



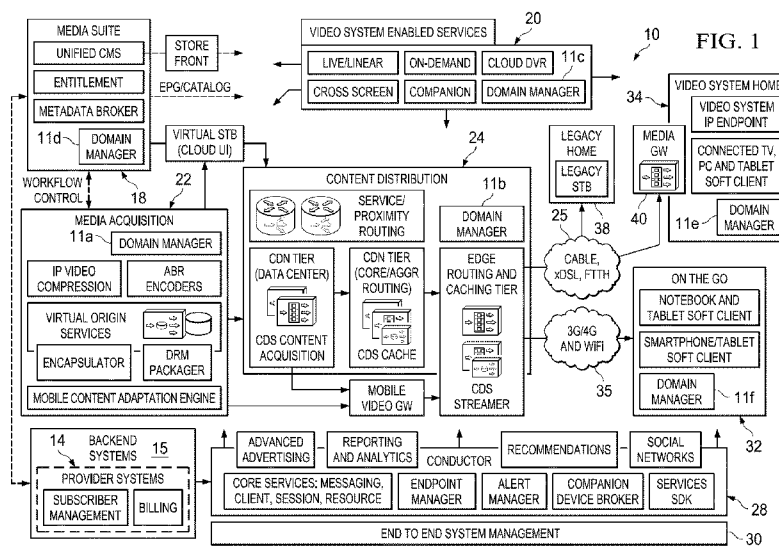
(43) International Publication Date
10 January 2013 (10.01.2013)

- (51) International Patent Classification:
H04L 29/08 (2006.01) H04L 29/12 (2006.01)
- (21) International Application Number:
PCT/US2012/045858
- (22) International Filing Date:
6 July 2012 (06.07.2012)
- (25) Filing Language:
English
- (26) Publication Language:
English
- (30) Priority Data:
61/505,358 7 July 2011 (07.07.2011) US
13/543,692 6 July 2012 (06.07.2012) US
- (71) Applicant (for all designated States except US): CISCO TECHNOLOGY, INC. [US/US]; SJC/10/2/1, 170 West Tasman Drive, San Jose, California 95134-1706 (US).

- (72) Inventors; and
- (75) Inventors/Applicants (for US only): WANG, Qi [CN/CN]; Room 904, Tian Hua Building, No. 276, Fu Rong Jiang Road, Shanghai, 200233 (CN). LI, Jerry Li-ansuo [CN/CN]; Room 602, Suite 286, Block 99, Wanding Road, Shanghai, 201104 (CN). YOU, Guangxin [CN/CN]; Room 602, Suite 8, Baiyang Road 1065#, District Pudong, Shanghai, 210124 (CN). SHE, Zhidong [CN/CN]; Room 701, No. 5, Nong 618, Road North Hut-ing, District Songjiang, Shanghai, 201615 (CN). POPE, Nick George [US/US]; 5123 Brendlynn Dr., Suwanee, GA 30024 (US). ANDREASEN, Flemming S. [DK/US]; 32 Burlington Drive, Marlboro, NJ 07746 (US).
- (74) Agent: FRAME, Thomas J.; Patent Capital Group, 2816 Lago Vista Lane, Rockwall, TX 75032 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ,

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR PROVIDING A MESSAGE AND AN EVENT BASED VIDEO SERVICES CONTROL PLANE



(57) Abstract: A method is provided in one example embodiment and includes establishing a connection between a client and a messaging fabric of a conductor element associated with a video system; defining a service having a set of features using a set of interfaces associated with an Extensible Messaging and Presence Protocol (XMPP); assigning a plurality of XML namespaces for the set of features of the service; assigning an identifier to the service; and registering the service in a service directory in order to create a mapping between the XML namespaces and the identifier.

WO 2013/006844 A1

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

SYSTEM AND METHOD FOR PROVIDING A MESSAGE AND
AN EVENT BASED VIDEO SERVICES CONTROL PLANE

TECHNICAL FIELD

[0001] This disclosure relates in general to the field of communications and, more particularly, to a system and a method for providing a message and an event based video services control plane.

BACKGROUND

[0002] Service providers face difficult challenges in the context of providing video services for a diverse group of end-users. Many service providers are gearing up to implement their 'TV Everywhere' initiatives, which can offer a level of freedom being demanded by consumers today. One aspect of this demand includes the ability to access content from any device at any time and from any location. Providing an effective integration of various technologies, while accounting for specific device options, specific location possibilities, specific user preferences, specific content and programming, etc. is a significant challenge for service providers.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] To provide a more complete understanding of the present disclosure and features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying figures, wherein like reference numerals represent like parts, in which:

[0004] FIGURE 1 is a simplified block diagram of a video system for providing a video platform in accordance with one embodiment of the present disclosure;

[0005] FIGURE 2 is a simplified block diagram illustrating possible example details associated with one embodiment of the video system;

[0006] FIGURE 3 is a simplified block diagram illustrating possible example details associated with one embodiment of the video system;

[0007] FIGURE 4 is a simplified block diagram illustrating possible example details associated with one embodiment of the video system;

[0008] FIGURE 5 is a simplified block diagram illustrating possible example details associated with one embodiment of the video system;

[0009] FIGURE 6 is a simplified block diagram illustrating possible example details associated with one embodiment of the video system;

[0010] FIGURE 7 is a simplified block diagram illustrating possible example details associated with one embodiment of a service policy;

[0011] FIGURE 8 is a simplified block diagram illustrating possible example details associated with one embodiment of virtual service and service instances of the video system; and

[0012] FIGURES 9-11 are simplified flowcharts illustrating potential operations associated with the video system in accordance with one embodiment of the present disclosure.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

OVERVIEW

[0013] A method is provided in one example embodiment and includes establishing a connection (e.g., wired, wireless, etc.) between a client and a messaging fabric of a conductor element associated with a video system. The method also includes defining a service having a set of features using a set of interfaces (e.g., hardware, software, applications, etc.) associated with an Extensible Messaging and Presence Protocol (XMPP). The service can be associated with any type of management, processing, delivery, formatting, or controlling of video data. The features can be associated with any type of activity, operation, task, or function associated with video. The method also includes assigning a plurality of XML namespaces (which includes any type of identifier) for the set of features of the service; assigning an identifier to the service; and registering the service in a service directory in order to create a mapping between the XML namespaces and the identifier.

[0014] In more particular embodiments, the service exposes non-XMPP-based features to be accessed by other protocols. In addition, the method could include

registering non-XMPP-based features in the service directory using a uniform resource locator (URL). In more specific implementations, the service is invoked by a device in order to support a particular feature of the video system.

[0015] In other implementations, the method can include accessing a protocol namespace in the service directory; retrieving a particular identifier associated with a particular service; and communicating a message using a messaging infrastructure of the conductor element. Additionally, a virtual service identifier can be defined for a particular service. In certain example instances, the method can include receiving a message associated with a particular service, and evaluating a service policy in the service directory to determine permissions associated with messaging for the particular service. The service policy can include matching criteria used to determine whether a particular client is permitted to send messages to the particular service. More specific methodologies can include evaluating which groups are associated with a particular client that sent the message, the service policy can include group permissions associated with a plurality of clients.

EXAMPLE EMBODIMENTS

[0016] Turning to FIGURE 1, FIGURE 1 is a simplified block diagram of a video system 10 configured for providing an integrated video platform in accordance with one embodiment of the present disclosure. Video system 10 may include a plurality of backend systems 15, which may further include a number of provider systems 14 that are inclusive of subscriber management and billing. In addition, video system 10 may include a media suite 18 for content and metadata management, which may be coupled to a media acquisition 22 for content processing. A video system enabled services element 20 may be suitably linked to media suite 18, media acquisition 22, and a content distribution 24.

[0017] Additionally, any number of networks may suitably couple content distribution 24 to a video system home 34, as well as an "on the go" component 32, which may be associated with wireless activities, roaming, WiFi, end-user devices more generally, etc. In one particular example being illustrated in FIGURE 1, a 3G/4G and WiFi network 35, along with a cable, xDSL, FTTH network 25 are being used to facilitate the activities of the video platform. FIGURE 1 also includes a conductor 28 video control plane, which can be

suitably coupled to media acquisition 22, content distribution 24, and an end to end system management 30. Note that the larger blocks of FIGURE 1 (e.g., conductor 28, content distribution 24, media suite 18, video system enabled services 20, vide system home 34, media acquisition, 22, etc.) can be viewed as logical suites that can perform certain activities of the present disclosure. Note that their functions, responsibilities, tasks, capabilities, etc. can be distributed in any suitable manner, which may be based on particular video needs, subscription models, service provider arrangements, etc.

[0018] In accordance with the teachings of the present disclosure, video system 10 is configured to offer service providers a number of valuable features. For example, video system 10 is configured to extend video services to a variety of devices ranging from smartphones, tablets, iPads, personal computers (PCs), to set-top boxes (e.g., n-screen), cable systems, etc. Additionally, this platform of video system 10 is configured to extend video services to any IP access network (un-tethering). The architecture can also provide unified content management between different devices, different networks, and different video services. Additionally, the architecture can provide a flexible platform and infrastructure that enables existing services to be modified (and for new services to be developed by the service provider) by leveraging a combination of Internet protocol (IP), hypertext transfer protocol (HTTP)/web-services, Extensible Messaging and Presence Protocol (XMPP) and a workflow-enabled infrastructure with open interfaces and both client and server software development kits (SDKs). An initial set of applications can also be provided (e.g., linear, time-shift, on-demand, etc.).

[0019] Additionally, the architecture can use adaptive bitrate (ABR) to facilitate video service delivery (independent of the access). This allows a video offering that can be targeted at consumers, which can offer "Anywhere, Any Access" that may be tied to subscription models. In addition, video system 10 can readily support unicast and multicast delivery with in-home cache optimizations for more efficient use of access network resources. This can include support for content protection, thereby enabling delivery of all content (not merely a subset of content). This also includes support for existing critical features such as Emergency Alert Service, Blackouts, Geo-Blocking, etc. Support is also provided for advertising (including dynamic ad support) and for legacy devices (primarily

existing endpoint devices (e.g., set-top boxes (STBs)) for a smooth migration of existing infrastructure.

[0020] The architecture can also support hybrid optimizations for access providers to implement (e.g., in order to enhance their offering). In this context, hybrid is referring to the combination of traditional service provider video delivery technologies (e.g., MPEG transport stream over quadrature amplitude modulation (QAM) in a cable hybrid fiber-coaxial (HFC) environment) with pure IP video delivery technologies (e.g., HTTP-based adaptive bitrate).

[0021] In operation, communication system 10 can support the following end-user oriented use cases: 1) content discovery; 2) linear services for managed IP STBs and unmanaged devices (where migration for existing linear services is supported equally); 3) on-demand services for managed IP STBs and unmanaged devices (where migration for existing on-demand services is supported); 4) time-shifted TV services (e.g., in the form of Cloud DVR/time-shifted TV across screens for managed IP STBs and unmanaged devices (where migration for existing DVR services is supported); 5) cross-screen experience in the form of companion devices, where a companion device (e.g., iPhone) can be used as a remote control for another video system device (e.g., IP STB), or the companion device can enhance the viewing experience through value add/context or programming aware metadata information (e.g., Facebook/twitter feeds, additional program detail, hyperlinks, etc.); 6) screen-shifting, where the user is able to change playback to another device (e.g., from iPad to TV), pause and resume programs across devices, or have multi-room DVRs; 7) dynamic advertising; and 8) value add applications, which enable service providers to offer value add user experiences (e.g., such as Facebook connect capabilities, access to Olympics Applications, etc.).

[0022] Note that video services have traditionally been provided in a siloed fashion. Linear TV services were provided by Cable, Telco, or Satellite companies over legacy non-IP based infrastructures with service offerings that expanded to include time-shift, on-demand, and DVR type services. Services were offered to managed devices (e.g., a STB) on managed networks only (e.g., QAM-based cable). As IP infrastructure with relatively high bandwidth became more prevalent, a second wave of IPTV-based video systems appeared. A common theme in these systems is an IP multicast-based linear service, real-time streaming protocol

(RTSP)-based on-demand (etc.) service, and a session initiation protocol (SIP)/ IP multimedia subsystem (IMS) plus RSTP control plane, and/or an HTTP/web services plus RTSP based control plane coupled with metadata management (e.g., electronic program guide (EPG)) towards the end-users typically based on HTTP/web services. IPTV content delivery was generally assumed to be a fixed bitrate over managed networks (either supporting resource reservations to satisfy certain levels of service or simply having plentiful bandwidth).

[0023] A new 3rd wave of systems is now being considered with a design principle of any content to any device anywhere at any time. HTTP adaptive bitrate enables this model in the content delivery domain; however, for a service provider to provide premium video services, a control plane infrastructure is still needed. The existing IPTV based control plane architecture and solutions fall short in a number of areas needed to support the above 3rd wave systems in today's web-based environment, including: 1) a lack of consideration and service for HTTP ABR based content delivery, which does not have the notion of a "network" or cloud session (e.g., for troubleshooting, diagnostics, statistics, policy enforcement (upper limit on sessions)), etc.; and 2) the HTTP Simple Object Access Protocol/REpresentational State Transfer (REST)(SOAP/REST) based video control plane architectures fall short in several areas. This includes an inability to work through NATs (e.g., to support notification type services to clients (emergency alerts, operator initiated messaging/diagnostics, etc.)). This also includes bidirectional communication support and a way for cloud-initiated communication to target households, users, and/or specific devices are missing (e.g., eventing), and authentication/authorization considerations around such cloud-initiated communication is missing as well. In addition, such models work as request-response protocols in the client-server computing model, and they are generally not session-stateful, which is needed for some premium video services. These HTTP-based services do not retain information or status of each user for the duration of multiple requests. Therefore, when HTTP-based web services are deployed over a large cluster, it is difficult to track the user's progress from one request to another, unless a centralized database is used to track it.

[0024] The SIP/IMS-based video control planes provide persistent connections with bi-directional support and notification services, which solve several of the shortcomings of the HTTP-based control planes. However, the SIP/IMS based architectures fall short in several other areas as well (e.g., they are defined only for SIP/IMS-based services to be

invoked and advertised). In today's world, ease of integration with HTTP and XML-based services is important. Additionally, SIP/IMS is based on a call setup model, whereby services are invoked as part of an incoming or outgoing session setup. Events within or outside of a session are supported as well. As a result of this, IMS service creation, composition, and interaction relies on the notion of IMS filter criteria, which are (statically defined) trigger points used to determine which of several IMS application servers (AS) to invoke.

[0025] Interaction between multiple application servers is handled by the (under-specified) Service Capability Interaction manager (SCIM) function. It is in many ways a more modern version of the classic Intelligent Network (IN) model used for telephony systems in the past. In the 3rd wave video system and today's increasingly web-based technology world, users and services both need to be considered as first-class citizens that are equally capable of initiating service to each other. Furthermore, an open framework of orchestrating such services is important, including responses to events in the system.

[0026] With SIP/IMS being designed around the need to establish a communication session (e.g., a call), it is not well suited to exchange structured data as part of a session by itself. For example, support for large messages is an issue over user datagram protocol (UDP), and SIP proxies are in general not intended to have frequent or substantial amounts of data sent through them. However, several video control plane services need that capability (e.g., remote scheduling, companion device experiences, interactive diagnostics, etc.).

[0027] Certain embodiments of video system 10 can offer an overall video services control plane architecture that addresses the above shortcomings. In accordance with one example implementation of the present disclosure, video system 10 can resolve the aforementioned issues (and potentially others) to provide a combination of cloud, network, and client capabilities that enables the service provider to offer its subscribers any content over any network to any device. The present disclosure provides the first complete instantiation of an end-to-end video platform solution supporting the full complement of managed video service offerings.

[0028] Within the platform of FIGURE 1, the functional components are logically grouped into different suites. Extending beyond the core platform are components that are assumed to be preexisting, within either the service provider or the content provider

networks. Specifically, service provider Business Support Systems/Operations Support Systems (SP BSS/OSS) represents a set of preexisting business and operations support systems. 3rd party web services are cloud-based services that the solution leverages, but are preexisting and can be leveraged in-place. Content provider control systems are preexisting or future systems that support the delivery of content into secondary distribution channels. A collection of different networks (both service provider managed networks and other networks) can also be provided that play a role in the delivery of the video service. Finally, the architecture can also include existing on-demand and linear content sources, representing both the origination of that content from the content provider/broadcaster, as well as the acquisition of that content within the service provider's network. The solid and dashed lines in this area represent the distinction between content metadata and content essence (the actual media files, etc.).

[0029] The cloud paradigm can extend the media and acquisition suites with enhanced capabilities for linear and time-shifted TV. The communication platform also introduces conductor and conductor services, providing an extensible service creation environment, common service capabilities, as well as massively scalable and persistent client connection technologies. Three additional suites are also provided, which includes the ad suite (represented as 'Advanced Advertising' in FIGURE 1) that provides a core set of advanced advertising capabilities that integrates a web ad decision server capabilities. In addition, an application suite (e.g., Video System Enabled Services) is provided that builds on the base soft client capability provided in QuickStart. It also provides a base set of core and value-add end-user applications across both managed and unmanaged devices. A management suite (e.g., end to end system management) is also provided for client and endpoint management; it facilitates management of the overall video platform suite of products.

[0030] Video system 10 also builds on the distribution suite capabilities for the efficient delivery of both on-demand and linear content to client devices. The content delivery network (CDN) capability can be responsible for taking content that originates from the Content management/media processing functions, and delivering it to clients at scale, efficiently, and with minimal end-to-end latency. The CDN can provide a high degree of deployment flexibility: scaling from more centralized deployments to highly-distributed

deployments using centralized root caching tiers, multiple intermediate caching tiers, and edge-caching tiers close to the client devices. CDN also provides intelligent content routing capabilities that are tied, through network proximity, to the real-time routing details of the underlying network elements. This enables the service provider to efficiently deliver content from the best edge cache resource, even during periods of network impairment.

[0031] The architecture also covers soft clients as well as managed devices. Specifically, the architecture includes a video system home gateway, as well as a video system IP STB. The home gateway, as an extension of the network, provides valuable linkage between managed and unmanaged devices within the home and the service provider cloud and network infrastructures. The IP STB, as well as all soft clients running on unmanaged devices, is designed to work across managed and unmanaged network environments. Soft client capabilities can be extended to include linear and time-shift capabilities, as well as leverage the evolving set of cloud and network APIs exposed by the various suites to provide a high-quality end-to-end user experience.

[0032] Video system 10 presents a migration to an all-IP based video and services infrastructure spanning the full service/content life cycle, from the video content and metadata acquisition, to content and metadata preparation, distribution, and delivery to the end-user. The video system encompasses a set of diverse products/suites with heterogeneous interfaces and implementations for these functions. The overall system follows a Service Oriented Architecture (SOA) development framework and, hence, supports multiple individual services, which are used via service orchestration and workflow engines. Each of the suites provides a set of well-defined services and associated interfaces, and it is with these services that end-user services are eventually provided. End-user services can be defined as including one or more services that users interact with to provide a user visible service. For example, a linear TV service provides features and logic to enable users to watch a particular channel in accordance with their subscription. The linear TV service does so by use of a number of underlying video system services and suites. Application suite services play a particular role in terms of providing certain application logic for one or more services. Users could be machines as well (e.g., for machine-to-machine oriented type services).

[0033] In certain implementations of the present disclosure, video system 10 can leverage a set of HTTP-based RESTful web services to support basic on-demand TV everywhere capabilities. These HTTP services, exposed to end-points by both the media suite and the distribution suite, can provide proven scalability, resiliency, and extensibility. In operation, the video platform can use a mix of HTTP RESTful web services and XMPP-based services, providing a powerful combination to support the enhanced capabilities for linear, time-shift TV, VOD, companion, and value-add applications.

[0034] Turning to FIGURE 2, FIGURE 2 illustrates a number of example content sources 45 (e.g., YouTube, Starz, HULU, etc.). Devices and services can be divided into client-facing and cloud-facing components. Client-facing components and services can involve interaction with a client. Cloud-facing components and services can include everything else. In either case, services provide well-defined XMPP and/or HTTP-based interfaces. XMPP-based services can rely on the conductor infrastructure and the features it provides (e.g., service virtualization or persistent connections), whereas HTTP-based services in the video system can follow a standard web-services model.

[0035] Clients may interface directly with a service or they may interact with a front-end application/service, which in turns orchestrates and invokes other services (e.g., by use of the flexible workflow engine provided by service orchestration). Similarly, services may also rely on backend application logic to implement higher-level applications/services, which again may rely on service orchestration of other services. On the client itself, there may be one or more applications installed, and applications may contain add-on modules. In either case, the client-side application interacts with the video system cloud via one or more service invocations (e.g., "Create Recording" to schedule an nDVR recording, which is supported by a service or application front-end via HTTP or XMPP).

[0036] In operation, the media suite (unified CMS, entitlement, metadata broker, LSMS/EPG manager, etc.), the distribution suite (which is the content distribution that includes the service router, service engine/edge cache, etc.), the advertising suite, and the application suite can expose services that clients consume. The client-facing interfaces can be HTTP-based, and for the video system, they can continue to be HTTP-based, or they as well as other applications and services may be HTTP and/or XMPP based. In either case, efficient mechanisms can be used for clients to initially discover these services, select the

instance of the component that can best fulfill service requests from that client, and manage the allocation of finite resources across all instances of that service. The video system can offer a unified service discovery capability through the conductor's service directory for both XMPP and HTTP-based services. For XMPP-based conductor services, service virtualization can be provided natively by the conductor infrastructure.

[0037] FIGURE 3 is a simplified block diagram highlighting the video system enabled services, along with the conductor capabilities. The acquisition suite services, while not directly consumed by client endpoints, provide critical media processing services to the media suite and the distribution suite and, therefore, are also considered. Service routing and service virtualization for the media suite, the acquisition suite, and the distribution suite can continue to leverage existing implementations. Specifically, the media suite currently provides a global server loadbalancing (GSLB)/Apache web services mechanism for service virtualization and loadbalancing. The acquisition suite can provide loadbalancing for video on demand (VOD) transcoding through its transcode manager server; expanded mechanisms for service virtualization and loadbalancing for linear and VOD transcoding and encapsulation can also be provided in the video system. The distribution suite provides a service router based mechanism for virtualization and edge cache selection. The ad suite message exchanges are stateless with transaction data being maintained and replicated across the virtualized service cluster allowing any virtual endpoint to process a message exchange. For services accessed using traditional HTTP message exchanges, an appliance, or other hardware loadbalancer may be used. Alternatively, a loadbalancer or a software loadbalancer may be adopted in alignment with the overall video system architecture. When the ad suite is accessed using XMPP, the integrated video system conductor service virtualization is leveraged for loadbalancing and high availability.

[0038] Video system users can subscribe to the video services through their service provider. One or more users and devices may be associated with an account for service, and associated with each is a profile to enable personalization of the video services. Devices range from IP set-top boxes to soft clients on a variety of devices such as PCs, Macs, tablets, smartphones, etc., and all of those devices may be used either on the service provider's access network (home), or another network (e.g., on the go). Users may also have a video system home gateway, which could be a residential NAT/firewall type device with additional

video features, such as media caching, and multicast-to-unicast conversion to optimize the end-user video experience and to reduce use of access network resources (especially when users have multiple devices accessing the same content). Cable and Telco (xDSL, Fiber, etc.) access networks are supported as managed networks, where quality of service and policy control enable a better end-user video experience than for unmanaged access network, that provide an over-the-top experience instead.

[0039] Users and devices can connect to the video system infrastructure using primarily persistent XMPP connections and stateless HTTP-based web services. The conductor provides the XMPP infrastructure to which clients (users/devices) connect via the connection manager and have their identity authenticated, thereby enabling a secure and personalized service experience. The conductor provides a basic set of connection management, messaging and core services, and additional services enablement features to allow for new services to be introduced. Services and applications can connect to the conductor, thereby enabling them to use the core services provided by the conductor, as well as exchange messages with each other through the XMPP messaging infrastructure.

[0040] Core services provided by the conductor include the client directory, which contains user and device profile information, and the publish-subscribe subsystem (PubSub), which enables listeners to subscribe to and be notified about events generated by publishers for a given topic. The session state manager tracks state associated with sessions (e.g., a video session when watching a movie), and the resource broker allows resources (e.g., network bandwidth), to be associated with that session. The application suite provides a set of supporting front-end and backend application logic to deliver the linear and time-shift TV, nDVR, on-demand, soft client download for certain platforms, value-added applications, and a web portal e-commerce platform for the on-demand storefront.

[0041] FIGURE 4 is a simplified block diagram illustrating the video systems cloud APIs and clients. In this particular example, a video system cloud API 50 is provided as being connected to a RESTful HTTP web services network 56. In addition, other instances of a video system cloud API 52, 54 are coupled to an XMPP messaging cloud 58. An instance of third-party services 60 is also being illustrated and is coupled to a video system managed IP set-top box 62. Additionally, a video system iOS tablet 64 and a video system Android smartphone 66 are suitably connected to a given network. The cloud APIs can enable a

consistent user experience. Additionally, the cloud APIs can leverage the best of XMPP and HTTP. The client SDKs can facilitate cloud API use across diverse platforms. Additionally, the cloud APIs can access third-party services.

[0042] FIGURE 5 is a simplified block diagram illustrating the content distribution suite and the media acquisition suite. In certain example implementations, the program guide retrieval and media delivery is HTTP-based. Video delivery supports adaptive bitrate, and it can utilize the distribution suite for efficient, service provider-scale video delivery. The distribution suite provides for distributed content caching throughout the network. HTTP requests for content can be sent to the service router (SR) first, which uses the proximity engine (PxE) to perform a proximity-based redirection of the HTTP request to a service engine (SE) for efficient media delivery. When the service engine receives the request, it either serves it from its cache, another service engine (higher in the caching hierarchy), or it contacts the content acquisition function, which retrieves the content from an origin server (in the acquisition suite). The distribution suite can be used for efficient delivery of any cacheable application object such as generic program guides, whereas personalized program guides may be retrieved directly from the media suite instead. In either case, clients may learn about new program guides being available by use of the PubSub XMPP service for program guide updates.

[0043] FIGURE 6 is a simplified block diagram illustrating additional details associated with the media suite, provider systems, etc. The media suite component receives content metadata and electronic program guide (EPG) information from a multitude of content providers that are serving up managed and unmanaged content. The media suite normalizes this information and produces program guides for the associated content. This can involve using the LSMS/EPG manager for mapping content to channels, respecting blackout indications for content in certain regions, determining Digital Rights Management (DRM) to be applied, etc. The program guides typically vary by region based on locally available content, and program guides may vary on a per-user basis as well (personalized program guides). Similar functionality is provided for on-demand content, which can be made available and visible to end-users. Once the associated content is available, the media suite can then publish the program guide and catalog information for that content. The media suite additionally supports a variety of time-shift TV experiences,

bridging the linear and on-demand domains; the DVR CMS function can provide content management functions in this regard. The media suite provides a unified entitlement capability, enabling the service provider to provide support for multiple leading DRM ecosystems. Individual assets (on-demand, linear channels, applications), both managed and unmanaged, are combined into offers by the media suite publisher capability. For example, the service provider may choose to provide a unified VOD catalog that contains a mix of actively managed content as well as unmanaged content from aggregators such as Hulu.

[0044] Metadata associated with this content can be served by the metadata broker, which also serves metadata associated with program guides and nDVR recordings. Managed content can be acquired, transcoded, encrypted, and delivered by the service provider's infrastructure (acquisition suite), whereas the unmanaged content processing and delivery is the responsibility of the aggregator. Assets from both can be seamlessly merged into unified offers and presented to the user in a common catalog. In the case of managed content, the client can interact with the media suite entitlement management server. If the user is entitled to the content, the content resolution server (CRS) function decides on one or more suitable formats to serve up the content for the client in question; the formats may in turn depend on certain content policies controlled by the content policy function. In the case of unmanaged content, the client will interface directly to the aggregator's backend entitlement/delivery systems at the time of asset playback.

[0045] Before a user is permitted to watch certain content, whether it is linear or on-demand, the content can be made available. Unmanaged content is neither cached nor processed by the video system network, but is instead delivered over-the-top (OTT) as any other IP traffic. However, managed content can be acquired from the content provider, and possibly transformed in a multitude of ways. The acquisition suite serves this role by (re)encoding the content in possibly several different formats (codecs, resolutions, etc.) to support a multitude of end-user devices and the adaptive bitrate delivery of said content. VOD transcoding is done by a transcode manager, linear transcoding can be done by the digital content manager (DCM) and media processor, and ABR formatting can be handled by the media encapsulator. Encryption for DRM can also be provided. The acquisition suite and media suite coordinate with each other to determine what content to acquire, when

the content is available and, hence, can be published in a catalogue, and which DRM to apply. Once the content has been transformed as appropriate, it can be stored on the origin server function, and the content is then available for distribution to endpoints. The content can then either be pushed out to the distribution suite (pre-fetching), or the distribution suite will retrieve and cache it when needed.

[0046] In spite of the use of HTTP ABR, some content may be served by multicast; the home gateway can translate between multicast delivery and unicast HTTP ABR to optimize access network and CDN (distribution suite) use. The multicast manager advertises statically and potentially dynamically provisioned multicast sessions defining the multicast cloud that determines the multicast senders, as well as the coverage for that multicast tree. The virtual origin service (VOS) embeds capabilities such as encapsulation, time-shifted representations, recording for nDVR, and multicast origination for multicast-cache fill; the service router function enables efficient service routing request handling across multiple VOS instances (e.g., to use a topologically close-by VOS).

[0047] Based on the program guide information, VOD catalog, etc., the client can have an HTTP URL for the content it wishes to acquire (e.g., a TV channel, a movie on-demand, etc.). When the client issues a request for said content, it will need to go through an entitlement check to determine if it's allowed to obtain the content requested. The entitlement check is performed by the media suite, which interfaces to the DRM/license servers to obtain DRM ecosystem-specific license keys that enable decryption of the DRM protected content.

[0048] The ad suite placement broker accepts advertising placement queries (e.g., in the form of an Society of Cable Telecommunications Engineers (SCTE) 130 Part 3 PlacementRequest message), from any initiating source (be it a client or the cloud). The placement broker gathers additional targeting criteria relative to both the content and the viewer from a combination of internal and external sources. For content specific metadata, the media suite's metadata broker and/or a 3rd party metadata source are queried using the SCTE 130 Content Information Service (CIS) interface. User or content viewer information is obtained from a combination of internal and/or 3rd party sources using the SCTE 130 Subscriber Information Service (SIS) interface. Example SIS metadata sources

include video system's geolocation service, conductor's client directory service, indirect access to the service providers subscriber data, or an external 3rd party such as Experian.

[0049] One or more placement opportunities (a more generalized form of a traditional linear element that includes metadata describing decision ownership, policy, ad unit structure) can be obtained from a component implementing the SCTE 130 Placement Opportunity Information Service (POIS) interface. Based on ownership and provisioned placement service criteria, the placement broker applies the appropriate metadata visibility policies and routes the individual placement opportunities to the correct advertising decision service. The advertising decision service may be a component of a 3rd party campaign manager or it may be the ad suite's web ADS router. The web ADS router forwards decision requests to a 3rd party web ad decision server such as DoubleClick or Freewheel using their native request format and receives an Interactive Advertising Bureau (IAB) Video Ad Serving Template (VAST) 2.0 response. The placement broker aggregates the sum of advertising placement decisions and returns the result to the initiating source using a SCTE 130 PlacementResponse message. The initiating source then intermixes the entertainment content and the selected advertising assets using the appropriate delivery platform specific assembly mechanism (for example, manifest manipulation for HLS, or player control for client HSS/Smooth, etc.).

[0050] The placement reporter acquires media session events including placement, playout, session, viewer, and remote control events, filters these events according to the provisioned placement service policies, and forwards the appropriate confirmation reports to the individual advertising decision services. The web ADS router provides an additional forwarding capability proxying to the VPAID format. The placement reporter also archives the data for later analysis and provides report generation support.

[0051] The management suite fulfills the management aspects (FCAPS) of the video system. The device manager performs basic hardware and firmware device management for video system managed devices (i.e., set-top boxes and home gateways, whereas the endpoint manager supports overall management for all video system clients in the form of application download, provisioning, event collection and reporting, etc.). Domain managers are sub-system managers for each product suite. A domain manager is either located in the management suite itself or it is a product in another suite that fulfills a dual role. Finally, the

video system manager of managers (MoM) can offer an overall manager for the various management components of the platform.

[0052] The video system architecture defines several third-party elements that are not associated with any particular suite. Within the video system box, the Authentication/Authorization/Single-Sign-On (AA/SSO) function provides a common backend AA and SSO solution that allows for common credentials and single sign-on between different suites and interfaces. The accounting function enables storage of accounting data (e.g., for quality statistics), and the DOCSIS and Telco Policy functions provide policy server functions for Cable and Telco access networks. Outside the video system box, a number of third-party elements for 3rd Party web services, service provider BSS/OSS, Content Provider (CP) Control Systems, as well as EPG schedule information, VOD and Linear Content Sources, Integrated Receiver Decoders (IRD), Emergency Alert System (EAS), and Public CDNs are defined as well.

[0053] Turning to the example infrastructure associated with present disclosure, the clients of FIGURE 1 can be associated with devices, customers, or end-users wishing to receive data or content in video system 10 via some network. The term 'client' is inclusive of devices used to initiate a communication, such as a receiver, a computer, a set-top box, an IRD, a cell phone, a smartphone, a tablet, a remote control, a personal digital assistant (PDA), a Google droid, an iPhone, an iPad, or any other device, component, element, or object capable of initiating voice, audio, video, media, or data exchanges within video system 10. The clients may also be inclusive of a suitable interface to the human user, such as a display, a keyboard, a touchpad, or other terminal equipment. The clients may also be any device that seeks to initiate a communication on behalf of another entity or element, such as a program, a database, or any other component, device, element, or object capable of initiating an exchange within video system 10. Data, as used herein in this document, refers to any type of numeric, voice, video, media, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another.

[0054] The networks of FIGURE 1 can represent a series of points or nodes of interconnected communication paths for receiving and transmitting packets of information that propagate through video system 10. The networks can offer a communicative interface

between sources and/or hosts, and may be any local area network (LAN), wireless local area network (WLAN), metropolitan area network (MAN), Intranet, Extranet, WAN, virtual private network (VPN), or any other appropriate architecture or system that facilitates communications in a network environment. A network can comprise any number of hardware or software elements coupled to (and in communication with) each other through a communications medium.

[0055] In one particular instance, the architecture of the present disclosure can be associated with a service provider digital subscriber line (DSL) deployment. In other examples, the architecture of the present disclosure would be equally applicable to other communication environments, such as any wireless configuration, any enterprise wide area network (WAN) deployment, cable scenarios, broadband generally, fixed wireless instances, fiber to the x (FTTx), which is a generic term for any broadband network architecture that uses optical fiber in last-mile architectures, and data over cable service interface specification (DOCSIS) cable television (CATV). The architecture of the present disclosure may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network.

[0056] Any of the suites, backend systems, the conductor, end to end system management, etc. can be representative of network elements that can facilitate the video management activities discussed herein. As used herein in this Specification, the term 'network element' is meant to encompass any of the aforementioned elements, as well as routers, switches, cable boxes, iPads, end-user devices generally, endpoints, gateways, bridges, STBs, loadbalancers, firewalls, inline service nodes, proxies, servers, processors, modules, or any other suitable device, component, element, proprietary appliance, or object operable to exchange content in a network environment. These network elements may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information.

[0057] In one implementation, these network elements can include software to achieve (or to foster) the video management activities discussed herein. This could include the implementation of instances of domain manager 11a-f. Additionally, each of these elements can have an internal structure (e.g., a processor, a memory element, etc.) to

facilitate some of the operations described herein. In other embodiments, these video management activities may be executed externally to these elements, or included in some other network element to achieve the intended functionality. Alternatively, these network elements may include software (or reciprocating software) that can coordinate with other network elements in order to achieve the video management activities described herein. In still other embodiments, one or several devices may include any suitable algorithms, hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof.

[0058] Turning to FIGURE 7, FIGURE 7 illustrates a service policy architecture 70 associated with the present disclosure. This particular configuration includes a service directory 72, a service policy 74, a service connection manager 76, a client directory 78, a system management console 80, and a Jabber session manager (JSM) 82. Note that 3GPP IMS defines a services infrastructure intended to enable an easy introduction of new services by the use of filter criteria and application servers. However, it relies on a much more static model where services are triggered by IMS filter criteria and invoked on behalf of users. Service discovery, scalability, policies, etc. are not considered as part of this paradigm. In particular, IMS does not address interconnecting services hosted on different platforms, communicating with different protocols and possibly developed in different languages. In addition, lacking is how services register, discover each other, and how to provide a virtual service experience at a generic service oriented plane level, while keeping certain policy and security enforcement functions. Furthermore, no system defines how to orchestrate multiple services into a virtual service in a flexible workflow (IMS filter criteria and application servers rely on a triggering message and fixed sequential order of invocation of the application servers). Therefore, it is still difficult and expensive for service providers to easily and quickly deploy various value-added services, such as a social network service, a companion device service, etc. Similarly, other existing video service architectures, also do not provide the features for end-users to discover both new and existing services, as well as notifying users about service change information, new services available, existing services update, etc.

[0059] The architecture of FIGURE 7 defines a services infrastructure, which treats services similar to any other entity in the control plane. The platform provides for a flexible,

scalable, reliable, secure, and policy-based control of services by providing: 1) a services directory, which enables XMPP and non-XMPP-based services to be registered and discovered and allows for those services to be virtualized; 2) a services publish, which can publish new deployed or new updated XMPP and non-XMPP-based services to subscribed devices, endpoints, users, and services in near real-time; 3) a service policy, which controls how services and clients can interact with each other, and is provided in a uniform and consistent manner to devices, endpoints, users, and services by relying on the authenticated JID associated with these; 4) a service virtualization, which enables XMPP service invocation to be done transparently to one of several service instances in accordance with a service routing policy; 5) a service orchestration, which can involve XMPP and non-XMPP-based services for different virtual services for various use cases; and 6) a service management, which can share the XMPP connection and messaging infrastructure with a service request/response at the video service control plane, centrally configure XMPP services, manage XMPP services' lifecycle, deploy XMPP services, and update XMPP services.

[0060] The architecture also provides a loosely coupled, flexible, open, scalable, and reliable services infrastructure suitable for a video services control plane. The platform supports a (transparent) virtualization of services with flexible service routing to specific service instances. It also supports an easy publishing and management of both XMPP and non-XMPP services. It further supports comprehensive service policies and allows for unified integration with client policies.

[0061] Note that central to the architecture of FIGURE 7 is an XMPP messaging fabric that integrates XMPP and non-XMPP (e.g., web services) based services. Additionally, the architecture includes a set of client endpoints/devices and users, each of which have an authenticated identity and various information stored in a client directory. The architecture also includes a (single) persistent connection to each client supporting bi-directional, authenticated, and authorized communication with other entities connected to the messaging fabric.

[0062] Additionally included in the architecture is an application and services model similar to the user/client model in terms of identities, connections, policies, etc. An orchestration engine is configured for supporting workflows and rules for flexible service implementation. The publish subscribe mechanism is provided for triggered communication

to a set of entities based on a variety of criteria (XMPP PubSub topic-based). A loosely coupled service oriented platform is provided across a multiprotocol message bus by connecting end-users closely. Then services can be abstractly decoupled from each other, and connected together through the platform as logical endpoints, which are exposed as virtual services to end-users. The architecture also includes various service-specific features in the form of a service directory, service publish, service policy, service security, service virtualization, service orchestration and service management.

[0063] In one particular example, the paradigm of FIGURE 7 is based on the notion of having devices (physical hardware), endpoints (e.g., a soft client), users, services, etc. all assigned a name (a Jabber Identifier (JID)) by which they can be identified and messages can be sent to them. Users can be associated with one or more devices (e.g., via login), thereby enabling personalized services as well as device-specific services. Devices, endpoints, and users are all registered in the client directory, where profile information is associated with them. Profile information includes services subscribed to, parental control settings, devices registered for services, content formats supported, etc. A cloud DVR service for user A can (for example) look at the devices user A has to determine suitable format(s) to record content in for user A. Users may belong to an account (household), that includes multiple users and devices with different settings for each.

[0064] JIDs can be authenticated and, hence, form the identity basis for authorization and service policies in the system. This applies universally to devices, endpoints, users, and services and hence provides for a simple and consistent security architecture inside the control plane. The XMPP control plane supports a number of native services/features (e.g., client directory, or publish/subscribe) and allows for additional non-native services/features to be connected to the control plane. Such services/features may be XMPP-based or they may be based on other protocols (e.g., HTTP web services) through the use of interface adapters, which can ensure those services are treated similar to XMPP-based services. All services can be registered in the service directory, where they can easily be discovered. All non-native entities (clients and services beyond the base platform services) are connected to the XMPP control plane by the use of a connection manager. Endpoints, devices, and users (also known as clients) can connect through a client connection manager, whereas services connect through a service connection manager. The

connection managers provide and ensure a single point of entry/exit for the entity, ensuring routing through the necessary service, security, and policy enforcement inside the control plane, authenticated identity for the entity, persistent connection with bi-directional communication support to the entity. Authentication is done once, and communication through NATs is supported. Examples of services include a session manager, which enables tracking of cloud-based session state for an HTTP ABR based content delivery service.

[0065] Using XMPP session state tracking, for example, enables the cloud to track on-going sessions for a user, initiate diagnostics sessions, and session coupling with the CDN infrastructure via the external interfaces supported. Notification-based services are enabled by the publish-subscribe functionality provided by XMPP. Note how this functionality applies to all entities in the system, including services and soft clients (i.e., not just media devices).

[0066] One key design principle is to enable service orchestration by the use of workflows and rules engines. Existing technologies can be used for this, where use of these technologies (being used by the services) are accessible via the XMPP control plane, addressable by a JID, and/or are subject to the control plane service policies enforced by the control plane. Compared to the IMS service model (filter criteria and SCIM), it is also different in that it does not rely on a triggering message (e.g., call setup/SIP INVITE), nor does it limit the rules or workflows that can be supported. Another advantage of the architecture is illustrated by the BSS/OSS adapter, which provides a unified interface to back-office subscriber management and billing systems typical of subscription-based service provider deployments. While the client directory provides the default place to store such data, service specific data may be better stored by an individual service. Services can register their interest in such new data, and the BSS/OSS adapter simply sends a message to the JID(s) that have expressed interest in such data.

[0067] In operation, the video control plane can provide a message and event based video services mechanism to enable 3rd wave video systems based on HTTP ABR content delivery technology, while supporting any content, to any device, anywhere, any time. The control plane enables secure, authenticated, and personalized bi-directional control plane services in a scalable manner for endpoints, devices, users, and services. The system can define and treat endpoints, devices, users, and services in a similar manner and provide a

consistent message routing infrastructure with security and service policy infrastructure. The control plane leverages and combines open web-based technologies in the form of XMPP and HTTP/web services, supports notification services, and is easily customizable and extensible by use of the orchestration engine.

[0068] Turning to FIGURE 8, FIGURE 8 is a simplified block diagram illustrating a virtual service and service instances framework. FIGURE 8 includes a plurality of service instances 90, which are coupled to a virtual service 85 that may involve multiple features. Note that the service platform of the present disclosure involves the following key concepts: a services directory, a services publish component, a service policy, a service virtualization, a service orchestration, and a service management component.

[0069] The services interconnection architecture (which can be referred to as the conductor) can be based on the concept of features. A feature can include any collection of functions (e.g., those advertised as a unit). Each feature can be implemented with an XMPP-based protocol and assigned a protocol namespace (an XML namespace). Features can be defined, for example, in a protocol document such as an XMPP Extension Protocol (XEP).

[0070] A key concept of the services architecture is that features are implemented with an XMPP-based protocol (possibly via a protocol adapter) and assigned a (static) protocol namespace (an XML namespace). Before a service connects to the fabric, the service should register its list of supported features and expose its external interface definition, such as WSDL for web service, or WADL for a REST based web service. The services and the list of supported features and interface definitions are tracked in the service directory. As part of the registration, the service can be (statically or dynamically) associated with a service JID. The service JID is an XMPP JID like all other JIDs in the system (e.g., for devices, endpoints, and users) and, hence, all other entities in the system can send messages to it. The service instance is the actual running copy of a service. Each service instance is also associated with a service instance JID. The service instance JID is an XMPP JID like all other JIDs in the system (e.g., for devices, endpoints, and users). Clients or other services looking for a service implementing a given feature can query the service directory by protocol namespace. The service directory can answer these namespace queries with the JID of the matching service.

[0071] The services architecture supports XMPP and non-XMPP services, both of which connect to the conductor via a Service Connection Manager. In the case of non-XMPP services, the complete set of features available may, or may not, be available via XMPP. Some services, such as Electronic Program Guides (EPGs) that involve potentially large amounts of cacheable data, may (for example) be provided more efficiently via web services interfaces. The service directory allows those services and features to register in the service directory as well. Where one or more features are not accessible via the XMPP-based interface, the mapping can be to a web service (or other) interface instead. This provides a single service directory and mechanism for identifying and addressing services in the system.

[0072] The namespace to identity (JID) mapping and the use of the service directory is important for achieving a loosely coupled services architecture. It means that new services can be developed, protocols designed, services and clients authored and deployed in a conductor system without any of the conductor components having to understand anything about the protocol used by that service. This engenders the service velocity and extensibility, which helps make the conductor different from traditional video middleware platforms.

[0073] Service publish is another key aspect of the video platform. With the use of a common messaging fabric that supports devices, endpoints, users, and services in a consistent manner, it is important to have mechanisms in place to publish the information of new deployed or new updated XMPP and non-XMPP-based services to other entities in near real-time. The service policy (as illustrated in FIGURE 7) is the next aspect of the video platform. With the use of a common messaging fabric supporting devices, endpoints, users and services in a consistent manner, it is important to have mechanisms in place to control who can interact with whom and how.

[0074] Four categories of service policy functionality are considered: 1) a policy repository holds the actual policy definitions (e.g., made up of rules); 2) a policy decision point analyzes the policy definitions from the policy repository to decide whether a given action will violate the rules in a definition; 3) the policy enforcement point enforces the decision of the policy decision point by blocking or dropping messages to or from services;

and 4) the policy console allows an operator to configure, view, deploy, and monitor policy definitions.

[0075] In the conductor, the service directory and client directory functions provide the policy repository role, holding the policy definitions for services and clients (endpoints, devices and users), respectively. Therefore, at service or event delivery stage, who can send or receive what kind of information can be retrieved from either service directory or client directory, and enforced at the corresponding enforcement point. The service policy module acts as the policy decision point. The policy section in the system management console GUI acts as the policy console. The service connection manager, through which all non-native services traffic is sent, acts as the policy enforcement point for services, whereas the Jabber Session Manager, through which all client traffic is sent, acts as the policy enforcement point for clients.

[0076] Each service has a policy definition in the service directory (possibly by use of a default policy). Policy definitions are made up of rules. Rules may, for example, contain permit or deny clauses followed by a match criteria, and rules may be ordered by preference. Possible match criteria include: 1) sending entity (JID); 2) sending entity group (client group or service group); 3) message type (XMPP stanza type + first child element name + first child element XMLNS); 4) receiving entity (JID); 5) receiving entity group (client group or service group); and 6) adapter type. Since the conductor framework may have many services and many more clients, groups are used to make definitions and policy decisions more efficient. Services and clients are assigned to one or more groups, tracked by the service directory and client directory.

[0077] Service virtualization (illustrated in FIGURE 8) is the fourth important aspect of the video platform. The conductor provides a service virtualization capability that allows a virtual service to be provided by multiple service instances for scale and high availability. Service virtualization makes the multiple instances appear as a single, virtual service so clients only see the single virtual service.

[0078] Service virtualization is provided by two mechanisms: Service Registration (described above), and service routing, which routes messages to specific instances of a given virtual service. A virtual service is assigned its own JID and clients querying a protocol

namespace that corresponds to a virtual service can see the virtual service JID in the response. The actual service instances (and their JIDs) are essentially invisible to clients.

[0079] Service registration happens when services connect to the Service Connection Manager and register in the service directory. All services that register a given protocol namespace are considered to be instances of the same virtual service. A virtual service JID can be assigned to each protocol namespace by an administrator. If more than one service instance registers with the same protocol namespace and a virtual service JID has not been configured by the administrator, a random virtual service JID is assigned by the system.

[0080] In certain embodiments of the present disclosure, individual service instances do not appear in the service directory. Instead, the virtual service appears in the service directory. That way, when service consumers (clients or other services) look up a service in the service directory, a single entry is returned for a given service (independent of the number of instances that make up that virtual service). This allows service consumers to interact with the single virtual service instead of the individual instances. It also means that policy and other subsystems act at the virtual service level and do not need to deal with individual service instances.

[0081] Once multiple instances of a service are registered and a virtual service JID is assigned, service routing takes care of routing messages addressed to the virtual service JID to one of the service instances. The algorithm used to route messages can be based on one of many possible service routing policies. The system administrator can configure the service routing policy for each virtual service. Various routing policies could be provided, including: 1) simple or weighted round-robin; 2) least loaded; 3) proximity-based; and 4) priority based on sender.

[0082] The service consumer is unaware of the service routing policy in use. As far as the service consumer is concerned, messages are sent to and received from the virtual service JID. This allows instances to be added, removed, or to fail without affecting the service consumer or the service. Alternatively, the service consumer could specify a desired service routing policy.

[0083] Service virtualization frees the service provider to deploy nodes in any configuration and distribute services among the nodes based in the specific needs of a given

deployment. Certain services can be centralized and others distributed, based on the traffic patterns generated by a given service.

[0084] Service orchestration is the fifth important aspect of the video platform. The conductor provides a service orchestration capability that allows a virtual orchestrated service to be provided by multiple virtual services and to be either invoked by the service consumer or triggered by a publishing system event. On the conductor platform, each service is a properly grained (trying to make it maximally reusable) self-contained software/application entity, for different business logics. Multiple services would be composed and coordinated to create higher-level business processes. Orchestration describes how services would interact with each other at the message level, including the business logic and execution order of the interactions.

[0085] Service orchestration can be provided by three mechanisms: service composition, service invocation, and service flow automation. Service composition occurs when services register in the service directory, and with the use of workflows and rules, the operator retrieves services' supported features and interface definition from the service directory, composite a virtual service, which is assigned a virtual service JID in turn and registered into service directory for other entity to consume. Hence, there is a unified consistent way for both 'real' services and 'composite' services, which are essentially invisible to clients. Since the service directory contains the service definition of both XMPP-based services and HTTP-based services, the conductor can composite both XMPP-based services and HTTP-based (or other) services into one workflow and present them as a virtual service.

[0086] Service invocation can occur in two different cases. The first one is a client or another service invoking a virtual composite service and expecting to receive a response in a short time. Another case involves a published event that triggers a virtual composite service and does not expect to receive a response. In either case, a message would be sent to the virtual composite service just like any other 'real' service through the XMPP service control plane, and the virtual composite service would then map to the orchestration engine by the service virtualization mechanism. When the orchestration engine receives the message, it would map the virtual composite service in the message to a workflow, and then execute the flow.

[0087] Service flow automation can occur when the conductor orchestration engine executes workflow and rules. The conductor orchestration engine can automatically dispatch service request to an actual service instance via the service virtualization and service routing mechanism of the XMPP service control plane. Since workflow can contain both XMPP-based services and HTTP-based services, one advantage is the conductor orchestration engine can automatically recognize the service protocol during workflow execution, and dispatch service request via adequate protocol. All of this is invisible to the invoker of the virtual composite service.

[0088] Service management is the sixth important aspect of the video platform, sharing the XMPP connection and messaging bus with the service request/response message. All management messages of a virtual service and its instance are also transported through the XMPP connection. Therefore, an operator does not need to maintain another channel or infrastructure to pass through management messages. The operator can still centrally control services and service instances from various points including creation, provision, deploy, configuration, life cycle, version control, etc.

[0089] In terms of an example conductor topology, conductor 28 can reflect a distributed messaging and service interconnect platform. A conductor cloud is made up of one or more conductor nodes. A node is a physical server or a virtual machine running the conductor platform software. Each node provides core message routing capabilities plus one or more of the core conductor functions. The nodes can be interconnected via encrypted TCP sockets.

[0090] Services in a conductor system can be hosted on application servers. There are typically conductor nodes dedicated to interconnecting with these application servers. These service interconnection nodes can run one or more service connection managers (SCM). One SCM is generally dedicated to a single application server. Clients in a conductor system connect via supported client connection protocols, such as XMPP and BOSH. There can be conductor nodes, distributed out in the network, dedicated to client connections. These distributed conductor nodes can run one or more client connection managers (also referred to as domain managers, as used herein in this specification).

[0091] Turning to FIGURE 9, FIGURE 9 is a simplified flowchart 900 illustrating example activities associated with basic service registration and lookup activities of the

present disclosure. The method may begin at 901, where a new service with one or more sets of features (collection of functions) is defined with a set of interfaces using XMPP. Adapters may be used for non-XMPP-based features as well. At 902, an XML namespace is assigned to each feature for the service. At 903, the service connects to the conductor infrastructure, and is assigned a JID. The service registers itself in the conductor service directory and, thereby, creates a mapping between the (static) XML namespaces for its features and the JID assigned to it. At 904, the service may also expose non-XMPP-based features accessed by other protocols (e.g., web services). XML namespaces for the non-XMPP-based features are registered in the service directory as for XMPP-based services; however, the mapping is not to a JID. Instead, a URL (e.g., for web services) or other mechanism is provided. At 905, a device, user, endpoint, or other service seeks to invoke a service that supports feature X. The entity looks up the protocol namespace in the service directory and retrieves the JID for the service. The entity then simply sends the message to the JID returned using the conductor messaging infrastructure. Note that for non-XMPP-based services, the service can be invoked as per the protocol returned (e.g., web services).

[0092] Turning to FIGURE 10, FIGURE 10 is a simplified flowchart 1000 associated with service virtualization and routing. This particular flow may begin at 1001, where multiple instances of a service are provided, all supporting the same set of features/services. At 1002, a virtual service JID is defined for the service in question. Alternatively, an XML namespace is noted as being for virtual services. At 1003, an instance of service S (S1) registers with the service directory. The service directory registers the service S1 under its virtual service JID (e.g., "VS") and internally associates "VS" with S1. If a virtual service JID had not been defined, one is assigned. At 1004, another instance of service S (S2) registers with the service directory causing "VS" to also be associated with S2. At 1005, a device, user, endpoint, or other service needs to invoke a service that supports feature X, which is supported by "VS" (via the Sx instances). The entity looks up the protocol namespace in the service directory and retrieves the JID for the service ("VS"). The entity then simply sends the message to the JID returned ("VS") using the conductor messaging infrastructure. The conductor infrastructure notes the JID is a virtual JID and uses internal service routing logic to route the message to a specific instance of the service.

Different service routing policies can be used (e.g., round-robin, least-loaded, proximity-based, etc.).

[0093] Turning to FIGURE 11, FIGURE 11 is a simplified flowchart 1100 associated with servicing communication policy. This particular flow may begin at 1101, where Client C (identified by JID "C") sends a message to service S (identified by JID "S") via the conductor messaging infrastructure. At 1102, the conductor infrastructure looks up the service policy for S in the service directory to determine if C is allowed to send messages to S. At 1103, various service policy matching criteria may be used to determine whether C is allowed to send a message to S. The service policy may also include group permissions; if so, the conductor Infrastructure looks up client C in the client directory to determine to which groups it belongs. At 1104, if C is allowed to send a message to S, normal JID-based service routing to S is performed. If not, the message is rejected. At 1105, Service Sx (identified by JID "Sx") sends a message to service Sy (identified by JID "Sy") via the conductor messaging infrastructure. At 1106, the conductor infrastructure looks up the service policy for Sy in the service directory to determine if Sx is allowed to send messages to Sy. At 1107, the various service policy matching criteria may be used to determine whether Sx is allowed to send a message to Sy.

[0094] The service policy may also include group permissions; if so, the conductor infrastructure looks up Service Sx in the service directory to determine to which groups it belongs. At 1108, if Sx is allowed to send a message to Sy, normal JID-based service routing to Sy is performed. If not, the message is rejected. At 1109, Client Ca (identified by JID "Ca") sends a message to client Cb (identified by JID "Cb") via the conductor messaging infrastructure. At 1110, the conductor infrastructure looks up the communication policy for Client Cb in the client directory to determine if Ca is allowed to send messages to Cb. At 1111, various policy matching criteria may be used to determine this, including group permissions. At 1112, if Client Ca is allowed to send a message to Cb, normal JID-based service routing to Cb is performed. If not, the message is rejected.

[0095] As identified previously, a network element can include software (e.g., domain manager 11a-f) to achieve the video management operations, as outlined herein in this document. In certain example implementations, the video management functions outlined herein may be implemented by logic encoded in one or more tangible, non-

transitory media (e.g., embedded logic provided in an application specific integrated circuit [ASIC], digital signal processor [DSP] instructions, software [potentially inclusive of object code and source code] to be executed by a processor [processors provided in any of the suites, in conductor 28, in media gateway 34, anywhere in legacy home 38, video system home 34, in backend systems 15, in end to end system management 30, etc.]). In some of these instances, a memory element [provided in any of the suites, in conductor 28, in media gateway 34, anywhere in legacy home 38, video system home 34, in backend systems 15, in end to end system management 30, etc.] can store data used for the operations described herein. This includes the memory element being able to store instructions (e.g., software, code, etc.) that are executed to carry out the activities described in this Specification. The processors can execute any type of instructions associated with the data to achieve the operations detailed herein in this Specification. In one example, the processor could transform an element or an article (e.g., data) from one state or thing to another state or thing. In another example, the activities outlined herein may be implemented with fixed logic or programmable logic (e.g., software/computer instructions executed by the processor) and the elements identified herein could be some type of a programmable processor, programmable digital logic (e.g., a field programmable gate array [FPGA], an erasable programmable read only memory (EPROM), an electrically erasable programmable ROM (EEPROM)) or an ASIC that includes digital logic, software, code, electronic instructions, or any suitable combination thereof.

[0096] Any of these elements (e.g., the network elements, etc.) can include memory elements for storing information to be used in achieving the video management operations as outlined herein. Additionally, each of these devices may include a processor that can execute software or an algorithm to perform the video management activities as discussed in this Specification. These devices may further keep information in any suitable memory element [random access memory (RAM), ROM, EPROM, EEPROM, ASIC, etc.], software, hardware, or in any other suitable component, device, element, or object where appropriate and based on particular needs. Any of the memory items discussed herein should be construed as being encompassed within the broad term 'memory element.' Similarly, any of the potential processing elements, modules, and machines described in this Specification should be construed as being encompassed within the broad term 'processor.'

Each of the network elements can also include suitable interfaces for receiving, transmitting, and/or otherwise communicating data or information in a network environment.

[0097] Note that with the examples provided above, interaction may be described in terms of two, three, or four network elements. However, this has been done for purposes of clarity and example only. In certain cases, it may be easier to describe one or more of the functionalities of a given set of flows by only referencing a limited number of network elements. It should be appreciated that video system 10 (and its teachings) are readily scalable and, further, can accommodate a large number of components, as well as more complicated/sophisticated arrangements and configurations. Accordingly, the examples provided should not limit the scope or inhibit the broad teachings of video system 10, as potentially applied to a myriad of other architectures.

[0098] It is also important to note that the steps in the preceding FIGURES illustrate only some of the possible scenarios that may be executed by, or within, video system 10. Some of these steps may be deleted or removed where appropriate, or these steps may be modified or changed considerably without departing from the scope of the present disclosure. In addition, a number of these operations have been described as being executed concurrently with, or in parallel to, one or more additional operations. However, the timing of these operations may be altered considerably. The preceding operational flows have been offered for purposes of example and discussion. Substantial flexibility is provided by video system 10 in that any suitable arrangements, chronologies, configurations, and timing mechanisms may be provided without departing from the teachings of the present disclosure.

[0099] Numerous other changes, substitutions, variations, alterations, and modifications may be ascertained to one skilled in the art and it is intended that the present disclosure encompass all such changes, substitutions, variations, alterations, and modifications as falling within the scope of the appended claims. In order to assist the United States Patent and Trademark Office (USPTO) and, additionally, any readers of any patent issued on this application in interpreting the claims appended hereto, Applicant wishes to note that the Applicant: (a) does not intend any of the appended claims to invoke paragraph six (6) of 35 U.S.C. section 112 as it exists on the date of the filing hereof unless the words "means for" or "step for" are specifically used in the particular claims; and (b)

does not intend, by any statement in the specification, to limit this disclosure in any way that is not otherwise reflected in the appended claims.

WHAT IS CLAIMED IS:

1. A method, comprising:
 - establishing a connection between a client and a messaging fabric of a conductor element associated with a video system;
 - defining a service having a set of features using a set of interfaces associated with an Extensible Messaging and Presence Protocol (XMPP);
 - assigning a plurality of XML namespaces for the set of features of the service;
 - assigning an identifier to the service; and
 - registering the service in a service directory in order to create a mapping between the XML namespaces and the identifier.
2. The method of claim 1, wherein the service exposes non-XMPP-based features to be accessed by other protocols.
3. The method of claim 1, further comprising:
 - registering non-XMPP-based features in the service directory using a uniform resource locator (URL).
4. The method of claim 1, wherein the service is invoked by a device in order to support a particular feature of the video system.
5. The method of claim 1, further comprising:
 - accessing a protocol namespace in the service directory;
 - retrieving a particular identifier associated with a particular service; and
 - communicating a message using a messaging infrastructure of the conductor element.
6. The method of claim 1, wherein a virtual service identifier is defined for a particular service.

7. The method of claim 1, further comprising:
receiving a message associated with a particular service; and
evaluating a service policy in the service directory to determine permissions associated with messaging for the particular service.

8. The method of claim 7, wherein the service policy includes matching criteria used to determine whether a particular client is permitted to send messages to the particular service.

9. The method of claim 7, further comprising:
evaluating which groups are associated with a particular client that sent the message, wherein the service policy includes group permissions associated with a plurality of clients.

10. The method of claim 1, further comprising:
providing a plurality of virtual services to the client using multiple service instances for each of the plurality of virtual services such that they appear as a single virtual service to the client.

11. The method of claim 10, further comprising:
providing service routing to route messages to specific instances of a particular one of the plurality of virtual services.

12. The method of claim 1, further comprising:
registering a protocol namespace in the service directory;
assigning a virtual service identifier for the protocol namespace; and
routing messages addressed to the virtual service identifier to a particular one of a plurality of service instances.

13. Logic encoded in one or more non-transitory media that includes instructions for execution and when executed by a processor is operable to perform operations, comprising:

establishing a connection between a client and a messaging fabric of a conductor element associated with a video system;

defining a service having a set of features using a set of interfaces associated with an Extensible Messaging and Presence Protocol (XMPP);

assigning a plurality of XML namespaces for the set of features of the service;

assigning an identifier to the service; and

registering the service in a service directory in order to create a mapping between the XML namespaces and the identifier.

14. The logic of claim 13, wherein the service exposes non-XMPP-based features to be accessed by other protocols.

15. The logic of claim 13, the operations further comprising:

registering non-XMPP-based features in the service directory using a uniform resource locator (URL).

16. The logic of claim 13, wherein the service is invoked by a device in order to support a particular feature of the video system.

17. The logic of claim 13, the operations further comprising:

accessing a protocol namespace in the service directory;

retrieving a particular identifier associated with a particular service; and

communicating a message using a messaging infrastructure of the conductor element.

18. An apparatus, comprising:

a memory element configured to store instructions;

a processor coupled to the memory element; and

a conductor element, wherein the apparatus is configured to:

establish a connection between a client and a messaging fabric of a conductor element associated with a video system;

define a service having a set of features using a set of interfaces associated with an Extensible Messaging and Presence Protocol (XMPP);

assign a plurality of XML namespaces for the set of features of the service;

assign an identifier to the service; and

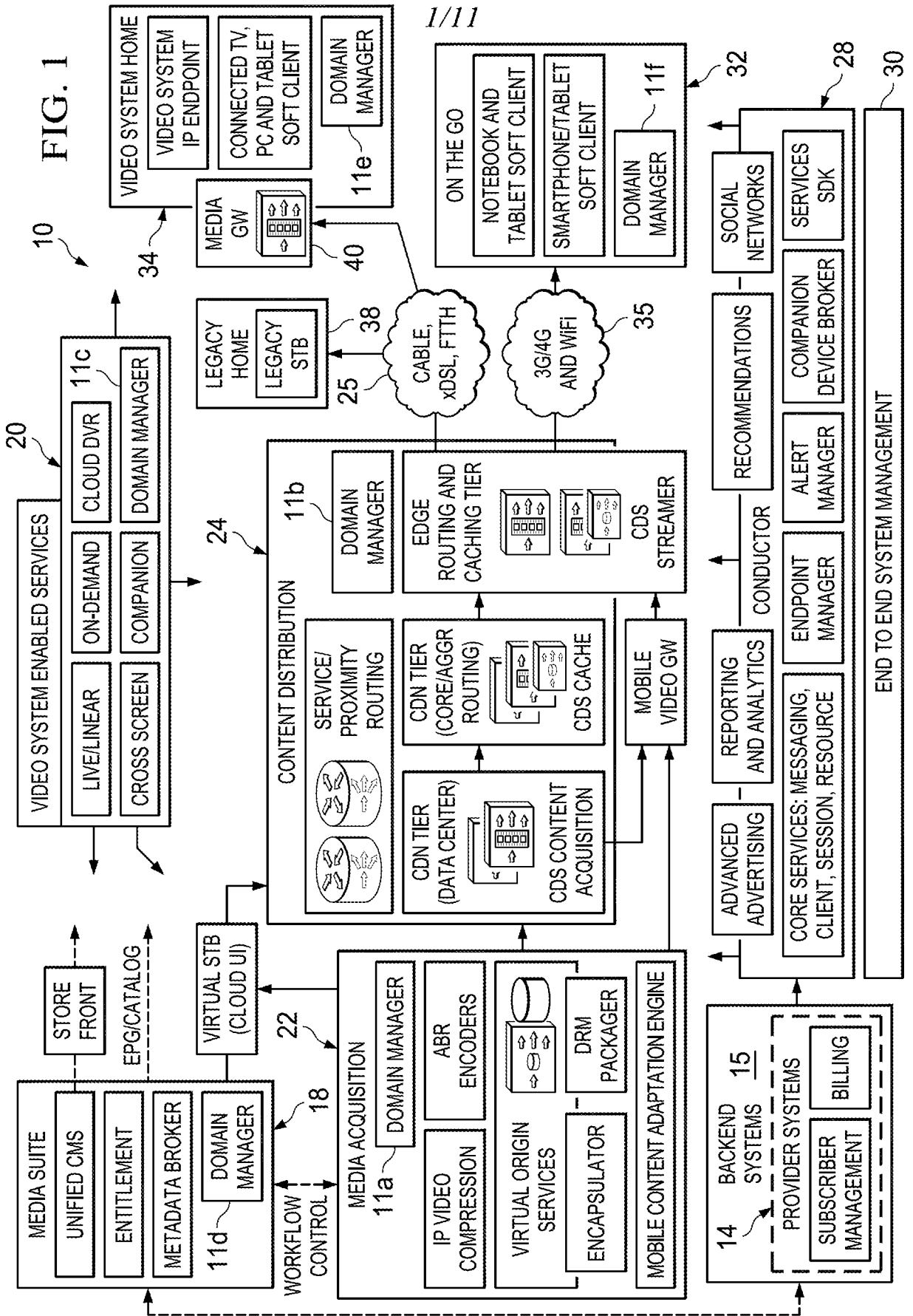
register the service in a service directory in order to create a mapping between the XML namespaces and the identifier.

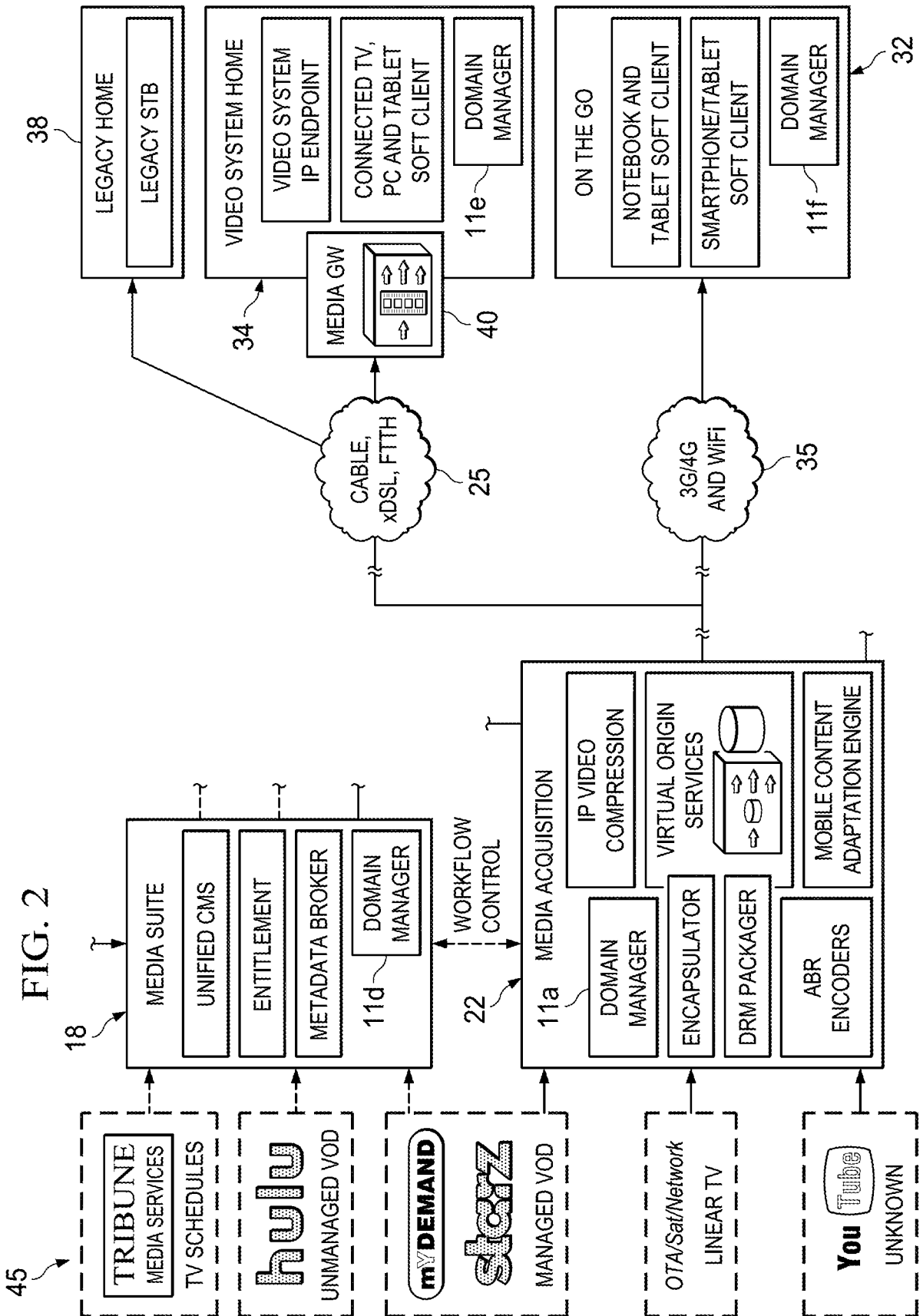
19. The apparatus of claim 18, wherein the service is invoked by a device in order to support a particular feature of the video system.

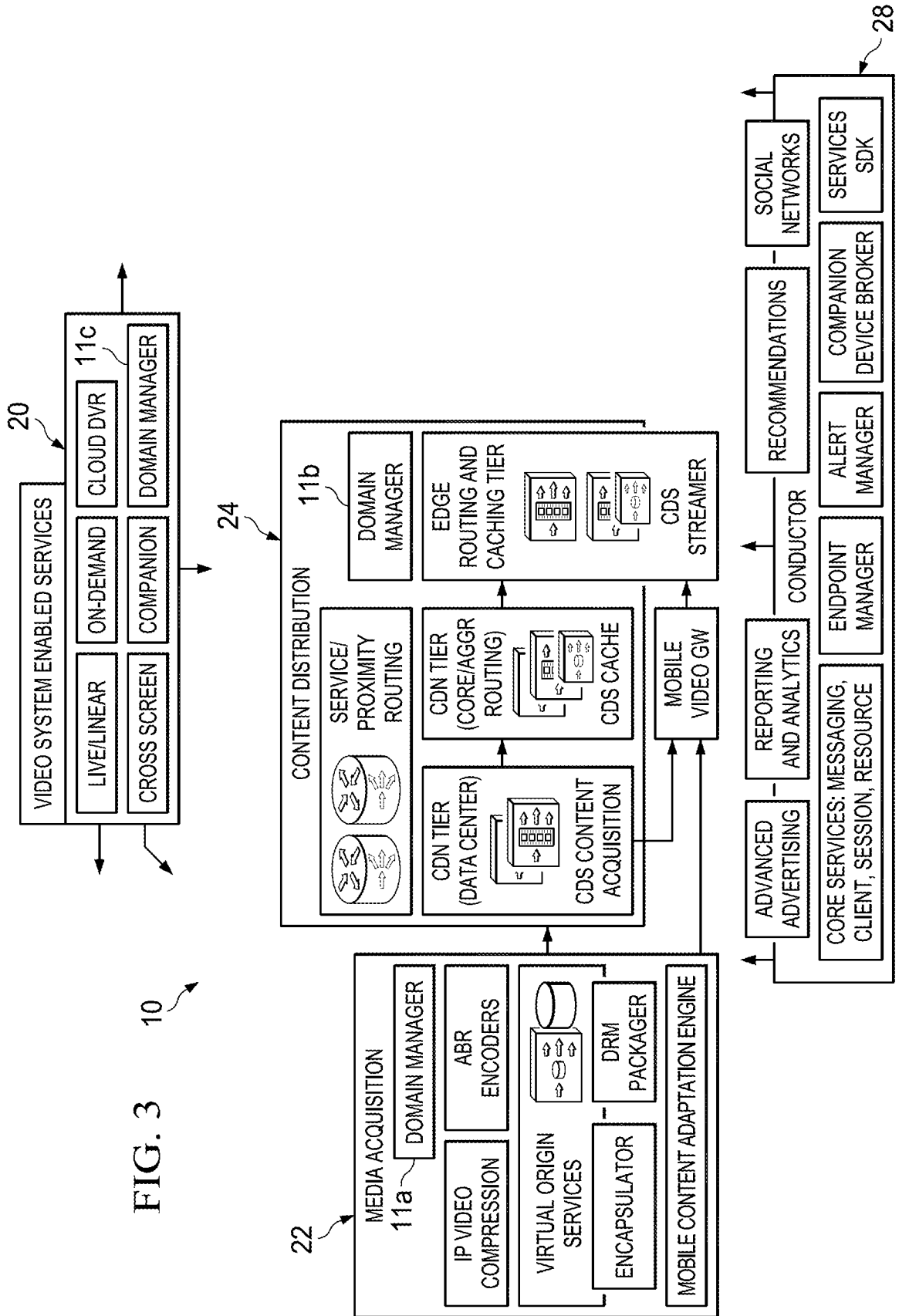
20. The apparatus of claim 18, wherein the apparatus is further configured to:

receive a message associated with a particular service; and

evaluate a service policy in the service directory to determine permissions associated with messaging for the particular service, wherein the service policy includes matching criteria used to determine whether a particular client is permitted to send messages to the particular service.







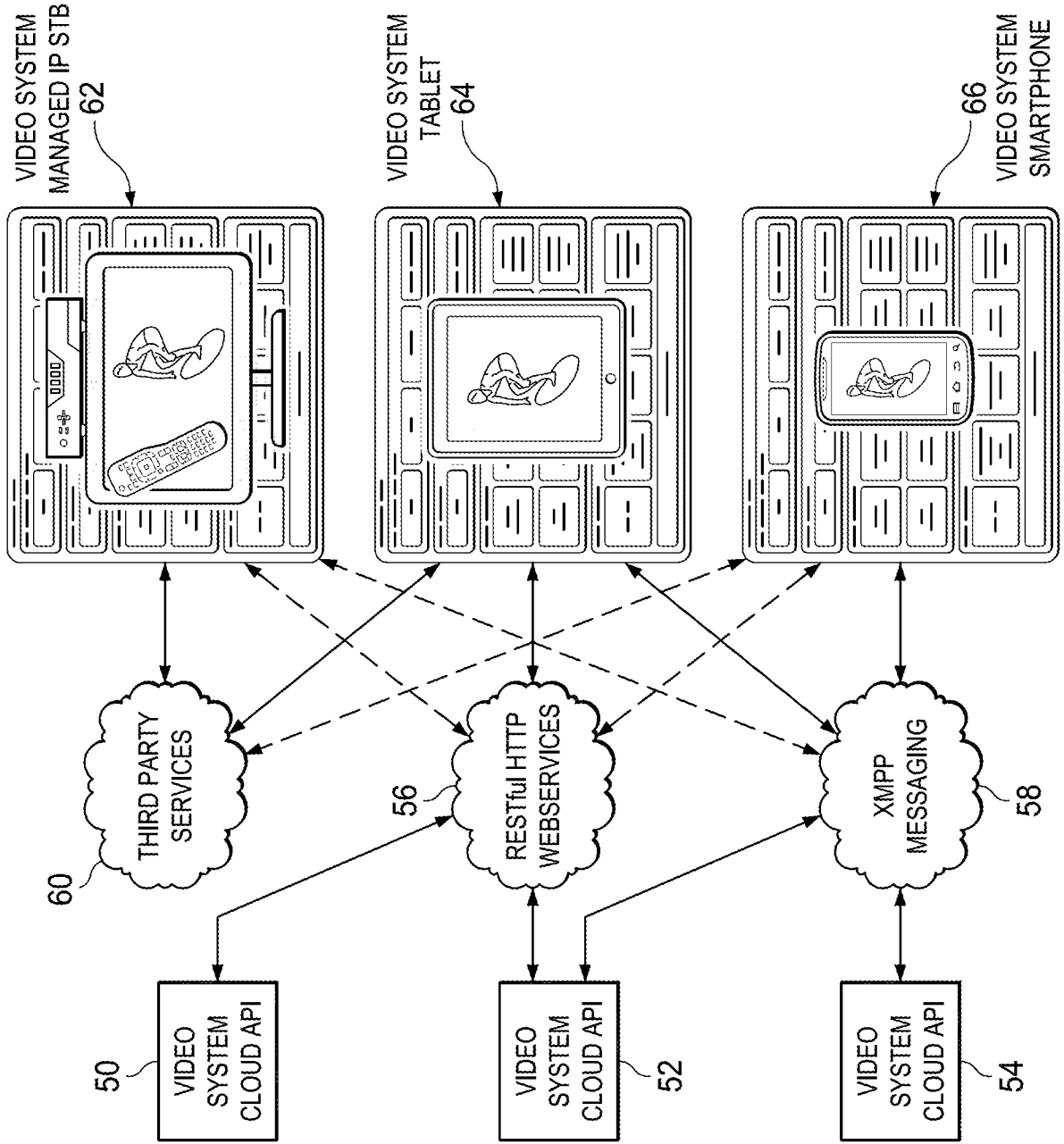


FIG. 4

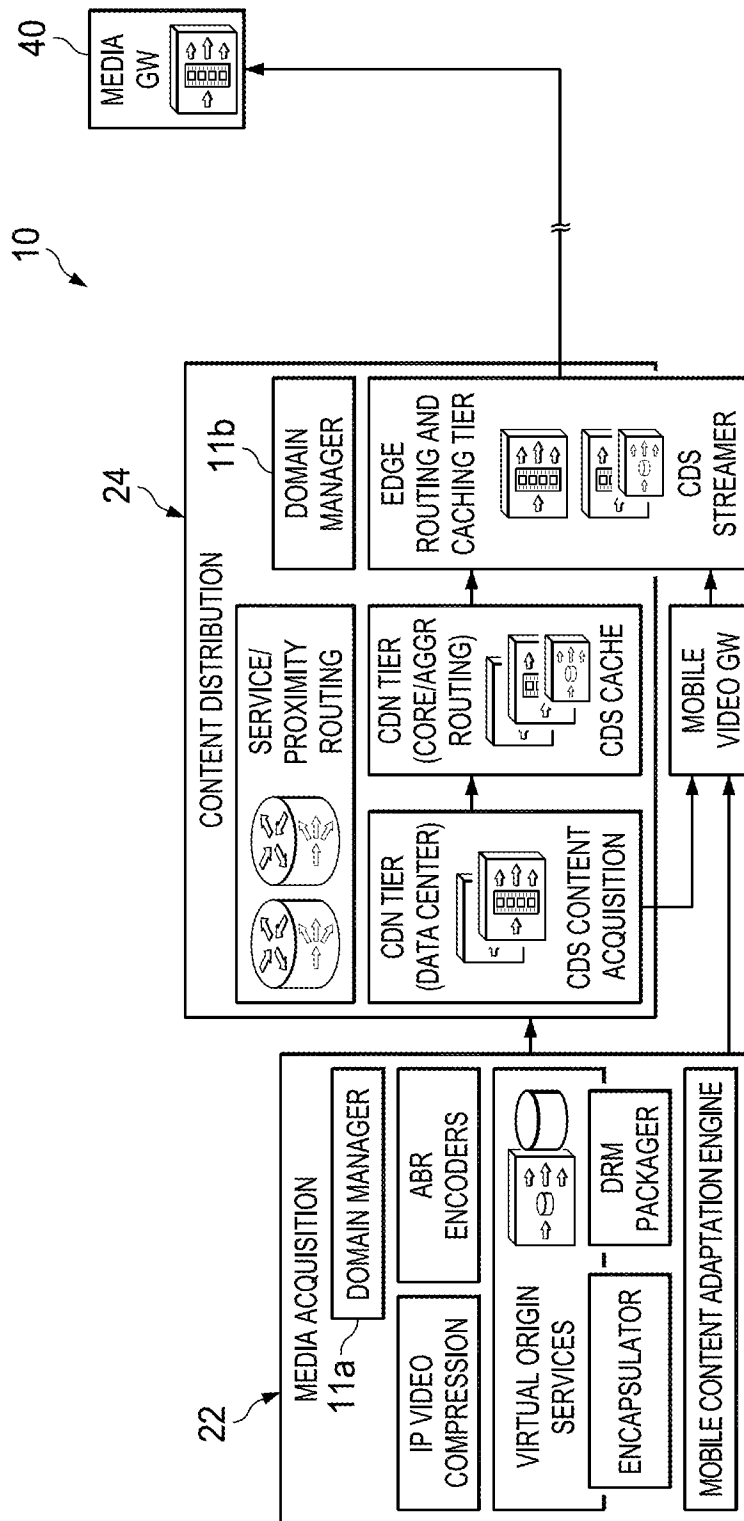
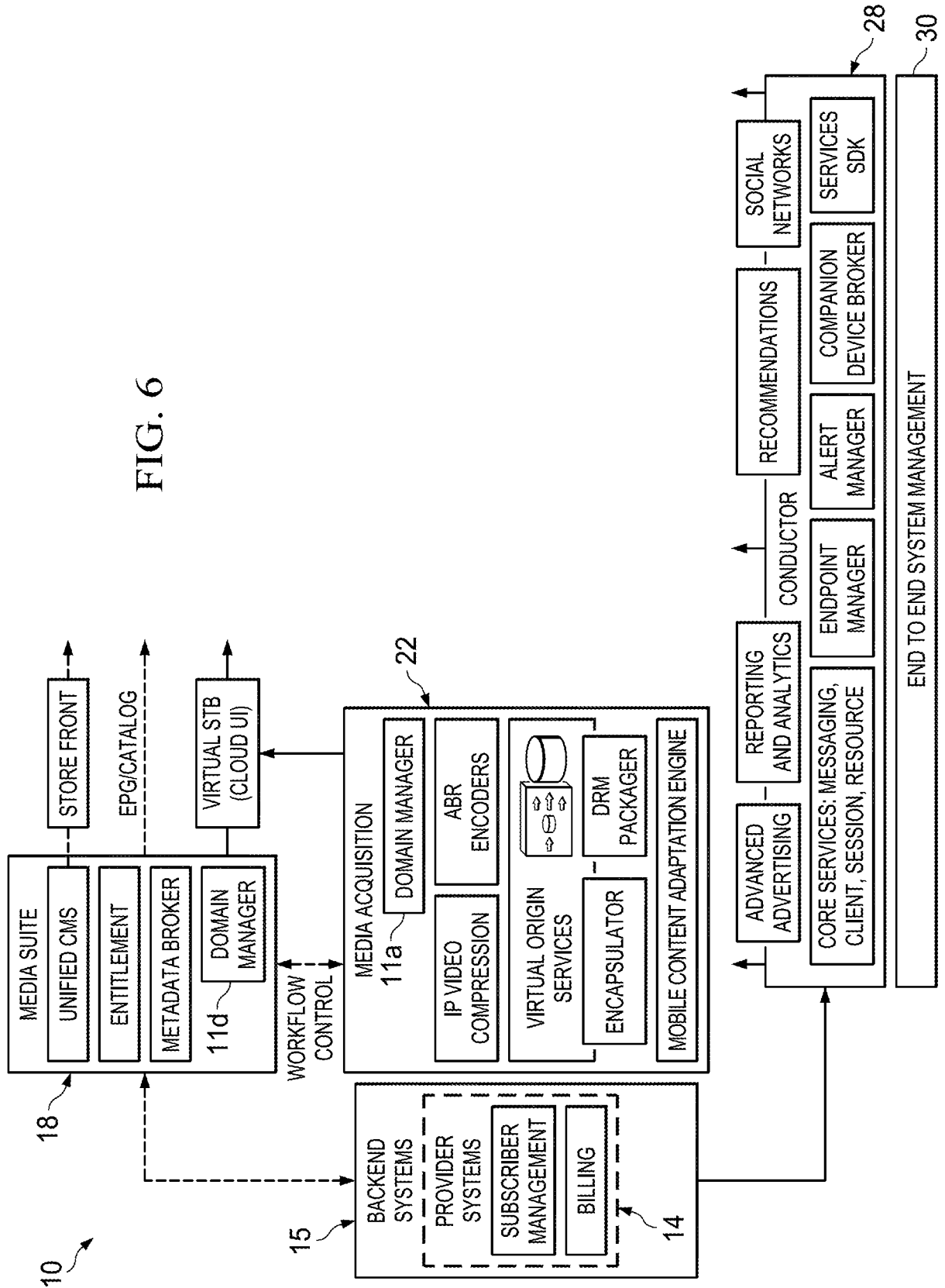
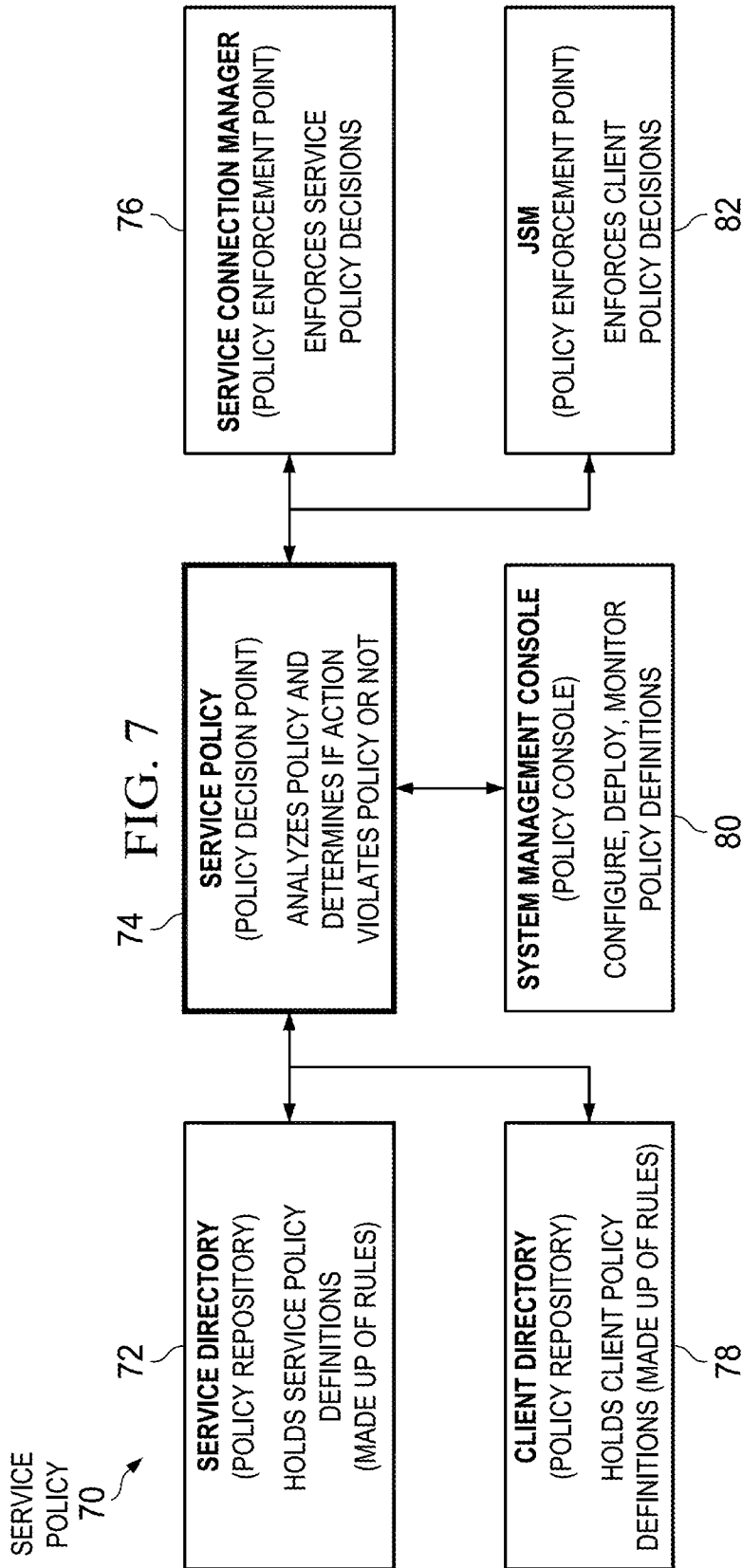
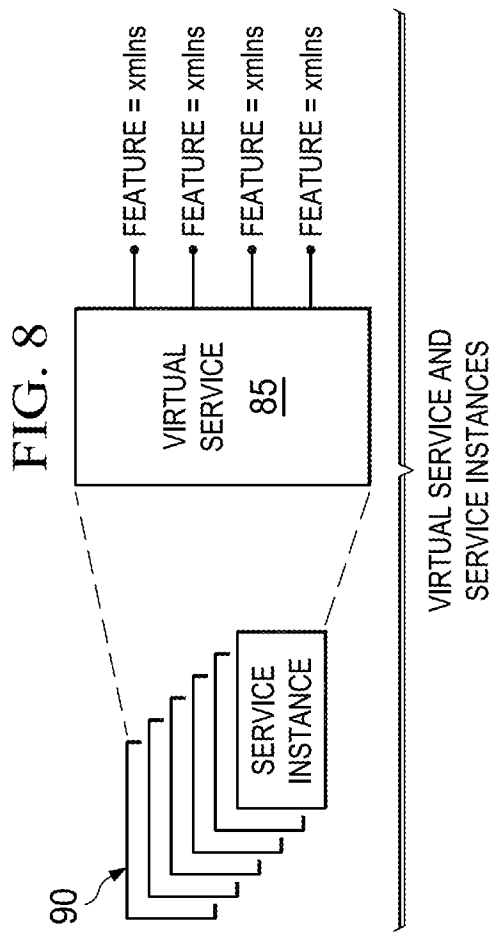


FIG. 5

FIG. 6







9/11

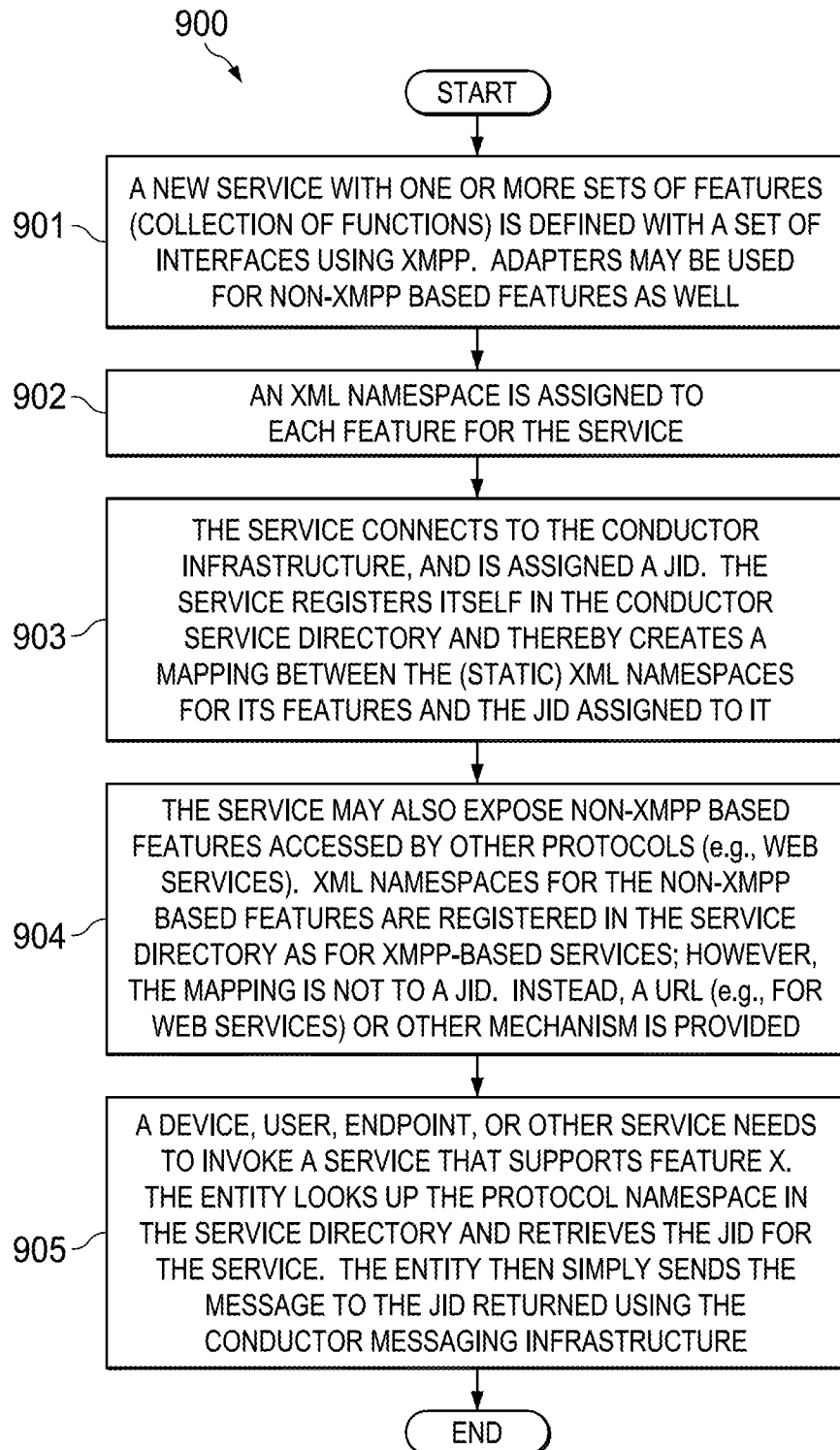


FIG. 9

10/11

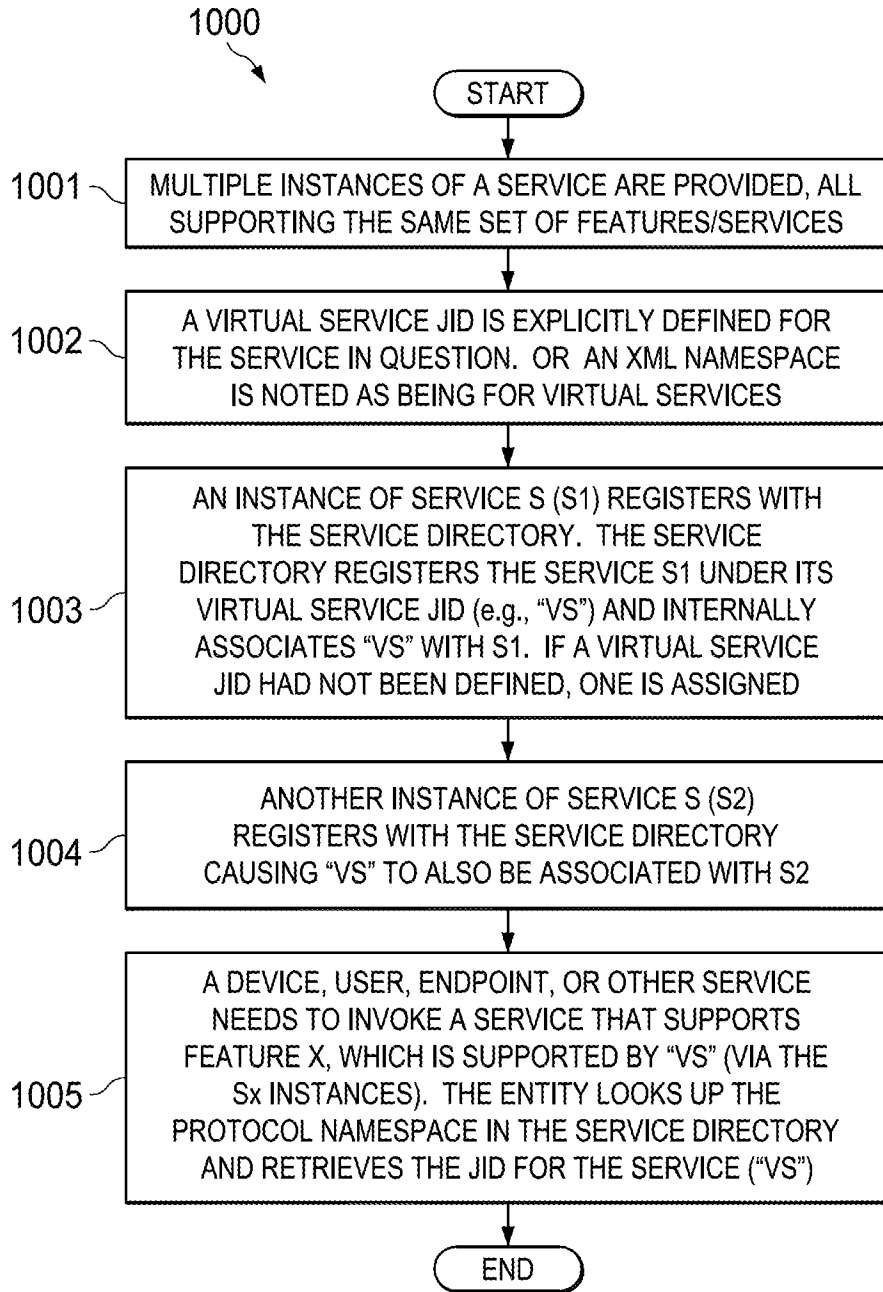


FIG. 10

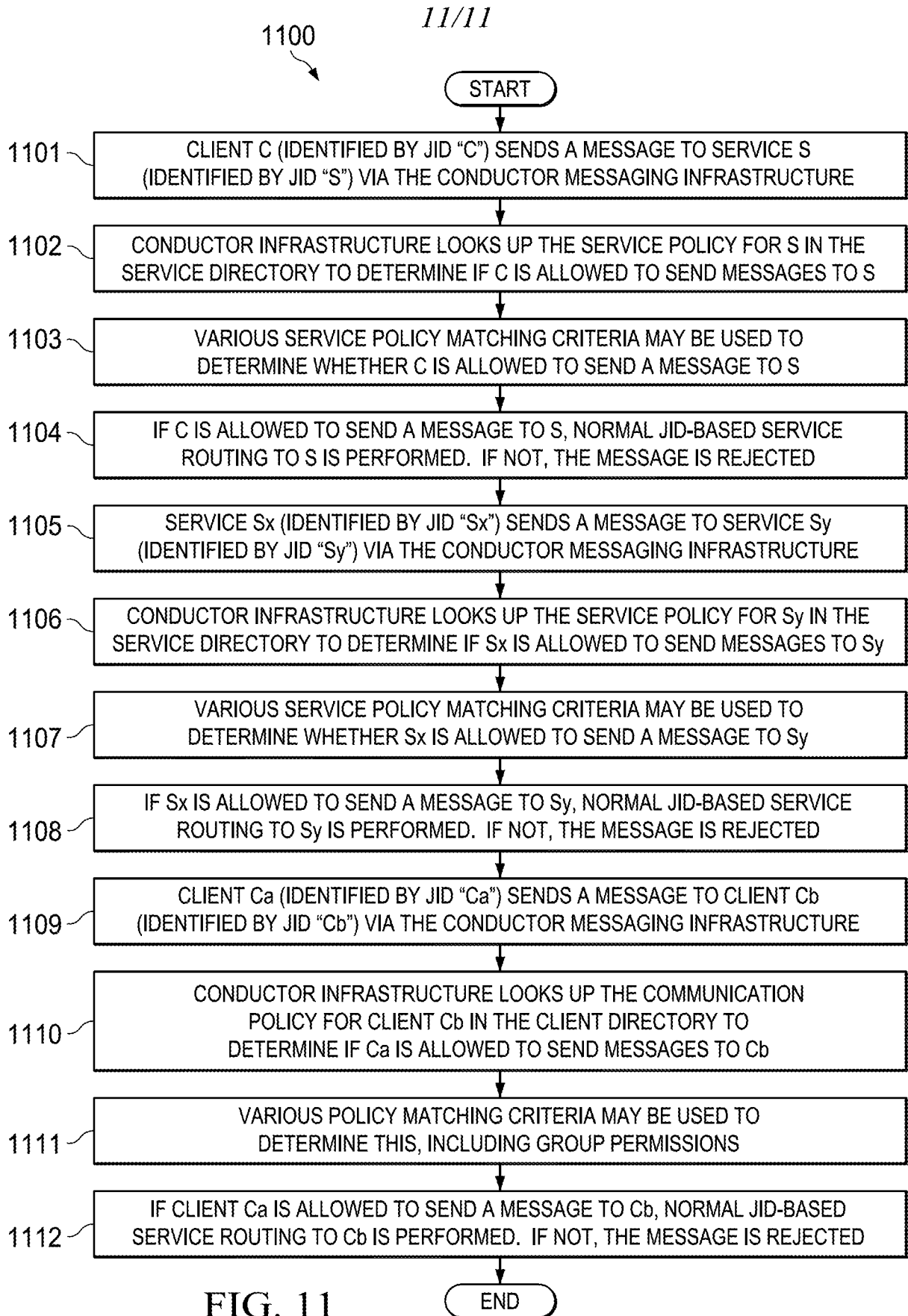


FIG. 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/045858

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/08 H04L29/12
ADD.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
H04L

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Lübke R: "Ein Framework zur Entwicklung mobiler Social Software auf Basis von Android", 15 March 2011 (2011-03-15), pages (i-v)1-115, XP002684306, Retrieved from the Internet: URL:http://www.rn.inf.tu-dresden.de/uploads/Studentische_Arbeiten/Diplomarbeit_L%C3%BCbke_Robert.pdf [retrieved on 2012-09-27] paragraph [04.1] paragraph [4.1.3] page 24, line 4 - line 35 paragraph [05.3] paragraph [05.4] paragraph [5.4.1] figures 18-25 ----- -/--	1-20

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search 28 September 2012	Date of mailing of the international search report 10/10/2012
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Tous Fajardo, Juan

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/045858

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>DANIEL SCHUSTER ET AL: "Service-based development of mobile real-time collaboration applications for Social Networks", PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS), 2010 8TH IEEE INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 29 March 2010 (2010-03-29), pages 232-237, XP031680036, ISBN: 978-1-4244-6605-4 paragraph [0111] figure 1</p> <p style="text-align: center;">-----</p>	1-20
X	<p>ROBERT LUBKE ET AL: "MobilisGroups: Location-based group formation in Mobile Social Networks", PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS), 2011 IEEE INTERNATIONAL CONFERENCE ON, IEEE, 21 March 2011 (2011-03-21), pages 502-507, XP031865724, DOI: 10.1109/PERCOMW.2011.5766941 ISBN: 978-1-61284-938-6 paragraph [001V] figure 1</p> <p style="text-align: center;">-----</p>	1-20