SEMS: The SIP Express Media Server

FRAFOS GmbH

Introduction

The SIP Express Media Server (SEMS) is a VoIP media and application platform for SIP based VoIP services. SEMS offers a wide selection of media services commonly found in VoIP networks such as announcement services, conferencing, Interactive Voice Response (IVR) services, voice-mail, click-to-dial and other voice messaging and recording services.

SEMS General Architecture

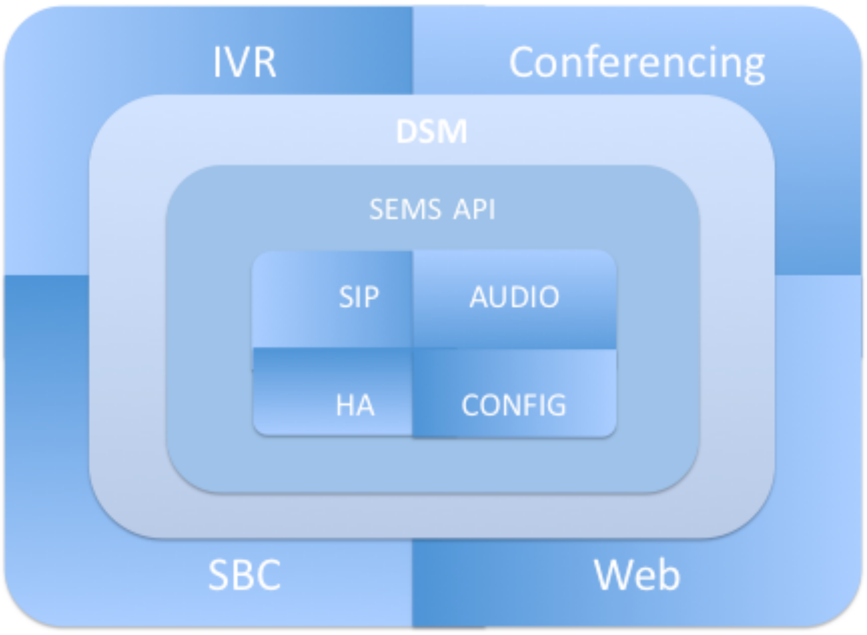

Figure 1 SEMS general architecture

On the one hand SIP media server platforms are expected to support a set of standard applications such as conferencing, voicemail and interactive voice response (IVR). On the other hand SIP media server platforms should provide developers with the necessary tools to easily innovate and implement new applications.

SEMS was designed as a general platform for implementing SIP-based media services and applications. To cover the basic needs of service providers and enterprises SEMS already incorporates a number of applications such as conferencing and IVR. To enable the development of new services and the customization of available services to the exact needs of our customers, SEMS incorporates flexible and open APIs.

As depicted in Figure 1 the SEMS architecture can be roughly divided into three parts;

namely core components, application programming interfaces and applications.

SEMS is implemented in the C++ and C programming languages and is designed as modular system. The core functionality (thread handling, event queues, processing of SIP signaling, media/RTP processing, file I/O) can be extended by different types of modules. Most important, CODECs and file formats can be implemented in audio modules, and specific applications on top of the core's API in application modules.

Core SEMS Components

The SEMS core includes the modules responsible for handling the audio as well as signaling messages. Aspects of high availability and general configuration and management are integrated into the core components.

Audio Processing

- The audio processing components handle the coding and decoding of audio signals.
- Currently SEMS is capable of working with G711u/a, G726, GSM, iLBC, L16,G722, Speex; on request: G729a, G729a/b, AMR.
- SEMS is capable of transcoding audio data from one format to the other.
- The audio processing architecture is designed in a modular manner enabling the easy introduction of new codecs.
- The audio processing modules include different algorithms for jitter handling, loss compensation, audio mixing and playout delay reduction

SIP Processing

- Compliant to RFC 3261
- Provide user agent client functionality
- Provide user agent server functionality
- Support third party call control

High Availability

- Active standby or cluster configuration
- State replication between active and standby servers so as to ensure a seamless

failover

- Seamless failover for signaling and media with no call interruption

SEMS Application Programming Interfaces

The SEMS core provides certain modules that provide audio and SIP processing, memory allocation and management, networking interfaces and so on. These modules can be reused and combined to develop new applications.

To have access to these modules and develop new application logic SEMS offers developers two types of APIs. The SEMS API is a low level API that provides the developers with a high level of flexibility but requires a solid understanding of SEMS and a good command of the C++ programming language. The Domain Specific Modeling (DSM) language provides an abstraction level of the details of the SEMS API and enables developers to implement new application logic based on already available components in a shorter time.

The SEMS API

The SEMS API is based on the event handler model. The application logic processing in SEMS is completely asynchronous, using call specific event queues, and independent threads per call, which optionally may be pooled. The media control and the media processing is synchronous, and the media processing is separated in another thread, so that the application logic processing is not blocked by the media processing. Likewise, all media control functions are implemented so as not to block the application/signaling thread for significant amounts of time. This thread and synchronous/asynchronous model has the properties of

- the consequent internal separation into signaling and media processing realms and the event based structure simplifies application development by keeping the application away from dealing with hard concurrency problems
- all application processing is essentially sequentialized, thus minimizing the need for locking
- most signaling/application threads are sleeping most of the time, and modern schedulers are able to cope with scheduling this type of load very effectively and

- multi-processor and multi-core systems can be utilized.

By using inheritance and polymorphism provided with the C++ language, applications can be written on different levels of abstraction from the underlying signaling and media infrastructure. This is achieved by default implementations of the event handlers, which can be selectively overwritten. For example, an application that only needs to execute application logic when a call starts (i.e. the media session starts) and ends can implement only the onSessionStart and the onBye event handlers and use the other event handlers of the default implementation. An application which needs to tie into the very details of the signaling can implement all signaling message event handlers, or for example the onSipRequest and onSipReply event handlers.

Every call in SEMS has an media input (source) and an output (sink), to which media processing elements can be attached via the API. Implemented in the SEMS core are basic media processing elements (audio file I/O, ring tone generation, conference connector, echo), and elements which can contain or mix basic audio elements (audio playlist, mixer, delay line) and audio filters. Audio elements can also be implemented by the application module. The execution context for all media processing is the media processing thread, and it is synchronous. Control of the media processing is done by locking the audio elements of a call and synchronous actions to change the audio chain; feedback to the application logic is implemented through event passing.

There are other useful building blocks for applications in the core (thread handling, garbage collector, SIP header parser) and also in component modules, which can extend the system's functionality through an internal, dynamically typed interfacing system. These components implement functionality as an in-memory database mainly used for monitoring, SIP registration client, RPC methods (XMLRPC and json-rpc), SIP UAC authentication, SIP session timers.

The Domain Specific Modeling Language

The DSM language is a domain specific modeling language, a dialect of statecharts represented in a textual format specifically designed for the implementation of multimedia services in an application server.

In contrast to simpler state transition diagram types like Moore state diagrams, their expressiveness and thus applicability to real world problems is increased by several measures.

First, short running processes, so called actions, can be bound to transitions and entry or exit of a state, which introduces flexibility in the way processing can be defined.

Second, through the introduction of hierarchical states, a system can be examined and modified at varying detail and abstraction levels, and common transitions can be bound together.

Third, concurrency can be handled by AND decomposition (or orthogonal states), which greatly reduces the complexity of implementation, as it prevents the so called state explosion which can with simpler state transition diagrams especially occur when handling concurrent interactions with other systems or distinct protocols in the same entity.

In order to support a high level of flexibility in the specification and implementation of telecommunication services and keep the implementation complexity at a sustainable level the DSM language was designed with the following goals in mind:

- **Readability**: The DSM language was designed to be as close as possible in structure to graphical call flow diagrams which are commonly used in the design of value-added services. A simple, human readable textual representation for the DSM scripts, and a scripted language (as opposed to a compiled language) is used.
- **Granularity**: As with the native C++ event handler API, it is possible to write applications using the media server on different abstraction levels. It is possible to write applications on higher abstraction levels with simplicity, while applications with fine control of the signaling can use lower level API functionality.
- **Extensibility**: The DSM language has a small core , is modular and easily extensible.
- **Manageability**: To achieve correct behavior also in the presence of internal and external failures, error handling is explicit, transparent and correct.
- **Performance**: The performance of an application server using DSM is not sacrificed when using the DSM system.

SEMS Applications

As part of the SEMS platform a number of ready applications are already implemented. These applications are implemented using the SEMS APIs and can be extended and customized by users of SEMS.

Multi-Party Conferencing

SEMS provides a scalable and efficient multi-party conferencing solution supporting bridging users that might be using different audio codes. Some of the features supported by the SEMS conferencing solution:

- Minimizing mouth-to-ear delay also in presence of jitter by advanced TD-WSOLA supported buffering
- Low Operational costs
  - Up to 5000 channels (G711) on current servers (Sun Fire X4170)
  - Conference bridge core enabler component for services
- Interfacing through REST, json-rpc, XMLRPC APIs

The conferencing solution can be used for dial-in and dial-out scenarios as well as be integrated into other applications using its REST and XMLRPC interfaces.

Interactive Voice Response (IVR)

The SEMS IVR implementation enables enterprises and service providers to interact with their users suing DTMF tones. DSM is used for drafting the logic and interaction menu. The flexibility and power of DSM enables the drawing of complex and lengthy interaction scenarios in a simple and manageable manner.

Announcement Server

SEMS can be configured as an announcement server that plays a certain message at the occurrence of a certain event. The logic for deciding which file to play at which event can be configured using DSM.

SIP Back-To-Back User Agent

The SEMS B2BUA is developed by combining the user agent client and user agent server modules of the SEMS core. Using the SEMS B2BUA it is possible to develop application scenarios such as third-party call control, security firewalls and so on.

SEMS Licensing and Availability

SEMS is available as an open source implementation under the GPL license. Usage of SEMS is free for developers of non-commercial or open-source applications and service providers and enterprises using SEMS to offer certain applications and services.

Developers of commercial applications can acquire a license from FRAFOS which would allow them to include SEMS in their non-open source products.

The target system for SEMS is POSIX environment, for example servers running Debian GNU/Linux(TM), SUN Solaris (TM) or Mac OS X. SEMS builds on top of POSIX threads (pthreads), GNU libc (glibc), Standard Template Library (STL) and BSD sockets. Optionally, specific functionality can be added by using functionality from the libraries spandsp, lame, libspeex, flite, OpenSSL, libev.

About FRAFOS

FRAFOS GmbH is a manufacturer of VoIP solutions with offices in Berlin and Prague. FRAFOS was incorporated as privately held company in May 2010, in Berlin, Germany.

The history of FRAFOS team and technology goes back to the late nineties. As researchers at the prestigious German public R&D institute Fraunhofer FOKUS, the FRAFOS founders were the among the first to work the SIP and RTP standards and to develop open source solutions that paved the way for the VoIP revolution.

FRAFOS offers SIP session management and security solutions of the latest generation that come either as a standalone solution or as a cloud ready implementation. The flagship product of FRAFOS, the ABC SBC, offers open interfaces and built in multimedia applications such as recording and announcements. The ABC SBC enables the operators to simplify their service infrastructure and prepares them for future challenges.