Chapter – 2
IP Multimedia Subsystem Architecture, Related Technologies & Tools

2.1 Requirement for IMS Architecture
   2.1.1 IP Multimedia Sessions
   2.1.2 IP connectivity
   2.1.3 Ensuring quality of service for IP multimedia services
   2.1.4 IP policy control for ensuring correct usage of media resources
   2.1.5 Secure Communication
   2.1.6 Charging arrangements
   2.1.7 Support of roaming
   2.1.8 Interworking with other networks
   2.1.9 Service control model
   2.1.10 Service development
   2.1.11 Layered design
   2.1.12 Access independence

2.2 Description of IMS-related entities and functionalities
   2.2.1 Call Session Control Functions (CSCF)
      2.2.1.1 Proxy Call Session Control Function (P-CSCF)
      2.2.1.2 Interrogating Call Session Control Function (I-CSCF)
      2.2.1.3 Serving Call Session Control Function (S-CSCF)
   2.2.2 Application Servers
      2.2.2.1 Gateways
   2.2.3 BGCF (Breakout Gateway Controller Functions)
   2.2.4 Media Resource Function (MRF)
      2.2.5 Reference points

2.3 IMS functional planes
   2.3.1 Transport plane
   2.3.2 Control plane
   2.3.3 Service plane

2.4 Databases

2.5 Service functions
   2.5.1 Interworking functions

2.6 IMS Related Technologies
2.6.1 Service Oriented Architecture
2.6.2 User Oriented Architecture
2.6.3 Event Driven Architecture
2.6.4 Web Services
   2.6.4.1 Extensible Markup Language (XML)
   2.6.4.2 Web Services Description Language (WSDL)
   2.6.4.3 Universal Description Discovery and Integration
   2.6.4.4 SOAP
   2.6.4.5 Web Services Invocation Framework
   2.6.4.6 Representation State Transfer (REST)
   2.6.4.7 OSA Parlay X
2.7 Service Orchestration
   2.7.1 Choreography v/s Orchestration
   2.7.2 Enterprise Service Bus
   2.7.3 Service Delivery Platforms
      2.7.3.1 Telephony Application Programming Interface (TAPI)
      2.7.3.2 SIP Common Gateway Interface (SIP CGI)
      2.7.3.3 Service Level Execution Environment
      2.7.3.4 SIP Servlets
      2.7.3.5 SDPs and IMS Requirements
2.8 Web 2.0 Principles
   2.8.1 General Characteristics
   2.8.2 Mashups
2.9 IMS Service Creation Environment & Toolkits
   2.9.1 IMS Service Development
   2.9.2 Unified Service Creation Environment (USCE)
   2.9.3 USCE (Unified Service Creation Environment) Toolkits for IMS
   2.9.4 Application Server Toolkit
   2.9.5 Perspective of AST V6.1 Tools
   2.9.6 SIP Servlet Application Development
      2.9.6.1 SIP only Applications
      2.9.6.2 Converged SIP/HTTP Applications
      2.9.6.3 SIP Servlet Deployment
      2.9.6.4 Sample SIP Services
2.10 Hardware & Software Requirement for Application Server Toolkit
2.10.1 Hardware Requirement
2.10.2 Supported Operating system

2.11 IMS Enablement Kit
2.11.1 IMS access gateways and enablers
2.11.2 IMS Enablement Tool Kit Components
2.11.3 Sample Application for IMS Foundation
2.11.4 Hardware & Software Requirement for IMS Enablement Tool Kit
2.11.5 Web Services Server Toolkit
2.11.6 Capabilities of Telecom Web Services Toolkit (TWSS)
2.11.7 Mediation Service
2.11.8 TWSS Access Gateway Mediation Primitives
2.11.9 TWSS Default Message Flow
2.11.10 Hardware & Software requirements for TWSS Toolkit

2.12 IMS Service Creation Tools
2.12.1 IMS service creation
2.12.2 Web tools platform for AST (Application Server Toolkit)
    2.12.2.1 Categories of web tool platform

2.13 Portlet Development Tool
2.13.1 Server Tool
2.13.2 Telecom Web services Tools

2.14 Tools Developing & Testing SIP and IMS Sample Applications
2.14.1 SIPp
2.14.2 Ethereal
2.14.2 SIPx Phone
2.14.3 Comparison of IMS traffic analyzing tools
2.14.4 DA 3400, DA 3600A VOIP Analysis [JDSUO]
2.14.5 PVA-1000 VOIP Analysis [JDSUO]
2.14.6 Packet Scan [GL Communication©]
2.14.7 Hammer Call Analyzer [Emprix©]
2.14.8 Hammer G5 [Emprix©]
2.14.9 THGNOTEBOOK [FinisarO]
2.14.10 VOIP testing [RadcomO]
2.14.11 nGenius (Infimtestream) [NETSCOUTO]

2.15 Summary
Chapter – 2
IP Multimedia Subsystem Architecture, Related Technologies & Tools

The IP-Multimedia Subsystem (IMS) defines the functional architecture for a managed IP-based network. It aims to provide a means for carriers to create an open, standards-based network that delivers integrated multimedia services to increase revenue, while also reducing network capital expenditure (CapEx) and operational expenditure (OpEx). IMS was originally designed for third-generation mobile phones, but it has already been extended to handle access from WiFi networks, and is continuing to be extended into an access-independent platform for service delivery, including broadband fixed-line access. It promises to provide seamless roaming between mobile, public WiFi and private networks for a wide range of services and devices.

The IMS architecture has been designed to enable operators to provide a wide range of real-time, packet-based services and to track their use in a way that allows both traditional time-based charging as well as packet and service-based charging. It has become increasingly popular both with wireline and wireless service providers as it is designed to increase carrier revenues, deliver integrated multimedia services, and create an open, standards-based network. The IP Multimedia Subsystem (IMS) is a standardized Next Generation Networking (NGN) architecture for telecom operators that want to provide mobile and fixed multimedia services. It uses a Voice-over-IP (VoIP) implementation based on a 3GPP-standardized implementation of SIP, and runs over the standard Internet Protocol (IP). Existing phone systems (both packet-switched and circuit-switched) are supported.

2.1 Requirement for IMS Architecture
There is a set of basic requirements, which guides the way in which the IMS architecture has been created evolve in the future. This section covers the most significant requirements. 3GPP IMS requirements are documented in [19].

2.1.1 IP Multimedia Sessions
Existing communication networks are able to offer voice, video and messaging type of services using circuit-switched bearers. Naturally, end-users’ service offerings should not decline when users move to the packet-switched domain and start using the IMS.

The IMS will take communication to the next level by offering enriched communication means. IMS users are able to mix and match a variety of IP-based services in any way they choose during a single communication session. Users can integrate voice, video and text, content sharing and presence as part of their communication and can add or drop services as and
when they choose. For example, two people can start a session as a voice session and later on add a game or video component to the same session.

### 2.1.2 IP connectivity

As the name IP Multimedia Subsystem implies, a fundamental requirement is that a device has to have IP connectivity to access it. Peer-to-peer applications require end-to-end reachability and this connectivity is easiest attainable with IP version 6 (IPv6) because IPv6 does not have address shortage. Therefore, 3GPP has arranged matters so that the IMS exclusively supports IPv6 [20]. However, early IMS implementations and deployments may use IPv4. 3GPP has created recommendations about how IP version interworking is handled in the IMS [21].

![Diagram](image)

**Figure - 2.1: IMS connectivity options when a user is roaming.**

IP connectivity can be obtained either from the home network or the visited network. The leftmost part of Figure 2.1 presents an option in which User Equipment (UE) has obtained an IP address from a visited network. In the UMTS network, this means that the Radio Access Network (RAN), Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN) are located in the visited network when a user is roaming in the visited network. The rightmost part of Figure 2.1 presents an option in which a UE has obtained an IP address from the home network. In the UMTS network this means that the RAN and SGSN are located in the visited network when a user is roaming in the visited network.

Obviously, when a user is located in the home network all necessary elements are in the home network and IP connectivity is obtained in that
network. It is important to note that a user can roam and obtain IP connectivity from the home network as shown in the figure. This would allow users to use new, fancy IMS services even when they are roaming in an area that does not have an IMS network but provides IP connectivity. In theory, it is possible to deploy an IMS network in a single area/country and use, say, General Packet Radio Service (GPRS) roaming to connect customers to the home network. In practice, this would not happen because routing efficiency would not be high enough. However, this deployment model is important when operators are ramping up IMS networks or, in an initial phase, when they are offering non- or near real-time multimedia services.

### 2.1.3 Ensuring quality of service for IP multimedia services

On the public Internet, delays tend to be high and variable, packets arrive out of order and some packets are lost or discarded. This will no longer be the case with the IMS. The underlying access and transport networks together with the IMS provide end-to-end Quality of Service (QoS). Via the IMS, the UE negotiates its capabilities and expresses its QoS requirements during a Session Initiation Protocol (SIP) session setup or session modification procedure. The UE is able to negotiate such parameters as:

- Media type, direction of traffic.
- Media type bit rate, packet size, and packet transport frequency.
- Usage of RTP payload for media types.
- Bandwidth adaptation.

After negotiating the parameters at the application level, UEs reserve suitable resources from the access network. When end-to-end QoS is created, the UEs encode and packetize individual media types with an appropriate protocol (e.g., RTP) and send these media packets to the access and transport network by using a transport layer protocol (e.g., TCP or UDP) over IP. It is assumed that operators negotiate service-level agreements for guaranteeing the required QoS in the interconnection backbone. In the case of UMTS, operators could utilize the GPRS Roaming Exchange backbone.

### 2.1.4 IP policy control for ensuring correct usage of media resources

IP policy control means the capability to authorize and control the usage of bearer traffic intended for IMS media, based on the signalling parameters at the IMS session. This requires interaction between the IP connectivity access network and the IMS. The means of setting up interaction can be divided into three different categories [19, 22, 23]:

- The policy control element is able to verify that values negotiated in SIP signalling are used when activating bearers for media traffic. This allows an
operator to verify that its bearer resources are not misused (e.g., the source and destination IP address and bandwidth in the bearer level are exactly the same as used in SIP session establishment).

- The policy control element is able to enforce when media traffic between the end points of a SIP session start or stop. This makes it possible to prevent the use of the bearer until session establishment is completed and allows traffic to start/stop in synchronization with the start/stop of charging for a session in IMS.

- The policy control element is able to receive notifications when the IP connectivity access network service has modified, suspended or released the bearer(s) of a user associated with a session. This allows IMS to release an ongoing session because, for instance, the user is no longer in the coverage area.

### 2.1.5 Secure Communication

Security is a fundamental requirement in every telecommunication system and the IMS is not an exception. The IMS has its own authentication and authorization mechanisms between the UE and the IMS network in addition to access network procedures (e.g., GPRS network). Moreover, the integrity and optional confidentiality of the SIP messages is provided between the UE and the IMS network and between IMS network entities regardless of the underlying core network (e.g., RAN and GPRS). Therefore, the IMS provides at least a similar level of security as the corresponding GPRS and circuit-switched networks: for example, the IMS ensures that users are authenticated before they can start using services, and users are able to request privacy when engaged in a session. An overview of applied security solutions is depicted in Figure 2.2.

![Figure - 2.2: Overview of IMS security](image)

### 2.1.6 Charging arrangements

From an operator or service provider perspective the ability to charge users is a must in any network. The IMS architecture allows different charging models to be used. This includes, say, the capability to charge just the calling party or to charge both the calling party and the called party based on used resources.
in the transport level. In the latter case the calling party could be charged entirely on an IMS-level session: that is, it is possible to use different charging schemes at the transport and IMS level. However, an operator might be interested to correlate charging information generated at transport and IMS (service and content) charging levels. This capability is provided if an operator utilizes a policy control reference point.

As IMS sessions may include multiple media components (e.g., audio and video), it is required that the IMS provides a means for charging per media component. This would allow a possibility to charge the called party if she adds a new media component in a session. It is also required that different IMS networks are able to exchange information on the charging to be applied to a current session [24, 25].

Figure 2.3 shows a simplified view of general charging arrangements in an IMS environment. The key observation is: the IMS adds the possibility to charge for user IP traffic in a more granular manner than before.

The IMS architecture supports both online and offline charging capabilities. Online charging is a charging process in which the charging information can affect in real time the service rendered and, therefore, directly interacts with session/service control. In practice, an operator could check the user’s account before allowing the user to engage a session and to stop a session when all credits are consumed. Prepaid services are applications that need online charging capabilities. Offline charging is a charging process in which the charging information does not affect in real time the service rendered. This is the traditional model in which the charging information is collected over a particular period and, at the end of the period, the operator posts a bill to the customer.

Figure 2.3 shows a simplified view of general charging arrangements in an IMS environment. The key observation is: the IMS adds the possibility to charge for user IP traffic in a more granular manner than before.
2.1.7 Support of roaming

From a user point of view it is important to get access to her services regardless of her geographical location. The roaming feature makes it possible to use services even though the user is not geographically located in the service area of the home network. Section 2.1.2 has already described two instances of roaming: namely, GPRS roaming and IMS roaming. In addition to these two there exists an IMS circuit switched roaming case. GPRS roaming means the capability to access the IMS when the visited network provides the RAN and SGSN and the home network provides the GGSN and IMS. The IMS roaming model refers to a network configuration in which the visited network provides IP connectivity (e.g., RAN, SGSN, GGSN) and the IMS entry point (i.e., P-CSCF) and the home network provides the remaining IMS functionalities. The main benefit of this roaming model compared with the GPRS roaming model is optimum usage of user-plane resources. Roaming between the IMS and the CS CN domain refers to inter-domain roaming between IMS and CS. When a user is not registered or reachable in one domain a session can be routed to the other domain. It is important to note that both the CS CN domain and the IMS domain have their own services and cannot be used from another domain. Some services are similar and available in both domains (e.g., Voice over IP in IMS and speech telephony in CS CN). Figure 2.4 shows different IMS/CS roaming cases.

2.1.8 Interworking with other networks

It is evident that the IMS is not deployed over the world at the same time. Moreover, people may not be able to switch terminals or subscriptions very rapidly. This will raise the issue of being able to reach people regardless of what kind of terminals they have or where they live. To be a new, successful communication network technology and architecture the IMS has to be able to
connect to as many users as possible. Therefore, the IMS supports communication with PSTN, ISDN, mobile and Internet users. Additionally, it will be possible to support sessions with Internet applications that have been developed outside the 3GPP community [23].

2.1.9 Service control model

In 2G mobile networks the visited service control is in use. This means that, when a user is roaming, an entity in the visited network provides services and controls the traffic for the user. This entity in the second generation (2G) is called a visited mobile service-switching centre. In the early days of Release 5 both visited and home service control models were supported. Supporting two models would have required that every problem have more than one solution; moreover, it would reduce the number of optimal architecture solutions, as simple solutions may not fit both models. Supporting both models would have meant additional extensions for Internet Engineering Task Force (IETF) protocols and increased the work involved in registration and session flows. The visited service control was dropped because it was a complex solution and did not provide any noticeable added value compared with the home service control.

On the contrary, the visited service control imposes some limitations. It requires a multiple relationship and roaming models between operators. Service development is slower as both the visited and home network would need to support similar services, otherwise roaming users would experience service degradations. In addition, the number of interoperator reference points increase, which requires complicated solutions (e.g., in terms of security and charging). Therefore, home service control was selected; this means that the entity that has access to the subscriber database and interacts directly with service platforms is always located at the user’s home network.

2.1.10 Service development

The importance of having a scalable service platform and the possibility to launch new services rapidly has meant that the old way of standardizing complete sets of teleservices, applications and supplementary services is no longer acceptable. Therefore, 3GPP is standardizing service capabilities and not the services themselves [24]. The IMS architecture should actually include a service framework that provides the necessary capabilities to support speech, video, multimedia, messaging, file sharing, and data transfer, gaming and basic supplementary services within the IMS.
2.1.11 Layered design

3GPP has decided to use a layered approach to architectural design. This means that transport and bearer services are separated from the IMS signalling network and session management services. Further services are run on top of the IMS signalling network. Figure 2.5 shows the design. In some cases it may be impossible to distinguish between functionality at the upper and lower layers. The layered approach aims at a minimum dependence between layers. A benefit is that it facilitates the addition of new access networks to the system later on. Wireless Local Area Network (WLAN) access to the IMS was added in 3GPP Release 6 and fixed broadband access to the IMS is being standardized in Release 7. The layered approach increases the importance of the application layer as services are designed to work independent of the access network and the IMS is equipped to bridge the gap between them. Whether the subscriber is using a mobile phone or a PC client to communicate, the same presence and group list functions in IMS will be used. Different services have different requirements. These include:

- Bandwidth
- Latency
- Processing power in the device

This means that in order for different services to be executed properly, the network has to be equipped with access-aware control and service logic for multimedia services. Multi-access functionality is built into the IMS architecture, which offers a way for fixed and mobile operators to deliver a fixed to mobile convergence solution. This will enable service providers to use the characteristics and capabilities of the currently selected device and its network access method and adapt to it dynamically.
2.1.12 Access independence

The IMS was originally designed to be access-independent so that IMS services can be provided over any IP connectivity networks (e.g., GPRS, WLAN, broadband access x-Digital Subscriber Line). Unfortunately, Release 5 IMS specifications contain some GPRS-specific features. In Release 6 (e.g., GPRS) access-specific issues are separated from the core IMS description and the IMS architecture returned to its born state (i.e., access independent). Figure 2.6 demonstrates the different types of access independent networks that the IMS can run on. These include Fixed Broadband, WLAN, GPRS and UMTS.

2.2 Description of IMS-related entities and functionalities

The IMS architecture defines the logical elements necessary to implement next generation multimedia services across multiple network types. It is important to note that these logical functions do not necessarily have a one-to-one relationship with physical equipment. The components of the IMS architecture refer to functions, not platforms. Multiple functions can be mapped to a single network device, and, conversely, a single function can conceivably be implemented across multiple physical platforms.

This section discusses IMS entities and key functionalities. These entities can be roughly classified in six main categories:

- Session management and routing family (CSCFs)
- Databases (HSS, SLF)
- Services (application server, MRFC, MRFP)
- Interworking functions (BGCF, MGCF, IMS-MGW, SGW)
- Support functions (PDF, SEG, THIG)
- Charging

![Figure - 2.6: Access-independent IMS](image-url)
It is important to understand that IMS specifications are set up so that the internal functionality of the network entities is not specified in detail. Instead, specifications describe reference points between entities and functionalities supported at the reference points. For instance, how does CSCF obtain user data from databases? Different reference points will be described in Section 2.3. Additionally, GPRS functions are described at the end of this section.

2.2.1 Call Session Control Functions (CSCF)

There are three different kind of Call Session Control Functions (CSCF): Proxy-CSCF (P-CSCF), Serving-CSCF (S-CSCF) and Interrogating-CSCF (I-CSCF). Each CSCF has its own special tasks and these tasks are described in the following subsections. Common to all CSCFs is that they all play a role during registration and session establishment and form the SIP routing machinery. Moreover, all functions are able to send charging data to an offline charging function. There are some common functions that P-CSCF and S-CSCF are able to perform. Both entities are able to release sessions on behalf of the user (e.g., when S-CSCF detects a hanging session or P-CSCF receives a notification that a media bearer is lost) and are able to check the content of the Session Description Protocol (SDP) payload and to check whether it contains media types or codecs, which are not allowed for a user. When the proposed SDP does not fit the operator’s policy, the CSCF rejects the request and sends a SIP error message to the UE.

2.2.1.1 Proxy Call Session Control Function (P-CSCF)

Proxy Call Session Control Function (P-CSCF) is the first contact point for users within the IMS. It means that all SIP signalling traffic from the UE will be sent to the P-CSCF. Similarly, all terminating SIP signalling from the network is sent from the P-CSCF to the UE. There are four unique tasks assigned for the P-CSCF: SIP compression, IPSec security association, interaction with Policy Decision Function (PDF) and emergency session detection.

As the SIP protocol is a text-based signalling protocol, it contains a large number of headers and header parameters, including extensions and security-related information, which means that their message sizes are larger than with binary-encoded protocols. For speeding up the session establishment 3GPP has mandated the support of SIP compression between the UE and P-CSCF. The P-CSCF needs to compress messages if the UE has indicated that it wants to receive signalling messages compressed.

P-CSCF is responsible for maintaining Security Associations (SAs) and applying integrity and confidential protection for SIP signalling. This is achieved during SIP registration as the UE and P-CSCF negotiate IPSec SAs. After the initial registration the P-CSCF is able to apply integrity and confidential protection of SIP signalling. The P-CSCF is tasked to relay session and media-related information to the PDF when an operator wants to apply SBLP. Based on the received information the PDF is able to derive authorized IP QoS information that will be passed to the GGSN when the GGSN needs to perform Service-Based Local Policy prior to accepting a secondary PDP context activation.
IMS emergency sessions are not yet fully specified (work ongoing in Release 7), therefore it is important that the IMS network detects emergency session attempts and guides a UMTS UE to use the CS network for emergency sessions. Such detection is a task of the P-CSCF. This functionality will not vanish when IMS emergency sessions are supported, as in certain roaming cases it is possible that the UE does not itself recognize that the user has dialed an emergency number. The planned function in Release 7 is that the P-CSCF is able to select an emergency CSCF to handle an emergency session. Such a selection is needed because in IMS roaming cases the assigned S-CSCF is in the home network and the home S-CSCF is unable to route the request to the correct emergency centre.

2.2.1.2 Interrogating Call Session Control Function (I-CSCF)

Interrogating Call Session Control Function (I-CSCF) is a contact point within an operator’s network for all connections destined to a subscriber of that network operator. There are four unique tasks assigned for the I-CSCF:

- Obtaining the name of the next hop (either S-CSCF or application server) from the Home Subscriber Server (HSS).
- Assigning an S-CSCF based on received capabilities from the HSS. The assignment of the S-CSCF will take place when a user is registering with the network or a user receives a SIP request while she is unregistered from the network but has services related to an unregistered state (e.g., voice mail).

2.2.1.3 Serving Call Session Control Function (S-CSCF)

Serving Call Session Control Function (S-CSCF) is the focal point of the IMS as it is responsible for handling registration processes, making routing decisions and maintaining session states, and storing the service profile(s). When a user sends a registration request it will be routed to the S-CSCF, which downloads authentication data from the HSS. Based on the authentication data it generates a challenge to the UE. After receiving the response and verifying it the S-CSCF accepts the registration and starts supervising the registration status. After this procedure the user is able to initiate and receive IMS services. Moreover, the S-CSCF downloads a service profile from the HSS as part of the registration process.

A service profile is a collection of user-specific information that is permanently stored in the HSS. The S-CSCF downloads the service profile associated with a particular public user identity (e.g., joe.doe@ims.example.com) when this particular public user identity is registered in the IMS. The S-CSCF uses information included in the service profile to decide when and, in particular, which application server(s) is contacted when a user sends a SIP request or receives a request from somebody. Moreover, the service profile may contain further instructions about what kind of media policy the S-CSCF needs to apply – for example, it may indicate that a user is only allowed to use audio and application media components but not video media components.
The S-CSCF is responsible for key routing decisions as it receives all UE-originated and UE-terminated sessions and transactions. When the S-CSCF receives a UE originating request via the P-CSCF it needs to decide if application servers are contacted prior to sending the request further on. After possible application server(s) interaction the S-CSCF either continues a session in IMS or breaks to other domains (CS or another IP network). Moreover, if the UE uses a Mobile Station ISDN (MSISDN) number to address a called party then the S-CSCF converts the MSISDN number (i.e., a tel URL) to SIP Universal Resource Identifier (URI) format prior to sending the request further, as the IMS does not route requests based on MSISDN numbers. Similarly, the S-CSCF receives all requests, which will be terminated at the UE. Although, the S-CSCF knows the IP address of the UE from the registration it routes all requests via the P-CSCF, as the P-CSCF takes care of SIP compression and security functions. Prior to sending a request to the P-CSCF, the S-CSCF may route the request to an application server(s), for instance, checking possible redirection instructions.

In addition, the S-CSCF is able to send accounting-related information to the Online Charging System for online charging purposes (i.e., supporting prepaid subscribers).

![Figure - 2.7: S-CSCF routing and basic IMS session setup](image)

### 2.2.2 Application Servers

Application servers essentially host and execute services for users and perform the function of SIP Application Servers. In other words, depending on the actual service, application server can operate in of the following modes:

- **SIP Proxy mode**
- **SIP User Agent mode**
• SIP Redirect Server mode
• SIP B2BUA (Back to Back User Agent - concatenation of two user agents)
• Application servers also interface with the HSS to upload and download user data.

2.2.2.1 Gateways
Several types of gateways are supported in the IMS architecture, for example the architecture includes gateways for converting signal from packet switched IMS network to a circuit switched PSTN or vice versa

 o Signaling gateways performs lower layer protocol conversion from one network to another.

 o Gateways to convert the media data - Media Gateway (MG) and Media Gateway Controller Function (MGCF). The MG interfaces the media planes of two networks. Hence, it converts the media over RTP (in IMS network) to the PCM (Pulse Code Modulation) based transport in the PSTN side.

2.2.3 BGCF (Breakout Gateway Controller Functions)
The BGCF is also a SIP server that performs routing functions when the call is addressed to a circuit switched network such as PSTN. It locates the appropriate gateway at the circuit switched destination network for routing the outgoing call.

The IMS will have to coexist with the circuit-switched domain of the network operator for some time. For example, GSM will have to exist with the IMS. This is not unusual since the EV-DO network of cdma2000 has to operate with the cdmaOne network that is now in place, which is the task faced by MNO-B Wireless. MNO-A Wireless needs to operate traditional GSM with the High-speed Packet Access services that are rolling out as part of UMTS. Clearly the IMS and the PSTN will also have to coexist.

To move from the IMS to the circuit-switched domain, the IMS uses the breakout gateway control function (BGCF), which determines where the breakout to the circuit-switched domain occurs. The outcome of the selection is not obvious at first since the breakout can be in the same network or to another network.

For example, MNO-A Wireless will implement IMS within the structure of its current GSM and this represents one network from the IMS perspective. Thus, an IMS MNO-A Wireless user could talk to a GSM MNO-A Wireless user and the BGCF would select the network in which it currently resides. In this case, the BGCF retains control and selects a media gateway control function (MGCF) to handle the session further. Suppose that the IMS MNO-A Wireless user needs to talk to a MNO-B Wireless user that is operating in the circuit-switched domain and not the IMS domain of the MNO-B Wireless network. Now the BGCF selects a breakout that is to another network. In this
case the BGCF hands control of the session to the BGCF in the selected network.

Now the BGCF in the selected network chooses the MGCF that will handle the session since it selects a circuit-switched domain that is part of its network. This scenario assumes that the selected network has the IMS as part of its network. If the IMS is not in place, then the handoff goes directly through the media gateway to the circuit-switched domain. In this case, some functions of the IMS could be lost at the gateway.

2.2.4 Media Resource Function (MRF)

An MRF performs several media functions for the home SIP network; such as mix media streams (in a conference bridge), transcoding functions, playing announcements. An MRF can be further broken into MRF Controller (MRFC) and MRF Processor (MRFP). The MRFC acts essentially as a SIP UA interfacing with the S-CSCF) to manage resources of the MRFP, while the MRFP performs all the media functions stated above.

The media gateway control function (MGCF) and the IMS media gateway function (IMS-MGW) are the Controller/processor Bridge between the IMS and circuit-switched domain. This visual examines the role of the MGCF.

Simply put, the MGCF enables the interworking between the IMS user and a user in the circuit-switched domain. On an incoming call from a user in the circuit-switched domain, the ISDN User Part (ISUP) or Bearer Independent Call Control (BICC) signals are converted to the appropriate SIP protocols and then the session is forwarded to the IMS. For a call from the IMS to a user in the circuit-switched domain, the reverse is followed. The MGCF also controls the sessions on the user-plane level via the IMS-MGW. As other entities, the MGCF can forward accounting information to the CCF and OCS.

2.2.5 Reference points

Performing functions in IMS network is realized through procedures, which define the flows between functional components. The interfaces exposed by the functional components and the control between the components is referred to as "reference points".

The following is the description of the reference points for the IP Multimedia Core Network Subsystem.

- **Cx** - Supports information transfer between CSCF and HSS
- **Dx** - The CSCF and SLF interface is used to retrieve the address of the HSS, which holds the subscription date for a given user. Not required in a single HSS environment
- **Gm** - Supports communication between the UE and a P-CSCF
- **ISC** - The interface between the CSCF and application servers for access to IMS services
- **Ma** - The interface between an application server and an I-CSCF
• **Mb** - Used to access IPv6 network services for user data transport

• **Mg** - Allows the MGCF to forward incoming session signalling (from the PSTN) to the CSCF for the purpose of interworking with PSTN networks. Uses SIP for signalling

• **Mi** - Allows the Serving CSCF to forward the session signalling to the BGCF for the purpose of interworking with PSTN networks. Uses SIP for signalling

• **Mj** - Allows the BGCF to forward session signalling to the MGCF for the purpose of interworking with PSTN networks. Uses SIP for signalling

• **Mk** - Allows the BGCF to forward session signalling to another BGCF. Uses SIP for signalling

• **Mm** - The interface between a CSCF/BGCF/IMS ALG and an IP multimedia network

• **Mr** - Supports information transfer between CSCF and MRFC. Uses SIP for signalling

• **Mw** - Allows the communication and forwarding of SIP signalling messaging between CSCFs

• **Mx** - The interface between a CSCF/BGCF and IBCF

• **Sh** - Used for communication from the SIP or OSA application server to the HSS

• **Si** - Used for communication from the CAMEL application server to the HSS

• **Ut** - The Ut interface resides between the UE and the SIP Application Server

### 2.3 IMS functional planes

The 3GPP architecture consists of logical planes or layers, which correspond to discrete functions. This is one of the most powerful concepts of the IMS functional architecture. Each plane consists of IMS functional components that together provide the functions supported by the plane. The logical functions in IMS are divided into the following three planes:

1. Transport plane
2. Control plane
3. Service plane
2.3.1 Transport plane

The transport plane provides support for the backbone IMS network and for the different means through which users can gain access to the IMS network. Included in this plane are IMS components such as routers, media gateways and switches as well as IMS user equipment devices. These components translate protocols between the IMS core network and the connecting network. The transport plane also shields the upper layers of the IMS architecture from the network access technologies by providing common access interface to the components in this plane.

2.3.2 Control plane

The primary function of the Control plane is to provide switching and session control in IMS networks. The key components in this plane are the SIP servers and proxies collectively called Call/Session Control Function (CSCF). CSCF handles SIP registrations and routing of the SIP signaling messages to appropriate application servers amongst the other control and signaling functions that it performs. CSCF also provides policy control and QoS management.

The other components in this plane include:
• **Home Subscriber Server (HSS):** The repository for users service profile. The user information is also used to provide authentication, authorization and accounting (AAA) functions.

• **Media Gateway Control Function (MGCF):** MGCF interworks SIP signaling with the signaling used by the media gateways, and manages the connections between the PSTN and the IP streams. It converts SIP messages into either Megaco or ISUP messages.

• **Media Server Function Control (MSFC):** The MSFC provides a similar function as the MGCF for media servers

### 2.3.3 Service plane
Residing in the service plane are application servers that perform telephony and non-telephony functions. Application servers interface with Call Session Control Function in the control plane using SIP, and operate in SIP proxy, SIP User Agent Server (UAS) or SIP B2BUA (back-to-back user agent) mode. An AS can be located in the home network or in an external third-party network. Types of application servers include:

- **SIP AS** - application servers that interfaces using SIP
- **OSA-SCS** (Open Service Access - Service Capability Server) – interfaces with Open Services Architecture (OSA) application servers using Parlay
- **IM-SSF (IP Multimedia Service Switching Function)** - CAMEL application servers, interfaces using CAMEL Application Part (CAP)

### 2.4 Databases
There are two main databases in the IMS architecture: Home Subscriber Server (HSS) and Subscription Locator Function (SLF). The HSS is the main data storage for all subscriber and service-related data of the IMS. The main data stored in the HSS include user identities, registration information, access parameters and service-triggering information [26]. User identities consist of two types: private and public user identities.

The private user identity is a user identity that is assigned by the home network operator and is used for such purposes as registration and authorization, while the public user identity is the identity that other users can use for requesting communication with the end-user. IMS access parameters are used to set up sessions and include parameters like user authentication, roaming authorization and allocated S-CSCF names. Service-triggering information enables SIP service execution.

The HSS also provides user-specific requirements for S-CSCF capabilities. This information is used by the I-CSCF to select the most suitable S-CSCF for a user.
In addition to functions related to IMS functionality, the HSS contains the subset of Home Location Register and Authentication Center (HLR/AUC) functionality required by the Packet-Switched (PS) domain and the Circuit-Switched (CS) domain. The structure of the HSS is shown in Figure 2.9. Communication between different HSS functions is not standardized. HLR functionality is required to provide support to PS domain entities, such as SGSN and GGSN. This enables subscriber access to PS domain services. In similar fashion the HLR provides support for CS domain entities, like MSC/MSC servers. This enables subscriber access to CS domain services and supports roaming to Global System for Mobile Communications (GSM)/UMTS CS domain networks. The AUC stores a secret key for each mobile subscriber, which is used to generate dynamic security data for each mobile subscriber. Data are used for mutual authentication of the International Mobile Subscriber Identity (IMSI) and the network. Security data are also used to provide integrity protection and ciphering of the communication over the radio path between the UE and the network. There may be more than one HSS in a home network, depending on the number of mobile subscribers, the capacity of the equipment and the organization of the network. There are multiple reference points between the HSS and other network entities. The SLF is used as a resolution mechanism that enables the I-CSCF, the S-CSCF and the AS to find the address of the HSS that holds the subscriber data for a given user identity when multiple and separately addressable HSSs have been deployed by the network operator.

### 2.5 Service functions

Three functions are categorized as IMS service-related functions – namely, Multimedia Resource Function Controller (MRFC), Multimedia Resource Function Processor (MRFP) and Application Server (AS). Keeping in mind the layered design, ASs are not pure IMS entities; rather, they are functions on top of IMS. However, ASs are described here as part of IMS functions because ASs are entities that provide value-added multimedia services in the IMS, such as presence and Push to talk Over Cellular. An AS resides in the user’s home network or in a third-party location. The third party here means a network or a standalone AS.

The main functions of the AS are:

- The possibility to process and impact an incoming SIP session received from the IMS.
- The capability to originate SIP requests.
- The capability to send accounting information to the charging functions.

The services offered are not limited purely to SIP-based services since an operator is able to offer access to services based on the Customized Applications for Mobile network Enhanced Logic (CAMEL) Service Environment (CSE) and the Open Service Architecture (OSA) for its IMS subscribers [11]. Therefore, AS is the term used generically to capture the behaviour of the SIP AS, OSA Service Capability Server (SCS) and CAMEL IP Multimedia Service Switching Function (IM-SSF). Using the OSA an operator may utilize such service capability features as call control, user interaction, user status, data session control, terminal capabilities, account management, charging and policy management for developing services [27].

Figure - 2.10: Relationship between different application server types.

An additional benefit of the OSA framework is that it can be used as a standardized mechanism for providing third-party ASs in a secure manner to the IMS, as the OSA itself contains initial access, authentication, authorization, registration and discovery features (the S-CSCF does not provide authentication and security functionality for secure direct third-party access to the IMS). As the support of OSA services is down to operator choice, it is not architecturally sound to support OSA protocols and features in multiple entities. Therefore, OSA SCS is used to terminate SIP signalling from the S-CSCF. The OSA SCS uses an OSA Application Program Interface (API) to communicate with an actual OSA application server. The IM-SSF was introduced in the IMS architecture to support legacy services that are developed in the CSE. It hosts CAMEL network features (trigger detection points, CAMEL Service Switching Finite State Machine, etc.) and interworks with the CAMEL Application Part (CAP) interface.

A SIP AS is a SIP-based server that hosts a wide range of value-added multimedia services. A SIP AS could be used to provide presence, messaging, Push to talk Over Cellular and conferencing services. Figure 2.9 shows how different functions are connected. From the perspective of the S-
CSCF SIP AS, the OSA service capability server and the IM-SSF exhibit the same reference point behavior.

An AS may be dedicated to a single service and a user may have more than one service, therefore there may be one or more ASs per subscriber. Additionally, there may be one or more ASs involved in a single session. For example, an operator could have one AS to control terminating traffic to a user based on user preferences (e.g., redirecting all incoming multimedia sessions to an answer machine between 5 p.m. and 7 a.m.) and another AS to adapt the content of instant messages according to the capabilities of the UE (screen size, number of colours, etc.). MRFC and MRFP together provide mechanisms for bearer-related services such as conferencing, announcements to a user or bearer transcoding in the IMS architecture. The MRFC is tasked to handle SIP communication to and from the S-CSCF and to control the MRFP. The MRFP in turn provides user-plane resources that are requested and instructed by the MRFC. The MRFP performs the following functions:

- Mixing of incoming media streams (e.g., for multiple parties).
- Media stream source (for multimedia announcements).
- Media stream processing (e.g., audio transcoding, media analysis) [23], [26].

Currently, the role of MRFC and MRFP in the IMS architecture is minor, as in IMS conferencing work [28] the MRFC is co-located with an AS and the reference point between the MRFC and MRFP is not yet well defined.

### 2.5.1 Interworking functions

This section introduces four interworking functions, which are needed for exchanging signalling and media between IMS and the CS CN. For breaking out the S-CSCF sends a SIP session request to the Breakout Gateway Control Function (BGCF); it further chooses where a breakout to the CS domain occurs. The outcome of a selection process can be either a breakout in the same network in which the BGCF is located or another network. If the breakout happens in the same network, then the BGCF selects a Media Gateway Control Function (MGCF) to handle the session further. If the breakout takes place in another network, then the BGCF forwards the session to another BGCF in a selected network [23]. The latter option allows routing of signalling and media over IP near to the called user. When a SIP session request hits the MGCF it performs protocol conversion between SIP protocols and the ISDN User Part (ISUP), or the Bearer Independent Call Control (BICC) and sends a converted request via the Signalling Gateway (SGW) to the CS CN.
The SGW performs signalling conversion (both ways) at the transport level between the IP-based transport of signalling (i.e., between Sigtran SCTP/IP and SS7 MTP) and the Signalling System No. 7 (SS7) based transport of signalling. The SGW does not interpret application layer (e.g., BICC, ISUP) messages, as is shown in Figure 2.10. The MGCF also controls the IMS Media Gateway (IMS-MGW). The IMS-MGW provides the user-plane link between CS CN networks and the IMS. It terminates the bearer channels from the CS network and media streams from the backbone network (e.g., RTP streams in an IP network or AAL2/ATM connections in an ATM backbone), executes the conversion between these terminations and performs transcoding and signal processing for the user plane when needed. In addition, the IMS-MGW is able to provide tones and announcements to CS users. Similarly, all incoming call control signalling from a CS user to an IMS user is destined to the MGCF that performs the necessary protocol conversion and sends a SIP session request to the I-CSCF for a session termination.

2.6 IMS Related Technologies

This section discusses architectures and frameworks relevant technology, which support IMS for Enhancing Service Provision on communication network.

2.6.1 Service Oriented Architecture

In its working group note on WS architecture, the World Wide Web Consortium (W3C) refers to Service Oriented Architecture (SOA) as a form of distributed systems architecture characterized by its logical view, message orientation, description orientation, granularity, network orientation and platform neutrality. From their description summaries a service in the SOA context as an entity that is:

- An abstract logical view of a process defined by what it does,
- Formally defined in terms of platform-neutral message exchange between its provider and requester agents,
- Described by machine processable meta data
- Typically carrying out few operations but with large complex messages.
From this description, this architecture is clearly not appropriate for all distributed applications. Here is a list of qualities that might indicate that an application is appropriate for SOA, as interpreted from the above-mentioned W3C note. Such an application:

- Operates over the Internet (typically no QoS guarantees);
- Does not have managed deployment mechanisms (group upgrades of requesters and providers);
- Has system components that are to some extent vendor and platform neutral;
- Already exists and needs to be exposed for use over a network (can wrap as a WS).

![Figure - 2.12: Relationship between the IMS & Enabling Technologies](image)

**2.6.2 User Oriented Architecture**

User Oriented Architecture (UOA) is not yet widely mentioned in the IT world. UOA can be described as user-centric SOA. It can be argued that IMS is a UOA. Due to the following facts:

- SIP is the central IMS protocol and SIP in itself has several user-centric features:
- SIP is designed to put intelligence in the end-nodes and even with an IMS between users, this inherent quality is still there.
- A SIP URI identifies a person rather than a device (unlike, for example telephone numbers)
- The SIP event notification [27] extension is (PUBLISH, SUBSCRIBE, NOTIFY) coupled with the growing collection of standardized SIP event packages, give users the power to create and publish their own service
events as well as to monitor events. Presence is the most well-known event-notification scenario but it is far from being the only one.

- IMS service chaining is steered by preferences set in the individual user's service profile.

Webalo [29] provides a UOA product called a User Proxy. Webalo claims that the User Proxy "achieves loose-coupling between users and systems". According to the information on their web site, the proxy is essentially a WS that other services utilize to interact with the user. Applications are dynamically customisable and appear to be tailor-made for the client to which they are downloaded.

2.6.3 Event Driven Architecture

Event Drive Architecture (EDA) are loosely coupled and distributed by nature and are therefore best suited to asynchronous systems. There is one style of EDA can be described as a style of SOA in the sense that either an event occurrence can trigger one or more services in a SOA environment, or, a service in a SOA can trigger the occurrence of one or more events [28]. However EDA covers a much broader scope of architectures than just event driven SOA.

In this context, it is define an event as a significant change of state [29]. In order for the event model to work well, the normal boundaries of state behavior must be well defined so that significant changes of state can be identified (expectation $\rightarrow$ deviation $\rightarrow$ response). Boundaries should be explicit, such as upper and lower time thresholds or a certain percentage of CPU time being idle. The event chain consists of an event source, an event channel, an event processing unit and an event driven activity.

SIP is a good candidate for implementation on EDA platforms because it is itself an event-oriented protocol with clear states. For example, SIP defines four different transactions: INVITE Client transactions, INVITE Server transactions, Non-INVITE Client transactions and Non-INVITE Server transactions. Further, depending on the combination of method type (INVITE or non-INVITE) and peer role (client or server), each transaction may be in or transition to one of six explicit states at any given time, namely: calling, trying, proceeding, completed, confirmed or terminated [24].

Figure 2.12 depicts a simplified state machine for the INVITE client transaction. State transitions are labeled with the events that trigger them. Triggers may be requests received (INVITE in this case), responses received (three digit status codes such as 200 which means "OK" or 180 which means "ringing"), an error signal or a timer signal. The action taken with each trigger is not shown in this diagram. In this case the event source is the SIP user agent or SIP stateful proxy; the event channel is the transaction itself and the event driven activity is the execution of the SIP method. For the sake of completeness can also mention that SIP also defines explicit states for dialogs.
Another example of a framework that implements an EDA is a Service Level Execution Environment (SLEE).

![Diagram of SIP INVITE Client Transaction - state machine](image)

**Figure - 2.13: SIP INVITE Client Transaction - state machine**

### 2.6.4 Web Services

SOA can be implemented without the use of WS and WS can be used without implementing SOA; yet WS is a technology typically used to implement SOA. There is no one WS specification, there are several, from a variation of commercial and standardization organizations. As there is no one specification, neither is their one authoritative definition. Use WSC's technical report on WS architecture [29] as a guideline. Supporting specifications from W3C and OASIS are referenced as needed. The Web Services Interoperability Organization (WS-I) provides what they call profiles which are a set of specifications aimed at best practices for WS interoperability. Figure 2.14 gives an overview of typical Web Services interactions.

Based on the various descriptions and implementations of WS here is a summary of protocol-agnostic requirements:

- A web service must be capable of interacting and coordinating with other services as well as with service consumers
- It must be discoverable and consumable over a network. This implies that:
• It must expose an interface to potential service consumers.
• The exposed interface must describe the service in terms of its capabilities and information needed to communicate with it.
• The service description must be machine-processable.

A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols.

According to common practice in WS development can say that WS is partially defined by the protocols and interfaces typically used in implementing these requirements.

### 2.6.4.1 Extensible Markup Language (XML)

XML is an open and extensible data object format that is the basis of several of the WS protocols. W3C standardizes it. One of XML’s main attractions is the ability it provides for using the same data structure across heterogeneous systems. This makes it particularly attractive for use across the Internet. In the WS context it is used to serialize data. However, the specifications and standards that describe XML in its entirety are vast, complex and ever growing. Its important features as utilised in WS are:

- **XML Namespaces (xmlns):** URI References that globally and uniquely qualify element and attribute names that might otherwise be ambiguous. For example `xmlns:env= “http://www.w3.org/2003/05/soap-envelope”` defines WSC's XML names pace for soap envelopes

- **XML Infoset:** a specification that provides an abstract description of different parts of a well-formed XML document for reference in other XML
specifications. Where a *well-formed* XML document is defined as one that has correct XML syntax.

- **XML Schema Definition (XSD):** A set of conformance rules to be applied for an XML document to be considered valid, where a *valid* XML document is defined as a well-formed XML document, which also conforms to an XML schema. For the sake of clarity, note that "XML Schema" is the name of WSC's XML schema and XSD is an instance of "XML Schema".

Figure - 2.15: Typical Web Services Protocol Stack

### 2.6.4.2 Web Services Description Language (WSDL)

WSDL is an XML language that is used to model and describe web services [30]. WSDL describes a service abstractly in terms of its message exchange patterns as well as concretely by specifying the protocol (e.g., SOAP) and the address (e.g., a URI) of the service port (in other words by specifying its binding information). It is important to note that WSDL 2.0 accepts binding to all HTTP request methods. WSDL 1.1 only accepted binding to the POST and GET methods.

In practical terms, WSDL allows a service to advertise its capabilities to potential clients. The client uses the information presented in the exposed WS interface to bind to, consume and interact with the service.
2.6.4.3 Universal Description Discovery and Integration (UDDI)

It is fully possible to have a functional WS architecture without the implementation of a WS discovery service. However, for anything other than small private networks with few services it is not practical. Without a discovery service, clients must have extensive knowledge of all available services in order to find, bind to and consume them. This would create an unacceptable overhead for a telecom system that wants access to a wide variety of services over the Internet so it is dismissed as infeasible.

The WS architecture does not specify how a WS discovery service obtains or handles service descriptions. However, UDDI is one popular option. It is standardised by OASIS as "a 'meta service' for locating web services by enabling robust queries against rich metadata" [31]. The current specification is UDDI 3.0 and it is designed to provide for building "flexible, interoperable XML Web services registries". XML schema is used to describe the UDDI data structures. The discovery process can be done at design-time or at runtime and must and can be autonomous or manual. These are important differences to consider in WS system design.

2.6.4.4 SOAP

SOAP is a protocol designed to facilitate the exchange of structured messages in a distributed network. W3C's SOAP 1.2 specification defines an XML-based messaging framework as well as three optional components, namely: (1) encoding rules for expressing application-defined data types, (2) SOAP's remote procedure calls (RFC) conventions, and (3) SOAP's HTTP/1.1 conventions.

SOAP messages can be sent within or on top of a variety of network protocols such as HTTP, SMTP, FTP, RMI, or even proprietary protocols. HTTP is the most widely used and the only one will refer to further discussion. The specification provides general rules for binding SOAP to different protocols. While other distributed messaging technologies such as the Distributed Component Object Model (DCOM) and the General Inter-Orb Protocol (IOP) are generally filtered by firewalls, SOAP tunnels through firewalls without problems. This presents a significant security risk for IMS Service Providers who want to open up their platforms to WS orchestration.

In addition the basic SOAP specification provides no security mechanisms. In other words, access control, confidentiality, integrity and non-repudiation are not addressed. Security in SOAP implementations can be added optionally as WS extensions.

A SOAP message is defined as an XML infoset, which consists of a mandatory envelope item, which is the message container, and within the envelope, a mandatory body item, which the actual message being transmitted, and an optional header item, which contains application specific information.
2.6.4.5 Web Services Invocation Framework

The Web Services Invocation Framework is a Java framework for invoking WSDL-described services regardless of the protocol being used for message exchange. This means with use of WSIF, one can invoke web services without the use of SOAP.

2.6.4.6 Representation State Transfer (REST)

Agents identify objects in the system, called resources, with Uniform Resource Identifiers (URI). Agents represent, describe, and communicate resource state via representations of the resource in a variety of widely-understood data formats (e.g., XML, HTML, CSS, JPEG, PNG). Agents exchange representations via protocols that use URIs to identify and directly or indirectly address the agents and resources.

REST is an architectural style for reliable Web applications first described by Fielding. According to the thesis, Fielding’s intention was to present a model of how the modern web should work. It provides principles which outline how resources on the web are defined and addressed. Further, application of REST architecture on the Web is meant to improve interaction scalability, reduction of interaction latency, and generality of interfaces, independent component deployment, security enforcement and legacy support.

REST web agents provide uniform interface semantics (create, retrieve, update, delete) rather than arbitrary or application-specific interfaces, and manipulate resources by exchanging representations. The messages do not depend on the stored state on the server, so the interactions are said to be stateless.

W3C identifies REST-compliant Web services as those whose primary purpose is to manipulate XML representations of Web resources using a uniform set of "stateless" operations as opposed to arbitrary Web services, that are identified as which may expose an arbitrary set of operations. SOAP 1.2 can be used in a REST compliant manner, however SOAP adds a layer to HTTP whereas REST in itself does not. This leads us to following criticism of SOAP. W3C has been criticised for promoting SOAP as a WS standard and for not trying to promote REST.

A frequent criticism of the SOAP standard is the verbosity of its messages. As a result RESTful is often recommended by SOAP critics as a way to minimize the messaging overhead. An example of the difference is shown in table 2.1: the same WS request for a Flickr [32] service called flickr.test.echo is first implemented in a RESTful fashion and then the same request is implemented using SOAP. One interesting observation is that while REST architecture stresses uniform interfaces SOA stresses varied described interfaces.

There are several indications from Marketplace leaders that suggest that REST is gaining ground over SOAP in large WS implementations. Here is a list of some of those indications:
Table – 2.1 Code compare for SOAP vs. REST

<table>
<thead>
<tr>
<th>Request service &quot;flickr.test.echo&quot; with SOAP:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;s:Envelope</code></td>
</tr>
<tr>
<td><code>xmlns:s=&quot;http://www.w3.org/2009/05/soap-envelope&quot;</code></td>
</tr>
<tr>
<td><code>xmlns:xsi=&quot;http://www.w3.org/1999/XMLSchema-instance&quot;</code></td>
</tr>
<tr>
<td><code>xmlns:xsd=&quot;http://www.w3.org/1999/XMLSchema&quot;</code></td>
</tr>
<tr>
<td><code>&lt;s:Body&gt;</code></td>
</tr>
<tr>
<td><code>&lt;x:Flickr Request xmlns:x=&quot;urn:flickr&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;method&gt;flickr.test.echo&lt;/method&gt;</code></td>
</tr>
<tr>
<td><code>&lt;name&gt;</code>value<code>&lt;/name&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/x:FlickrRequest&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/s:Body&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/s:Envelope&gt;</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Request &quot;flickr.test.echo&quot; with REST:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>http://api.flickr.com/services/rest/?method=flickr.test.echo&amp;name=value</code></td>
</tr>
</tbody>
</table>

2.6.4.7 OSA Parlay X

Parlay X [33] is set of telecom Web Services offered by the Open Services Architecture (OSA) group. The OSA SCS is one of the two optional application servers supported in IMS. Through this server, IMS gains limited support for Web Services. So, though out of scope for implementation it is discussed here due to its relevance to WS.

The OSA SCS in the IMS service layer provides limited WS capability to the IMS domain. It is attached to an external OSA WS gateway via the OSA API. The WS gateway in turn connects to an OSA AS over standardized Parlay X [34] interfaces as well as publishes to a WS Registry using standard WS interfaces. It can be argued than an advantage of using OSA to access external WS is that OSA SCS is in itself a secure gateway and therefore security features (authorization and authentication) are built into its architecture. However attempting to find a native service orchestration solution for IMS, which has additional capabilities to include a wide variety of (external and internal) web services in its service compositions.

There are several limitations to using the OSA approach to implement Web Services in IMS. Firstly according to the current IMS specification, the OSA SCS is the only OSA element specified within the IMS domain. Since the OSA SCS is only a gateway this implies that none of the OSA/Parlay services can be hosted inside the IMS domain. In order to use Parlay to implement native SIP messages, all SIP proxies involved must be call-stateful and all SIP requests must be mapped to Parlay requests. Cursory consideration implies that mandatory mapping of all SIP requests within a native SIP environment is redundant and inefficient. So the OSA application cannot participate in SIP
signalling flow. Instead, SIP requests are mapped to OSA API invocations via the OSA SCS. OSA services are typical telecom services such as 3rd party call control and SMS. Given that an IMS would be under the control of a telecom operator, it is counter-intuitive to adopt an architectural feature that does not allow typical telecom services to be hosted in the IMS. From a business standpoint, this is clearly not an attractive option for an IMS provider/operator and argues that it serves as a deterrent against wide adoption of the OSA CSC in IMS. Secondly, the general OSA API is not based on widely used WS standards. It is based on CORBA and it is deemed to be complex. Again, this serves as a deterrent against wide adoption. Thirdly, Parlay X Web Services is limited to a finite set of telecom services, which for version 2.1 (equivalent to 3GPP TS 29.199-1 through 14) are namely:

- Third Party Call,
- Call Notification
- Short Messaging (SMS),
- Multimedia Messaging,
- Payment,
- Account Management,
- Terminal Status,
- Terminal Location,
- Call Handling,
- Audio Call,
- Multimedia Conference
- Address List Management and
- Presence

### 2.7 Service Orchestration

The term "service orchestration" typically refers to Web Services orchestration. In this context, WS orchestration automates interactions between loosely coupled business processes that together represent a composite web service.

#### 2.7.1 Choreography v/s Orchestration

It is useful to note the differences between service orchestration and service choreography. The difference may seem subtle because they both deal service interactions, but the functional differences are important.
According to [30] choreography refers to collaboration between processes where all parties have a similar level of influence on the composite product while orchestration refers to one steering process, which gives instructions to other processes that are necessary for its composite product. There are clear parallels to the difference between orchestration and choreography in the music world. Figure 2.16 illustrates the differences.

Telco provider and controller of an IMS domain - it is the orchestration of services that is of immediate importance. However, one must note that orchestration and choreography are not mutually exclusive. They are, for example, both necessary elements in a complete web services scenario. An example of orchestration in WS is modeling of participant behavior in a single workflow as done in executable processes in the Business Process Execution Language (BPEL). An example of choreography in WS is the public message exchange between independent parties in a BPEL abstract process.

### 2.7.2 Enterprise Service Bus

The term Enterprise Service Bus (ESB) refers to a logical architecture that is used to communicate with distributed, abstracted services. An ESB is responsible for passing messages between the services that are attached to it. It is usually used in an enterprise environment between services that represent business processes.

An ESB can be classified as an EDA. In fact, A SLEE can be considered to be a
2.7.3 Service Delivery Platforms

At time of writing there was no standard definition for SDR. However, the Telemanagement Forum (TMF) is in the process of standardizing such a definition. One of the most recent efforts to this end was the TMF Service Delivery Platforms Summit in June/July 2008. One of the aims of TMF appears to be to create a strong association between SDP and SOA. One definition given of SDP in summit literature was a "framework for exposing, managing and controlling Service Provider network assets". The motivation for utilizing Service Delivery Platforms (SDP) is to achieve better integration of telecom infrastructure for rapid development of converged services. Service creation, orchestration and execution are key steps. SOA by way of web services is used sparingly in SDP, specifically for business systems. SDPs are offered not only by telecom companies but also by traditional Internet companies and system integrators. A Service Level Execution Environment (SLEE) can be considered to be a type of SDP.

2.7.3.1 Telephony Application Programming Interface (TAPI)

TAPI is a computer telephony integration API developed by Microsoft and Intel. It is integrated with Microsoft operating systems (since Windows 95) and allows the use of telephone services on individual computers and computer networks. The Java telephony API (JTAPI) provides a cross platform solution for telephony call control.

2.7.3.2 SIP Common Gateway Interface (SIP CGI)

The original CGI specification was developed by W3C to facilitate service creation and deployment in a Web Services environment. Concretely it serves as an interface to HTTP Servers. Given the parallels between HTTP and SIP, SIP CGI was standardized by IETF (RFC 3250) for the creation and deployment of services in a SIP environment. Interfacing in this case with SIP Servers.

2.7.3.3 Service Level Execution Environment

JAIN SLEE or JSLEE is the Java implementation of a SLEE. It has an EDA, which by definition indicates its support for applications that require low latency high throughput signalling. In fact, JSLEE has put an upper latency limit that must be met by JSLEE certified application servers. JSLEE 1.0 (JSR22) was finalized already in 2004 while the final release of JSLEE 1.1 (QSR 240) came July 15, 2008.

JSLEE's greatest advantage is perhaps its flexibility. It is modular. It supports many network protocols. It upholds ACID (Atomicity, Consistency, Isolation, Durability) properties and it is event driven. These properties make it a strong candidate for development of composite telecom services. However, its complexity may be a deterrent for many developers. Developers who are already at home in complex J2EE environments or/and who have experience with Enterprise Java Beans (EJB) may be better at adopting JSLEE.
2.7.3.4 SIP Servlets

As of version 1.1, SIP Servlet has adopted a central application invocation logic that is well suited for IMS application chaining. This new logic is implemented by way of a new component in the Servlet container called the Application Router (AR).

For a given framework, all SIP Servlet applications have only one main Servlet to relate to. The AR is a central component that interacts with the service container through a Java interface. On start-up, the container sends an initial query to the AR. The AR then uses a selection of relevant information it has access to (for example SIP headers, user profiles or service profile) to choose and return the name of the first application to be invoked. It also passes state information to the container at this time. The container can then use this information to make further requests as necessary. Note that this functionality matches the IMS specification for service provisioning through implementation of initial Filter Criteria (iFC).

The greatest advantage of the SIP Servlets is perhaps its simplicity especially when contrasting it with JSLEE. It is lightweight and integrated with the SIP stack and its model resembles that of HTTP Servlets, which SOA developers are already familiar with.

2.7.3.5 SDPs and IMS Requirements

In [35], Khelifi and Gregoire present a useful overview on available implementation technologies for IMS application servers and how they fit into IMS service requirements. The discussion presented provides a suitable framework for choosing a technology for the SSD platform. A list of six requirements are used to measure SIP Programming techniques as exemplified by SIP Common Gateway Interface (CGI)
and SIP Servlets, versus Application Programming Interfaces (API) as exemplified by OSA Parlay and Telephony API (TAPI), and service logic execution environment (SLEE) as exemplified by JSLEE. On ease of service customization all options received an equally ambiguous score of "possible". For the criteria on which they differed, Table 2.2 summarizes the differences.

<table>
<thead>
<tr>
<th>Table - 2.2: Overview of SDP Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>Rapid service creation</td>
</tr>
<tr>
<td>Multilayer support</td>
</tr>
</tbody>
</table>

2.8 Web 2.0 Principles

The term Web 2.0 can be considered a so-called buzzword used to describe recent trends on the World Wide Web. It is not architecture, nor is it a specification, nor is it even an official term of any sort. Never the less, there have been distinct changes in the nature in which services are provided and consumed over the Internet and these changes have on impact on traditional service providers. It is therefore deemed relevant to a discussion on WS orchestration. Since an official definition is unavailable, a representative explanation of the concept seems more fitting:

Tim Oreilly made an attempt at a concise definition of Web 2.0 in [31]. Though the resulting definition is not so concise, the principles mentioned seem to capture the Web 2.0 concept well. The terms "harnessing collective intelligence" and "Don't fight the Internet. " are two descriptive phrases from Web 2.0 evangelism that are used by O’reilly as well as his own quote from a 2005 description of web 2.0: "Data is the Next Intel Inside",

Further, in the following list, paraphrased from [31], O’reilly offers these abstract principles for building services that are compatible with the Web 2.0 concept:

"Don't treat software as an artifact, but as a process of engagement with your users. Open your data and services for re-use by others, and re-use the data and services of others whenever possible. Don't think of applications that reside on either client or server, but build applications that reside in the space between devices. Remember that in a network environment, open APIs and standard protocols win, but this doesn't mean that the idea of competitive advantage goes away. Chief among the future sources of competitive advantage will be data, whether through increasing returns from user-generated data, through owning a namespace, or through proprietary file formats."

2.8.1 General Characteristics

Some important features of Web 2.0 services are summarized in the following list:
• **Social**
Web 2.0 services are inherently social in nature and are named as such.

• **Dynamic in Nature**
Web 2.0 services are never static. Here is a list of some its dynamic service elements that illustrate this key aspect:

• **Web Feeds**
This is a method of publishing frequently updated content to users that subscribe to such alerts. Content providers syndicate content and content consumers collect an aggregation of updates from different sources into one application (called a newsreader). Newsreaders may be web-based or desktop. Really Simple Syndication (RSS) is one popular feed format for syndicated news. Facebook provides proprietary news feed of what friends were last "seen" doing on Facebook (e.g., "Bob updated his profile picture", "Alice added Bob as a friend")

• **Tags**
Tagging is the use of keywords to describe web content such as pictures, parts of a picture (e.g. a specific person in a photo) and bookmarked web pages. Tags are usually subjective and a single resource usually has several tags. Users of social networking sites like Facebook [36], social bookmarking sites like del.icio.us and picture sharing sites like Flickr [32] use tagging extensively. In fact, tagging is an integral part of del.icio.us' and Flickr's concept.

• **Collaboration**
This is a reference to services that allow many individulas to collaborate on the development of a composite product. Example applications include wikis which are collectively edited web pages, the most well known of which is www.wikipedia.org, a heavily used wiki encyclopedia.

• **User Generated Content and Services**
The fact that content and services are now actively generated by Internet users is perhaps the most revolutionary part of the Web 2.0 concept. Here are some concrete examples:

- The afore-mentioned **tags** are user generated.
- The afore-mentioned **wikis** are user generated.
- **Podcasts** (syndicated media files available over the Internet)
- **Web logs (blogs)** were one of the first Web 2.0 trends. Special blog formats include audio-blogs (usually as podcasts), video-blogs, photo-blogs and micro-blogs (e.g., twitter.com), micro-blogs are for sending very short updates such as "I'm hungry" or "just woke up". There is usually a very low restriction on maximum number of words/characters. It is therefore suitable for a mobile-blog (**verb: moblogging**) where updates or "tweets" as they are called in Twitter can be easily published via SMS. There is a plethora of freely available blog platforms on the web which make it
easy for even the less technically inclined to create and maintain blogs. Popular alternatives include blogger.com [37] and wordpress.com [38].

- The chance for users to develop and deploy their own loosely coupled services is becoming increasingly popular. Facebook [36], Myspace [39], and iGoogle [40] are some of the pioneers in this field. Generally speaking, the platform in question offers an API that customers use to develop compatible applications. Licensing and ownership of the finished product varies.

### 2.8.2 Mashups

A Mashup is a hybrid web application that combines application data from several sources into one blended offering. The main advantage of Mashups is that the unique hybrid offers a service that could not have been offered singly by any of the constituent services. As long as the appropriate interfaces are available from the potential services and APIs are available for interacting with them then only imagination and time restrict what kind of mashups are possible.

A so-called telecom mashup combines services from different providers into one integrated user experience. Service delivery platforms (SDP) are used to develop and deploy telecom mashups. Telecom mashups are deemed to be complex and present a challenge, especially in light of security, privacy and low-latency requirements of typical telecom services. In addition, telecom services are built on business models that rely on billing for the use of services, whether individually or as subscription package. Mashups evolved on the Web which is a much more open environment, its projection onto the telecom domain provides a billing challenge but it is an issue that must be addressed soon as telecom trends move towards all over IP.

### 2.9 IMS Service Creation Environment & Toolkits

#### 2.9.1 IMS Service Development

The main characteristics of IP convergence and IMS architecture show ability for the development and deployment of new services. According to the full cycle of the service development, software development personal manage new services developments, and operations team takes care of running and deployment of these services. There are some steps of full cycle service development. These steps include “Modeling the Service”, which is the simulation of alternative scenarios, capturing flows as well as service activities. “Investigate Requirements” defines the requirements of the service, by visualizing different flows and user interaction models.

It also captures the requirements for business and technology. “Design & Construct” are for the creation of high quality services; the team follows a development paradigm, and also translates the requirements for better service. “Test” conducted for the acceptable performance. This work is done
as part of quality assurance of the software / system. “Deployment” services are appropriate deployed to get complete coordination and management for best environment in service execution. In “Monitor” the feedback is taken by the comparison in-between actual values and projected values. After that important adjustments are made for batter business.

2.9.2 Unified Service Creation Environment (USCE)

To support the full cycle of IMS service development, need such kind of service environment, which is robust and has strategic importance. USCE has vast variety of set of tools, which offers best professional develops business driven services. These tools support different functions during service creation and roles in all software development life cycles. There are some areas of software development life cycles, which are mapped by these tools. These areas are defined as “Requirements & Analysis”, which are Integration tools for data modeling use case development, business modeling, and requirement management. Design & Construction tools for run time analysis activities, architecture and design modeling, model driven development, component testing.

Software Quality defines functionality; reliability and performance are the three dimensions of software quality. And these tools address theses dimensions. Software Configuration Management defines any change including version control is simplified and managed. Solutions are available for change and defect tracking, software asset management. Process and Portfolio Management is the integrated solutions, which can implement proven development process, assess and report progress and help the teams to manage the change and requirements.

2.9.3 USCE (Unified Service Creation Environment) Toolkits for IMS

For the development of SIP and IMS applications, USCE includes some toolkits. These toolkits create SIP and IMS applications. The toolkits are named as, application server tool kit, IMS enablement toolkit and web services toolkit. In service creation environment, SIP and SIP/HTTP converged applications are created by application server tool kit, IMS foundation applications are created by IMS enablement toolkit and composite IMS applications are created by web services toolkit.

In runtime environment, converged container for SIP and application server through SIP supports HTTP servlets during runtime process. Business process security, transactional integrity, consistency is executed by process server, which is high performance business engine. Service continuity, service oriented integration, Java message service (IMS) is provided by enterprise service bus.
2.9.4 Application Server Toolkit

This toolkit is used to deploy, test and create applications that run within the application server. Application server toolkit is eclipse based integrated application development environment. The tools used in application server toolkit are integrated into a workbench, which is based on web tools platform and eclipse technology. This toolkit focuses from the application deployment tool to the application development tool and has some features.

Tools for jython development:

Web tools platform version V 1.0.2 is included Tools support for JSR 168 port lets development For the testing and publishing applications, tools are more improved. Tools supporting JSR 116 SIP servlets application development

The application server toolkit workbench toolkit includes J2EE application, which can allow the creation of web services from different files like enterprise Java beans, Java beans, WSDL. By this characteristic can create, deploy and modify enterprise Java beans, container managed persistence (CMP) can be do-bottom up, back-end database can be supported for enterprise Java bean (EJB) deployment code. Recently it was not possible and could only do deploying and assembling of enterprise Java bean (EJB), and could only modify their deployment descriptor.

2.9.5 Perspective of AST V6.1 Tools

The SIP application development tools supported by application server toolkit version 6.1 are translated through J2EE perspective. Converged SIP/HTTP and SIP projects are two projects namely Converged SIP/HTTP and SIP applications are represented by application server toolkit. SIP Servlet Development (JSR 16) application server toolkit version 6.1 indorses the capability for the development of SIP servlet, which is based on JSR 116 specification.

This is an addition to HTTP servlet development. Import/ Export of SAR packages define that, from inside application server toolkit, Sip applications cannot be directly installed on application server like other J2EE or Web applications. So for this purpose have to use import/export tool to export SIP application. It will be done to export SIP application as a part of enterprise archive (EAR) or standalone SAR package. Then use the administrative console to install the package to the application server. SIP Deployment Descriptor Editor defined for the packaging and configuring SIP applications, editor is used.

SIP application resource (SAR) is a new construct in which SIP applications are packaged. Http servlets and SIP servlets are encapsulated by these editors and are fully supported by application server toolkit version 6.1. In the application of the servlet, different parameters can be specified and configured by using SIP Deployment Descriptor Editor. For example if want to add servlet mappings by using SIP Deployment Descriptor Editor, SIP
messages routes to the exact servlet for processing. [41]

Some other tool features for application server version 6.1 are available.

- Server Tools used for the support of unit testing and debugging.
- They support SIP and Jython tools for application server specific-extensions
- Graphical editors are available for deployment descriptors and application server property files.

2.9.6 SIP Servlet Application Development

JSR 116 servlets, which are also known as siplets are developed by the application server tool kit. In this way develop and test SIP applications in the same manner test and deploy J2EE applications. The group of source files, servlets, and resources, which are also known as session initiation protocol (SIP) applications can be managed in single unit. By using the application server toolkit, can develop SIP only for HTTP and converged SIP servlets. [41]

2.9.6.1 SIP only Applications

For the development of new SIP application, needed a new SIP project wizard. The wizard creates SIP project and has additional information in the project menu like, SIP servlet, SIP project and converged project. SIP servlet give the application functions and gives the definition of servlet class name, super class, package & location. In the unique SIP mapping section, use the deployment descriptor information to enter the SIP servlet deployment descriptor information. The wizards also define the interfaces and different methods. In the servlet class, stubs are created and implement the servlet by developing code for stubs.

2.9.6.2 Converged SIP/HTTP Applications

The creation of hypertext transfer protocol (HTTP) and session initiation protocol (SIP) applications is supported by application server toolkit. The dynamic web project which is also known as converged project with SIP contents and also have web deployment descriptor files and SIP deployment descriptor files. If the web project already has HTTP servlets according to SIP specifications, it can make easy to check the symmetry between web.xml and sip.xml. SIP servlets can be added from the deployment descriptor editor, when the SIP deployment descriptor is created.

2.9.6.3 SIP Servlet Deployment

The SIP application resource (SAR) file is a deployable SIP application which is a basic Java archive with ".sar" file extension. It also includes SIP
deployment descriptor file. The extensions of web archive resource (WAR) export and import wizards, are SIP application resource (SAR) export and import wizards. The SIP project exporting is allowed by the SAR (SIP application resource) export wizard and tells us for include export files or not. During export process, SAR export operation has additional validations, which ensures the compliance with SIP specifications. This is the specific case of converged SIP/HTTP application. Servlet context manages the initialization parameters for SIP and HTTP, while the WAR file may have the converged application. The reverse of import wizard is the export wizard, while in import wizard can import previously exported SAR file in workspace. The creation of SIP project occurs when SAR file is a SIP only application. Otherwise SAR file is a converged SIP/HTTP application and then project with contents of SIP and HTTP will be created.

The SIP deployment descriptor (DD) displays about the deployment of SIP applications. For the syntax of the deployment descriptor file, XML is used. For the SIP applications, information is stored in "*.xml" file and for HTTP applications, information is stored in "web.xml". While exporting the SIP application, deployment descriptor is used to build the SAR file. Application server tool kit gives provisions to SIP deployment descriptor editor for the maintenance and creation of sip.xml descriptor file and also supports to web development descriptor editor to maintain and create web.xml.

For specify deployment information for modules, web deployment descriptor editor gives this provision and these modules are created in web development environment. And the information of modules will be appeared in web.xml file. Using web deployment descriptor can set web deployment descriptor attributes. The information, which is necessary for deploying web application module, is in web.xml file, which is also used for creating a WAR file. Web deployment descriptor includes many views and also a dynamic, which defines many settings and properties while deploying descriptor.

SIP deployment descriptor editor has multiple tabbed pages like variables, references, security, servlet, source page and overview. can use these pages as to get summary of contents can add, remove or change the contents. The overview page extends web deployment descriptor overview page. The irrelevant pages to SIP are removed and relevant pages to SIP are added, e.g. "login section". The sections of overview page include general information, servlets, login, security, icons, listeners, references, environment variables and context parameters. Servlet page enable adding, removing or modifying of servlets in the project. Servlet page has the same sections as web deployment descriptor except some changes. And these changes are URL mapping section, which are replaced with mapping section for SIP. Also extension sections and programming model extensions are removed. Security roles and constraints of project are managed by security page. Security page of web deployment descriptor editor is extended by security page. In security page, web resource collection section is replaced by SIP resource collection section. The variable page enables to manage the list of listeners, environment variables and context parameters in the project. Reverences pages manage the project resource references. The references pages support remote EJB reference, local EJB reference, resource environment, and resource reference. The source page finally shows sip.xml source code.
2.9.6.4 Sample SIP Services

The capabilities of session initiation protocol are presented by many samples, and these samples are contained by application server toolkit. These samples are call-blocking sample, call forwarding sample, third party call control.

2.10 Hardware & Software Requirement for Application Server Toolkit

2.10.1 Hardware Requirement

- Intel Pentium III 880 MHz Processor (1.0 GHz Recommended)
- 512 MB RAM (Minimum), (1.0 GB is recommended)
- 900 MB disc space, and 100 MB TEMP disc space is needed during Installation
- Display 1024X768 (Minimum)

2.10.2 Supported Operating system

- Windows XP
- Windows 2000
- Windows Server 2003
- Red Hat Enterprise Linux 3.0
- Red Hat Desktop Linux 3.0
- SUSE Linux Enterprise Server (SLES) 9

2.11 IMS Enablement Kit

For getting access to core IMS enablers, IMS enablement tool kit adds more features in application server toolkit. These features describe the group list and presence from SIP servlet. The plug-in of IMS enablement promotes the J2EE development, Java 2 platform to reinforce the foundation level creation properties of IMS applications. The enhanced J2EE focus on different perspectives, which are,

- To accelerate the development of JSR 116 SIP servlets, it enhances the service Creation with new servlet wizards and SIP project.
- For the editing of SIP deployment descriptions like other J2EE components, it supports wizard and SIP archive (SAR) format.
- For the acceleration deployment to runtime servers, it packs the SIP components and J2EE.
- To decrease risk of errors, it compiles the automatic inclusion of SIP
application server (SIP A/S).

- It has a gallery of SIP sample services.
- It has a specific help plug-in for IMS

The enablers like IMS enablers, location services; short message services (SMS) are used by the foundation level applications, which provides end-user encapsulated services. To make more rich IMS composite applications, foundation level applications gives the provisions of more interfaces.

2.11.1 IMS access gateways and enablers

The access gateways and enablers are, Presence Server, which communicates to users and manages, distributes and collects real time information regarding user's availability, access and willingness. In Group List Server Component, group list server provisions Network based groups are managed and created by administrators and users, and these users and administrators. Permissions, access lists, other service-specific properties are maintained by group list server. In IMS Connector Industry leading application server platform is induced by some specific IMS-interfaces from IMS connector. This function is performed to deliver full IMS standards-complaint SIP application server.

2.11.2 IMS Enablement Tool Kit Components

Following are the components of IMS enablement toolkit

A. Diameter Resources

- **Rf Interface Libraries:** It provides diameter-messaging interface to IMS application server application. And the purpose is to run the application, so it may send offline accounting messages to billing or accounting servers.

- **Sh Interface Libraries:** The data of the user profile is updated and retrieved from home subscriber server (HSS). And the function is performed by Sh subscriber profile web services.

- **Corresponding web services description languages (WSDLs) of Sh and Rf services:** Through web services, interaction is made in-between IMS application server and user profile server. And due to this interaction, charging is performed. It also indices a function to notify the status of the specified user.

- Test clients of Rf and Sh.

B. Presence Resources

- **Authorization APIs Library:** this library develops Customized permission policies. These APIs also develop the plug-able
applications. These plug-able applications allow or disallow, authorize and give information to subscription to presentity.

C. Parlay X.21 Resources

These are web services description languages (WSDLs) and include call notification, third party call, payment, SMS, terminal status.

2.11.3 Sample Application for IMS Foundation

There are three sample applications of IMS toolkit. One is IMS service control (ISC) SIP servlet and two diameter clients. These samples give demonstration of using diameter Sh and Rf clients and IMS service control (ISC) & session initiation protocol (SIP) servlet API.

A. IMS Service Control (ISC) Interface Sample

It gives the presentation about the implementation of back-to-back user agent (B2BUA) service by using ISC and SIP servlet API.

![Figure 2.19: ISC Demo Call flows [35]](image)

The figure 2.19 shows the SIP server, CSCF (call session control function) and interaction between user agents. User agent 1 sends a call to ISC demo SIP servlet and ISC demo SIP servlet receives INFO on same SIP session. This SIP session provides user agent 2 (UA2) address. Through serving call session control function (S-CSCF), ISC demo makes a call to user 2 and sends INFO to user 1 and user 2 separately. Then the call will be hanged in-between user 1 and user 2 by ISC demo. ISC demo is a SIP servlet, which is used to implement different methods like doinvite, doAck, doError, doBye, doRequest, doResponse, doSuccess, doInfo. IMS service control demo App (ISC demo App) holds the getters and setters and holds all the states. IMS service control demo App Handler (ISC Demo App Handler)
used to handle the request. It performs some specific actions about the moving from current state to new state. The processing action INVITE from user 1 is also handled by IMS service control demo App Handler (ISC Demo App Handler).

B. Diameter Client Sample

There are two-diameter client samples.

- Diameter Rf test client, which uses offline charging WSDL DiameterRfService.wsdl
- Diameter Sh test client uses the user profile management WSDL DiameterShService.wsdl

In general both applications processing are similar in simulation and software configurations.

2.11.4 Hardware & Software Requirement for IMS Enablement Tool Kit

A. Hardware Requirements

- Intel Pentium III 880 MHz Processor (1.0 GHz Recommended)
- 512 MB RAM (Minimum), (1.0 GB is recommended)
- 900 MB disc space, and 100 MB TEMP disc space is needed during Installation
- Display 1024X768 (Minimum)

B. Supported Operating Systems

- Windows XP
- Windows 2000
- Windows Server 2003
- Red Hat Enterprise Linux 3.0
- Red Hat Desktop Linux 3.0
- SUSE Linux Enterprise Server (SLES)

2.11.5 Web Services Server Toolkit

To create the higher level of services, web services server toolkit is used by the IMS composite applications. These IMS composite applications contain service enablers and foundation level applications. Service enablers and foundation level applications require principles of service-oriented architecture (SOA), or say that service enablers and foundation level applications acts as service oriented architecture (SOA) service implementations. New composite services are created when plan these services and enablers by using BPEL. These new services are also service-
oriented architecture (SOA) services and can create high level of services. For the composite IMS application, need to install web service toolkit and also require integration developer, which is powerful and integrated platform.

2.11.6 Capabilities of Telecom Web Services Toolkit (TWSS)

One of the important components of telecom platform is telecom web services toolkit (TWSS). The capabilities and exposure of the network is enhanced and illuminated by technology and language independent high level web service interfaces. These interfaces may be access through custom integrated services, direct connect access to network protocols, SIP or Diameter, PSTN functionality through parlay.

A. TWSS Service Implementation

These implementations have many reusable components, which are deployed at application server. They provide the best implementations and high-level service interfaces, which reinforces the network for convenient access of it.

B. WSS Access Gateway

It provides the capabilities like authorization, message capture, management, policy driven traffic monitoring. This is a gateway in-between service ends points and clients and this gateway implements such policies on all such requests and responses occurred in-between these service end points and clients. This gateway enhances the system capability by adding policy driven processing elements. It has mediation primitives, which are the components used to assemble customized message process flow. The programming model and mediation primitives interface creates the custom function and gives the points for extensibility.

C. Service Policy Manager

It provides the capability of access to service policies, data administration, access mechanism and definition of administration interfaces, storage. Attached policies, requesters and service definitions can be taken out by the usage of Service Policy Manager. Service Policy Manager also defines associated service subscriptions to requesters.
2.11.7 Mediation Service

The message processing between service providers and service requesters are allowed by enterprise service bus and mediation is a function of this enterprise service bus. By using mediation modules, these services are implemented and they modify and intercept messages passing in-between providers and requesters. The mediation modules in mediation flows, which process the messages, provide the logic. By using the mediation flow editor these flows are maintained as well as created. These flows are executed in a sequence and they consist of series of processing steps. The end nodes are translated by using these mediation flows and these nodes are based on source operation. For the creation of request and response flows, mediation primitives are added and help in execution of sequence between end-nodes. These request and response flows give the processing logic. The receiving of messages, processing them and send the processed message to next node is the main function of mediation primitives.

2.11.8 TWSS Access Gateway Mediation Primitives

The several mediation primitives are contained by telecom web services access gateway and they are divided into two steps. [18]

A. Mandatory Mediation Primitive

These are the primitives from the base of the TWSS gateway flow and configuration. The add-on mediation primitive function is supported and base services are provisioned. These primitives are:

- **Transaction Recorder Mediation Primitive**
  
  Within the table, the transaction information is recorded by the transaction recorder mediation primitive and is referenced by other mediation primitives

- **Policy/Subscription Mediation Primitive**
  
  Policy/Subscription Mediation Primitive retrieves the policy data that contains service, requester and operation being called. During the mediation primitive execution, policy data acts as decision parameter.
• **Service Invocation Mediation Primitive**
  
  Appropriate end points from the messages are extracted and further the message will be prepared for dynamic service invocation on the next step.

**B. Optional Mediation Gateway**

TWSS gateway flow uses following optional plug-ins:

- **Network Statistics Mediation Primitives:**
  
  The results of the exit information are stored in the database and the entries are entitled of records messages.

- **Service Authorization Mediation Primitive:**
  
  For web services operations it provides fine-grained authorization.

- **SLA (Service Level Agreement) Enforcement Mediation Primitive:**
  
  It enforces the policies recommended by service level agreement and measure the system usage.

- **Group Resolution Mediation Primitive:**
  
  For the implementation of parlay-X services, use resolution mediation primitive, for the given operation, these parlay-X implementations accept the group of URI's within the list of target destinations. The member URI's are replaced and expanded by the group URI's by Group Resolution Mediation Primitive.

- **CEI Event Emitter Mediation Primitive:**
  
  For the implementation of common base event (CBE), this primitive is used by alarm and fault common component. By this implementation, can pick up fault information by different monitoring systems.

**2.11.9 TWSS Default Message Flow**

![Figure 2.21:TWSS Default Message Flow [18]](image-url)
The default flow implementation is the provision of telecom web services access gateway. For different capabilities like traffic level enforcement, service/operation level authorization, message capture regulatory purpose and accounting of requests are supported by the default flow model, which is message processing function by a service provider.

2.11.10  Hardware & Software requirements for TWSS Toolkit

A. Hardware Requirements [418]

- Intel Pentium III - 1GHz Processor (Higher is Recommended)
- 1 GB RAM (Minimum 1 to 2 GB RAM is recommended)
- 5.5GB of Disk Space (If file system is FAT32 and not NTFS, more space will be required.)
- 1 GB for the TEMP directory is required
- Display 1024X768 Minimum (1280x1024 is Recommended)

B. Software requirements

- Windows 2000
  - Windows 2000 Advanced Server With SP3 and SP4
  - Windows 2000 Server With SP3 and SP4
  - Windows 2000 Professional With SP3 and SP4

- Windows 2003
  - Windows Server 2003 Enterprise Edition

- Windows XP
  - Windows XP Professional With SP1 and SP2

- Linux
  - Red Hat Enterprise Linux 3.0 WS Update 2
  - SuSE Linux Enterprise Server 9

2.12  IMS Service Creation Tools

2.12.1 IMS service creation

For the service creation of IP multimedia subsystems (IMS), need such kind of tools, which are helpful for an appropriate functionality of IMS. Different IMS toolkits, application server toolkit, IMS enablement toolkit and web service toolkit and their capabilities. In this part will discuss different tools used by these toolkits for the service creation of IMS.
2.12.2 Web tools platform for AST (Application Server Toolkit)

For the building of J2EE and web applications, web tool platform provides such a platform for these applications. This web tools platform is an eclipse based project and shows the development of many tools and wizards. There are two standards of web tool platform, one is a Web standard tool (WST) and other is J2EE standard Tools (JST). The JST gives all the specifications of J2EE and these specifications are Enterprise Java Beans (EJB), Java Server Page (JSP), servlets and Java web services. WST has all language neutral functionality for web applications building. WST also has document generators for languages like (SQL, Extendable Markup Language (XML), Hyper Text Markup Language (HTML)), validators, and editors. [41]

2.12.2.1 Categories of web tool platform

Within the development environment, server tools provide the management to J2EE servers. By the help of server tools, can set up server connections, deploy and test applications and work with servers. These applications and activities can be viewed at the bottom of application server tool kit workbench. Web Tools provides creation of several web artifacts, which includes style sheets, Hyper Text Markup Language (HTML), Java scripts and others; web tool gives the provision of many editors and wizards. Web tools also present TCP/IP monitor and embedded web browser.

XML Tools provides validation and building of schemes, XML artifacts, Document Type Definition (DTDs), XML files, XML tools offer editors and wizards. Web Services Tools provides transformation of J2EE artifacts into other web services, web services tools gives the provision of web services wizards and Web Service Desription Language (WSDL) editor. J2EE Tools defines creation and working with J2EE artifacts, like EJBs and servlets, these tools represents a number of editors and wizards. J2EE also represents special editors for deployment descriptors.

Data Tools gives an interaction with variety of databases. For the generation of Enterprise Java Bean (EJB) deployment code, which is added to many databases, are provisioned and supported by back end databases. [41]

2.13 Portlet Development Tool

These tools are compliant with the Java Specification Request (JSR) 168 portlet specifications. These tools support the creation of portlets and portlet projects and they also presents portlet deployment descriptor editor. The generated portlets extends generic portlet class and they also have stubs for required portlet methods. By using the application server tool kit (AST), also import portlet Web Archive (WAR) files. Resources can be imported from a WAR file to proposed project [41]
2.13.1 Server Tool

By the help of this tool, it can be publish any running instance of application server locally or remote. This tool improves support and defines the connection to data source and also gives the support for additional back end databases. The tool improves the Enterprise Archive (EAR) capability, which represents preparation and packing of applications for publishing application server. In the case of enterprise deployment descriptor, this improved EAR acts as deployment page. This tool is updated to give different provisions like adding of resource adaptors, message queues, connection factories and topics to an enterprise application on the target for the application server. This tool helps to launch profile management tool from the workbench and also helps to remove configuration and registry files, which are related with the workbench [21]

2.13.2 Telecom Web services Tools

Telecom Web services Tools are divided into two parts. One is an application-testing tool and the other is an application development tools. The application development tools consist of snippets, telecom application templates, cheat sheets, WSDL import wizard, and telecom web samples. The application-testing tools consist of simulator and runtime views and simulator configuration editor. [41] Parlay X2.1 WSDL Import Wizard is used to import parlay 2.1 WSDL into the existing project; Parlay X2.1 WSDL Import Wizard used. Telecom Simulator Configuration Wizard is used for creation of new custom simulator configuration file; use Telecom Simulator Configuration Wizard.

Using telecom application template sample, cheat sheets gives the provision and guidance for the creation of new applications. Each cheat sheet is addressed and designed to complete some tasks and it makes some steps with sequences to achieve the goal. These cheat sheets are designed to develop and test telecom web services. To develop and test parlay X 2.1 telecom client application, cheat sheet provides four steps. These four steps are:

- **Create a web project:** This step reinforces and helps throughout the process of loading the template of telecom web application

- **Create the JSP servlet for your application:** To call a telecom web service, this step guides the process thoroughly to create a client. This client may be a JSP or a servlet

- **Add telecom web service call:** by using the telecom snippets, this step guides the process to add web service call in the servlet client or JSP.

- **Test your application:** By using the web services client simulator, the step guides the process to test user client application. The web services client simulator runs on integrated application server in rational application developer.
Snippets are used to insert pre-defined working code in Java server page or Java class. These snippets are used to call different parlay X web services, which are multimedia media message service, short message service, Terminal location, accounting, payment, terminal status, third party call, audio call, notification administration, presence, group management, wireless access protocol push. The Categories of Snippets includes Telecom Message Service Category, Telecom Terminal Location and Status Web Service Category, Telecom Account Web Service Category, Telecom Notification Administration Web Service Category, Telecom Payment Web service Category, Telecom Call Web Service Category, Telecom Audio Call Web Service category, Telecom Group and Group Management Web Service Category, Telecom Member Web Service Category, Telecom Presence Web Service Category, Telecom Call Handling Web Service Category, Telecom Address List Management Web Service Category, Telecom Wireless Access Push Web Services Category.

The samples are used to make a toolkit that can be tested and deployed on web services client simulator. Web services client simulator provides the test suite and emulates the parlay X gateway to test the web applications of user development parlay X. And it does not needs a real network for this. For the configuration of test data, the simulator uses configuration file, which is xml.wss file and has static configuration file, which tells us the behave of the service.

2.14 Tools Developing & Testing SIP and IMS Sample Applications

2.14.1 SIPp

It is a free open source test tool/traffic generator for the SIP protocol. It is available under GNU general public license. This tool has a current version, which is 1.1rc5. It also represents some basic SipStone user agent scenarios like UAC and UAS. These tools read the SIP messages contained by XML scenario files and these files. These tools can describe different call flows leading from simple to complex. [42]

2.14.2 Ethereal

It is a network packet software analyzer, which is available for both Linux and Windows environments. This is a release under GNU general public license and open source software. It captures the network packets, which are flowing from or to the selected network interface and displays in real time with protocol independent information. It supports SIP protocol and represents filtering capabilities. [42]

2.14.2 SIPx Phone

It runs on Microsoft Windows and Linux. It is fully functional SIP soft phone. The phone client supports hold, multiple simultaneous calls, and client
mixed conferencing, mute, consultative transfer, authentication, multiple line appearances and extendable Java based application environment. It is developed as open source and hosted as part of SIPX line projects. These projects are available from SIP foundry. It is licensed under LGPL. [43]

2.14.3 Comparison of IMS traffic analyzing tools

These tools are regarding to different tools kits like application server tool kit, IMS enablement tool kit, web services server tool kit and also discussed some other tools which help in IMS applications. According to task need to have some information in which may analyze SIP message, time stamp and regarding information. Which give information about the decoding of SIP messages and other useful functions. The name and capabilities of measurement tools for IMS are as under

2.14.4 DA 3400, DA 3600A VOIP Analysis [JDSUO]

A. Capabilities:
- Real time analysis of 64000 simultaneous calls (DA-3600A)
- Real Time Analysis of 8000 simultaneous calls (DA-3400)
- Simultaneous VOIP and IP data Analysis
- MOS/R factor and detailed statistic for each call
- Extensive display customization and filtering
- Signaling analysis with call signaling trace
- Support post capture analysis and play back with PVA-1000

2.14.5 PVA-1000 VOIP Analysis [JDSUO]

A. Capabilities
- MOS and R factor Analysis
- Jitter and Packet loss analysis
- Audio playback with multiple CODEC support
- Signaling Analysis with call trace
- Signaling support for SIP, Cisco, SCCP, MGCP and DOCSIS/NCS
- Compatible with wide range of JDSU test and analysis equipment
- Compatible with Wireshark (PCAP)
- Distributed automated VOIP call capture agent option
2.14.6 Packet Scan [GL Communication©]

A. Capabilities: [45]

- Monitor progress of up to 500+ simultaneous calls with bidirectional RTF traffic. Supported protocols include SIP, Megaco3525, Megaco3015, MGCP, H323/H225 and RTF.
- Call capturing based on call agents or trigger actions such as MOS, R-factor, Jitter, packet loss, duplicate packets or called/calling numbers (SIP/H323).
- Provision for H263+ video capture and video conference monitoring capability.
- Decode AMR in all packet format and G.726 RTF and AAL packing types.
- Real time audio/video monitoring of RTF streams using audio playbacks, record video, and write to file features.
- Call quality of service for all calls with E-model based G.107 MOS and R-factor with individual and summary statistics presented in graphical and tabular formats.
- User can get real time call trace information based on SIP, H.323 calls.
- Ability to configure sipport.ini file for customization of decoding options.
- Calculate minimum, maximum and average RTD values for SIP calls.
- Provide summary, detail, hex-dump and call detail records view of captured traffic.
- Packet analysis displays call information in graphical format as well as in tabular format.

2.14.7 Hammer Call Analyzer [Emprix©]

A. Capabilities: [46]

- Real time, multi stage, multi protocol call flow display.
- Auto association of messages across signaling domains.
  - RTF media quality analysis, MOS scrolling for voice and video quality, save and play back.
- Full VOIP protocol decode.
- Intuitive, protocol aware searching, filtering, and capture.
- Pre-trigger packet capture.
- Import standard libpcap traces for analysis.
- Export standard format for documentation.
• Analyze VOIP, IMS and TDM call using a single integrated analysis tool and interface

2.14.8 Hammer G5 [Emprix®]

A. Capabilities: [46]
• Multiple VOIP signaling protocols for IMS and NGN testing
  • Emulate real end points with common, separate and unique characteristics for signaling and media
  • Simulate real world subscriber behavior with pre-defined or customized caller profiles and load patterns
• Multiple codec and media types for wire line and wireless applications
• Active and passive verification and analysis of media quality
• Customizable protocol messaging and signaling behavior
• Remote access and control
• Comprehensive test monitoring and reporting
• Flexible test creation and execution options
• Secure transport testing

2.14.9 THGNOTEBOOK [Finisar®]

A. Capabilities: [47]
• THGnotebook features 7-layer packet decode and analysis, match including real time network statistics, advance alarm sating and action, multi state pattern, filters and automatic name table updating
• Provides a cost effective full power portable solution for measuring, analyzing or monitoring 10/100/1 Gbit Ethernet

2.14.10 VOIP testing [Radcom®]

A. Capabilities: [48]
• VOIP testing analyzer used to capture, filter and analyze raw and decoded data on a wide variety of networks
• It decodes over 600 telecom and datacom protocols including standard protocols such as ITU-T, 3 GPP and 3GPP2 and country/vendor specific variant as well including protocols such as SIP, H.323, RTF, Megaco and MGCP
• All layers of protocols are supported with VOIP testing
2.14.11  nGenius (Infmitestream) [NETSCOUTO]

A. Capabilities: [49]

- Packet troubleshooting recording
- Voice convergence management
- Data Granularity
- Network visibility
- Sophisticated Sniffer Intelligence analysis
- Response time Analysis/KPI
- Network Management Marries
- Alarming and Event Identification

2.15  Summary

During research initiation, all existing protocols and architecture studied. Discussion with each protocol and entities works in IP Multimedia Subsystem discussed in this chapter.

Study for all technologies, protocols and tools support IMS. Aim for this project is to use Open Source software maximum. Then also deep root study of each and every tool and their capability with respective purpose of use during proposed research mentioned. Although majority of existing tools discussed over here, among a many used during practical work including IMS implementation, result, testing, security observation etc.