Media Routes
Cloud Communications Platform

Media Routes Service Delivery

OVERVIEW:

Software Defined Services Infrastructure that enables developers to create a broad range of SIP services via simple scripting. Users can create their own SIP infrastructure components to realize use cases such as:

- Session Border Control for both Access and Peering
- Soft-Switches (Class 4 & Class 5)
- Load Balancing
- Session Management and Mediation for interworking between heterogeneous Unified Communications Systems e.g. removing SIP incompatibilities etc.
- API Mash-ups and service orchestration for integrating third party business and communication applications
- Federation to extend the functionality of Enterprise Communication Systems across geographical boundaries and multiple sites.

Users may also create several individual Enterprise Communication features and applications such as IP-PBX, Campaign Management and Auto-Dialers, Automatic Call Distribution and Call Center applications to name a few.

JAVASCRIPT AND CCXML SCRIPTING

The platform is programmable via two options:

- Combination of CCXML and JavaScript. CCXML is an XML based scripting language standardized by W3C that is extremely easy to learn. CCXML allows JavaScript code to be embedded in its XML document pretty much like JavaScript is embedded in HTML documents. Several CCXML based primitives for communications are exposed as service API to the script writers.
- Pure JavaScript using the JavaScript based service API exposed to the script writers.

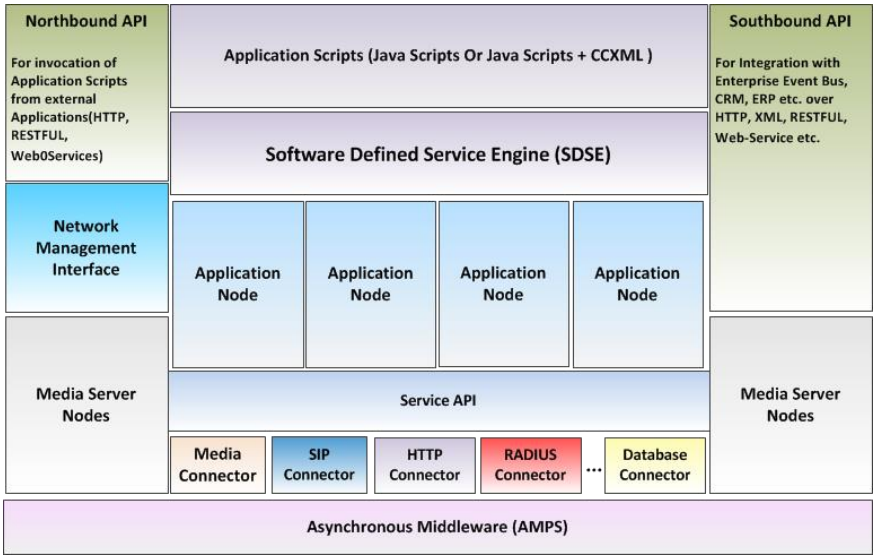
SOFTWARE DEFINED SERVICE ENGINE (SDSE)

SDSE is the central execution engine in the Platform that provides the run-time environment for business logic, service orchestration and user defined call flows.

AMPS MIDDLEWARE

The platform's core is written in ANSI C. It is built on AMPS middleware (www.openamps.org). AMPS is an event-driven middleware supporting asynchronous I/O that facilitates fast development of protocol servers. AMPS has also been developed and owned by Media Routes.

SOFTWARE ARCHITECTURE



PROTOCOLS SUPPORTED

The Platform supports several protocols called ***connectors*** in its architecture that expose their API in the form of CCXML primitives or JavaScript API to programmers. The currently supported protocols include:

- SIP
- RTP
- HTTP (with RESTFUL API support)
- RADIUS (as a client)
- DIAMETER (as a client)
- SMPP
- Database access
- Raw XML over sockets
- Media Handling (MSCML over SIP)

SERVICE API

SDSE exposes a rich set of API (Application Programming Interfaces) for application developers. These are primitive operations for different tasks that a typical application may require. Examples include:

- Creating and receiving telephony calls over SIP
- Playing and recording media files with the help of Media Server Nodes
- Collecting DTMF digits with the help of Media Server Nodes
- Sending and receiving SMS over SMPP
- Creating Conferences
- Authentication Authorization and Accounting over RADIUS and DIAMETER
- Access database and performing direct Stored Procedure calls
- Accessing Web-services over HTTP using RESTFUL API and other methods such as calling server side scripts.
- Sending and receiving XML packets directly over sockets

MEDIA HANDLING SERVER

The platform includes a Media Server that handles Media related tasks and is separated from the main scripting engine. Media Server has a corresponding Connector available to Application scripts running in SDSE nodes that exposes its own set of service APIs for media handling. Media Connector communicates with Media Server using standard SIP protocol with a standardized XML based language called MSCML (Media Server Control Markup Language) embedded in the body of SIP INFO messages. As a result of separate Media serving nodes, Media handling (which is typically more CPU intensive than signaling) can be scaled and distributed across multiple CPU cores, multiple physical servers, and even geographically distributed servers in isolation from the processes running application scripts. Media handling involves all media related tasks such as:

- Media Relaying, Proxy and Trans-Coding
- Multi-party Conference handling
- Media Recording
- DTMF collection
- Streaming of Pre-stored Media files for use cases such as Ring-back tones and Dial tones, Announcements and IVR prompts and menus

SCALABILITY

The platform scales seamlessly by adding hardware resources since it internal architecture is composed of individually scalable modules running as processes with built-in software load balancer. The internal architecture supports a highly concurrent system that is able to handle thousands of concurrent sessions for very large scale deployments.