Department of Computer Science

Zhiyang Lu

COTS Component Database for Standardized Telecom Platform Integration

Computer Science
D-level thesis (20p)

Date:	06-09-26
Supervisor:	Donald F. Ross
Examiner:	Thijs Holleboom
Serial Number:	D2006:08

Karlstads universitet 651 88 Karlstad
Tfn 054-700 10 00 Fax 054-700 14 60
Information@kau.se www.kau.se

This dissertation is submitted in partial fulfillment of the requirements for the Master's degree in Computer Science. All material in this report which is not my own work has been identified and no material is included for which a degree has previously been conferred.

Zhiyang Lu

Approved, 2006-09-26

Advisor: Donald F. Ross

Examiner: Thijs Holleboom

Opponent: Katarina Asplund

Abstract

To help modern telecom corporations gain an advantage in the competition with other companies, the standards based platforms built using commercial off-the-shelf (COTS) components represent a new trend within the industry. This new development method increases efficiency, reduces the cost and shortens the time to market. The goal of this dissertation project is to construct a prototype database to hold information about COTS components. This dissertation first presents a general overview of the telecom industry. Several standards issuing organizations and the main content of some of the standards are then presented, such as PICMG[1] (ATCA[2]), OSDL[3] CGL WG[4] (CGL RD[5]), and SAF[6] (HPI[7]&AIS[8]). Five standards structuring organizations (SCOPE[9], PICMG RES[10], CP-TA[11], MVA[12], ITU-T[13] OCAF[14]) are also described. The dissertation then presents a synthesis of the information from the given standards related organizations and analyses the information from the standard compliable COTS components. Finally, the prototype of the COTS component database is defined as a prototype tool for the platform integration department of TietoEnator to actively integrate information about telecom components, platforms, and systems in the future.

[1] PICMG: PCI Industrial Computer Manufacturers Group
[2] ATCA: Advanced Telecom Computing Architecture (AdvancedTCA)
[3] OSDL: Open Source Development Lab
[4] CGL WG: Carrier Grade Linux Working Group
[5] CGL RD: Carrier Grade Linux Requirement Definition
[6] SAF: Service Availability Forum
[7] HPI: Hardware Platform Interface
[8] AIS: Application Interface Specification
[9] SCOPE: SCOPE Alliance
[10] PICMG RES: PICMG Requirements Engineering Subcommittee
[11] CP-TA: Communications Platforms Trade Association
[12] MVA: Mountain View Alliance
[13] ITU-T: International Telecommunication Union-Telecommunication Standardization Sector
[14] OCAF: Open Communication Architecture Forum

Contents

List of Figures

List of tables

Declaration

The work for this project was carried out at TietoEnator, Karlstad under the supervision of Gunnar Lorentzon. The work on the project was carried out by Zhiyang Lu (the author of this document) from Karlstad University and Pär Råstram fro Chalmers Institute of Technology.

Each of the collaborants will present their work as a dissertation submitted to their respective institutions. This dissertation is the version submitted to Karlstad University as part of the requirements towards a Master's degree.

1 Introduction

1.1 Background

To help modern telecom corporations gain an advantage in the competition with the other companies, the standards based platforms which are built using commercial off-the-shelf (COTS) components are encouraged to be used. This increases the efficiency, reduces the cost and shortens the time to market. Procuring, developing and maintaining a system by using the open standards and the COTS components, is very different from the traditional method of using customized components. Under the traditional method, the system producer develops the system independently. They develop their own products, owns all the detailed technology, and bears responsibility for the third party who produces the component for their system. The new open telecom platform solutions changed the roles of the system producers from components developer to be components vendees since the COTS components became an important part in open systems. As the vendees of the COTS components, the system producers are only required to know about the interfaces, the functionalities and the services of the components but not the components' inner structure detailed technologies. [15]

Knowledge of existing standards and COTS components is extremely important for the marketing analyst, the project manager, the vendor representative, the quality assurance manager and the system analyst. Since the information of the open standards specifications and the components datasheets are dispersed on the internet and the format of the data varies, there is an obvious need for a tool to help managing all the information which is related to telecom systems. To build a COTS component database is one of the solutions to help collect, format, analyze, and share the information. [15]

This project was specified by TietoEnator. The department with responsibility for this project is the Telecom Platform business unit which is a sub unit of the Telecom R&D Infrastructure business unit. The Telecom Platform business unit works with systems for telecom access, core and service networks. This department is a development and integration service provider and its main task is assembling and testing the telecom platforms, telecom systems or subsystems. Normally the department does not sell any products. [14]

The aim of this project is to set up a prototype of the COTS component database which actively and easily helps those who are working on the platform integration purchasing the COTS components, and integrating platforms systems. [13]

1.2 Project development

The dissertation work was accomplished in four steps: [13]

1) Firstly, a study was performed to gain insight into which information was needed in the database to make it a useful tool for the platform integration department. We started with some general papers about the modern telecom industry, open platform, standards, and COTS components. Later, we found a set of standards related organizations which focused on different area of a typical telecom system.
2) Secondly, a study of the standards specifications was made to gain a deeper understanding of the telecom area. Some of the normal attributes and the interface information of the COTS components were found from the standards specifications. The work from a set of standards structuring organizations has been mentioned as complementarities.
3) Thirdly, a search of example COTS components was performed. By analyzing the components datasheets, the requirements specification for the database began to be more specific and clearer. The process of finding and collecting the components also helped us to understand more about the purposes, and the issues involved in this project.
4) Finally, the prototype of the database was designed and populated with information reflecting a sample set of hardware and software components. The prototype was used as a demonstration and can be considered as a verification of the requirements specification for this project.

1.3 Dissertation structure

This dissertation first presents the general knowledge and terms which will be used in the dissertation, such as telecom industry, COTS components, and open standards. Then the dissertation describes some of the core ideas from several standards issuing organizations and the main content of their standards, such as PICMG (ATCA), OSDL CGL WG (CGL RD), and SAF (HPI&AIS). Organizations involved with structuring and organizing the standards and testing the standards specifications are also introduced in the dissertation, such as SCOPE, PICMG RES, CP-TA, MVA, and ITU-T OCAF. This dissertation analyzes information from a number of standard compliable COTS components and examines typical attributes from the different parts of a telecom system, such as the manufacturers, sellers, list prices, complied standards, functionalities and services, requirements, and verified interoperability. Finally, the dissertation describes the prototype design of the COTS component database. Current issues and future work are included at the end of this dissertation.

2 Background

2.1 The telecom industry

2.1.1 Introduction

The core task of the telecom industry is to transmit information over a distance. Transmission techniques nowadays can be both wire (using cable - electrical or fiber-optic) and wireless, where the form of the information transfer is electromagnetic, such as digital and analog signals. [78]

The broad view of the telecom industry includes radio, television, telegraph, fixed-line telephone, mobile phone, and computer networks, and a narrower view includes communications and data service networks. This dissertation only focuses on platforms for communications and data service networks. [78]

2.1.2 System overview

The system reference model is used to describe the system architecture, building blocks and interfaces between blocks. The system integrator must have a system reference model before they start developing the system. [15]

In general, the structure of a telecom system is the following (Figure 2.1):

5	Telecom Applications
4	Telecom Services and Protocols
3	Clustering Software/Middleware
2	Operating System (OS)
1	System Hardware

Figure 2.1: The telecom system overview [6]

1. The system hardware includes the central processing unit (CPU), storage, switch, and I/O.
2. The operating system can be for example Linux, or Solaris.
3. The clustering software/middleware includes fail-over, load-balancing, and data replication.
4. The telecom services and protocols can be for example signaling system 7 (SS7), H323, session initiation protocol (SIP), and media gateway control protocol (MGCP).

5. The telecom applications can be for example billing, provisioning, interactive voice response (IVR), directory services, and home location registers (HLRs).

The telecom industry helps people to communicate with each other expediently, efficiently and easily. More and more people are using the Internet to get information, using E-mail to contact each other and using the short message service (SMS) to send messages. The telecom industry has become a very important part of the modern world. With the development of the telecom industry, services have become more varied, systems are more complicated, and the technology is more mature today. [78][10]

2.2 Commercial off-the-shelf (COTS) component

2.2.1 Component

A component is a part, a constituent element, or a piece of a system or a composite entity. A component can be constituted by a set of sub-components. For example: a computer is made up of software and hardware, and the hardware is made up of for example a motherboard/system board, CPU, random access memory (RAM), hard disk, sound card, and graphics card. [36][15][10]

A component in the telecom industry is a functionally identifiable independent element in the system which can be replaced by another component with the same functionality and interfaces. Each component in the system should have an interface specification and a set of encapsulated functions or services, and the architecture of the system must be well defined. Generally, the type of components in a telecom system can be categorized as hardware, operation system, middleware, and other software. [36][15][10]

2.2.2 Customized component and off-the-shelf component

Generally, there are two types of components, one is the customized component which is made for a certain user and the user has to wait for the manufacturer or their own developer to produce it. The other type is the off-the-shelf component where the user does not need to wait for the component to be produced and may use the component directly.

Off-the-shelf components may be either cost-free or paid for. For example, open source software is normally available for everyone to download with no payment. There are a large number of off-the-shelf components which the consumer has to pay for as well, such as the operating systems from Microsoft. [15]

2.2.3 COTS component

A commercial off-the-shelf (COTS) component is an off-the-shelf product which is made by a second manufacturer and is available for purchase. The COTS components are similar to the normal commodities in stores which can be bought and used directly without any major modifications. [36][15]

COTS components are in effect special kinds of off-the-shelf components with the following characteristics: [15]

1) The purpose of producing COTS components is to make profit from selling or leasing.
2) There are always a certain number of the same COTS components available for purchase.
3) The core technology and intellectual property rights of the COTS components are held by the vendor and the vendee may only use the component (generally without being able to change the functionalities and the services).

2.3 COTS component for the telecom industry

2.3.1 The requirements of the telecom COTS components

In order to have the best performance, all the COTS components used in a telecom system must have the following characteristics:

1) Carrier Grade

Carrier grade, in the telecom area means that the system or the components have high availability, high performance, high security, uniform interface, efficient interoperability, and good scalability. [34]

2) High Availability

Availability depends on both the reliability and the reparability, and is a property of both the components and the system as a whole. The reliability is measured by the Mean Time Between Failure (MTBF) and reparability is measured by the Mean Time To Repair (MTTR). Availability is then defined as MTBF/(MTBF+MTTR). A carrier grade system is expected to have an availability that is equal or better than 99.999% (also referred to five nines), which means that in each year the system only has about five minutes unscheduled downtime. [1]

3) Interoperability

Interoperability is the ability of a component or a system to exchange information with another system or component and to be able to use that information to work together. Since a telecom system is built by various independent components, it is important that the components in the system are easy to integrate with each other. Interoperability is one of the most important abilities for a COTS component. [48]

4) Hot Swapping

Hot swapping is the ability of a system to be able to remove and replace its components during uptime. To implement hot swapping the system must be able to identify defective components and if a new hot swappable component is added in, the system should not be required to shut down or reboot but can continue working with no downtime. The hardware, software and middleware in the system should have the ability to redeploy the components instantly. Hot swapping is one of the most important abilities for a high availability system. [47]

2.3.2 The advantages of using COTS components

TEMs today face radical business challenges. Pressure arises from new competitors and new requirements. TEMs have to improve their services, lower the cost price, and speed the product to market. Using COTS components and open source components instead of the customized components is a good solution for TEMs to achieve a dominant position in today’s market. Table 2.1 shows the developing processes of a telecom system by using customized components and COTS components. [13]

Step	Customized Component	COTS Component
1	Analyze the system requirement with the knowledge of the private system.	Analyze the system requirement with the knowledge of the existing COTS components and open standards, and the open system.
	Analyze the customized components requirements.	Analyze the COTS components requirements.
2	Develop the customized components.	Check the COTS components datasheets.
		Order the COTS components.
3	Test the customized components.	Test the COTS components.
4	Integrating the customized components for the system.	Integrating the COTS components for the system.
	Test the system.	Test the system.
5	Maintenance and update the system.	Maintenance and update the system.

Table 2.1: The developing processes of a telecom system by using customized components and COTS components

In Table 2.1, the main differences of the processes between using customized components and using COTS components are step 1 and 2. Step 5 may also be considered as simpler for systems using COTS components.

Compared with customized components, COTS components bring advantages for both system developers and component manufactures.

For the system developer, they do not need to order or develop any special customized products and wait for the manufacturer or their own developers to develop them. As a component vendee, they only need to select the COTS components which have the corresponding functionalities and services for their requirements. Using COTS components can help the customers to increase efficiency and shorten the development time. Using COTS components can decrease the reliance on proprietary solutions thus the system is more flexible and easier to maintain and update.

For the manufactures of COTS components, the development cost and time for each unit decreases since they do not have to provide any customizations for any particular individual customer. As the component vendors, the telecom equipment manufactures (TEMs) can product a number of the same COTS components and sell them to different component vendees. [36][15]

2.3.3 The possible disadvantages of using COTS components

There are also some possible disadvantages of using COTS components.

Using COTS components may reduce the cost for a single product, but, on the other hand, COTS components may not match all the specific requirements from the vendees. Thus, sometimes the vendees need to buy several more COTS or customized components to obtain all the functionalities and services for the requirements, and the cost for the whole system may increase. There is also the possibility that the COTS component has more functionality and services than the requirements from the vendees, and then the vendees may pay more for the unused parts as well.

Buying COTS components to integrate a system may shorten the time to market. But if we consider the whole life cycle of producing a COTS component, the time to create the open standards which are used to define a COTS component should be included. Usually a long time (several years) is needed to make a standard and get it accepted by most of the manufacturers. Thus, when system integrators need some components with some special requirements that are not compliable with any existing standards, they have to wait until the standard is created or finally order the customized component. [15]

2.4 Open standards

2.4.1 Open standard

A standard is “A publicly available document that defines specifications for interfaces, services, processes, protocols, or data formats and that is established and maintained by group consensus.” [15]

Since more and more COTS components from different manufactures are used to build the modern telecom system, using open standards would be a good solution to define the interfaces and increase interoperability and compatibility between various COTS components. [4]

A completely open standard is a public standard which can be accessed and used by everyone without cost. Users may also suggest changes without having to pay a membership fee to the relevant standards issuing organization. Standards range from a completely open standard to a completely proprietary standard. Completely proprietary means exclusively owned and private. [40]

Most of the existing open standards are not completely open. The states of the open standards specifications which are mentioned in this dissertation are the following:

Standard	Cost free access	Cost free use	Cost free change
PICMG—ATCA	No	No	No
OSDL CGL WG —CGL RD	Yes	Yes	No
SAF—HPI&AIS	Yes	Yes	Free for academic researchers only

Table 2.2: The states of the open standards specifications used in this dissertation [76][64][54][46]

Since the purpose of producing standards is to standardize the components in the market, the fee for accessing and using the standards should be appropriate.

2.4.2 Standards issuing organizations

The open standards must be issued by some professional organizations and be accepted by the companies involved. The more companies that accept the standard, the more reliability the standard has.

There are a large number of organizations which create and/or promote standards for the telecom industry. The number of telecom consortia on the list from "ConsortiumInfor.org" is around 69 by Sep, 2006, and some of the organizations are missing, such as Open Source Development Lab (OSDL). [37] [38]

This dissertation chooses three of the standards issuing organizations which produce the standards accepted by most of the large telecom manufacturers. Their standards specifications cover the hardware, the operation system and the middleware of telecom systems. The telecom services and the application software are too multifaceted and varied, thus there is no very typical professional organization which produces a standard which is widely accepted.

The names and the acronyms of the standards issuing organizations are listed in Table 2.3:

Organization	Acronyms	Member Number (Sep, 2006)	Example Members
PCI Industrial Computer Manufacturers Group	PICMG	110+273	Artesyn Technology, Huawei, Kontron, Radisys, Sun Microsystems
Open Source Development Lab	OSDL	70	Novell, NTT Data Intellilink, Rad Hat, TurboLinux, Wind River Systems
Service Availability Forum	SAF	39	Ericsson, GoAhead Software, Huawei, Kontron, Oracle, Radisys, Solid Information Technlolgy, Sun Microsystems, TietoEnator, Wind River Systems

Table 2.3: The names and members of the standards issuing organizations which are mentioned in this dissertation [52] [63] [71]

The standards issuing organizations are mainly made up of representatives from large companies. Alcatel, Fujitsu Limited, IBM, Intel, MontaVista Software, Motorola, NEC, Nokia, NTT Corporation, and Siemens are the members of all the three organizations which are mentioned in Table 2.3.

Not only companies but also individuals are allowed to register the standards issuing organizations. Each member can choose a level of participations, from the free memberships for access and using to the most expensive level where the member may join the discussion about creating and modifying the standards specifications. [52] [63] [71]

2.4.3 Open standards for telecom industry

Each of the standards issuing organizations focuses on a particular part of the system. Since the purposes of telecom systems are varied, there are special standards for different kinds of telecom systems from the same organization. The standards which are mainly used in this dissertation are shown in Table 2.4.

Organization	Standard	System Layer
PICMG	ATCA (PICMG3.0-3.6)	Hardware Platform
OSDL – CGL GW	CGL RD v3.2	Operating System
SAF	HPI B.01.01	API[15]s: Hardware Platform Management – SA[16] Middleware
	AIS A.01.01, B.02.01	APIs: SA Middleware – Application Software

Table 2.4: Open standards

The Advanced Telecom Computing Architecture (ATCA/AdvancedTCA/PICMG3.0-3.6) is the standard made by PICMG which describes the shelf hardware platform. The ATCA standard has seven specifications with version numbers from PICMG3.0 to PICMG3.6. The ATCA hardware includes for example the shelves/chasses, CPU blades, storage blades, server blades, switch blades, and I/O blades. [79]

[15] API: Application Programming Interface
[16] SA: Service Availability

The Carrier Grade Linux Requirements Definition (CGL RD) v3.2 is the standard produced by the Carrier Grade Linux Working Group (CGL WG), which is a subsection of OSDL. CGL WG is in charge of specifications for both Carrier Grade Linux (CGL) OS and the software development tools. The CGL RD v3.2 has eight specifications and covers seven areas of the CGL OS: availability, clustering, hardware, performance, security, serviceability and standards. [33]

The SAF specifications describe the Service Availability (SA) Middleware Interfaces and currently have two existing subsections: Hardware Platform Interface (HPI) and Application Interface Specification (AIS). HPI defines the APIs between the hardware platform management and the SA middleware; AIS defines the APIs between the SA middleware and the application software. The APIs allow the SA middleware to be an independent block in the system. [1]

2.4.4 Standards structuring organizations

Since the standards issuing organizations mainly focus on their special areas of interest in the telecom system, there is no seamless integration available between the standards issued by the different organizations. The standards may have gaps between them or overlap each other. To resolve this problem and to try to supply a reference model that covers an entire system and not only a single layer, some other organizations have emerged. These organizations do not produce any standards but identify and fill gaps, produce requirements and profiles to fit certain business areas for the telecom market. Examples of the standards structuring organizations are SCOPE Alliance (SCOPE), PICMG Requirements Engineering Subcommittee (PICMG RES), Communications Platforms Trade Association (CP-TA), Mountain View Alliance (MVA), and Open Communication Architecture Forum (OCAF).

2.5 COTS component database

A database is a collection of information which is organized in a systematic way by the logical relationships between the data. A database helps people to easily search, share, and analyse the information. The modern telecom system contains a large amount of information such as the information from the corporations, the standards specifications, the system stack information, and the components information.

There are three benefits of having a COTS component database:

1) For the COTS component vendees, a COTS component database may help them make purchase decisions. A database with the component information can be a good tool for

the vendees to compare the functionalities and services, and the list prices of a set of components.

2) For the COTS component vendors, a public COTS component database may help them to market their products easily. It is better to have a single database instead of the multiple websites from which people can find all kinds of components information from different vendors.

3) For the component developers and the system integrators, a COTS component database with standards information can help them to do a general test of the compliance and the adaptability between different components.

2.6 Summary

Figure 2.2 shows a simple reference model of a telecom system with the open standards from different standards issuing organizations. The next generation of telecom systems will be based on open standards and COTS components.

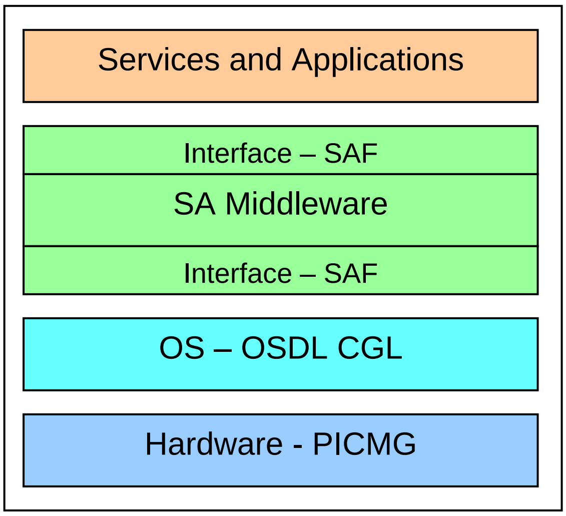


Figure 2.2: A simple reference model of the telecom system

The process of developing the next generation telecom systems is changing because more COTS components are used for standardized platform integration. The system developers should be better aware of the information concerning the interrelated standards and the COTS components in the present market before they start specifying requirements for a system.

Instead of developing their own products, the system integrators select and buy the COTS components from the market. It is necessary to set up a database with all the related information of the COTS components for the telecom industry. The components in a COTS component database have to follow certain open standards to ensure interoperability. [13]

3 Open standards and related organizations

3.1 PICMG and ATCA

Note: The information in this section has been compiled from information provided on the web by various organizations.

3.1.1 PICMG

The PCI Industrial Computer Manufacturers Group (PICMG) is a non-profit organization which develops open standards specifications for the carrier grade hardware platform for "telecomunications and industrial computing applications" [19]. PICMG was founded in 1994, and by July of 2006, there were 58 specifications documents on the list. The standards from PICMG can be classified into several sections, such as CompactPCI, AdvancedTCA, AdvancedMC, CompactPCI Express, COM Express, and SHB Express. [77] [19]

PICMG is one of the most mature standards issuers and has more than 450 involved industrial companies. The specifications from IPCMG have been issued since 1994, and in the last twelve years PICMG has continuously developed open standards specifications for different purposes and types of components for open telecom hardware platforms, such as chassis/shelves, carrier boards, switches and hubs, AMC[17] modules, single board computers, storage blades. Products based on the PICMG standards contain most of the hardware blocks from different telecom system models. PICMG offers the open specifications to all the manufacturers and helps the system integrators to increase the use of COTS components. [19][42]

3.1.2 ATCA specifications

Note: The information in Table 3.1 has been compiled from [77] and [24].

The Advanced Telecom Computing Architecture (ATCA/AdvancedTCA) standard is a set of specifications which have been incorporated from PICMG3.0 to PICMG3.6. The ATCA

[17] AMC: Advanced Mezzanine Card

standard has been developed by PICMG since November 2001 with 105 companies. [27] The ATCA documents from PICMG are shown in Table 3.1.

name	version	status	issued time	description
AdvancedTCA Base	PICMG3.0 ECN001	adoptable	2004-1-21	
AdvancedTCA Base	PICMG3.0 ECN002	adoptable	2006-4-29	
AdvancedTCA Base	PICMG3.0 R1.0	obsolete	2002-12-30	
AdvancedTCA Base	PICMG3.0 R2.0	adoptable	2005-3-18	*Core specification defining architecture mechanicals, power, system management, fabric connectors, and Base interface (10/100/1000 Base-T). [24]*
AdvancedTCA Ethernet	PICMG3.1 R1.0	adoptable	2003-1-22	*Specification for Ethernet and Fibre Channel Fabric interface. [24]*
AdvancedTCA InfiniBand	PICMG3.2 R1.0	adoptable	2003-1-22	*Specification for InfiniBand* Fabric interface. [24]*
AdvancedTCA StarFabric	PICMG3.3 R1.0	adoptable	2003-5-21	*Specification for StarFabric*/Advanced Switching interface. [24]*
AdvancedTCA PCI Express	PICMG3.4 R1.0	adoptable	2003-5-21	*Specification for PCI Express* and Advanced Switching Fabric interface. [24]*
AdvancedTCA RapidIO	PICMG3.5	adoptable	2005-9-21	*Define how Serial RapidIO transport is mapped onto PICMG 3.0. [77]*
AdvancedTCA PRS	PICMG3.6	member review		*Define how Packet Routing Switch (PRS) is mapped onto PICMG 3.0. [77]*

Table 3.1: ATCA documents [77][24]

Most of the ATCA specifications are not cost free for access. The standard user needs to pay to access the full text documents. However, there is a short form version of PICMG3.0 which can be downloaded without payment. [23]

The basic specification PICMG3.0 consists of 430 pages and describes the properties of the mechanical configuration, power, cooling, interconnection, reliability, availability, serviceability and manageability (RASM) of the ATCA components. The ATCA standard also designs the sub-components such as front boards, rear transition modules (RTMs), mezzanine cards, and ATCA backplanes which a shelf is made up of. The technologies of the mechanical configuration, power, cooling, shelf management, and data transport are described in the basic ATCA specification as well. Subsidiary specifications describe different schemes of the backplane fabric interfaces for data transport. [3][25]

3.1.3 ATCA properties

Note: The information in Table 3.2 has been cited from [28].

Since the standard is used to help manufacturing the production, there are a number of attributes about the components in the speculations. Table 3.2 shows the properties of the components from ATCA standard speculations.

ATCA Mechanical Configuration
▪ 8U boards in 12U chassis ▪ 1.2" board pitch allows heat sinks plus rear SMT ▪ Forced air cooling for up to 200 watts per slot ▪ Front and rear fiber bend area in 600mm depth ▪ Simplified sheet metal construction ▪ ETSI & NEBS vibration, shock and serviceability
ATCA Power & Cooling
▪ -48V/-60 VDC power input ▪ Redundant power inputs ▪ Distribution of ringing voltages ▪ Capacity of over 3,200 Watts per shelf ▪ Local power conversion on each board
ATCA Shelf Management
▪ Monitor & control low-level aspects of ATCA boards and other field replaceable units (FRUs) within a shelf ▪ Watch over basic health of the shelf, report anomalies, take corrective action when needed ▪ Retrieve inventory information & sensor readings ▪ Receive event reports and failure notifications from boards and other intelligent FRUs ▪ Manage power, cooling & interconnect resources in the shelf ▪ Enable visibility into a shelf for a logical System Manager

▪ Overall, sophisticated shelf management
ATCA Data Transport
▪ ATCA backplane is designed for four different fabric topologies including Star, Dual Star, Dual-Dual Star, and Full Mesh ▪ In the Star topology, each node board is connected to one central switching board. Applications for the star topology include non-carrier-grade with little latency sensitive data traffic. ▪ In the Dual-Star topology, there are 2 connected redundant switching boards that are centralized, and each node board connects to both switching boards. Redundancy decreases downtime from failure. Applications for the dual-star topology include carrier-grade with non-latency-sensitive data requirements like a modular server. ▪ In the Dual-Dual Star topology, there are 2 distinct and redundant (i.e 4 in total) centralized fabrics, one for control and one for data. Each node board is linked to all four of the fabrics. This topology is useful for carrier grade applications with latency sensitive streaming data requirements and significant control and management, TCP/IP-based, workload. Data throughput is optimized since the control and data planes are separated. ▪ In the Full Mesh topology, each board can be directly connected to every other board. Switching and management is distributed across all boards, with no centralized switching board. This topology is suited for carrier-grade applications with large data throughput requirements, like routers. The full mesh topology is highly redundant and scalable, as all boards can intercommunicate simultaneously. A 16-slot ATCA shelf with a full mesh topology of 10Gbps per connection has a total bandwidth of 2.4 Tbps! ▪ Fabric alternatives defined for ATCA (within PICMG 3.x specifications) include Ethernet, FiberChannel, InfiniBand, StarFabric, PCI-Express, and Rapid I/O

Table 3.2: ATCA properties [28]

3.1.4 ATCA hardware platform

An ATCA hardware platform is comprised of the shelf hardware (interconnection), the blade hardware and the hardware platform management software, and is used for wireless networks, access servers, edge servers, and telecom servers. See Figure 3.1. [50]

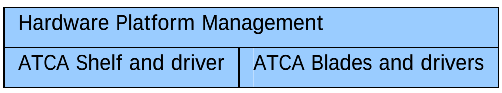


Figure 3.1: ATCA hardware platform

The ATCA standard mainly describes the shelf hardware for the carrier grade system. As long as the structure, the size and the interfaces of the shelf are standard, the blades which are used to fill in the shelf can be easily designed as well. One example of a hardware platform with the COTS components is illustrated in Figure 3.2:

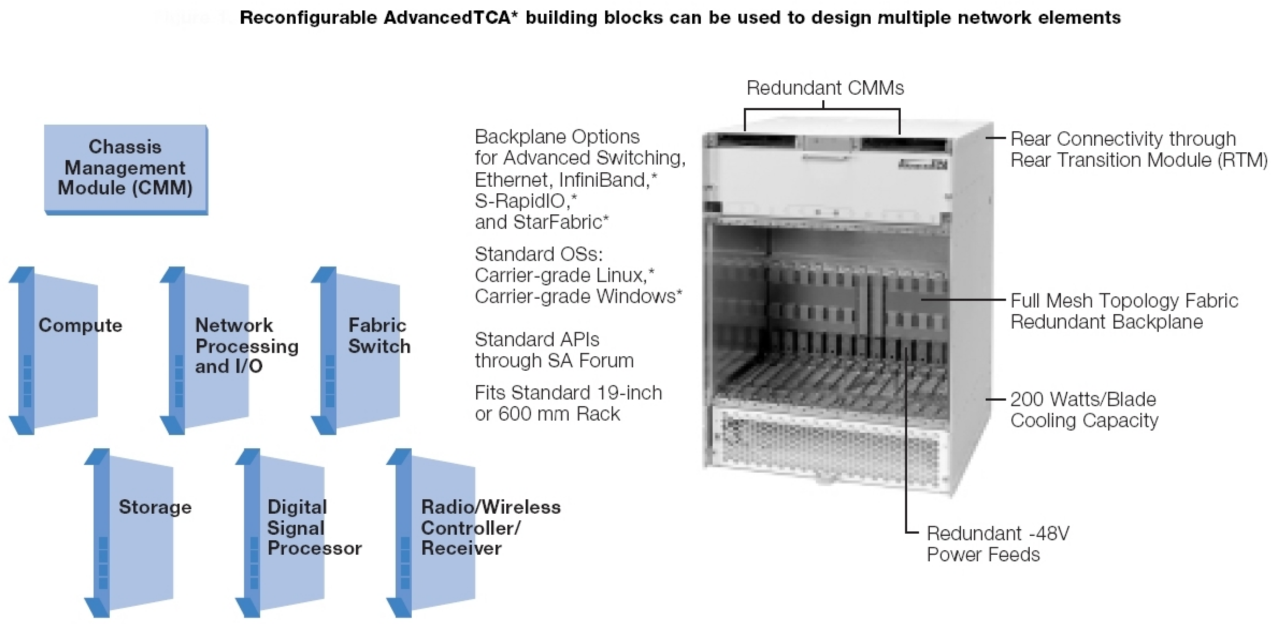


Figure 3.2: ATCA building blocks [79]

Note: The information about the hardware platform management has been compiled from [67].

The hardware platform management is a piece of control software which oversees the underlying hardware. The hardware platform management takes care of such aspects as [67]:

1) *Inventory: vendor name, product number, serial number etc. of each component that is installed in the system*
2) *(Over)voltage, (over)current, temperature etc. of the components in the system,*
3) *Events / alarms if there is something wrong with the above,*
4) *Recognizing Field Replaceable Units entering and leaving the system,*
5) *Powering components on and off,*
6) *Switching audible / visible information (LEDs, beepers, small text messages on the front panels) on and off,*
7) *Taking basic actions to assure the well-being of the system (like increasing the fan speed if the temperature of the components increase),*
8) *Others, depending on the attributes of the system.*

‘*Traditionally, the Hardware Platform Management "authority" is only restricted to basic actions when the system starts behaving wrongly. For instance, when a temperature of a replaceable component goes up, the control software would increase the fan speed, but will not switch the component off.*’ [67]

As low-level control software, the hardware platform management can be executed on any OS or even with out any OS [67].

3.1.5 Summary

The PICMG has been developing a series of specifications for the next generation of carrier grade hardware platforms. PICMG3.x is a set of mature specifications which describe the structures, parameters, methods and other technologies to make an ATCA shelf hardware platform. The ATCA hardware platform is a dynamic, high performance platform and it is one of the basic portions in a telecom system. The software, such as the OS and the SA middleware, is running on the hardware platform. Figure 3.3 shows the reference model from the ATCA.

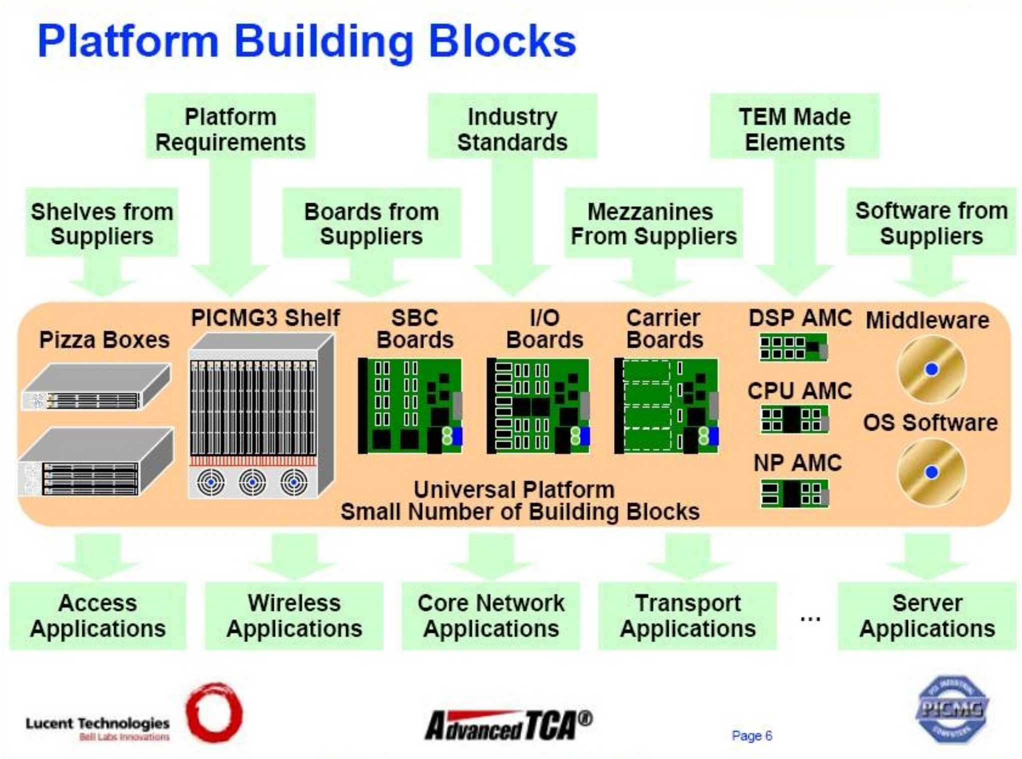


Figure 3.3: ATCA reference model [3]

The ATCA standard is widely accepted by most of the big TEMs and the ATCA COTS components have become an important part in the current telecom market. The PICMG

ATCA is helping the telecom corporations to change the traditional proprietary network architecture to the open standard architecture.

3.2 OSDL, CGL WG, and CGL RD

3.2.1 OSDL

Note: The material in this section is mainly a synthesis of material from the following sources.

- ♦ OSDL Webpage: [63][62][60][11][66][65][51][69]
- ♦ Other Webpages: [10] [61] [12][9][8]

The Open Source Development Lab (OSDL) is a nonprofit organization which supports the equipment and infrastructure of the open source developer to hasten the adaptation of Linux based OS platform in the telecom industry. OSDL also supports open hardware resources, “tests and reports on open source software” [61], and “employs a number of Linux developers” [61]. OSDL was founded in 2000 and currently has four main working groups: [62][60]

1) Carrier Grade Linux (CGL) Working Group: “was established in January 2002 to enhance Linux for use in a highly available, secure, scalable, and maintainable carrier-grade system” [11].
2) Data Center Linux (DCL) Working Group: “was established in August 2002 to develop the roadmap for Linux platform software that supports commercial software products and corporate IT requirements, enabling developers to create Linux-based solutions for the data center market segment” [10].
3) Desktop Linux (DTL) Working Group: was established on January 20th, 2004 “to create a forum where a range of desktop usage models can be studied, with recommendations on improvements to encourage broader adoption of Linux” [10].
4) Mobile Linux Initiative (MLI) Working Group: was established on October 18th, 2005 “to accelerate adoption of Linux on next-generation mobile handsets and other converged voice/data portable devices” [66][65].

The OSDL is mainly supported by Computer Associates, Fujitsu, Hitachi, Ltd., Hewlett-Packard, IBM, Intel Corporation and Nippon Electric Corporation. OSDL also receives help from a large number of other “software vendors, end-user companies and educational institutions” [61]. The OSDL had a total of 70 members on the roster by Sep, 2006, which includes 10 Linux vendor companies: Bull, Fujitsu, Hitachi, HP, IBM, Mitsubishi Electric, NEC, Stratus Technologies Inc., Toshiba Solutions, and Unisys. The members of OSDL cover most of the industry segments, such as end users, academic/university, equipment suppliers, system integrators or service providers, applications ISV[18]s, middleware ISVs, Linux OS providers, system vendors, and silicon suppliers. [63][51] With the help of OSDL, more and more corporate data carters, telecom networks, desktop computers, and mobile handsets start to use Linux based OS platforms [61].

3.2.2 CGL WG

The Carrier Grade Linux Working Group (CGL WG) "consists of leading network equipment providers, system integrators, hardware platform providers, Linux distributors, and carriers, working to define, collect and prioritize requirements that enhance standard Linux for demanding carrier environments" [11]. The CGL WG was founded in January 2002 and released a series of the CGL requirements definitions. By September 2006, CGL had released three main sets with 25 requirement definitions on the list. [69]

The CGL working group consists of three sub-groups: [11]

1. *Steering sub-group: This sub-group oversees the CGL initiative and provides directions to the marketing and technical sub-groups.*
2. *Marketing sub-group: The Marketing sub-group collects market requirements from member companies, prepares marketing documents and presentations, organizes meetings, conferences and panels, and prepares press releases. The Marketing sub-group is also active in the process of registration of CGL distributions through its Registration task team.*
3. *Tech Board sub-group: The CGL Tech Board sub-group is responsible for the CGL Specifications and the CGL Development task teams. The Specifications task team collects and defines requirements for carrier grade enhancements in Linux, for carrier applications. […] The Development task team solicits generates and consolidates documents detailing the design of CGL features and technology.*

[18] ISV: Independent Software Vendor

3.2.3 CGL RD v3.2

Note: The information in Table 3.3 has been compiled from [69].

The CGL Requirements Definition (CGL RD) v3.2 is the newest version from the CGL WG which was released on February 27th, 2006. CGL RD v3.2 has 8 documents which include one main requirements definition and seven accessional requirements. CGL RD v3.2 describes the availability, serviceability, performance, clustering, standards, hardware, and security of a Linux OS. [69]

name	version	status	issued time	description
CGL Requirements Definition Overview	CGL RD v3.2	Adoptable	2006-2-27	This document gives a summary of the documents in the V3.2 document set (Availability, Serviceability, Performance, Clustering, Standards, Hardware and Security).
CGL Availability Requirements Definition	CGL RD v3.2	Adoptable	2006-2-27	Availability requirements apply to the Linux kernel, core libraries, and tools essential to a carrier-grade system. These availability requirements are related to single system availability.
CGL Serviceability Requirements Definition	CGL RD v3.2	Adoptable	2006-2-27	Serviceability requirements define the tools and methods used by system administrators to manage, install, maintain, upgrade, and monitor a carrier grade system.
CGL Performance Requirements Definition	CGL RD v3.2	Adoptable	2006-2-27	Performance requirements have a significant bearing on application performance. Carrier grade applications have some unique requirements, although they also share many needs with more general applications.
CGL Clustering Requirements Definition	CGL RD v3.2	Adoptable	2006-2-27	Clustering requirements are aimed at supporting clustered applications in a carrier-grade environment as an effective way to achieve highly available services inside a network element.
CGL Standards Requirements Definition	CGL RD v3.2	Adoptable	2006-2-27	Standards requirements cover the intersections of the OSDL CGL requirements with those of "recognized" standards bodies, such as POSIX, IETF, and DMTF. This document serves as a reference for the standards referred to in the CGL 3.2 functional requirements.

CGL Hardware Requirements Definition	CGL RD v3.2	Adoptable	2006-2-27	The telecommunication industry is migrating from proprietary platforms toward a commercial off the shelf (COTS) building blocks infrastructure. Important hardware building blocks for carrier grade systems and the open source software needed to support this hardware are identified in this document.
CGL Security Requirements Definition	CGL RD v3.2	Adoptable	2006-2-27	The security objectives and requirements in this document are aimed at analyzing and mitigating threats and to improve resiliency to attacks on CGL systems.

Table 3.3: CGL RD v3.2 documents

The CGL RD v3.2 documents are all very short; the number of pages is between 13 and 32. The CGL RD v3.2 is available for downloading cost free.

3.2.4 CGL platform

The CGL RD mainly describes the Linux based OS for the carrier grade system. "A Linux kernel with Carrier Grade characteristics is an essential building block component of such platforms and architectures" [12]. Figure 3.4 shows the components in a CGL OS platform.

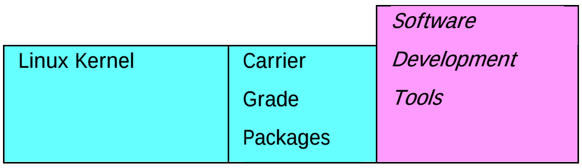


Figure 3.4: CGL OS platform

One example of a Linux OS platform which is used in the access and corporate networks is illustrated in Figure 3.5:

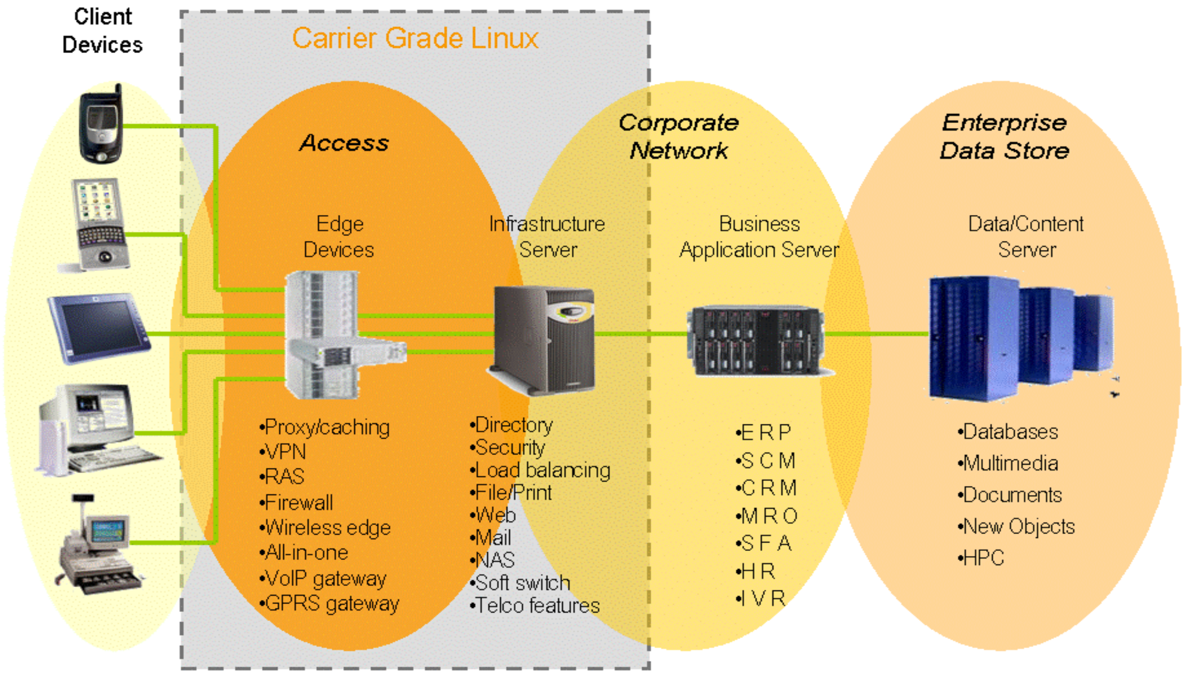


Figure 3.5: The context of Carrier Grade Linux in access and corporate networks [9]

3.2.5 Summary

The CGL WG is a sub-portion of OSDL which is working to improve the Linux based OS platform for the next generation of telecom systems. The requirements definitions from CGL WG help in achieving greater standards-compliance, more hardware support, more availability, more clustering, more security, more performance and more serviceability for the Linux based OS platforms. [12]

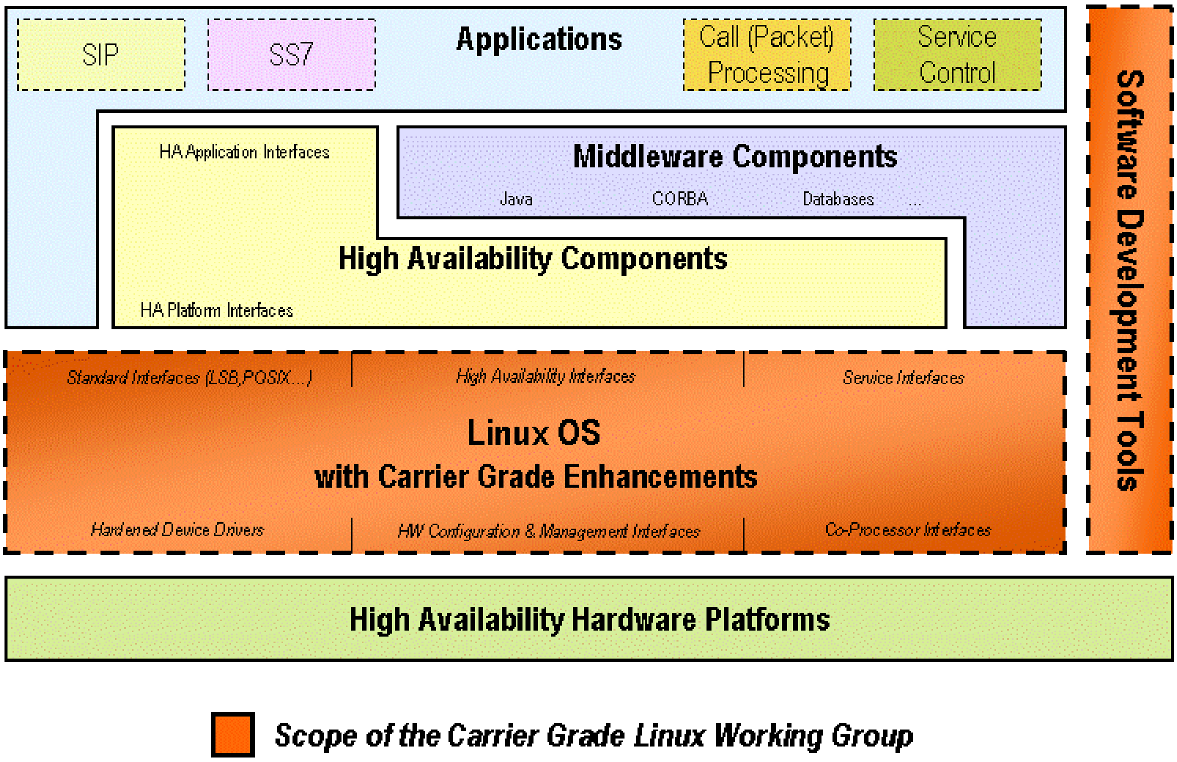


Figure 3.6: CGL reference model [8]

Figure 3.6 shows the reference model and the scope of the CGL WG. The CGL WG is supported by several large telecom companies and related organizations. The CGL OS platform works well with the ATCA hardware platform and the SA middleware. [12]

3.3 SAF and HPI&AIS

3.3.1 SAF

"The Service Availability Forum™ is a consortium of industry-leading communications and computing companies working together to develop and publish high availability and management software interface specifications" [59]. SAF was founded in 2001 by 20 member companies, and there are currently more than 30 members [17].

The goals of SAF are to supply standard interfaces for defining a SA middleware and to contribute in the work of creating a COTS based carrier grade system by collaboration with other standards issuing organizations [21]. SA middleware has three interfaces: Hardware Platform Interface, Application Interface, and Systems Management Interface. See Figure 3.7.

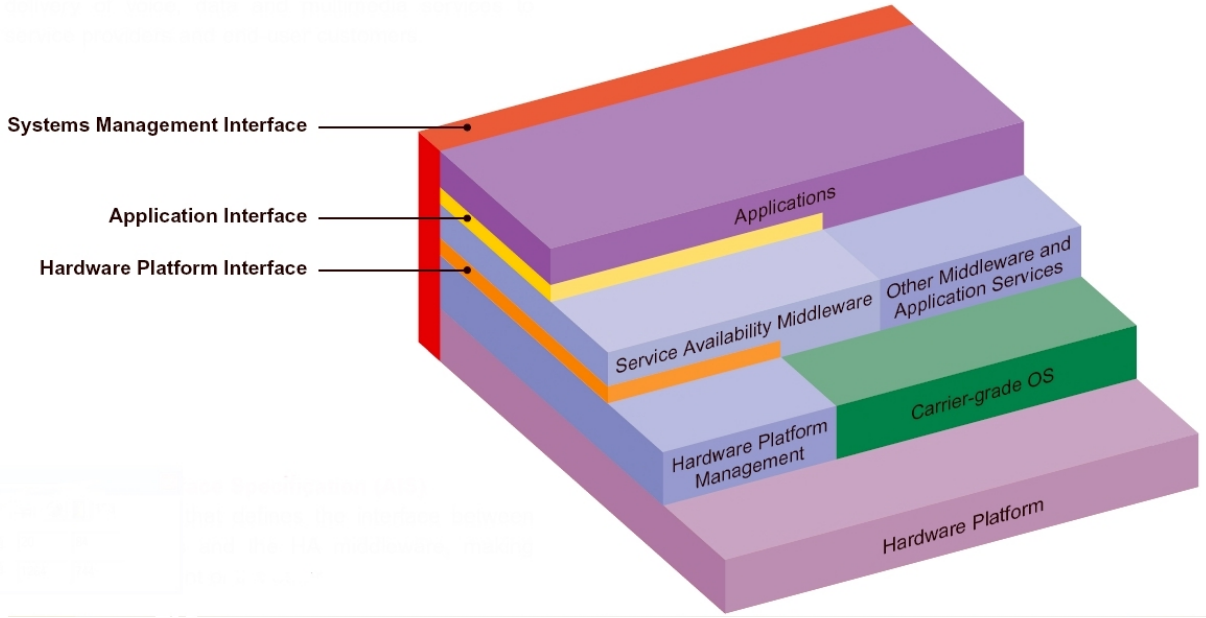


Figure 3.7: SAF reference model [58]

3.3.2 SAF Specifications

Note: The information in Table 3.4 has been compiled from [74].

There are three main specifications from SAF: [58]

1) Hardware Platform Interface (HPI): defines the API between the hardware platform management and the SA middleware.
2) Application Interface Specification (AIS): defines the API between the applications and the SA middleware.
3) System Management Specification (SMS):
    a) “An interface specification that enables service event and error reporting by AIS and HPI” [58].
    b) “SNMP and Web-based interface that provides distributed monitoring and control access to AIS and HPI Management Functionality” [58].

The adoptable version of the HPI specification was released in March, 2004, and the newest version of the AIS was released at the end of 2005 with the version number B.02.01. SMS is currently under development. The latest versions of the main specifications from SAF are shown in Table 3.4.

name	version	status	issued time	description
Overview	SAI-Overview-B.02.01	Adoptable	2005	Provides a brief guide to the remainder of the Service Availability Forum (SA Forum) Interface Specifications documents and includes a description of the SA Forum Information Model.
AIS: Availability Management Framework Service	SAI-AIS-AMF-B.02.01	Adoptable	2005	Describes the Availability Management Framework API.
AIS: Checkpoint Service	SAI-AIS-CKPT-B.02.01	Adoptable	2005	Describes the Checkpoint Service API.
AIS: Cluster Membership Service	SAI-AIS-CLM-B.02.01	Adoptable	2005	Describes the Cluster Membership Service API.
AIS: Event Service	SAI-AIS-EVT-B.02.01	Adoptable	2005	Describes the Event Service API.
AIS: Information Model Management Service	SAI-AIS-IMM-A.01.01	Adoptable	2005	Describes the Information Model Management Service API.
AIS: Lock Service	SAI-AIS-LCK-B.02.01	Adoptable	2005	Describes the Lock Service API.
AIS: Log Service	SAI-AIS-LOG-A.01.01	Adoptable	2005	Describes the Log Service API.
AIS: Message Service	SAI-AIS-MSG-B.02.01	Adoptable	2005	Describes the Message Service API.
AIS: Notification Service	SAI-AIS-NTF-A.01.01	Adoptable	2005	Describes the Notification Service API.
Distributed Systems Management for AIS-SNMP[19]	SAI-AIS-SNMP-A.01.01	Adoptable	2005	Describes the AIS SNMP MIB[20]s.

19 SNMP: Simple Network Management Protocol

[20] MIB: Management Information Base

HPI-to-AdvancedTCA Mapping Specivication	SAIM-HPI-B.01.01-ATCA	Adoptable	2005	Describes the mapping on ATCA platforms.
Hardware Platform Interface: Specification	SAI-HPI-B.01.01	Adoptable	2004	Describes the HPI API.
Distributed Systems Management for HPI-SNMP	SAI-HPI-SNMP-B.01.01	Adoptable	2005	Describes the HPI SNMP MIBs.

Table 3.4: SAF specifications [74]

The number of pages of the SAF Specifications in Table 3.4 is between 44 and 344 and all the documents are available for downloading cost free.

3.3.3 SAF properties

Note: The information in this section has been compiled from information provided on the web by SAF and various sources.

From the HPI and AIS specifications, the SA middleware mainly has two sets of properties:

HPI resource has the capability to support hot-swappable components and manage the hardware components by sensors, controls, inventory data repositories, watchdog timers, and annunciators. [46][18]

1) **Sensors:** Sensors are used to monitor the environment and the physical characteristics of the hardware components. Sensors measure the temperatures, the fan speeds, the input/output voltages, and report the static configuration data.
2) **Controls:** Controls are used to command the hardware components. There are five types of controls: digital, discrete, analogy, stream, and text.
3) **Inventory data repositories:** Every hardware component in the system should have inventory data. The inventory data contains manufacturer, model number, revision level, serial number, and static configuration. All the inventory data are kept in the repositories. The user can read, add, change, or delete inventory data via the HPI interface.

4) **Watchdog timers:** Watchdog timers are used to manage the implementation time of a hardware component. The functions include start a watchdog timer, time the timer, and tell the hardware component what it should do when the time is finished.
5) **Annunciators:** Annunciators are used to display the fault condition and other status information. The hardware platform should support the entities which can show the visible or audible information to people. HPI is independent from the hardware platform but it can limit the ability of the HPI user by control which and how does the alarm work.

AIS defines the availability management formwork and eight core services: [74]

1) The Availability Management Framework is used "for overall management of the cluster wide services" [16].
2) The Cluster Membership Service is used "for book-keeping of processing nodes joining and leaving the cluster" [16].
3) The Checkpoint Service is used "for replicating information between redundant nodes" [16].
4) The Event Service is used for "Event notification and response facilities within the cluster" [16].
5) The Message Service is used "for cluster wide messaging" [16].
6) The Lock Service is used "for cluster wide synchronization services" [16].
7) The Information Model Management Service is used for managing the SAF information model which is specified in UML [16].
8) The Log Service "enables applications to express and forward log records through well-known log streams that lead to particular output destinations such as a named file" [74].
9) The Notification Service is "to a great degree, based on the ITU-T Fault Management model as found in the X.700 series of documents as well as many other supportive recommendations" [74].

3.3.4 SA middleware platform

The SA middleware is managing the hardware, detecting errors and handle the replacement or introduction of new hardware. For example, when the temperature of a replaceable component goes up, the management software can not switch the component off but the SA middleware can [67].

The current SA middleware platform contains the body of the SA middleware and two types of interfaces. See Figure 3.8.

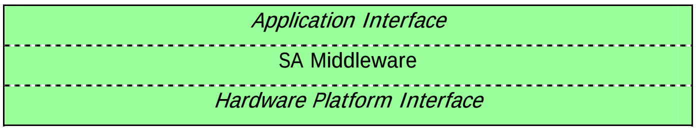


Figure 3.8: SA middleware platform

The SAF interfaces are a set of “C” programs and they are strictly OS independent. By using the standard based interfaces, the SA middleware is independent of the other building blocks in the system [50].

3.3.5 Summary

SAF is working to define standard interfaces between the hardware platform management, the SA middleware, and the application software. By using standard interfaces, the system can be divided into several independent layers, and the component developer needs only to focus on their own blocks of the system. Same as the hardware interfaces, the consistent standard software interfaces enable consistency for different layer of the components in a system. [21]

SAF collaborates with PICMG, and OSDL to define comprehensive open standards and COTS components based system [58].

3.4 Standards structuring organizations

3.4.1 SCOPE

The SCOPE Alliance is a very new industry alliance founded on January 1st, 2006 “by Alcatel, Ericsson, Motorola, NEC, Nokia and Siemens” [20].

The goal of SCOPE is to perform market research to choose the main profiles of the COTS components from the existing standards which best meet the requirements from the telecom market. SCOPE also analyzes and identifies the gap areas of the existing standards and works together with the standards issuing organizations to fix the gaps. [20][73]

SCOPE has following the specific deliverable: [73]

1) Specification profile[21]s: the necessary and sufficient options or attributes from the existing standards specifications which are really match the real industry requirements. The profiles from ACTA and CGL are done. See Table 3.5 for an example.
2) Gap analysis: the missing options or attributes from the existing standards specifications.
3) Reference model: the reference model of the carrier grade based platform by using the open standards based COTS and FOSS[22] components. See Figure 3.9.
4) Technical position paper: the description of the SCOPE reference model.
5) Strategy white-paper: "the strategy and benefit of SCOPE to the industry" [73].

Category	Attribute Description	Comment
Fabric Interface	Fabric is Ethernet technology 1/2/4 with aggregation technology or 10 Gbits per slot (PICMG 3.1. OPT. 1, 2, 3, 9).	General trend is 10G and in short term 1G could be sufficient.
Fabric Interface topology	Dual star	
Power Budget	Maximum power dissipation per front blade shall be 200W.	
Power Supply	Shelf to require redundant power feed of 48V DC	-60V DC as option
Shelf	14 or 16 slots	
Update Interface	Interface between logical paired slots using 10 differential pairs between two slots.	
…		

Table 3.5: An example of the ATCA profiles formatted by SCOPE [29]

[21] Profile: "A profile is a subset of existing open specification from bodies like PIGMG, OSDL, SA Forum and others. This subset – or profile – reflects the technical requirements regarding the interfaces and building blocks to form a Carrier Grade Base Platform to meet the Service Providers' requirements." [73]
[22] FOSS: Free and Open Source Software

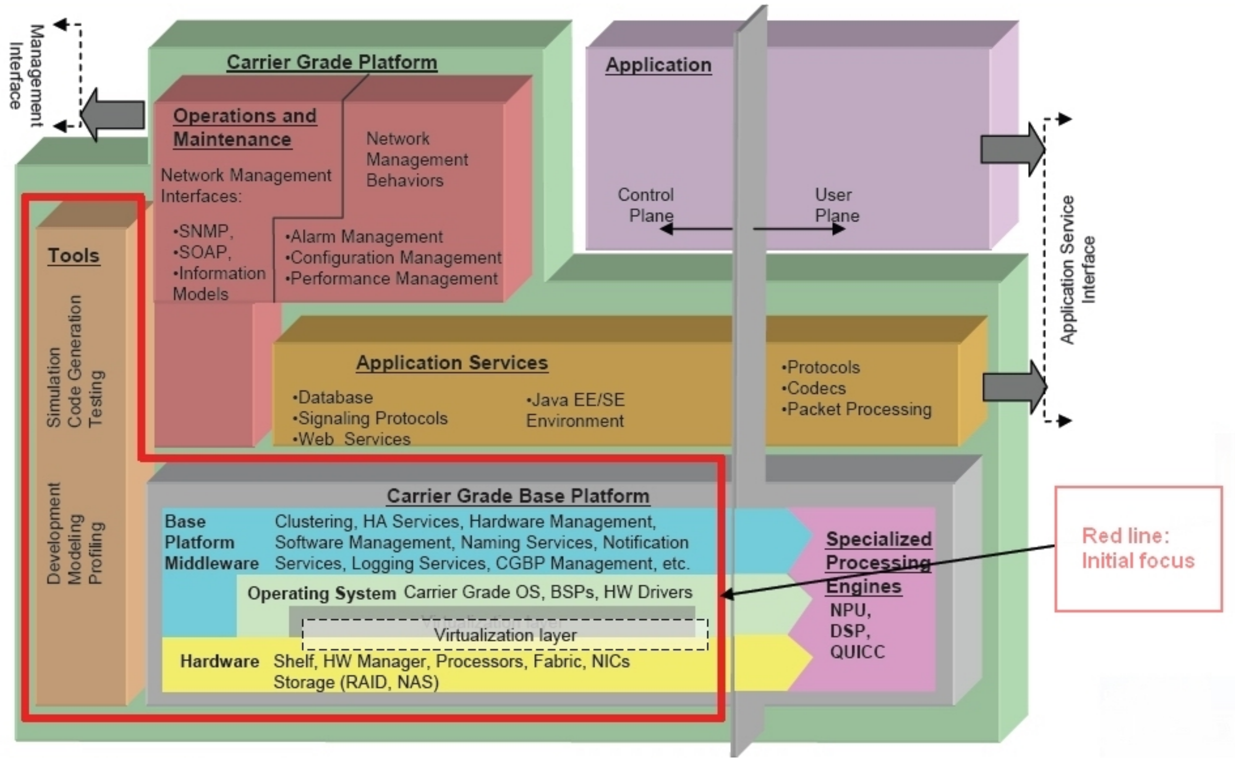


Figure 3.9: SCOPE reference model [72]

The standards which SCOPE are using are from PICMG, OSDL CGL WG, SAF as well as other standards issuers.

3.4.2 PICMG RES

The PICMG Requirements Engineering Subcommittee (PICMG RES) was approved by the PICMG executive members as a subcommittee of PICMG on 25th April 2006 [70]. PICMG RES currently has more than 65 member companies, and the executive sponsors are Lucent, Nokia, Kontron, Radisys, Artesyn, Alcatel, Tyco, Huawei and Pigeon Point Systems [56].

The work of PICMG RES is mainly based on the PICMG standards, especially ATCA and AMC currently. The PICMG RES summarizes and categorizes the mandatory and optional sections of the standards specifications. The scope of the PICMG standards is very wide, which means that they cover several options for different market segments and different elements within each market segment. The various options make it complicated for the standards user to find out the right profiles, which only match their special requirements for one specific market segment. The goal of PICMG RES is to provide a set of recommended requirements to reduce the design options and develop a framework to help industry groups to obtain the vertical market profiles from the standards specifications more easily. [56]

PICMG RES also works with SAF, OSDL, SCOPE, CP-TA and the MVA [70].

3.4.3 CP-TA

The Communications Platforms Trade Association (CP-TA) was launched on 27th April 2006 by Adtron, Artesyn, Continuous Computing, Diversified Technology, ESO Technologies, Fujitsu Siemens Computers, HP, Intel, Kontron, MontaVista Software, Motorola, Nortel Networks, Pentair-EP, Pigeon Point Systems and Wind River [5]. TietoEnator is one of the adopter members of CP-TA [53].

In order to develop a COTS components based system, the products from different vendors must be interoperable. CP-TA has the aim to help the telecom companies to make the interoperability testing. The testing requirements “will be developed along with detailed test procedures and supported by industry-harmonized automated test suites and benchmarks” [5]. Currently, CP-TA focuses on the existing standards from PICMG, OSDL and SAF, See Figure 3.10. The standards profiles which CP-TA is using as a foundation for the testing are from SCOPE. [5] CP-TA supplies a compliance mark “CP-TA certified” for the products which pass the compliance test [43].

The reference model and working scope of CP-TA is as follows.

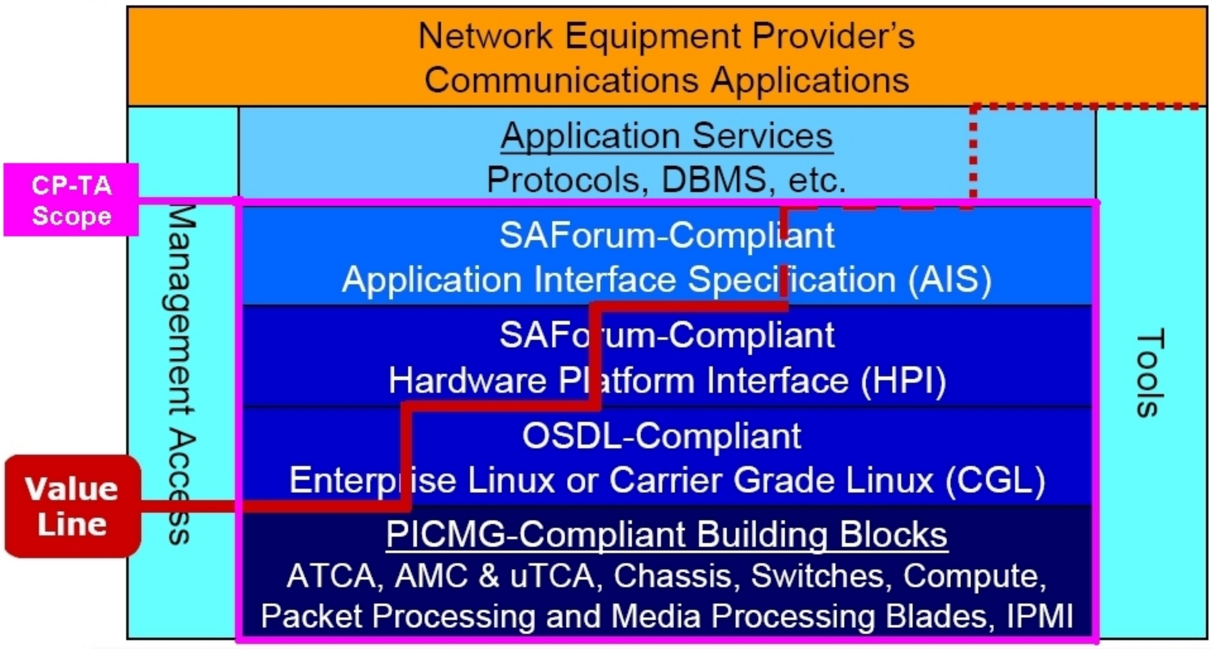


Figure 3.10: CP-TA reference model [45]

3.4.4 MVA

The Mountain View Alliance was founded in June 2005 by the Optical Internetworking Forum (OIF), PICMG and SAF. The RapidIO Trade Association, SCOPE, CP-TA and OCAF joined MVA later. There are more than 500 companies involved with MVA's member organizations. The MVA works for the whole telecom industry which includes equipment manufacturers, service provides, and end users. [22] [7]

The goal of the MVA is to improve the adoption of COTS components and the platforms which are based on the specifications from the members' standards. The MVA works as a coordinator for its members and arranges forums and conferences. The events usually include "education sessions, interactive round table discussions as well as presentations from technology and business leaders from the entire ecosystem" [7]. During the events, the members of the MVA can exchange experience, information and technology. The MVA helps the members to find out the gaps, overlaps, and contradictions in the standards and heartens them to resolve the problems. [7] Figure 3.11 shows the reference model from the MVA.

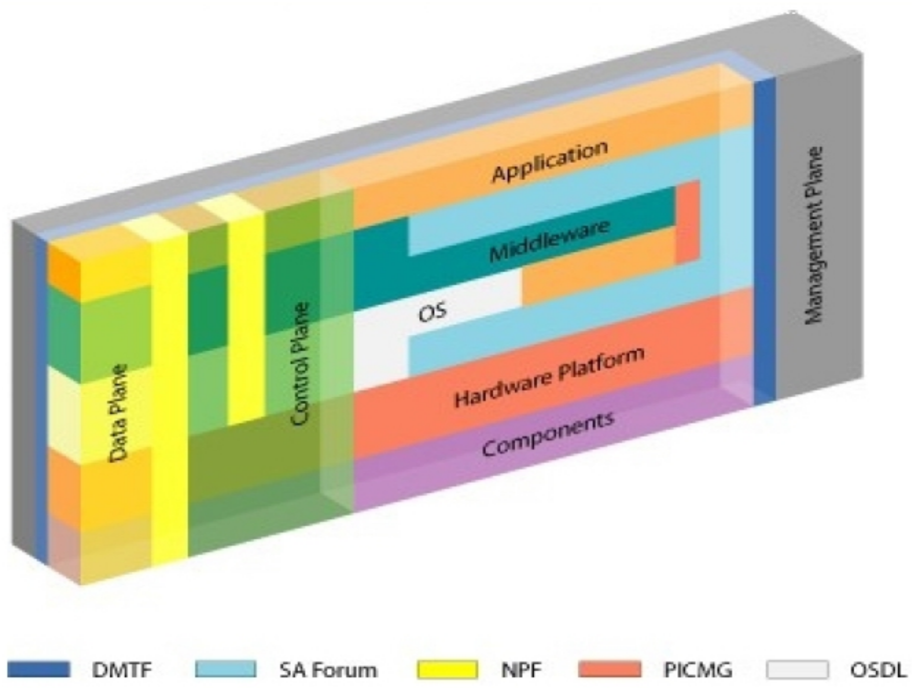


Figure 3.11: MVA reference model [39]

3.4.5 ITU-T OCAF

The Open Communications Architecture Forum (OCAF) Focus Group is a sub group of the International Telecomunication Union–Telecomunication Standardization Sector (ITU-T) and was founded on 20th May, 2004 [57]. ITU-T currently has 350 member companies [49].The members of the OCAF are: Avaya, Cisco, Comcast, Deutsche Telekom, France Telecom, Lucent, IBM, Nortel Networks, NTT, Telecom Italia and Siemens" [41].

The main task of OCAF is to define a common open reference model for the next carrier grade telecom system. The reference model in which a set of COTS components categories for the whole telecom system is defined is called the Carrier Grade Open Environment (CGOE). [41] See Figure 3.12.

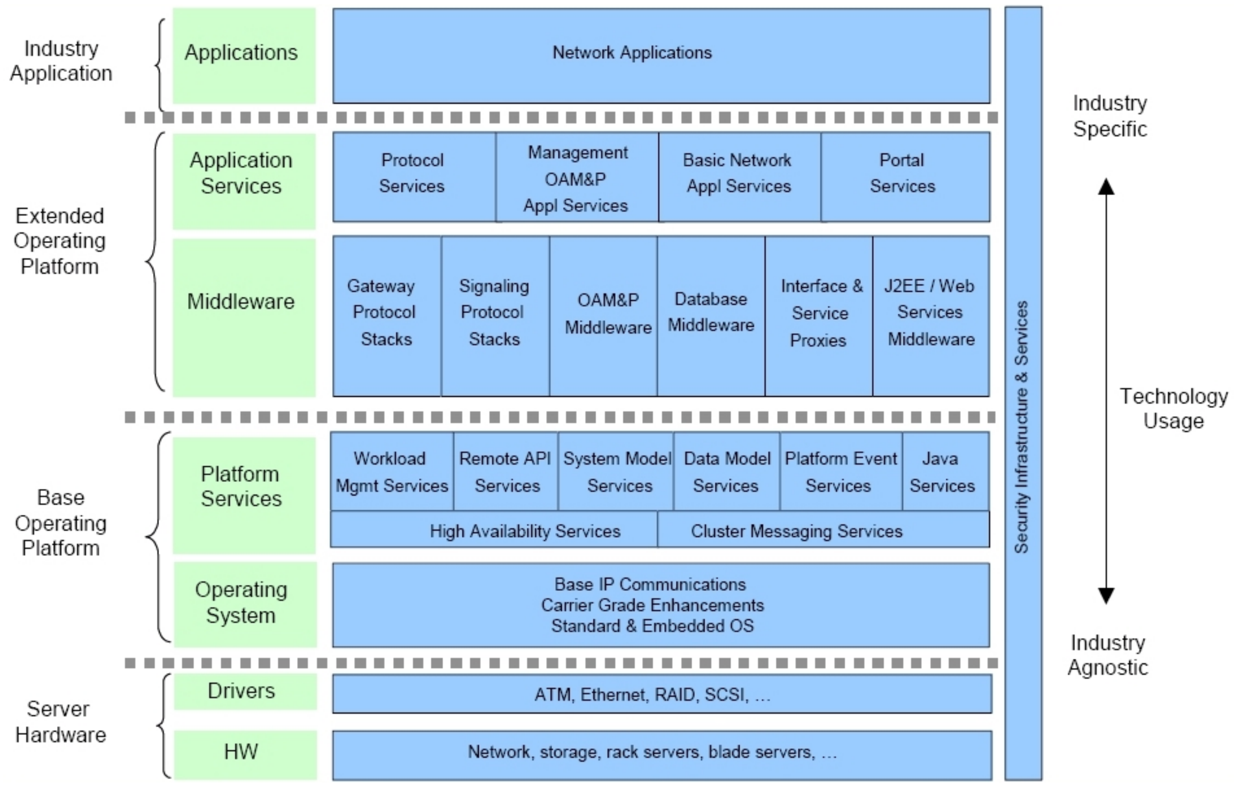


Figure 3.12: OCAF reference model: CGOE [34].

The OCAF has the cooperation with several other organizations, such as SAF and OSDL. [41]

3.4.6 Summary

SCOPE, PICMG RES, CP-TA, MVA, and ITU-T OCAF, all these organizations are newly founded. The main goal of these standards structuring organizations is to increase the interoperability of the components in an open standard based telecom system. These standards structuring organizations identify what is demanded by the industry, collect and test the existing open standards, and make the reference models of the telecom system based on open standards and COTS components. Table 3.6 shows the working scopes of SCOPE, PICMG RES and CP-TA.

<table>
<tr><td></td><td></td><td></td><td>Process Flow</td></tr>
<tr><td rowspan="4">SCOPE</td><td></td><td></td><td>Market Research</td></tr>
<tr><td></td><td></td><td>Profile Creation</td></tr>
<tr><td rowspan="8">PICMG RES</td><td></td><td>Profile Formatting</td></tr>
<tr><td></td><td>Sequencing of Profiles</td></tr>
<tr><td></td><td></td><td>Enumeration of Requirements</td></tr>
<tr><td></td><td></td><td>Formatting of Matrix</td></tr>
<tr><td></td><td></td><td>Population of Matrix Body</td></tr>
<tr><td></td><td></td><td>Distillation of Outputs</td></tr>
<tr><td></td><td rowspan="7">CP-TA</td><td>Test Plan Creation</td></tr>
<tr><td></td><td>Test Script Creation</td></tr>
<tr><td></td><td></td><td>Test Lab Construction</td></tr>
<tr><td></td><td></td><td>Test Execution</td></tr>
<tr><td></td><td></td><td>Issue Certifications</td></tr>
<tr><td></td><td></td><td>Outbound Marketing</td></tr>
</table>

Table 3.6: Interoperability Specification Process Flow [70]

In Table 3.6, SCOPE and PICMG RES mainly focus on the requirements design and CP-TA focuses on testing. The difference between SCOPE and PICMG RES is: SCOPE does the market researching and compares the market requirements with the standards specifications, but PICMG RES mainly focuses on the profiles from the standards specifications and tries to format them to the vertical market profiles.

MVA organizes events to help the related organizations having the chance to discuss together. OCAF makes a reference model describing the building blocks for a typical carrier grade telecom system.

3.5 Summary

The open standards from PICMG, OSDL CGL WG and SAF describe the hardware platform, the OS platform and the SA middleware respectively in a telecom system. The technical properties of the COTS components can be found from the open standards specifications. Figure 3.13 shows a part of the system reference model which is based on the open standards mentioned in this dissertation.

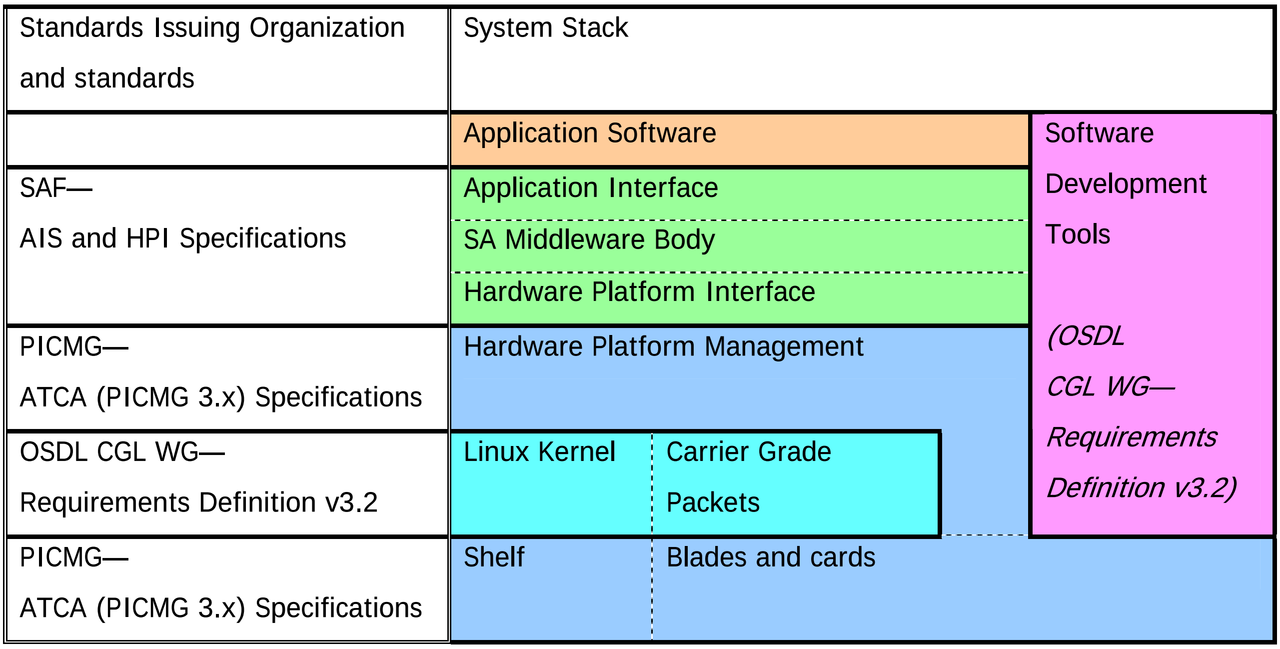


Figure 3.13: System reference model with the ATCA, CGL and SA middleware components

Apart from the standards issuing organizations (PICMG, OSDL CGL WG, SAF), several other organizations are working on analyzing, testing, and structuring the existing standards, for example: SCOPE, PICMG RES, CP-TA, MVA, and ITU-T OCAF. Both the standards issuing organizations and the standards structuring organizations are working together to improve the telecom industry toward the open standard system. Table 3.7 shows that there exist relationships between the organizations mentioned in this dissertation. “Y” means there are some kinds of relationships between the organizations, and the relationships can be for example: member, standards user, cooperate.

	Standards Organization	PICMG	OSDL CGL WG	SAF	SCOPE	PICMG RES	CP-TA	MVA	ITU-T OCAF
Issuing	PICMG	—	Y	Y	Y	Y	Y	Y	
	OSDL CGL WG	Y	—	Y	Y	Y	Y	Y	Y
	SAF	Y	Y	—	Y	Y	Y		Y
Structuring	SCOPE	Y	Y	Y	—	Y		Y	
	PICMG RES	Y	Y	Y	Y	—	Y	Y	
	CP-TA	Y	Y	Y		Y	—	Y	
	MVA	Y	Y		Y	Y	Y	—	Y
	ITU-T OCAF		Y	Y				Y	—

Table 3.7: Existence of relationships between the different standards related organizations

4 Components information

4.1 The system reference model

In order to build a system with different components, the system reference model must be defined first.

Each organization and system supplier has their own view of the system reference model. The reference model diagrams from PICMG, OSDL CGL WG and SAF respectively focus on the hardware platform, OS, and SA middleware but not on the entire system. Since SCOPE is mainly working on analysing the standards from PICMG, OSDL CGL WG at present, the scope of the reference model from SCOPE is restricted to the hardware platforms including the hardware, the Linux OS and SA (service availability) middleware. PICMG RES which is the sub group of PICMG covers the almost the same issues as SCOPE but focuses on the hardware part. The model diagrams from CP-TA and MVA only divide the system into several layers and are too simple. The OCAF CGOE model describes the entire carrier grade system with example building blocks. The system reference model which is used for the COTS component database for TietoEnator is mainly a synthesis of the reference models from OCAF (Figure 4.1) and other standards related organizations (Figure 3.2, Figure 3.3, Figure 3.6, Figure 3.7, Figure 3.9, Figure 3.10, Figure 3.11).

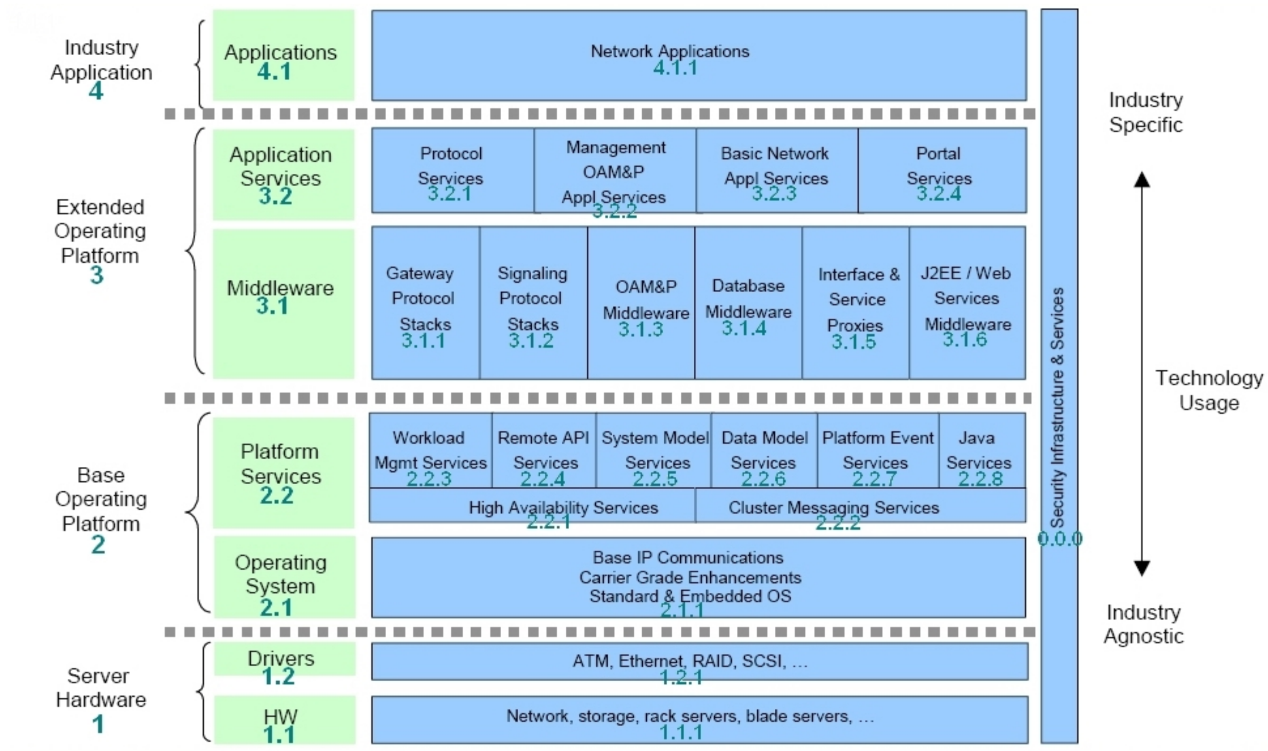


Figure 4.1: System reference model—OCAF CGOE [34]

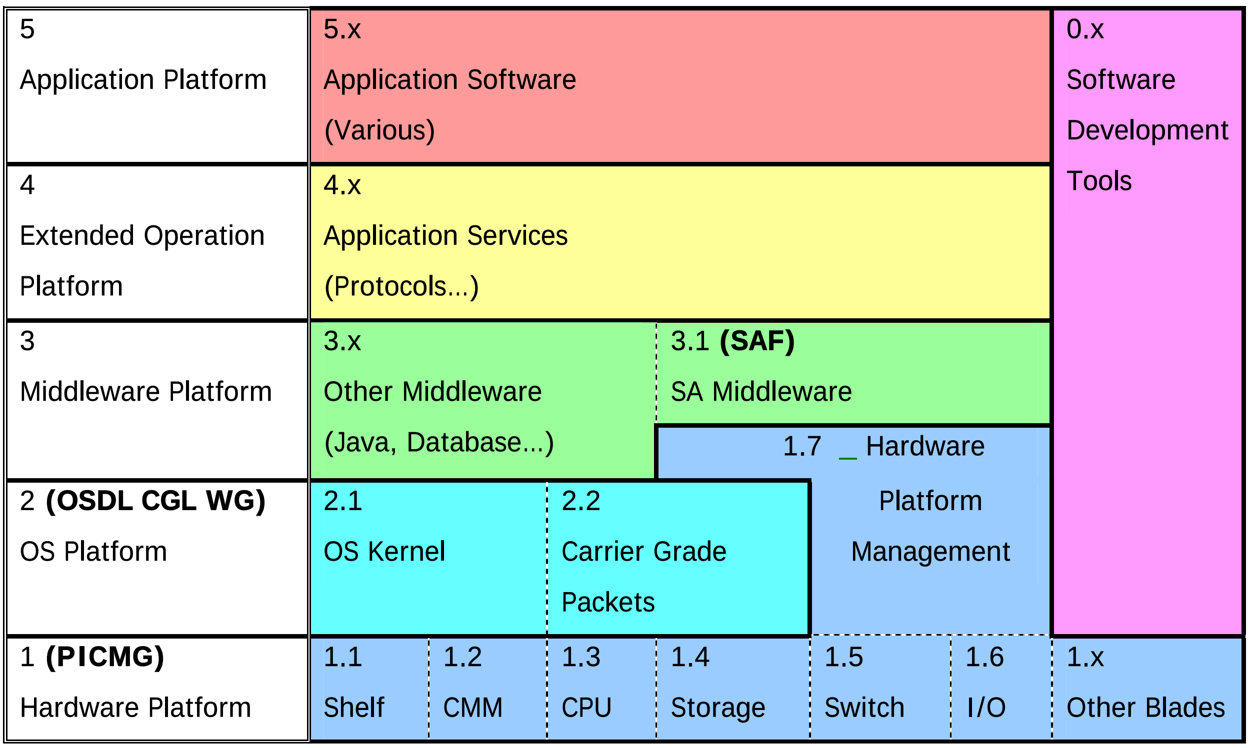


Figure 4.2: System reference model—Database

The CGOE model (Figure 4.1) is a clear model in that it divides the telecom system into different functionalities and services. Since the reference model which is used for the COTS component database (Figure 4.2) is based on the COTS components, the building blocks in the model should be mainly based on the real industry products. The views of the system model are not really the same between CGOE and the database user. The mappings between the CGOE model (Figure 4.1) and the database model (Figure 4.2) are explained below.

The hardware block (Figure 4.1, 1.1) in the CGOE model contains a set of COTS components which are different blocks in the database model (Figure 4.2, 1.1-1.x). For example, the hardware can be divided into shelf and blades; examples of the blade components are CMM, CPU, storage, switch, and I/O. The vendors sometimes separate the blade to card and board, and for this case, each card or board is one COTS component and belongs to one system block as well. Since the drivers of the hardware come together with the hardware, the vendees do not need to buy the drivers separately. Therefore the drivers are not considered as real COTS components. Thus the database model does not present the drivers as a single block (Figure 4.1, 1.2) but includes them in the hardware blocks (Figure 4.2, 1.1—1.x).

One further example is the “signaling”. In the model from CGOE, signaling (Figure 4.1, 3.1.2) belongs to software, but in the real market, signaling sometimes belongs to hardware since it is sold as a blade and the user can not obtain the software if they do not buy the hardware. In this case, the signaling blade is in the block 1.x of Figure 4.2. [26]

One component (building block in the database model) may support a set of system functionalities and services, which means one building block in the database model may contain a set of CGOE blocks. For example, the SA middleware in the database model is represented as one system block (Figure 4.2, 3.1) but is represented as two blocks in the CGOE model (Figure 4.1, 2.2.1, 2.2.2).

Certain software is not considered as a COTS component but as an open source component, for example, Java, MySQL database and most of the protocols. The database model synthesizes the models from both CGOE and other standards related organizations, and defines the building blocks 3.x, 4.x, 5.x and 0.x in Figure 4.2. The detailed information is as follows:

Figure 4.2, 3.x = Figure 4.1, 3.1

Figure 4.2, 4.x = Figure 4.1, 3.2

Figure 4.2, 5.x = Figure 4.1, 4.1

Figure 4.2, 0.x = Figure 3.6, Software Development Tools

4.2 Standards

The standards which are stored in the database contain the definition of the structure, the interfaces, the requirements, the functionalities and the services of the components. Not all the COTS components are 100% compliant with the standards. Sometimes the vendee needs to compare the detailed parameters between the component datasheet and the standards specifications.

The standards can also be a requirement of a component in the database. For example some of the SAF components only work on the ATCA components. The compatibility between different standards compliable components can be found from the standards specifications as well.

One component can be based on a set of standards specifications, as for example most SAF components. One component can also be based on a specific part of one standard specification, such as most ATCA components.

The current standards cover different layers of the whole telecom system. The standards for hardware and the standards for software contain different types of the components attributes. The hardware standards have more physical attributes and most of them can be described by figures or very short key words. The short descriptions of the attributes are very suitable for a database to manage, such as index, order by, group by, and obtain statistics.

Table 4.1 (a synthesis of Table 3.1, Table 3.3, Table 3.4) is an example of the standards list with the basic information which can be kept in the database.

Standard Issuing Organization	Standard Name	Standard Version	System Block
OSDL/CGL WG	Carrier Grade Linux Availability Requirements Definition	CGL RD v3.2	OS
OSDL/CGL WG	Carrier Grade Linux Clustering Requirements Definition	CGL RD v3.2	OS
OSDL/CGL WG	Carrier Grade Linux Hardware Requirements Definition	CGL RD v3.2	OS
OSDL/CGL WG	Carrier Grade Linux Performance Requirements Definition	CGL RD v3.2	OS
OSDL/CGL WG	Carrier Grade Linux Requirements Definition Overview	CGL RD v3.2	OS
OSDL/CGL	Carrier Grade Linux Security Requirements	CGL RD v3.2	OS

WG	Definition		
OSDL/CGL WG	Carrier Grade Linux Serviceability Requirements Definition	CGL RD v3.2	OS
OSDL/CGL WG	Carrier Grade Linux Standards Requirements Definition	CGL RD v3.2	OS
PICMG	AdvancedTCA Base	PICMG3.0 R2.0	Hardware platform
PICMG	AdvancedTCA Ethernet	PICMG3.1 R1.0	Hardware platform
PICMG	AdvancedTCA InfiniBand	PICMG3.2 R1.0	Hardware platform
PICMG	AdvancedTCA StarFabric	PICMG3.3 R1.0	Hardware platform
PICMG	AdvancedTCA PCI Express	PICMG3.4 R1.0	Hardware platform
PICMG	AdvancedTCA RapidIO	PICMG3.5	Hardware platform
PICMG	AdvancedTCA PRS	PICMG3.6	Hardware platform
SAF	Hardware Platform Interface: Specification	SAI B.01.01	SA Middleware
SAF	Distributed Systems Management for HPI-SNMP	SAI B.01.01	SA Middleware
SAF	HPI-to-AdvancedTCA® Mapping Specivication	SAI B.01.01	SA Middleware
SAF	Service Availability Interface: Overview	SAI B.02.01	SA Middleware
SAF	Availability Inrerface Specification: Availability Management Framework Service	SAI B.02.01	SA Middleware
SAF	Availability Inrerface Specification: Checkpoint Service	SAI B.02.01	SA Middleware
SAF	Availability Inrerface Specification: Cluster Membership Service	SAI B.02.01	SA Middleware
SAF	Availability Inrerface Specification: Event Service	SAI B.02.01	SA Middleware
SAF	Availability Inrerface Specification: Lock Service	SAI B.02.01	SA Middleware

SAF	Availability Interface Specification: Message Service	SAI B.02.01	SA Middleware
SAF	Availability Inrerface Specification: Information Model Management Service	SAI A.01.01	SA Middleware
SAF	Availability Interface Specification: Log Service	SAI A.01.01	SA Middleware
SAF	Availability Interface Specification: Notification Service	SAI A.01.01	SA Middleware
SAF	Distributed Systems Management for AIS-SNMP	SAI A.01.01	SA Middleware

Table 4.1: An example of the standards list

4.3 Database component definition and attributes

The components which are used to build a telecom system are varied, but the components which are planned to be kept in the COTS component database are particular and need to have some special attributes, including suppliers, list prices, complying standards, system blocks, component name and version, system functionalities and services, and requirements.

4.3.1 COTS components

Each component in the COTS component database corresponds to a product and is purchasable, and can usually be found on the internet. The datasheet which contains the information about the component is available and is easily accessible for the vendees. One component in the database is usually produced by one manufacturer and can be bought from several suppliers. The price of a component depends on the different suppliers and may change over time.

4.3.2 System blocks

The components in the database are used to build a telecom system, thus each component has to provide some functionalities and services which relate to a telecom system. A typical telecom system stack includes hardware, operating system, middleware, and other application software. The hardware and the software can be divided into more hierarchical sub-stacks/blocks. The components which are kept in the database correspond to the detailed blocks which are functionally identifiable in a telecom system.

4.3.3 Compliant to standards

Since the database is used for the standardized platform integration, the components in the database are mostly complying with some standards. The current components collection mainly uses the standards from the standards issuing organizations which this dissertation has mentioned before, such as the PICMG (ATCA), the OSDL CGL WG (CGL RD v3.2), and the SAF (HPI&ASI).

4.3.4 Other attributes and requirements

Apart from the vendors and their list prices, the position in a system, and the standard compliant status, each component in the database has more attributes, such as the specific functionalities and services, the interfaces, the detailed compositions, and the run-time environment requirements. All these attributes and requirements will be kept in the database.

4.3.5 Components datasheets

The information about a component which is used to fill the database is mainly taken from the component datasheet. Examples of component datasheets are presented in appendix B.

To find a component on the internet, the vendor corporations, the system blocks, and the related standards can all be searched for to help gain access to a particular component.

The structure and the content of the components datasheets are presented in different formats.

Some of the component datasheet does not have much information about the component attributes which can be kept in the database, such as the datasheet of the "Solaris Operating System" from Sun [75].

Some of the components datasheets include a lot of information but not all of them will be kept in the database tables directly. The database is used for searching and comparing components, thus the information which is about the internal structure and how to use the component, will not be kept in the database. The information in the database mostly focuses on the external information of a component, such as the interfaces parameters.

For example, the datasheet of "AXP1600 AdvancedTCA Shelf" from Motorola (Figure 4.3) [32] contains the information "BLOCK DIAGRAM" (Figure 4.4) which will not be kept in the database tables, and also contains the "SHELF HARDWARE" (Figure 4.5) attributes which will be kept in the database. See the photo cut from the datasheet of "AXP1600 AdvancedTCA".

Figure 4.3: AXP1600 AdvancedTCA Shelf datasheet [32]

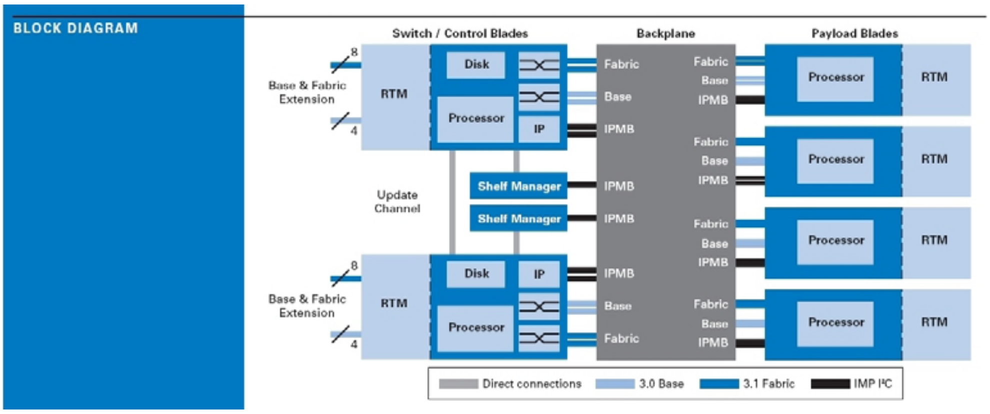
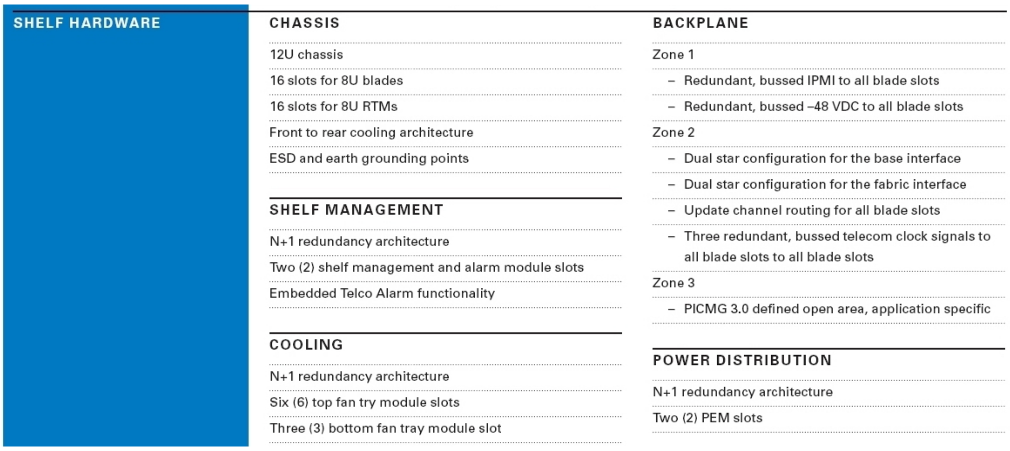


Figure 4.4: AXP1600 AdvancedTCA Shelf datasheet: BLOCK DIAGRAM [32]



SHELF HARDWARE

CHASSIS
- 12U chassis
- 16 slots for 8U blades
- 16 slots for 8U RTMs
- Front to rear cooling architecture
- ESD and earth grounding points

SHELF MANAGEMENT
- N+1 redundancy architecture
- Two (2) shelf management and alarm module slots
- Embedded Telco Alarm functionality

COOLING
- N+1 redundancy architecture
- Six (6) top fan try module slots
- Three (3) bottom fan tray module slot

BACKPLANE

Zone 1
- Redundant, bussed IPMI to all blade slots
- Redundant, bussed –48 VDC to all blade slots

Zone 2
- Dual star configuration for the base interface
- Dual star configuration for the fabric interface
- Update channel routing for all blade slots
- Three redundant, bussed telecom clock signals to all blade slots to all blade slots

Zone 3
- PICMG 3.0 defined open area, application specific

POWER DISTRIBUTION
- N+1 redundancy architecture
- Two (2) PEM slots

Figure 4.5: AXP1600 AdvancedTCA Shelf datasheet: SHELF HARDWARE [32]

4.4 Summary

Methods for classifying the components in the database vary. The main ideas are to classify the components by the vendors, the standards, and the system blocks.

The way to classify the components by the vendors has certain advantages. The components in different system blocks may have a better interoperability between each other when they are from the same manufacturer, and the components in the same system block may have the same datasheet format which allows the user to easily compare them when they are from the same supplier. The price of the components may be lower if the vendees buy a set of components from the same vendee.

Another way to classify the components is by standards. The systems built by COTS components are used for different purposes, such as communications server, media server, switch, and router [70]. Each standard focuses on certain types of the systems, such as the standards from PICMG: “AMC.0 R1.0” is used for defining the Advanced Mezzanine Card (AMC), and “MicroTCA” will be used for defining the low cost infrastructure for AMCs [42]. Classifying the COTS components by standards will help the user to obtain the specific components for different interest areas of the system.

Classifying the components by the system blocks is the primary method. The components that have similar system functionalities and services always have the similar attributes, and the datasheets of the components may have the similar structure and the similar fields. Grouping the components by their system stacks can help the user to compare similar components, especially for hardware components.

These three classification methods can be used both individually and together. In the database, in general, more than one way is used to select the components.

5 Database design

5.1 Database structure

There are four modules in the database and each module contains one basic table and a set of additional tables. These four modules are the organization module, the standard module, and the system block module, and the component module. Each module has one main table with general information which contains the identity fields. The information of a COTS component can be divided into these four modules. These four modules have relationships between each other. In Figure 5.1, the boldface type shows the main primary keys in each module.

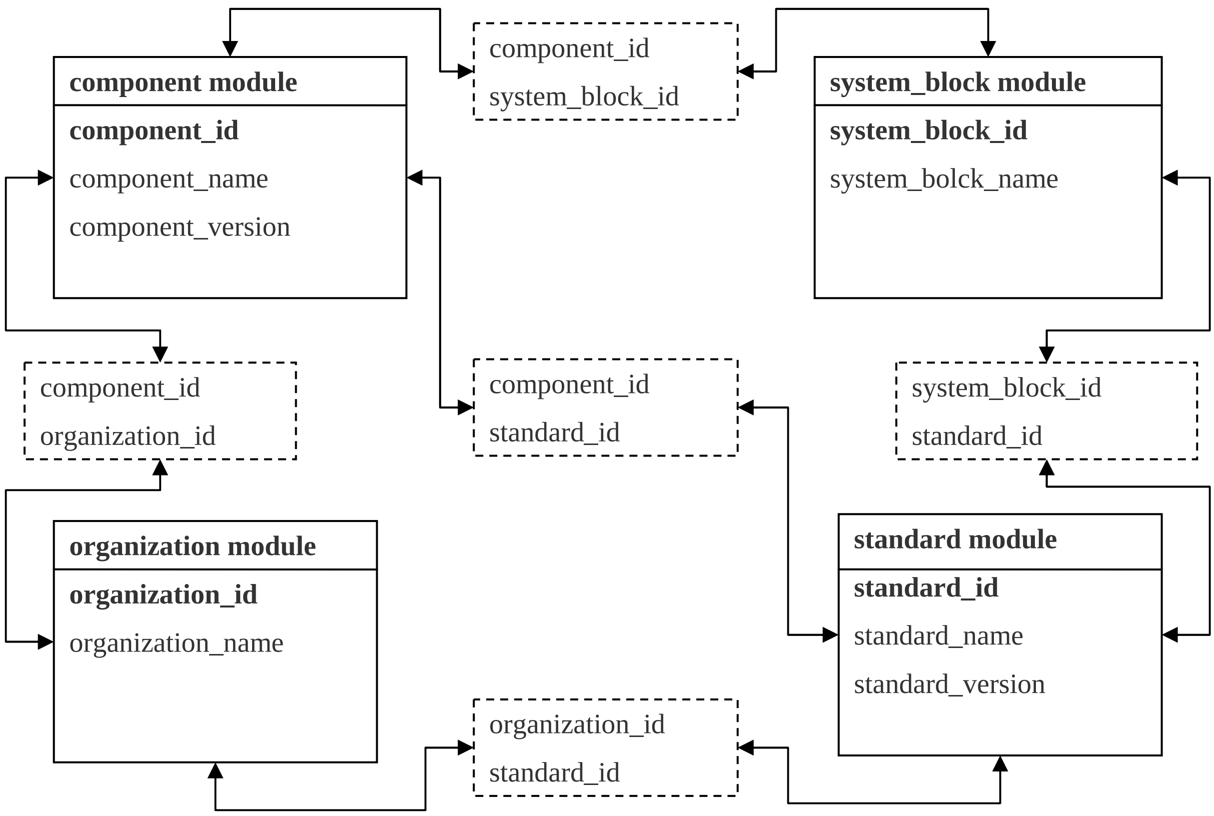


Figure 5.1: Database structure

5.2 Organization module

The organization module is used to keep the information of all the organizations, such as the components manufacturers, the components franchisers and the standards related organizations.

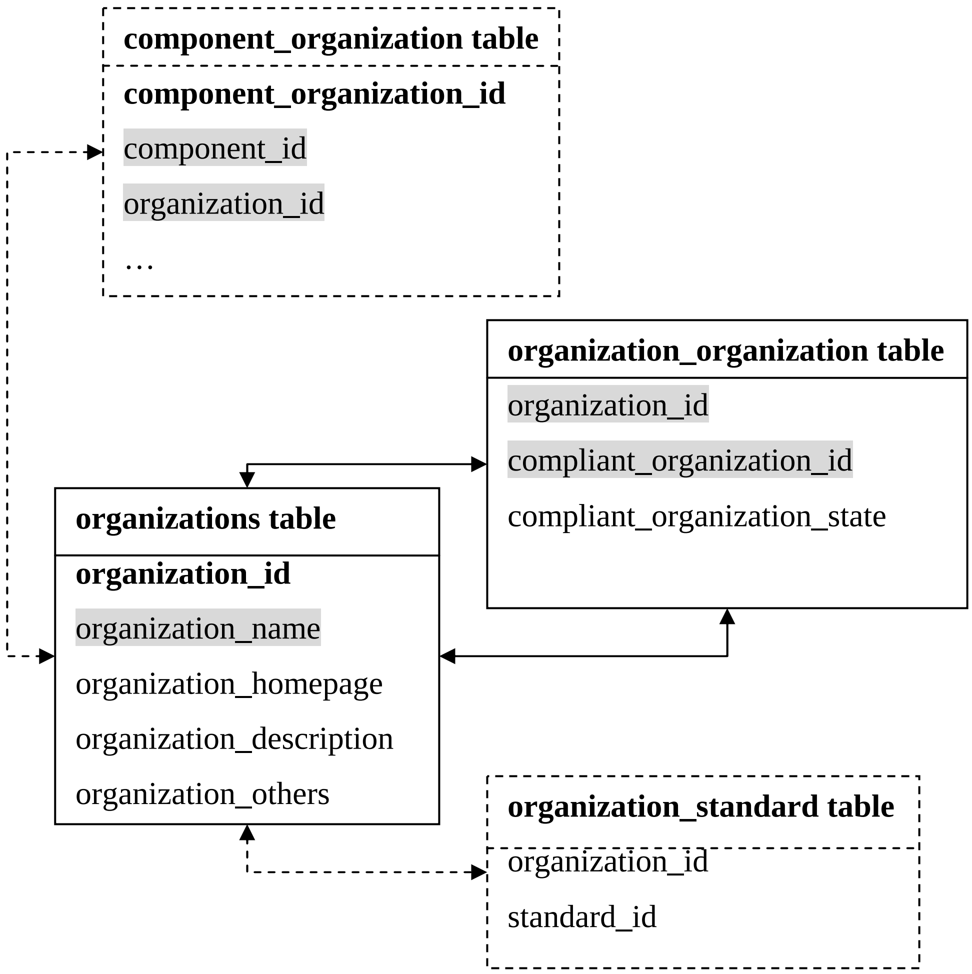


Figure 5.2: Organization module

The organization module has four tables and relates to the standard module and the component module. The organization id is the primary key, and it relates to the organization name which has to be unique in the organizations table.

The organizations table has fields giving the general description and the contact information of each organization.

The organization organization table is used to show the relationships between organizations. For example, one standard publisher has a set of member companies. Intel is the member of SAF, CGL WG, and PICMG, and the membership can be executive and associate. Another example is that sometimes the manufacturer does not sell the components by themselves. The component “Promentum ATCA-6000 - 12U 14-Slot ATCA Shelf” made by Intel can not be bought from Intel directly, vendees can only order if from another company which is called RadiSys [68].

The component organization table is used to show the relationships between a component and an organization. One component can be made by a set of manufacturers and be sold by several vendors.

The organization standard table is used to show the relationship between the standards and the standards issuing organizations.

5.3 Standard module

The standard module is used to keep the information of the component compliant standards.

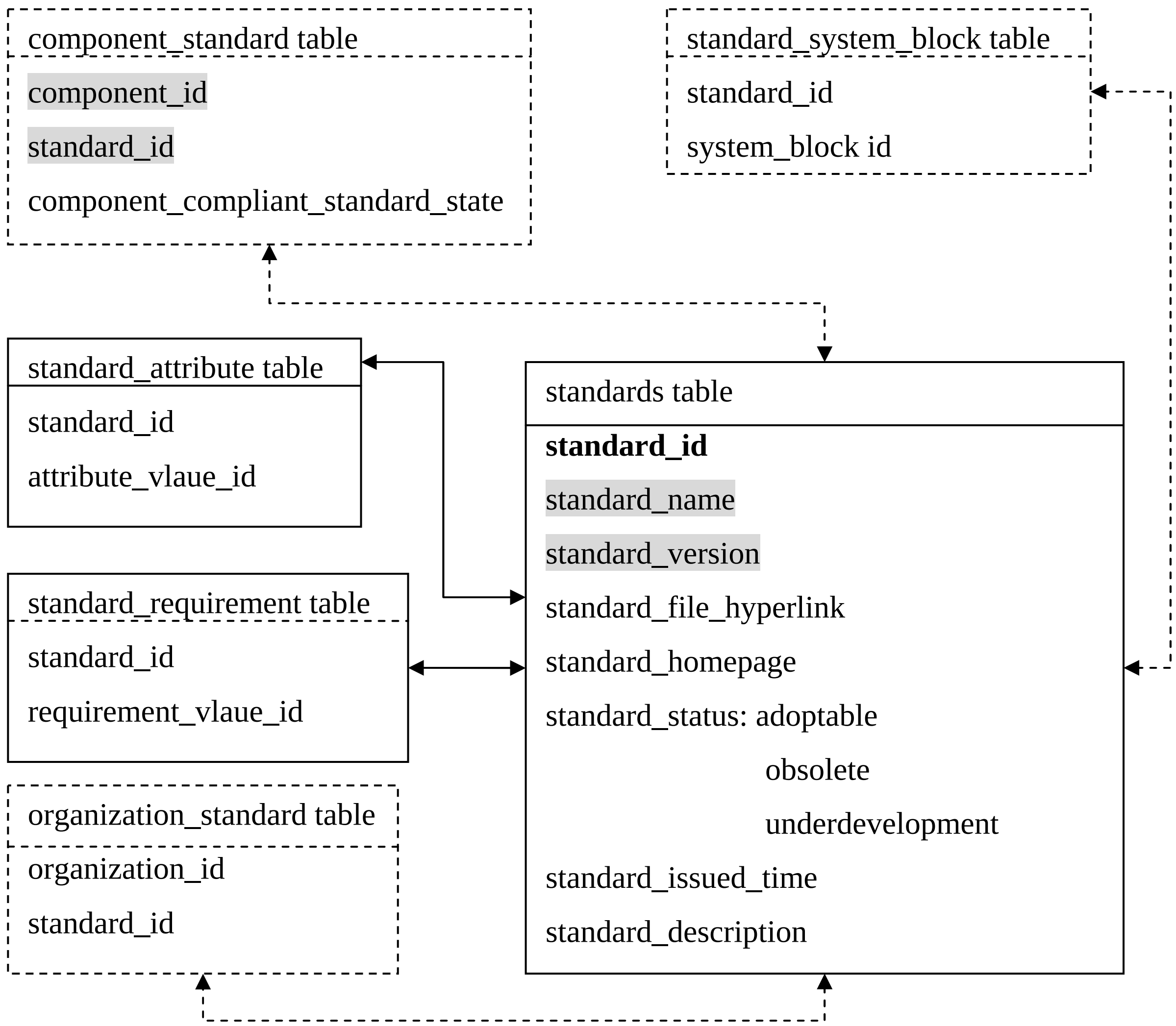


Figure 5.3: Standard module

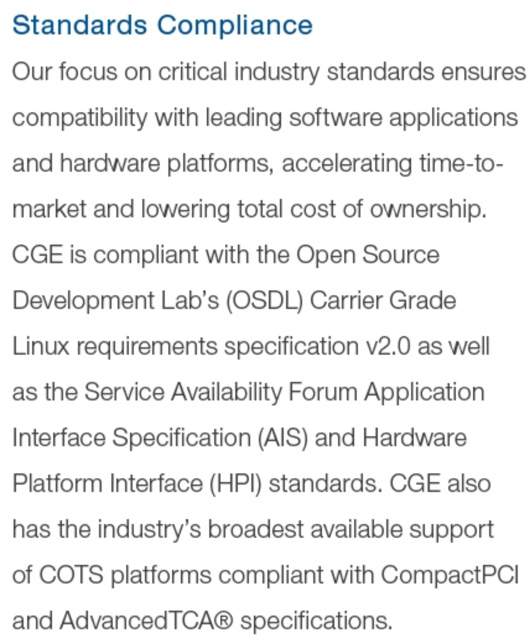
The main table in the standard module is the standards table. The primary key is the standard id, and it relates to both standard name and standard version.

The standard table only keeps the general information about the standards. To access the standards specifications, there are two ways:

1. One is accessing the standard homepage.
2. Another one is using the hyperlink to reach the documents which is stored in the database.

The standard system block table shows the relationship between the standard and the system layers.

The component standard table shows how the components compile to different standards. For example, the standards compliance information from the datasheet of "MontaVista Linux Carrier Grade Edition 4.0" is as following [55].



Standards Compliance

Our focus on critical industry standards ensures compatibility with leading software applications and hardware platforms, accelerating time-to-market and lowering total cost of ownership. CGE is compliant with the Open Source Development Lab's (OSDL) Carrier Grade Linux requirements specification v2.0 as well as the Service Availability Forum Application Interface Specification (AIS) and Hardware Platform Interface (HPI) standards. CGE also has the industry's broadest available support of COTS platforms compliant with CompactPCI and AdvancedTCA® specifications.

Figure 5.4: MontaVista 4.0 datasheet: standard compliance [55]

The standards issuing organizations can be searched for through the organization standard table.

Since the standard is used to describe how to make a component, some of the component attributes, services, requirements, and interfaces information can be found from the standards. The standard attribute table and the standard requirement table keep the standard attribute and requirement. For example, an ATCA shelf should have 14 or 16 slots and supports each blade 48V DC [28]. The information placed in the standard attribute table and the standard requirement tables are mainly from the standards specifications. Some of the standards structuring organizations such as SCOPE and PICMG RES provide core data from the standards which can be used in this table as well.

5.4 System block module

The system block module is used to keep the information about the blocks in a telecom system and it has one basic table and two sub tables. The system block module relates to the component module and the standard module.

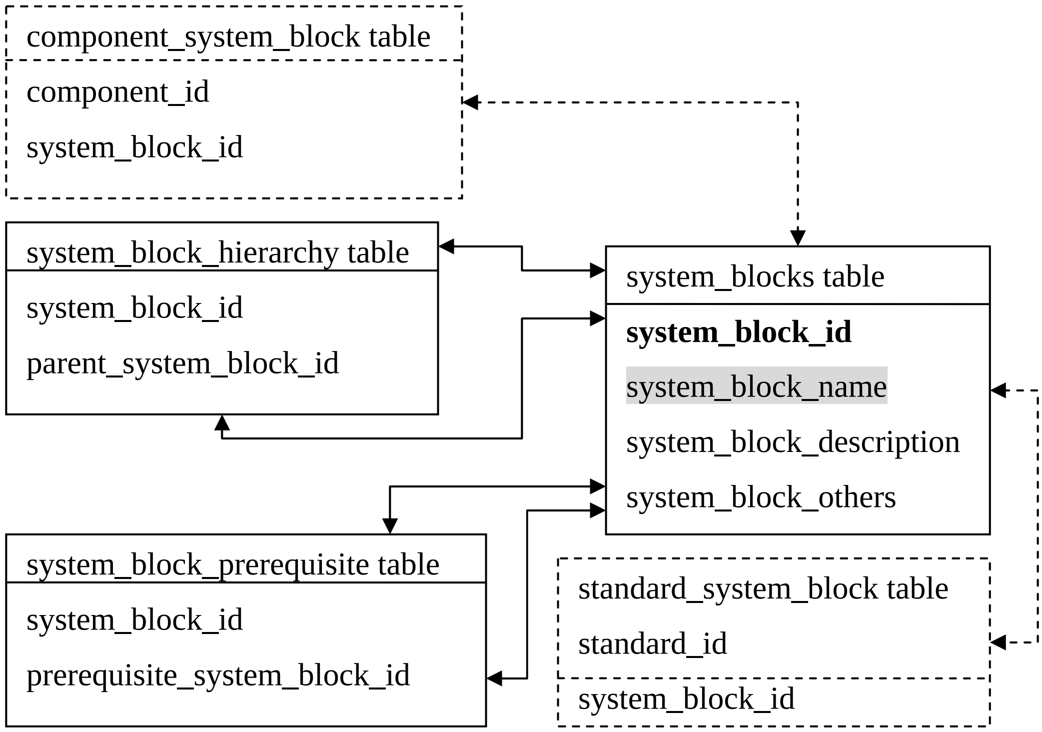


Figure 5.5: System block module

The system block table is the main table in the system block module which is used to keep the name and the description of the system blocks. This design is suitable for the various views of the system reference model. The primary key in the system block table is the system block id which relates to the system block name. The system block name is only the simple words, such as hardware, shelf, CPU, operation system, and SA middleware.

The component system block table shows which part of the system the component belongs to.

The standard system block table shows which part of the system the standards focus on.

The system block hierarchy table is used to keep the inclusive relationship between the system blocks. The system block hierarchy table uses the parent and child structure to present the relationship between the block and the sub blocks. For example, “hardware” is the parent system block of “shelf” and “blade”. Most of the time, the blocks with the same parent block are on the same system layer.

The system block prerequisite table is used to keep the relationship between the system blocks which are from different system layers. Here the database requires that the block from the lower system layer is the prerequisite block of the block from the higher system layer. For example, hardware is the prerequisite block of operating system, and shelf is the prerequisite block of the blades. The interface attributes of the prerequisite block maybe the interface requirements of the upper layer block, for example, an ATCA shelf supports 48V DC for each slot and 48V DC is the requirement power of the blades.

5.5 Component module

5.5.1 Component module

The component module is used to keep the information of the COTS component, and it is the main module in the database.

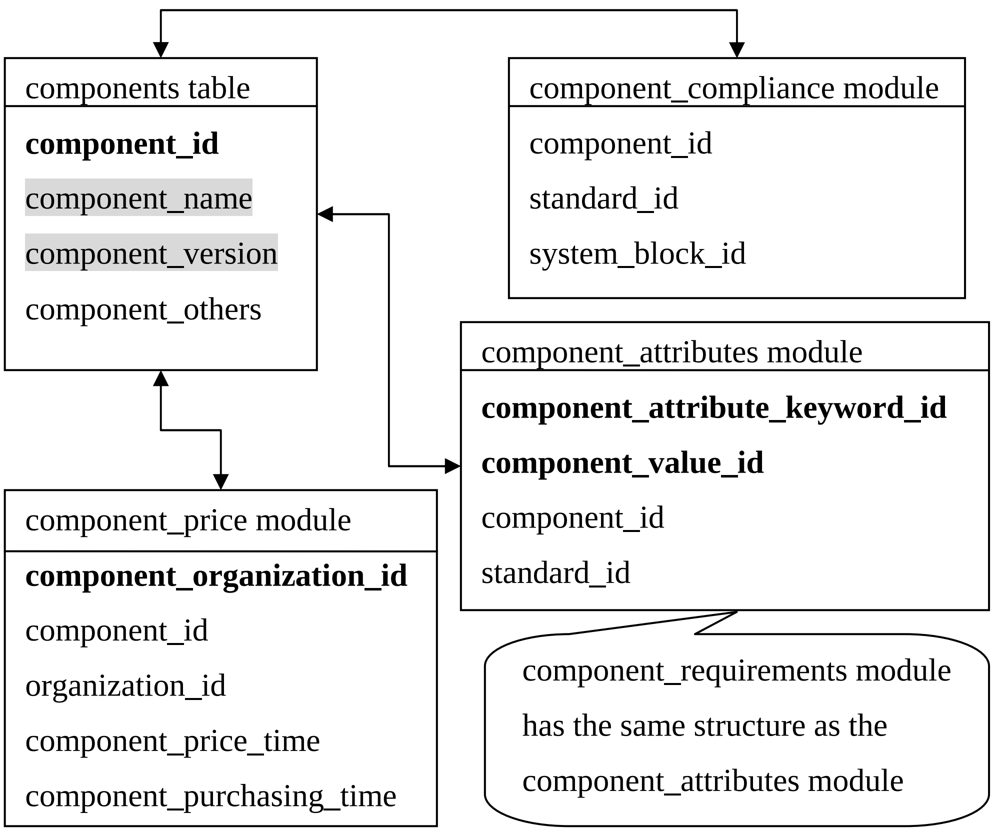


Figure 5.6: Component module

The component module has one basic table and three sub modules: component price module, component attributes module, component compliance module. The component module has relationships with all the other modules in the database and it stores all the information of the component.

The basic table in the component module is the component table, which is used to keep the general information about a component. The component name and the component version are the main fields in the component table and the primary key is the component id.

The information about a component such as the system function and the compliant standards which relates to the system stack module and the standard module are kept in the component compliance sub module.

The other attributes of the component such as the requirement and some more detailed attributes are stored in the component attributes sub module.

The component price sub module is used to show the relationships between component and organizations and the price of the component.

5.5.2 Component price module

The component price module relates to the organization module.

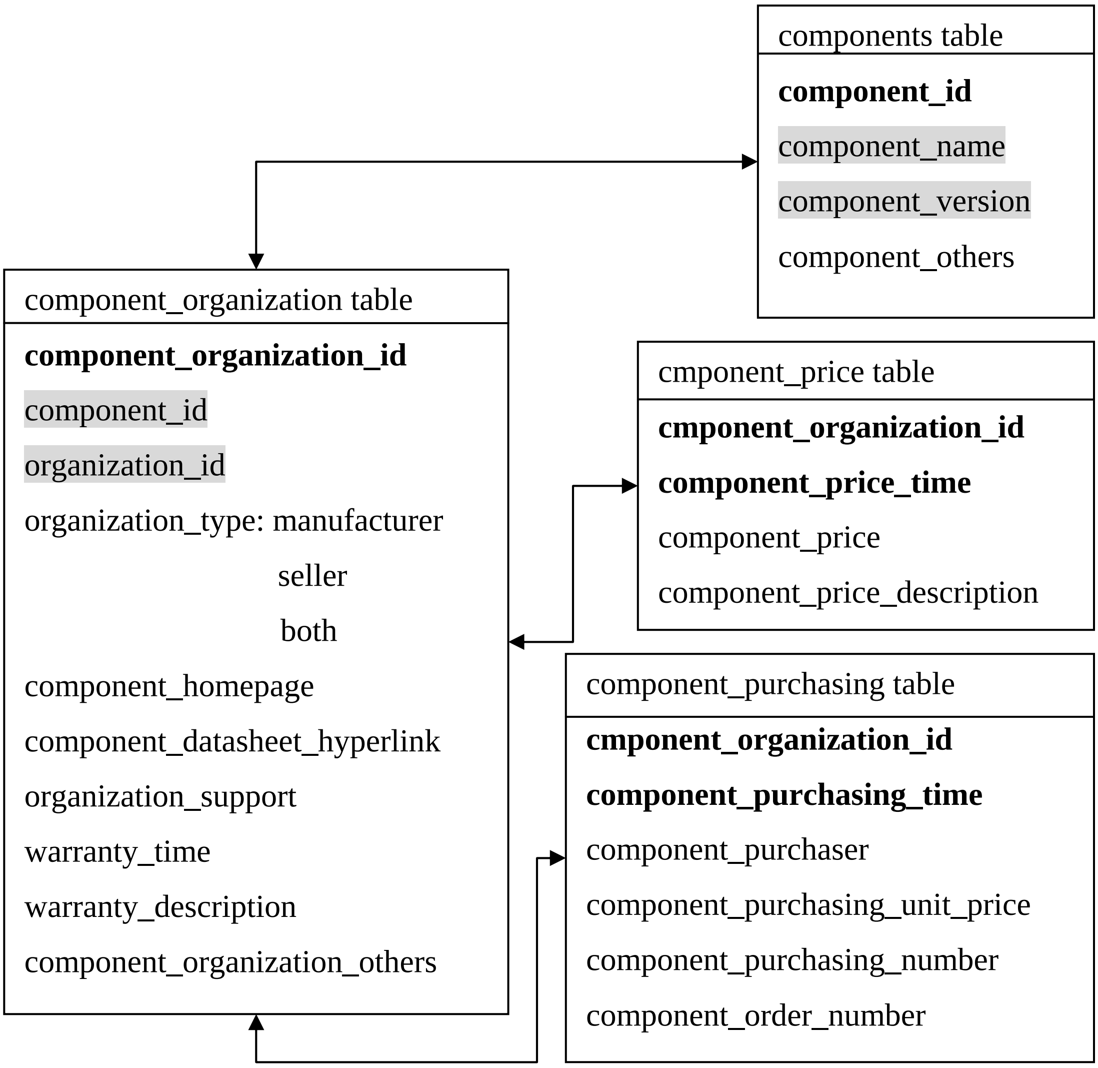


Figure 5.7: Component price module

Since the component can be sold by different sellers, the services and the prices from different sellers maybe different. The same component from different vendors is created as a new entity in this module.

The main table in the component module is the component organization table. The primary key is the component organization id, and it relates to both the component id and the organization id.

The component organization table keeps the information of the support from the seller, such as the “warranty” information.

The links to the component datasheets are kept in the component organization table as well. The way to get the component datasheets are the same as to get the standards specifications:

1. One is accessing the component homepage from different sellers.
2. Another one is using the hyperlink to reach the documentation from the database.

The price of the same component may be different from different sellers. The price of the same component from one seller is changing over time as well. The component price table is used to keep all the price information:

1. List the price from different sellers.
2. List the price changing from one seller by time.

The component purchasing table is used to keep the information about the components which have already been bought before, such as the time of purchase, people who bought them, price, and number.

5.5.3 Component compliance module

The component compliance module is used to keep the relationships: component—component, component—standards, component—system blocks.

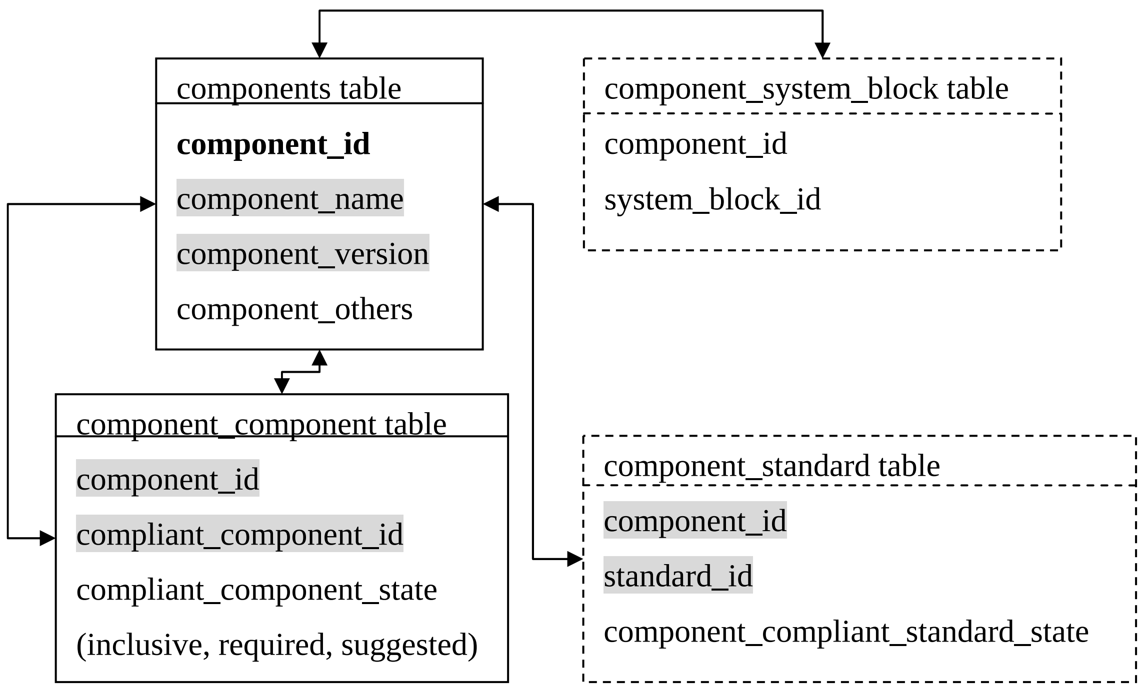


Figure 5.8: Component compliance module

There are at least three kinds of relationships between two components: required, optional, and recommended. See the following examples.

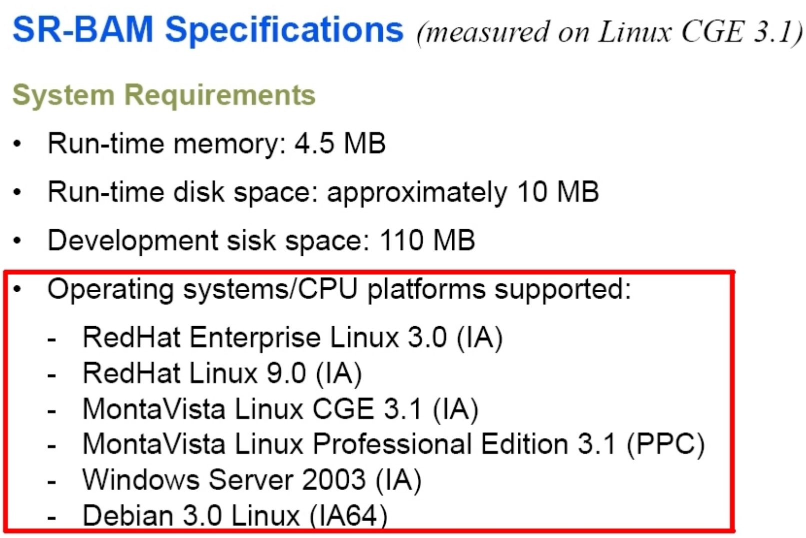
Required: the COTS component actually is a package with a set of components. An example is as following.

1) COTS component: Motorola Centellis 3600 AdvancedTCA System Platform [35]
2) Inclusive components:
   A. AXP1600 AdvancedTCA Shelf [32]

      "ORDERING INFORMATION: The AXP1600 shelf is sold as part of a complete Centellis 3600 platform, therefore not sold separately. Please contact your local Motorola representative to custom configure your Centellis 3600 platform."[14]

   B. ATCA-7221,R3: AdvancedTCA Processor Blade [30]
   C. ATCA-S100,R3: AdvancedTCA Storage Blade [31]

Optional: the components are from different system layers and the required component is from the requirements part of the datasheet. For example: SelfReliant Basic Availability Management (SR-BAM) [44]. See the following photo cut from the datasheet of the middleware from GoAhead.



SR-BAM Specifications *(measured on Linux CGE 3.1)*

System Requirements

- Run-time memory: 4.5 MB
- Run-time disk space: approximately 10 MB
- Development sisk space: 110 MB
- Operating systems/CPU platforms supported:
  - RedHat Enterprise Linux 3.0 (IA)
  - RedHat Linux 9.0 (IA)
  - MontaVista Linux CGE 3.1 (IA)
  - MontaVista Linux Professional Edition 3.1 (PPC)
  - Windows Server 2003 (IA)
  - Debian 3.0 Linux (IA64)

Figure 5.9: SR-BAM datasheet: system requirements [44]

In this case, the OS components (RedHat, MontaVista, Windows Server, Debian) which are mentioned in the datasheet requirement part are the required components of the GoAhead middleware component “SR-BAM”.

Recommended: the components from the same manufacturer always have better interoperability and the manufacturer will give the consumer some kind of suggestion about the other components from the same company. For example, at the end of the datasheet of “Promentum ATCA-6000 - 12U 14-Slot ATCA Shelf” [68], it gives the suggestion as following:

“The shelf provides the platform infrastructure to host a multitude of AdvancedTCA modules – switch, CPU, line cards and storage modules, supporting applications ranging from Network Elements to Data Servers to High Performance Computing Platforms.”[68]

5.5.4 Component attribute/requirement module

The component attribute module is used to keep the attributes of the components. The component requirement module has the same structure as the component attribute module.

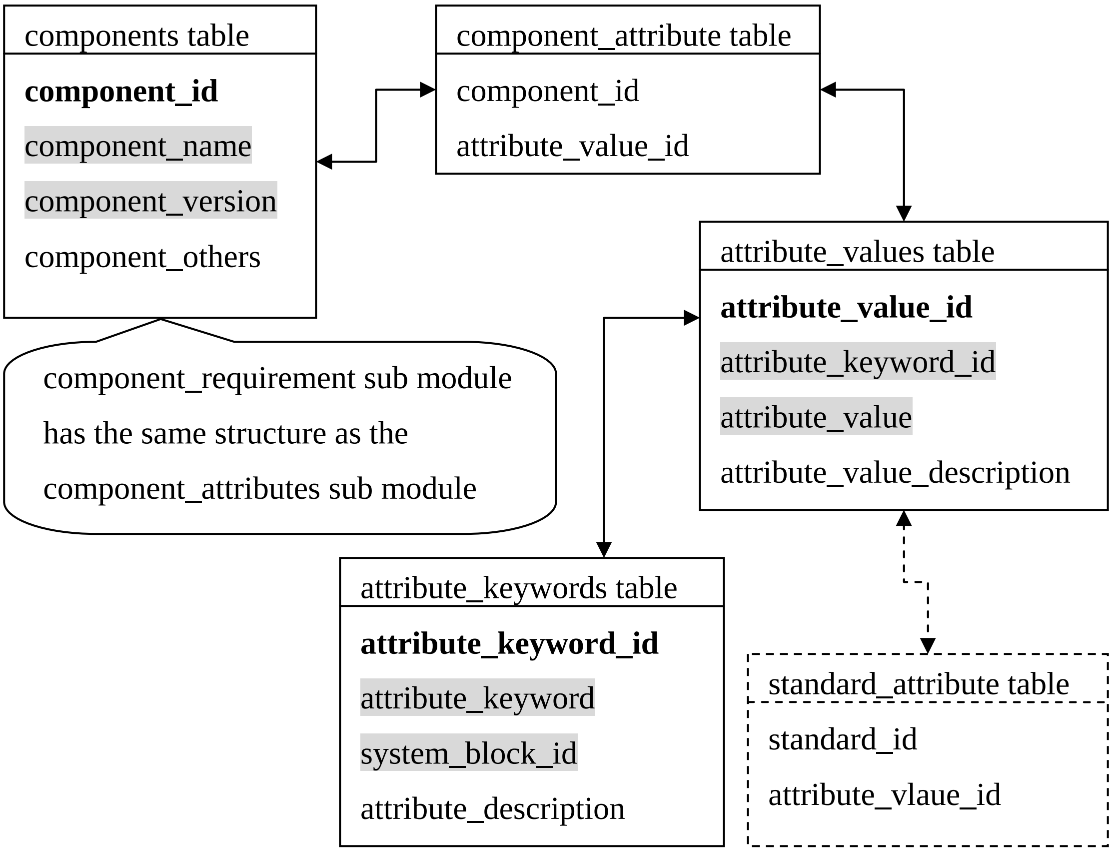


Figure 5.10: Component attribute/requirement module

The main table in the component attribute module is the attribute keywords table and the attribute values table.

The attribute keywords table is used to keep the keyword of the components attributes from the components datasheets. The primary key is the attribute keyword id, and it relates to both system block id and attribute keyword. For example: (Table 5.1)

Attribute_keyword_id	System_block_id	Arrtibute_keyword	Attribute_description
1	5(shelf)	Slot number	
2	7(storage)	Memory size	

Table 5.1: Attribute keywords table

The attribute values table is used to keep the key values of the components attributes. The primary key is the attribute value id, and it relates to both attribute keyword id and attribute value. For example:Table 5.2

Attribute_value_id	Attribute_keyword_id	Arrtibute_value	Arrtibute_value_description
1	1(shelf-slot number)	14	
2	1(shelf-slot number)	16	
8	2(storage-memory size)	150G	
9	2(storage-memory size)	200G	

Table 5.2: Attribute values table

The component attribute table is used to show the components attributes with the values from the components datasheets. The two main fields are component id and attribute value id. For example:Table 5.3

Component_id	Attribute_value_id
11(Motorola ATCA shelf M1)	1(shelf-slot amount: 14)
12(Motorola ATCA shelf M2)	2(shelf-slot amount: 16)
23(Intel ATCA storage blade I1)	8(storage-memory size: 150G)
24(Intel ATCA storage blade I2)	9(storage-memory size: 200G)

Table 5.3: Component attribute table

The standard attribute table is used to show the components attributes with the value from the standards specifications.

The structure of the component attribute/requirement module gives the possibility of some comparison: between components, between component and standard, and between standards.

1. For the components in the same system block, the comparison of the same attributes may help the vendees to know which components are closer to the requirements and better than others.
2. For the components from different system blocks, the comparison between the components attributes and the components requirements may help the vendees to know which components have a better interoperability, such as the support power of a shelf component and the required power of a blade component must be equal.
3. By comparing the attribute value from a component datasheet and the attribute value from the component compliable standard, the database user may get an in-depth idea about the component compliant standard state. This kind of comparison can be used as a simple test of the component before incorporating them in a real system.
4. There is the possibility that the standards are not always compatible. The comparison of the attributes values between standards may help the standards issuing organizations testing the interoperability between different standards as well.

5.6 Extra module

The extra module comprises a synonym table. See Table 5.4 for example

term_id	term_description
1	blade
1	board
2	CPU
2	central processing unit
2	computer
2	computing
2	processor
3	shelf
3	chassis
3	motherboard
3	rack mount
3	platform

3	system
4	OS
4	operating system
5	ATCA
5	AdvancedTCA
5	Advanced Telecom Computing Architecture
6	TE
6	TietoEnator
…	…

Table 5.4: Synonym table

Since different standards related organizations and different manufacturers have their own definitions of the terms, the same object may have different descriptions. In the real world, people can judge the synonym by themselves, but in the database, the computer compares two descriptions by strings. Even SQL uses "LIKE" to resolve this issue; it does not work in all the cases. For example, in the database, "CPU" and "central processing unit" are two different objects. So if the user uses both "CPU" and "central processing unit" in the database, when searching "CPU", the database will not select the "central processing unit" and present it at the same time. The user can use "c%" to select all the records which start with "c", but then the results will be very general.

There are server methods to resolve this problem. One is to standardize the input terms, which means when the user adds a new item in the database; the database administrator must input the standard term but not various descriptions. For example, when the standard term is "CPU", even if the datasheet uses "central processing unit", the user has to input "CPU" instead. By using this solution, the searching problem is solved. But another issue is that the information which is kept in the database is not the same as the source information. Sometimes the descriptions are just similar but not totally the same, such as "blade" and "board". By using the first solution, the input terms may lose some semantic meaning.

Another solution is using a table to keep all these kinds of synonyms. See Table 5.4. In the synonym table, all the terms which may have similar meaning are using the same term id. See Figure 5.11: term_id = blade AND board.

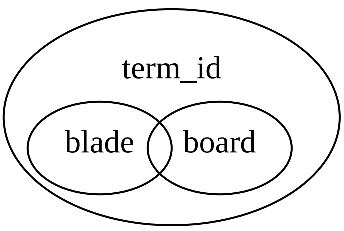


Figure 5.11: Synonyms

Each time the database read in an input term, the system will check the synonym table first to see if this term is already there. Figure 5.12 shows the process of adding a new item in the database.

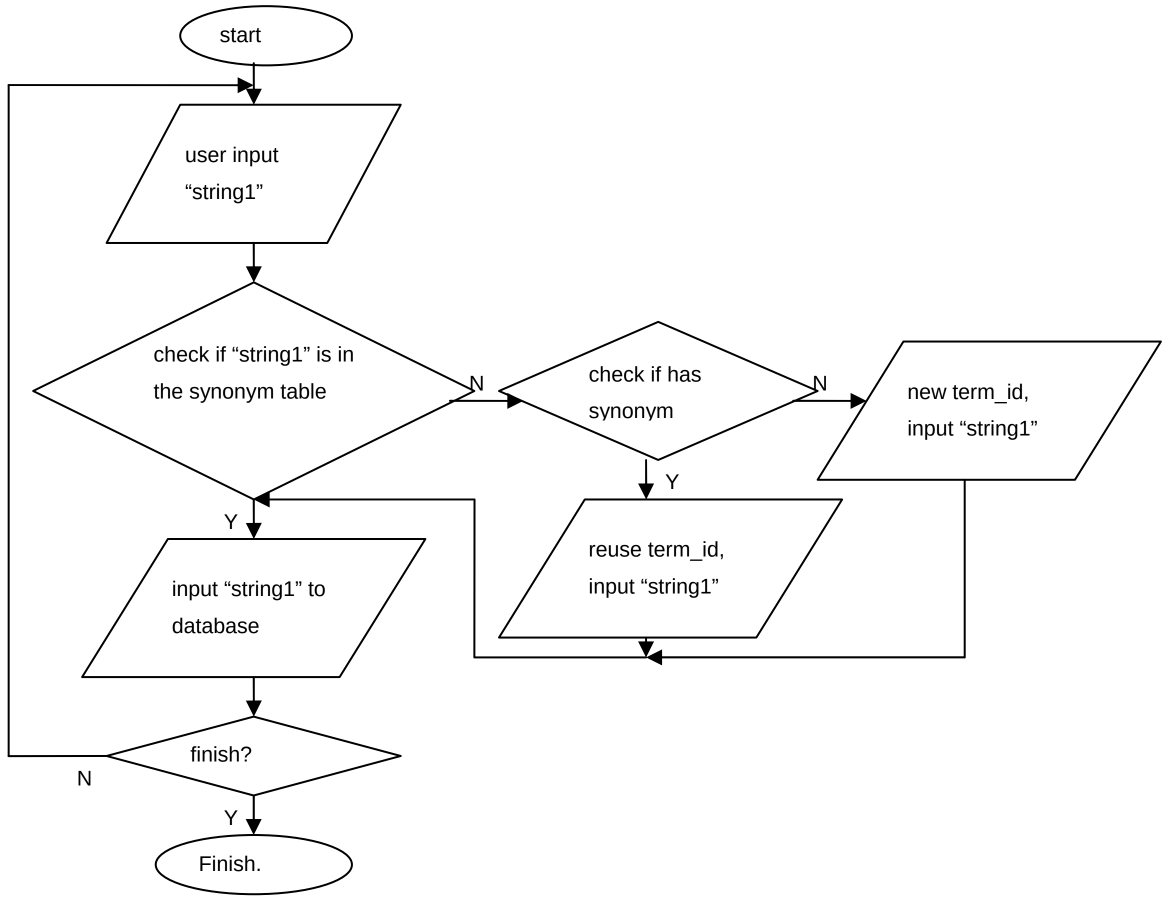


Figure 5.12: Adding process

For searching, when the system gets the input string from the user, it will first check the synonym table. If the input string has a synonym, then the database will select all the information with all these terms. Otherwise the database will perform a regular search and only return the result with the specific string.

5.7 Database functionality

5.7.1 Basic functionality

There are four basic functionalities of the database:

- Add: component, standard, system block, organization
- Search: component, standard, system block, organization
- Delete: component, standard, system block, organization
- Update: component, standard, system block, organization

For the component datasheet, only the external attributes about the component functionalities, services and interfaces will be kept in the database tables. The information such as the internal structure and the user manual, which are of no use for searching or comparing, should not be kept in the table of the database directly. The database user can access the component datasheet through the database and then read the whole documentation.

5.7.2 File management

The database is a good tool to help managing the files and documents:

- component datasheets
- standards specifications
- software files, for example hardware drivers

Since the web information of each component and standard is kept in the database, compared with using a general web searcher, it is more direct to obtain the particular information from the database

A hyperlink is another solution to help the database user to access the specific documents. Current file systems are organized hierarchically and not compliant for the multiplex searching requirement. By using a database, managing documents is more flexible and convenient.

The software files can be kept in the database as well. The database can store all the related files. This, however, is not implemented in the current version.

5.7.3 Testing

The structure of the database gives the possibilities to do some simple testing work by comparing the information in the database tables:

- standards—standards (contradictions)

- component—standards (compliance)
- component—component (interoperability)

By using the profiles from SCOPE and PICMG RES, the core standard attributes from PICMG, OSDL CGL WG, and SAF can be identified. Keep the profiles in the database and then the comparison between different standard attributes may be definite.

Comparing the attributes between component and the compliable standard is a primary work for testing the compliance of a component. Thus the database can be a useful tool for CP-TA as well.

The interoperability between components can be identified from the component attribute/requirement model.

5.8 Summary

The COTS component database has four main modules and the fields cover most of the involved information of a component. The structure is flexible for most kinds of telecom system and a variety of forms of information. The database can be a good tool for the user to make the decision of component acquisition, and managing and sharing the information easily. For details of the database tables and fields please see appendix C – SQL create table.

Figure 5.13 shows the overview of the database.

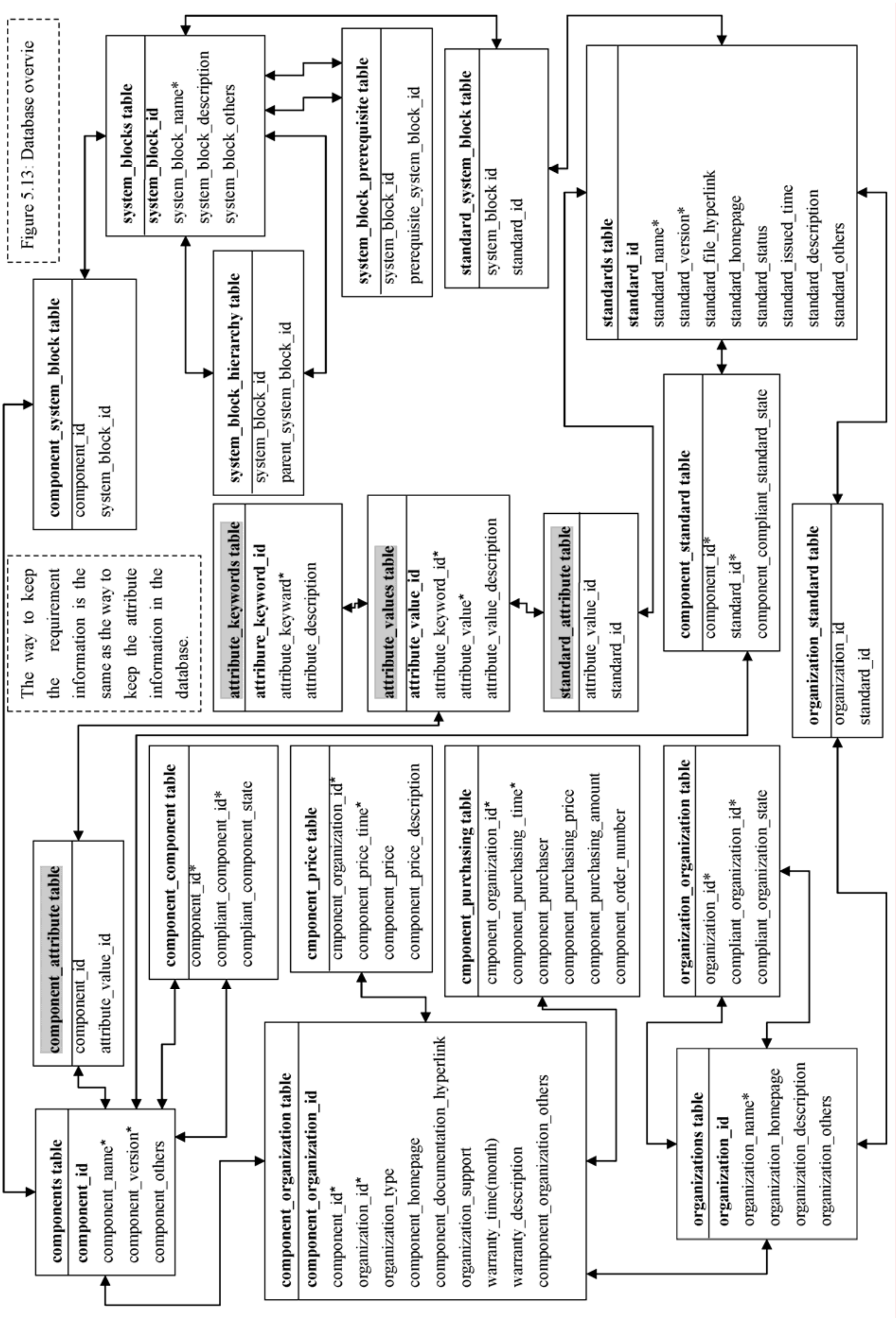


Figure 5.13: Database overview

6 Conclusions, evaluation, and future work

6.1 Conclusions

In this dissertation project, basic knowledge of a telecom system has been gained by studying industrial articles, organizations datasheets, standards specifications, and components datasheets. Academic books about software engineering, management information systems, and computer database have also been read as the background material.

The next generation of telecom systems will be based on COTS components and open source components rather than customized components. Thus the acquisition and integration methods will change. In conclusion, the system integrator nowadays must have knowledge of both their requirements and the state of the current component market. This project is a primary attempt to help the platform integration department of TietoEnator to manage the information from the industrial market.

A telecom system is mainly separated into hardware platform, OS, middleware platform and applications. There are no standards for the applications and the standards sources for the hardware platform, OS, and SA middleware are described below.

The main materials of the open standards are from the following issuing organizations:

1) PICMG: produces the standards specifications for hardware platform, this dissertation used ATCA (PICMG3.0-3.6) standard as an example.
2) OSDL CGL WG: produces the standards requirements definitions for the Carrier Grade Linux. The CGL RD v3.2 is used as the example.
3) SAF: produces the standards for the interfaces of the SA middleware. The standards specifications used as the examples are HPI B.01.01, AIS A.01.01, and AIS B.02.01.

Several standards structuring organizations are mentioned in the dissertation since their work can be a reference for this project.

1) SCOPE: analyzes the requirement from the market and makes the profiles of the standards from PICMG, OSDL, and SAF.
2) PICMG RES: formats the profiles of the ATCA standards as the vertical market profiles.
3) MVA organizes events to help finding the gap and overlap between different standards.

4) OCAF makes a reference model describes the building blocks by functionalities and services for a typical carrier grade telecom system.

After collecting, compiling, synthesizing and analyzing all these information, a system reference model for the COTS component database has been created (Figure 4.2). This model contains most of the components which are used to build a telecom system and the mentioned standards are the main part we have focused on.

Information about the COTS components is mainly taken from the components datasheets. Thus, a study of the components datasheets from several manufacturers has been done as well.

To build a COTS component database, certain information must be included: manufacturer & supplier, compliance standard, system block and attribute & requirement. Thus the database has four main models: organization model, standard model, system block model, and component model. Each model contains a set of basic tables which keep most of the related information about a COTS component.

6.2 Evaluation

The techniques and technology which are used in this dissertation project are old, however, the contribution of this project is to collect the available information and allow comparisons and checking in one database.

The existing databases are mostly held by certain companies, thus it is hard for the user to compare the COTS components from different suppliers. The database in this project covers all the related organizations, and makes the comparisons of the COTS components with similar functionalities and services but from different suppliers can be compared directly.

The scope of the existing databases from different COTS components suppliers mainly focuses on one system layer. This dissertation covers the entire system, thus there is the possibility to test the interoperability between more kinds of components, such as hardware & OS, hardware & middleware, and OS & middleware.

The databases from different COTS components suppliers use their own data format to store info, and the formats vary between databases. The database in this project will try to format the data from various companies; the database structure is very simple and flexible, suitable for various information formats.

The current database only achieves the basic functionalities (keeping, searching, and sharing) and can be used as storage to manage the documents and most of the related information. The design is extensible, which means that more functionality can be gained from the database in the future.

6.3 Future work

6.3.1 Database initialization

The core model of the COTS component database is the component model. The other three models (organization, standard, system block) are access models. These three models and the synonym table contain the information which can be kept in the database in advance. For example, the list of the current big telecom companies and the standards related organizations can be kept in the main table of the organization model. The existing influential standards specifications for telecom industry are enumerable and can be kept in the standard model in advance as well. The system reference models vary with the purposes of the systems, but the terms of the basic unit in each model are similar. Thus, the name of the system block can be filled in the system model in advance.

All this work should be done by professional experts who know the telecom market very well. This work will help the user filling in the components information faster in the future and reduces the redundancies of the synonym table by supporting the propositional input terms.

6.3.2 Wider scope

The current work for this discourse covers only three standards issuing organizations[23] and five standards structuring organizations[24]. There are more organizations with more standards for the current telecom industry; examples are Network Processing Forum (NPF), RapidIO, TeleManagement Forum (TMF), and Distributed Management Task Force (DMTF). It will be really helpful to extend the scope and collect more standards and system reference models for the database. Since the structure of the database is very flexible, not only COTS components but also other components and documents will be able to be kept in the database.

The current database only supports the basic functionalities for storage and simple searching. In order to get more comparative results and statistics from the database, more information about the components must be kept in the tables. The current structure of the standard and component attributes/requirements table is very flexible for adding a new record, but not as good for comparing data and producing statistics. In the future, the domains of the

[23]standards issuing organizations: PICMG, OSDL CGL, SAF
[24]standards structuring organizations: SCOPE, PICMG RES, CP-TA, MVA, OCAF

attributes need to be limited and formatted. This work needs to have a candidate terms list first and then the professional people can select the best set from the list.

From a single datasheet to a database, the future development of the information management is data warehousing and grid computing. The more information is put in the database, the more information could be found from the database.

6.3.3 Open database

In order to collect more information and let the database to have more availability, one solution is to put the database on the web as an open source. The database holder can make different user rights to control the database on the website and obtain payment for the services. For the component provider and the organization issuer, it is possible for them to add and modify their data. For the component vendee, the system integrator and the end user, it is possible for them to search the information from the database. The current problem about the list price of a COTS component, that it cannot be updated in real time, can be solved by opening the database to the components supplier as well.

References

[1] Associates, Demac. Standards for a Service Availability Solution. 2002.

http://www.saforum.org/about/solution_backgrounder.pdf (2006-03-02)

[2] Byers, Chuck. AdvancedTCA® Tutorial: AdvancedTCA® Applications Studies. 2004-06-23.

http://www.picmg.org/pdf/Supercomm_Tutorial.pdf (2006-02-02)

[3] Byers, Chuck. AdvancedTCA® Tutorial: AdvancedTCA® Introduction. 2004-06-23.

http://www.picmg.org/pdf/Supercomm_Tutorial.pdf (2006-02-02)

[4] Cai, Xia. Lyu, Michael.R. Wong, Kam-Fai. Ko, Roy. Component-Based Software Engineering: Technologies, Development Frameworks and Quality Assurance Schemes. 2000-12-17.

http://www.cse.cuhk.edu.hk/~lyu/paper_pdf/apsec.pdf (2006-08-18)

[5] Campbell, Susan J. Communications Platforms Trade Association Officially Launched. 2006-04-27.

http://news.tmcnet.com/news/2006/04/26/1613161.htm (2006-09-13)

[6] Cress, Andy. The Telecom System View: Linux Clustering Software for Telecom. 2004-01-22.

http://developer.osdl.org/cherry/cluster-whitepapers/cluster_intel.pdf (2006-05-03)

[7] Davidson, Rob. MVA Launches Communications Ecosystem Conference. 2006-09-05.

http://www.rapidio.org/news/pr/view?item_key=c1bede5ce1f513a783963086f213b89a033b2a14 (2006-09-13)

[8] Haddad, Ibrahim. Carrier Grade Linux: Adoption and Deployments Figure 5: The scope of the Carrier Grade Linux Initiative. 2005-07-14.

http://linuxplanet.com/linuxplanet/reports/5931/2/screenshot3417/ (2006-09-08)

[9] Haddad, Ibrahim. Carrier Grade Linux: Adoption and Deployments Figure 6: The context of Carrier Grade Linux in Access and Corporate networks. 2005-07-14

http://linuxplanet.com/linuxplanet/reports/5931/2/screenshot3418/ (2006-09-08)

[10] Haddad, Ibrahim. Moving Toward Open Platforms. 2004-04-19.

http://www.saforum.org/press/articles/Moving_Toward.pdf (2006-02-28)

[11] Haddad, Ibrahim. Open Standards for Carrier Grade Linux (CGL Datasheet). 2004.

http://www.osdl.org/docs/cgl_data_sheet.pdf (2006-09-07)

[12] Haddad, Ibrahim. The OSDL Carrier Grade Linux Initiative:

A presentation for the MultiService Forum. 2005-07-19.

http://groups.osdl.org/apps/group_public/download.php/1301/CGL_MSF.pdf

(2006-09-08)

[13] Lorenzon, Gunnar. COTS component database for standardized platform integration Thesis Specification Rev 1.0. 2006-02-03.

[14] Lorenzon, Gunnar. Private Communication (E-mail). 2006-08-23

[15] Meyers, B.Craig. Oberndorf, Patricia. Managing Software Acquisition: open systems and COTS products. 2001.

[16] Reitenspiess, Manfred. Delivery of the SA Forum Application Interface Specification. 2003-10-05. http://www.saforum.org/press/presentations/SRDS-WS-AIS-presentation.pdf (2006-03-02)

[17] Strass, Hermann. Inaugural International Service Availability Symposium held in Munich. July 2004. http://www.compactpci-systems.com/columns/Technology_Update/pdfs/7.2004.pdf (2006-09-18)

[18] Zhang, Xiaolu. The Service Availability Forum Platform Interface Overview and Model. April 2003. http://www.cs.helsinki.fi/u/niklande/opetus/SemK03/Zhang.pdf (2006-02-06)

[19] About PICMG. http://www.picmg.org/aboutus.stm (2006-09-05)

[20] About SCOPE Alliance. http://www.scope-alliance.org/about.html (2006-09-12)

[21] About the Forum. http://www.saforum.org/about/ (2006-09-08)

[22] About the Mountain View Alliance. http://www.mountainviewalliance.org/ (2006-09-13)

[23] AdvancedTCA Specifications for Next Generation Telecommunications Equipment. http://www.picmg.org/newinitiative.stm (2006-09-08)

[24] AdvancedTCA*. http://www.intel.com/technology/atca/ (2006-02-28)

[25] AdvancedTCA: PICMG 3.0 Short Form Specification. January 2003. http://www.picmg.org/pdf/PICMG_3_0_Shortform.pdf (2006-02-28)

[26] AdvancedTCA™ Signaling Blade: product sheet. http://www.signaling.tietoenator.com/filer/ProductSheets/ProdSheet_AdvancedTCA%20Signaling%20Blade.pdf (2006-09-18)

[27] AMCC AdvancedTCA: AMCC provides AdvancedTCA Evaluation boards and Software for its Switch Fabric, Packet Processors and Embedded PowerPC Processors.

http://www.amcc.com/ATCA/ (2006-09-08)

[28] ATCA - The NexGen Telecom Standard: Advanced Telecom Computing Architecture. http://www.nuvation.com/corporate/news/newsletter/summer2004/atca.html (2006-03-03)

[29] ATCA PROFILE: Version 1.0. 2006-03-30. http://www.scope-alliance.org/pr/SCOPE_ATCA_Profile_v1.0.pdf (2006-09-12)

[30] ATCA-7221,R3: AdvancedTCA Processor Blade. http://www.motorola.com/mot/doc/6/6311_MotDoc.pdf (2006-09-16)

[31] ATCA-S100,R3: AdvancedTCA Storage Blade. http://www.motorola.com/mot/doc/6/6312_MotDoc.pdf (2006-09-16)

[32] AXP1600 AdvancedTCA Shelf—Datasheet. http://www.motorola.com/mot/doc/5/5403_MotDoc.pdf (2006-09-16)

[33] Carrier Grade Linux. http://www.osdl.org/lab_activities/carrier_grade_linux/ (2006-02-27)

[34] Carrier Grade Open Environment Reference Model: Basis Version 1.0. May 2005. http://www.itu.int/ITU-T/ocaf/files/cgoe-rm-v1.0.pdf (2006-01-09)

[35] Centellis 3600 - AdvancedTCA System Platform. http://www.motorola.com/content.jsp?globalObjectId=5663 (2006-09-16)

[36] Commercial off-the-shelf. 2006-07-07. http://en.wikipedia.org/wiki/Commercial_off-the-shelf (2006-07-08)

[37] consortium and standards list. http://consortiuminfo.org/links/ (2006-08-24)

[38] consortium and standards list: telecom. http://consortiuminfo.org/links/cats.php?ID=26 (2006-09-01)

[39] Coordination of specifications enables rapid COTS system development. http://www.mountainviewalliance.org/MVAWhitePaper.pdf (2006-09-13)

[40] Definition of open standards. June 2004. http://www.oio.dk/files/040622_Definition_of_open_standards.pdf (2006-03-03)

[41] FAQ. 2004-07-29. http://www.itu.int/ITU-T/ocaf/faq.pdf (2006-09-13)

[42] FREE 30 x 20" AdvancedTCA standards poster. http://www.nmscommunications.com/devplatforms/whitepapers/abstracts/advancedtca.htm (2006-09-07)

[43] Frequently Asked Questions. http://www.cp-ta.org/about/faq/ (2006-09-13)

[44] GoAhead SelfReliant: Basic Availability Management (SR-BAM).
http://www.goahead.com/pdf/dataSheet_srBAM_2005.06.27.pdf (2006-09-16)

[45] Goals and Charter. 2006.
http://www.mountainviewalliance.org/pdfs/CP-TA_Goals_and_Charter.pdf
(2006-09-13)

[46] Hardware Platform Interface: Specification SAI-HPI-B.01.01. 2004-03-17.
http://www.saforum.org/specification/getspec_content/SAI-HPI-B.01.01.pdf
(2005-12-20)

[47] Hot swapping. 2006-07-25.
http://en.wikipedia.org/wiki/Hot_swap (2006-08-19)

[48] Interoperability. 2006-08-08.
http://en.wikipedia.org/wiki/Interoperability (2006-08-19)

[49] ITU-T Sector Members.
http://www.itu.int/cgi-bin/htsh/mm/scripts/mm.list?_search=ITU-T&_languageid=1
(2006-09-13)

[50] Key Advantages of the AdvancedTCA* Architecture. 2005-09-15.
http://mfile.akamai.com/10430/wmv/cim.download.akamai.com/10430/con/ATCA_video.asx (2006-09-11)

[51] Linux Industry Segments.
http://groups.osdl.org/about/industry_segments?segment=sysvendors (2006-09-08)

[52] List of Executive Members.
http://www.picmg.org/picmg_cfmfiles/generalv3/executivemembers.cfm (2006-08-18)

[53] Member Companies.
http://www.cp-ta.org/about/roster/ (2006-09-13)

[54] Membership Levels and Benefits.
http://www.saforum.org/join/levels_and_benefits/ (2006-03-03)

[55] MontaVista Linux Carrier Grade Edition 4.0 Data Sheet.
http://www.mvista.com/dswp/ds_cge.pdf (2006-09-18)

[56] New PICMG® Subcommittee to Enhance AdvancedTCA® Interoperability. 2006-04-25.
http://www.picmg.org/news.stm#pr042506 (2006-09-13)

[57] Open Communications Architecture Forum (OCAF) Focus Group.
http://www.itu.int/ITU-T/ocaf/ (2006-09-13)

[58] Open Specifications for Service Availability: SA Forum datasheet.
http://www.saforum.org/press/kit/SA_Forum_datasheet.pdf (2006-06-16)

[59] Open Specifications for Service Availability™.
http://www.saforum.org/home (2006-09-08)

[60] OSDL.
http://www.osdl.org/ (2006-09-07)
[61] OSDL. 2006-08-25.
http://en.wikipedia.org/wiki/OSDL (2006-09-07)
[62] OSDL FAQ.
http://osdl.org/about_osdl/faq.html/document_view (2006-08-07)
[63] OSDL Member Companies and Affiliates: OSDL Member Roster.
http://groups.osdl.org/osdl_members/osdl_roster (2006-08-31)
[64] OSDL Membership and Affiliate Programs.
http://groups.osdl.org/join/#member (20006-03-03)
[65] OSDL Mobile Linux Initiative.
http://osdl.org/lab_activities/mobile_linux/mli/document_view (2006-09-07)
[66] OSDL Mobile Linux Initiative: Accelerating Linux Adoption for Mobile/Wireless (MLI Datasheet). 2006-02-07.
http://osdl.org/lab_activities/mobile_linux/mli_datasheet.pdf (2006-09-07)
[67] Private Communication (E-mail). SAF. 2006-03-07.
[68] Promentum ATCA-6000 Data Sheet: 12U 14-Slot ATCA Shelf.
http://www.radisys.com/china/products/datasheet_page.cfm?productdatasheetsid=1173&CFID=70450&CFTOKEN=12025364 (2006-09-16)
[69] Requirements.
http://www.osdl.org/lab_activities/carrier_grade_linux/requirements.html/document_view (2006-09-08)
[70] Requirements Engineering Subcommittee Overview. 2006-02-07.
http://www.mountainviewalliance.org/pdfs/RES_Overview.pdf (2006-09-12)
[71] SA Forum Members.
http://www.saforum.org/about/companies/ (2006-08-18)
[72] SCOPE Alliance. 2006-06-02.
http://www.scope-alliance.org/docs/scope_in_brief_060602.pdf (2006-09-12)
[73] SCOPE FAQ.
http://www.scope-alliance.org/pr/scope_faq.pdf (2006-09-12)
[74] Service Availability Interface: Overview SAI-Overview-B.02.01. 2005.
http://www.saforum.org/specification/getspec_content/SAI-Overview-B.02.01.pdf (2006-01-10)
[75] Solaris Operating System—Data Sheets.
http://www.sun.com/software/solaris/ds/index.jsp#2 (2006-09-15)
[76] Specification Order Forms.

http://www.picmg.org/specorderformsec.stm (2006-03-03)
[77] Specifications. 2006-07-10.
http://www.picmg.org/specifications.stm (2006-09-05)
[78] Telecommunication. 2006-08-06.
http://en.wikipedia.org/wiki/Telecommunications (2006-08-06)
[79] The Advantages of AdvancedTCA*: The Telecom Industry's Open Standard for Equipment Manufacturers. 2004.
ftp://download.intel.com/design/network/papers/30265501.pdf (2006-08-31)

Appendix A – List of abbreviations

AIS	Application Interface Specification
AMC	Advanced Mezzanine Card
API	Application Programming Interface
ATCA	Advanced Telecom Computing Architecture (AdvancedTCA)
CGL	Carrier Grade Linux
CGL RD	Carrier Grade Linux Requirement Definition
CGL RDs	CGL Requirements Definitions
CGL WG	Carrier Grade Linux Working Group
CGOE	Carrier Grade Open Environment
COTS	commercial off-the-shelf
CP-TA	Communications Platforms Trade Association
CPU	central processing unit
DCL	Data Center Linux
DMTF	Distributed Management Task Force
DTL	Desktop Linux
FOSS	Free and Open Source Software
HLRs	home location registers
HPI	Hardware Platform Interface
ISV	Independent Software Vendor
ITU-T	International Telecommunication Union-Telecommunication Standardization Sector
IVR	interactive voice response
MGCP	media gateway control protocol
MIB	Management Information Base
MLI	Mobile Linux Initiative
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
MVA	Mountain View Alliance
NPF	Network Processing Forum
OCAF	Open Communication Architecture Forum
OIF	Optical Internetworking Forum
OS	operating system

OSDL	Open Source Development Lab
PICMG	PCI Industrial Computer Manufacturers Group
PICMG RES	PICMG Requirements Engineering Subcommittee
RAM	random access memory
RASM	reliability, availability, serviceability and manageability
RTMs	rear transition modules
SA	Service Availability
SAF	Service Availability Forum
SCOPE	SCOPE Alliance
SIP	session initiation protocol
SMS	short message service
SMS	System Management Specification
SNMP	Simple Network Management Protocol
SS7	signaling system 7
TEMs	telecom equipment manufactures
TMF	TeleManagement Forum

Appendix B – Component datasheet examples

1. SA Middle from GoAhead: SelfReliant Basic Availability Management (SR-BAM)
2. CGL OS from MontaVista: Linux Carrier Grade Edition 4.0
3. ATCA Blade from TietoEnator: AdvancedTCA Signaling Blade
4. ATCA Hardware platform from Motorola: Centellis 3600 - AdvancedTCA System Platform
    a. ATCA Shelf from Motorola: AXP1600 ATCA Shelf
    b. ATCA Blade from Motorola: ATCA-S100,R3 ATCA Storage Blade
    c. ATCA Blade from Motorola: ATCA-7221,R3 ATCA Processor Blade

GoAhead SelfReliant

Basic Availability Management (SR-BAM)

For application and system developers who require high availability without complexity, GoAhead® SelfReliant® Basic Availability Management (SR-BAM) provides the surest and fastest path to 99.999% availability. Targeted for systems running on rack-mounted, pedestal, or bladed servers, SR-BAM allows developers to quickly make applications highly available. It rapidly detects, isolates, recovers, and sends notifications for application, system, and external services failures. Recovery actions include application start, stop, restart, and fast switchover to a redundant resource.

SR-BAM offers a layered method to quickly and easily implement SelfReliant HA functionality. Many customers start by adding simple, transparent application failover to ensure continuous service. This approach to availability requires no application modification. It is ideal when stateless failover is sufficient HA functionality, or when it is not possible or desirable to modify the application. Transparent availability management can be applied to applications on up to 64 nodes. For many projects, this SR-BAM implementation fully meets the system's availability requirements.

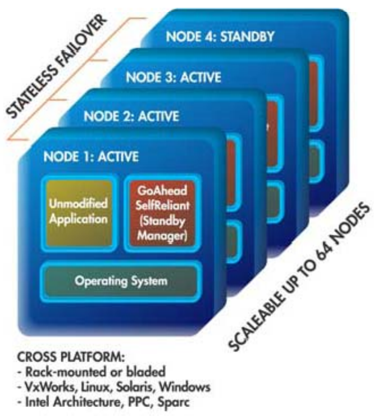
For developers who require more availability control over their applications and services, SR-BAM offers a set of checkpointing APIs to implement failover across application and/or node pairs to ensure critical state is preserved during any type of failure. Active/active and active/standby configurations are supported.

Key SR-BAM Benefits

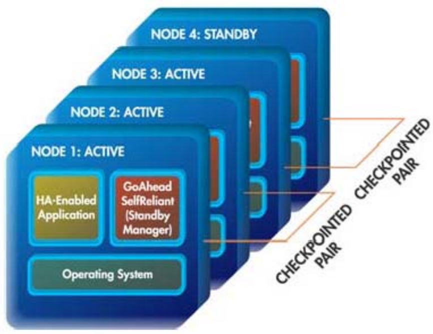
- **Enables development team to quickly implement fundamental HA capabilities**
- **Pre-integration with third party hardware and software eliminates unproductive development effort and insulates development teams from changes in underlying hardware and software**
- **Platform independence enables development teams to more quickly respond to new technologies and platforms**
- **By seamlessly upgrading to the Advanced Suite, development teams can leverage more sophisticated capabilities**

For more complex systems that require more sophisticated failover policies and hardware management, SR-BAM can be easily upgraded to the SelfReliant Advanced Suite (SR-AS). Building upon their initial SR-BAM high availability features, project teams can gracefully upgrade to the Advanced Suite for even greater control of all critical system resources.

Multi-Node Switchover with No Application Modifications



Simple Set of APIs for Checkpointing of Application and/or Node Pairs



GoAhead SelfReliant

GOAHEAD

Who Uses SR-BAM

With an easy-to-use interface and a simple set of APIs, SR-BAM is ideal for:

- Project teams who are under severe time pressures to deliver fundamental HA capabilities
- Developers who are building solutions that require only fundamental HA capabilities
- Architects who need a phased approach to HA functionality—starting with fundamental HA capabilities and moving to more sophisticated HA capabilities over time
- Project teams that need to quickly implement a prototype HA solution for field trials, then add more sophisticated HA capabilities for later releases and commercial deployment
- Developers of legacy or commercial applications that must be managed for availability, but the code cannot be modified

Basic Availability Management Services

Transparent Availability Management Service (TAMS)

This feature provides high availability for legacy and third-party applications, as well as applications that preserve state through other methods.

- Application monitoring and control to start, stop, restart, and switchover monitored applications and/or nodes
- Sub-second failover for clusters up to 64 nodes
- Application role assignments: active, standby, unassigned
- Failover of active application to standby application automatically upon fault detection, via manual request or external management interface
- Individual or group failover support, including failover of resources and processes that depend on a failing application
- Configurable retry policies (frequency and time period) to attempt, upon failure, to restart the application on the current node prior to a switchover
- Application and node monitoring via customer-defined text or binary protocol (such as HTTP or PING)
- Virtual IP address support
- Ability to call external applications (such as sending an SNMP trap, set, or external command line) upon a start, stop, restart, and switchover of the application
- Browser-based management console enables remote access to applications in the cluster to start, stop, restart, monitor health, and determine role assignment

Simplified Availability Management Service (SAMS)

This feature provides checkpointing between application and node pairs. It is intended for applications that require fast failover with preservation of application state.

- Simple HA APIs for pairs of applications and/or nodes in active/standby and active/active configurations
- APIs enable replication of application state
- Millisecond stateful switchover
- Support for individual, group, service group, and dependency failover scenarios

SR-BAM Specifications *(measured on Linux CGE 3.1)*

System Requirements

- Run-time memory: 4.5 MB
- Run-time disk space: approximately 10 MB
- Development sisk space: 110 MB
- Operating systems/CPU platforms supported:
  - RedHat Enterprise Linux 3.0 (IA)
  - RedHat Linux 9.0 (IA)
  - MontaVista Linux CGE 3.1 (IA)
  - MontaVista Linux Professional Edition 3.1 (PPC)
  - Windows Server 2003 (IA)
  - Debian 3.0 Linux (IA64)

Performance and Scalability

- Modified application failover time: as low as 10 msec
- Unmodified application failover time: as low as 300 msec
- Steady-state CPU usage: <0.1%
- Number of nodes: up to 64

Languages supported

C and C++ (only applicable for application modification)

www.goahead.com
10900 NE 8th Street, Suite 1200 Bellevue, Washington 98004-1455
PHONE +1.425.453.1900 **FAX** 1+.425.636.1117 **CONTACT** info@goahead.com

MontaVista Linux Carrier Grade Edition 4.0

The Premier Linux Platform for Network Equipment

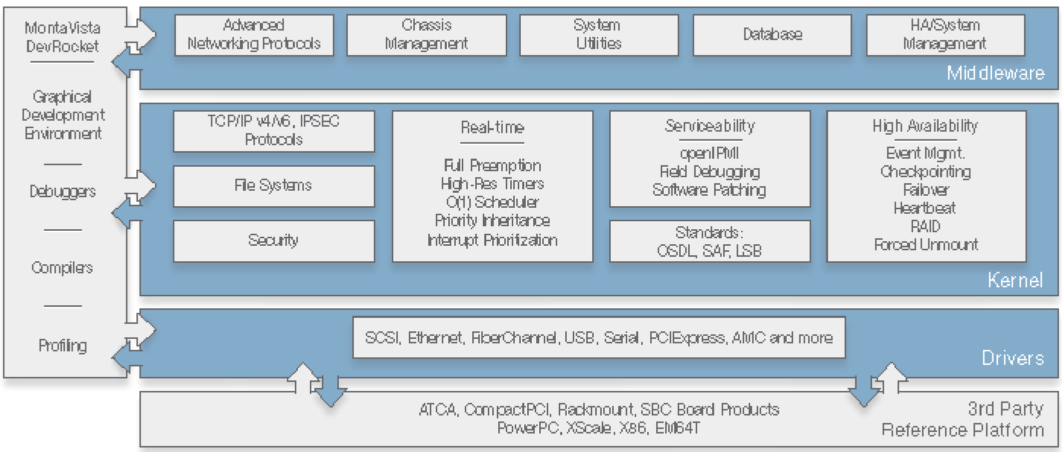
As carriers and service providers accelerate the deployment of next generation network services, they must leverage a new class of infrastructure, based on open standards. As the framework behind this infrastructure shifts, Network Equipment Providers (NEPs) turn to Commercial-Off-The-Shelf (COTS) platforms, such as Linux, to help bring down costs and accelerate their time-to-market. MontaVista Linux Carrier Grade Edition (CGE) is designed with the most advanced real-time, high-availability, serviceability and true carrier grade functionality available in any Linux today.

The Foundation for Network Infrastructure Services

When determining an operating system strategy, it is critical that you choose partners that will help build and support your business into the future. MontaVista Linux Carrier Grade Edition enables you to migrate smoothly and efficiently to next generation services and technologies that leverage the latest advancements in open source and telecommunications technologies. CGE, the only Linux solution trusted by the world's top NEPs, is the premier COTS-based Linux platform designed specifically for next generation networks. Focused on high availability, reliability and performance, CGE adds many carrier-grade features that are not available in enterprise Linux – ensuring the ability for carriers to achieve up to six "9s" of reliability with their services.

Comprehensive Product Offering

CGE includes comprehensive development tools (based on the latest open source Eclipse framework), world-class technical support and services, and is the most complete carrier-grade operating system in the industry. CGE gives you online access to MontaVista Zone®, a world class web portal, which provides the latest downloads, development tips, and comprehensive documentation. Additionally, MontaVista works closely with leading Independent Software Vendors to deliver fully validated middleware, advanced protocols, tools, and applications. These offerings give you the ability to create reliable, cost-effective, best-of-breed solutions that achieve the performance levels that carrier-grade systems require.



CGE OS Platform

Technology Leadership

MontaVista has unmatched experience and expertise in delivering proven carrier grade extensions to Linux. We make key contributions to most of the recognized carrier grade open source projects, including clustering, availability, real-time, and serviceability.

Clustering support, authored by MontaVista Software, includes the OpenAIS project for creating and managing fail-over clusters with Service Availability® Forum compliant interfaces for fault detection, event management, and checkpointing enabling safe application failover with protection of vital state information.

Availability extensions include redundant network and file systems support (RAID and DRBD), forced unmount of file systems, and watchdog timers.

Real-time enhancements have always been a hallmark of MontaVista Software with kernel preemption, high resolution timers, and interrupt handler scheduling offering performance that remains significantly ahead of the standard Linux capabilities.

Serviceability enhancements in CGE deliver unequaled support for the managing, monitoring and debugging of deployed systems. Comprehensive, extended dump and performance analysis tools ensure critical data will be captured and analyzed efficiently. A unique kernel resource monitoring framework works in conjunction with the event logging system to track critical events in a single place, enabling high-availability management middleware to take appropriate action. CGE also includes the MontaVista Software-authored OpenIPMI system as well as SA Forum Hardware Platform Interface (HPI) services for comprehensive hardware management capabilities. Finally, only MontaVista delivers a commercial grade, non-intrusive field debugging and function-level application patching facility with the MontaVista Field Safe Application Debugger (FSAD).

Standards Compliance

Our focus on critical industry standards ensures compatibility with leading software applications and hardware platforms, accelerating time-to-market and lowering total cost of ownership. CGE is compliant with the Open Source Development Lab's (OSDL) Carrier Grade Linux requirements specification v2.0 as well as the Service Availability Forum Application Interface Specification (AIS) and Hardware Platform Interface (HPI) standards. CGE also has the industry's broadest available support of COTS platforms compliant with CompactPCI and AdvancedTCA® specifications.

The MontaVista Advantage

Leading telecom equipment providers worldwide have adopted the MontaVista Linux family of products for use in their carrier equipment. In fact, MontaVista Linux Carrier Grade Edition is the only Linux deployed on multiple generations of these systems. Our open source and Linux experience, our reputation for innovation, and our proven technology are just a few reasons for selecting MontaVista Software as a partner in your success.

For the latest information on CGE, visit www.mvista.com/products/cge

Hardware Support
- Intel Pentium
- Intel EM64T
- IBM PowerPC 32-bit and 64-bit
- Freescale PowerPC
- ATCA
- PICMG 2.16 Systems

For a complete list, visit http://www.mvista.com/products/boards.html

Development Hosts
- Red Hat Linux 9.0
- Red Hat Enterprise Linux 3.0
- Sun Solaris 8.0, 9.0
- Native development on CGE 4.0

Performance and Scalability
- Fully Preemptible Linux kernel with priority based interrupt handlers
- Four user-selectable modes for Real-Time
- Real-time scheduler with CPU affinity
- Native POSIX Thread Library (NPTL) support
- Microsecond resolution POSIX timers
- Fast reboot with kexec facility

High Availability
- OpenAIS clustering services
- Persistent Device Naming
- Watchdog timer support
- RAID 1 and DRDB disk mirroring support
- Ethernet bonding driver for ethernet aggregation and failover
- Forced Unmount of file systems
- Journaling file systems

Serviceability
- IPMI Hardware Management with ATCA extensions
- OpenHPI - SAF Hardware Platform Interface
- MontaVista Field Safe Application Debugger (FSAD)
- Updated Logical volume manager (LVM2)
- In-kernel debugger (kdb)
- Extensive Crash Dump facilities
- Remote boot capability
- Kernel resource monitoring
- POSIX event logging and event broker

Networking and Security
- Linux Security Module (LSM)
- IPSecv4/v6
- Encryption support including SSL, PF_KEY, PKI_CA
- IPv4/v6 native stack and USAGI extensions
- MobileIPv6
- SCTP

MontaVista DevRocket
(Advanced IDE designed for platform & application developers)
- Platform Image wizard
- Library footprint optimization
- Platform Execution Tracing (Linux Trace Toolkit)
- Latest 3.4 based GNU Tool chains
- Memory leak detection
- GUI-based debugging
- MontaVista Target Tools

MontaVista Software, Inc. 2929 Patrick Henry Drive Santa Clara, CA 95054 USA t: 408.572.8000 f: 408.572.8005 www.mvista.com

AdvancedTCA™ Signaling Blade

Signaling Solutions from TietoEnator provides a turnkey signaling platform for usage in telecom core nodes like MSC, GSN and HLR. It is also ideal for integration with telecom applications like location-based or prepaid services, messaging, voice over IP, and softswitch solutions. It complies with ITU, ANSI, TTC, and Chinese standards.

Introduction

The telecom industry is today facing very high demands in terms of functionality, capacity and reliability from their customers. On the other hand, there is also a demand on reducing cost and development lead-time. One way of reducing cost and lead-time, while maintaining operating characteristics, is to use a standards-based Modular Communications Platform (MCP). The new Advanced Telecom Computing Architecture (AdvancedTCA) is clearly a big step towards a harmonized and standardized MCP specifically for core telecom products like MSC, GSN and HLR.

TietoEnator strongly believes in standardized building blocks both with regards to hardware and software. By using a standardized platform, telecom equipment manufacturers can rely on TietoEnator and other component suppliers to deliver platform components with the right characteristics. This will lead to lower development costs, shorted lead-time and lower maintenance costs.

Solution overview

The AdvancedTCA Signaling Blade is an integrated signaling sub-system consisting of standard hardware components and TietoEnator signaling middleware. It is a flexible system consisting of a combination of hardware and signaling software for AdvancedTCA-based systems. It is designed to meet the extreme requirements on reliability, capacity and scalability, normally required in telecom network core nodes. The AdvancedTCA Signaling Blade is intended for telecom equipment manufacturer, system integrators, application developers and network operators implementing telecom core & radio network connectivity in a AdvancedTCA system.

The software is optimized for the Intel IA32 architecture, taking advantage of the latest improvements in CPU design.

Key benefits

The main benefit of the AdvancedTCA Signaling Blade is the extremely good scalability and reliability achieved by TietoEnator's unique Horizontal Distribution architecture. Other benefits of the solution are:

- Very high link density:
- up to 256 TDM links per controller board
- up to 16 Ethernet links (SIGTRAN) per board
- up to 8 OC3/STM-1 ATM interfaces per board
- Simultaneous support for all types of transport:
- TDM (E1/T1/J1)
- ATM (OC3/STM-1)
- IP (SIGTRAN)
- Horizontally distributed signaling processing:
- Load sharing over multiple signaling CPU blades
- N + X redundancy scheme
- High capacity, more than 10,000 TCAP transactions per second / signaling CPU blade
- Designed for high availability, 99.999%

Features

- Signaling solution adopted for AdvancedTCA
- Protocols: MTP, SCCP, TCAP, MAP, ANSI41, INAP, BSSAP, CAP and ISUP in compliance with ITU, ANSI, TTC and Chinese SS7 standards
- SIGTRAN in compliance with IETF standards, M3UA (RFC 3332) and SCTP (RFC 2960)
- Access to Signaling protocols through distributed easy-to-use APIs that are linked together with the application
- Thread-safe, call-back C/C++ protocol APIs available for Sun Solaris®, Windows® NT, Windows® 2000, Windows 2003 Server®, Red Hat® Enterprise Linux®, Montavista® CG Linux® HP-UX® and IBM AIX® in application environment
- Java APIs available for TCAP; ETSI MAP; INAP; CAP
- Possibility for multiple applications to use the same protocol stack with load sharing between application instances
- Link capacity extension without installing new communica tion boards or rebooting the system
- Horizontally Distributed architecture, scalable from one signaling CPU blade and upwards
- Graphical Java-based configuration & management tools

TietoEnator

Building the Information Society

System architecture
The typical signaling solution consists of one or several ATCA Signaling Blades networked together using the ATCA chassis Gigabit Ethernet Base Channels.
Each ATCA Signaling Blade can be equipped with up to four interface modules for TDM, ATM or IP. Incoming and outgoing traffic are load-shared over the available Signaling Blades.

Hardware architecture
The AdvancedTCA Signaling Blade is equipped with:
- Intel® Pentium-M® 1.6 GHz CPU
- Two 1000baseT Base Channel i/f
- On-board Gbit Ethernet switch connecting:
  - P4 CPU
  - 4 PMC slots
  - Base Channels
  - Fabric Channel 1 & 2
- 1 - 4 SS7-PMC interface boards
  - 4 x E1/T1/J1 or 2 x HSL
- 1 - 4 Ethernet interface boards
  - 4 x 100BaseT
- 1 - 4 ATM interface boards
  - 2 x STM1/OC-3
- AdvancedTCA PICMG 3.0 compliant
- Montavista Carrier Grade Linux
- Software boot via onboard hard disk, solid state memory or network boot via Ethernet
- Integrated Platform Management Interface (IPMI) support
- Prepared for SA Forum High Availability architecture
- Watchdog timer monitoring of software and hardware
- Blade Hot swap support

Prerequisites
- ATCA-compatible chassis
- Application environment: SUN Solaris®, Windows® 2000, Windows® NT, Windows® 2003 Server, Red Hat®, Enterprise Linux®, Montavista® CG Linux® or IBM AIX

AdvancedTCA Signaling Blade

Adress: Kanikenäsbanken 12
S-651 15 Karlstad, Sweden
Telephone: +46 54 29 40 00
E-post: signaling@tietoenator.com
URL: www.tietoenator.com

AXP1600

AdvancedTCA Shelf

DATASHEET

MOTOROLA

KEY FEATURES

16-slot, 23" AdvancedTCA® shelf

Small footprint ,12U form factor

Including redundant shelf management, fan tray and power entry modules

Front access for all blades, shelf manager and fan tray modules

Rear access for power entry modules and most I/O connectivity

Complies with AdvancedTCA PICMG® 3.0 and 3.x standards

Dual star configuration support for base and fabric interfaces

NEBS compliant air filter design

The AXP1600 was specifically designed to address the unique carrier-grade requirements of the telecommunications industry. Application examples include wireless infrastructure, packetized voice, wireline data, and cable network head-end equipment.

The AXP1600 shelf includes all the sub-components required to create a complete AdvancedTCA platform and is ready for billable application blades. Redundant shelf manager and alarm modules, redundant power entry modules (PEMs), network synchronization clock (NSC) distribution modules and an N+1 cooling architecture via nine tray modules are all included.

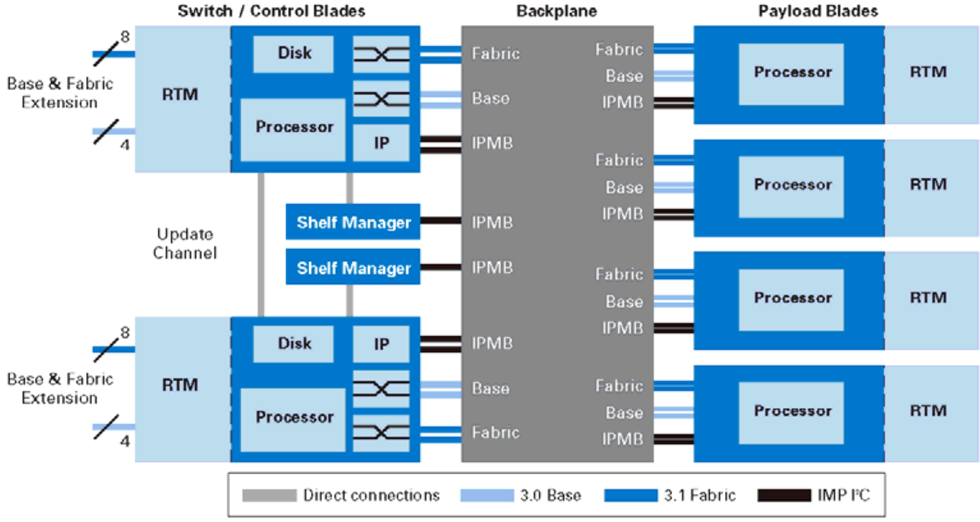
A replaceable, NEBS compliant air filter is also included. The 12U form factor allows for three complete AdvancedTCA shelves configured in a standard 42U telecom rack with several U-slots available for breaker panels and management elements.

By supporting both the PICMG 3.0 base interface and PICMG 3.1 fabric interface switching the AXP1600 shelf offers a wide range of configurations including separation of the control and data plane traffic – a key feature for many telecommunications applications.

AdvancedTCA®

The AXP1600 shelf is a heart of the Centellis™ 3600 AdvancedTCA Application-Enabling Platform from Motorola. The Centellis 3000 series represents a quantum leap forward in platform outsourcing by providing highly integrated and verified hardware and software components, reducing development costs, and accelerating time-to-market. This allows telecommunications equipment manufacturers (TEMs) to focus their development efforts on critical, differentiating features that provide a competitive advantage.

BLOCK DIAGRAM



SYSTEM ARCHITECTURE

Special attention was paid to architecting the AXP1600 shelf such that it could accommodate optional features as well as different PICMG 3.x future fabric interface technologies. The fabric interface backplane routing was implemented to utilize any PICMG 3.x compliant fabric technology by populating the shelf with the appropriate switch (hub) and application (node) blades. Motorola's first generation hub and node blades support PICMG 3.1 Gigabit Ethernet fabric technology. The PICMG 3.1 fabric standard allows for several configuration options in order to provide greater bandwidth to particular slots within the shelf. When the AXP1600 shelf is operating with the ATCA-F101 system controller and switch blade, the following PICMG 3.1 bandwidth options are available:

– PICMG 3.1, option 1 (1.0Gbps) — Slots 4, 6, 8, 11, 13, 15
– PICMG 3.1, option 2 (2.0Gbps) — Slots 3, 5, 7, 10, 12, 14
– PICMG 3.1, option 3 (4.0Gbps) — Slots 2, 9

SHELF HARDWARE

CHASSIS

12U chassis
16 slots for 8U blades
16 slots for 8U RTMs
Front to rear cooling architecture
ESD and earth grounding points

SHELF MANAGEMENT

N+1 redundancy architecture
Two (2) shelf management and alarm module slots
Embedded Telco Alarm functionality

COOLING

N+1 redundancy architecture
Six (6) top fan try module slots
Three (3) bottom fan tray module slot

BACKPLANE

Zone 1
– Redundant, bussed IPMI to all blade slots
– Redundant, bussed –48 VDC to all blade slots
Zone 2
– Dual star configuration for the base interface
– Dual star configuration for the fabric interface
– Update channel routing for all blade slots
– Three redundant, bussed telecom clock signals to all blade slots to all blade slots
Zone 3
– PICMG 3.0 defined open area, application specific

POWER DISTRIBUTION

N+1 redundancy architecture
Two (2) PEM slots

SHELF LAYOUT & DIMENSIONS

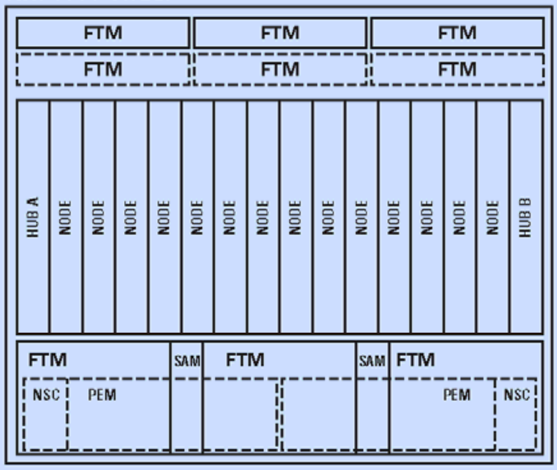
FRONT (TOP TO BOTTOM)
Three (3) top fan try module slots
16 vertical 8U blade slots
Air inlet/air filter tray
Cable management system
Three (3) bottom fan tray module slot
Two (2) shelf management and alarm module slots

REAR (TOP TO BOTTOM)
Air outlet
Three (3) top fan try module slots
16 vertical 8U RTM slots
Two (2) PEM slots
Two (2) NSC distribution modules

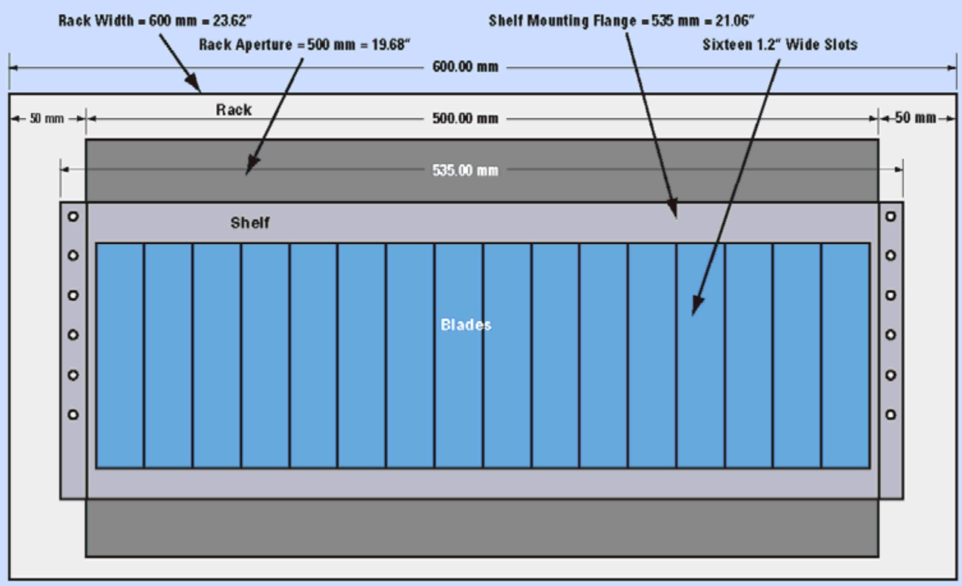
SHELF DIMENSIONS
Height: 20.97″/532.6 mm (12U)
Width: 19.50″/495.3 mm
Depth: 19.70″/500.4 mm

The figure below shows the general expected physical locations of modules in the AXP1600 Shelf. Dotted lines denote modules located in the rear of the shelf.

AXP1600 SHELF



"STANDARD" 600 MM ETSI 300 119-3 RACK



(From PICMG-3.0 Rev 1.0 Table 2-22 (pg 101)

SHELF COMPONENTS

SHELF MANAGEMENT AND ALARM MODULE

The purpose of shelf management, as defined by the PICMG 3.0 standard, is to assure proper operation of AdvancedTCA blades and other shelf components within one or more shelves. The shelf management entity continually monitors all low-level hardware functionality (inventory, sensor, and status data, etc.) and reports status to the system manager. It also provides control access to these attributes. Management access to this information is provided via local console and Ethernet interfaces as well as the Service Availability™ Forum (SA Forum) defined HPI interface. Each blade and major shelf accessory has an Intelligent Platform Management Controller (IPMC) that is responsible for providing this information to the shelf management entity.

The AXP1600 shelf provides redundant shelf management functionality utilizing an active/standby architecture. In addition, the Telco Alarm functionality is integrated into the same module to maximize critical real estate within the system, this functionality is also redundant. Visual indicators, as well as physical interfaces are provided for direct, front panel access.

FRONT PANEL ACCESS

- RS-232, console, RJ-11
- 10/100BaseT Ethernet, RJ-45
- Telco Alarm interface, dry relay contact, DB-15

TELCO ALARM LED STATUS INDICATORS

- Critical/major/minor: red/red/amber
- In service/out-of-service: green/red (US) or green/amber (Europe)

SHELF MANAGER LED STATUS INDICATOR

- Hot swap: blue

FAN TRAY MODULES

The AXP1600 shelf provides fault-tolerant cooling to all front blades and RTM slots in an N+1 cooling architecture that is implemented using nine fan tray modules; six on the top of the shelf and three at the bottom. Each fan tray module has redundant fans and a complete IPMC complex to interface with the shelf management entity.

GENERAL CHARACTERISTICS

- Front blade cooling capacity: 200 watts per blade
- RTM blade cooling capacity: 25 watts per blade
- Automatic fan speed control
- Front access
- Operating range: 5°C to 55°C

TOP FAN TRAY MODULE

- Two (2) fans per fan tray module
- Local IPMC

LED STATUS INDICATORS

- In service/out-of-service: green/red (US) or green/amber (Europe)
- Hot swap: blue

BOTTOM FAN TRAY MODULE

- Two (2) fans per fan tray module
- Local IPMC

POWER ENTRY MODULE (PEM)

Power conditioning for the AXP1600 shelf is provided by a pair of redundant PEMs. They provide wiring studs for connection to redundant –48 VDC and/or –60 VDC power sources and provide power to the backplane on the redundant –48 VDC power rails for blades and other shelf components.

GENERAL CHARACTERISTICS

- Input voltage range (–40 VDC to –72 VDC)
- 100 amp maximum capacity via four (4) 25 amp circuits
- EMI filtering
- Transient voltage suppression
- Internal redundant fans
- Rear access

ATCA-S100, R3

AdvancedTCA Storage Blade

PRELIMINARY DATASHEET

MOTOROLA

KEY FEATURES

Fibre Channel based AdvancedTCA® storage blade

RoHS (5 of 6) compliant

PICMG® 3.0 compliant

Single-slot storage solution

Up to 300GB capacity

Multiple configurations available

1.0Gbps and 2.0Gbps support

Carrier Grade Linux drivers included with host blades

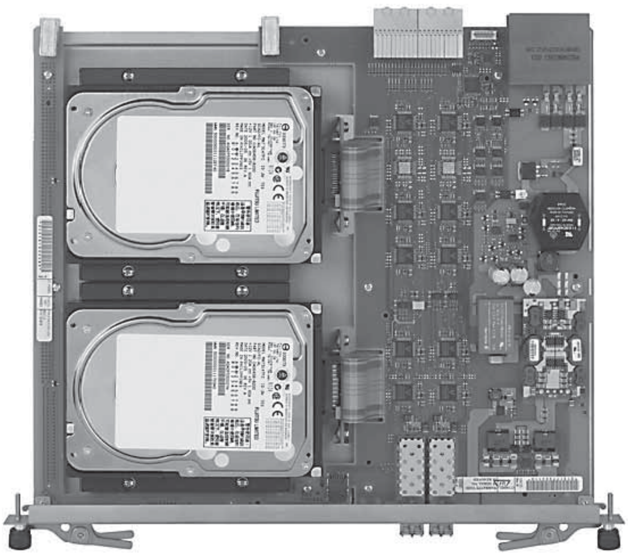
Designed for NEBS and ETSI compliance

The ATCA-S100 storage blade provides single or dual Fibre Channel based hard drive devices on a single blade. Several storage architectures are supported including RAID-0, RAID-1 and RAID-10.

The ATCA-S100 blades are RoHS (5 of 6) compliant, eliminating the need for customers to incur the time, resource and expense associated with creating and/or converting existing product to meet this international requirement.

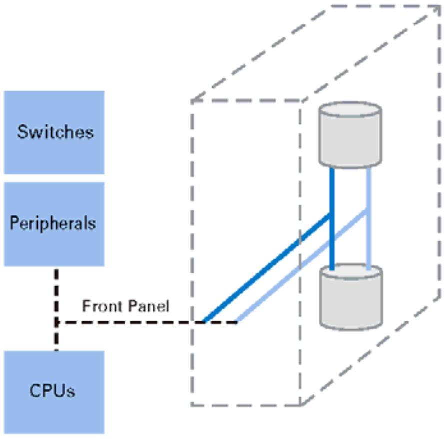
Redundant front panel connections, via small form factor pluggable (SFP) connectors, enable external connections to Fibre Channel enabled ATCA® host blades as well as any Fibre Channel compliant storage array.

The ATCA-S100 storage blade is equipped with one or two highly reliable 3.5-inch, enterprise class hard drive devices. Storage capacity scales from 150GB to 300GB per blade. Multiple blades can be used within an AdvancedTCA shelf.



The Motorola ATCA-S100 AdvancedTCA storage blade allows native Fibre Channel based hard disk drive (HDD) expansion for AdvancedTCA systems. This scalable blade-based architecture, together with on-board interface redundancy, provides cost-effective disk capacity expansion for state-of-the-art high availability AdvancedTCA platforms.

BLOCK DIAGRAM



STANDARD NETWORK SUPPORT

Two standard Fibre Channel interfaces are provided on the front panel for external network connectivity.

ADVANCEDTCA INTELLIGENT PLATFORM MANAGEMENT

The PICMG 3.0 AdvancedTCA standard specifies a low-level, environmental management architecture referred to as intelligent platform management interface (IPMI). The ATCA-717 blade implements this functionality using an off-the-shelf hardware and software-based IPM controller that monitors all local, blade-specific environmental information. Management access to this information is provided through the Service Availability™ Forum (SA Forum) defined HPI interface.

HARDWARE

MEMORY CAPACITY

ATCA-S100/150-1/5E (150GB)

ATCA-S100/150-2/5E (300GB)

EXTERNAL INTERFACES

Front Panel

- Small form factor pluggable (2)
- Copper and optical options

POWER REQUIREMENTS

Dual-redundant –48V rail

Input range: 39.5 – 72 VDC

Typical power: 120W maximum

THERMAL CHARACTERISTICS

Operating range: –5° C to 55° C

BLADE SIZE

8U form factor, 280 mm X 322.5 mm, single slot

RELEVANT STANDARDS

PICMG 3.0 (form factor, IPMI, hot swap, RTM)

ORDERING INFORMATION

Part Number	Description
ATCA-S100/150-2/5E	ATCA storage blade with 300GB and Fibre Channel interface (RoHS 5/6)
ATCA-S100/150-1/5E	ATCA storage blade with 150GB and Fibre Channel interface (RoHS 5/6)
ATCA-S100-F/6E	Fiber cable (1m) with two fibre SFP transceivers for ATCA-S100 (RoHS 6/6)
ATCA-S100-C/6E	Copper cable (1m) with two fibre SFP transceivers for ATCA-S100 (RoHS 6/6)

REGULATORY COMPLIANCE

Item	Description
Designed to comply with NEBS	GR-63-CORE, NEBS Physical Protection, Level 3
	GR-1089-CORE, Electromagnetic Compatibility and Electrical Safety — Generic Criteria for Network Telecommunications Equipment. Level 3, Equipment Type 2
Designed to comply with ETSI	ETSI Storage, ETS 300 019-2-1, Class 1.2 equipment, Not Temperature Controlled Storage Locations
	ETSI Transportation, ETS 300 019-2-2, Class 2.3 equipment, Public Transportation
	ETSI Operation, ETS 300 019-2-3, Class 3.2 equipment, Partly Temperature Controlled Locations
Designed to comply with Acoustic	ETS-300-753, Equipment Engineering (EE); Acoustic noise emitted by telecommunications equipment
EMC	EN-300-386 Electromagnetic compatibility and Radio spectrum Matters (ERM); telecommunication network equipment; ElectroMagnetic Compatibility (EMC) requirements, Telecommunication equipment room (attended)
	FCC 47 CFR Part 15 Subpart B (US), Class A
	EMC Directive 89/336/EEC (EU)
	AS/NZS 3548 (Australia/New Zealand), Limits and Methods of Measurement of Radio Disturbance Characteristics of Information Technology Equipment
	VCCI Class A (Japan), Voluntary Control Council for Interference by Information Technology Equipment
Safety	Compliance to UL/CSA 60950-1, EN 60950-1 and IEC 60950-1 CB Scheme. Marked with U.S. NRTL, Canadian Safety and CE Mark. Safety of information technology equipment, including electrical business equipment
	ETS 300-132-2 Environmental Engineering (EE); Power supply interface at the input to telecommunications equipment; Part 2: Operated by direct current (dc)
RoHS/WEEE compliance	DIRECTIVE 2002/95/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS)
	DIRECTIVE 2002/96/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on waste electrical and electronic equipment (WEEE)

SOLUTION SERVICES

Motorola provides a portfolio of solution services optimized to meet your needs throughout the product lifecycle. Design services help speed time to market. Deployment services include global 24x7 technical support. Renewal services enable product longevity and technology refresh. And solution extras include enhanced warranty and repairs.

Sales Offices

Tempe, AZ U.S.A. 1 800 759 1107 or +1 602 438 5720
Paris, France +33 1 69 35 25 88
Munich, Germany +49 89 608 14-0
Loughborough, UK +44 1509 634300
Tel Aviv, Israel +972 3 568 4385
Shanghai, China +86 215292 5693
Tokyo, Japan +81 3 5424 3101
Hong Kong, China +852 2966 3209

MOTOROLA

www.motorola.com/computing

ATCAS100R3-D0 06/06

ATCA-7221, R3

AdvancedTCA Processor Blade

PRELIMINARY DATASHEET

MOTOROLA

KEY FEATURES

High performance processor blade with SMP support

RoHS (6 of 6) compliant

Dual 2.8 GHz Intel® Xeon™ LV processors

Complete software operating environment including OS

PICMG 3.0 Gigabit Ethernet base interface support

PICMG 3.1, Option 1 and 2 fabric interface support

PMC site for I/O flexibility

SATA or SAS hard disk drive options

Service Availability™ Forum (SA Forum) compliant HPI and AIS software interfaces

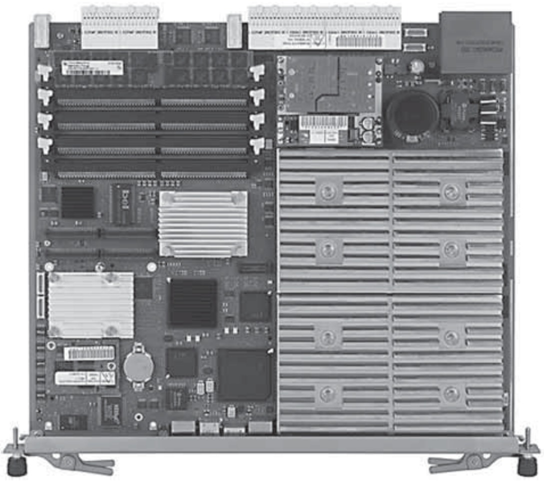
Designed for NEBS and ETSI compliance

The ATCA-7221 processor blade delivers a combination of performance, functionality and cost efficiency to help drive the successful implementation of next generation telecom networks. It builds on the AdvancedTCA® (ATCA®) standard to provide the right product at the right time to meet the needs of the telecom industry.

The ATCA-7221 blades are RoHS (6 of 6) compliant, eliminating the need for customers to incur the time, resource and expense associated with creating and/or converting existing product to meet this international requirement.

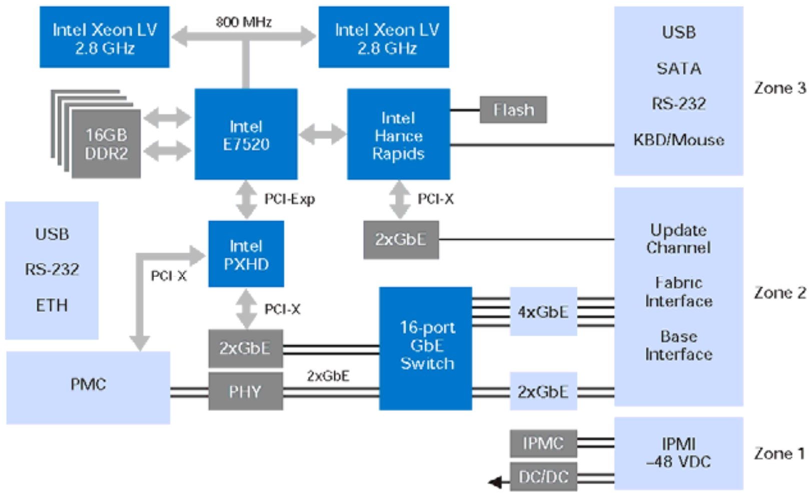
With dual Intel Xeon processors, the ATCA-7221 processor blade is the highest performance processing blade in the AdvancedTCA form factor. It also provides Gigabit Ethernet interfaces to the PICMG® 3.0 base interface and the PICMG 3.1 fabric interface in a dual star configuration. This allows the base interface to be used for control, signaling, or management while the fabric interface can be used for high performance data transport, providing control and data traffic separation.

An array of memory options, including up to 16GB of main memory, SAS and SATA storage interfaces and multiple hard drive options add to the performance and flexibility of the ATCA-7221 processor blade.



The Motorola ATCA-7221 processor blade is designed to operate within both the Motorola Centellis™ 3000 and Avantellis™ 3000 series of communications servers. Dual Intel Xeon LV processors operating at 2.8 GHz in a two-way symmetric multiprocessing (SMP) configuration deliver outstanding performance. The ATCA-7221 blade is ideal for implementing almost any control plane applications for next generation wireless and wireline telecommunications infrastructure equipment.

BLOCK DIAGRAM



STANDARD NETWORKING SUPPORT

The ATCA-7221 processor blade provides PICMG 3.0 base interface connectivity in a dual star configuration using standard Gigabit Ethernet technology. The PICMG 3.1 fabric interface is also supported and several configurations are available depending on application bandwidth requirements.

- PICMG 3.1, Option 1 – Single, redundant Gigabit Ethernet pair (1.0Gbps)
- PICMG 3.1, Option 2 – Dual, redundant Gigabit Ethernet pairs (2.0Gbps)

PROCESSOR COMPLEX

Surrounding the dual Intel Xeon SMP processor configuration is an array of high performance components that combine to form a powerful processor complex.

FEATURES INCLUDE:

- Up to 16.0GB, ECC-protected DDR2 SDRAM. Supported configurations: 2, 4, 8GB (16GB future)
- Intel® E7520 high performance chip set
- Dual 64MB flash, dual bank architecture
- SATA or SAS hard disk drive (HDD) on RTM
- Dual, external Ultra320 SCSI interfaces on RTM
- Dual external Fibre Channel interfaces on RTM

SOFTWARE SUPPORT

All Motorola ATCA blades can be configured with optional software that, when combined with the hardware, create a fully integrated and verified telecom platform. Two software packages are available:

- Centellis 3000 software package
- Avantellis 3000 software package

The Centellis-Avantellis 3000 platforms come complete with, and are verified with, a standard Carrier Grade Linux (CGL) distribution; MontaVista CGE 4.0, the market leader in Carrier Grade Linux. MontaVista CGE 4.0 comes complete with all required Linux Support Packages (LSPs) to support Motorola ATCA blades as well as several userland applications.

The Centellis 3000 software package comes complete with:

- MontaVista CGE 4.0
- Basic Blade Services

Basic Blades Services (BBS) software is provided to enable a set of ATCA hardware and software components into a fully integrated and verified telecom platform – The Centellis 3000 platform. This platform is ready for customers HA middleware and application environment.

Basic Blade Services (generic to all ATCA blades):

- Hardware Platform Management including local IPMC, LED, EKeying and blade extraction software
- Firmware upgrade utility
- Local management access (SNMP, CLI)

The Avantellis 3000 software package comes complete with:

- MontaVista CGE 4.0
- Platform Control Software
- NetPlane® Core Services

NetPlane Core Services (NCS) is Motorola's SA Forum compliant HA middleware combined with select complimentary services to create a complete service availability solution. Platform Control Software (PCS) is lower level software that binds the platform independent NCS software to the specific hardware platform. These software packages, combined with MontaVista CGE 4.0, provide a high availability platform ready for customer applications – the Avantellis 3000 platform.

RELEVANT STANDARDS

- Open Source Development Labs (OSDL), rev. 1.0
- SA Forum
  - Hardware Platform Interface (HPI) – rev. 1.0, A .01.01
  - Application Interface Specification (AIS) – rev. 1.0, A .01.01

For more information on the Centellis and Avantellis 3000 platforms, please refer to the Centellis 3000 and Avantellis 3000 series datasheets.

INTELLIGENT PLATFORM MANAGEMENT CONTROL

The PICMG 3.0 AdvancedTCA standard specifies a low-level, environmental management architecture referred to as intelligent platform management interface (IPMI). The ATCA-7221 blade implements this functionality using an off-the-shelf hardware and software based IPM controller that monitors all local, blade-specific environmental information. Management access to this information is provided through the SA Forum defined HPI interface.

REAR TRANSITION MODULES

Motorola offers two rear transition modules (RTMs) for the ATCA-7221 processor blade for the ultimate flexibility in storage interconnect and capacity options as well as external Gigabit Ethernet connectivity.

PMC SITE

The ATCA-7221 blade includes a single width IEEE1386.1-2001 PMC site. The PMC site supports PCI-X 64-bit 133 MHz capable PMC modules.

HARDWARE

PROCESSOR

Dual 2.8 GHz Intel Xeon LV processors in an SMP configuration

1MB L2 on-chip cache

800 MHz frontside bus

Intel® E7520 system controller

MEMORY

Up to 16GB, ECC-protected SDRAM. Supported configurations – 2GB, 4GB, 8GB, 16GB (future)

256Byte CMOS NVRAM for BIOS configuration

1.0MB boot flash, dual bank architecture

32MB application flash, dual bank architecture

16MB CPU reset-persistent memory

COUNTERS/TIMERS

Real-time clock

Programmable watchdog timer

PCI MEZZANINE CARD

PMC site with 64-bit 133 MHz PCI-X interface

Dual Gigabit Ethernet interface

BASE AND FABRIC INTERFACES

Dual star configuration

PICMG 3.0 base interface compliant, Gigabit Ethernet (1.0Gbps)

PICMG 3.1 fabric interface compliant, Gigabit Ethernet

- PICMG 3.1, Option 1 – Single, redundant Gigabit Ethernet pair (1.0Gbps)
- PICMG 3.1, Option 2 – Dual, redundant Gigabit Ethernet pairs (2.0Gbps)

EXTERNAL INTERFACES

Front Panel

- USB 2.0, mini USB Type AB (2)
- Serial, RJ-45 (2)
- 10/100/1000BaseT Ethernet, RJ-45 (1)

Via Optional RTM (RTM-ATCA-7221/SCS/6E)

- USB 2.0, Type A (2)
- Serial, RJ-45 (2)
- SAS interface (1)
- Keyboard/mouse, PS2 (1)
- SCSI interface (2)
- Gigabit Ethernet (2)
- SAS HDD slot (1)

Via Optional RTM (RTM-ATCA-7221/FC/6E)

- USB 2.0, Type A (2)
- Serial, RJ-45 (2)
- Keyboard/mouse, PS2 (1)
- SATA HDD slot (1)

POWER REQUIREMENTS

Dual-redundant –48V rail

Input range: 39.5 – 72 VDC

Typical power: 120 – 140W

THERMAL CHARACTERISTICS

Operating range: –5° C to 55° C

RELEVANT BLADE SIZE

8U form factor, 280 mm X 322.5 mm, single slot

RELEVANT STANDARDS

PICMG 3.0 (form factor, IPMI, base interface, hot swap, RTM)

PICMG 3.1, Option 1 and 2

ORDERING INFORMATION

Part Number	Description
ATCA-7221/2G/6E	ATCA processor blade with dual Intel Xeon processors (2.8 GHz), 2GB memory and 1 PMC slot (RoHS 6/6)
RTM-ATCA-7221/FC/6E	RTM for the ATCA-7221 blade with dual FC interfaces (copper), optional SATA HDD (RoHS 6/6)
RTM-ATCA-7221/SCS/6E	RTM for the ATCA-7221 blade with dual SCSI & GE interfaces, optional SAS HDD (RoHS 6/6)
ATCA-7221/HD/SATA/6E	60GB SATA HDD and mounting kit for the RTM-ATCA-7221/FC/6E (RoHS 6/6)
ATCA-7221/HD/SAS/6E	72GB SAS HDD and mounting kit for the RTM-ATCA-7221/SCS/6E (RoHS 6/6)
ATCA-7221/MEM-2G/6E	2GB DDR memory module for ATCA-7221 (RoHS 6/6)

Appendix C – SQL create table

attribute_keywords

SQL:

```
CREATE TABLE `attribute_keywords` (
    `attribute_keyword_id` INT( 10 ) NOT NULL ,
`attribute_keyword` VARCHAR( 32 ) NOT NULL ,
`system_block_id` INT( 10 ) NOT NULL ,
`attribute_description` VARCHAR( 64 ) ,
PRIMARY KEY ( `attribute_keyword_id` ) ,
UNIQUE (
    `attribute_keyword` ,
`system_block_id`
    )
    ) TYPE = MYISAM ;
```

attribute_values

SQL:

```
CREATE TABLE `attribute_values` (
    `attribute_value_id` INT( 10 ) NOT NULL ,
`attribute_keyword_id` INT( 10 ) NOT NULL ,
`attribute_value` VARCHAR( 64 ) NOT NULL ,
`attribute_value_description` VARCHAR( 64 ) ,
PRIMARY KEY ( `attribute_value_id` ) ,
UNIQUE (
    `attribute_keyword_id` ,
`attribute_value`
    )
    ) TYPE = MYISAM ;
```

components

SQL :

```
CREATE TABLE `components` (
    `component_id` INT( 10 ) NOT NULL AUTO_INCREMENT ,
`component_name` VARCHAR( 64 ) NOT NULL ,
`component_version` VARCHAR( 64 ) ,
`component_others` VARCHAR( 64 ) ,
PRIMARY KEY ( `component_id` ) ,
UNIQUE (
    `component_name` ,
`component_version`
    )
    ) TYPE = MYISAM ;
```

component_attribute
SQL:

```
CREATE TABLE `component_attribute` (
    `component_id` INT( 10 ) NOT NULL ,
`attribute_value_id` INT( 10 ) NOT NULL
    ) TYPE = MYISAM ;
```

component_component
SQL:

```
CREATE TABLE `component_component` (
    `component_id` INT( 10 ) NOT NULL ,
`compliant_component_id` INT( 10 ) NOT NULL ,
`compliant_component_state` VARCHAR( 64 ) ,
UNIQUE (
    `component_id` ,
`compliant_component_id`
    )
    ) TYPE = MYISAM ;
```

component_organization

SQL:

```
CREATE TABLE `component_organization` (
    `component_organization_id` INT( 10 ) NOT NULL AUTO_INCREMENT ,
`component_id` INT( 10 ) NOT NULL ,
`organization_id` INT( 10 ) NOT NULL ,
`organization_type` VARCHAR( 64 ) ,
`component_homepage` VARCHAR( 64 ) ,
`component_documentation_hyperlink` VARCHAR( 64 ) ,
`organization_support` VARCHAR( 64 ) ,
`warranty_time` VARCHAR( 64 ) ,
`warranty_description` VARCHAR( 64 ) ,
`component_organization_others` VARCHAR( 64 ) ,
PRIMARY KEY ( `component_organization_id` ) ,
UNIQUE (
    `component_id` ,
`organization_id`
    )
    ) TYPE = MYISAM ;
```

component_price

SQL:

```
CREATE TABLE `component_price` (
    `component_organization_id` INT( 10 ) NOT NULL ,
`component_price_time` VARCHAR( 32 ) NOT NULL ,
`component_price` VARCHAR( 32 ) NOT NULL ,
`component_price_description` VARCHAR( 64 ) ,
UNIQUE (
    `component_organization_id` ,
`component_price_time` ,
`component_price`
    )
    ) TYPE = MYISAM ;
```

component_purchasing

SQL:

```
CREATE TABLE `component_purchasing` (
    `component_organization_id` INT( 10 ) NOT NULL ,
`component_purchasing_time` DATE NOT NULL ,
`component_purchaser` VARCHAR( 32 ) ,
`component_purchasing_unit_price` VARCHAR( 32 ) ,
`component_purchasing_number` INT( 10 ) ,
`component_order_number` VARCHAR( 32 ) ,
UNIQUE (
    `component_organization_id` ,
`component_purchasing_time`
    )
    ) TYPE = MYISAM ;
```

component_requirement

SQL:

```
CREATE TABLE `component_requirement` (
    `component_id` INT( 10 ) NOT NULL ,
`requirement_value_id` INT( 10 ) NOT NULL
    ) TYPE = MYISAM
```

component_standard

SQL:

```
CREATE TABLE `component_standard` (
    `component_id` INT( 10 ) NOT NULL ,
`standard_id` INT( 10 ) NOT NULL ,
`component_compliant_standard_state` VARCHAR( 64 ) ,
UNIQUE (
    `component_id` ,
`standard_id`
    )
    ) TYPE = MYISAM ;
```

component_system_block

SQL:

```
CREATE TABLE `component_system_block` (
    `component_id` INT( 10 ) NOT NULL ,
`system_block_id` INT( 10 ) NOT NULL
    ) TYPE = MYISAM ;
```

organizations

SQL:

```
CREATE TABLE `organizations` (
    `organization_id` INT( 10 ) NOT NULL AUTO_INCREMENT ,
`organization_name` VARCHAR( 64 ) NOT NULL ,
`organization_homepage` VARCHAR( 64 ) ,
`organization_description` VARCHAR( 64 ) ,
`organization_others` VARCHAR( 64 ) ,
PRIMARY KEY ( `organization_id` ) ,
UNIQUE (
    `organization_name`
    )
    ) TYPE = MYISAM ;
```

organization_organization

SQL:

```
CREATE TABLE `organization_organization` (
    `organization_id` INT( 10 ) NOT NULL ,
`compliant_organization_id` INT( 10 ) NOT NULL ,
`compliant_organization_state` VARCHAR( 64 ) ,
UNIQUE (
    `organization_id` ,
`compliant_organization_id`
    )
    ) TYPE = MYISAM ;
```

organization_standard

SQL:

```
CREATE TABLE `organization_standard` (
    `organization_id` INT( 10 ) NOT NULL ,
`standard_id` INT( 10 ) NOT NULL ,
UNIQUE (
    `organization_id` ,
`standard_id`
    )
    ) TYPE = MYISAM ;
```

requirement_keywords

SQL:

```
CREATE TABLE `requirement_keywords` (
    `requirement_keyword_id` INT( 10 ) NOT NULL ,
`requirement_keyword` VARCHAR( 32 ) NOT NULL ,
`system_block_id` INT( 10 ) NOT NULL ,
`requirement_description` VARCHAR( 64 ) ,
PRIMARY KEY ( `requirement_keyword_id` ) ,
UNIQUE (
    `requirement_keyword` ,
`system_block_id`
    )
    ) TYPE = MYISAM
```

requirement_values

SQL:

```
CREATE TABLE `requirement_values` (
    `requirement_value_id` INT( 10 ) NOT NULL ,
`requirement_keyword_id` INT( 10 ) NOT NULL ,
`requirement_value` VARCHAR( 64 ) NOT NULL ,
`requirement_value_description` VARCHAR( 64 ) ,
PRIMARY KEY ( `requirement_value_id` ) ,
UNIQUE (
```

```
`requirement_keyword_id` ,
`requirement_value`
)
) TYPE = MYISAM
```

standards

SQL :

```
CREATE TABLE `standards` (
`standard_id` INT( 10 ) NOT NULL AUTO_INCREMENT ,
`standard_name` VARCHAR( 64 ) NOT NULL ,
`standard_version` VARCHAR( 64 ) ,
`standard_file_hyperlink` VARCHAR( 64 ) ,
`standard_homepage` VARCHAR( 64 ) ,
`standard_status` VARCHAR( 32 ) ,
`standard_issued_time` VARCHAR( 10 ) ,
`standard_description` VARCHAR( 64 ) ,
`standard_others` VARCHAR( 64 ) ,
PRIMARY KEY ( `standard_id` ) ,
UNIQUE (
`standard_name` ,
`standard_version`
)
) TYPE = MYISAM ;
```

standard_attribute

SQL:

```
CREATE TABLE `standard_attribute` (
`standard_id` INT( 10 ) NOT NULL ,
`attribute_value_id` INT( 10 ) NOT NULL
) TYPE = MYISAM ;
```

standard_requirement

SQL:

```
CREATE TABLE `standard_requirement` (
    `standard_id` INT( 10 ) NOT NULL ,
`requirement_value_id` INT( 10 ) NOT NULL
    ) TYPE = MYISAM ;
```

standard_system_block

SQL:

```
CREATE TABLE `standard_system_block` (
    `system_block_id` INT( 10 ) NOT NULL ,
`standard_id` INT( 10 ) NOT NULL ,
UNIQUE (
    `system_block_id` ,
`standard_id`
    )
    ) TYPE = MYISAM ;
```

system_blocks

SQL:

```
CREATE TABLE `system_blocks` (
    `system_block_id` INT( 10 ) NOT NULL ,
`system_block_name` VARCHAR( 64 ) NOT NULL ,
`system_block_description` VARCHAR( 64 ) ,
`system_block_others` VARCHAR( 64 ) ,
PRIMARY KEY ( `system_block_id` ) ,
UNIQUE (
    `system_block_name`
    )
    ) TYPE = MYISAM ;
```

system_block_hierarchy

SQL:

```
CREATE TABLE `system_block_hierarchy` (
    `system_block_id` INT( 10 ) NOT NULL ,
`prent_system_id` INT( 10 ) NOT NULL ,
) TYPE = MYISAM ;
```

system_block_prerequisite
SQL: CREATE TABLE `system_block_prerequisite` (`system_block_id` INT(10) NOT NULL , `prerequisite_system_block_id` INT(10) NOT NULL) TYPE = MYISAM ;