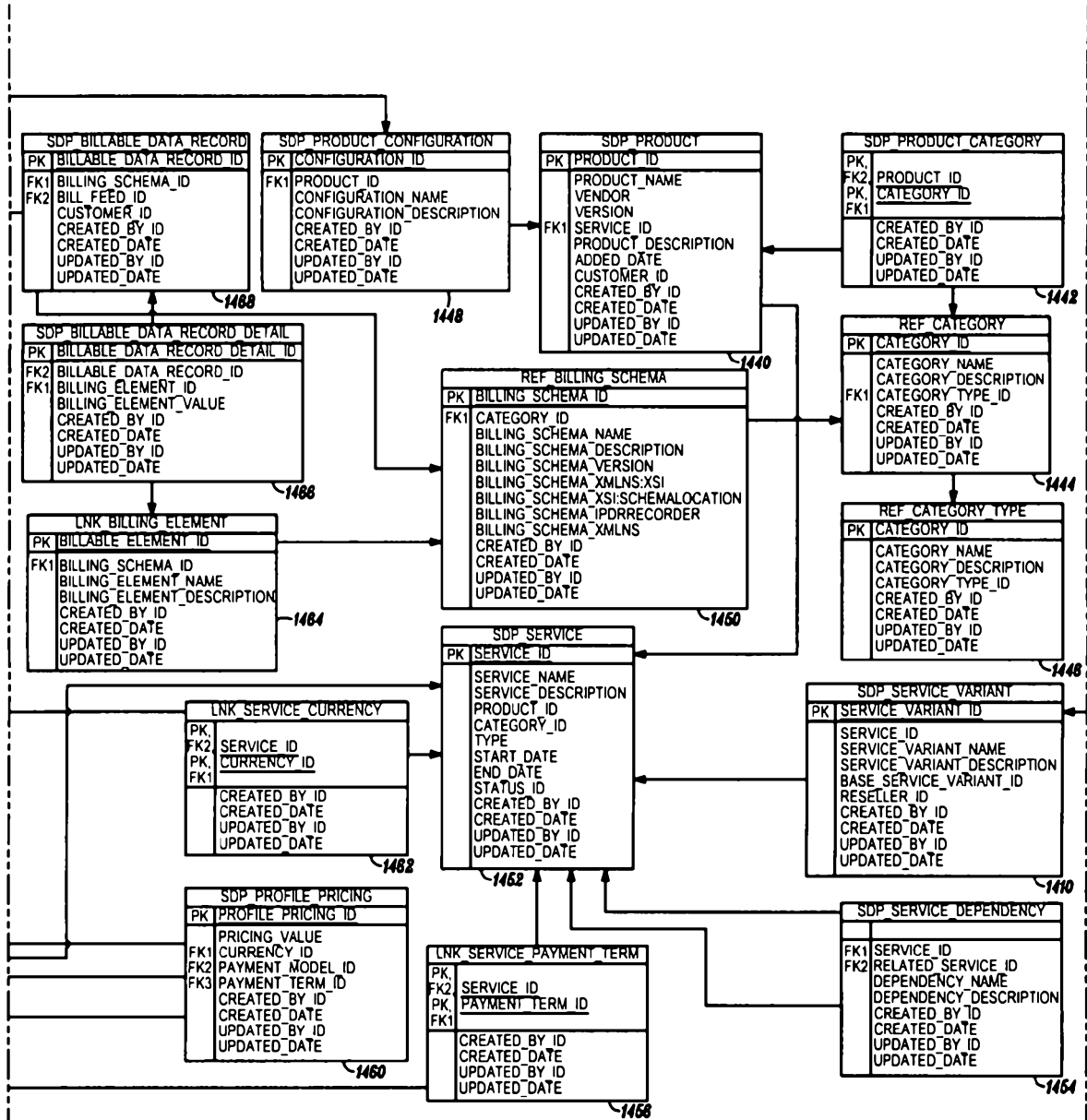


FIG. 14B

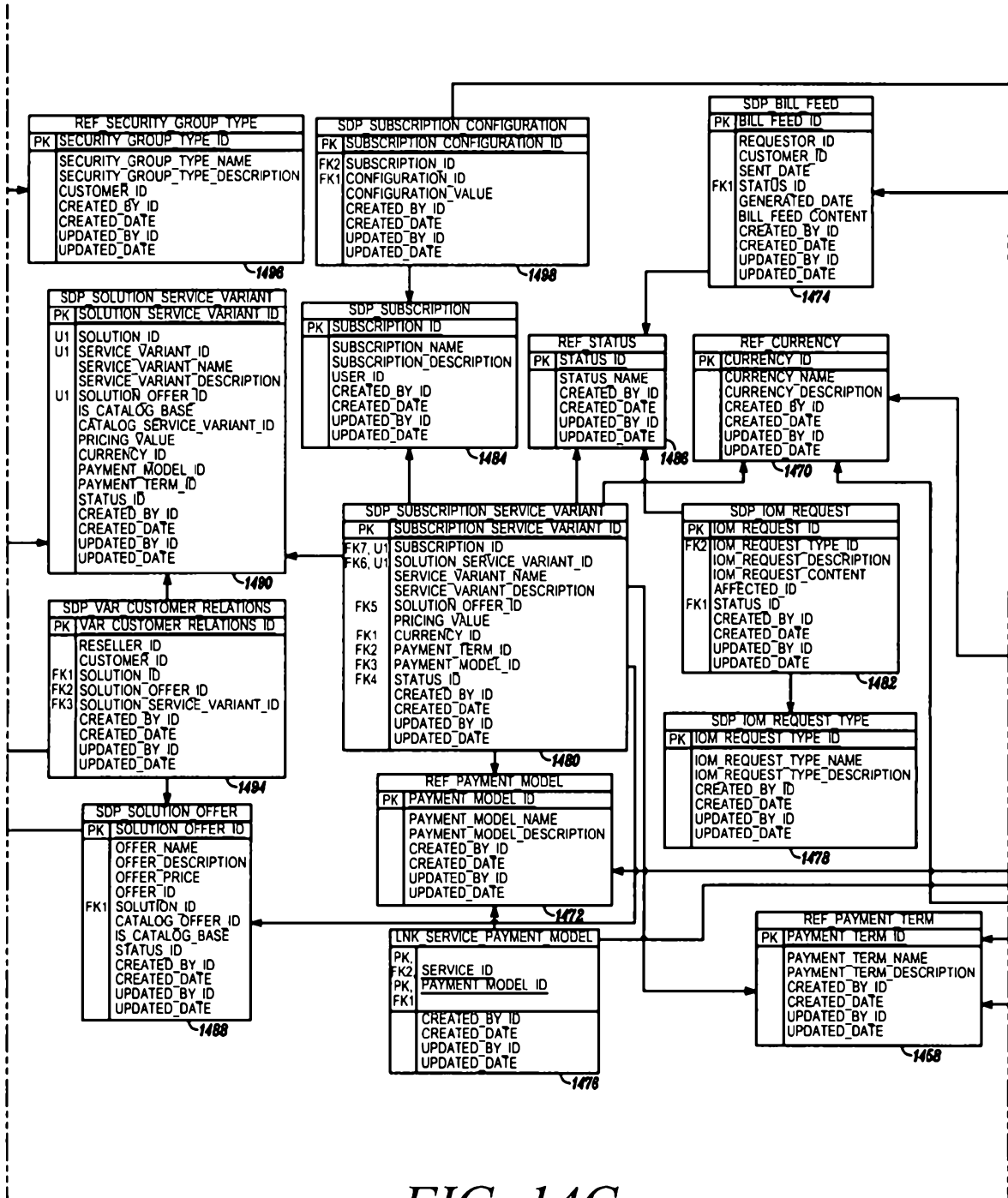


FIG. 14C

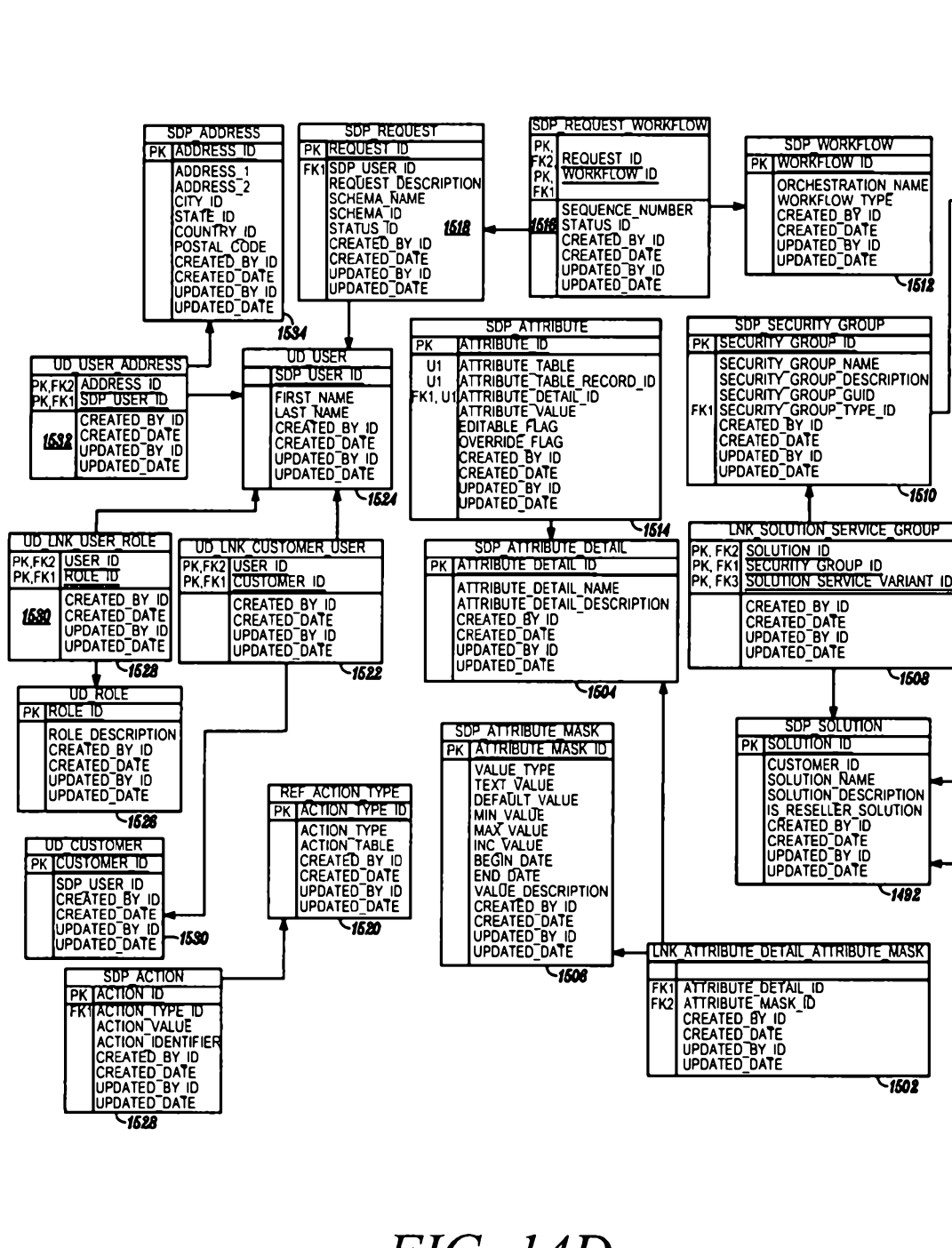


FIG. 14D