

Open Multimedia Platform framework

Continuous changes in business environments as well as the convergence of media, entertainment, and communication businesses and solutions require a new approach to system design, pricing, product packaging, deployment, and support.

BO ANDRÉN

Technologies in the IT industry are evolving to the point where they provide telecom-grade levels of availability without the need for extensive in-house solutions. For select network segments, operators can now attain this level of availability via the network solution and not just in individual nodes.

Where multimedia applications are concerned, Ericsson is leveraging the evolution of basic technologies and adding value in the application domain.

Ericsson promotes and embraces the evolution of broad industry technologies and advocates open standards for designing scalable, reliable, and efficient systems. It does so by being an early adopter, an active participant in standardization, and by working with the supply chain.

Platforms come in many shapes and sizes. Where telecommunications access and core networks are concerned, they must be clearly defined in terms of both content and characteristics. This means that the choices of technology and configurations must be verified in a controlled environment.

The concept of multimedia applications spans such a broad range of services that one cannot adequately define a multimedia platform for all applications. Accordingly, the framework described in this article is not a traditional telecommunications platform; instead, it

- ✦ describes a generic execution environment for applications that adhere to open standards; and
- ✦ enables constant additions to, as well as modifications or substitutions of, its component parts.

Traditionally, telecommunications networks have been built as a hierarchy of integrated vertical solutions that guarantee compliance with stringent requirements, among other things, for

BOX B Service Availability Forum

The SA Forum is a consortium of companies from the telecom and computing industries that works to develop and publish high-availability and management software specifications. Its main goal is to develop an ecosystem that enables the use of off-the-shelf building blocks for high-availability systems.

availability and retainability. Meeting these requirements has often called for proprietary solutions, from the smallest screw to processors for the middleware on which applications rely. As telecommunications has evolved, user demands for more features, more speed, more throughput, and faster change have given rise to a horizontally layered network. This evolution has also been accompanied by wider use of standard, mainstream components.

Today, we face even greater change as network architecture hierarchies are being flattened and the signaling infrastructure moves to all-IP. Accompanying these changes are pooled resources, a clear separation of client and server gateways, and the ability to build networks that more or less fulfill availability requirements through network redundancy. This, in turn, helps to relax the requirements put on individual network elements, which, of course, affects how service providers approach vendors.

Some nodes must still always adhere to strict definitions (specifically, nodes of the call-handling type, which employ a distributed-state model to restore calls when a state is lost due to failure). However, growing demand for swift new features in the multimedia and service parts of the network have led to the use of standard, mainstream solutions in both the execution and development environments.

Standardization

In anticipation of the evolution described above, many players in the IT and telecom industries initiated a variety of layered, open-standardization activities at the opening of the new

BOX A Terms and abbreviations

AIS	Application interface specification	NTF	Notification service
AMF	Availability management framework	OAM	Operations, administration, and maintenance
CKPT	Checkpoint service	OMP	Open multimedia platform framework
EJB	Enterprise JavaBeans	PM	Performance management
IETF	Internet Engineering Task Force	SAF	Service Availability Forum
IMM	Information model management	SGCS	Sun GlassFish communication server
IMS	IP Multimedia Subsystem	SIP	Session initiation protocol
JCP	Java Community Process	SNMP	Simple network management protocol
JEE	Java Platform, Enterprise Edition	SSH	Secure shell
JSR	Java Standardizations Request	TEM	Telecom equipment manufacturers
MIB	Management information base	VIP	Virtual internet protocol
NEP	Network equipment providers		
NETCONF	Network configuration protocol		

❖ millennium. Some of these efforts, such as Carrier Grade Linux, were also driven as open-source initiatives backed by large companies with the same intent—to enable open, horizontal isolation and exchangeability, and to create a lively and healthy ecosystem. One activity that has prevailed is the Service Availability Forum (SA Forum), whose objective is to define a high-availability middleware environment, primarily for telecom but also for other industry segments.

In 2000, Java had not yet been widely deployed as a programming language in telecom systems. But there was an understanding in the industry that this was the way forward, especially for multimedia and service layer applications. Accordingly, Java has become one of the most important engines in response to the expectations and requirements of a faster-moving universe that anticipates new features at ever-increasing speed.

A major driving force is the ongoing convergence and consolidation of networks. These changes also impel an IP-centric, management-driven approach toward element management, in order to achieve a uniform north-bound interface from the base platform. Typical examples are the network configuration protocol (NETCONF) and its modeling language, Yang, specified by

the Internet Engineering Task Force (IETF). These initiatives stem from the need to centrally manage and configure huge numbers of more or less complex network elements.

Numerous companies were involved in the standardization and specification efforts described above but without any specific coordination. The result was a range of standards or specifications that described many aspects of a basic execution platform but could not be combined into useful implementations. For this reason, most major telecom equipment manufacturers joined forces in 2006 to create the SCOPE Alliance, which

- ❖ guides the industry's use of existing specifications and standards; and
- ❖ drives the requirements and requests for missing functionality toward the specification and standardization bodies.

Other developments in this direction include the establishment of the OpenSAF Foundation, to facilitate the open-source project that is implementing the SA Forum specifications.

Ecosystem

Notwithstanding the many accomplishments of the above-mentioned initiatives, the industry has yet to define and establish a generic implementation of the layered architecture. Besides components, a platform contains many pro-

BOX C The OpenSAF Foundation

was established by leading communications and enterprise computing companies to facilitate the OpenSAF project and to accelerate the adoption of the OpenSAF code base in commercial products. Members include Emerson, Enea, Ericsson, HP, Huawei, Nokia Siemens Networks, Rancore Technologies, Sun Microsystems, and Wind River. OpenSAF is an open-source project established to develop a base platform middleware that, under the LGPLv2.1 license, is consistent with Service Availability Forum (SA Forum) specifications. www.opensaf.org

cedures and ways of working that ultimately define how flexible and usable it is. These procedures and ways of working include the enforcement of strict layered interfaces to ensure replaceability and flexibility, and ways of developing, testing, and deploying applications.

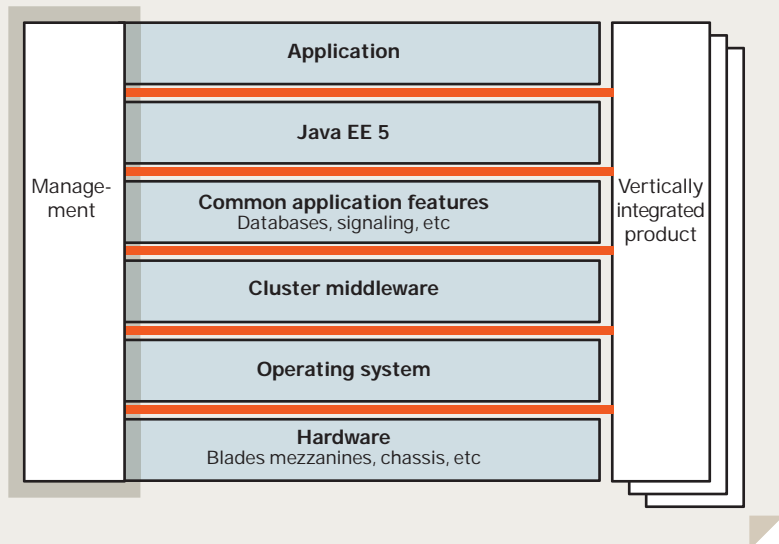
Ericsson has thus defined and created the Open Multimedia Platform framework (OMP) in order to embrace an existing and evolving ecosystem that provides the necessary layered components. In addition, it sends a forceful message to suppliers, users, and purveyors of third-party products: this is the environment that will shape the service and multimedia layer. The industry should design toward these goals. Doing so will ensure a diversity of tools and components and create the vibrant ecosystem and efficient supply chain we all crave.

The main functional layers of OMP comprise a software applications execution framework. This is complemented by application services, such as a database management system, protocol stacks, and directories, for implementing and deploying solutions.

OMP is intended to embrace the largest-possible ecosystem, which is why adapting to standards is crucial. It includes the Java Platform Enterprise Edition (JEE) execution environment and all major layers in the stack. Applications need not make any assumptions about hardware or operating systems at design. In today's market, no solution fully addresses telecom requirements from the telecom industry and offers a diverse ecosystem of independent hardware and software providers.

The framework allows for an application to be configured with the necessary distribution, availability, and supporting services characteristics. Over time, services will become available that can fully use the service of the availability layer. But an application deployed on the high-availability JEE can also use services that are deployed outside the cluster itself and still achieve necessary characteristics. In other words, the use of OMP is not limited to solutions that use all the included components. Usage is based on the scalability, availability and performance requirements of the specific solution.

FIGURE 1 Layers of the Open Multimedia Platform framework (OMP) architecture.



Open Multimedia Platform framework

OMP is meant to serve as a model for building server nodes with open interfaces and standard components without sacrificing service availability and low cost of operation. The prime target is to support the design of portable JEE applications and to align the northbound management interface with operations support systems.

The horizontal layering in OMP architecture is coupled with the basic elements of any application — the hardware, operating systems, middleware, and application code. The ability to combine third-party software with proprietary software is crucial in terms of time to market.

The layers of OMP (Figure 1) consist of

- ❖ a third-generation application server;
- ❖ standardized middleware providing high availability;
- ❖ an operating system;
- ❖ any hardware system; and
- ❖ management components that provide the generic management features to the application server and the applications.

The interfaces between the layers are based on existing and evolving standards or specifications. The components in each layer are expected to comply with these standards. The separation of layers via well-defined interfaces is a fundamental requirement for achieving replaceability (of components).

Applications based on this architecture use selected framework components to achieve the required characteristics. The integration and dimensioning is part of the application design. In other words, there are no limitations or restrictions regarding how the architecture compiles, selects or applies components. Moreover, applications need not use all layers.

Figure 2 shows the current selection of components.

The application server layer consists of the third-generation application server (Sun Glassfish Communication Server, SGCS), which includes the SIP container contributed by Ericsson via the open-source SailFin project. Integration with the SAF middleware provides OAM for applications as well as for the application server. Finally, reusable components, such as licensing, and perfor-

BOX D SCOPE Alliance

is a generic requirement organization led by network equipment providers to drive open standards/source/ecosystems in order to supply more suitable free and open-source software and commercial off-the-shelf solutions for an open carrier-grade base platform (CGBP). See SCOPE's technical position paper for a more thorough description of the CGBP. www.scope-alliance.org

mance management (PM) features, are added in the JEE domain. These additions are examples of differentiators and are the main focus of in-house OMP design.

To guarantee portability to any application server, all applications that run on top of the application server should use only standardized interfaces.

Ericsson uses SAF middleware (based on the open-source OpenSAF project) in the cluster middleware layer. As an optional service for load balancing, this may be complemented with virtual internet protocol (VIP).

The operating system is an enterprise distribution of Linux.

The OAM components use a model-driven approach to provide an Ericsson-aligned northbound interface (NBI).

The software and hardware in the OMP are decoupled, which for instance means that operation, administration and maintenance of hardware and software are handled separately. It also means that it is possible to use a wide range of hardware.

The application server

The application server layer integrates a Sun GlassFish communication server with features that are needed to align the operation, administration, main-

tenance, and provisioning of the server itself and applications using the server.

The application development environment is Java EE. The integration of SGCS brings all the positive, standardized aspects of a JEE development environment to Ericsson's community of application developers.

The SGCS uses the same successful programming model for SIP as for HTTP, the servlet. SIP, which is needed for more advanced multimedia services, is thus available to a huge developer community outside Ericsson, which will nurture the growth of an ecosystem.

Thanks to the use of Enterprise JavaBeans (EJB), the model and enterprise business logic are part of the application server, allowing applications to combine business logic and communication needs in the same process in memory (previously, the performance of HTTP- or SIP-intense applications that used EJB to solve business needs was quite poor).

JEE application developers need no knowledge of the SAF middleware implementation, because all necessary middleware services are made available through standard Java interfaces. The JEE software development kit (SDK) encompasses the Java SDK from Sun, all additional APIs, and example appli- ❖

FIGURE 2 Components in the Open Multimedia Platform framework (OMP) architecture.

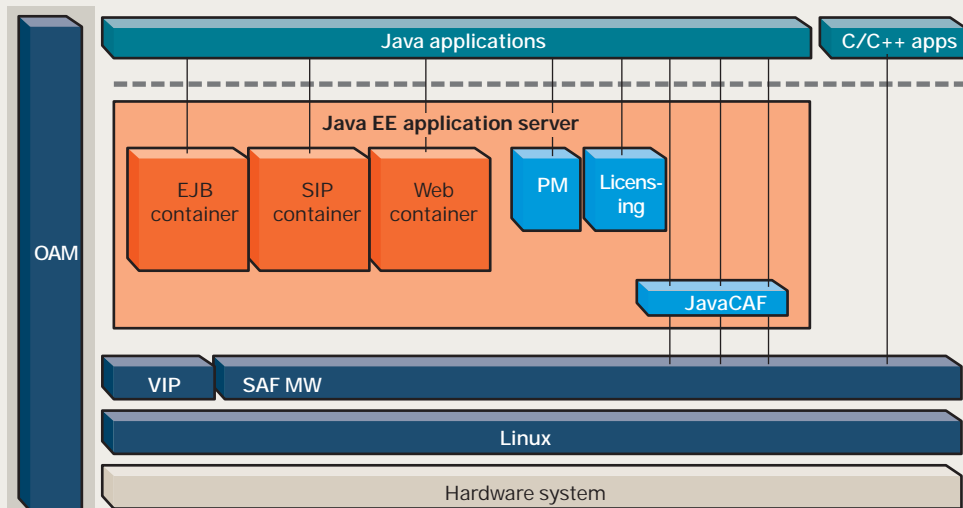
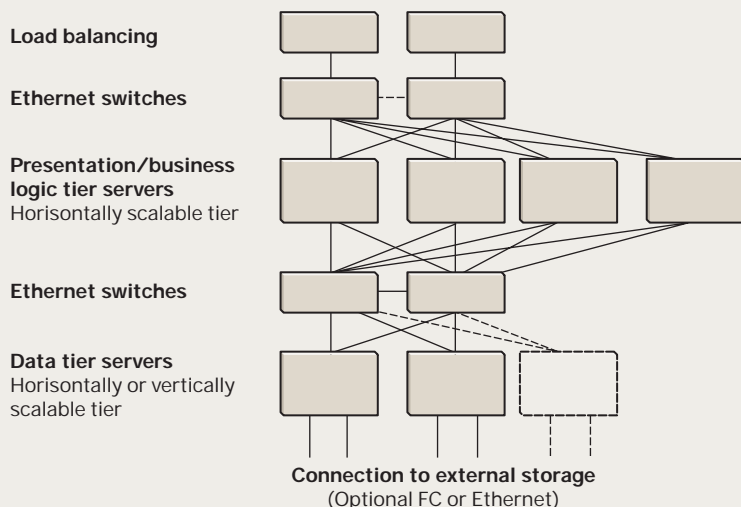


FIGURE 3 Overview of the hardware architecture, server part.



❖ cations. The development environment is Eclipse. In addition to the application server, the Service Development Studio (available through Ericsson Mobility World) emulates a complete IMS network including a mobile terminal.

The adaptation to SAF middleware adds functions that are particular to a telecom-type application environment and not otherwise available in the community. Over time, more and more of these features will be migrated into the application server.

Important new specifications from the Java community are JSR 289 (SSA 1.1), JSR 281 (IMS Services API), JSR 319 (Availability Management), and JSR 313 (JEE version 6).

Middleware

SAF middleware is a component that provides interfaces that comply with the standardized SA Forum application interface specification (AIS). These are needed for developing high-availability applications in a clustered system. The most important services provided by SAF middleware are:

- ❖ availability management framework (AMF),
 - ❖ checkpoint service (CKPT),
 - ❖ information model management (IMM), and
 - ❖ notification service (NTF).
- The availability management frame-

work coordinates redundant resources within a cluster and specifies an abstract system model to represent resources under its control. An application that is managed by the AMF is structured into logical entities according to the system model specified by the AMF specification.

The checkpoint service manages a set of checkpoints that allow a process to save its state to minimize the impact of failure.

The information model management service provides a base infrastructure to manage various information models. The objects in an information model are provided with their attributes and administrative operations. IMM allows object managers, such as configuration management, to create, access, and manage the objects of the information model.

The notification service reports alarms and alerts to fault management.

The SAF middleware is optional for strictly JEE-based applications, or for applications that put limited requirements on availability and scalability.

OAM

The OAM is a system component that enables centralized or remote operation, administration, maintenance and provisioning of OMP-based systems. It provides a common, approved north-

bound interface to the operations support system. OAM provides configuration management (CM), fault management (FM), and software management (SWM) capabilities.

Configuration management is a system function for administering the configuration data of a system. Operators access configuration management functionality via a northbound interface using the NETCONF protocol (machine-machine interface) or a command line interface (man-machine interface).

Fault management is the system function that allows operators to detect and act on faults in a system. The function uses the simple network management protocol (SNMP) to deliver alarms to the OSS. It receives alarms generated by applications or OMP layers via the NTF service.

Software management is a system function that provides support for installing or upgrading a system. It supports a rolling upgrade method without system downtime, and a one-step upgrade with system downtime.

OAM includes tools and documentation support for the vital areas of modeling, installation, backup or restore, and software upgrade.

OAM facilitates a homogenous and centralized handling of models, making the customer product documentation and the model-driven command line interface and NETCONF interface reliable and consistent. By using the modeling tool, it is possible to create a management model of the application that can be loaded into the system as part of an installation or an upgrade.

Hardware architecture

Applications on the Open Multimedia Platform framework are not aware of or designed for the hardware on which they run when deployed. The actual choice of hardware can thus be made late in the process. A common hardware architecture has been defined to ensure that the OMP components and applications can run with proper characteristics and without putting unnecessarily diverse or odd requirements on the hardware.

The architecture describes common principles and guidelines for building the software part in order to make it target a hardware setup that can cater for

these characteristics. This does not tie the software to specific (or even physical) hardware. Instead, it gives guidelines on how applications should be built in order to arrive at a common and sensible view on the underlying hardware.

This approach makes it unnecessary to design multiple solutions to the same problem. Instead, one can reuse others' solutions as well as co-deploy and share resources.

Figures 3 shows the principal hardware architecture. A tiered approach promotes but does not absolutely stipulate horizontal scaling as the only option, particularly for the data tier.

The use of a common architecture facilitates efficient setup for providing the actual hardware platform, since the overall structure is known and unified, regardless of the application that runs on it.

Reuse of components

Previous platforms within Ericsson were developed and maintained by large, dedicated platform organizations. Unfortunately, this made it difficult to prioritize application developers' needs. And today, where expectations regarding what the platform should support are even higher, one group simply cannot develop everything that needs to go into the platform. Hence, the Open Multimedia Platform framework.

With OMP, organizations that develop applications can contribute reusable objects (software or plain knowledge) to the software stack or framework. The three main systems in place to support this way of working are

- ❖ software object inventory;
- ❖ community forums; and
- ❖ governance structure.

The software object inventory stores metadata related to software objects. It contains candidates for reusable objects as well as objects that are maintained as reusable objects. It stores third-party software that is brought into application design to prevent duplication of effort. And it includes a search engine that operates on all metadata for the objects in the inventory.

The community forums share knowledge about developing applications on the OMP as well as the actual development of platform components. The

forums also contain discussions concerning what objects should be included in the framework and what ongoing application projects can be used as vehicles for developing them. In addition, the community discusses the software objects that are under development, and whether they should be maintained as reusable objects for application designers or incorporated as reusable components in the framework.

The information system that supports this process has functions both for pushing information to the development community and for collecting functionality from application developers. Overall, the development of the open platform and its applications is gradually evolving into an internal open-source environment.

To foster and govern the process of contributing and reusing items, a council has been established that has the authority to decide how new objects and systems for OMP should be developed and maintained and who should develop and maintain them. ❖

Bo Andréén,



❖ who joined Ericsson in 1979, is the current head of R&D Management, Common Technologies within Business Unit Multimedia. In this position he is responsible for implementing a strategy for basic technologies within the business unit. Bo has held numerous positions in R&D and product management while working with several of Ericsson's telecom platforms. He has also been heavily involved in the standardization of hardware and software technologies.

Trademarks

❖ Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.