COMMETREX

Open Telecommunications Framework

Open Telecommunications Framework (OTF) is a hardware-independent system kernel that complies with ECTF S.100 and provides the core S.100 system services plus Commetrex's OTF Addressing Framework in the Windows NT environment. As an open-architecture S.100-compliant kernel, OTF can be extended by the OEM to support multi-vendor system resources and client APIs. Client APIs can be either S.100 compliant or proprietary. With OTF, an open-telecommunications OEM can assume full control of its strategic platform in every dimension, while the development of complex inter-vendor switching applications is fully supported over the H.100 bus and its OTF extensions.

OTF adds a comprehensive, open-architecture signaling protocol to H.100 that allows the OEM to easily implement complex switching functionality, even when system resources are furnished by different vendors, finally opening high-end switching applications to the CT OEM. The OTF Addressing Framework adds the signaling protocols and switch-fabric control necessary for full inter-vendor inter-operability of system resources. OTF finally brings "programmable switching" to computer-telephony.

Industry-standard PCM highways, such as H.100, do not support anonymous inter-vendor signaling because no standard has yet to specify the protocols needed to support inter-vendor call setup. OTF removes that impediment by including a comprehensive open-architecture signaling system which finally supports the economic development of complex switching applications such as PBX, ACD, and enhanced-services platforms. The signaling protocol is complemented by OTF's implementation of the S.100 Call Channel Resource which offers the connections set up by the OTF connection-management facility as CCR objects to the system's S.100 Group Manager.

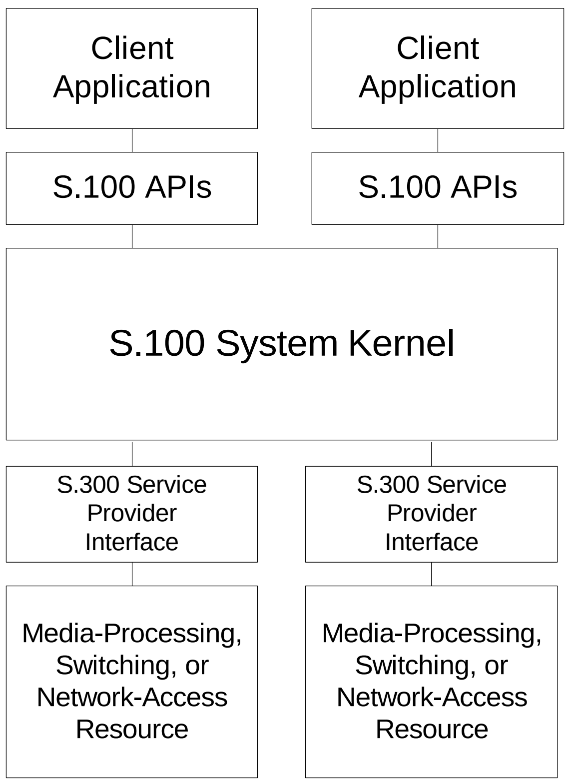
Features

- S.100 kernel
- Group Manager
- Connection Manager
- Session/Event Manager
- Container Manager
- System Call Router
- OTF Addressing Framework
- Open "everywhere"
- Vendor independent
- Hardware independent
- OTF Call Control interface
- Windows NT

Benefits

- Avoid captive technologies
- Anonymous inter-vendor switching
- Control of product strategy
- Lower cost
- Highest system quality by choosing "best-of-breed" system resources
- Low-cost development of complex switching systems
- Economically develop programmable-switch applications: PBX, ACD, enhanced services platform

As a modular building block, OTF does not include the S.100 "resource APIs" or any vendor-specific "Service Provider Interfaces" (SPIs), making it truly independent. It is the responsibility of the system-resource vendor to provide the software, which exposes the resource's functionality to the OTF's SPI. S.100 resource APIs for client applications may also be furnished by independent developers. Commetrex provides SPIs for the system-resource functions it offers on the MSP/CX, such as MultiFax for MSP/CX via an MSP Consortium M.100-compliant software environment.



OTF is the first product to support the economic development of standards-based open-telecommunications systems that are truly vendor independent. Prior to OTF, all computer-telephony software systems were tied to the vendor's hardware products, and they were only "open" at the client-API level. Moreover, the vendor offered no way to remove the bundled-in system-resource or hardware component and substitute another. With OTF, any developer is free to add any system resource and any client-side resource API, where the API can be either S.100-compliant or otherwise.

The Computer-Telephony Middleware Challenge

Where does the CT OEM get the middleware, or the software framework, for system development? This is the software that manages system resources, handles call control and routing, system configuration and security, and so on. The answer, of course, is either make or buy. If the OEM develops the CT framework, he is buying into not just an unending development effort, but perhaps an insurmountable task of developing proprietary media-processing system resources.

In the simpler times of single-media single-PC systems this was feasible. But with today's reality of extensible client-server integrated-media systems, the investment in capital and time-to-market makes this an unjustifiable investment.

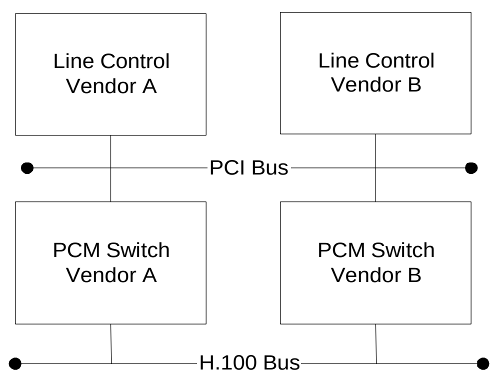
But does purchasing the CT system framework mean the OEM must relinquish control of his product strategy? Usually it does. Other than OTF, there are no modular, industry-standard, hardware-independent CT system software systems. All are sold with integrated support for the vendors traditional media-processing resource boards: voice boards for Dialogic and NMS; fax boards for Brooktrout and Gammalink. Using a third-party voice board with Dialogic's CT Media Access or NMS's CT Access isn't an option. Why not? They are captive technologies. Voice and call control, parameter and resource management are considered the base-level functions. Additional media, if available from the primary vendor are add-on functions.

The System Resource Challenge

So if the software framework is bundled with a media-processing or network-access product, must every "technology-resource" vendor develop a CT middleware product? How can the developer of a single CT system resource, such as voice, data, text-to-speech, or voice codecs, offer the system developer a multi-media system platform?

Without OTF there are few answers, and none are very attractive alternatives. One would be to form a technology partnership with a company that already has a software platform. This approach has inherent addressable-market limitations and gives up marketing control to the partner. The second alternative is to develop an integrated-media client-server software system, either proprietary or based on S.100, the only specification available. This is a large investment which will only pay off if third-party media-processing resource vendors offer compatible products.

But OTF offers an alternative with low investment hurdles, the broadest addressable markets, and strategic control perhaps greater than in-house development, which is often severely constrained by investment and time-to-market obstacles.



Traditional Switch Model

The S.100 specification has two of the critical features needed to support the development of a modern, integrated-media computer-telephony system: client-server and vendor independence. But, S.100 does not include a capability essential to bring programmable switching to computer-telephony: there is no support for multi-vendor signaling, and there is no support for system-wide control of switching. This means S.100 provides no support for connecting a call received by a network interface from vendor A that should be connected to a station interface from vendor B.

Inside S.100

S.100 is a strong foundation on which an effective client-server CT system can be built. The meat of S.100 is the kernel services and their client APIs, the client APIs for “technology resources”, and interfaces to the telephony service providers (S.300). There is one product available today, CT Media Services from Dialogic, which is based on S.100 (perhaps, visa versa). It includes the technology-resource APIs. And, perhaps more important, it also includes the Dialogic voice-board products as fully integrated resources. So, although it may be based on S.100, it’s hardly vendor-neutral and open. (Try to substitute a different voice board.)

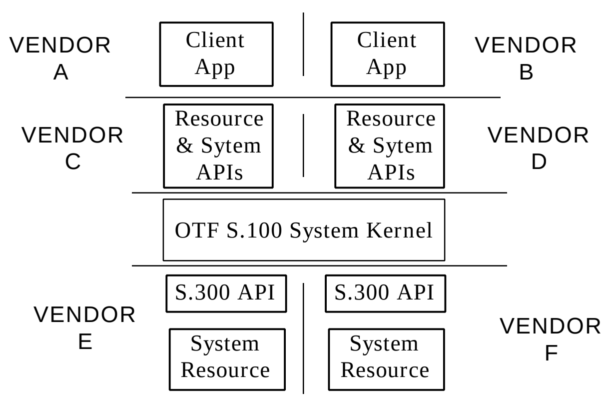
Another issue has to do with the availability (or lack thereof) of S.300. It’s not yet available, and until it is made available by the ECTF any ability to achieve vendor-independence at the resource-board level has to be somewhat suspect. One approach is to furnish a temporary S.300 substitute and replace it with the S.300 interface when it’s available.

Another interface needed for a truly modular system is between the client-level technology-resource APIs and the system kernel. Here, the ECTF has published the S.200 protocol interface. It separates the client API from the kernel while it supports remoting the client API through a telecommunications transport such as TCP/IP.

S.100 Connection Management

But S.100 suffers from the distributed-switching model of all the widely accepted PCM highways (MVIP, H.100, etc.) in its ability to deliver a robust and comprehensive connection-management facility. Because each vendor’s resource board may contain a switch, each board represents a potential source of an unintended switch setting, as shown on the previous page This arrangement presents the need for a hierarchy of S.100 System Call Routers (SCRs). The SCR, an optional S.100 system element, shields applications from the S.100 Connection and Group Mangers. It normalizes vendor-specific call-control to S.100 functions, such as answer call, make call, and disconnect call. But since S.100 does not define a signaling protocol, the process of connecting a call from the network-access interface of Vendor A, for example, to the station interface of Vendor C is extremely complex. Either a rather complex application must be developed which is vendor aware and uses the S.100 Connection and Group Managers, or the SCR must be used and its functional limitations accepted.

It's gets even more complicated: Before a media stream can be presented to an application it must be presented to the S.100 Group Manager as a Call Channel Resource (CCR) through an undefined interface. So there are additional interfaces to each vendor-specific line control entity.



The solution to this complicated puzzle is to create a CCR proxy and define a protocol for each resource to communicate with the proxy. Think of the proxy as a central office and the resource as terminal equipment or a subscriber line. The protocol, then, is a telephony signaling protocol. This notion can be extended to any OTF addressable entity, be it a host-based application or a resource-board-based application. If the application does not reside on the board, or entirely on the board, a host-resident proxy can do the job, as shown below.

In the traditional model, the switch configuration is controlled through vendor-specific switch control and signaling. For example, a line interface from one vendor may use channel-associated signaling while another uses a host-based call-control protocol. In a basic S.100 system each vendor presents significant complications, forcing the OEM towards a single-vendor approach.

But what if the switch fabric were controlled by one connection manger in a manner similar to a PBX? From the system's perspective, calls would originate at the trunk interface. Call setup would then occur between the interface (or its host-based proxy) and the OTF Connection Manager. Once the connection establishment has proceeded to the "call proceeding" state, the media stream is made available to the S.100 kernel as a Call Channel Resource (CCR).

OTF furnishes a host-based signaling proxy and API to simply development of OTF-compatible resources. But a far better approach, and one slated for future releases of OTF, is for the intelligent line card to host the protocol engine and signal the OTF Connection Manager through the H.100 pin reserved for the purpose.

OTF System Services

OTF includes the system services defined in S.100 that constitute the core services of the system. They include

- Session/Event Manager
- Group Manager
- Connection Manager
- Container Manager
- System Call Router

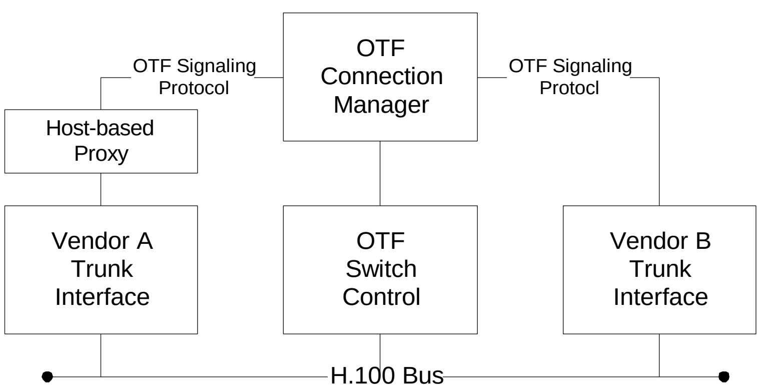
Session/Event Management

The OTF Session Manger is used by all OTF-Addressable Entities (OAEs) to register with the OTF Connection Manager (CM). Once registered, an OAE can be communicated with by any other OAE, either explicitly by name or indirectly through group names or implicitly by a resource request. Once registered, all OAEs have a “nailed-up” signaling connection with the CM to support command-event signaling.

Group Manager

An S.100 “Group” is an object that presents a unified interface for allocation, configuration, interconnection, and hand-off between applications of the resources needed to perform computer-telephony functions. The resource that represents a media stream is the “Call Channel Resource” (CCR). An example of a group for a play-message function is a CCR, a “player”, and a “Signal Detector” to detect touch tones. Groups may be explicitly reserved by an application or they may be implicitly reserved by invoking a higher-level function.

The system resource used to abstract a switch-fabric connection is the Switch Port (SP). Each Group has an SP as a virtual resource. The Connection Manager allows an application to make connections between SPs.



OTF Signaling

Connection Manager

The Connection Manager and its API give the application an easy-to-use facility to switch calls from one party to another, to monitor two other parties, and to control a Conferencing Resource, if one is available.

Container Manager

“The S.100 Container Manger (and its API) provides an operating system-independent mechanism for the storage and interchange of system data between system services, resources, and the application.” A “container” is an object made up of data, usually media data, and a set of attributes. The S.100 Container Manager extends the functionality usually found in file systems with features required by Resources to manipulate media data in a convenient manner.

System Call Router

The System Call Router (SCR) shields the application from the details of the Connection and Group Managers and the need to be aware of any vendor-specific call control. If provides API functions such as answer, make, and disconnect call, greatly simplifying the routine placement of outbound call and accepting inbound calls in traditional CT applications.

OTF Addressing Framework

All entities—applications and network-connected entities—known to the OTF system are addressable by the system and any other OTF Addressable Entity (OAE) for which permission has been granted. OAEs place and accept calls in a manner similar to a terminal on a network. The network-layer protocol used by OTF is Q.931. It’s the same protocol used in ISDN, H.323, and similar to that in ATM (B-ISDN) and SS-7.

The OTF Developer's Kit includes the OTF Addressing API, and each API that requires it is packaged with it. The developer can use the API to implement OTF signaling support directly on a resource board or in a host-resident proxy. The latter approach allows the OEM to integrate existing boards from any vendor into the OTF Addressing Framework.

Typically, the CM is able to resolve most connection requests by matching the OAE's address with the name of a Group owner listed in its Connection Map. However, there will be situations where the CM has insufficient information to resolve the connection request, such as a "dialed number" in a PBX application. The OTF Addressing Framework provides for such occurrences by allowing the CM to pass the request off to an OAE registered to resolve the connection requests of either a single address or an address group previously defined to the CM. These OAEs are known as CM "Controlling Entities".

The CM supports the following types of addressing:

- Explicit
- Address Group (not S.100 Group)
- Dialed Number
- Implicit
- Service Requests

In Explicit Addressing the calling OAE provides the CM with the OTF Address of the called entity. For example, a System Kernel service would use explicit addressing to send an unsolicited event to an application.

The CM treats Address-Group Addressing in a manner similar to Explicit Addressing. (Here, the Address Group is a homogeneous group of OAEs, not the S.100 Group). In Group Addressing, the CM must first resolve the Group Address to a specific OAE Address. This is done by the CM placing a query to the OTF Resource Manager (RM), passing in the Group Name. The RM will then translate the name into an explicit address via tables, hunting algorithms, etc. The resulting session request is then processed as an explicit addressing request.

In Implicit Addressing the calling entity furnishes no "to" address at all. If a Controlling Process is registered for that calling entity the CM will simply pass the request to there, otherwise. If no Controlling Entity is registered the request will be denied.

Service Requests differ from the other session requests because the calling OAE does not specify in any way the address of the called party. Instead, it requests some service from the network (the OTF Kernel). This is the mechanism used to implement the S.100 Resource APIs.

OTF includes a facility similar to an Intelligent Network Service Control Point (SCP). The CM, upon detecting the Service Request will communicate the request to the OTF SCP which will attempt to locate a service provider within the OTF system. The SCP will then return the service provider's OTF Address to the CM, which is then able to complete the call.

Commetrex Corporation
6400 Atlantic Blvd.,Suite 190
Norcross, GA 30071
Voice: (770) 449-7775, Fax: (770) 242-7353
http://www.commetrex.com
e-mail: marketing@commetrex.com