Universidade de Aveiro
2008
Departamento de Electrónica, Telecomunicações e Informática

Gabriel Fartaria Ferreira

Serviços IP Multimédia em Redes VoIP/3G

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. Rui Luís Andrade Aguiar, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri

presidente	Professor Doutor José Carlos Silva Neves Universidade de Aveiro
orientador	Professor Doutor Rui Luís Andrade Aguiar Universidade de Aveiro
arguente principal	Professor Doutor Manuel Alberto Pereira Ricardo Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Eng. da Univ. do Porto

agradecimentos Dedicado aos meus pais e irmã, sem eles nada disto teria sido possível.

palavras-chave IMS, SDP, SOA, OSE, Convergência, Serviços, Redes, Mobilidade

resumo Esta dissertação tem como objectivo o estudo da arquitectura ‘IP Multimedia Subsystem’, seus protocolos e entidades constituintes, e averiguar a implementação de uma plataforma de entrega de serviços que siga os conceitos e directrizes presentes na arquitectura orientada a serviços, mais especificamente a plataforma ‘OMA Service Environment’.

keywords IMS, SDP, SOA, OSE, Convergence, Services, Networks, Mobility

abstract The main goal of this dissertation is to study the IP Multimedia Subsystem, its constituting protocols and entities, and to evaluate the implementation of a service delivery platform that follows the concepts and directives present in service oriented architecture, more specifically the OMA Service Environment platform.

ÍNDICE

Lista de Figuras

Lista de Tabelas

1. Introdução

As redes de comunicações têm uma história que poderá se definir como evolutiva, por vezes novos conceitos dando origem a novas redes e outras vezes o inverso, com o intuito de preencher as necessidades da época. Neste caso verifica-se a necessidade de convergir as diferentes redes existentes que poderá ser conseguido através de um conjunto variado de soluções sendo a arquitectura 'IP Multimedia Subsystem' (IMS) uma dessas soluções. A de convergência de diferentes tipos de redes de telecomunicações vem da necessidade de integrar aplicações dentro e entre diferentes negócios e seus processos que poderão estar utilizando diferentes tecnologias sendo necessário encontrar um modo de permitir com que interoperem quebrando o modelo anterior de redes baseadas em silos onde serviços eram construídos de um modo independente e proprietário. Também a pressionar essa convergência vem a necessidade das operadoras de telecomunicações de diversificarem os seus modelos de negócio estendendo o modelo anterior baseado no fornecimento de serviços de voz e dados que se tem estagnado ao nível de lucro médio por assinante para um modelo baseado no fornecimento de múltiplos serviços personalizados ao assinante fornecidos em qualquer lado a qualquer altura que poderão ser desenvolvidos, continuamente melhorados e reutilizados de um modo rápido e por terceiros reduzindo os custos de implementação de novos serviços e assim reduzindo os riscos associados com a sua criação. Para isso as redes IMS apenas fornecem a plataforma base de sinalização que irá convergir as redes tradicionais de comutação de circuitos com as redes de comutação de pacotes fornecendo ao utilizador um acesso agnóstico à rede utilizada, para a gestão de serviços e seu fornecimento o IMS apenas estandardizou um conjunto de interfaces para a interacção com entidades que fornecerão os serviços deixando um vazio que pode por em causa a sua adopção. É aí que entram os conceitos da arquitectura orientada a serviços, 'Service Oriented Arquitecture' (SOA), que trazem a capacidade de desenvolver serviços altamente granularizados com base na utilização de standards comuns e reutilização de serviços existentes fornecendo os meios para a gestão do seus ciclos-de-vida e utilização de políticas de gestão. Mas o SOA trata-se apenas de um paradigma que para ser aplicado numa rede IMS terá de ser adaptado e posto em prática numa plataforma de entrega de serviços, 'Service Delivery Platform' (SDP), que define as directrizes principais, entidades

lógicas e suas interacções sendo através deles que se poderão aplicar correctamente os conceitos de SOA. Existem diferentes implementações de SDP com diferentes objectivos e destinatários, para as redes IMS o 'Open Mobile Alliance Service Environment' (OSE), definido pela 'Open Mobile Alliance' (OMA), é uma concretização de um SDP que permite o desenvolvimento e execução de 'enablers' OMA que fornecerão os tão necessários serviços preenchendo assim aquele vazio existente no IMS. A aplicação de muitos destes conceitos está em decurso em muitas empresas de telecomunicações que tentam aproveitar logo de inicio as novas potencialidades destas arquitecturas começando principalmente pela exposição de serviços em interfaces, API's, em ambiente de Internet.

Esta dissertação tem como objecto de estudo a aplicação dos conceitos de 'Service Oriented Arquitecture' (SOA) através de um 'Service Delivery Platform' (SDP) nas redes 'IP Multimedia Subsystem' (IMS) sendo necessário o estudo detalhado da arquitectura IMS e dos protocolos que a compõem e de que modo o IMS poderá realizar os conceitos de SOA. Das plataformas SDP que aplicam os conceitos de SOA, o OSE prevê uma interligação com o IMS através do 'enabler' 'IMSinOMA' e através da pesquisa e analise teórica das especificações do IMS e do OSE espera-se mostrar conceptualmente como ambas as plataformas poderão cooperar num ambiente conjunto onde as características de convergência da arquitectura IMS se poderão aliar aos conceitos de SOA e SDP permitindo assim com que a tradicional inflexibilidade das redes baseadas em silos possa ser minimizada ou até abolida resultando num fácil desenvolvimento de serviços compostos baseados na reutilização e interoperabilidade com outros serviços já existentes. Para isso no capitulo seguinte, capítulo 2, começar-se-á com o estudo dos principais protocolos que constituem a arquitectura IMS definindo de um modo geral o seu funcionamento e alguns detalhes protocolares, seguir-se-á o estudo da arquitectura 3GPP IMS, nuclear e serviços, as entidades que a compõem e suas interligações, os aspectos de segurança, facturação e qualidade-de-serviço, 'Quality of Service' (QoS), terminando com a comparação entre o IMS estandardizado e especificado pelo consórcio 3GPP e o IMS análogo do consórcio 3GPP2. No capítulo 3 são esclarecidas as definições de SOA e SDP, os conceitos e aplicabilidade de SOA, os benefícios de uma plataforma SDP e sua arquitectura geral seguido do estudo da plataforma OSE onde será explicada a sua arquitectura. Segue-se a análise geral de algumas interfaces resultantes dos esforços de empresas de telecomunicações para aplicar estes conceitos expondo serviços sob a forma

de API's permitindo com que terceiros possam utiliza-los em aplicações personalizadas fornecendo novos serviços e novas potencialidades aos seus assinantes. Finalmente estudar-se-á a nível conceptual e lógico a aplicação destes conceitos numa rede IMS interligando-a com a plataforma OSE onde recursos poderão ser partilhados por ambas num ambiente cooperativo e as consequências que isso trará.

2. IMS

2.1. Introdução

A arquitectura IMS afirma-se como a melhor escolha para a convergência de redes fixas e móveis. Com os requisitos de convergência, interoperabilidade, controle de serviços e negociação de sessões multimédia, com qualidade-de-serviço e suporte para 'roaming', realiza o paradigma de Internet móvel criando uma plataforma comum que fornece o controlo de sessões multimédia, suporte para a implementação de serviços sem especificar como estes deverão ser implementados, disponibilizando aos operadores capacidades que lhes permitirão construir e disponibilizar serviços de valor acrescentado e interligação com as redes tradicionais de comutação de circuitos minimizando o impacto da transição entre as redes tradicionais de comutação de circuitos e as redes de comutação de pacotes. Para além disto a capacidade do IMS de ser agnóstico à rede de acesso do dispositivo móvel age como um factor decisivo na escolha desta arquitectura pelas operadoras como base nas suas redes de nova geração. Deste modo as operadoras de telecomunicações têm um ponto de partida para aumentar as margens de lucro pelo aumento do retorno médio por assinante causado pela utilização extra dos novos serviços. Como se pode ver o IMS é uma tecnologia que representa mais um passo na evolução das redes permitindo a coexistência com as redes existentes deixando um espaço para que possam coexistir futuras tecnologias e simultaneamente tentar cumprir a promessa de fornecer quaisquer serviços a qualquer hora em qualquer lugar. [66] [1] [2] [8]

A descrição do IMS neste capítulo seguirá a arquitectura definida pelo consórcio '3rd Generation Partnership Project' (3GPP) mas não seguirá nenhuma versão específica resultante dos processos de estandardização, em vez disso falar-se-ão de características que são gerais às últimas versões tentando assim não comprometer toda a arquitectura com base em detalhes de especificação que podem, ou não, ser alterados numa próxima versão. Então neste capitulo falar-se-á dos princípios gerais da arquitectura IMS e alguns dos seus conceitos, dos protocolos utilizados para a suportar, entidades que a compõem e como foram resolvidas as preocupações de segurança, facturação e qualidade-de-serviço,

adicionalmente comparar-se-ão as redes IMS estandardizadas pelos 2 principais consórcios, 3GPP e 3GPP2 observando as suas principais diferenças.

2.2. Princípios Gerais

O IMS baseia-se numa rede IP cujas entidades cooperam de modo a permitir o controlo de sessões multimédia, essa funcionalidade serve de plataforma para que serviços possam ser disponibilizados a utilizadores sem necessidade de estandardizar detalhes de implementação. Para além disso o IMS fornece independência da rede de acesso significando que qualquer dispositivo poderá aceder aos serviços numa rede IMS desde que a sua rede de acesso forneça uma conectividade IP, inicialmente foi definido exclusivamente o IPv6 devido à sua imensa capacidade de endereçamento embora algumas implementações possam utilizar o IPv4 tendo que conter estruturas de interoperabilidade. Como se baseia em IP foram seleccionados protocolos já existentes e estandardizados, que serão apresentados mais à frente, para suportar as funcionalidades necessárias à concretização de alguns dos requisitos definidos inicialmente para o IMS: suporte de estabelecimento de sessões IP multimédia, mecanismos de negociação de qualidade-de-serviço e interoperabilidade com a Internet e redes de comutação de circuitos. No IMS são introduzidos conceitos, herdados das redes sem fios, de rede natal, rede do operador do utilizador onde está aprovisionada a sua subscrição, e rede visitada que é uma rede de outro operador que poderá fornecer ao utilizador serviços de 'roaming' para a sua rede natal. Também na identidade do utilizador é introduzido um conceito de identidade que consiste no agrupamento de identidades públicas e identidades privadas, um utilizador poderá ter-lhe atribuído pelo operador uma ou mais identidades públicas, em formato URI SIP ou TEL URI que serão utilizadas para encaminhar a sinalização SIP e tornar o utilizador contactável, também poderá ser atribuída uma ou mais identidades privadas a um utilizador mas estas identidades, que poderão ser em formato 'Network Access Identifier' (NAI), serão utilizadas apenas para identificação da subscrição do utilizador e sua autenticação. Cada utilizador poderá ter associado a cada identidade privada uma ou mais identidades

públicas podendo estas ser partilhadas entre diferentes identidades privadas. Em questão de identidades também é introduzida a identidade 'Public Service Identities' (PSI), com formato de um SIP URI ou um TEL URI, utilizada para identificar um serviço fornecido por um servidor de aplicações. O 3GPP também definiu uma aplicação a ser contida no 'smartcard' presente em dispositivos móveis, o 'Universal Subscriber Identity Module' (USIM), que contém todos os dados necessários à identificação da subscrição do utilizador e sua autenticação. A nível de segurança os mecanismos de autenticação no IMS fornecem dupla autenticação, autenticação da rede pelo utilizador e do utilizador pela rede, e a ligação entre o dispositivo móvel e o primeiro ponto de contacto da rede IMS utilizará procedimentos de protecção de dados, a nível de facturação estão especificados dois mecanismos, 'online' e 'offline' que permitem a facturação instantânea dos créditos gastos nos serviços utilizados, comum em assinaturas pré-pagas, ou a facturação dos créditos após a utilização dos serviços, utilizada em assinaturas de pagamentos periódicos. [8] [1] [2]

Em seguida descrevem-se os principais protocolos seleccionados para suportar a sinalização e transporte de dados na arquitectura IMS.

2.3. Protocolos

Durante o início do desenvolvimento das redes IMS o consórcio 3GPP teve que decidir entre a criação de novos protocolos e a utilização de protocolos já existentes. Como a rede IMS é baseada em IP fez todo o sentido a utilização de protocolos já existentes aproveitando a experiência das entidades que os estandardizaram e reduzindo assim o tempo e custos associados com a criação de novos protocolos que afinal iriam executar as mesmas funções que poderiam ser fornecidas por protocolos já estandardizados. Foram escolhidos vários protocolos para o IMS para executar diferentes funcionalidades que faziam parte dos requisitos da nova arquitectura de rede. Para o controlo de sessões foi seleccionado o protocolo SIP devido a um conjunto de características chave muito importantes para IMS: a facilidade de estender funcionalidades a partir das especificações base, o modelo cliente-servidor, derivado de protocolos como o SMTP e HTTP que

também permite com que ferramentas de desenvolvimento e utilização destas tecnologias, especialmente HTTP, possam ser reutilizadas para o SIP, e de se apresentar em modo texto sendo, desse modo, mais legível facilitando a sua compreensão, extensão e depuração. Para os processos de AAA foi seleccionado o protocolo 'Diameter' pois também permite a sua extensão, neste caso pela criação de aplicações 'Diameter' que estendam as funcionalidades do protocolo base, tendo sido criadas duas aplicações: uma para que no IMS se possa haver interacção com o SIP em situações de criação e configuração de sessões e outra para executar o controlo de crédito e facturação. Outros protocolos também importantes seleccionados para serem utilizados no IMS foram o COPS para a transmissão de políticas entre pontos de decisão e pontos de aplicação de políticas, 'Policy Decision Points' (PDP's) e 'Policy Enforcement Points' (PEP's), o H.248, também conhecido como MEGACO, para a sinalização ao nível dos dados áudio/vídeo e o RTP juntamente com o RTCP para o transporte dos fluxos de áudio/vídeo. A decisão de utilizar protocolos já estandardizados foi muito importante pois a utilização de tecnologias já provadas e largamente testadas permitiu a desenvolvimento mais rápido da arquitectura e uma melhor compreensão desta pela comunidade. [1]

Nos pontos seguintes descrevem-se estes protocolos, suas especificações e alguns detalhes extra das suas aplicabilidades e modos de utilização no IMS.

2.3.1. SIP

SIP (Session Initiation Protocol) é um protocolo da camada de aplicação (modelo OSI) que permite o estabelecimento, modificação e terminação de sessões multimédia. [58]

O SIP não foi desenhado de raiz e as suas ideias básicas, que partem do conceito de telefonia aplicado a redes IP, já vêm desde 1992 quando a análise de conferências 'multicast' estava em decurso e em 1996 já existiam 2 protocolos do IETF propostos para a mesma função, o SIPv1 (Session Invitation Protocol) e o SCIP (Simple Conference Invitation Protocol), foi a junção das melhores características de cada um que deu origem ao SIPv2 que nessa altura foi submetido como 'draft' e posteriormente, em 1999, foi

publicado no RFC2543 definindo as suas funcionalidades básicas que foram, desde então, sendo expandidas com a publicação de vários RFC's. [5] [7] [8]

Foi principalmente devido à sua expansibilidade e independência da rede de acesso que foi escolhido como protocolo de criação e gestão de sessões multimédia no IMS pois assim não limitará o futuro desenvolvimento da arquitectura. À medida que forem encontradas falhas ou a necessidade de novas funcionalidades que impliquem alterações no SIP, este assim o permitirá. Para além disso a sua capacidade de anexar quaisquer dados de tamanho reduzido a uma mensagem SIP e de ser um protocolo baseado em texto como o http também tiveram grande importância na escolha do SIP para o IMS, a anexação de dados potencia ainda mais as capacidades do SIP como protocolo e do IMS como arquitectura não só para a negociação como para o envio de dados de tamanho reduzido e a utilização de texto tal como no HTTP permite melhor compreensão do protocolo e dos dados que estão a ser trocados numa negociação permitindo assim o melhor desenvolvimento e despistagem de erros. [2]

Apesar de suportar o estabelecimento de sessões numa rede IP, foi publicado em Maio de 2005 o RFC4083 expandindo o SIP para suportar muitas das características intrínsecas da rede IMS referentes ao '3GPP Release 5'e poderão ser publicados outros RFC's definindo requisitos adicionais. [2] [60]

É por isso então o protocolo principal de sinalização do IMS.

Camadas Protocolares no SIP

O SIP divide-se em várias camadas protocolares que permitem com que diferentes módulos possam funcionar de modo independente e com fraco acoplamento.

Existem 4 camadas, começando com as de mais baixo nível, a camada de sintaxe e codificação, a camada de transporte, a camada de transacção e a camada de utilização de transacção.

A camada de sintaxe e codificação engloba as características da sintaxe e gramática do SIP que seguem uma forma melhorada da gramática Backus-Naur Form (BNF) definida na generalidade no RFC2234 e em particular no SIP no RFC3261. [62] [8]

A camada de transporte é responsável pela transmissão de pedidos e respostas em protocolos de transporte pela rede e pela gestão de ligações persistentes. Faz identificação da ligação a utilizar para o caso da utilização de um transporte 'orientado à ligação' e um mapeamento de tuplos contendo IP, porto e protocolo. [62] [8]

A camada de transacção é responsável pelo agrupamento de cada resposta a cada pedido. Utiliza a camada de transporte para enviar e receber as várias mensagens SIP e lida com as retransmissões e 'timeouts' (expirações por excesso de tempo) que são ambos dependentes do protocolo utilizado. Uma transacção compreende tipicamente de um pedido enviado por um cliente para um servidor juntamente com todas as respostas enviadas por esse servidor de volta ao cliente.

Esta camada consiste em 4 máquinas de estado ligadas à transacção, as transacções de cliente para 'INVITE' (convite) e para não-'INVITE' (não-convite) e as transacções de servidor para 'INVITE' (convite) e para não-'INVITE' (não-convite), cada uma tem diferentes definições de temporizadores e regras de retransmissão de pacotes e terminação de ligações. [8] [62]

A camada de utilização de transacção é responsável pela criação de transacções no cliente e no servidor. Instancia uma transacção de cliente com IP de destino, porto e protocolo de transporte quando é necessário enviar um pedido SIP. Pode ser definida como servidor ou como cliente tendo comportamentos diferentes para cada caso. Todas as entidades SIP, à excepção de um 'stateless proxy', são utilizadoras de transacções podendo cada uma cancelar qualquer transacção criada por si através de um pedido 'CANCEL' que por si só é uma transacção mas referencia a transacção a cancelar. [8] [62]

Arquitectura SIP

A arquitectura SIP compreende dois tipos de entidades, entidades utilizadoras referidas de 'User Agents' (UA) e entidades intermediárias, servidores.

Um UA é um ponto de terminação de um diálogo, que envia e recebe pedidos e respostas SIP, e de fluxos multimédia. Na maioria dos casos, embora não exclusivamente, um UA corresponde ao equipamento do utilizador. Um UA é constituído por duas entidades lógicas, um cliente, 'User Agent Client' (UAC), que inicia um pedido e o UA agirá como tal durante toda a transacção referente a esse pedido e um servidor, 'User Agent Server' (UAS), que recebe um pedido e gera uma resposta a esse pedido, tal como no UAC o UA agirá como um UAS durante toda a transacção referente ao pedido recebido. A utilização destas entidades lógicas do UA é transitável de transacção para transacção, o UA que age como UAC ao iniciar uma chamada agirá como UAS ao receber um pedido 'BYE' do destinatário ao terminar a chamada. [8] [62]

As entidades intermediárias dividem-se em:

Servidores Proxy que recebem e reencaminham mensagens SIP. Podem interpretar e alterar partes da mensagem que não interfiram com o estado do diálogo, incluindo o corpo da mensagem. Um pedido recebido pode ser enviado para várias localizações simultaneamente de modo paralelo ou sequencial num processo chamado de 'forking'. Existem 3 variantes de servidores proxy no SIP, o 'Dialog-Statefull Proxy' que retém o estado do diálogo desde o pedido iniciador até ao terminador, o 'Transaction-Statefull Proxy' que instancia uma máquina-de-estado transaccional durante o processamento de um pedido e o 'Stateless Proxy' que simplesmente reencaminha todos os pedidos e respostas recebidas. [8]

Servidores de redireccionamento que associam os endereços nos pedidos para os endereços destino correctos permitindo o redireccionamento dos pedidos embora não participe na transacção. [8]

Servidores de localização que armazenam e mantêm actualizada a localização actual de cada utilizador. [8]

Servidores de Registro, 'Registrar', que receberão pedidos 'REGISTER' e após registro bem sucedido armazena uma ligação explícita entre o endereço SIP e o endereço da máquina onde se localiza o utilizador ou onde este deseja receber futuros pedidos. [8]

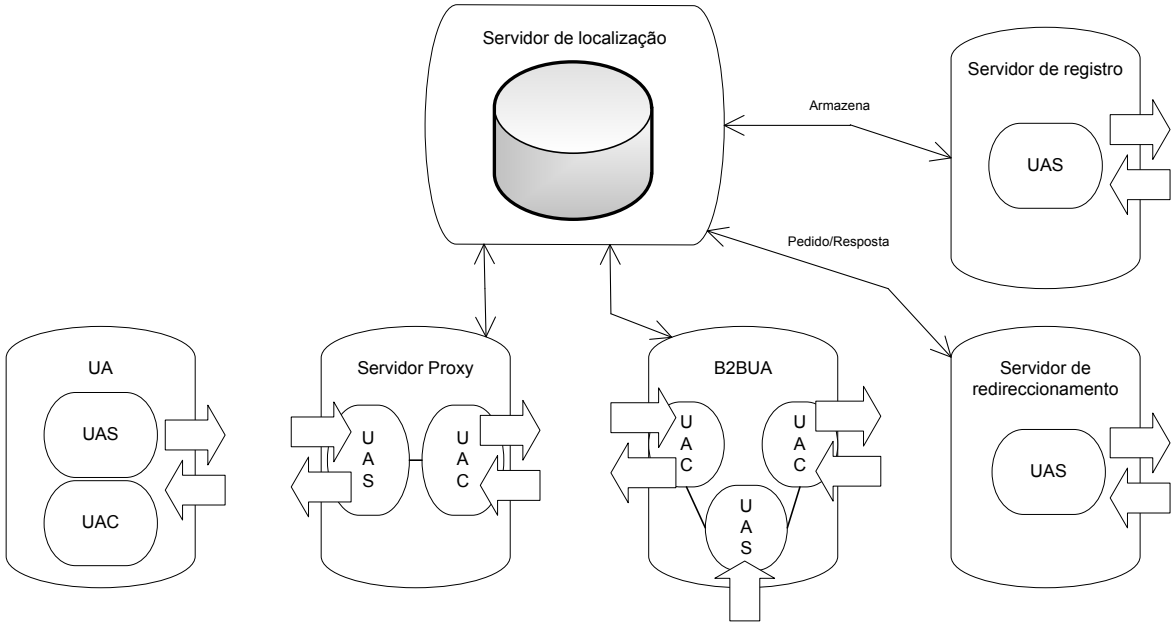

Figura 2.1 – Entidades SIP [11]

Para o fornecimento de serviços aos utilizadores existem duas entidades, o 'Aplication Server' (AS) e o 'Back-to-Back User Agent' (B2BUA). O AS é um servidor onde serviços podem ser lançados, os mais comuns são serviços de presença e conferência e o B2BUA que é semelhante a um 'proxy server' mas mantém um estado do diálogo e terá de agir em todos os diálogos estabelecidos por si dos pedidos recebidos, consiste numa fusão de um UAC e um UAS pois um pedido recebido por si é tratado do mesmo modo que um UAS mas para determinar a resposta esse pedido é alterado criando-se um outro pedido relacionado que é enviado segundo o comportamento típico de um UAC.

Numa rede IMS as entidades intermediárias SIP correspondem às entidades CSCF-x e HSS e para fornecer serviços aos utilizadores ambas as entidades de fornecimento de serviços estão presentes. [8] [62]

Mensagens SIP

As mensagens SIP, que podem ser um pedido ou uma resposta, seguem o formato básico definido no RFC2822 mas com uma sintaxe e um conjunto de caracteres diferentes. Cada mensagem está estruturada em 3 secções, a linha de início, os cabeçalhos da mensagem e o corpo da mensagem. A linha de início é a primeira linha da mensagem e contém o método da mensagem, a versão SIP a que corresponde e o URI do destinatário no caso de um pedido, no caso de uma resposta contém o código de estado e a versão SIP. Os cabeçalhos da mensagem estão entre a linha de início e uma linha vazia que marca o fim dos cabeçalhos, usualmente depois do campo 'CONTENT-LENGTH', e o corpo da mensagem, cujo tamanho é definido no campo atrás referido, apresenta-se imediatamente a seguir à linha vazia que estará presente quer esteja presente ou não um corpo da mensagem. [2] [62]

```
Session Initiation Protocol (SIP as raw text)
  REGISTER sip:open-ims.test;transport=udp SIP/2.0\r\n
  Call-ID: 3f8b2a28c0abba2bc9439c00f336fb19@172.16.2.128\r\n
  CSeq: 1 REGISTER\r\n
  From: <sip:bob@open-ims.test>;tag=123abc\r\n
  To: <sip:bob@open-ims.test>;tag=123abc\r\n
  Via: SIP/2.0/UDP 172.16.2.128:5060;branch=UserAgent\r\n
  Max-Forwards: 70\r\n
  Contact: <sip:172.16.2.128:5060>\r\n
  Authorization: algorithm=MD5 username="bob",realm="open-ims.test",uri="sip:open-ims.test;transport=udp"\r\n
  P-Access-Network-Info: 3GPP; cell-id=CE34DF\r\n
  Security-Client: ipsec-3gpp; alg=hmac-sha-1-96;spi-s=23456789;port-s=1301;spi-c=12345678;port-c=2501\r\n
  Require: sec-agree\r\n
  Proxy-Require: sec-agree\r\n
  Supported: path\r\n
  Content-Length: 0\r\n
  \r\n
```

Figura 2.2 – Exemplo de mensagem SIP

Endereços SIP

Existem dois tipos de endereços utilizados no SIP, os endereços SIP URI e os endereços de telefone URI (TEL URI). Os endereços SIP URI que foram introduzidos inicialmente no RFC2543 seguem o mesmo formato que um endereço de correio electrónico com a adição de 'sip:' a preceder o endereço. Mais tarde foi introduzida no RFC3261 uma variação do

SIP URI com segurança, que utiliza TLS numa ligação TCP, ficando SIPS URI e tem o mesmo formato que o SIP URI mas é adicionado 'sips:'. [8]

Um endereço SIP(S) URI tem duas variantes, uma é endereço de registo (Address Of Record (AOR)) que é o endereço identificador de um utilizador e que necessita de ser resolvido outra é endereço totalmente qualificado (Fully Qualified Domain Name (FQDN)) que identifica directamente o dispositivo no qual o utilizador está conectado não necessitando de resolução. [8]

Um endereço de telefone URI (TEL URI), que foi definido no RFC3966, identifica um número de telefone num URI, deste modo pedidos podem ser enviados para números de telefone e pode ser utilizado no SIP tal como um SIP(S) URI. [8]

Um URI SIP(S) apresenta-se com a forma:

'sip(s):nome_de_utilizador_ou_telefone@dominio[parâmetros][cabeçalhos]'

Dos quais o campo 'parâmetros' conterá parâmetros da ligação, tais como transporte e tempo de vida, e o campo 'cabeçalhos' poderá conter qualquer informação extra que poderá ser necessária. [8]

E um TEL URI tem a forma:

'tel:número_de_telefone_global' ou 'tel:número_de_telefone_local;parâmetros'

O campo parâmetros poderá conter múltiplos campos de contexto 'phone-context' separados por ';' contextualizando o número local tornando-o globalmente único.[63] [8]

Diálogos e sessões SIP

Um diálogo, que é uma relação que se estabelece entre 2 elementos que trocam mensagens, é constituído de vários estados que permitem com que as mensagens enviadas possam ser encaminhadas e compreendidas dentro do contexto adequado. Cada diálogo tem um identificador, 'Dialog-ID', que consiste num identificador de chamada, 'Call-ID', e de

duas etiquetas, ‘tags’, uma local e uma remota. As etiquetas, remota e local, são anexadas como parâmetros nos campos de cabeçalho ‘From:’ e ‘To:’ respectivamente para o UAS e vice-versa para o UAC do diálogo. Por sua vez um estado consiste de um ‘Dialog-ID’, 2 números de sequência, um local e um remoto, 2 URI’s, local e remoto, um destino remoto, uma variável Booleana com o nome de ‘secured’ e um caminho.

Uma sessão multimédia é um conjunto de fluxos de dados, juntamente com as suas origens e destinos, que utiliza diálogos. Uma sessão é iniciada através de um pedido ‘INVITE’ e da sua negociação bem sucedida e termina após um pedido ‘BYE’.

Pedidos SIP

Um pedido SIP contém na linha inicial o método da mensagem que informa o destinatário, também presente no URI SIP dessa mesma linha, e as entidades intermédias da acção que está a ser pedida. Um pedido não é nada mais do que uma chamada para estabelecer uma ligação virtual entre dois pontos, vulgo diálogo. Um diálogo ficará estabelecido assim que a negociação dos parâmetros seja bem sucedida e terminada com uma aceitação final, ‘acknowledgement’. A linha inicial terá o seguinte formato:

MÉTODO {espaço} URI SIP {espaço} VERSÃO SIP {crlf}

O método informa a acção que está a ser requisitada, o URI SIP é o endereço do destinatário da mensagem e a versão SIP informa a versão do protocolo SIP utilizada. [2]

Existem sete métodos diferentes na especificação nuclear do SIP definida no RFC3261 que são: ‘ACK’, ‘BYE’, ‘CANCEL’, ‘INFO’, ‘INVITE’, ‘OPTIONS’ e ‘REGISTER’ e para o IMS são utilizados também os métodos ‘MESSAGE’, ‘NOTIFY’, ‘PRACK’, ‘PUBLISH’, ‘REFER’, ‘SUBSCRIBE’ e ‘UPDATE’ que foram definidos como extensões do SIP.

O método ‘ACK’ é o passo inicial de um ‘handshake’ entre duas entidades que é necessário para estabelecer o diálogo e iniciar a sessão, reconhece assim a recepção de uma resposta final a um pedido. [2] [1]

O método ‘BYE’ termina uma sessão decorrente, podendo ser enviado por qualquer dos participantes da sessão. A sessão ficará terminada imediatamente após a recepção de um ‘ACK’ confirmando a terminação da sessão. [2]

O método ‘CANCEL’ cancela um pedido pendente, ou seja, um pedido para o qual ainda não foi recebida uma resposta do destinatário. [2] [1]

O método ‘INFO’ permite o envio de informação entre os terminais durante uma sessão. A informação pode ser muito variada desde parâmetros de sinalização, dígitos que foram marcados num terminal até ficheiros de dados pois a especificação deste método não define de modo exacto a sua utilização. [2]

O método ‘INVITE’ permite o estabelecimento de uma sessão entre dispositivos permitindo a negociação dos seus parâmetros. [2]

O método ‘OPTIONS’ permite a interrogação de uma entidade de modo a averiguar as suas capacidades de modo a que se possa saber que serviços se poderão utilizar antes de se tentar estabelecer uma sessão. Na resposta a este método se receberá um cabeçalho ‘ACCEPT’ informando os métodos suportados e outras capacidades relevantes. [2]

O método ‘REGISTER’ permite o emparelhamento de um URI público com a localização actual do utilizador. O seu objectivo principal é a notificação da rede da localização actual do dispositivo através do seu endereço de modo a poder encaminhar-lhe mensagens a si destinadas. Devido às características intrínsecas da mobilidade um só endereço da localização do dispositivo não será suficiente, deste modo uma mensagem destas trará credenciais únicas do dispositivo. [2]

O método ‘MESSAGE’ foi criado pelo 3GPP para permitir o envio de mensagens sem que seja necessário utilizar o método ‘INVITE’ e, desse modo, facilitar serviços de mensagens de texto e serviços de mensagens instantâneas. A informação é enviada no corpo do pedido SIP. [2]

O método ‘NOTIFY’ é utilizado, numa rede IMS, pelo S-CSCF para notificar os AS’s, que se tenham subscrito à notificação de eventos, de um ou vários utilizadores, que ocorreu um evento. Essa notificação, que poderá conter detalhes sobre o evento, terá que ser requisitada com um pedido ‘REQUEST’. [2] [84]

O método ‘PRACK’ cujo nome vem de ‘Provisional Response ACKnowledge’ foi criado para permitir o suporte de uma transmissão fiável de respostas provisionais. Uma entidade para se certificar que uma mensagem chegou ao destino enviará um ‘PRACK’ para o qual receberá uma resposta. [2] [105]

O método ‘PUBLISH’ permite a transferência de informação para um servidor. Um pedido ‘PUBLISH’ pode criar, modificar ou remover um estado associado com um endereço-de-registo numa entidade gestora. [1] [85]

O método ‘REFER’ instrui o destinatário a contactar uma terceira entidade cujo contacto se encontra na informação de contacto presente no pedido. [86]

O método ‘SUBSCRIBE’ permite com que se obtenha uma notificação sobre o estado actual e consequentes notificações de actualização, acerca de um evento específico a partir de uma entidade notificadora. Após a subscrição a entidade notificadora enviará notificações sempre que o estado se altere até que a subscrição expire ou seja cancelada. No IMS é utilizada pelos AS’s para pedir actualizações do S-CSCF e do HSS das alterações do estado de registro de um subscritor. [1] [84] [2]

O método ‘UPDATE’ permite a alteração de características de uma sessão que está a decorrer ou cuja negociação ainda não completou sem ter impacto no estado do dialogo entre os participantes. Sem este método o fluxo de mensagens numa negociação seria inconsistente. [87] [2]

Respostas SIP

O modelo utilizado nas respostas SIP foi baseado no HTTP onde se utilizam códigos numéricos para reportar um estado. Muitos dos códigos utilizados são iguais ao HTTP e outros foram adicionados especificamente para o SIP. [5]

Estas diferem de um pedido SIP principalmente nos campos da linha inicial que numa resposta SIP são 3, a versão do SIP utilizado, o código numérico com o estado definindo o resultado do pedido e uma frase descrevendo o estado ou a razão do estado. [8]

Formato da linha de inicio:

VERSÃO SIP {espaço} CÓDIGO SIP {espaço} DESCRIÇÃO {crlf}

Cada resposta pertence a 1 de 6 grupos de classificação, provisório, sucesso, redireccionamento, falha no cliente, falha no servidor e falha global que se identificam pelo primeiro dígito do código SIP que a acompanha, os restantes dois dígitos especificam a resposta dada. [2]

O grupo provisório, com o número 1xx, representa eventos relacionados com o envio de pedidos informando da correcta recepção de um pedido e do seu consequente processamento por nós intermédios ou destinatário. As respostas deste grupo evitam na generalidade que o pedido inicial seja reenviado continuamente até que se receba uma resposta final e evitando assim tráfego desnecessário. As respostas de sucesso utilizam números na gama 2xx, este grupo representa a recepção correcta e aceitação do pedido pelo destinatário. As respostas pertencentes a este grupo são enviadas pelo destinatário do pedido inicial. As respostas de redireccionamento utilizam números na gama 3xx e notificam que o pedido enviado poderá ou terá de ser reenviado para outro destino pelo que o elemento originário do pedido terá que tomar uma acção. Uma resposta utilizando números na gama 4xx refere-se a uma falha no cliente, indica que o pedido enviado não foi compreendido correctamente, provavelmente por um erro de sintaxe, ou não pode ser aceite por um elemento intermédio ou pelo seu destinatário. No caso da falha ser no elemento que está a actuar como servidor no diálogo o erro terá números na gama de 5xx. Para todos os restantes casos não cobertos pelos grupos anteriores há o grupo de falha global que utiliza números na gama 6xx e geralmente indica que o pedido não pôde ser executado em nenhum servidor, provavelmente devido à inacessibilidade do destinatário ou então devido à incapacidade de suportar a sessão. [2] [8]

É claro que só um número indicando a resposta não é suficiente. Uma resposta também conterá parâmetros presentes nos cabeçalhos, em alguns casos também no corpo da mensagem, que fornecerão mais detalhes. [2] [8]

Elementos de cabeçalho

O cabeçalho de uma mensagem situa-se a seguir à linha de início de uma mensagem e detalha não só a informação enviada, relacionada com o pedido ou resposta, mas também as características do corpo da mensagem em elementos individuais que seguem o formato:

nome_do_elemento ':' valor_do_elemento [8] [62]

Existem elementos que têm de estar obrigatoriamente presentes numa mensagem que são os campos 'To:', 'From:', 'Call-ID:', 'CSeq:', 'Via:', 'Max-Forwards:' e o 'Contact:', dos quais o 'Contact:' apenas é obrigatório para pedidos que iniciam um diálogo. [8] [2]

Para o funcionamento do SIP no IMS foram adicionados alguns elementos de cabeçalho, os mais importantes foram os iniciados por 'P-', que estão incluídos na seguinte lista podendo se encontrar a maioria deles definidos no RFC3261.Outros foram definidos em RFC's próprios.

Accept	CSeq	P-Associated-URI	Require
Accept Encoding	Date	P-Called-Party-ID	Retry-After
Accept Language	Error-Info	P-Charging-Function-Address	Route
Alert Info	Event	P-Charging-Vector	Server
Allow	Expires	P-Preferred-Identity	Subject
Authorization	From	P-Visited-Network-ID	Supported
Call-ID	In-Reply-To	Privacy	Timestamp
Call-Info	Max-Forwards	Priority	To
Contact	Min-Expires	Proxy-Authenticate	Unsupported
Content-Disposition	MIME Version	Proxy-Authorization	User-Agent
Content-Length	Organization	Proxy-Require	Via
Content-Transfer-Encoding	P-Access-Network-Info	Record-Route	Warning
Content-Type	P-Asserted-Identity	Reply-To	WWW-Authenticate

Tabela 2.1 – Elementos de cabeçalho SIP IMS.

Extensões SIP IMS

Segurança

De modo a proteger o SIP de ataques maliciosos mais comuns foi definido um método de negociar um mecanismo de segurança entre um UA e a próxima entidade SIP que lhe está ligada, 'next-hop', no RFC3329 onde são definidos 3 novos campos de cabeçalho, 'security-client, 'security-server' e 'security-verify' para permitir a negociação. No SIP também é utilizado o mecanismo AKA-Digest, que é um mecanismo de desafio-resposta, 'challenge-response', numa autenticação 'Digest' definido para o HTTP no RFC3310. [88] [89]

Reserva de recursos

Uma sessão só deverá ser estabelecida se a rede puder suportar o tráfego que os seus fluxos de dados irão gerar e com os requisitos de QoS necessários. Para isso terá que se certificar que não se estabelecerá uma sessão sem que existam as condições de largura de banda e qualidade de serviço necessárias.

Foi então definido no RFC3312 o conceito de pré-condições, que são um conjunto de características que têm de existir para que a sessão possa ser estabelecida, para uma negociação SIP em conjunto com SDP. As pré-condições são enviadas juntamente com a descrição da sessão no corpo da mensagem, assim ambos os participantes de uma negociação poderão estar informados das características actuais e das necessárias para estabelecer a sessão e não a estabelecerão sem que as pré-condições seja atingidas primeiro.

Mais tarde o RFC3312 foi actualizado no RFC4032 de modo a suportar novos tipos de pré-condições e situações de mobilidade onde pontos de uma sessão necessitam de transitar para outro local. [90] [91]

Autorização de media

Para o controlo de qualidade de serviço, através do controlo da utilização de recursos da rede, e assim também proteger a rede contra ataques de DoS numa rede administrada foi

definida uma extensão SIP que permite com que uma sessão possa ser autorizada por um 'proxy' SIP e pelo ponto de controlo de políticas que garante uma qualidade de serviço. Foi criado um novo campo de cabeçalho, 'P-Media-Authorization', que conterá 'tokens' de autorização fornecidos por uma entidade independente de reserva de recursos que deverão estar presentes durante a negociação da sessão. [92]

Compressão SigComp

Como o SIP é baseado em texto uma mensagem não terá um tamanho óptimo resultando em mensagens de pelo menos umas centenas de bytes. Embora uma mensagem nessa escala de tamanho não seja problemática numa utilização típica de uma rede cuja largura de banda seja elevada o mesmo não será verdade para redes com larguras de banda mais constrangidas como é o caso de redes celulares. O SigComp foi desenvolvido de modo a minimizar esse problema comprimindo as mensagens de texto geradas por protocolos baseados em texto tal como o SIP sem perda de informação. Apresenta-se como uma camada de abstracção que está situada entre a aplicação e a camada de transporte fornecendo à aplicação o serviço fornecido pela camada de transporte mais compressão suportando vários protocolos de transporte tais como TCP, UDP e SCTP. O mecanismo de sinalização da utilização de SigComp no SIP, para além das situações mais apropriadas é definido no RFC3486. [93] [94]

Cabeçalhos 'P'

No IMS foi necessária a adição de alguns campos de cabeçalho para permitir funcionalidades como facturação e transição de redes. Esses campos de cabeçalho são, 'P-Associated-URI'. 'P-Called-Party-ID', 'P-Visited-Network-ID', 'P-Access-Network-Info', 'P-Charging-Function-Addresses', 'P-Charging-Vector', 'P-Asserted-Identity' e 'P-Preferred-Identity'. Estes campos estão presentes atrás na tabela 2.1 da secção de cabeçalhos e foram definidos no RFC3455 e RFC3325.

Registro móvel

O IMS permite mobilidade entre várias redes o que causa com existam vários ‘proxies’ SIP entre o UA e elementos da sua rede natal, ‘Home Network’, o problema é que um pedido enviado para um UA da sua rede natal tem que passar por esses ‘proxies’ e não havia um método para armazenar esse caminho para futura utilização. Para isso foi definido um campo de cabeçalho ‘Path’ no RFC3327 que permite com que cada ‘proxy’ intermédio que necessita de estar no caminho se adicione a esse campo e desse modo informar o percurso da mensagem. O campo ‘Path’ é semelhante ao ‘Record-Route’ mas só é utilizado em pedidos ‘REGISTER’e consequentes respostas do grupo ‘2xx’ nos casos em que existem ‘proxies’ intermediários entre um UA e o seu ‘Registrar’, o caminho de ida de um pedido é exactamente igual ao caminho de volta e os ‘proxies’ intermediários não estão presentes em tabelas de encaminhamento, registos de DNS ou semelhantes mecanismos.

Também é necessário, para UA’s situados fora da rede natal, um ‘proxy’ de contacto para fornecer serviços que possam originar quer na rede natal quer no UA e para isso foi necessário um mecanismo que permita a sua descoberta e associação. Esse mecanismo, chamado de ‘Service Route’ foi definido no RFC3608 onde foi adicionado um campo de cabeçalho ao SIP como nome de ‘Service-Route’. Quando um ‘registrar’ responde com sucesso, resposta de grupo ‘2xx’, a um pedido de registro de um UA inclui no cabeçalho da resposta o campo ‘Service-Route’ com o caminho que o UA deverá utilizar nos seus pedidos. Este mecanismo é utilizado quando um UA se regista num ‘registrar’, o ‘registrar’ tem conhecimento do ‘proxy’ de serviço a utilizar pelo UA e da topologia de rede de modo a poder construir um caminho de serviço e utilizará esse caminho para todos os contactos associados com o mesmo AoR. Também esta extensão só será utilizada quando outros mecanismos de fornecimento de um caminho de serviço não estejam disponíveis ou sejam inapropriados. [95] [96]

2.3.2. SDP

SDP (Session Description Protocol) é um protocolo textual destinado à descrição de sessões multimédia que está ao nível da camada de aplicação e apesar de ser referido como um protocolo o SDP não passa de um formato de descrição se sessões multimédia que numa rede IP conterá, tipicamente, endereços IP e números dos portos onde a informação terá de ser enviada e os 'codecs' utilizados para a codificação de voz e/ou vídeo. Foi inicialmente definido no RFC2327 que foi depois tornado obsoleto com a publicação do RFC4566. No SIP uma mensagem SDP está presente no corpo de um pedido ou resposta descrito como 'application/sdp' e conterá linhas de atributos identificados em grupos de letra-valor na forma de '<tipo>=<valor>'.

Existem 3 tipos de categorias de informação numa mensagem SDP e terão de aparecer na seguinte ordem: descrição ao nível da sessão, descrição da temporização e descrição de média. [2] [1] [8] [97]

A descrição ao nível de sessão que incluirá um identificador da sessão e parâmetros da sessão tais como endereço IP e assunto tem os seguintes campos obrigatórios: [2] [1] [8] [97]

Campo	Descrição	Comentário
'v='	Versão de protocolo	
'o='	Originador e identificador de sessão	
's='	Nome da sessão	

Tabela 2.2 – Campos de sessão SDP obrigatórios.

E os seguintes opcionais:

‘i=’	Informação da sessão	
‘u=’	URI da descrição	
‘e=’	Endereço de e-mail	
‘p=’	Número de telefone	
‘c=’	Informação da ligação	Opcional se for incluído em todos os média
‘b=’	Informação sobre a largura de banda	(0 ou +)
‘z=’	Ajuste de fuso horário	
‘k=’	Chave de criptografia	
‘a=’	Atributos da sessão	(0 ou +)

Tabela 2.3 – Campos de sessão SDP opcionais.

A descrição da temporização contém tempos de início e paragem, tempos de repetição e pelo menos uma descrição de média utilizada.

Tem o campo obrigatório:

‘t=’	Duração da sessão	

Tabela 2.4 – Campos de temporização SDP obrigatórios.

E o campo opcional:

‘r=’	Número de repetições	(0 ou +)

Tabela 2.5 – Campos de temporização SDP opcionais.

A descrição de media, caso seja necessária, contém o protocolo de transporte, número dos portos e alguns parâmetros acerca da média com o seguinte campo obrigatório:

‘m=’	Nome da média e endereço	

Tabela 2.6 – Campos de media SDP obrigatórios.

E os opcionais:

‘i=’	Titulo da média	
‘c=’	Informação da ligação	Opcional se incluido ao nível de sessão
‘b=’	Informação da largura de banda	(0 ou +)
‘k=’	Chave de criptografia	
‘a=’	Atributos de média	(0 ou +)

Tabela 2.7 – Campos de media SDP opcionais.

A seguinte figura, figura 2.3, exemplifica uma mensagem SDP anexada ao corpo de uma mensagem SIP.

```
v=0
o=Bob 234562566 236376607 IN IP6 1080::8:800:200C:417A
s=Trabalho em curso
c=IN IP6 1080::8:800:200C:417A
t=0 0
m= video 8382 RTP/AVP 98 99
b=AS:75
a=curr:qos local none
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos none remote sendrecv
a=rtpmap:98 H263
a=fmtp:98 profile-level-id=0
a=rtpmap:99 MP4V-ES
m=audio 8283 RTP/AVP 97 96
b=AS:25.4
a=curr:qos local none
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos none remote sendrecv
a=rtpmap:97 AMR
a=fmtp:97 mode-set=0,2,5,7; maxframes=2
a=rtpmap:96 telephone-event
```

Figura 2.3 – Exemplo de uma mensagem SDP típica. [1]

A mensagem da figura 2.3 encontra-se dividida em duas partes, a descrição da sessão e a descrição dos fluxos media propostos.

Na primeira parte é identificada a versão do protocolo no campo ‘v’, no campo ‘o’ identifica-se a origem da sessão com o seu nome de utilizador (‘Bob’), o identificador da sessão (‘234562566’), a versão da descrição da sessão cujo valor inicial não é relevante e será incrementado sempre que uma alteração é feita (‘236376607’), o ‘IN’ identifica o tipo da rede utilizada (‘IN’ refere-se a Internet) seguido do protocolo e endereço, endereço 1080::8:800:200C:417A IPv6. No campo ‘s’ descreve-se o assunto da sessão que neste caso é ‘Trabalho em curso’, no campo ‘c’ são descritos os detalhes da ligação, neste caso a ligação é feita numa rede de tipo ‘Internet’ para o endereço IP versão 6 de valor 1080::8:800:200C:417A, e no campo ‘t’ identifica-se a temporização de inicio e fim da sessão, como a sessão deverá decorrer a partir do momento em que será recebida os seus valores estão a zero.

Na segunda parte, identificada pela primeira vez que o campo ‘m’ aparece existem 2 descrições de fluxos de media, ambos iniciados pelo campo ‘m’ que no primeiro caso define a media utilizada, ‘video’, o porto utilizado, ‘8382’, o protocolo de transporte, ‘RTP/AVP’ e a identificação dos formatos que poderão ser utilizados na sessão, ’98’ e ‘99’, o campo ‘b’ define a largura de banda necessária, ‘75’ que, neste caso, a interpretação do

seu valor está a cargo da aplicação, ‘Aplication Specific’ (AS). Após este campo são definidos outros atributos nos seguintes campos ‘a’ relacionados com o fluxo representado em ‘m’ que os precede imediatamente. Nesta situação estão representados os estados actuais e desejados a nível de qualidade-de-serviço (QoS) no modo de envio e recepção, ‘sendrecv’, sendo que a sessão apenas poderá se iniciar após as condições actuais se verificarem serem iguais ou melhores às pedidas e que um terminal após verificar que o seu estado de QoS se alterou para uma condição igual ou melhor à que tinha enviará uma actualização ao terminal oposto, e outros detalhes específicos dos ‘codecs’ utilizados. No segundo caso a media descrita é ‘audio’ sendo a sua descrição análoga à verificada com o fluxo vídeo. [2] [1] [8] [97]

2.3.3. Diameter

Diameter é um protocolo que fornece serviços de AAA ou seja autenticação, autorização e facturação ou contabilização (‘Accounting’). É baseado no protocolo ‘Remote Authentication Dial In User’ (RADIUS) definido no RFC2865 que fornece o mesmo tipo de serviços mas devido à sua falta de funcionalidades e baixa escalabilidade foi desenvolvido o Diameter. O Diameter divide-se em 2 partes: o protocolo base, definido no RFC3588, e as aplicações do Diameter que permitem com que se extenda o protocolo quando são necessárias novas funcionalidades. O protocolo base, que utiliza TCP e SCTP como protocolos de transporte, é responsável pela entrega das unidades de informação (‘data units’) do Diameter pela negociação das capacidades e gestão de erros disponibilizando assim funcionalidade às aplicações do Diameter. Na parte das aplicações do Diameter, cada aplicação define e fornece funções e unidades de informação personalizadas estendendo as funcionalidades do protocolo base para uma utilização mais específica. Como são as aplicações que fornecem serviços, terá de ser utilizada pelo menos uma aplicação em conjunto com o protocolo base. [106]

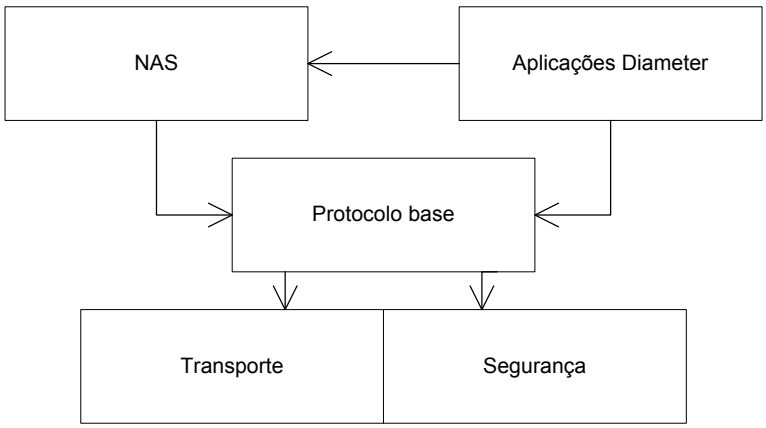

Figura 2.4 – Relações conceptuais no Diameter.

Na figura 2.4 podem-se observar os relacionamentos existentes entre os vários blocos conseptuais. O bloco 'Transporte' refere-se à necessidade de fiabilidade na entrega das mensagens, para isso uma ligação entre quaisquer dois nós utiliza TCP ou SCTP que terão de ser suportado pelo agente Diameter, um cliente Diameter poderá suportar apenas um. [106]

No bloco 'Protocolo Base' definem-se a maioria dos elementos constituintes do Diameter, atributos e estrutura das mensagens, iterações entre os vários agentes, conceitos associados com os protocolos de transporte utilizados e o conceito de aplicações Diameter. [106]

O bloco 'Segurança' refere-se à protecção da informação enviada contra terceiros. Utilizam-se protocolos de encriptação na camada de transporte, TLS [98], ou na camada de ligação, IPSec [99]. CMS [100] foi uma tentativa de aplicar segurança em porções seleccionadas das mensagens em situações onde intermediários teriam que inspeccionar apenas alguns campos. [106]

No bloco 'NAS' representa-se a aplicação 'Network Access Server' (NAS) [101]. Como a maioria das aplicações e serviços que lidam com AAA necessitam de utilizar um NAS, são reunidos nesta aplicação as funcionalidades e serviços comuns de modo a que cada aplicação não necessite de os implementar. [106]

As 'Aplicações Diameter' são protocolos ou serviços, cuja especificação é efectuada em documentação independente, que utilizam o protocolo Diameter ou servidores Diameter tendo de suportar as especificações presentes no protocolo base. A cada aplicação é atribuído um identificador pela IANA de modo a que os nós Diameter possam informar,

nas negociações, as aplicações que suportam e assim averiguar quais é que podem ser suportadas por todo o conjunto. As directrizes de criação de uma nova aplicação estão descritas no RFC3588. [106]

Devido à possibilidade de estender funcionalidades o Diameter é ideal para fornecer as funções de AAA no IMS pois poderá ser melhor adaptado às suas especificidades. Das aplicações Diameter existentes foram utilizadas algumas no IMS e foi criada outra para fornecer serviços mais específicos. [2] [1] [8] [102]

Entidades Diameter

O Diameter é um protocolo distribuído par-a-par, 'point-to-point' (P2P), onde cada nó pode enviar mensagens assincronamente a qualquer outro nó directamente ligado via TCP ou SCTP, vulgo nó vizinho ('peer').

Existem 3 tipos de nós Diameter: cliente, servidor e agente. Um cliente Diameter é uma entidade situada à entrada da rede que executa um controlo de acesso, tipicamente um 'Network Access Server' (NAS) que autentica os terminais conectados à rede para que possam utilizar recursos. Um servidor Diameter gere pedidos AAA num ambiente fechado ('realm'). E um agente Diameter que tem funções de balanceamento de carga, de administração e manutenção do sistema, processamento adicional de mensagens e pedidos, há 4 tipos de agentes Diameter dependendo da sua função: retransmissão ('relay'), reencaminhamento ('proxy'), redireccionamento ou tradução. Um agente Diameter de retransmissão reenvia mensagens Diameter baseado em tabelas de encaminhamento baseadas nos 'realms' e nós vizinhos conhecidos modificando apenas informação de encaminhamento numa mensagem, é agnóstico ao protocolo e à aplicação Diameter utilizada. Um agente Diameter 'proxy' faz encaminhamento como um agente de retransmissão mas também executa decisões baseadas em politicas de utilização de recursos e aprovisionamento, também pode alterar o conteúdo das mensagens para fazer reflectir essas decisões. Um agente Diamenter de redireccionamento resolve o nome do

'realm' a um servidor Diameter respondendo com uma mensagem de redireccionamento em vez de reencaminhar a mensagem. E um agente Diameter de tradução traduz mensagens protocolares entre o Diameter e outros protocolos de AAA permitindo a sua interacção. [8] [1] [106]

Mensagens Diameter

Estrutura das mensagens

Uma mensagem Diameter consiste de um cabeçalho seguido por um ou mais pares de atributo-valor, 'attribute value pair' (AVP), que conterão informação tipicamente referente a operações de AAA. [106] [1]

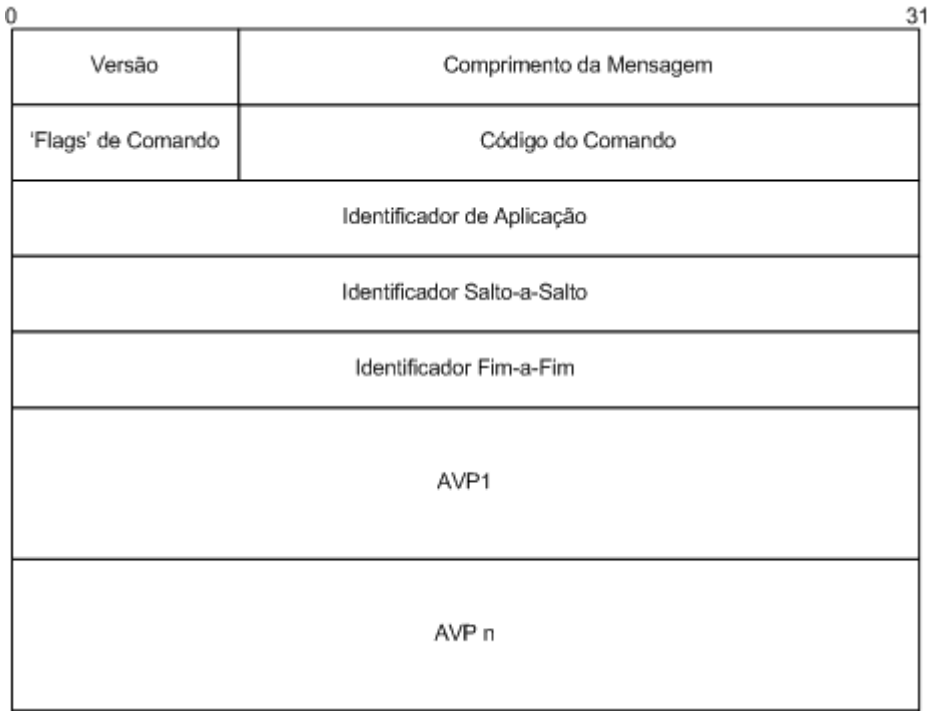

Figura 2.5 – Formato de uma mensagem Diameter com campos do cabeçalho. [1]

Dos campos presentes na imagem o campo ‘Versão’ indica a versão do protocolo Diameter a que se refere a mensagem, o campo ‘Tamanho da mensagem’ informa o tamanho total do cabeçalho e AVP’s, o campo ‘Flags de Comando’ informa se uma mensagem é um pedido ou uma resposta, se pode ser reenviada por um ‘proxy’, se é referente a um erro no protocolo e se é uma mensagem que foi retransmitida, o campo ‘Código do Comando’, que é utilizado em pedidos e em respostas, indica o comando do pedido ou o detalhe da resposta, o campo ‘Identificador de Aplicação’ identifica a aplicação Diameter para qual a mensagem é destinada, o campo ‘Identificador Salto-a-Salto’ (Salto-a-Salto = ‘Hop-by-Hop’) é utilizado para identificar a que pedido uma resposta corresponde ao nível da ligação entre dois nós vizinhos imediatos e o campo ‘Identificador Fim-a-Fim’ tem a mesma função só que ao nível da ligação entre originador e destinatário da mensagem. [106] [1] [8]

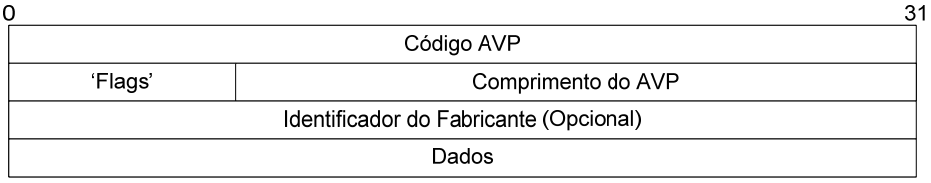

Figura 2.6 – Detalhe de um AVP. [1]

Um AVP contém informação sobre elementos de AAA e outros elementos relacionados com encaminhamento, segurança e outras informações que poderão ser relacionados com o comando do pedido ou resposta presente no cabeçalho. Como se pode observar na figura 2.6 os campos ‘Código AVP’ e ‘Identificador do Fabricante’ (Vendor-ID), que é opcional, identificam o atributo, o campo ‘Flags’ informa da necessidade ou não de garantir segurança ‘Fim-a-Fim’, se o suporte para este AVP é obrigatório ou opcional e se o campo do ‘Vendor-ID’ está presente ou não, o campo ‘Comprimento do AVP’ indica o comprimento de toda a informação presente no próprio AVP incluindo o campo de ‘Dados’ que pode ser enviado num número variado de formatos e contém informações específicas ao próprio AVP. [106] [1] [8]

Um AVP pode ser agrupado com uma sequência de outros AVP’s no seu campo de ‘Dados’.

Processamento de mensagens e erros

Uma mensagem é encaminhada de acordo com o seu identificador de acesso à rede, 'Network Access Identifier' (NAI) e o seu processamento segue um procedimento diferente dependendo do seu valor, correspondendo cada um a um agente Diameter diferente. Um procedimento pode ser: de tipo local onde uma mensagem é processada localmente não necessitando de ser enviada para outro vizinho imediato, de retransmissão em que a mensagem é enviada para o vizinho imediato identificado na tabela de encaminhamento sem qualquer modificação, de reencaminhamento em que a mensagem é enviada para o vizinho identificado na tabela de encaminhamento depois de terem sido aplicadas as politicas necessárias ou de redireccionamento onde uma mensagem é reenviada de volta para o vizinho originador para que possa ser enviada por este para outro local.

A tabela de vizinhos, que está presente em cada nó, contém as características de cada um estando cada entrada da tabela associada a uma identidade atribuída dinamicamente, com tempo de expiração, ou estaticamente. Uma tabela de vizinhos é referenciada por tabelas de encaminhamento baseado em 'realms' e cada entrada da tabela de encaminhamento é associada a um identificador da aplicação que lhe corresponde. Cada caminho pode ser estático ou dinâmico.

Existe para cada vizinho uma máquina de estados finita, 'Diameter Peer State Machine', que gere o estado das ligações entre os nós e os seus vizinhos. Todos os nós mantêm um estado de transacção, que pode ser adicionado a um pedido Diameter quando é necessário que um agente 'stateless' possa manter o estado actual da sessão, e de cada sessão que pode ser terminada explicitamente ou por expiração e que está associada a um serviço de uma aplicação. [8]

Os erros no Diameter dividem-se em erros de protocolo e erros de aplicação. Erros de protocolo correspondem a ocorrências ao nível do protocolo base onde a 'flag' de erro é assinalada no cabeçalho da mensagem de resposta e é anexado um AVP com o código correspondente ao erro que é tratado na próxima entidade a recebê-lo. Os erros de aplicação não necessitam do envolvimento dos agentes Diameter e por isso não é assinalada a 'flag' de erro no cabeçalho, apenas é anexado o AVP com o código e informação adequada ao erro que será tratado na entidade emissora do pedido inicial. [8]

Serviços

O protocolo base fornece às aplicações serviços de autenticação, autorização e contabilidade (‘accounting’) dos quais os serviços de autenticação e autorização podem ser utilizados em separado ou combinados podendo cada aplicação definir os seus próprios comandos e AVP’s para os utilizar, no caso dos serviços de contabilidade, que podem ser utilizados independentemente, a sua definição dos comandos, presente no protocolo base, é partilhada sendo apenas os AVP’s específicos a cada aplicação.

Comandos

Na seguinte figura, figura 2.8, apresentam-se os comandos definidos no protocolo base que fornecem as funcionalidades básicas do Diameter às aplicações que podem estendê-los e especificar para si novos comandos. Como se viu nas mensagens o código de um comando é igual para um pedido ou resposta e o que muda é a ‘flag’ que informa se é um pedido ou resposta.

Comando	Abreviação	Código
Abort-Session-(Request/Answer)	ASR/ASA	274
Accounting-(Request/Answer)	ACR/ACA	271
Capabilities-Exchange-(Request/Answer)	CER/CEA	275
Device-Watchdog-(Request/Answer)	DWR/DWA	280
Disconnect-Peer-(Request/Answer)	DPR/DPA	282
Re-Auth-(Request/Answer)	RAR/RAA	258
Session-Termination-(Request/Answer)	STR/STA	275

Tabela 2.8 – Comandos do protocolo base. [1]

Autenticação e autorização

O protocolo base fornece serviços de autorização em modo ‘stateless’ ou ‘statefull’ onde cada modo corresponde a uma máquina de estados finita diferente que tem que ser suportada pelos nós que fazem parte desses serviços. A diferença é que no modo ‘statefull’ o servidor mantém um estado da sessão de autorização que terá uma duração finita que por sua vez consiste no tempo de vida da autorização mais um tempo de “graça” ou poderá ser terminada a pedido do cliente, abortada pelo servidor ou re-autorizada no fim da sua duração. Ambos os serviços estão interligados pois um pedido de ‘Auth’ feito por um cliente fará uma autenticação, uma autorização ou ambos consoante os AVP’s presentes na mensagem. [8]

Contabilidade

Sem uma sessão de contabilidade activa não há recursos de rede reservados para ela quer no cliente ou servidor Diameter, só após um Accounting Request (ACR) bem sucedido é que uma sessão de contabilidade ficará activa na qual registros de contabilidade são trocados. Esses registos de contabilidade podem pertencer a serviços de comprimento mensurável, onde é criado um registo no início do período e outro quando termina ou podem ser criados registos em intervalos, ou a serviços pontuais, que representam serviços sem comprimento mensurável onde o início coincide com o fim sendo gerado apenas um registo de contabilidade. A decisão do tipo de registo a utilizar fica a cargo do servidor de contabilidade ou do servidor que autoriza a sessão do utilizador num procedimento chamado de “modelo de contabilidade orientado ao servidor” onde o servidor instrui o cliente a utilizar o método adequado à situação incluindo, caso seja necessário, o intervalo da criação de registos.

Para prevenir contra a perda de registos de contabilidade o protocolo de contabilidade do Diameter correlaciona registos de contabilidade com o AVP de identificação da sessão presente em todas as mensagens de AAA, para além disso também pode utilizar um AVP de identificação de sessões simultâneas se um serviço consiste num conjunto de sessões tendo cada uma um AVP de identificação de sessão diferente. [8]

Serviços no IMS

No 3GPP IMS foram utilizadas duas aplicações Diameter como referência para as aplicações das suas interfaces: a aplicação Diameter SIP, 'Diameter Session Initiation Protocol aplication', definida no RFC4740 de onde são baseadas as aplicações das interfaces Cx, Dx, Sh e Dh e a aplicação Diameter de controlo de crédito, 'Diameter Credit Control aplication', definida no RFC4006 que baseia a aplicação das interfaces Ro e Rf para fornecer a funcionalidade de facturação 'online' e 'offline', respectivamente. Os códigos dos comandos, 300 até 313, foram reservados, para a Rel.5 do 3GPP IMS, no RFC3589. Para além disso utiliza o SIP na interface ISC ('IMS Service Control'). [8]

As interfaces Cx e Dx implementam uma aplicação Diameter chamada 'Diameter aplication for the Cx interface' e está especificada no 3GPP TS 29.228/229. [1]

2.3.4. COPS

O protocolo 'Common Open Policy Service' (COPS), especificado no RFC2748, é utilizado para a administração, configuração e aplicação de politicas entre um ponto decisor de politicas, 'Policy Decision Point' (PDP), e um ponto de aplicação de politicas, 'Policy Enforcement Point' (PEP), transmitindo informação relacionada com as politicas de rede. As mensagens são trocadas em modo de pedido/resposta através de uma ligação TCP que é iniciada pelo PEP e pode ser protegida com IPSec ou TLS, sem essa protecção apenas poderá haver protecção de integridade que é suportada nas mensagens COPS. O COPS é extensível sendo que para suas extensões podem ser descritos formatos de mensagens e objectos de politicas, que são auto-identificáveis, sem necessitar de alterações no protocolo. Dos vários tipos de PEPs suportados, que são identificados pelo 'Client-Type' na mensagem, um exemplo comum é um router com suporte de 'ReSource reserVation Protocol' (RSVP) [103] para fornecer serviços de qualidade-de-serviço. [1] [8]

O COPS tem dois modelos de controlo de politicas: terceirização (‘outsourcing’), e aprovisionamento (configuração). [1]

No modelo de terceirização o PEP tem que contactar o PDP sempre necessita de uma decisão, o PDP por sua vez após receber o pedido faz a decisão adequada e responde-lhe. [1]

No modelo de aprovisionamento o PEP é configurado pelo PDP com as politicas a ser aplicadas e desse modo o PEP poderá fazer as decisões e aplica-las sem ter que contactar o PDP. [1]

Mensagens COPS

Uma mensagem COPS é baseada em código binário e consiste de um cabeçalho seguido de objectos de politicas. [1] [8]

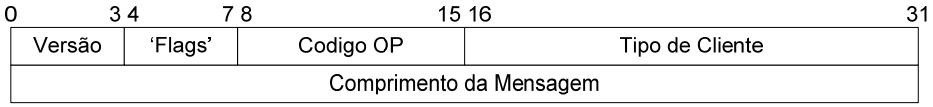

Figura 2.7 – Cabeçalho de uma mensagem COPS. [8]

No cabeçalho o campo ‘Versão’ indica a versão do protocolo COPS a que a mensagem se refere, o campo ‘Flags’ contém apenas o valor 0x1 se a mensagem foi pedida por outra mensagem COPS, o ‘Código OP’ indica que tipo de mensagem COPS se trata, operação a que se refere (ver tabela 2.9), o ‘Tipo de Cliente’ informa o receptor do tipo de cliente de politicas que enviou a mensagem de modo a podê-la interpretar correctamente (apenas as mensagens ‘Keep-Alive’ têm este valor a 0) e o campo ‘Comprimento da mensagem’ que informa o tamanho da mensagem medido em octetos. [1] [8]

Valor	Tipo de Mensagem
1	Pedido
2	Decisão
3	Reportar Estado
4	Apagar Estado de Pedido
5	Sincronizar Estado de Pedido
6	Cliente-Abrir
7	Cliente-Aceitar
8	Cliente-Fechar
9	Keep-Alive
10	Sincronização Completa

Tabela 2.9 – Códigos de Operação COPS.

Um objecto de politicas contém os campos: 'Comprimento' que indica o tamanho do objecto em octetos, 'C-Num' que informa o tipo de informação presente nos conteúdos do objecto e o campo 'C-Type' que identifica a que subtipo pertence essa informação cuja interpretação está dependente do valor do 'C-Num' (tabela 2.10). [1] [8]

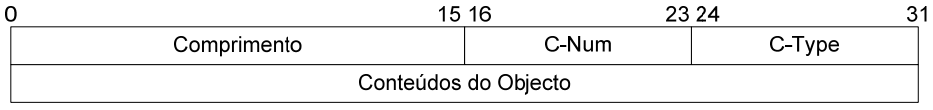

Figura 2.8 – Objecto de politicas de uma mensagem COPS.

Os valores possíveis o campo ‘C-Num’ estão presentes na tabela seguinte:

Valor	Tipo de Objecto
1	‘Handle’
2	Contexto
3	Interface de Entrada
4	Interface de Saída
5	Codigo Descritivo (‘Reason’)
6	Decisão
7	Decisão PDP
8	Erro
9	Informação Relativa ao Cliente
10	Temporizador ‘Keep-Alive’
11	Identificação PEP
12	Tipo de Reporte
13	Endereço de Redirecção PDP
14	Último Endereço PDP
15	Temporizador de Contabilização (‘Accounting’)
16	Integridade da Mensagem

Tabela 2.10 – Valores do Campo ‘C-Num’.

Aprovisionamento de Politicas COPS (COPS – PR)

No modelo de aprovisionamento, especificado no RFC3084, as politicas a aplicar são fornecidas ao PEP codificadas em documentos PIB ('Policy Information Base') ao contrário do modelo de terceirização em que o PEP tem que questionar o PDP quando necessita de uma decisão. Um PIB está estruturado em árvore cujas ramificações são classes de aprovisionamento, 'Provisioning Classes' (PRC), sendo que cada classe contém um conjunto de instanciações de aprovisionamento, 'Provisioning Instances' (PRI), que são folhas dessa classe que as contém. Cada PRI tem um identificador, 'Provisioning Instance Identifier' (PRID), que é o nome transportado pelo objecto COPS que identifica a instanciação da classe de aprovisionamento.

Em vez fornecer informação sobre um determinado evento quando contacta o PDP como acontece no modelo de terceirização, um PEP fornece informação acerca de si e com essa informação um PDP decide que políticas enviar de volta ao PEP. [1] [8]

COPS no IMS

No IMS o COPS é utilizado na interface Go para realizar a autorização dos fluxos, áudio e vídeo, e sua facturação utilizando extensões do COPS-PR. [8]

2.3.5. H.248

O protocolo H.248, também conhecido como 'MEdia GAteway COntrol Protocol' (MEGACO) é utilizado para a sinalização e gestão da sessão entre um 'media gateway' e o seu controlador e está definido no RFC3525. Ambas as entidades funcionam em modo 'master/slave'. [8]

Arquitectura H.248

A arquitectura do protocolo consiste em 'Media Gateways' (MGs) que fazem conversão de formatos de fluxos áudio/vídeo entre duas redes diferentes e 'Media Gateway Controllers' (MGCs) que controlam o estabelecimento e terminação desses fluxos nos MGs. [3]

O seu modelo conceptual define terminações, contextos, transacções, descritores, pacotes e mensagens baseando-se o protocolo em transacções entre MGCs e MGs, cada transacção envolve um pedido, 'TransactionRequest', e uma resposta, 'TransactionReply', sendo que o formato das mensagens pode ser em modo texto ou em modo binário. As terminações são entidades lógicas num MG que são fontes ou terminações de fluxos de áudio/vídeo, podem ser físicas, sendo uma ligação a uma linha analógica ou a uma rede, ou efémeras, representando fluxos lógicos de áudio/vídeo, que são criadas ou destruídas através de comandos. Os contextos são associações de terminações com o objectivo de partilhar fluxos de áudio/vídeo entre várias terminações que podem ser adicionadas, movidas ou removidas de um contexto podendo apenas coexistir num. As transacções consistem na passagem de pedidos e respostas relacionados com terminações dentro de contextos. Os comandos entre um MGC e um MG são agrupados em transacções de modo a que comandos relacionados com um contexto não interfiram com outro contexto. Os descritores são estruturas de dados que fornecem informações adicionais, que podem ser necessárias ou opcionais, enviadas juntamente com os comandos e respostas. Existem vários tipos de descritores: descritor de modem, multiplexagem, fluxos de áudio/video, eventos, sinalização, auditoria, pacotes, alteração de serviço, marcação de números, estatísticas, eventos observados, erros e topologia. E pacotes são conjuntos de propriedades implementados por uma terminação que os terá que suportar, cada pacote conterá sinalizações, eventos e estatísticas aos quais são atribuídos valores e identificadores, um MGC poderá fazer uma auditoria a uma terminação acerca dos pacotes que suporta. Estão definidos na especificação do H.248 vários pacotes básicos. [3]

Mensagens H.248

Uma mensagem, em modo texto, inicia-se com a palavra MEGACO seguido de '/', a versão do protocolo, o identificador da mensagem (mID) que tipicamente pode ser o nome do domínio ou um endereço IP da origem e em baixo o corpo da mensagem. O corpo da mensagem poderá conter múltiplas transacções diferentes com múltiplos comandos no interior de cada uma. A ordem com que as transacções se apresentam numa mensagem não implica a sua ordem de execução ao contrário do que se passa na ordem dos comandos no interior de cada transacção. [3]

A diferença existente entre uma mensagem de comando e uma de resposta é a etiqueta 'Reply' que estará presente substituindo a etiqueta 'Transaction' estando assim a referir a que transacção corresponde a resposta que depois conterá os descritores adequados. [3]

Comandos H.248

Existem 8 comandos definidos para o controle e manipulação de contextos e terminações dos quais 6 são enviados do MGC para o MG que são o 'Add', 'Modify', 'Subtract', 'Move', 'AuditValue' e 'AuditCapabilities'. O 'Notify' apenas é enviado do MG para o MGC e o 'ServiçeChange' que é enviado nos dois sentidos. [3]

O comando 'Add'adiciona uma terminação a um contexto implicitamente criando um contexto caso seja a primeira terminação adicionada. O comando 'Modify' modifica propriedades do estado de uma terminação e propriedades referentes a fluxos de áudio/vídeo. O comando 'Subtract' remove uma terminação de um contexto implicitamente removendo um contexto caso seja a única terminação. O 'Move' move uma terminação de um contexto para outro. O 'AuditValue' obtém as propriedades de uma terminação. O 'AuditCapabilities' obtém todas as propriedades suportadas por uma terminação para além das sinalizações e eventos suportados pelo MG. O 'Notify' é

utilizado pelo MG para notificar o MGC de certos eventos. E o 'ServiceChange' é utilizado pelo MG para informar o MGC de alterações nos serviços fornecidos. [8]

H.248 no IMS

No IMS o MEGACO é utilizado no MRF para a comunicação entre o MRFC e o MRFP na interface Mn e entre MRFC e o IMS-MGW na interface Mp. Ambas as interfaces são baseadas no H.248 principalmente para reservar ou iniciar terminações, iniciar ou terminar cancelamento de eco, tons e anúncios em terminações e enviar ou receber tons DTMF, embora possam ter de ser adicionadas novas funções. [8] [1]

2.3.6. RTP

O 'Real-time Transport Protocol' (RTP), definido no RFC3550, é um protocolo fim-a-fim ('end-to-end') para a transmissão de informação a tempo real com fiabilidade utilizando protocolos de transporte sem garantias de entrega como o UDP. É utilizado em conjunto com o 'RTP Control Protocol' (RTCP) e seu objectivo é permitir que informação áudio/vídeo, que tendo restrições a nível de temporização, seja transmitida através de uma rede em tempo real sem atrasos significativos impedindo a sua correcta visualização e/ou audição. Fornece capacidades de identificação da origem e tipo de dados transportados, números de sequência, marcadores de tempo e monitorização de entrega ao mesmo tempo impedindo com que uma parcela temporal de informação seja repartida por diferentes pacotes aumentando as probabilidades de perda dessa informação. A identificação da origem e o tipo de dados transportados é conseguida através de um identificador em cada pacote, desse modo permitindo a identificação da fonte emissora que é extremamente útil no caso de uma conferência de múltiplos participantes e a identificação do 'codec'

utilizado. Existem dois géneros de dados transportados, estáticos e dinâmicos, os estáticos correspondem sempre ao mesmo identificador, os dinâmicos podem ter identificadores diferentes dentro da mesma sessão que são acordados durante a negociação da mesma. A utilização de números de sequência permite o calculo estatístico do número de pacotes que são perdidos na rede não chegando ao destino e assim fornecer essa informação às camadas superiores de modo a que possam ajustar características relacionadas com a sessão ou ligação tais como o tamanho do 'buffer' ou o 'codec' utilizado. Os marcadores de tempo permitem com que se possam organizar no 'buffer' de entrada os dados de áudio/vídeo de acordo com a porção de tempo a que correspondem e não a ordem com que chegam pois numa rede IP normal uma sequência de pacotes frequentemente não chegará ao destino na mesma ordem com que foi enviada e desses pacotes alguns não chegarão a tempo ou até faltarão sendo que as acções típicas das aplicações de camadas superiores sejam a interpolação da informação presente ou a repetição do pacote anterior para colmatar essas falhas. Para a protecção do tráfego existe o 'Secure RTP' (SRTP) definido no RFC3711 que fornece confidencialidade, autenticação e protecção contra ataques de repetição onde os vários participantes, através da utilização de um protocolo de gestão de chaves, criam uma chave mestre utilizada para gerar uma chave de sessão que é constantemente actualizada. [1] [8]

Cabeçalho RTP

O cabeçalho RTP, que pode ser observado na figura 2.9, contém as informações necessárias para que a aplicação no destino possa reconstituir correctamente os dados enviados e é anexado a uma ou mais amostras temporais e tipicamente enviado utilizando UDP. [1] [8] [3]

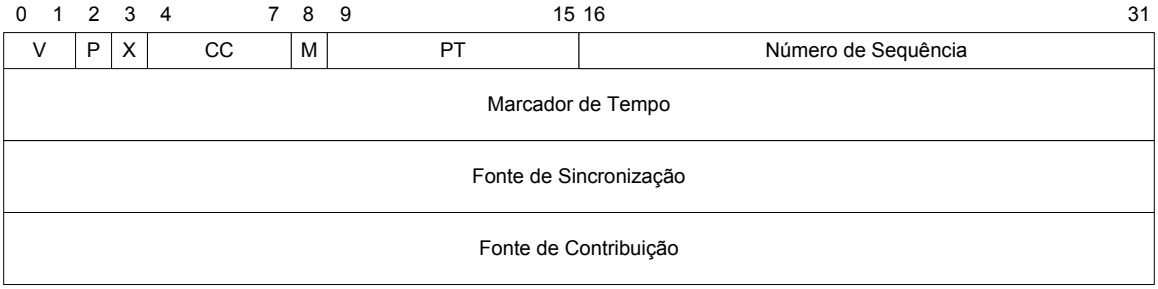

Figura 2.9 – Cabeçalho RTP [3]

V – Versão do protocolo RTP utilizada.

P – Indica se o pacote RTP contém octetos de 'padding'.

X – Indica a existência de uma extensão ao cabeçalho.

CC – Indica o número de fontes contributivas que se seguem ao cabeçalho.

M – Marcador cuja interpretação varia com o perfil utilizado.

PT – Identifica o 'codec' dos dados transportados.

Número de sequência – Permite reordenar pacotes que chegam ao destino fora de ordem e determinar perdas de pacotes.

Marcador de tempo ('timestamp') – Indica o tempo da amostragem do primeiro octeto dos dados transportados.

Fonte de sincronização – Identifica a fonte dos pacotes RTP.

Fonte de contribuição – Lista as fontes dos pacotes que contribuíram para o fluxo de informação se este é resultado de uma mistura de vários. Permite um máximo de 15 fontes e se exceder esse valor apenas os primeiros 15 são identificados. [1] [8] [3]

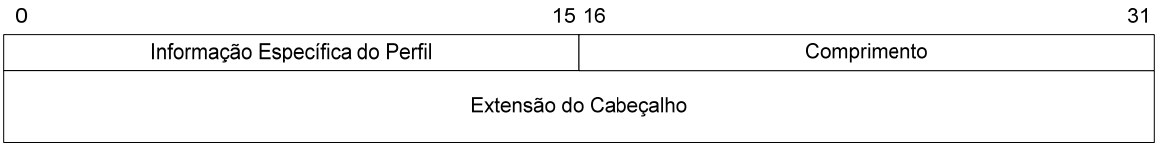

Figura 2.10 – Extensão de cabeçalho RTP [3]

Qualquer informação, que não esteja privilegiada no cabeçalho, pode ser incluída nos dados transportados se assim estiver definido no perfil dos dados transportados, caso contrário poderá ser enviada numa extensão de cabeçalho, figura 2.10, cuja presença é indicada no campo X do cabeçalho e será colocado entre as fontes contributivas e os dados transportados. [3]

RTP no IMS

No IMS o RTP é utilizado em conjunto com o protocolo de transporte UDP não sendo utilizada qualquer segurança, SRTP, que é assumida como implementada às camadas inferiores do IMS. [1]

2.3.7. RTCP

O protocolo 'RTP Control Protocol' (RTCP), definido juntamente com o RTP no RFC3550, é utilizado em conjunto com o protocolo RTP e permite a troca periódica de informações de controlo entre participantes de uma sessão com o objectivo de monitorizar a qualidade das ligações e sincronizar dados entre os vários fluxos de dados, opcionalmente permite associar os identificadores das fontes com nomes legíveis. O protocolo em si baseia-se na transmissão periódica pacotes de controlo a todos os participantes de uma sessão utilizando o mesmo modo de transmissão que os pacotes de dados o que significa que os pacotes RTCP são enviados pelo mesmo canal que os pacotes RTP contribuindo para os problemas que se propõe a corrigir, para resolver isso é definido no RFC1889 o algoritmo que calcula o tamanho dos pacotes e o seu espaçamento temporal, para além disso, é recomendado que a largura de banda utilizada para o controlo seja cerca de 5% da largura de banda total da sessão. [3] [1] [8] [104] [39]

Pacotes RTCP

O RTCP tem 5 tipos de pacotes que serão sempre enviados em conjuntos de pelo menos 2 pois desse modo é diminuído o impacto da utilização de um canal de controlo na rede para além da necessidade de qualquer novo participante receber a identidade das várias fontes da sessão o mais rápido possível. Opcionalmente cada pacote poderá ser codificado, nesse caso um prefixo será adicionado ao topo do conjunto dos pacotes.

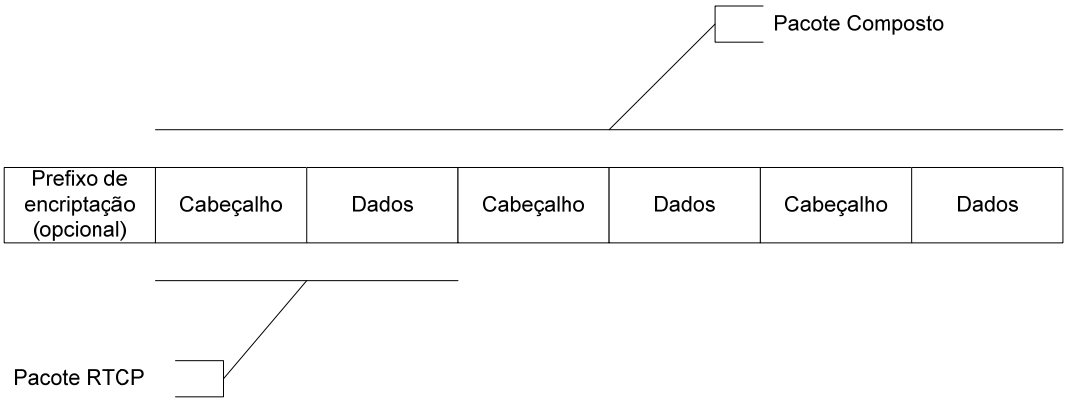

Figura 2.11 – Pacote RTCP composto [3]

Podem ser do tipo 'Sender Report' (SR), 'Receiver Report' (RR), 'Source Description' (SDES), 'BYE' e 'APP'.

Um SR é enviado por fontes de fluxos áudio/vídeo activas informando estatísticas de transmissão e recepção.

Um RR é enviado por fontes de fluxos áudio/vídeo inactivas informando estatísticas de recepção.

Um SDES contém informações descritivas da fonte.

Um BYE termina uma participação.

Um APP contém funções ou informações, específicas a uma aplicação, definidas num perfil RTP que conterá especificações de formatos de cabeçalhos e suas extensões, 'codecs' utilizados, serviços e detalhes comportamentais.

2.4. Arquitectura IMS

O objectivo principal da arquitectura IMS é o controlo de sessões fornecendo suporte para que outros serviços e utilizadores possam usufruir dessa capacidade de um modo agnóstico à rede de acesso por parte dos utilizadores. A arquitectura IMS está estruturada em camadas lógicas: camada de acesso, de controlo e de aplicações. A camada de acesso refere-se aos procedimentos necessários para que o dispositivo possa aceder à rede IMS e seus serviços. A camada de controlo corresponde à rede IMS nuclear contendo os 'Call Session Control Functions' (CSCF's), a base de dados 'Home Subscriber Server' (HSS), as várias 'gateways' que permitem a interoperabilidade com redes tradicionais de comutação de circuitos e outras redes IMS de outros domínios e o 'Media Resource Function' (MRF). E a camada de aplicações que compreende os vários servidores de aplicações, 'Aplication Servers' (AS's), que são a plataforma base para o desenvolvimento de serviços a disponibilizar para os utilizadores da rede. Para o funcionamento destas entidades e interacções entre si foram apenas especificadas funcionalidades, protocolos e operações deixando em aberto os detalhes das implementações internas de cada uma das entidades e para as comunicações entre entidades são descritos pontos de referência para interfaces específicas de comunicação estandardizando assim o modo como uma entidade específica comunicará com outra. Na figura seguinte podem-se observar os vários blocos conceptuais que fazem parte da arquitectura IMS, existem também entidades para funções de facturação e qualidade-de-serviço que não estão presentes na figura pois funcionam como uma estrutura de suporte para a arquitectura. [8] [1] [2]

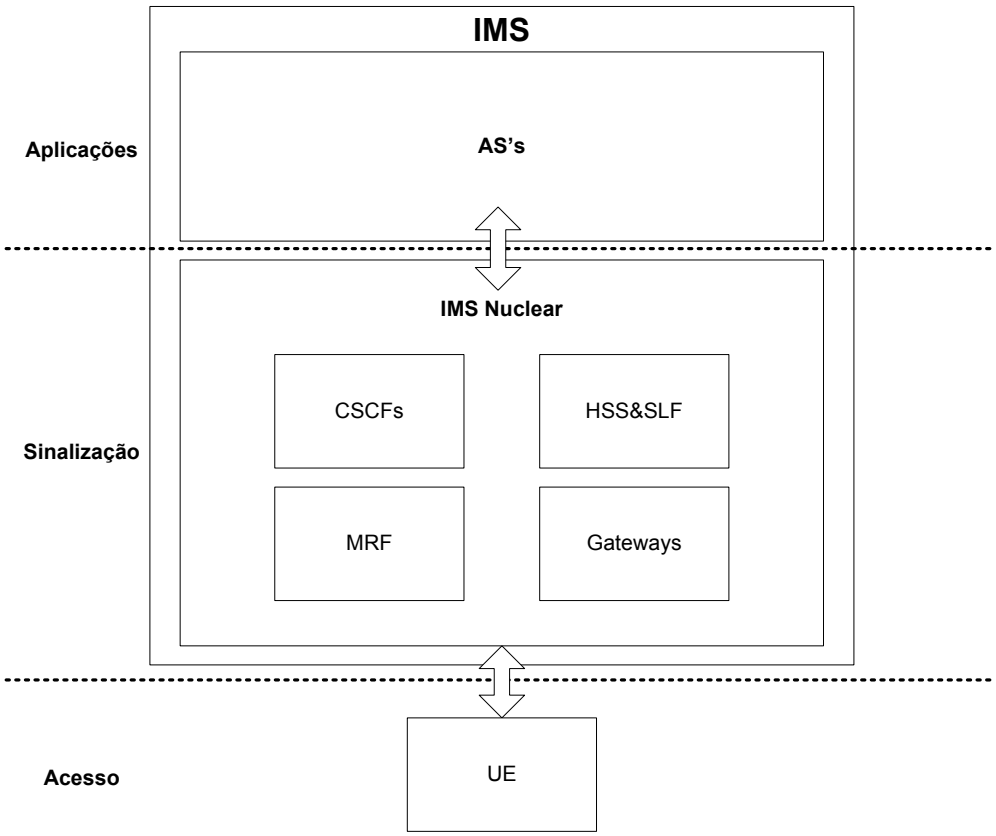

Figura 2.12 – Visão geral da arquitectura IMS

Em seguida, as entidades e interfaces aqui presentes, são definidas com maior detalhe em dois grupos, IMS nuclear e camada de serviços.

2.4.1. IMS Nuclear

2.4.1.1. HSS e SLF

O ‘Home Subscriber Server’ (HSS) é uma base-de-dados que funciona como um repositório para as informações relacionadas com os utilizadores da rede. É responsável por conter informações de identificação, endereçamento, segurança, localização e perfis de cada utilizador, para além disso é responsável pela geração de informações de segurança para funções relacionadas com integridade, codificação e autenticação mútua que são disponibilizadas para outras entidades da mesma rede de modo a que possam identificar e fornecer serviços aos utilizadores. Poderão existir vários HSS’s na mesma rede de modo a acomodar mais utilizadores, nesse caso terá de existir um ‘Subscription Locator Function’ (SLF) que actuará como um servidor de redireccionamento Diameter ou seja fornecerá um serviço de resolução permitindo ao S-CSCF, I-CSCF ou AS descobrir o nome do HSS que conterá as informações de uma determinada identidade de utilizador. [8] [13]

O SLF é apenas necessário num ambiente de múltiplos HSS e onde as entidades não estão pré-configuradas para aceder apenas a um determinado HSS. Recebe pedidos do I-CSCF e do S-CSCF durante o registo e durante a criação de uma sessão pela interface Dx, também do AS pela interface Dh e do servidor 3GPP AAA pela interface Dw de modo a obter o nome do HSS que contem a informação do subscritor. A sua interface principal é a Dx, é utilizada conjuntamente com a interface Cx e ambas utilizam o protocolo Diameter. Quando um I-CSCF ou S-CSCF não sabe para qual HSS deve enviar um pedido, envia-o para um SLF que responde com uma mensagem de redireccionamento Diameter e por sua vez, a entidade que enviou o pedido, envia-o de novo para o HSS indicado na mensagem de redireccionamento. [13]

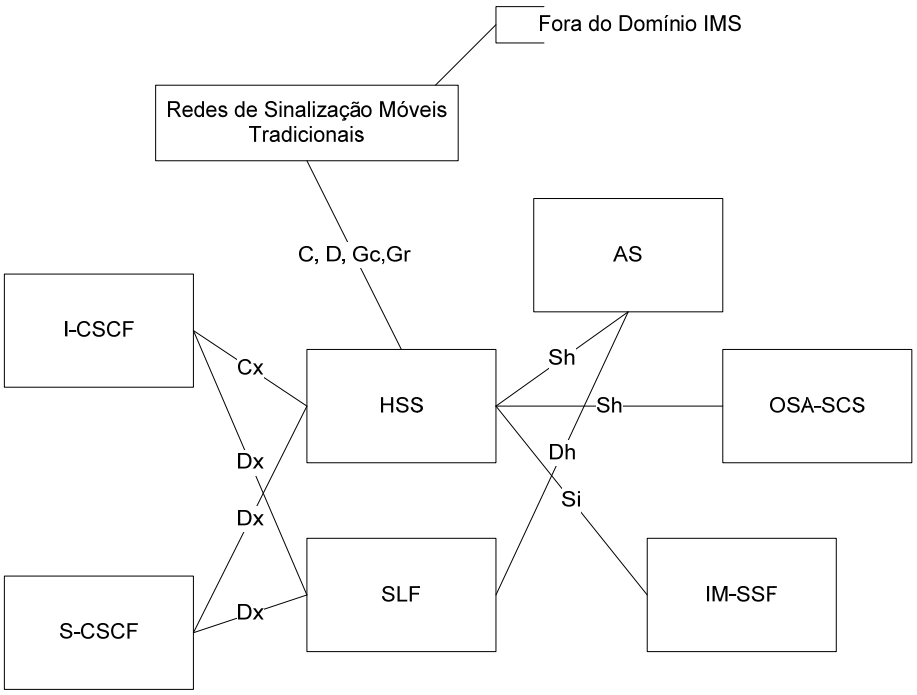

Figura 2.13 – HSS, SLF e suas ligações [13] [1]

O HSS está ligado ao I-CSCF e S-CSCF pela interface Cx, ao AS e OSA-SCS pela interface Sh, ao IM-SSF pela interface Si e poderá se ligar às redes de sinalização móveis tradicionais ('legacy') pelas interfaces C, D, Gc e Gr. [13]

A interface mais relevante é a Cx que utiliza o protocolo Diameter. As suas funções pertencem a três categorias: gestão de localização, manipulação de informação e autenticação. [8]

A gestão de localização divide-se em duas partes: registo e localização. A parte de registo refere-se a quando o I-CSCF recebe um SIP REGISTER do P-CSCF e envia um 'User-Authorization-Request' (UAR) ao HSS que por sua vez lhe responderá, se o pedido for bem sucedido, com um 'User-Authorization-Answer' (UAA) que conterá o nome do S-CSCF que está atribuído ao utilizador caso o utilizador já esteja registado ou as capacidades do S-CSCF a atribuir caso não esteja registado ou ambos se assim for explicitamente pedido pelo I-CSCF. Também se refere a quando o S-CSCF recebe um SIP REGISTER do I-CSCF que previamente descobriu qual o S-CSCF a contactar e por sua vez o S-CSCF envia um 'Server-Assignement-Request' (SAR) ao HSS informando-lhe

qual o S-CSCF que estará atribuído ao utilizador enquanto o registo estiver activo, caso o registo expire será enviado um outro SAR de modo a informar que deixou de estar atribuído ao utilizador. Após a recepção do SAR o HSS responderá com um 'Server-Assignement-Answer' (SAA) que conterá o perfil do utilizador baseado nas opções do SAR. Para além disso o HSS pode tomar a iniciativa de terminar o registo do utilizador em situações atípicas, nesse caso envia um 'Registration-Termination-Request' (RTR) ao S-CSCF que responde com um 'Registration-Termination-Answer' (RTA) indicando o resultado da operação. A parte de localização é relacionada com a procura do S-CSCF associado ao utilizador quando o método SIP utilizado não é um SIP REGISTER, nesse caso é utilizado um 'Location-Info-Request' (LIR) que o I-CSCF envia ao HSS que por sua vez responde com um 'Location-Info-Answer' (LIA) contendo o nome do S-CSCF associado ou, caso não exista nenhum associado, conterá as capacidades a serem fornecidas pelo S-CSCF que serão necessárias. [8]

A manipulação de informação refere-se às situações em que é necessária a alteração dos dados do utilizador e serviços que previamente tinham sido transferidos para o S-CSCF exceptuando os casos em que o utilizador associado a um S-CSCF não está registado e a alteração é referente a dados ou serviços do utilizador quando registado. Nessas situações o HSS envia um 'Push-Profile-Request' (PPR) ao S-CSCF que responderá com um 'Push-Profile-Answer' (PPA) contendo o resultado da operação. [8]

A autenticação no IMS baseia-se numa palavra secreta pré-partilhada, que está armazenada no 'IP Multimedia Services Identity Module' (ISIM) e no HSS. Para a autenticação de cada utilizador o HSS contém um ou mais vectores de autenticação organizados por número de prioridade que são constituídos de um método de autenticação, informação de autenticação, uma chave de integridade e, caso seja necessário, uma chave de confidencialidade. No IMS é o S-CSCF que é responsável pela autorização de um utilizador e para isso envia um 'Multimedia-Auth-Request' (MAR) ao HSS que, caso exista o utilizador, responderá com um 'Multimedia-Auth-Answer' (MAA) que conterá informação para a autenticação do utilizador constituída de um ou vários vectores de autenticação. [8]

2.4.1.2. CSCF

O 'Call/Session Control Function' (CSCF) é um servidor SIP essencial no IMS que processa a sinalização das chamadas ou sessões. As suas funções são distribuídas pela rede em várias entidades permitindo uma melhor escalabilidade e segurança. Existem quatro entidades que colectivamente fazem o CSCF sendo que as três primeiras são fundamentais para o funcionamento do IMS, são: 'proxy' CSCF (P-CSCF), 'serving' CSCF (S-CSCF), 'interrogating' CSCF (I-CSCF) e 'emergency' CSCF (E-CSCF). O P-CSCF é o primeiro ponto de contacto do UE no IMS, o S-CSCF gere os estados de sessão na rede, o I-CSCF é principalmente o ponto de contacto para todas as ligações vindas de uma rede IMS fora da rede do operador incluindo os casos de 'roaming' e o E-CSCF que, não sendo fundamental no IMS, gere alguns aspectos relacionados com o estabelecimento de sessões em situações de emergência. Cada uma destas entidades, que têm funções específicas embora possam partilhar algumas delas, actua principalmente como um 'proxy' 'stateful' mantendo detalhes das sessões a decorrer e do estado do registo de cada UE. [8] [13] [6] [2] [1]

P-CSCF

O P-CSCF é o primeiro ponto de contacto do terminal do utilizador, 'User Equipment' (UE), na rede IMS, pode se localizar quer na rede visitada ou na rede natal dependendo da sua rede de acesso e das preferências do operador. Comporta-se como um proxy SIP, definido no RFC3261, não pode modificar o 'Request URI' de uma mensagem 'SIP INVITE' mas pode actuar como um agente SIP gerando ou terminando transacções SIP. As suas funções são: reencaminhar os pedidos SIP 'REGISTER' enviados pelo UE baseando-se no domínio e as mensagens para o S-CSCF que foi atribuído ao UE após um registo bem sucedido, assegurar-se que a informação da rede de acesso, caso esteja presente nas mensagens, esteja correcta e actualizada, reencaminhar os pedidos e respostas SIP para os UE's a si ligados, detectar e gerir pedidos de estabelecimento de sessões de emergência, gerar os CDR's, manter uma associação de segurança, 'Security Association' (SA), entre si e cada UE a si ligado (TS33.203), executar a compressão/descompressão de

mensagens SIP e efectuar a autorização de recursos de rede e gestão de QoS (TS33.203). [14] [1]

O P-CSCF liga-se ao S-CSCF e ao E-CSCF através da interface Mw, ao 'Interconnection Border Control Function' (IBCF) através da interface Mx, ao UE através da interface Gm e ao 'Policy and Charging Rules Function' (PCRF) através da interface Rx. [13]

Para se ligar ao P-CSCF o UE é pré-configurado durante o aprovisionamento inicial ou descobri-lo-á através do procedimento DHCP/DNS (TS23.228). [14]

S-CSCF

O S-CSCF é o ponto fulcral para o funcionamento do IMS, localiza-se sempre na rede natal. É o nó central da camada de sinalização no IMS que permite aos operadores controlar todo o fornecimento de serviços e estabelecimento de sessões, actua essencialmente como um servidor SIP com funções de controlo de sessão e um 'registrar' SIP fazendo a gestão dos processos de registo mantendo os perfis dos utilizadores que são obtidos do HSS através da interface Cx, faz o reencaminhamento das mensagens de e para o UE mantendo o estado actual de cada negociação e também aplica as politicas de rede impedindo com que cada utilizador execute operações que não são autorizadas quer a si ou na rede. Tipicamente existem vários S-CSCF numa rede de modo a fornecer escalabilidade, redundância e segurança, a não ser que tenha poucos utilizadores, pois cada S-CSCF servirá um conjunto de UE's baseado na quantidade de UE's que suporta simultaneamente e nos serviços necessários por cada um deles. [1] [2] [8] [8]

Cada S-CSCF comunica com o I-CSCF e com o P-CSCF usando a interface Mw, com redes 'IP Multimédia' utilizando a interface Mm, com o HSS usando a interface Cx, com o SLF usando a interface Dx, com o AS usando a interface ISC, com o IBCF usando a interface Mx, com o MRFC usando a interface Mr, com o MGCF usando a interface Mg e com o BGCF usando a interface Mi. [13]

I-CSCF

O I-CSCF é o ponto de contacto dentro do âmbito de uma rede de um operador para todas as ligações destinadas a um utilizador dessa rede ou um utilizador 'roaming' na área de serviço desse operador, actua como um 'proxy' SIP e seu endereço é listado nos registos de DNS. Tem funções relacionadas com registo, sessões, facturação e utilização de recursos que podem ser resumidas em 4 tarefas: obter o nome do próximo salto a partir do HSS, atribuir um S-CSCF a um UA baseado nas suas capacidades que obteve do HSS, reencaminhar pedidos vindos de fora da rede ('incoming') para o S-CSCF ou AS adequado e fornecer, opcionalmente, a funcionalidade 'Topology Hiding Inter-network Gateway' (THIG) que consiste em codificar partes das mensagens SIP que contêm informação sensível acerca da rede interna de modo a oculta-las ao exterior. O I-CSCF também actua como uma 'firewall' que permitirá ou não o acesso a outras redes protegendo entidades dentro do domínio de confiança tais como o S-CSCF e o HSS de acessos exteriores não autorizados. Uma rede IMS típica conterá múltiplos I-CSCF por motivos de escalabilidade e redundância sendo normalmente localizado na sua rede natal, embora, em algumas situações, em que fornece a funcionalidade THIG, possa ser localizado numa rede visitada. [2] [1] [8] [14]

Um I-CSCF comunica com o S-CSCF utilizando a interface Mw, com SLF utilizando a interface Dx, com o HSS utilizando a interface Cx, com o AS utilizando a interface Ma, com o MGCF utilizando a interface Mg. [13]

E-CSCF

O ‘Emergency’ CSCF (E-CSCF) é responsável por permitir a criação de sessões de emergência reencaminhando pedidos de sessões de emergência enviados pelo P-CSCF para um centro de emergências, o ‘Public Safety Answering Point’ (PSAP), ou o BGCF adequado baseando-se na localização do UE e no tipo de serviço de emergência presente no pedido. A localização do UE poderá ser adicionada pelo P-CSCF que a obterá do ‘IP Connectivity Access Network’ (IP-CAN) ou, caso não esteja presente e seja necessária, o E-CSCF terá de a poder obter do ‘Location Retrieval Function’ (LRF) que é responsável pela obtenção de informação de localização do UE que está a tentar iniciar uma sessão de emergência IMS. O E-CSCF e o P-CSCF têm que estar sempre localizados na mesma rede o que significa que no caso do UE estar numa situação de ‘roaming’ terá de ser utilizado o E-CSCF da rede visitada, nessa situação, ou caso receba uma resposta indicando a necessidade de o fazer ou caso não esteja registado na rede, o UE terá de fazer um registo de emergência utilizando um identificador público especial para o efeito no pedido SIP REGISTER que será reconhecido pelo S-CSCF como tal. [15]

O E-CSCF comunica com o P-CSCF utilizando a interface Mw, com o BGCF utilizando a interface Mi, com o LRF utilizando a interface Ml e com redes IP Multimédia utilizando a interface Mm. [13]

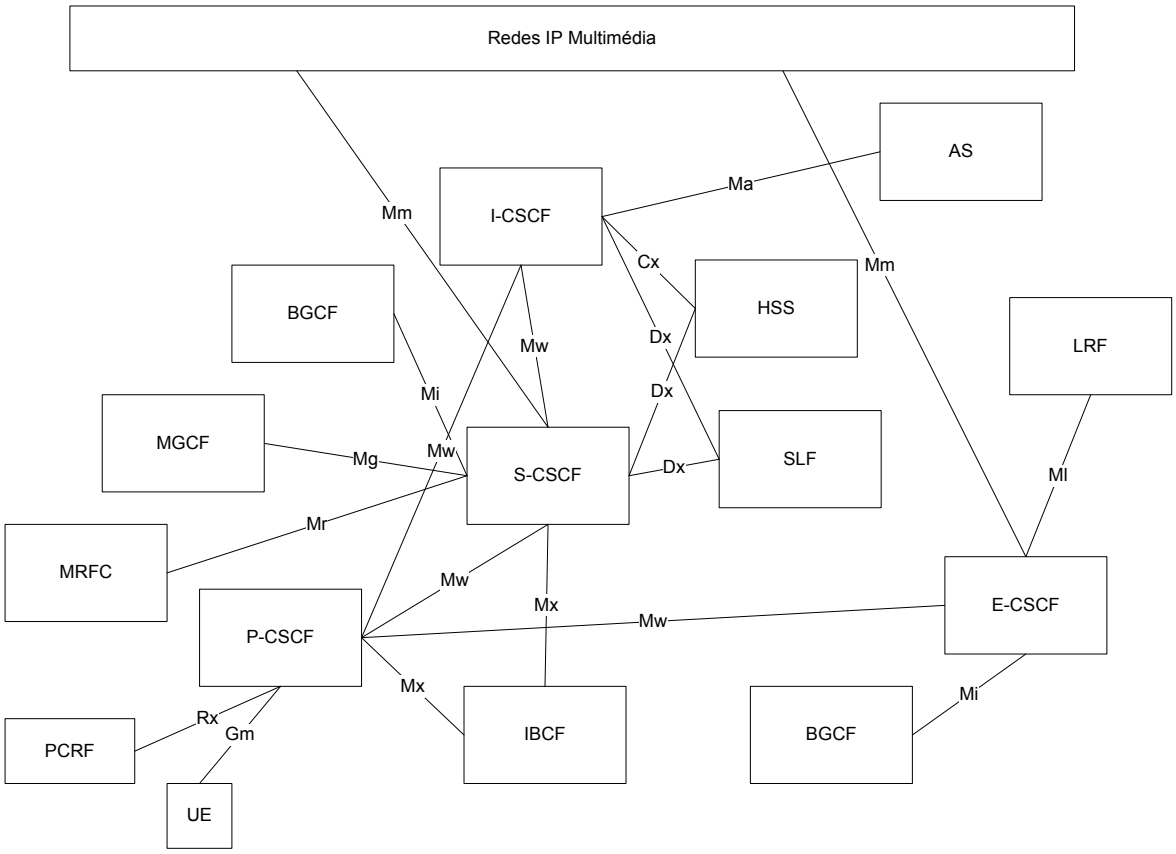

Figura 2.14 – Entidades do CSCF e suas ligações [13] [15]

As interfaces mais relevantes para o CSCF são a Cx, Dx e Mw. As interfaces Cx e Dx já foram referidas no capítulo do HSS. A interface Mw, que é baseada no protocolo SIP, é utilizada para as ligações entre o P-CSCF e o S-CSCF, o P-CSCF e o E-CSCF, o P-CSCF e o I-CSCF e entre o S-CSCF e o I-CSCF. As suas funcionalidades dividem-se em 3 grupos: registo, controlo de sessão e transacções. [8]

Nos procedimentos de registo, a interface Mw é utilizada pelo P-CSCF para reenviar o pedido de registo do UE para o I-CSCF que por sua vez utiliza a interface Mw para reenviar o pedido para o S-CSCF escolhido, a resposta a esse pedido fará o caminho contrário ao pedido. Para além disso a interface Mw também é utilizada pelo S-CSCF para informar o UE acerca de procedimentos de terminação de um registo, ou re-autenticação, iniciados pela rede e para informar o P-CSCF que liberte recursos referentes a determinado utilizador. [8]

Nos procedimentos de controlo de sessão existem duas situações: sessões que originam de um contexto móvel e sessões que terminam num contexto móvel. Nas sessões originadas de um contexto móvel a interface Mw é utilizada para reenviar pedidos do P-CSCF para o S-CSCF e deste para o I-CSCF. Nas sessões terminadas num contexto móvel a interface Mw é utilizada para reenviar pedidos do I-CSCF para o S-CSCF e deste para o P-CSCF. Para além disso a interface Mw é utilizada para o envio de dados de facturação e para a terminação de sessões sob iniciativa da rede, um exemplo será quando o P-CSCF inicia uma terminação de uma sessão em curso quando é informado pelo PDF que as fontes de áudio/vídeo pararam de transmitir. [8]

Nos procedimentos de transacção a interface Mw é utilizada para transmitir pedidos que não criam um diálogo e receber as suas respostas. [8]

2.4.1.3. Gateways

Uma rede IMS de modo a ser interoperável com outras redes contém um conjunto de entidades que fornecem as funções necessárias para a conversão e emulação da sinalização e dados enviados nas sessões, estas entidades que poderão alterar parcialmente ou totalmente o tráfego que as atravessa, funcionam de modo transparente e em vários níveis de abstracção, fazem conversão de sinalização, conversão de dados multimédia ou conversão de protocolos de endereçamento, também existem entidades que agregam ou controlam um conjunto de 'gateways' que podem comunicar entre si ou com um controlador através de um protocolo, proprietário ou aberto.

Numa rede típica IMS existem várias entidades que fornecem serviços de interoperabilidade, que são: 'Breakout Gateway Control Function' (BGCF), 'Interconnection Border Control Function' (IBCF), 'Transition Gateway' (TrGW), 'Media Gateway Control Function' (MGCF), 'IMS – Media Gateway Function' (IMS-MGW), 'Signaling Gateway' (SGW) e 'Security Gateway' (SEG). [13]

BGCF

O ‘Breakout Gateway Control Function’ (BGCF) é essencialmente um servidor SIP que reencaminha mensagens baseado em informação recebida no protocolo de sinalização, no TEL URI, na informação administrativa e/ou no acesso a uma base de dados. Quando o S-CSCF decide que uma mensagem é destinada a um utilizador presente numa rede de comutação de circuitos esta é reencaminhada para o BGCF que terá de determinar onde ocorrerá a transição para essa rede, caso essa transição seja efectuada na mesma rede a mensagem será reenviada para o MGCF dessa rede, caso seja efectuada noutra rede fora do domínio a mensagem será enviada para o BGCF dessa rede contactando o I-CSCF dessa rede permitindo desse modo o envio da sinalização e dados áudio/vídeo em IP para o destinatário ou se necessário poderá ser enviada para o IBCF que o reenviará para o domínio adequado. As funções de um BGCF são: determinar o próximo salto para o reencaminhamento SIP em situações de transição de protocolar e geração de CDR’s, poderão haver vários BGCF na mesma rede e por motivos de segurança deverão apenas interagir com outros BGCF de outras redes. [2] [13] [8] [1] [13] [14]

Um BGCF situa-se na rede natal do utilizador e interage com o E-CSCF e o S-CSCF pela interface Mi, com o IBCF pela interface Mx, com outros BGCF da mesma rede e com outras redes IP Multimédia pela interface Mk e com o MGCF pela interface Mj. [13]

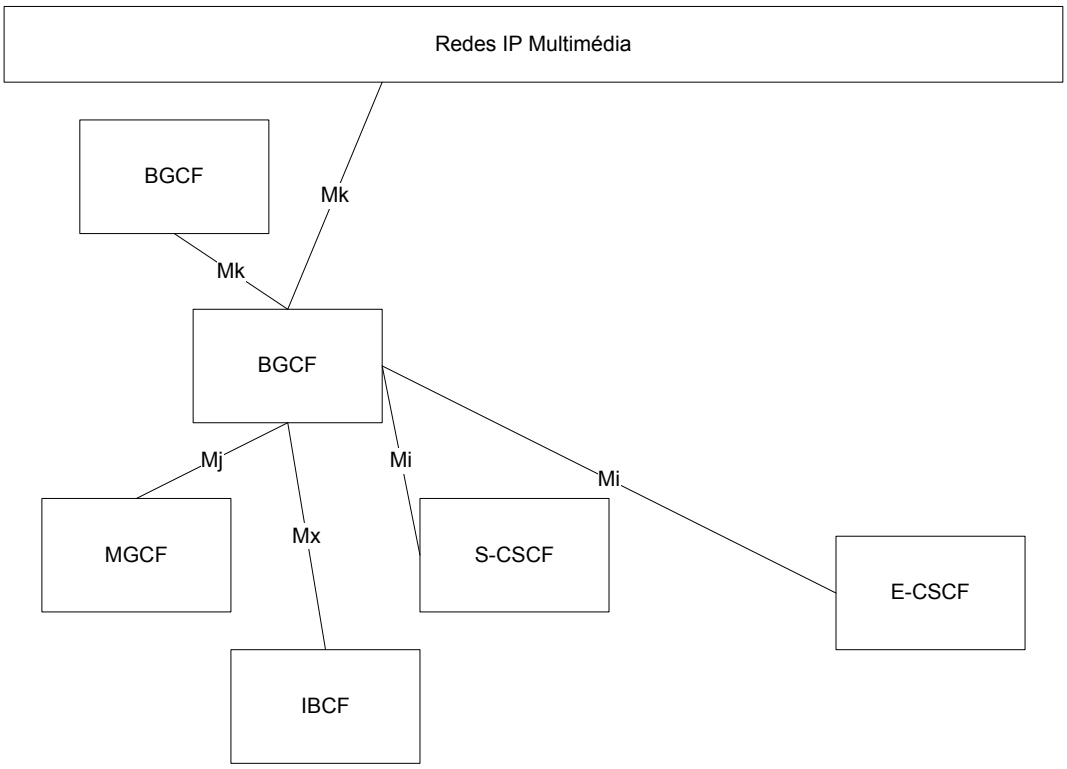

Figura 2.15 – BGCF e suas ligações [13]

As interfaces mais relevantes são as Mk, Mi e Mj. A interface Mk é baseada em SIP e permite a comunicação entre dois BGCF, entre um BGCF e um IMS ALG ou entre um BGCF e uma rede IP multimédia. A interface Mi é baseada em SIP e liga um S-CSCF, ou um E-CSCF, a um BGCF, é utilizada pelo S-CSCF, ou E-CSCF, para o reenvio de pedidos SIP para o BGCF quando estes necessitam de ser reencaminhados para uma rede de comutação de circuitos. E a interface Mj é baseada em SIP e permite a comunicação entre um BGCF e um MGCF quando a transição para uma rede de comutação de circuitos é efectuada na mesma rede que o BGCF. [8] [14]

IBCF

O ‘Interconnection Border Control Function’ (IBCF) actua como um separador entre dois domínios diferentes fornecendo funções ao nível da camada de aplicação no SIP/SDP de modo a permitir a sua interligação. Fornece as funcionalidades de IMS-ALG modificando as mensagens SIP de modo fazendo a conversão entre IPv4 e IPv6, controla o TrGW que executa a conversão ao nível dos fluxos de áudio/vídeo, pode implementar a funcionalidade THIG onde informações sensíveis sobre a topologia da rede interna são codificadas ocultando a topologia de rede ao exterior, selecção do ponto de interligação de sinalização adequado, poderá executar a invocação de um IWF (‘InterWorking Function’) quando é necessária a interligação de perfis SIP ou protocolos diferentes e proceder à geração de CDR’s. [1] [13] [14] [6]

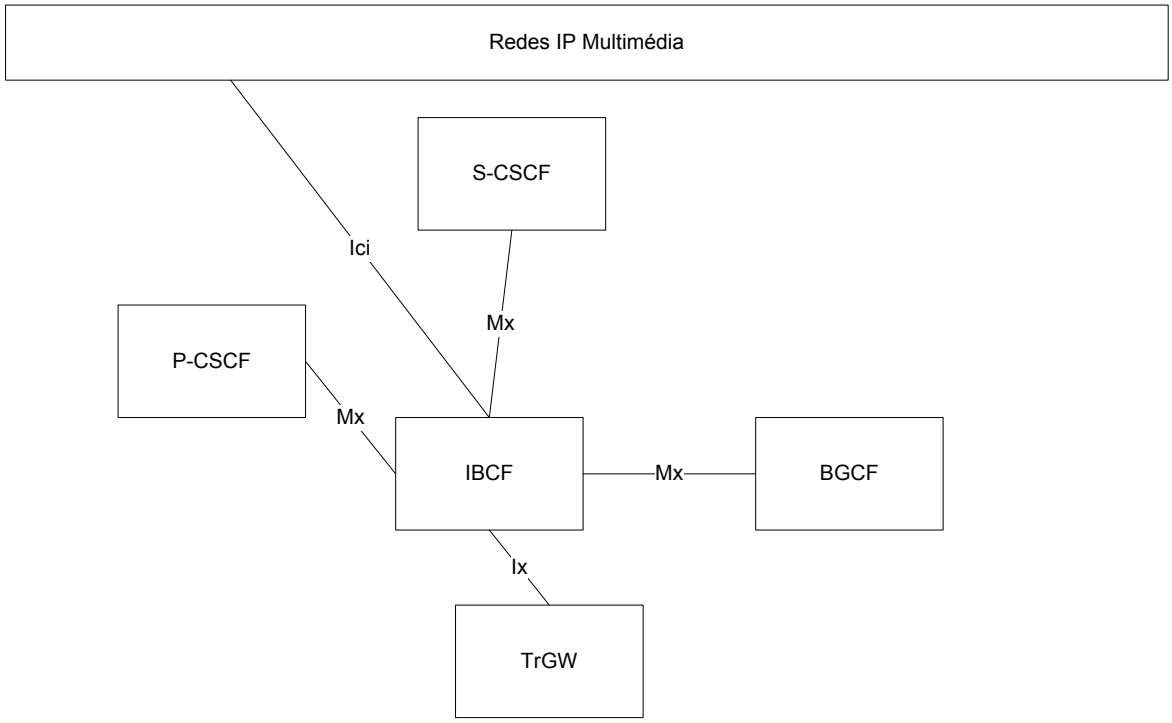

Figura 2.16 – IBCF e suas ligações [13]

O IBCF localiza-se na rede natal do utilizador e interage com o P-CSCF, o S-CSCF e o BGCF pela interface Mx, com o TrGW pela interface Ix e com outros IBCF situados noutras redes IP multimédia pela interface Ici. [13]

As interfaces mais relevantes são a Mx e a Ici. A interface Mx é baseada em SIP e permite a comunicação e reenvio de mensagens de sinalização entre um S-CSCF, P-CSCF e BGCF e um IBCF. A interface Ici permite com que dois IBCF's comuniquem de modo a trocar e reenviar mensagens de sinalização entre redes IP multimédia. A interface Ici juntamente com a interface Izi faz a interface rede-a-rede IMS, 'Inter-IMS Network to Network Interface' (II-NNI). [13] [14] [1]

TrGW

O 'Transition Gateway' (TrGW) é na prática um 'Network Address Port Translator Protocol Translator' (NAT-PT/NAPT-PT) que utiliza um conjunto de endereços globais únicos IPv4 para atribuir dinamicamente a entidades IPv6 e vice-versa para sessões iniciadas entre fronteiras de domínios IP de diferentes versões, IPv4/IPv6. É controlado por um IBCF e a tradução, de endereços e portos, é feita ao nível dos fluxos de áudio/vídeo alterando os endereços e portos contidos no cabeçalho IP e na informação de carga de cada pacote, deste modo fornece um reencaminhamento transparente entre ambos os domínios de diferentes versões IP sem necessitar de alterações nos terminais das ligações. [1] [13] [16]

O TrGW situa-se no caminho dos fluxos de áudio/vídeo e interage com o IBCF através da interface Ix e com outro TrGW, situado noutra rede IP Multimédia, através da interface Izi. [13]

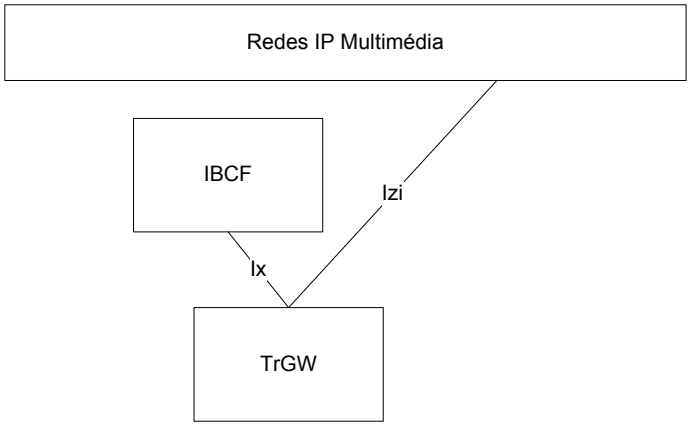

Figura 2.17 – TrGW e suas ligações [13]

A interface Ix é utilizada pelo IMS-ALG ou IBCF com função IMS-ALG para controlar o TrGW de modo a que possa, por exemplo, pedir a tradução de um fluxo de dados áudio/vídeo. A interface Izi é utilizada para reenviar um fluxo áudio/vídeo de um TrGW para outro, juntamente com a interface Ici faz a interface rede-a-rede IMS, 'Inter-IMS Network to Network Interface' (II-NNI). [13] [14] [1]

IMS-ALG

O 'IMS Aplication Layer Gateway' (IMS-ALG) é uma entidade por si só ou poderá estar embebida num S-CSCF ou num BGCF. Processa tráfego ao nível da sinalização de controlo reescrevendo campos relativos a endereços IP e portos nas mensagens SIP e SDP de modo a executar a conversão dos protocolos IPv4-IPv6 agindo como um SIP B2BUA mantendo duas zonas de sinalização independentes, uma para a rede IMS interna e outra para a rede externa, zonas essas que utilizam diferentes versões do protocolo IP, para coincidirem com os IP's e portos atribuídos pelo TrGW para a passagem do fluxo de informação áudio/vídeo. Também interage com o I-CSCF para tráfego vindo de fora e com o S-CSCF para tráfego com origem no interior da rede. Esta função pode estar embebida num S-CSCF ou num BGCF. [1] [6]

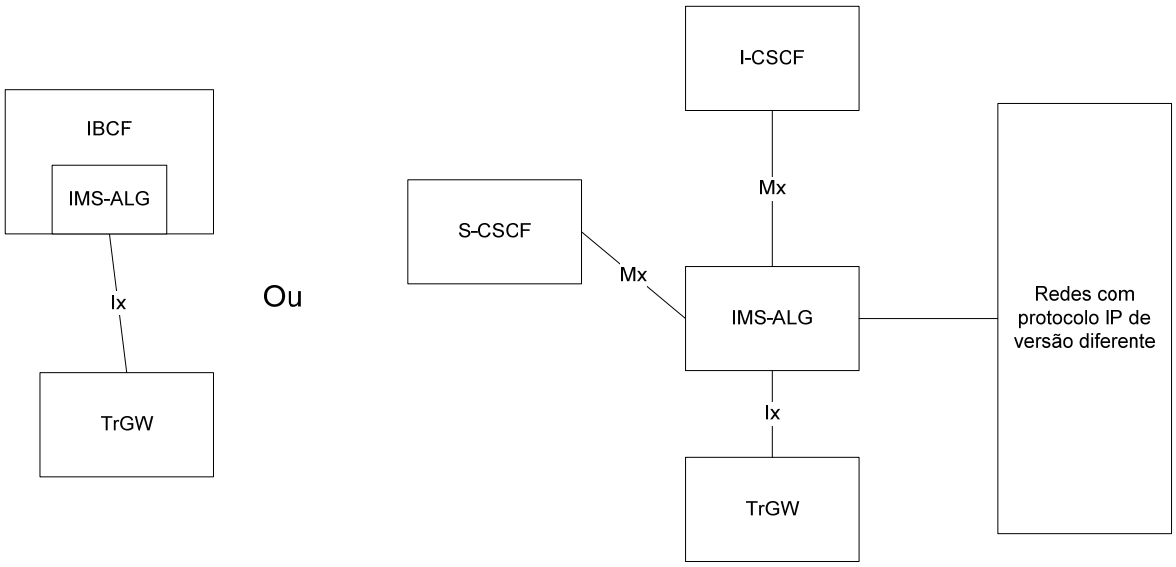

Figura 2.18 – IMS-ALG [1] [6]

A funcionalidade IMS-ALG está tipicamente presente num IBCF e poderá ser incluído num S-CSCF se um UE está por detrás de um NAT ou se é necessária tradução ao nível de IP entre o IP-CAN e o domínio IMS para os fluxos de áudio/vídeo. O IMS-ALG utiliza a interface Ix para controlar o TrGW. [13] [14]

IMS-MGW

O nome 'IMS Media Gateway Function' (IMS-MGW) refere-se à entidade 'Media Gateway Function' (MGW) de uma rede IMS, no caso de uma rede de um domínio de comutação de circuitos a entidade será referida como CS-MGW que tem as mesmas funcionalidades só que do ponto de vista de uma rede de comutação de circuitos, quando não é necessário distinguir as entidades será apenas utilizado o nome MGW. O IMS-MGW fornece, juntamente com o MGCF, a interligação entre uma rede IMS e uma rede de comutação de circuitos baseada no ISUP ou BICC. Faz a terminação dos canais de uma rede de comutação de circuitos, onde troca dados de sinalização e voz, e dos fluxos de áudio/vídeo de uma rede de comutação de pacotes, onde troca pacotes RTP com as redes IMS através da interface Mb, fornecendo serviços de conversão e processamento de áudio, controlo dos portadores e processamento dos dados transportados. Poderá também fornecer

tons e anúncios a utilizadores da rede de comutação de circuitos. Interage com o MGCF que reserva os recursos necessários ao nível dos fluxos de dados. [13] [17] [6] [1]

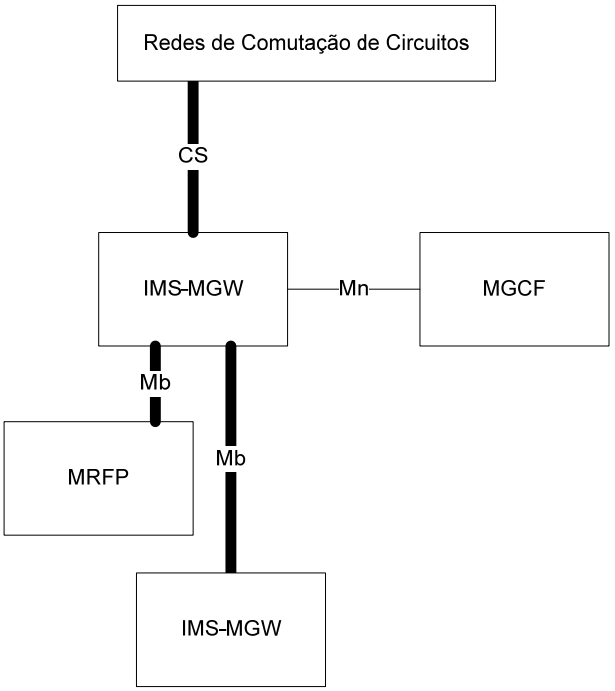

Figura 2.19 – IMS-MGW [13]

O IMS-MGW interage com o MGCF através da interface Mn, transfere informação para redes de comutação de pacotes através da interface CS e para o MRFP e outros IMS-MGW's através da interface Mb. [13]

A interface CS é utilizada para o envio de fluxos de voz e utiliza o protocolo ISUP ou BICC para a sinalização. A interface Mn é baseada no protocolo H.248, permitindo a gestão dinâmica de recursos, físicos e de transmissão, e com procedimentos adicionais para a gestão das terminações, quer do lado de comutação de pacotes ou do lado de comutação de circuitos, fazendo o controlo do IMS-MGW pelo MGCF. Comparativamente a interface Mn é equivalente à interface Mc, exceptuando procedimentos mais específicos de uma rede de comutação de pacotes, que é usada para o controlo do CS-MGW. [8] [13]

MGCF

O ‘Media Gateway Control Function’ (MGCF) implementa uma máquina de estados que faz conversão do protocolo SIP para ‘ISDN User part’ (ISUP) ou ‘Bearer Independent Call Control’ (BICC) e envia o pedido convertido para a rede de comutação de circuitos. A interacção do MGCF com rede de comutação de circuitos só se dará dentro do mesmo fornecedor de serviços que a rede IMS, para outra rede serão encaminhados pelo BGCF. Para além disso também controla o IMS-MGW usando o protocolo H.248 e interage com o S-CSCF e o I-CSCF usando o protocolo SIP. A sua função no IMS é gerir chamadas de voz de uma rede de comutação de circuitos convertidas para uma rede de comutação de pacotes que entram no seu domínio IMS e passar o controlo dessas chamadas através o MGCF para o CSCF controlando partes dos estados das chamadas relacionadas com o controlo de ligação para os canais de transporte num IMS-MGW e comunicando com CSCF, BGCF e entidades de redes de comutação de circuitos para além de determinar o próximo salto dependendo do número de destino, convertendo-o para um TEL URI, para chamadas vindas de redes de voz tradicionais. Também poderá gerar CDR’s. Será utilizada uma ‘Signaling Gateway’ (SGW) em conjunto com o MGCF no caso da rede de comutação de circuitos utilizar o protocolo de transporte MTP, a SGW fará a conversão entre ambos os protocolos de transporte, MTP e SCTP/IP, caso contrário o SGW não será utilizado. [1] [9] [2] [6] [13]

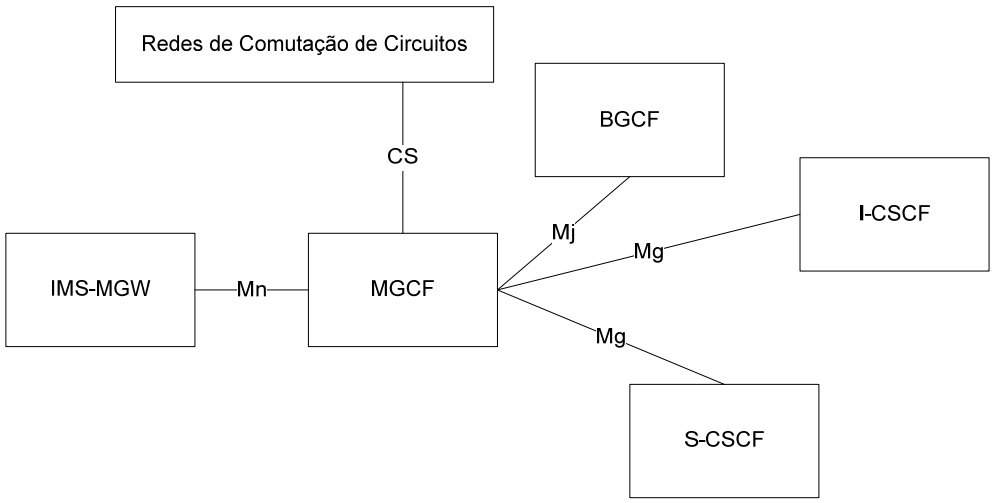

Figura 2.20 – MGCF e suas ligações [13]

O MGCF interage com um BGCF pela interface Mj, com um I-CSCF e um S-CSCF pela interface Mg e com um IMS-MGW pela interface Mn, está também relacionado com a arquitectura de facturação onde poderá enviar informações de facturação para um 'Charging collection Function' (CCF) pela interface Rf. [13] [1]

A interface Mj que é baseada no protocolo SIP permite a troca de sinalização de uma sessão entre um BGCF e um MGCF para a interligação com redes de comutação de pacotes ou em situações de seguimento da sinalização para redes externas. A interface Mg é baseada no protocolo SIP para o envio de sinalização entre o MGCF e o I-CSCF ou o S-CSCF ligando efectivamente um domínio de comutação de circuitos a uma rede IMS. [13] [1]

SGW

O 'Signaling Gateway' (SGW) é utilizado para interligar redes de sinalização que utilizam protocolos de transporte diferentes, numa rede IMS executa a conversão do protocolo de transporte utilizado na sinalização das chamadas sem as interpretar. Permite a transição do protocolo de transporte MTP em SCTP transformando ISUP ou BICC transportado em MTP para ISUP ou BICC transportado sobre SCTP/IP e vice-versa. No caso da rede de comutação de circuitos utilizar o SCTP/IP deixa de ser necessária a sua utilização. [1] [6] [14]

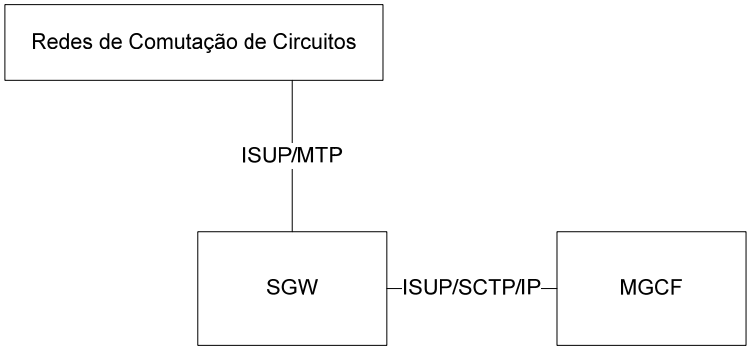

Figura 2.21 – SGW [1] [13]

A SGW troca mensagens de sinalização BICC/ISUP utilizando o protocolo de transporte SCTP/IP com o MGCF e com a rede de comutação à qual está ligada utilizando o protocolo de transporte MTP. [6]

SEG

Os 'Security Gateway' (SEG) são entidades nas fronteiras dos domínios de segurança de uma rede IP, todo o tráfego que sairá do domínio de segurança terá de passar por um SEG e cada um estará configurado para comunicar com apenas um grupo específico de domínios de segurança. São responsáveis pela execução das políticas de segurança para a interligação com outras redes. O tráfego entre SEG's é protegido utilizando IPsec ESP ('Encapsulated Security Payload') e as associações de segurança são estabelecidas e mantidas utilizando IKE ('Internet Key Exchange') para o tráfego dentro do mesmo domínio a troca de informação com cada SEG é feito utilizando IPsec. [1]

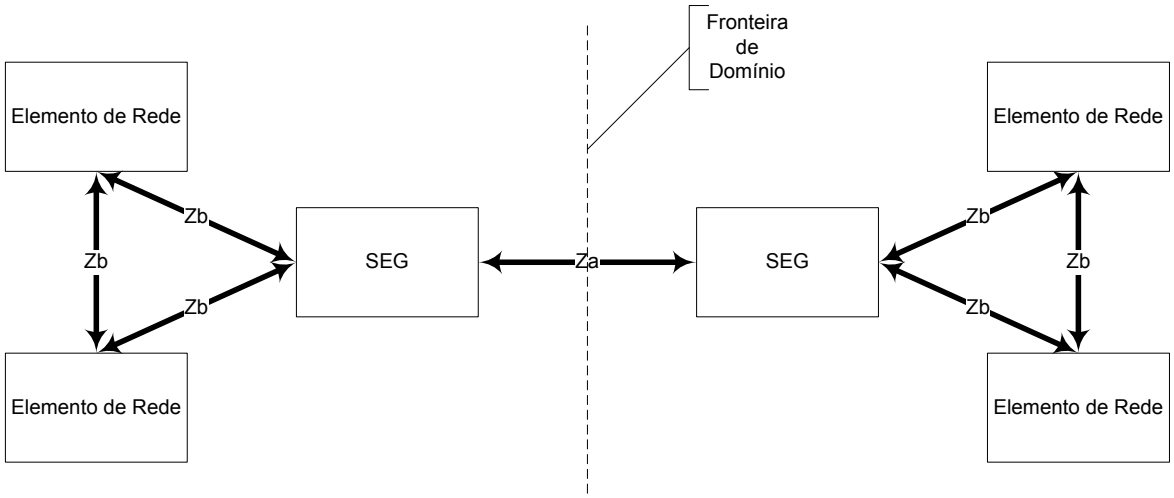

Figura 2.22 – SEG [19]

Um SEG utilizará a interface Za para comunicar com outros SEG's e a interface Zb para comunicar com o CSCF do seu domínio. A interface Za é utilizada para a comunicação entre SEG's de diferentes domínios sendo obrigatória a utilização de autenticação, integridade de dados e codificação utilizando IPsec com ESP. A interface Zb é utilizada para proteger o tráfego entre diferentes SEG's, entre um SEG outra entidade e entre outras entidades dentro do mesmo domínio de segurança transportando apenas tráfego intra-operador, a sua utilização é opcional e caso seja implementada é obrigatória a utilização de funções de integridade de dados sendo a codificação opcional. Utiliza ESP em modo de túnel + IKE sendo o ESP em modo de transporte opcional. [1] [13] [19]

2.4.1.4. MRF

O 'Media Resource Function' é um nó distribuído que fornece serviços de áudio/vídeo na rede natal com capacidade de transmitir mensagens, misturar fluxos de áudio/vídeo, transmutar entre 'codecs' diferentes, obter análises estatísticas e que se divide num 'Media Resource Function Controller' (MRFC) e num 'Media Resource Function Processor' (MRFP) localizando-se sempre na rede natal. O MRFC age como um SIP UA que interage com o S-CSCF usando o SIP e baseado nisso controla os recursos do MRFP através de uma interface usando o protocolo H.248, para além disso também gera CDR's. O MRFP, que é controlado pelo MRFC, fornece e termina fluxos de áudio/vídeo, executa funções de transmissão de anúncios de mensagens e tons, processa os fluxos de áudio/vídeo fornecendo serviços de análise, transmutação e mistura e gere acessos a recursos partilhados num ambiente de conferência. Um exemplo típico será o S-CSCF ou o AS enviar um 'INVITE', requisitando tons ou mensagens, para o MRF que lhe responderá caso seja bem sucedido executando o que foi pedido. [1] [8] [6] [11] [8] [13]

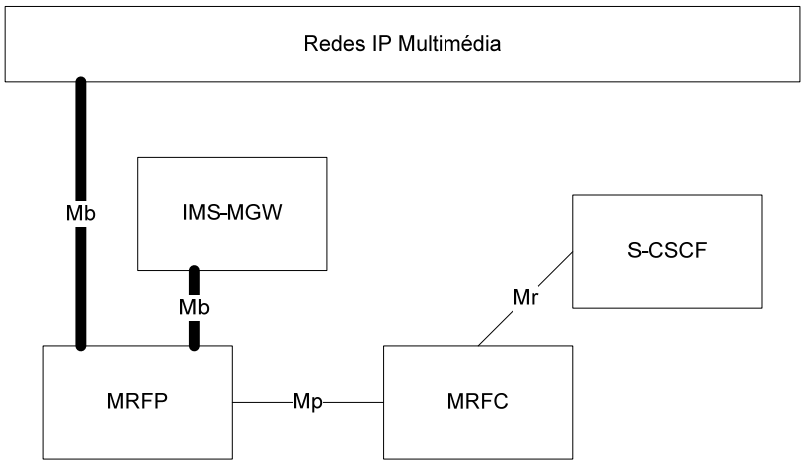

Figura 2.23 – MRF e suas ligações. [13]

O MRFC interage com o S-CSCF através da interface Mr e com o MRFP através da interface Mp. O MRFP troca dados com redes 'IP Multimédia' e com o IMS-MGW usando a interface Mb. [13]

A interface Mr é baseada em SIP e permite a troca de sinalização entre o S-CSCF e o MRFC. A interface Mp é baseada no protocolo H.248 e permite a um MRFC controlar fluxos áudio/vídeo e outros recursos num MRFP. A interface Mb fornece serviços de transporte de dados RTP/UDP/IP numa rede IPv6. [13] [6]

2.4.2. Camada de Serviços

Numa rede IMS existem entidades que implementam e fornecem serviços aos seus utilizadores, essas entidades são os servidores de aplicações, 'Aplication Servers' (AS). Um AS pode operar em 3 modos distintos, em modo SIP 'proxy', SIP UA ou SIP B2BUA e poderá enviar informações de facturação para as entidades adequadas. [1] [8]

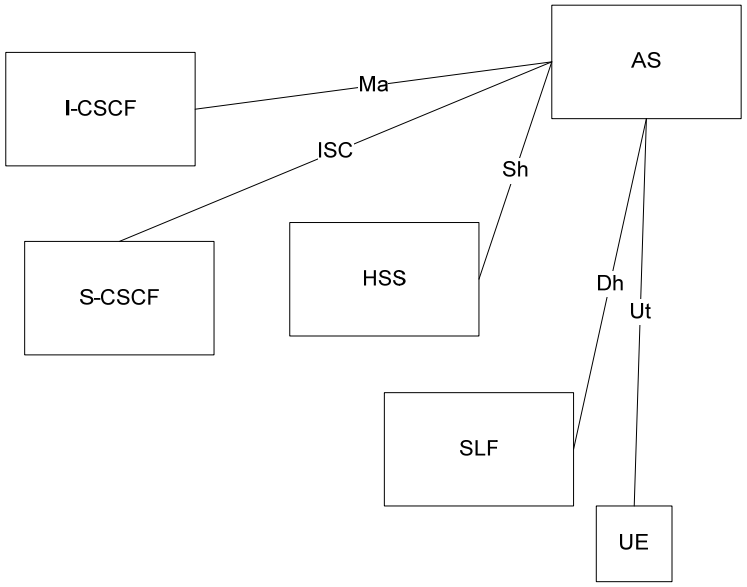

Figura 2.24 – AS e suas ligações. [13]

Um AS pode-se localizar na rede natal ou em qualquer rede externa para a qual o operador tem um acordo de serviço, neste último caso o AS não poderá aceder ao HSS. Um AS interage com o I-CSCF através da interface Ma, com o S-CSCF através da interface ISC, com um HSS através da interface Sh ou Si caso se trate de um IM-SSF, com o SLF (caso exista) através da interface Dh e com um UE através da interface Ut. A interface Ma permite a interligação de um AS com o I-CSCF sendo usada para o reenvio de pedidos SIP destinados a uma identidade contida no serviço que o servidor AS fornece. A interface ISC que significa 'IMS Service Control' é basicamente uma interface SIP com suporte para mais do que uma transacção que permite a interacção da sinalização de chamadas/sessões com a plataforma de serviços cujos procedimentos se dividem em duas categorias: reencaminhamento de um pedido para um AS ou pedidos iniciados pelo AS, o

primeiro sucede-se quando o P-CSCF recebe um pedido SIP e decide que o pedido deverá ser enviado para um AS de modo a ser processado onde o AS por sua vez termina, redirecciona ou transmite o pedido, o segundo relaciona-se com os casos onde o AS inicia um pedido em nome do utilizador para o qual está a executar um serviço. A interface Sh, cuja implementação é opcional, é baseada no protocolo Diameter com um conjunto de novos AVP's permitindo a um AS gerir dados no HSS que mantém uma lista dos AS que estão autorizados a obter ou armazenar dados e também fornece um serviço de subscrição e notificação de alterações no HSS. A interface Dh é utilizada em conjunto com a interface Sh, é baseada no protocolo Diameter e permite a um AS contactar com um SLF que é basicamente um agente de redireccionamento Diameter de modo a saber qual o HSS que deverá contactar para um determinado utilizador. E a interface Ut, que permite a comunicação entre um AS e um UE, é baseada em HTTP1.1 transportando dados em formato 'XML Configuration Access Protocol' (XCAP), esta interface é utilizada apenas para permitir ao utilizador configurar listas de recursos e outros tipos de manipulação de informação. [1] [11] [8]

Existem três tipos de AS's: o SIP AS, o 'Open Service Access-Service Capability Server' (OSA-SCS) e o 'IP Multimedia Service Switching Function' (IM-SSF). Todos se comportam como um AS na rede IMS, a interface para o HSS é opcional e no caso do IM-SSF a interface com o HSS é baseada no 'Mobile Aplication Part' (MAP) em vez do Diameter. Do ponto de vista de um S-CSCF não há diferenciação entre o SIP AS, o OSA-SCS e o IM-SSF. Adicionalmente existe o 'Service Capability Interaction Manager' (SCIM) que fornece serviços de orquestração e gestão de contextos para iterações de serviços mais complexas que interagem com o resto da rede IMS com as mesmas interfaces que um AS típico. [1] [13]

2.4.2.1. SIP AS

Um 'SIP Aplication Server' (SIP AS), é o servidor de aplicações nativo de uma rede IMS para a disponibilização e execução serviços baseados em SIP permitindo a criação e lançamento de novos serviços numa rede IMS. [1]

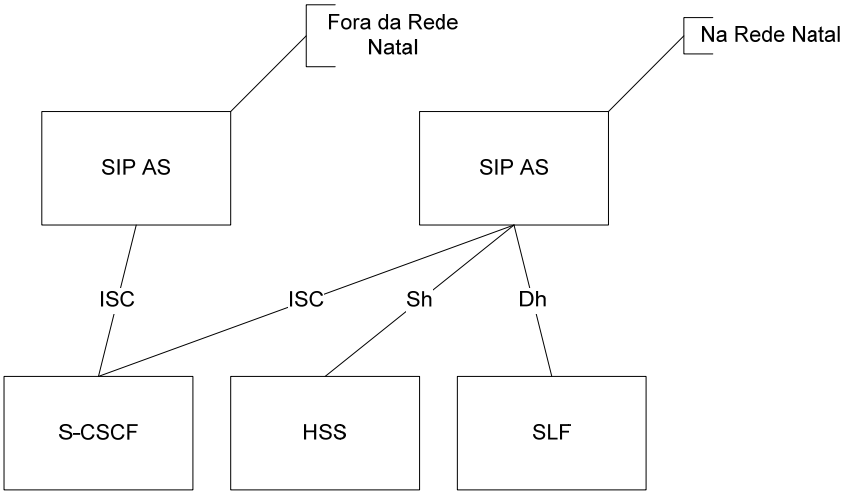

Figura 2.25 – SIP AS e suas ligações. [1]

2.4.2.2. IM-SSF

O 'IP Multimedia Service Switching Function' (IM-SSF) permite a reutilização de serviços CAMEL ('Costumized Applications for Mobile Network Enhanced Logic'), que foram desenvolvidos para o GSM, no IMS. Permite com que um gsmSCF ('GSM Service Control Function') controle uma sessão IMS ou que serviços CAMEL possam ser invocados a partir da rede IMS. Age como um AS típico no lado da rede IMS e como um SSF ('Service Switching Function') no outro. Utiliza a interface Si para comunicar com HSS que se baseia no protocolo MAP ('Mobile Aplication Part'), ao contrário dos outros 2 tipos de AS's que utilizam a Sh, e a sua interacção com o gsmSCF utiliza o protocolo CAP ('CAMEL Aplication Part'). [1]

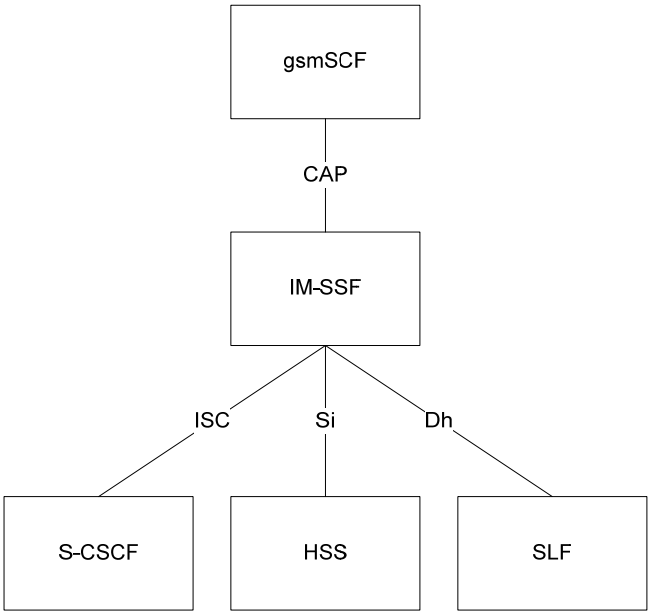

Figura 2.26 – IM-SSF e suas ligações. [1]

2.4.2.3. OSA-SCS

O 'Open Service Access-Service Capability Server' (OSA-SCS) fornece uma interface para o servidor de aplicativos da 'framework' OSA permitindo a execução de serviços OSA numa rede IMS. Actua como um AS típico no lado do IMS e tem uma interface para a API OSA no outro. É deste modo uma 'Gateway' para a execução de serviços OSA. [1]

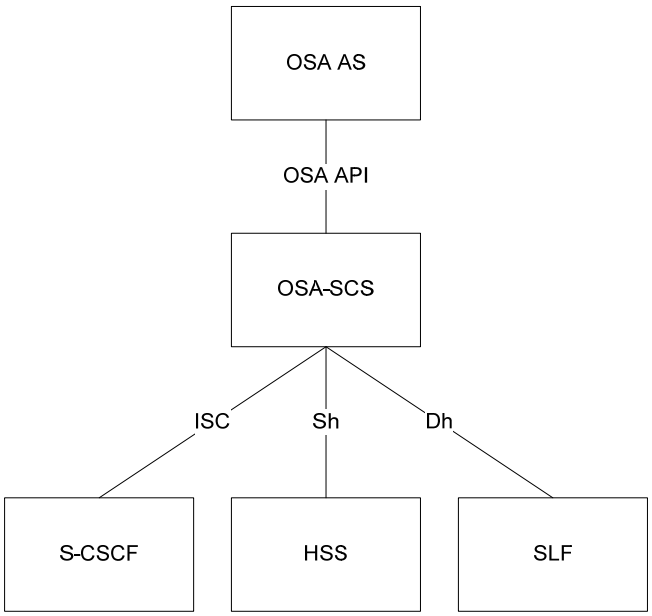

Figura 2.27 – OSA-SCS e suas ligações. [1]

2.4.2.4. SCIM

O 'Service Capability Interaction Manager' (SCIM) é uma entidade cujo papel se centra na gestão das interacções, no interior e com o exterior, da camada de serviços fornecendo serviços de mediação na distribuição de eventos e controlo de serviços entre diferentes AS's, convencionalmente esta funcionalidade está presente no S-CSCF num modelo mais simplificado que não permite a utilização de contexto nem de composição de serviços com outros 'enablers' da mesma camada de serviços. A sua definição presente no 3GPP TS23.002 é muito simples e abrangente criando uma grande indefinição e ao mesmo tempo liberdade que permite com que grande parte das suas implementações sejam proprietárias. Apesar disso pode-se determinar que o SCIM actua em conjunto com o S-CSCF para a sequenciação e invocação das aplicações a serem executadas onde o S-CSCF executa a função de filtragem dos serviços a ser ou não utilizados e o SCIM a mediação em cenários de maior complexidade de interacção entre serviços cujo suporte e disponibilidade depende da filtragem executada pelo S-CSCF. A capacidade de utilizar dados contextuais, num ambiente em que sessões poderão conter outras sub-sessões, permite fornecer uma

experiência melhorada ao utilizador pela orquestração dos diferentes ‘enablers’ disponíveis numa mesma aplicação. [56] [66] [13]

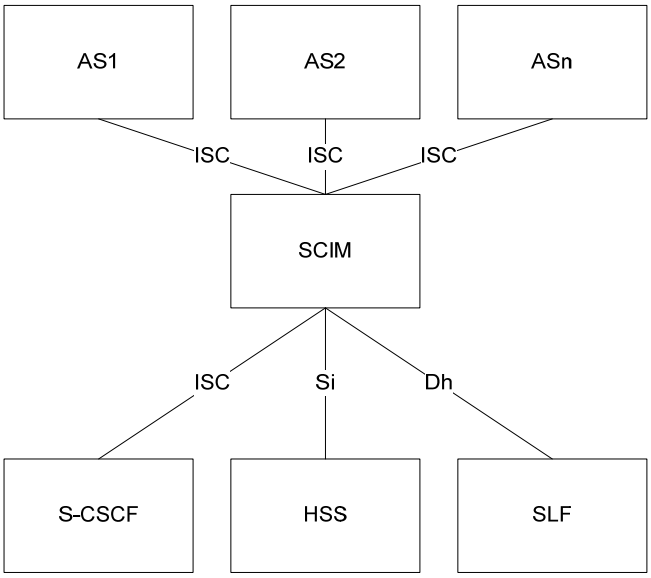

Figura 2.28 – Posicionamento de um SCIM no IMS. [13]

O SCIM está definido como fazendo parte dos AS’s estando apenas definidas as suas interfaces com o seu exterior, desse modo terá de suportar a interface ISC para comunicar com o S-CSCF e com os outros AS’s, a interface Sh para comunicar com o HSS e Dh para comunicar com o SLF, caso seja necessário, sendo visto como outro AS do ponto de vista da rede nuclear. [56]

Devido à sua breve especificação o SCIM deixa um grande espaço de dúvidas sobre a sua aplicabilidade, suas funcionalidades e o futuro que terá à medida que as redes IMS vão evoluindo.

2.5. Segurança

A segurança numa rede IMS é implementada em três áreas: segurança no acesso à rede, segurança no domínio e segurança dos aplicativos. A segurança aplicada no acesso à rede diz respeito à autenticação do utilizador e protecção dos dados trocados entre o terminal e a rede com o objectivo principal de proteger contra usurpação de identidade e ataques DoS ('Denial of Service'), a segurança no domínio diz respeito ao tráfego entre as diferentes entidades na rede ou em redes exteriores que está implementada nas interfaces Za e Zb com o IPsec e a segurança dos aplicativos refere-se à segurança fim-a-fim implementada por cada aplicativo que utiliza a rede. O IMS é essencialmente uma camada arquitectural funcionando sobre uma rede de comutação de pacotes com baixa interdependência, desse modo poderão existir outros esquemas de segurança que podem funcionar a uma camada superior ou inferior dos quais o IMS é independente. [1] [9] [8] [20]

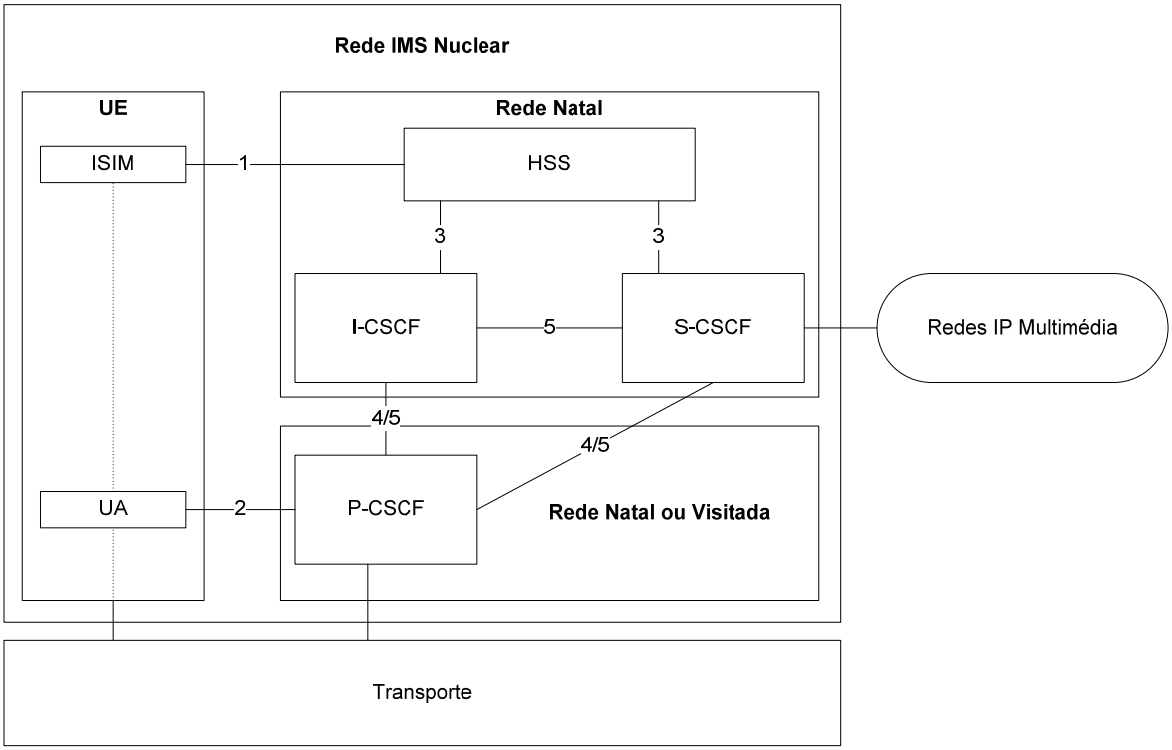

Figura 2.29 – Arquitectura de segurança. [20]

Como se poderá observar na figura 2.29, a segurança no acesso corresponde às ligações ‘1’ e ‘2’ enquanto que as ‘3’, ‘4’ e ‘5’ correspondem à segurança no domínio, ‘Network Domain Security’ (NDS).

2.5.1. Segurança no acesso

Para aceder aos serviços de uma rede IMS um utilizador tem que ser autenticado e autorizado na sua rede natal e terá que ser estabelecida uma associação de segurança entre o UE e o P-CSCF do ponto de vista da rede de acesso. O processo de autenticação, autorização e estabelecimento de associações de segurança, que utilizam IPsec, ocorre simultaneamente utilizando o pedido SIP REGISTER onde o S-CSCF, usando os vectores de autenticação obtidos a partir do HSS, autentica e autoriza o utilizador enquanto que o P-CSCF negoceia dois pares de associações de segurança unidireccionais com o UE que por sua vez também autentica a rede de modo a certificar-se que não se está conectando a uma rede falsa. O processo de autenticação no IMS é de desafio-resposta e utiliza tipicamente o algoritmo IMS AKA, ‘IMS Authentication and Key Agreement’, que permite a autenticação mútua entre a rede e o UE e a partilha de chaves necessária para o estabelecimento das associações de segurança entre o UE e o P-CSCF de modo a proteger o tráfego. Para se registar na rede cada utilizador tem que utilizar uma identidade privada, ‘IM PrIvate identity’ (IMPI), e pelo menos uma identidade pública, ‘IM PUblic identity’ (IMPU), juntamente com uma chave secreta previamente partilhada estando todos armazenados num módulo de identidade, ‘IM Services Identity Module’ (ISIM), que é uma aplicação contida num ‘smartcard’, Universal Integrated Circuit Card’ (UICC), que contém um conjunto de dados de segurança relativos à rede IMS com funções de cálculo e comparação das chaves de autenticação. Numa única subscrição IMS podem haver várias IMPI’s que são associadas a uma ou mais IMPU’s que por sua vez podem ser partilhadas entre as IMPI’s da mesma subscrição. [20] [9] [6] [1] [21]

A geração dos vectores de autenticação que contêm um número aleatório (RAND), uma resposta esperada (XRES), uma chave de encriptação (CK), uma chave de integridade (IK) e uma chave de autenticação, 'Authentication Token' (AUTN), é feita pelo HSS a pedido do S-CSCF, quando recebe um pedido SIP REGISTER de um UE, e após a sua recepção obtém a partir dela um desafio ('challenge') que o UE terá de responder correctamente como se pode observar na figura 2.30. Para além disso existe um contador, 'sequence number' (SQN), individual para cada utilizador, que está guardado no ISIM deste e no HSS que impede a reutilização de um desafio de uma tentativa de autenticação anterior. Existem três situações típicas no processo de autenticação e autorização de um utilizador: autenticação iniciada pelo utilizador bem sucedida (figura 2.30), autenticação iniciada pela rede bem sucedida (figura 2.32), e autenticação falhada que pode ser devido a uma resposta incorrecta ou devido à falta de sincronização (figura 2.31). No caso da autenticação iniciada pela rede é o S-CSCF que enviará um pedido ao UE para iniciar o re-registo e caso este não o faça será automaticamente des-registado da rede. Do ponto de vista das associações de segurança cada registo que inclui uma tentativa de autenticação vai produzir novas associações de segurança que substituirão a antiga já estabelecida caso a autenticação seja bem sucedida como se pode observar na figura 2.30 onde o tráfego protegido pela SA se apresenta em tracejado, na situação de falha de autenticação devido a uma resposta incorrecta haverá também uma falha de integridade IPsec no P-CSCF que por sua vez apaga a nova associação de segurança. No caso de falha na autenticação da rede ou falha de sincronismo o P-CSCF descarta a associação de segurança nova pois o UE enviará outro SIP REGISTER através de uma associação de segurança já estabelecida, caso não haja nenhuma associação de segurança estabelecida o pedido será enviado para os portos 5060/5061. E no caso de uma autenticação incompleta o P-CSCF descarta a associação de segurança a estabelecer e dados do processo de registo iniciado devido à expiração do seu tempo de vida ou por receber um SIP REGISTER de uma nova tentativa de registo. [21] [20]

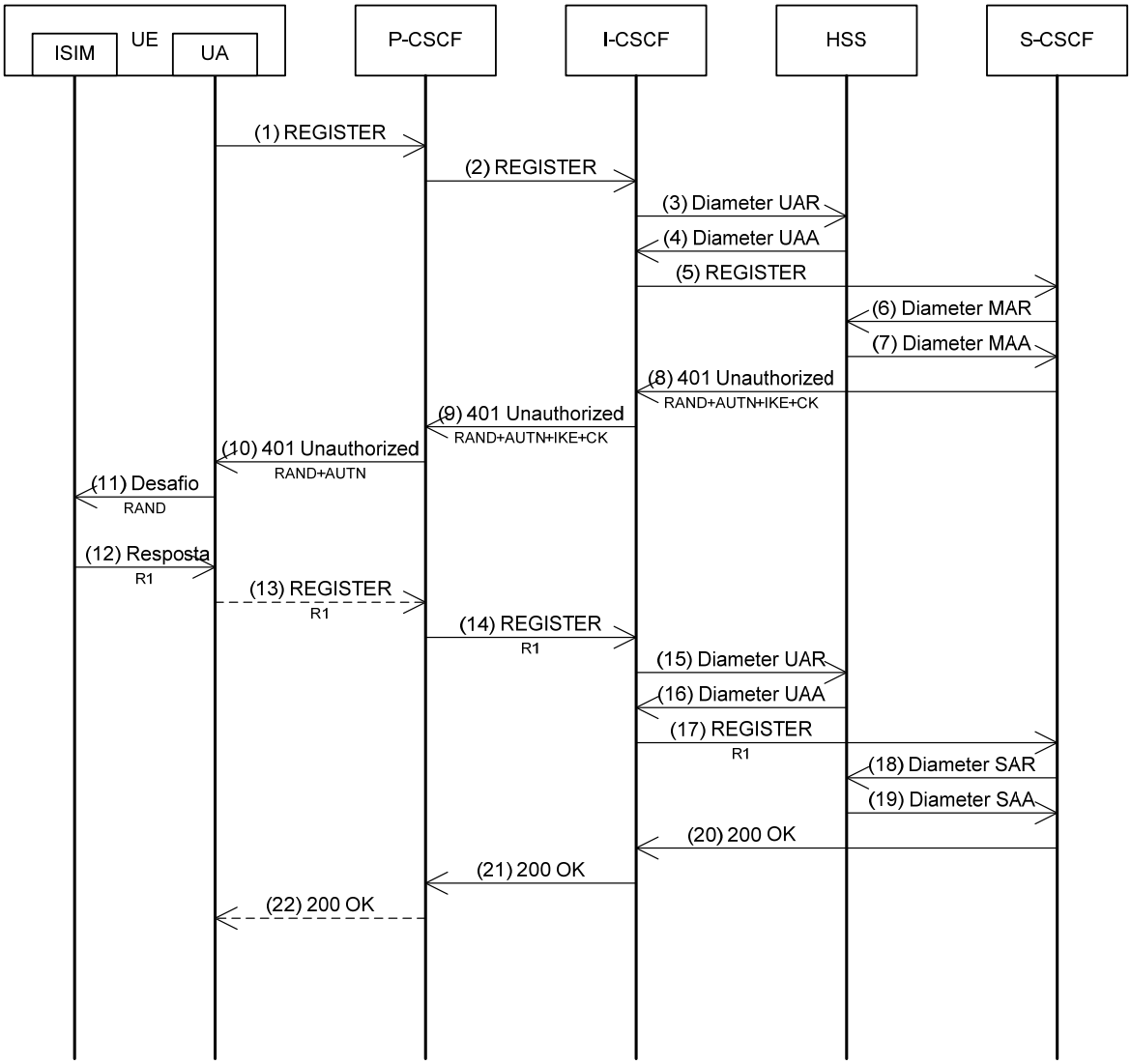

Figura 2.30 – Autenticação bem sucedida (com SA a tracejado). [20] [9] [6] [1]

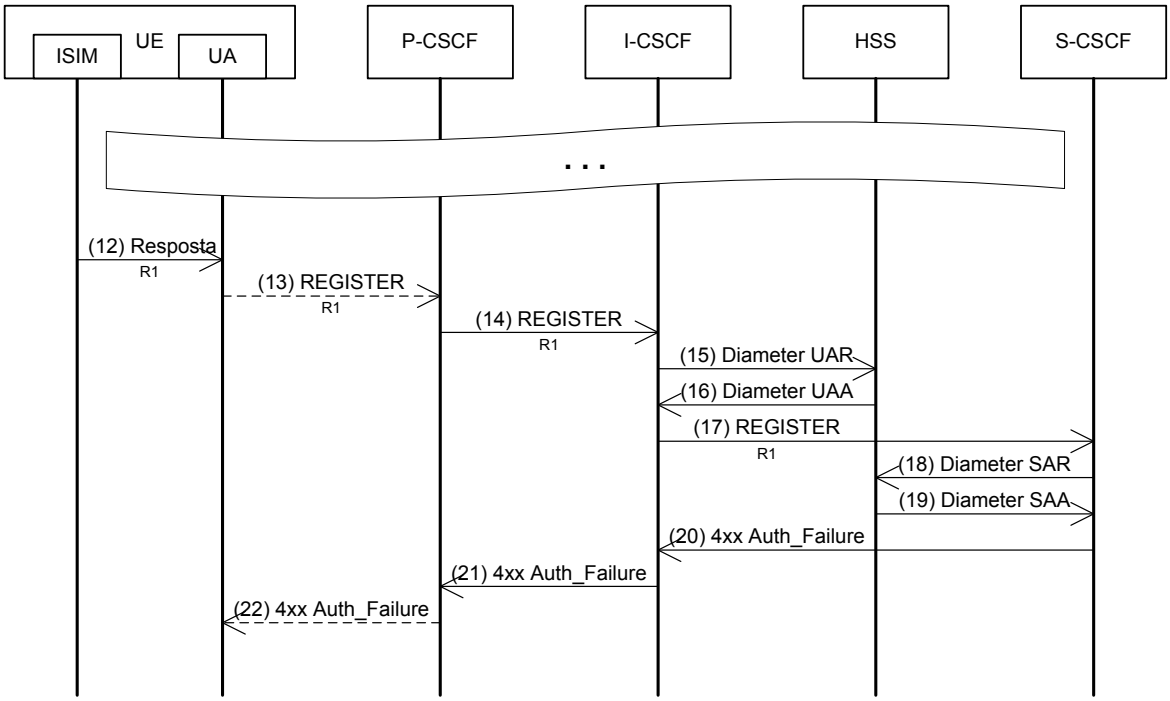

Figura 2.31 – Autenticação falhada por resposta incorrecta ou falta de sincronismo. [20] [9] [6] [1]

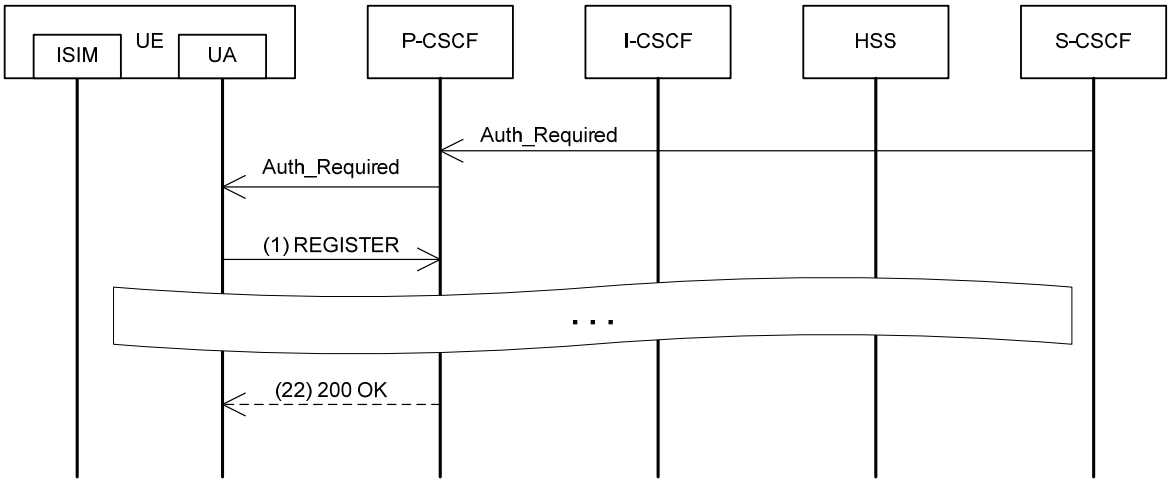

Figura 2.32 – Autenticação iniciada pela rede. [20]

2.5.2. Segurança no domínio

A segurança no domínio numa rede IMS tem como objectivo a protecção do tráfego dentro de um domínio de segurança e entre diferentes domínios de segurança cujos limites tipicamente coincidem com os limites do seu operador. Essa protecção é conseguida através da utilização da segurança de domínios de rede para redes IP, 'Network Domain Security for IP' (NDS/IP), e de uma estrutura de autenticação, 'Network Domain Security Authentication Framework' (NDS/AF). No NDS/IP existem duas interfaces, uma para o tráfego entre diferentes domínios, Za, de utilização obrigatória e outra para o tráfego dentro do mesmo domínio, Zb, cuja implementação é opcional. A NDS/AF fornece a capacidade de autenticação dos SEG's de diferentes domínios de segurança através da interface Za no âmbito do NDS/IP embora possa ser adaptado para as interfaces Zb cuja implementação está à escolha de cada operador. [22] [19] [1] [6] [8]

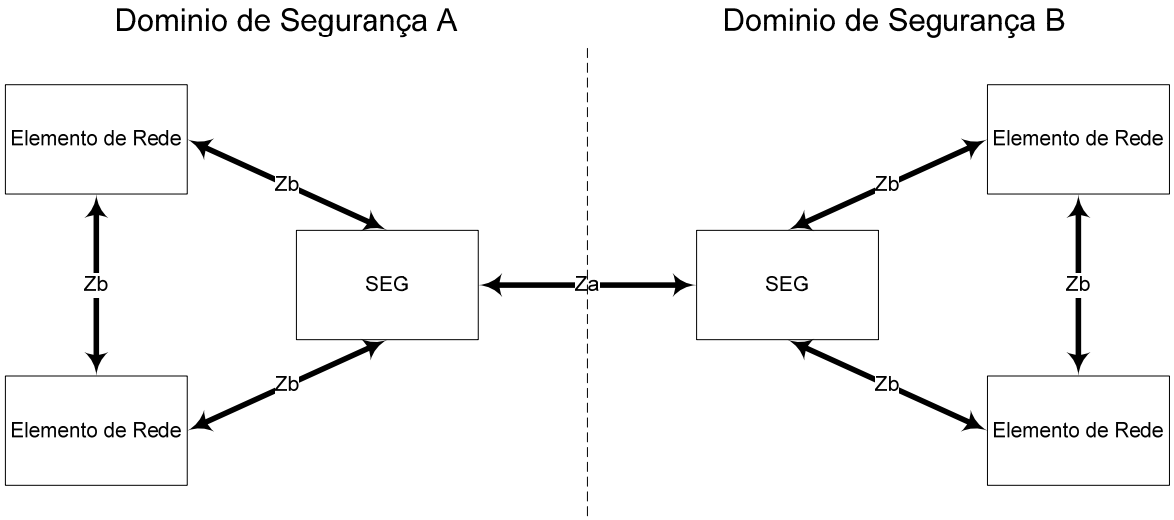

Figura 2.33 – Arquitectura NDS. [22]

A gestão das associações de segurança ESP em ambas as interfaces é feita utilizando IKE. Entre SEG's de diferentes domínios é estabelecido um túnel IPsec com ESP pela interface Za que terá que fornecer, obrigatoriamente, autenticação, protecção de integridade e encriptação do tráfego, todo o tráfego que transita entre diferentes domínios terá de

obrigatoriamente passar por dois SEG's, um de cada domínio. Dentro do mesmo domínio de segurança as diferentes entidades podem comunicar entre si e com os vários SEG's existentes no mesmo domínio usando IPsec pela interface Zb, caso tenha sido implementada, que terá de fornecer obrigatoriamente protecção de integridade e apenas opcionalmente a encriptação de dados. Apenas os SEG's comunicarão com entidades fora do domínio de segurança e cada domínio terá mais do que um SEG por motivos de gestão de falhas. As políticas de segurança estabelecidas pela interface Za poderão estar sujeitas a acordos de 'roaming' ao contrário das politicas de segurança da interface Zb que estão dependentes das decisões do seu operador e a comunicação entre diferentes domínios de segurança é feita sempre através dos SEG's e nunca directamente entre outros elementos de rede de domínios diferentes. [22] [19] [1] [6] [8]

2.6. Facturação

A facturação no IMS é suportada pelo protocolo Diameter e uma arquitectura específica que permite outros modelos de facturação para além da facturação por volume ou por tarifa fixa que são a facturação por sessão, evento ou serviço. Isso é muito importante pois permite com que novos modelos de serviços possam ser disponibilizados aos utilizadores apresentando novas possibilidades para as operadoras. Existem 2 tipos de facturação que são suportados por duas arquitecturas de facturação diferentes, a facturação pré-paga que é suportada por uma arquitectura de facturação 'online' e a facturação pós-paga suportada por uma arquitectura de facturação 'offline'. A facturação 'online' necessita de um sistema de verificação e actualização das contas dos utilizadores antes de permitir a utilização de um serviço, enquanto que o sistema 'offline' necessita de um sistema de recolha de dados de facturação para mais tarde cobrar ao utilizador. O protocolo Diameter é utilizado por todas as entidades envolvidas para comunicar detalhes de facturação e informações de utilização a um sistema centralizado que por sua vez agrega todos os dados e constrói um registo de informação de facturação, 'Charging Data Record' (CDR). [8] [2]

2.6.1. Arquitectura ‘Offline’

A facturação ‘offline’ é tipicamente utilizada para cobrar periodicamente serviços a utilizadores com uma subscrição ou assinatura permanente. Tal como na arquitectura ‘online’ vai depender dos ‘Charging Trigger Function’s’ (CTFs) que estarão presentes nas várias entidades da rede para obter informações de facturação que são recolhidas pelo ‘Charging Data Function’ (CDF) pela interface Rf que criará um CDR por cada evento ou por múltiplos eventos ocorridos no mesmo elemento de rede segundo um conjunto de regras definidas pelo operador e as envia para um ‘Charging Gateway Function’ (CGF) pela interface Ga. O CGF que pode ser integrado no mesmo elemento de rede que o CDF age como um ‘gateway’ entre a rede IMS e o domínio de facturação enviando-lhes os CDRs criados pelo CDF através da interface Bi podendo também ter algumas funcionalidades de pré-processamento de CDRs e gestão de erros. Como estão envolvidos vários elementos de rede para uma mesma sessão haverão múltiplos CDRs de cada um deles, então de modo a identificar a sessão é utilizado um parâmetro global único, não reutilizável por um período não inferior a 1 mês, ‘IMS Charging Identifyer’ (ICID), no cabeçalho ‘P-Charging-Vector’ do pedido SIP que é criado e adicionado à mensagem pelo primeiro elemento que o recebe, normalmente o P-CSCF, que estará presente em todas as mensagens SIP subsequentes identificando a informação de facturação relacionada com essa transacção. Para além disso o cabeçalho ‘P-Charging-Vector’ poderá conter um ‘Inter Operator Identifier’ (IOI) e um identificador de facturação da rede de acesso. O IOI consiste em dois parâmetros no ‘P-Charging-Vector’, o ‘orig-ioi’ e o ‘term-ioi’, que contêm o identificador da rede origem e destino da sessão respectivamente, cada um é gerado por cada lado da rede permitindo com que se identifique a rede originária da chamada e a rede destino de modo a que ambas possam partilhar informações de facturação. E o identificador de facturação da rede de acesso que está presente no parâmetro ‘access-network-charging-info’ permite a correlação da informação de facturação da rede de acesso e a rede IMS. Para situações em que os participantes de uma sessão pertencem à mesma rede e nenhum está em situação de ‘roaming’, o que significa que o P-CSCF e o S-CSCF de cada um se encontram na mesma rede reportando os eventos de facturação para os mesmos CCFs, é utilizado mais um cabeçalho SIP, ‘P-Charging-

Function-Address', por um dos S-CSCF envolvidos para enviar os endereços dos CCF's para os outros P-CSCF e S-CSCF, também os P-CSCF enviam ao S-CSCF que contactam, no 'P-Charging-Vector', informações acerca do modo com que elementos da rede de acesso estão a gerir a sessão. [23] [2] [6] [1] [27]

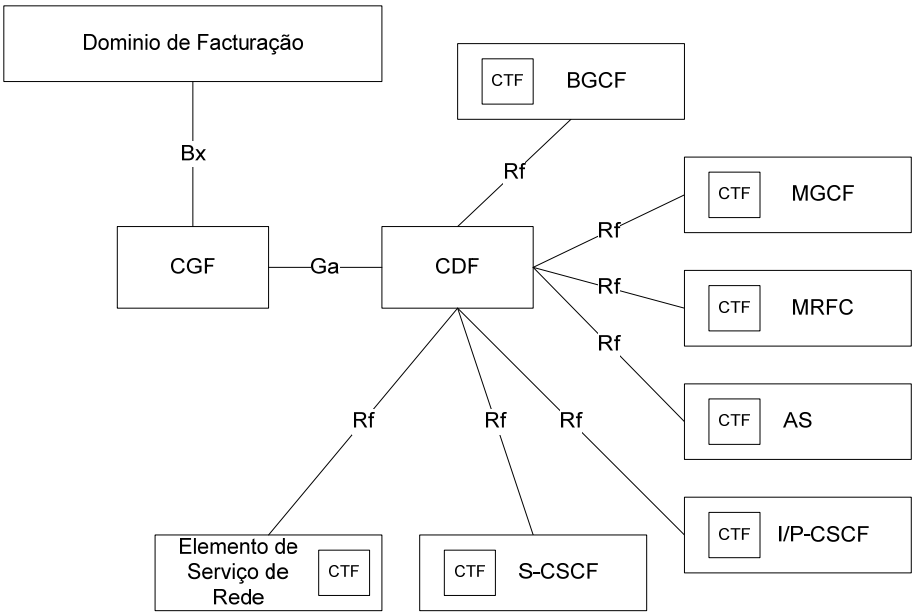

Figura 2.34 – Arquitectura 'Offline'. [6] [2]

Como se pode observar na figura 2.34 com a arquitectura de facturação 'offline' a interface Rf, permite a interacção entre o CTF e um CDF transportando eventos de facturação 'offline' e confirmações da sua recepção, nesta interface é utilizado o protocolo Diameter que tem que suportar transacções em tempo real com mecanismos próprios de retransmissão e fiabilidade e modos de operação 'statefull' e 'stateless'. Do ponto de vista do Diameter o CTF não é nada mais do que um cliente Diameter e o CDF um servidor Diameter. A interface Ga permite a interacção entre um CDF e um CGF transportando CDRs e confirmações das recepções, o protocolo de transporte utilizado é derivado do 'GPRS Tunneling Protocol' (GTP) chamado de 'GTP Prime' (GTP') que terá de suportar transacções quase tempo real, o envio de múltiplos CDRs numa só mensagem, comutação para um destinatário alternativo caso o primeiro não esteja disponível e mecanismos próprios de retransmissão e fiabilidade. A interface Bx permite a interacção entre um CGF

e o domínio de cobrança através da transferência de CDRs em formato de ficheiro, para isso é recomendada a utilização de um protocolo simples de transferência de ficheiros tal como o FTP. [23] [26]

2.6.2. Arquitectura 'Online'

A facturação 'online' permite um controlo do crédito do utilizador impedindo com que aceda aos serviços caso não tenha crédito suficiente. A arquitectura 'online' tem como elemento central de decisão o 'Online Charging System' (OCS) que vai depender dos vários 'Charging Trigger Function' (CTF) que estão contidos nos elementos de rede e para lhe fornecer as informações de facturação. O OCS é constituído por um 'Online Charging Function' (OCF), um 'Account Balance Management Function' (ABMF) e uma 'Rating Function' (RF). O OCF interage com os CTFs recebendo informações de facturação e consiste de dois módulos: o 'Session Based Charging Function' (SBCF) que é responsável pela facturação de sessões individuais e o 'Event Based Charging Function' (EBCF) que executa a facturação para conteúdos ou serviços fornecidos por um AS ou outro elemento de fornecimento de serviços. O ABMF cuja implementação é dependente do operador contém os dados relativos a créditos ou valor da conta do subscritor. E o RF executa a determinação das taxas unitárias, valor monetário ou unidades de crédito, para um volume de unidades, tempo ou eventos antes e depois do fornecimento de serviços e aplicações. Devido ao funcionamento em tempo real da facturação 'online', não são necessários CDR's pois o valor da conta é verificado e actualizado pelo ABMF antes do fornecimento do serviço. Tipicamente existirão múltiplas entidades envolvidas na mesma sessão e para a identificação das sessões e eventos a facturar é utilizado um identificador na sinalização SIP, o 'IMS Charging Identifyer' (ICID) que é um parâmetro no cabeçalho 'P-Charging-Vector'da mensagem, para além disso é utilizado o parâmetro CCF para fornecer os endereços dos CDFs e o parâmetro ECF para o endereço do OCS no cabeçalho 'P-Charging-Function-Addresses'. Quando uma sessão necessita de ser terminada por falta de

crédito o IMS-GWF comunica com o S-CSCF pela interface ISC terminando-a. [23] [6] [25] [24]

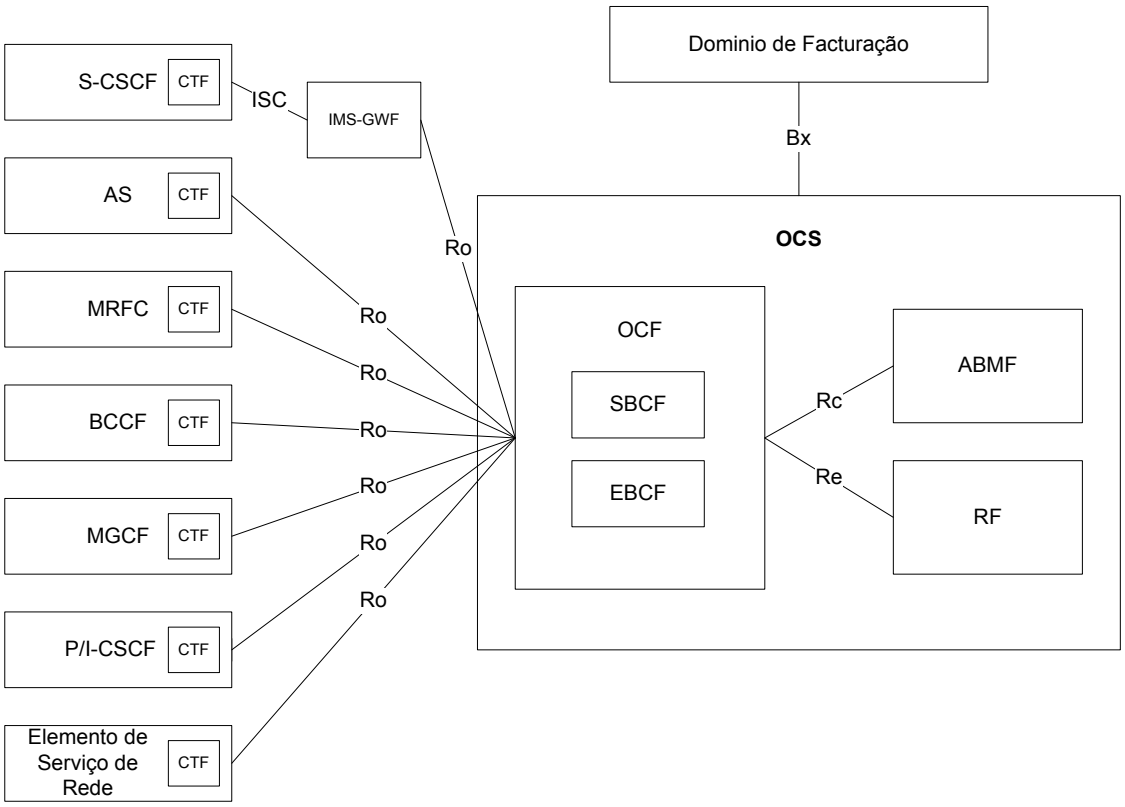

Figura 2.35 – Arquitectura 'Online'. [2] [1] [6]

A interface Ro é análoga à interface Rf, permite a interacção entre um CTF e um OCF pela transferência de eventos de facturação do CTF para o OCF e confirmações da sua recepção, o protocolo utilizado para o efeito terá de suportar transacções em tempo real com mecanismos próprios de retransmissão e fiabilidade e modos de operação 'stateless' e 'statefull', para esse efeito é utilizada a aplicação de controlo de crédito do Diameter, 'Diameter Credit Control Application' (DCCA). A interface Rc permite a interacção entre um OCF e um ABMF fornecendo acesso à conta do subscritor. A interface Re suporta a interacção entre um OCF e um RF permitindo a obtenção do valor, monetário ou não, dos eventos a facturar e é baseada no protocolo Diameter. [23] [25] [26]

2.7. Qualidade-de-Serviço (QoS)

A capacidade de permitir a negociação de uma qualidade de serviço, 'Quality of Service' (QoS), é um componente importante no IMS. Os valores para uma sessão com QoS serão determinados por factores como a largura de banda máxima que poderá ser reservada para o utilizador baseado na sua subscrição, podendo haver distinções entre diferentes grupos de utilizadores, ou no estado actual da rede se actualmente está mais sobrecarregada ou se está previsto haver mais tráfego naquela altura do dia. A especificação 3GPP TS23.207 define a plataforma para a QoS fim-a-fim em GPRS complementando a especificação 3GPP TS23.107 que define a plataforma para a QoS em UMTS, o objectivo não é discutir estas especificações devido ao enfoque na independência do IMS à rede de acesso, em vez disso apresenta-se brevemente o mecanismo que permite à estrutura presente na rede de acesso efectuar reservas dos recursos de rede para aplicar uma QoS sendo que o IMS não garante a QoS, em vez disso permite com que uma estrutura de rede sob si possa receber informações do P-CSCF que ao escutar a sinalização SIP/SDP entre os utilizadores instrui um 'Policy Decision Point' (PDP) de modo a que possa reservar os recursos necessários para o estabelecimento das sessões suportando vários modelos de QoS. Os dispositivos terminais poderão utilizar protocolos a nível da camada de ligação tais como PDP, 'Resourse Reservation Protocol' (RSVP) ou o 'Differentiated Services' (DiffServ) enquanto que as redes utilizam RSVP ou DiffServ. Um mecanismo implementado no SIP/SDP é a utilização de uma etiqueta 'preconditions', que podem ser de acesso ou fim-a-fim, quando um terminal recebe uma oferta com esta etiqueta só responderá ao utilizador, com um pedido SIP UPDATE, quando as condições de QoS actuais são iguais ou melhores que as pré-condições exigidas presentes no SDP. A informação de QoS poderá ser transmitida aos terminais pelo P-CSCF que modificará os campos adequados presentes no SDP do corpo das mensagens SIP. [1] [11] [10]

2.8. 3GPP versus 3GPP2

O ‘Third Generation Partnership Project’ (3GPP) e o ‘Third Generation Partnership Project 2’ (3GPP2) foram iniciados em paralelo em 1998 para coordenar requisitos e criar especificações para a terceira geração de comunicações sem fios a partir da rede GSM (mercados Europeus) e IS-95 (mercados Norte Americanos e Asiáticos) respectivamente. O 3GPP está organizado num ‘Project Co-ordination Group’ (PCG) que é responsável pela sua gestão e vários ‘Technical Specification group’ (TSG) responsáveis pelo trabalho técnico enquanto que o 3GPP2 tem um Steering Committee (SC) análogo ao PCG e vários TSGs, ambos produzem relatórios técnicos, ‘Technical Reports’ (TR), e especificações técnicas, ‘Technical Specifications’ (TS) criando cada um a sua versão do IMS. A semelhança de objectivos tais como a interoperabilidade IPv4/IPv6 e migração para IPv6, fiabilidade e qualidade de serviço ao nível das redes tradicionais e suporte de dispositivos abrangentes para além o facto de ambos os grupos colaborarem com o IETF e utilizarem os chamados protocolos Internet obtiveram-se duas redes bastante semelhantes. [11] [1] [4]

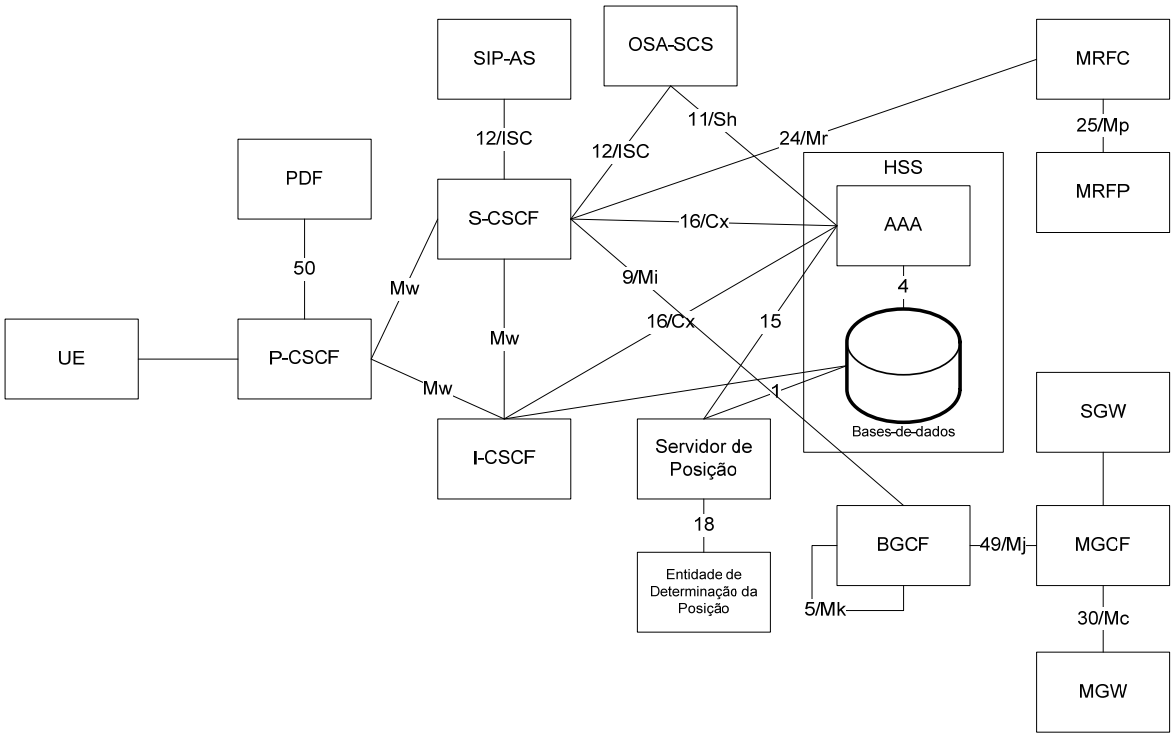

Figura 2.36 – Arquitectura 3GPP e 3GPP2 sobrepostas. [1] [31]

Como se pode observar na figura 2.36 onde se sobrepõem de um modo generalizado ambas as arquitecturas, as diferenças passam pelas nomenclaturas a alguns conceitos e convenções. O 3GPP2 IMS utiliza o conceito de um domínio multimédia, 'MultiMedia Domain' (MMD) que se divide num sistema de comutação de pacotes, 'Packet Data System' (PDS), que fornece conectividade IP no IMS fornecendo as funcionalidades IMS semelhantes ao 3GPP IMS. A rede de acesso no 3GPP2 IMS baseia-se no CDMA2000 e no 3GPP2 'Wireless IP Network' enquanto que o 3GPP IMS é agnóstico à rede de acesso havendo um foco nas redes WCDMA e GPRS sendo o 'PDP context activation' do GPRS para a descoberta do P-CSCF não suportado no 3GPP2 IMS, para além disso o P-CSCF no 3GPP IMS tem que estar sempre localizado na mesma rede que o GGSN, ao contrário do 3GPP2 IMS onde o P-CSCF e o análogo ao GGSN, o 'Packet Data Service Node' (PDSN), podem se situar em diferentes redes, também a ancoração destas entidades de suporte da rede de acesso com o UE no 3GPP IMS, que é feita antes do registo, é fixa enquanto que no 3GPP2 IMS pode mudar mesmo que esteja a decorrer uma sessão. Em relação às interfaces, no 3GPP2 IMS são identificadas numericamente e a sua correspondência com as interfaces do 3GPP IMS é quase directa embora haja excepções, a interface Go não é suportada no 3GPP2 IMS e a interface Sh pode fornecer informações de localização que são especificas ao 3GPP IMS e não são suportadas no 3GPP2 IMS, também devido à utilização do conceito 'all-IP' o 3GPP2 IMS não tem sistemas herdados de CAMEL não havendo o equivalente à interface Si entre o HSS e o IM-SSF. Para o armazenamento dos dados de configuração e segurança no UE para o 3GPP IMS é utilizada uma aplicação USIM ou ISIM num UICC enquanto que no 3GPP2 IMS pode ser armazenada directamente no dispositivo ou num 'Removable User Identity Module' (R-UIM). Existe também uma diferença dos formatos predefinidos, os codecs de voz são diferentes em ambas as redes, AMR no 3GPP IMS e EVRC e SMV no 3GPP2 IMS. Em relação às entidades, o 3GPP2 IMS tem entidades explícitas para o AAA que agregadas correspondem ao HSS no 3GPP IMS. Na mobilidade a gestão está implícita no 3GPP2 IMS pois está construído em cima do conceito de IP móvel enquanto que no 3GPP IMS a gestão de mobilidade é feita através da sua rede de acesso para além disso o 3GPP2 IMS utiliza um servidor de posição 'Position Server' e uma entidade de determinação da posição 'Position Determining Indentity' (PDE) que não existem no 3GPP IMS. [11] [1]

3. SDP/SOA

3.1. Introdução

As redes IMS fornecem a capacidade de controlo de sessões multimédia e de fornecimento de serviços para os seus utilizadores mas as suas especificações apenas definem um conjunto de interfaces para os serviços deixando um grande espaço em branco sobre como aproveitar todas as capacidades e recursos da rede apresentando-se principalmente como uma plataforma de sinalização onde assentarão outras plataformas de aplicações e serviços. É então neste ponto que entram novos conceitos de gestão e suporte de processos e serviços. O SOA, 'Service Oriented Architecture', é um desses conceitos que foi criado para lidar com processos distribuídos de negócio prometendo agilizar a criação e gestão de serviços baseando-se nos conceitos de reutilização e continuo melhoramento. Como SOA é um paradigma e não uma plataforma de suporte para aplicar os seus conceitos será necessário o desenvolvimento de uma plataforma de entrega de serviços, 'Service Delivery Platform' (SDP), tratando-se assim de um conjunto de princípios arquitecturais que que poderão aplicar os conceitos presentes no SOA. Uma plataforma SDP ideal para utilizar será a 'OMA Service Environment' (OSE) que se apresenta como uma arquitectura para a gestão e execução de 'enablers' OMA que poderão fornecer vários serviços compostos por funcionalidades e serviços mais pequenos e actuar em conjunto permitindo a reutilização de funcionalidades e serviços seguindo os conceitos OSA e utilizando interfaces estandardizadas que permitem uma grande interoperabilidade. Ao longo deste capítulo falar-se-á acerca de SOA, seus conceitos e metodologias, da arquitectura do SDP, seus benefícios, e de OSE como a realização de um SDP. Em seguida analisar-se-ão API's resultantes dos esforços pelas operadoras de expor os seus serviços através da Internet para os seus assinantes comparando as funcionalidades que cada uma permite. Finalmente aplicar-se-ão estes conceitos a uma rede IMS tentando criar uma arquitectura conjunta trazendo para o IMS os conceitos de SOA e mais especificamente a plataforma OSE que serão necessários para aproveitar as suas potencialidades como plataforma de sinalização e suporte de serviços e desse modo tornar mais atractiva e rentável a sua implementação.

3.2. ‘Service Oriented Architecture’ (SOA)

3.2.1. Definição

O ‘Service Oriented Architecture’ (SOA), em português, arquitectura orientada a serviços, não é uma tecnologia nem um produto, é um paradigma arquitectural orientado à integração baseada no conceito de serviços. Foi desenvolvido para lidar com processos de negocio, distribuídos por um grande conjunto de sistemas heterogéneos, fazendo com que esses sistemas ajam como um só, o que implica que não se possa comprar um produto, ferramenta ou aplicação que implemente um sistema novo em vez disso terão de ser feitas decisões adequadas ao contexto de cada sistema seguindo as directrizes deste paradigma. [35] [34] [38]

SOA é deste modo uma infra-estrutura de suporte de sistemas de informação que, focando-se nos processos de negócio, flexibiliza as empresas não estando dependente de nenhuma tecnologia nem metodologia concreta. [35] [34] [38]

3.2.2. Conceito e Aplicabilidade

A razão principal para implementar SOA é de que ao flexibilizar e optimizar os vários processos dos negócios de uma empresa optimiza-se consequentemente o funcionamento e produtividade dessa empresa. Para atingir esse objectivo tem que se lidar com os vários papéis de cada elemento de um processo e o papel desse processo na empresa, daí a utilização do conceito de serviços tomando a forma de serviços ‘web’ pois fornecem uma maior interoperabilidade e algum nível de protecção do código que está por detrás e o facto de poderem ter uma vinculação fraca (‘loosely coupled’) pela utilização de um descritor, o mais comum é o WSDL, que é disponibilizado e deste modo publicando uma interface. [35] [34] [38]

Assim SOA tem três conceitos técnicos: serviços, interoperabilidade e vinculação fraca. Um serviço é uma abstracção de uma funcionalidade de negócio ou processo de uma empresa através do qual se estruturam os sistemas distribuídos, é o bloco de construção em SOA. A interoperabilidade é a base para a implementação dos serviços que estão dispersos por vários sistemas. E a vinculação fraca é crucial para a versatilidade pois é com ela que se conseguem minimizar as dependências entre os vários serviços permitindo reutilização e uma maior flexibilidade quando são necessárias alterações a processos ou serviços contribuindo para uma maior tolerância a falhas. [35] [34] [38]

Para definir que tipos de serviços deverão de ser suportados para implementar SOA, está definida uma arquitectura de referência que permite o planeamento adequado de cada fase. Essa arquitectura de referência está presente na figura seguinte e inclui as características a implementar independentemente dos produtos ou serviços utilizados. [35] [34] [38]

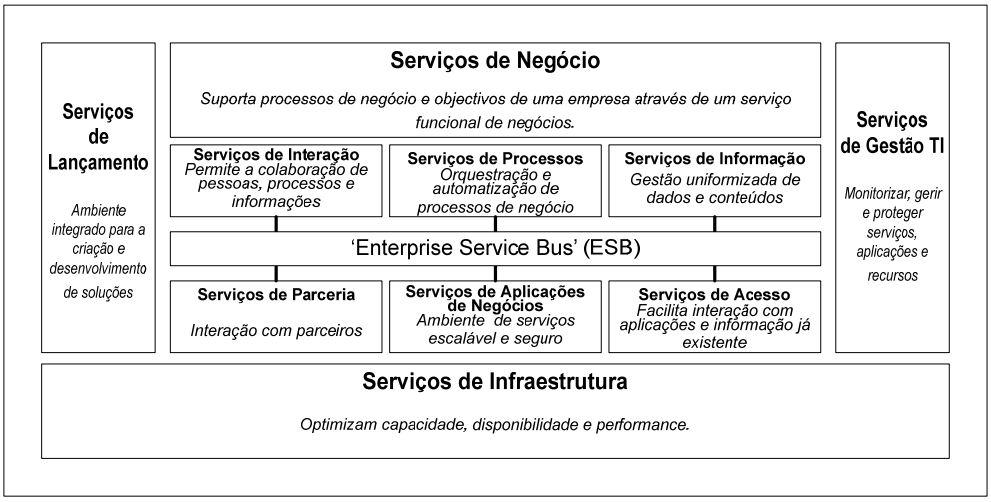

Figura 3.1 – Arquitectura referência SOA. [35]

Cada característica deverá ser implementada de um modo faseado pois uma implementação não estruturada ou demasiado rápida poderá por em causa toda a transição. Das várias características presentes na arquitectura referência (figura 3.1) destacam-se as mais importantes: ESB, serviços de parceria, serviços de aplicações de negócios, serviços de lançamento e serviços de gestão de tecnologias de informação (TI). O ESB é um

componente mediador que fornece o suporte para a implementação de SOA através da integração e gestão dos serviços, da desvinculação de cada serviço com a sua implementação e dos aspectos técnicos das interacções entre os vários serviços tais como capacidades de encaminhamento de mensagens entre os vários serviços, conversão de protocolos e formatos de mensagem, gestão de eventos de negócios de múltiplas fontes e certificação da qualidade-de-serviço baseada em segurança, fiabilidade e iterações de cada transacção. Os serviços de parceria fornecem os protocolos, as definições de interacção e gestão de parceiros que são necessários para os processos que necessitam de interacções com outros parceiros e outros fornecedores de serviços. Os serviços de aplicações de negócios permitem que novos componentes sejam integrados no sistema. Os serviços de lançamento que são necessários para o desenho, implementação e lançamento de novos sistemas. E finalmente, os serviços de gestão de tecnologias TI que fornecem a escalabilidade e performance necessária através da utilização correcta dos recursos existentes. [35] [34] [38]

Para o desenvolvimento e integração de SOA deverá começar-se pela implementação do seu ciclo de vida que, como se pode observar na figura seguinte (figura 3.2), começa com a fase de modelação onde são reunidos os requisitos para os processos e é feito o seu desenvolvimento, simulação e optimização. Segue-se a fase da construção onde o processo é implementado segundo as características definidas na fase de modelação. Depois de implementado o processo é posto em acção, é lançado, e nesta fase é feita a certificação de que todos os elementos do processo estão correctamente ligados entre si e a cooperar correctamente. Na última fase, fase de gestão, o processo é monitorizado de modo a averiguar possíveis melhorias ou alterações que possam ser feitas face a novos requisitos que poderão entretanto ter surgido. Este ciclo de vida é assim fechado criando um processo contínuo de melhoramento flexibilizando e abrindo as portas à inovação a qualquer sistema que se baseie em SOA adaptando-se continuamente a novos requisitos e oportunidades. [35] [34] [38]

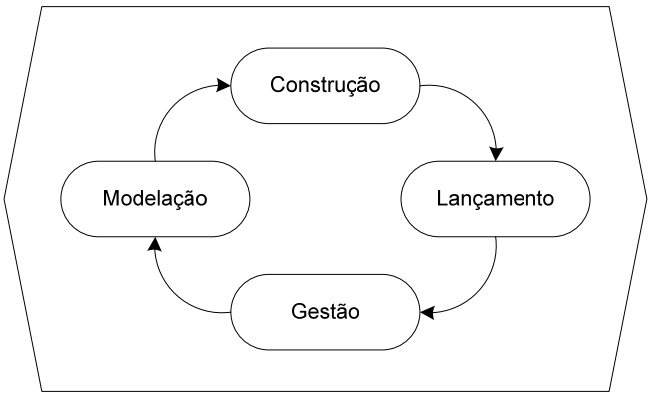

Figura 3.2 – Ciclo de vida SOA. [35]

Por debaixo da camada de abstracção do ciclo de vida de SOA está a camada de governança, 'SOA Governance', cujo objectivo é agilizar o processo de fazer decisões, não do ponto de vista de gestão onde se fazem e implementam as decisões, mas sim determinar quem faz essas decisões. Assim a governança em SOA (figura 3.3) começa pelo planeamento onde se irá determinar uma necessidade a ser correspondida, seguidamente terão de ser definidos os processos de governança, o desenho de políticas, mecanismos que as executam e o estabelecimento do que é considerado um sucesso para cada um dos processos. Após esta definição tudo é posto em prática de um modo faseado como já foi estabelecido no estágio anterior. Finalmente o novo, ou melhorado, processo é posto em prática, sendo monitorizado e gerido com consequente avaliação para determinar o nível do seu sucesso. [35] [34] [38]

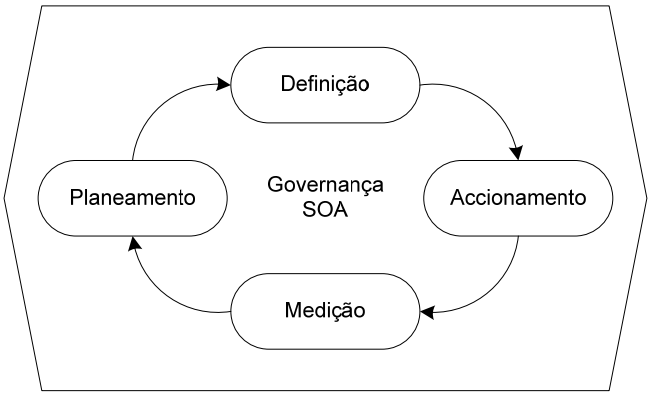

Figura 3.3 – Governança SOA como base para o ciclo de vida SOA. [35]

A governança SOA está envolvida no ciclo de vida de SOA certificando-se que este é executado correctamente quando os serviços são desenvolvidos e postos em acção, depois disso, a governança SOA irá monitoriza-los do ponto de vista de politicas operacionais previamente estabelecidas pelo seu método de governança. [35] [34] [38]

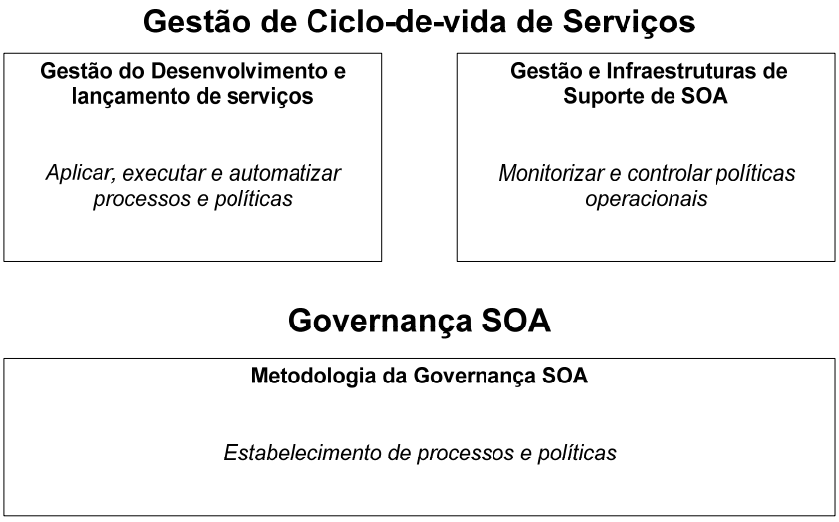

Figura 3.4 – Metodologia de governança SOA [35]

A gestão, pela governança SOA, das fases de modelação, construção e lançamento corresponderá à gestão do ciclo de vida do serviço e divide-se em duas secções (figura 3.4): secção de gestão do desenvolvimento e lançamento de um serviço e secção de suporte e gestão de SOA numa determinada infra-estrutura. A secção de gestão do desenvolvimento e lançamento de um serviço lida com políticas de acesso, quem ou quando poderá ou não alterar o serviço, com a certificação de que todos os requisitos previamente definidos estão a ser respeitados e que a performance do serviço, sujeita a monitorização, está a ser adequada certificando-se também que os princípios de reutilização e fiabilidade estão a ser correspondidos e que as políticas estabelecidas estão a ser bem seguidas. [35] [34] [38]

O suporte e gestão de SOA numa infra-estrutura lida com os aspectos de segurança dos serviços tais como a gestão de identidades, controlo de acessos, politicas de segurança e policiamento, gestão dos serviços do ponto de vista dos processos como também a previsão

e gestão de alterações a serem efectuadas, para além disso também lida com a virtualização dos serviços de modo a averiguar e certificar que a localizações e capacidades destes sejam adequados, com suporte à escalabilidade dos recursos, com a prioritização das aplicações compostas (múltiplos serviços) e com optimização da performance pela correcta distribuição dos recursos disponíveis. [35] [34] [38]

Pode-se então concluir que SOA é principalmente indicado para sistemas distribuídos heterogéneos com múltiplos domínios administrativos diferentes onde a gestão de serviços é pouco escalável e caótica sendo nesses ambientes que os processos SOA melhor se aplicam e mais vantagens oferecem pelo que SOA não é uma solução universal pois poderá não ser rentável a sua aplicação para sistemas mais pequenos onde é apenas necessária uma separação binária ou ternária dos seus componentes com poucas necessidades de expansão e optimização. [35] [34] [38]

3.3. 'Service Delivery Platform' (SDP)

3.3.1. Definição

Não parece haver um consenso para a definição especifica de 'Service Delivery Platform' (SDP) mas os pontos comuns a todas as definições é que se trata de um conjunto de princípios arquitecturais que foram criados para facilitar a transição de redes organizadas em silos para redes, organizadas em serviços de rede horizontais, baseadas em SOA. Inicialmente tinha como objectivo a abstracção de recursos de rede de modo a que a sua partilha possa ser facilitada sendo mais tarde anexados aspectos de governança, gestão, tecnologias de informação e serviços 'web'. O SDP segue os conceitos de SOA sendo um conjunto de soluções e serviços que formam uma estrutura que fornece e gere serviços de valor acrescentado e conteúdos criados tanto pela própria empresa como por terceiros. Concretamente um SDP é uma porção da infra-estrutura de rede de um operador que interage quer com os sistemas OSS/BSS, AAA e nuclear da rede suportando o aprovisionamento de serviços de valor acrescentado em interfaces 'web' e fornecendo um

acesso aos serviços da rede de um modo abstracto, um exemplo será o 'OMA Service Environment' (OSE). [42] [48] [11]

3.3.2. Benefícios

A utilização de um SDP fornece múltiplos benefícios comparando com a abordagem mais comum, baseada em silos. Um SDP permite que serviços e conteúdos possam ser criados por terceiros sem conhecimento das características internas da rede e que sejam lançados de forma padronizada minimizando adaptações e custos associados a estas tirando a necessidade de detalhar processos de autorização e aprovisionamento. Permite padronizar a conexão de serviços de terceiros transparentemente permitindo a orquestração de vários serviços e fontes de informação de modo a que a implementação de serviços mais complexos seja facilitada e flexível, logo mais rápida e de menor impacto monetário. Cria vários níveis de desenvolvimento para a criação de serviços: um nível interno onde um serviço é desenvolvido com o conhecimento profundo da estrutura interna da rede do operador, um nível externo onde são utilizadas estruturas de desenvolvimento tais como linguagens de programação, tirando a necessidade de ter conhecimento técnico da rede interna, e um nível que se situa entre ambos, vulgo intermédio, onde são utilizadas características de ambos. Isto permite uma melhor adaptação das informações a disponibilizar às necessidades da equipa de desenvolvimento impedindo a divulgação de detalhes privados ou extremamente complexos expondo excessivamente a estrutura ou impedindo as limitações associadas à ocultação excessiva desses detalhes. Fornece assim uma abstracção de recursos através de serviços 'web' e interfaces standard, tais como Parlay X e SIP, permitindo com estes possam ser facilmente partilhados e reutilizados com segurança, deste modo é facilitado o desenvolvimento de novos serviços sem a necessidade do profundo conhecimento da rede reduzindo o esforço e tempo necessário. [42] [48]

3.3.3. Arquitectura

A arquitectura SDP, de um modo genérico, está assente em funcionalidades comuns separadas em entidades ou plataformas lógicas para a execução de serviços ou entrega de conteúdos e em camadas de abstracção para as funcionalidades de rede ou exposição de serviços, também é de referir a necessidade de integrar os seus mecanismos com o OSS/BSS, criação e gestão de serviços também como funções de AAA também como fornecer suporte à gestão do ciclo-de-vida dos serviços e a transparência da rede de acesso.

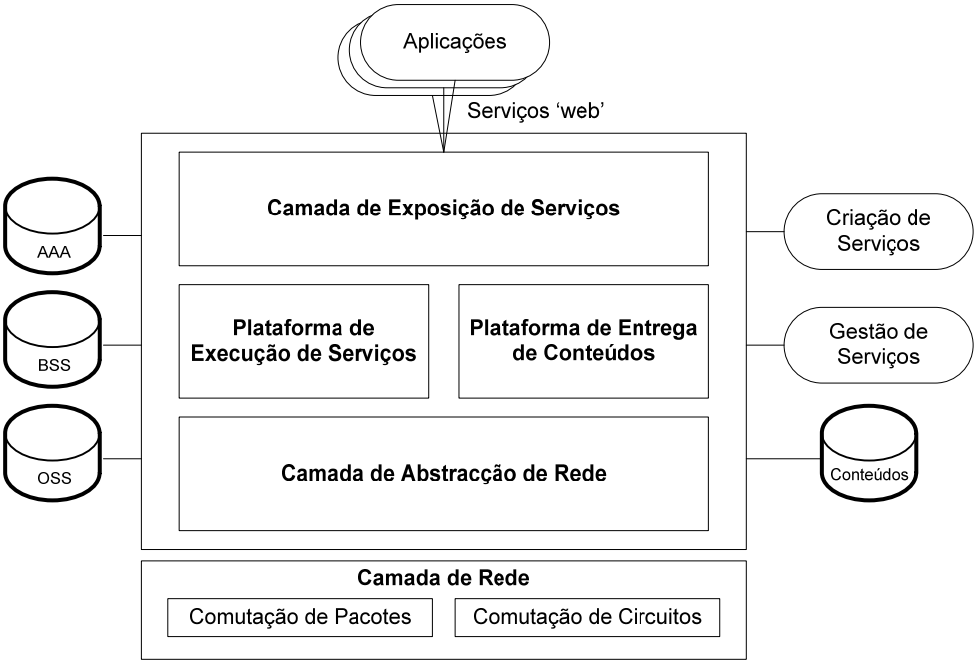

Figura 3.5 – Arquitectura SDP generalizada. [42] [11]

Da figura anterior, figura 3.5, observam-se 4 elementos essenciais para uma plataforma SDP: a camada de abstracção de rede, a plataforma de execução de serviços, a plataforma de entrega de conteúdos e a camada de exposição de serviços. A camada de abstracção de rede deverá fornecer um conjunto de interfaces estandardizadas de modo a permitir o acesso às capacidades e funcionalidades da rede tais como voz, chamadas multimédia e mensagens. A plataforma de execução de serviços deverá fornecer suporte para o ciclo-de-

vida dos serviços tal como gestão de integridade e eventos, permitir o acesso às interfaces da camada de abstracção de rede fornecendo assim um ambiente para o lançamento e execução dos serviços. A plataforma de entrega de conteúdos será importante para redes de convergência onde o fornecimento de serviços para o ambiente móvel traz a necessidade de uma plataforma para a disponibilização de conteúdos multimédia. E a camada de exposição de serviços que sendo um elemento opcional na arquitectura SDP é extremamente importante para a convergência do mundo de telecomunicações e a Internet permitindo um acesso às funcionalidades de rede e entrega de conteúdos e serviços a um nível de abstracção ainda mais elevado do que o da camada de abstracção de rede. Deste modo aplicações externas poderão aceder a serviços, que podem ou não ter um elevado nível de granularidade, implementados na camada de execução que por sua vez se baseiam em funções e capacidades fornecidas pela camada de abstracção de rede. Pode se concluir então que embora não haja consenso para uma definição exacta em toda a industria do que constitui um SDP, esta arquitectura generalizada fornece os componentes chave comuns a todas para a concretização do seu objectivo principal que é o desacoplamento e abstracção de recursos de rede para a redução e partilha com terceiros do risco de desenvolvimento e lançamento de novos serviços aos seus utilizadores, sejam particulares ou empresas. [11] [42] [51]

3.4. ‘OMA Service Environment’ (OSE)

3.4.1. Introdução

O ‘Open Mobile Alliance Service Environment’ (OSE) é uma arquitectura lógica, flexível e extensível criada para fornecer uma estrutura e um conjunto de regras comuns para a especificação, desenvolvimento e lançamento de ‘enablers’, que fornecem um, ou vários, serviços através da disponibilização de uma, ou mais, interfaces públicas. OSE é, no fundo, um conjunto de ‘enablers’ especificados por OMA (Open Mobile Alliance), ‘OMA Enablers’, que fornecem componentes estandardizados que interagem entre si formando um ambiente no qual serviços podem ser desenvolvidos e lançados. A sua utilização permite o rápido desenvolvimento e lançamento de novas aplicações, a abertura desta capacidade a terceiros protegendo os activos do fornecedor de serviços, a reutilização de ‘enablers’ e a redução da organização de estruturas em silos. A criação desta arquitectura foi motivada pelo aumento da necessidade de fornecer e desenvolver serviços, por sua vez isto criou uma necessidade de flexibilizar e agilizar esse processo pois, para uma estrutura baseada em silos, o aumento de serviços criará uma duplicação de dados e recursos, para além disso também torna o desenvolvimento e lançamento de novos serviços mais complicado, de maior custo e com facilidade cria uma inconsistência nas interfaces dos vários serviços dificultando a integração e cooperação entre eles. Outras motivações para a sua criação foram a necessidade de reutilizar os ‘enablers’ OMA que já tinham sido especificados fornecendo vantagens ao reduzir as dificuldades associadas ao lançamento de uma nova aplicação permitindo assim que esta possa ser lançada em diferentes ambientes com o mínimo de adaptações e que funções comuns possam ser estandardizadas num só ‘enabler’ OMA e assim serem reutilizadas em vez de ser necessário recria-las para cada novo serviço necessite de as utilizar. Finalmente, também é necessário o melhoramento da percepção de continuidade para o utilizador final sendo que se os vários serviços a que o utilizador acede são consistentes na funcionalidade e integrados do mesmo modo com a operadora e entre operadoras, assim a sua utilização será facilitada e reduzirá a descontinuidade criada principalmente pela comutação de diferentes operadores e serviços num ambiente de mobilidade com as consequentes necessidades de transição de informações de utilizador e suas configurações pessoais. [50]

3.4.2. Arquitectura

A arquitectura OSE tem quatro princípios chave: a delegação e reutilização de ‘enablers’, protecção de ‘enablers’ e recursos, funcionalidade intrínseca e expansibilidade. O principio da delegação de reutilização de ‘enablers’ diz que as especificações de um ‘enabler’ deverão ser reutilizadas sempre que possível reduzindo o problema do silo vertical simplificando a integração de novas aplicações ou ‘enablers’ num domínio OSE, assim uma implementação de um ‘enabler’ pode, deste modo, invocar qualquer função estandardizada ou ‘enabler’ presente no seu domínio ou noutro domínio OSE externo para satisfazer as suas necessidades intrínsecas. A protecção de ‘enablers’ e recursos apenas levanta a importância de que, em cada domínio, seja controlado o acesso aos recursos pela exposição controlada destes. O principio da funcionalidade intrínseca é importante para catalogar funções que estão subjacentes a operações e por isso terão uma maior probabilidade de ser reutilizadas e refere que uma função é intrínseca quando é essencial para fornecer uma determinada tarefa a um ‘enabler’ tal como uma função de autenticação será intrínseca ao ’login’ mas a mesma função pode ser não-intrínseca para outro ‘enabler’ sendo esta definição subjectiva e, por isso, definida em cada ‘enabler’. E a expansibilidade relaciona-se com a independência das aplicações e ‘enablers’ que utilizam interfaces expostos por outros ‘enablers’ que, por sua vez, se interligam com os recursos reais, com este principio na utilização de múltiplos fornecedores de software que suportam diferentes tecnologias as suas aplicações ligam-se dinamicamente às interfaces registadas num ‘enabler’ de descoberta de serviços sem serem afectadas por diferentes implementações. O seu modelo arquitectural foca-se principalmente na identificação dos diferentes elementos e suas relações dando liberdade ao modo como são criados e lançados os ‘enablers’ não obrigando à existência de um elemento de aplicação de políticas nem sendo definido o modo como o serviço será lançado. [50]

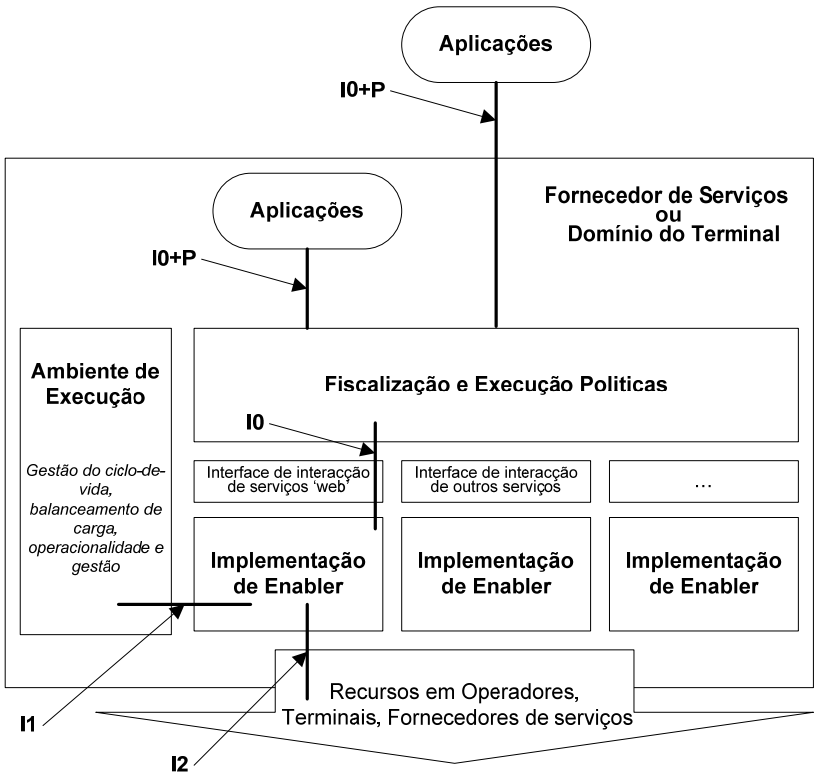

Figura 3.6 – Modelo arquitectural OSE 1.0 e interfaces. [50]

Do modelo da arquitectura, figura 3.6, tem-se:

O 'enabler' que é o produto principal do OMA que deverá especificar uma ou mais interfaces públicas. [50]

A implementação do 'enabler' onde não há restrições no modo como as especificações deste são implementadas embora definindo que as funções disponibilizadas deverão ser estandardizadas e que terão de expor interfaces de gestão do seu ciclo-de-vida permitindo com que no domínio seja utilizada a infra-estrutura para gestão dos componentes do 'enabler' que, por sua vez, poderá ser invocado por aplicações ou outras implementações. [50]

As interfaces que são definidas para invocar funções intrínsecas de cada 'enabler' permitindo a interoperabilidade entre entidades de um 'enabler' e a gestão do seu ciclo-de-vida. [50]

As interfaces de interacção de ‘enablers’ onde se definem associações específicas à linguagem de programação ou protocolo de rede para cada interface apesar dessas interfaces serem agnósticas à linguagem utilizada. [50]

Os recursos que, sendo elementos arquitecturais, representam capacidades no domínio do fornecedor de serviços que podem ser acedidas ou invocadas directamente por uma implementação de um ‘enabler’. [50]

As aplicações cuja localização não é específica, podendo se localizar em qualquer lado no ambiente de serviço incluindo no terminal móvel, representam uma implementação de um conjunto de funções que realizam tarefas para um ou mais serviços. [50]

O ambiente de execução que fornece suporte para funções de gestão do ciclo-de-vida contendo várias funções de suporte e monitorização de processos que permitem ao domínio OSE controlar os ‘enablers’que também podem invocar essas funções caso seja necessário.

E o fiscalizador e executor de politicas (‘Policy Enforcer’) que sendo um elemento arquitectural de muita importância gere e aplica políticas de acesso criadas para proteger os recursos de acessos não autorizados, através da facturação adequada, do registo de eventos e da aplicação das preferências do utilizador e sua privacidade. Os métodos utilizados nesta entidade para a aplicação de politicas e sua implementação não são definidos no OSE. [50]

Para além disso uma plataforma OSE gere também os procedimentos que poderão ser aplicados entre ‘enablers’ e as aplicações que podem residir ou não no mesmo domínio ou ambiente. [50]

Também na figura se poderão observar as interfaces OSE que são definidas em categorias abstractas:

A categoria I0 é a categoria de interface para as funções intrínsecas, é exposta a aplicações e ‘enablers’ quando as politicas a ser aplicadas não necessitam de parâmetros adicionais ou quando não há nenhuma politica a aplicar. Inclui eventos assíncronos e métodos para subscrever elementos de escuta para esses eventos. [50]

A categoria I0+P combina as características da interface de categoria I0 com a aplicação de políticas (P), satisfazendo as políticas definidas para aquele ambiente e que têm que ser

aplicadas quando é exposta a interface I0 do ‘enabler’ tipicamente para entidades externas. Esta categoria também é exposta para as aplicações e ‘enablers’ do mesmo domínio quando as políticas a ser aplicadas necessitam de parâmetros adicionais. [50]

A categoria I1 compreende as interfaces entre os ‘enablers’ e o ambiente de execução. [50]

A categoria I2 compreende as interfaces usadas pelos ‘enablers’ descrevendo como poderão invocar recursos. Essas interfaces normalmente correspondem a interfaces e recursos de redes de suporte assim como para outros recursos secundários. [50]

Do ponto de vista de segurança é apenas definido que um domínio OSE pode ser protegido utilizando mecanismos que reduzam os riscos de segurança sem especificar nenhum método em concreto principalmente porque as ameaças de relevância e o modo de as contrariar depende do risco da aplicação e das políticas de segurança consideradas. [50]

3.5. API’s para exposição de serviços

3.5.1. 3GPP Parlay X

O Parlay X começou em 2001 da necessidade de definir interfaces simples permitindo com que o desenvolvimento de aplicações possa ser efectuado sem o conhecimento básico de protocolos e redes de telecomunicações. Os seu objectivo principal é de permitir com que a comunidade TI e sua vasta comunidade de programadores possa desenvolver aplicações sem necessitar do profundo conhecimento dos protocolos de suporte utilizados e para isso aplicou os conceitos de Parlay em tecnologias de desenvolvimento de aplicações no âmbito da Internet, mais familiar ao grupo alvo. A ideia base parte da definição de blocos de funcionalidades abstractos expondo os serviços das interfaces Parlay já existentes definindo serviços ‘web’ e suas funcionalidades sem especificar detalhes de

implementação, tais como o ambiente e como e onde será lançado, que foram evoluindo descrevendo novas funcionalidades para além das fornecidas pela API do Parlay. Em relação ao Parlay, as suas API's, nas quais se baseia o Parlay X, evoluíram em paralelo com as APIs OSA do 3GPP que surgiu mais tarde até que ambos se uniram numa especificação Parlay/OSA a partir da especificação Parlay 3.0/OSA release 5, o mesmo se passa com o Parlay X onde existem especificações equivalentes do consórcio 3GPP. Na figura seguinte está representada localização relativa de uma aplicação que utiliza a API do Parlay X, seu relacionamento com o Parlay, onde o Parlay X fornece um maior nível de abstracção, e o modo como poderá aceder às capacidades de rede. Como se pode observar na seguinte figura (figura 3.7) poderá tanto utilizar as capacidades fornecidas pelo Parlay como utilizar as capacidades de rede de um modo directo. [56] [57]

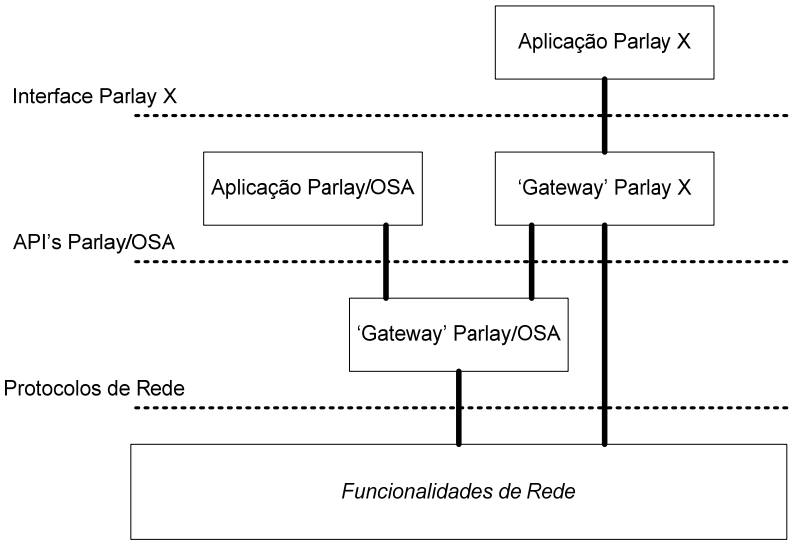

Figura 3.7 – Relação entre Parlay X e Parlay/OSA. [56]

Os serviços 'web' Parlay X 3.0 (draft) dividem-se em 20 blocos onde as especificações equivalentes do 3GPP disponibilizam 2 blocos adicionais de funcionalidades. Como se pode ver na tabela seguinte, tabela 3.1, a API Parlay X é extremamente completa

estandardizando o maior número de funcionalidades possível para desse modo a permitir às aplicações tomar melhor proveito das capacidades e serviços das redes, poderá observar-se também que o Parlay X não se trata de uma simples exposição de funcionalidades de uma dada rede de telecomunicações, fornecendo outros serviços e capacidades que outrora estavam apenas disponíveis em aplicações específicas do terminal do utilizador tais como localização e gestão de contactos.

Bloco	**Nome**	**Especificação de Parlay X Web Services Versão 3.0 (draft)**	**Especificações 3GPP Equivalentes**
1	Common	3rd Draft ES 202 504-1 Parlay X 3.0	TS 29.199-01
2	Third Party Call	5th Draft ES 202 504-2 Parlay X 3.0	TS 29.199-02
3	Call Notification	3rd Draft ES 202 504-3 Parlay X 3.0	TS 29.199-03
4	Short Messaging	3rd Draft ES 202 504-3 Parlay X 3.0	TS 29.199-04
5	Multimedia Messaging	3rd Draft ES 202 504-4 Parlay X 3.0	TS 29.199-05
6	Payment	3rd Draft ES 202 504-5 Parlay X 3.0	TS 29.199-06
7	Account Management	4th Draft ES 202 504-6 Parlay X 3.0	TS 29.199-07
8	Terminal Status	4th Draft ES 202 504-7 Parlay X 3.0	TS 29.199-08
9	Terminal Location	2nd Draft ES 202 504-8 Parlay X 3.0	TS 29.199-09
10	Call Handling	4th Draft ES 202 504-9 Parlay X 3.0	TS 29.199-10
11	Audio Call	2nd Draft ES 202 504-10 Parlay X 3.0	TS 29.199-11
12	Multimedia Conference	3rd Draft ES 202 504-11 Parlay X 3.0	TS 29.199-12
13	Address List Management	3rd Draft ES 202 504-12 Parlay X 3.0	TS 29.199-13
14	Presence	2nd Draft ES 202 504-13 Parlay X 3.0	TS 29.199-14
15	Message Broadcast	6th Draft ES 202 504-14 Parlay X 3.0	TS 29.199-15
16	Geocoding	5th Draft ES 202 504-15 Parlay X 3.0	TS 29.199-16
17	Application-driven Quality of Service	4th Draft ES 202 504-16 Parlay X 3.0	TS 29.199-17
18	Device Capabilities and Configuration	3rd Draft ES 202 504-17 Parlay X 3.0	TS 29.199-18
19	Multimedia Streaming Control	1st Draft ES 202 504-18 Parlay X 3.0	TS 29.199-19
20	Multimedia Multicast Session Management	3rd Draft ES 202 504-19 Parlay X 3.0	TS 29.199-20
21	Content management	--	TS 29.199-21
22	Policy	--	TS 29.199-22

Tabela 3.1 – Mapa de funcionalidades Parlay X.

3.5.2. Web21C SDK da ‘British Telecom’ (BT)

O Web21C é um conjunto de bibliotecas cujo objectivo é permitir o desenvolvimento simplificado de aplicações para o uso das interfaces expostas pela ‘British Telecom’ (BT).

As capacidades expostas foram divididas em 6 pacotes de API’s. O serviço de autenticação que fornece serviços de autenticação para utilizadores e grupos de utilizadores, o ‘CallFlow’ que permite o desenvolvimento de aplicações de voz, o ‘Conference Call’ que permite a criação e gestão de conferências, o ‘Inbound SMS’ que permite a recepção de ‘SMSs’, o ‘Messaging’ que permite o envio de ‘SMS’, o ‘VoiceCall’ que permite efectuar e gerir chamadas e o ‘email’. Cada aplicação necessita de ser registada para obter um certificado que lhe permite aceder aos serviços e o seu desenvolvimento é feito através de um SDK disponível em Java, .NET, PHP e Python. Os serviços são disponibilizados num servidor usando REST e SOAP, as mensagens são assinadas com um certificado X.509 referente à aplicação e anexadas num HTTP POST. Os serviços podem ser acedidos sem a utilização do SDK utilizando ‘SOAP Web Serviços’ em HTTP/HTTPS incluindo WS-Security, WS-Trust, WSE, X.509 PKI e SAML.

São disponibilizados dois servidores, um para testes (ambiente ‘Sandbox’) no endereço ‘acorn.ws.bt.com’ e outro de produção no endereço ‘oaktree.ws.bt.com’, ambos necessitam do registo independente das aplicações ou seja é necessário um certificado diferente para cada aplicação em cada servidor. Neles estão expostos 5 serviços: ‘MessagingOneWay’, ‘SessionThirdPartyCall’, ‘SessionConferencing’, ‘WhiteLabelAuthentication’ e ‘MessagingInbound’. Cada serviço está disponível no endereço:

‘https://{endereço_do_servidor}/endpoint/{nome_do_serviço}/{versão}’

O formato das mensagens SOAP a utilizar será:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

  xmlns:sdk="http://sdk.bt.com/{versão}/{nome_do_serviço}"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsse=
  "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu=
  "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

  <soap:Header>
        <wsa:MessageID>urn:uuid:{endereço_uuid} </wsa:MessageID>
        <wsa:Action>http://sdk.bt.com//{versão}/{nome_do_serviço}#{acção _a_executar}</wsa:Action>
        <wsse:Security>
        <ds:Signature …
        <wsu:Timestamp …
        <wwse:BinarySecurityToken …
        </wsse:Security>
  </soap:Header>

  <soap:Body wsu:Id="BodyID">
        <sdk:{acção _a_executar}>
        {Conteúdos definidos no WSDL}
        </sdk:{acção _a_executar}>
  </soap:Body>

</soap:Envelope>
```

Para o WS-Adressing o campo MessageID identifica unicamente a mensagem, recomenda-se a utilização de um UUID definido no RFC4122 não reutilizado em mensagens subsequentes, e o campo Action é utilizado para reencaminhar a mensagem para a entidade responsável pela operação.

Para o WS-Security utiliza-se o elemento 'Signature' que contém que partes da mensagem foram assinadas, o valor calculado da assinatura e a chave utilizada para a gerar. O atributo 'wsi:Id=' do corpo da mensagem é referenciado pela assinatura digital presente no cabeçalho da WS-Security.

Exemplo Autenticação

De modo a utilizar uma aplicação é necessário obter um certificado para ela no endereço:

http://sdk.bt.com/application_registration

Ou pode ser gerado manualmente com o 'Certificate Tool' de modo a que se possam identificar todas as aplicações que utilizam os serviços fornecidos.

Exemplo 'ThirdPartyCall'

```
...
tpcManager = new SessionThirdPartyCallManager();
ThirdPartyCall tPartyCall = tpcManager.makeCall("tel:+4479XXXXXXXX", "sip:user@domain.com");
...
```

Com o SDK em Java uma chamada resume-se a 2 linhas de código. Podem-se obter também informações sobre a chamada e o seu estado pela invocação de outros métodos disponíveis.

Esta API baseia-se principalmente nas seguintes tecnologias:

SAML ('Security Assertion Markup Language'), certificados X.509, 'SOAP Web Services', HTTP/HTTPS, WS-Security, WS-Trust e WSE ('Web Services Enhancements').

Recentemente a BT adquiriu a 'Ribbit Corporation' que fornece uma API baseada em Adobe Flash e Flex estando planeado suporte a outras linguagens. As APIs Ribbit inicialmente suportam chamadas de voz, reprodução de voz e texto, transcrição de voz, listagens de directorias e acesso a dados do utilizador. Dado isto a BT vai suspender a API Web21C transitando para a API do Ribbit que será baseada em REST. [59]

3.5.3. ‘Orange API’ da Orange

A empresa Orange focou-se na rápida implementação dos conceitos Web2.0, SaaS (‘Software as a Service’) e redes sociais.

Tem no total 14 API’s, 6 pertencem a um grupo apelidado de ‘Personal API’s’ onde, após a autenticação, podem ser combinadas na aplicação personalizando o serviço. A ‘Autentication API’ fornece um serviço de autenticação e privacidade e é necessária para a utilização das restantes ‘Personal API’s’, a ‘Personal Calendar API’ permite a adição de eventos nos calendários de clientes, ‘Personal Contacts API’ permite a visualização e adição de contactos na lista de contactos dos clientes, ‘Personal Messages API’ permite acesso à caixa de mensagens do cliente, a ‘Personal Photo API’ permite adição e visualização de fotos e álbuns dos clientes e o ‘Personal Profile API’ permite o acesso à informação do perfil do cliente. Para além destas tem também as APIs ‘contact everyone’, ‘multimedia conference’ e ‘ device capability enabler’ para fornecer serviços que permitam a interação de múltiplos utilizadores. O ‘contact everyone’ permite o envio de SMS, mensagens de voz, correio electrónico e fax, o ‘multimedia conference’ permite a marcação e iniciação de conferências, convite de elementos e partilha de documentos e aplicações durante a conferência e o ‘device capability enabler’ fornece informação acerca das capacidades de cada dispositivo móvel baseado na marca e modelo, código TAC ou ‘User-Agent’. As restantes 5 são na realidade os serviços de telecomunicações que foram expostos com algumas adições, a ‘SMS API’ fornece serviço de envio e recepção de mensagens texto, a ‘email API’ que permite o envio, recepção e gestão de emails, a ‘location API’ permite a obtenção da localização de um utilizador, o ‘click-to-call API’ permite estabelecer chamadas telefónicas entre dois números através de uma aplicação ‘web’ e a ‘voicemail API’ permite o acesso a mensagens do correio de voz através de uma aplicação ‘web’. Também fornecem 2 plataformas, a ‘bubbletop developer platform’ e a ‘pikeo developer platform’, para o desenvolvimento de aplicações que utilizem as API’s sendo a última também uma API de partilha de imagens. A exposição é feita utilizando REST com os métodos HTTP/HTTPS GET e POST com XML-RPC.

Detalhes dos serviços expostos

‘click-to-call API’

http://sandbox.alpha.orange-api.net/call/createCall.xml?{parâmetros}

call.createCall

‘email API’

http://mail.alpha.orange-api.net/mail/{método}.xml?{parâmetros}

mail.sendMail

mail.getMailList

mail.getMail

mail.deleteMail

mail.markMailRead

mail.markMailUnread

mail.countMail

‘location API’

http://location.alpha.orange-api.net/location/{método}.xml?{parâmetros}

location.createAuthorization

location.getLocation

location.buyLocationPass

‘SMS API’

http://sms.alpha.orange-api.net/sms/sendSMS.xml?{parâmetros}

sms.sendSMS

‘voice mail API’

http://voicemail.alpha.orange-api.net/voicemail/{método}.xml?{parâmetros}

getMessageList

getMessage

markMessageRead

markMessageUnread

deleteMessage

updateAnnouncement

deleteAnnouncement

3.5.4. Sipgate API da Sipgate

A Sipgate fornece uma API única para a utilização dos serviços de mensagens de voz, listagem de chamadas, controlo de custos, ‘forking’ de chamadas, lista telefónica com ‘click-2-dial’ e envio de SMS. Baseia-se no protocolo SIP para a sinalização de chamadas com suporte para um número limitado de ‘codecs’ e para a utilização do serviço utiliza XML-RPC onde os métodos a invocar são representados em XML e enviados no corpo de uma mensagem HTTP em modo ‘text/xml’, a protecção do tráfego é garantida com a

utilização de SSL/TLS em HTTPS. Para o desenvolvimento de aplicações são fornecidos exemplos da implementação da API em linguagem Perl, na interface gráfica KDE do Linux e também um SDK em .NET, para além disso são disponibilizadas extensões de integração da API com o PowerShell e o Firefox.

Listagem dos métodos:

Obrigatórios:

system.listMethods

system.methodHelp

system.methodSignature

system.serverInfo

Opcionais:

samurai.AccountStatementGet

samurai.BalanceGet

samurai.ClientIdentity

samurai.HistoryGetByDate

samurai.ItemizedEntriesGet

samurai.OwnUriListGet

samurai.PhonebookEntryGet

samurai.PhonebookListGet

samurai.RecommendedIntervalGet

samurai.ServerdataGet

samurai.SessionClose

samurai.SessionInitiate

samurai.SessionInitiateMulti

samurai.SessionStatusGet

samurai.TosListGet

samurai.UmSummaryGet

samurai.UserdataGreetingGet

samurai.UserdataSipGet

3.5.5. T-Online Developer API da Deutsche Telekom

A Deutsche Telekom tem em desenvolvimento 2 serviços, o envio de SMS permitindo o envio de mensagens texto para outro dispositivo móvel ou subscritor fixo e chamadas de voz, 'Voicecall', que permite a iniciação e terminação de uma chamada com 2 participantes e a obtenção de informações acerca da mesma. Cada serviço é disponibilizado numa API em separado, cujos métodos são invocados num serviço 'web' através de mensagens SOAP em HTTP/HTTPS num servidor apelidado de 'sandbox' para testes e desenvolvimento contendo números virtuais simulando situações típicas. Para o desenvolvimento de aplicações é fornecido um SDK em Java e em C#.

Antes da utilização de um serviço é necessária a obtenção de uma chave de segurança válida por 8 horas no serviço, também em SOAP, disponibilizado no endereço:

https://samsts.t-online.com/SAMClientTokenProvider

Como parâmetros de entrada é necessário um nome de utilizador e palava passe e todos os serviços necessitam da chave de segurança e da identificação do ambiente em que o

serviço está a ser utilizado, ou seja, produção, teste ou simulação.

O serviço ‘web’ de ‘Voicecall’ está disponibilizado no endereço:

https://odg.t-online.de/odgvoicebutler/services/VoiceButlerService

E fornece três funcionalidades, iniciação de uma chamada, pedido de informações acerca da chamada e terminação da chamada. Para a iniciação da chamada necessita da chave de segurança, o ambiente, o número do participante A, o número do participante B, tempo de expiração do pedido da chamada e o tempo máximo de duração da chamada. Para o pedido de informações acerca da chamada ou sua terminação é necessário o identificador da sessão da chamada devolvido na resposta da sua iniciação. A mensagem de resposta conterá a mesma chave de segurança, a identificação do ambiente e da sessão, caso tenha sido iniciada, e, dependendo da situação, o status da chamada ou o identificador do erro ocorrido.

Exemplo de um pedido de chamada:

```
<soapenv: Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns: soapenv =
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.voicebutler.odg.tonline.de"> xmlns: web = "http://webservice.voicebutler.odg.tonline.de">
<soapenv:Body>
<web:newCall>
<request>
<samsToken>PEVuY3J5cHRlZEFzc2V[...]Y3J5cHRlZEFzc2VydGlvbj4=</samsToken> <samsToken>
PEVuY3J5cHRlZEFzc2V [...] Y3J5cHRlZEFzc2VydGlvbj4 = </ samsToken>
<environment>2</environment> <environment> 2 </ environment>
<aNumber>+49-6151-11223344</aNumber> <aNumber> +49-6151-11223344 </ aNumber>
<bNumber>+49-6151-11223355</bNumber> <bNumber> +49-6151-11223355 </ bNumber>
<expiration>100</expiration> <expiration> 100 </ expiration>
<maxDuration>100</maxDuration> <maxDuration> 100 </ maxDuration>
</request> </ request>
</web:newCall> </ web: newCall>
</soapenv:Body> </ soapenv: Body>
</soapenv:Envelope> </ soapenv: Envelope>
```

Exemplo de uma resposta:

```
<soapenv: Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns: soapenv =
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> xmlns: xsi = "http://www.w3.org/2001/XMLSchema-
instance">
<soapenv:Body>
<newCallResponse xmlns="http://webservice.voicebutler.odg.tonline.de"> <newCallResponse
xmlns="http://webservice.voicebutler.odg.tonline.de">
<return xmlns=""> <return xmlns="">
<err_msg>The call was initiated successfully.</err_msg> <err_msg> The call was initiated successfully. </ err_msg>
<sessionId>2A91A23F-47B45C64000BC231-B791CBB0</sessionId> <sessionId> 2A91A23F-47B45C64000BC231-
B791CBB0 </ sessionId>
<status>0</status> <status> 0 </ status>
</return> </ return>
</newCallResponse> </ newCallResponse>
</soapenv:Body> </ soapenv: Body>
</soapenv:Envelope> </ soapenv: Envelope>
```

Para o serviço ‘web’ ‘SMS’ ainda não está disponibilizado um endereço. Um pedido de envio de um SMS terá como parâmetros de entrada a chave de segurança e o ambiente tal como no serviço ‘Voicecall’, o número do destinatário e a mensagem a enviar. A resposta conterá um código e uma mensagem do estado do pedido.

Exemplo de um pedido:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web= *Service URL*>
<soapenv:Body>
<web:sendSMS>
<request>
<samsToken>PEVuY3J5cHRlZEFzc2V[...]Y3J5cHRlZEFzc2VydGlvbj4=</samsToken> <samsToken>
PEVuY3J5cHRlZEFzc2V [...] Y3J5cHRlZEFzc2VydGlvbj4 = </ samsToken>
<environment>2</environment> <environment> 2 </ environment>
<number>+49-6151-11223344</number> <number> +49-6151-11223344 </ number>
<message>Das ist eine Beispiel-SMS</message> <message> This is an example SMS </ message>
</request> </ request>
</web:sendSMS> </ web: sendSMS>
</soapenv:Body> </ soapenv: Body>
</soapenv:Envelope> </ soapenv: Envelope>
```

Exemplo de uma resposta:

```
<soapenv: Envelope xmlns: soapenv = "http://schemas.xmlsoap.org/soap/envelope/" xmlns: xsd =
"http://www.w3.org/2001/XMLSchema" xmlns: xsi = "http: / / www.w3.org/2001/XMLSchema-instance ">
<soapenv:Body>
<newCallResponse xmlns= *Service URL* > <newCallResponse xmlns= *Service URL*>
<return xmlns=""> <return xmlns="">
<err_msg>The SMS was sent successfully.</err_msg> <err_msg> The SMS was sent successfully. </ err_msg>
<status>0</status> <status> 0 </ status>
</return> </ return>
</newCallResponse> </ newCallResponse>
</soapenv:Body> </ soapenv: Body>
</soapenv:Envelope> </ soapenv: Envelope>
```

3.5.6. Open Movilforum (Telefonica)

Open Movilforum é uma iniciativa da empresa de telecomunicações Telefonica. O fórum é apresentado como uma plataforma para criação de API's abertas e projectos "mistura" ('mashup') onde estas são integradas numa aplicação ou num serviço 'web'. Cada utilizador pode propor um projecto de uma aplicação ou de uma API que poderá ser desenvolvido em comunidade usando as API's já existentes no fórum e qualquer outra externa, como base o fórum fornece várias 'API's' resultantes da exposição de serviços da operadora, a 'API Recepción de SMS' e 'API de envio de SMS' que permitem, respectivamente, a recepção e envio de mensagens texto (SMS) na caixa de correio electrónico, a 'API Copiagenda' que permite guardar, de forma automática, os contactos da agenda do cartão SIM ou dispositivo móvel, a 'API Localizame' para a localização de uma pessoa, a 'API SMS2.0' que permite saber quais os contactos é que estão conectados ao serviço e iniciar conversações instantâneas tal como nos programas de 'Instant messaging' (IM), o 'Envío http de MMS' que permite compor e enviar mensagens multimédia contendo texto, vídeo, som e imagem, 'Recepção de videochamadas' para a recepção de videochamadas e armazenamento dos fluxos de vídeo/áudio e o 'WAP Push' para o envio automático mensagens WAP a um utilizador. As documentações fornecidas baseiam-se na descrição das APIs abstraindo os utilizadores das tecnologias utilizadas na exposição dos serviços, como se pode ver no exemplo da utilização da API de envio de SMS não é utilizado SOAP nem XML-RPC, apenas um conjunto de linhas de texto em UTF-8 com uma etiqueta para cada pedaço de informação, o que é conceptualmente semelhante mas sem as vantagens que o SOAP ou o XML-RPC fornecem. Em todas as API's são

fornecidos SDKs em várias linguagens, PHP, Ruby, Python, Java, C++ e para várias plataformas, Windows/Mobile, Machintosh, Symbian e IPhone. Para além disso é fornecido o SDK 'Nori' de gestão de dispositivos móveis e estabelecimento de ligações sobre interfaces WWAN ('Wireless Wide Area Network').

Exemplo de utilização da API de envio de SMS

O método da API de envio de SMS em Java é:

"SendMessage(nome_de_utilizador, palavra_passe, destinatário, mensagem)"

O método, como pode ser visto no código seguinte, é executado anexando a uma mensagem HTTPS os dados de autenticação (nome de utilizador e passe), o destinatário e a mensagem a enviar. Por sua vez a mensagem HTTPS é enviada para o url:

https://opensms.movistar.es/aplicacionpost/loginEnvio.jsp

E resposta HTTPS do envio é devolvida pelo método à aplicação.

Dados anexados

```
String data = URLEncoder.encode("TM_ACTION", "UTF-8") + "=" + URLEncoder.encode("AUTHENTICATE", "UTF-8");
data += "&" + URLEncoder.encode("TM_LOGIN", "UTF-8") + "=" + URLEncoder.encode(Login, "UTF-8");
data += "&" + URLEncoder.encode("TM_PASSWORD", "UTF-8") + "=" + URLEncoder.encode(Pwd, "UTF-8");
data += "&" + URLEncoder.encode("to", "UTF-8") + "=" + URLEncoder.encode(Dest, "UTF-8");
data += "&" + URLEncoder.encode("message", "UTF-8") + "=" + URLEncoder.encode(Msg, "UTF-8");
```

Propriedades do cabeçalho:

```
connection.addRequestProperty("Content-type", "application/x-www-form-urlencoded");
connection.addRequestProperty("Accept-Encoding", "gzip, deflate");
connection.addRequestProperty("Accept", "image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword,
*/*");
connection.addRequestProperty("Connection", "Keep-Alive");
```

Utilização do método:

```
SMSSender oSmsSenderHttps = new SMSSender();
oSmsSenderHttps.SendMessage(Login, Pwd, Dest, Msg);
```

3.5.7. Comparação das várias API's

As API's analisadas resultam da necessidade das indústrias de telecomunicações de fornecer mais serviços e a partir de múltiplas plataformas. A abertura dos recursos das suas redes abstraindo-as de modo a que aplicações exteriores possam invocar recursos enquanto que escondendo simultaneamente os detalhes internos da rede, não sendo praticável no passado, tornou-se possível e assim o tempo de desenvolvimento e os custos de um serviço podem ser partilhados com os seus utilizadores e parceiros externos que os utilizam e desenvolvem aplicações que tomam partido deles. A escolha da estrutura das API's dos recursos de rede a disponibilizar ao exterior apresenta-se como um desafio pelas restrições de segurança, fiabilidade e facturação necessárias para o seu correcto funcionamento e retorno de investimento, para além disso as semelhanças de funcionalidades entre serviços causa dúvidas sobre o melhor modo de as disponibilizar com o melhor índice de reutilização e simplicidade. Segue-se uma breve comparação das características das diferentes API's ao nível de segurança, de nível de exposição, de funcionalidades adicionais não directamente ligadas a serviços de telecomunicações e SDK's disponibilizados.

Segurança

Em termos de segurança para a protecção dos dados transmitidos todas utilizam HTTPS, para a autenticação existem diferentes opções, a 'Orange API' (OAPI) e o 'T-Online Developer API' (TODAPI) utilizam um 'token', com validade finita, que é obtido antes da utilização do serviço, o 'Open Movilforum' (OMF) utiliza 'JavaServer Pages' (JSP) em HTTS, o 'Sipgate API' (SGAPI) uma autenticação HTTP básica em SSL/TLS e no caso do 'Web21C SDK' (W21C) são emitidos certificados X.509, com validade de 1 mês, para cada utilizador. A excepção é de que o '3GPP Parlay X' (Parlay X) não define nenhum modo de segurança deixando essa escolha para as especificações 'WS-Security' do serviço 'web' implementado.

Exposição

A quantidade de serviços expostos e suas características variam imenso. Enquanto que por um lado o Parlay X permite a utilização de 20 blocos de API's com uma extensa gama de funções extremamente detalhadas por outro o SGAPI fornece uma para um pequeno grupo de funcionalidades básicas e o TODAPI que fornece duas, uma para voz e outra para SMS, sem funcionalidades detalhadas. Das restantes a que fornece melhor nível de exposição é a OAPI com 14 API's em 3 grupos de funcionalidades agregadas de um modo específico estando cada grupo adequado a uma utilização diferente seguida pela OMF com 10 API's de funcionalidades básicas com diferentes objectivos e pelo W21C com 6 API's.

Funcionalidades adicionais

As funcionalidades adicionais compreendem serviços que tipicamente estariam ao cargo de aplicações do dispositivo do utilizador sem necessitar do operador, tais como gestão de lista de contactos ou agenda, e serviços adicionais de presença e localização que se tornaram uma extensão aos serviços típicos de voz e dados. De todas as API's o Parlay X é

o mais completo com vários serviços tais como presença, gestão de lista de contactos, ‘geocoding’, QoS, etc. A OAPI também fornece muitas funcionalidades adicionais, permitindo o armazenamento e partilha de fotos, a disponibilização do perfil do utilizador, gestão de contactos e calendário com eventos. No caso do OMF existe também uma grande variedade de serviços adicionais mas com a intervenção dos utilizadores, para a criação de novas API’s, sem terem que necessariamente utilizar os serviços básicos e poderem utilizar outros sítios de Internet. Das restantes API’s a SGAPI permite apenas a listagem de contactos para o serviço de ‘click2dial’, o W21C a criação de aplicações de serviços baseados em voz na API de ‘CallFlow’ e o TODAPI que para além de não fornecer todas as funcionalidades básicas não fornece nenhuma adicional.

SDK’s

Para o desenvolvimento de aplicações são disponibilizados SDK’s de modo a facilitar esse processo ao programador e desse modo abstrair ainda mais as funcionalidades disponibilizadas. Para o Parlay X cada bloco de API’s é disponibilizado um descritor WSDL e um XSD para a descrição do serviço ‘web’ e definição dos dados respectivamente, para o OMF os SDK’s disponibilizados dependem da API existindo SDK’s em linguagem Java, C++, Python e Ruby. No SGAPI é disponibilizado em Perl, .NET e são fornecidas extensões para ‘PowerShell’, ‘Firefox’ e ‘KDE Kicker’. No W21C é disponibilizado um SDK em Java, .NET, PHP e Python. O TODAPI só está disponível em Java e o OAPI apenas disponibiliza a documentação não fornecendo nenhum SDK específico.

3.6. Aplicação em redes 'IP Multimedia Subsystem' (IMS)

3.6.1. Estado actual e objectivos

A implementação das redes IMS permite um aumento das receitas das operadoras obtidas pelo tráfego de voz e dados reduzindo simultaneamente os custos pela partilha de funções pelos diversos serviços oferecidos. Tudo isto foi resultante das capacidades inerentes da arquitectura e são apenas o ponto de partida pois durante o seu desenvolvimento não foi dada a importância necessária ao modo como seriam implementados os serviços ficando apenas uma arquitectura de suporte à convergência do mundo fixo, tradicionalmente serviços de voz e redes baseadas em comutação de circuitos, com o mundo móvel e serviços multimédia em redes baseadas em comutação de pacotes mas não é o suficiente para atingir todos os objectivos principais que motivam esta transição, a redução do tempo necessário e dos riscos associados no desenvolvimento e lançamento de novos serviços, o aumento da receita média por assinante, 'Average Revenue Per User' (ARPU), o suporte para novos standards e seu constante desenvolvimento, a aplicação do conceito de eterno desenvolvimento onde um serviço ou aplicação está continuamente a ser melhorado e a redução dos custos OPEX e CAPEX. É com estas finalidades que o IMS, SDP, OSE e SOA se vão interligar e complementar, será necessária a utilização de um SDP e dos conceitos de SOA para que se obtenha uma plataforma realmente extensível e flexível maximizando as capacidades de toda a estrutura e assim assegurar as metas propostas. O IMS fornece o controlo centralizado para a lógica das aplicações e a capacidade de fornecer serviços de um modo agnóstico à rede de acesso e o OSE é a concretização de um SDP e a implementação dos conceitos de SOA fornecendo uma abstracção das tecnologias de rede necessárias para camada de serviço onde serviços possam ser geridos de um modo flexível e de baixos custos.

No final podemos dizer que se tornou necessária a transposição das características hoje encontradas na Internet para o mundo dos serviços de telecomunicações para o aumento da dinâmica dos serviços de valor acrescentado oferecidos ao utilizador final e deste modo

aumentar os lucros das empresas de telecomunicações. [52] [53] [54]

O objectivo principal nesta secção é discutir a implementação de uma plataforma OMA OSE e a exposição de serviços através de API's numa rede IMS aproveitando as suas potencialidades e serviços já existentes. O resultado final deverá ficar organizado em 3 camadas lógicas horizontais, a camada de aplicações, de serviços e de recursos como se poderá observar na figura seguinte, figura 3.8. Na camada de aplicações estarão presentes as aplicações que utilizarão e disponibilizarão serviços aos subscritores, é para essa camada que serão disponibilizados serviços para que clientes e empresas externas possam desenvolver uma infinidade de aplicações personalizadas. Na camada de serviços estarão presentes os ambientes onde serão lançados 'enablers' e outros servidores de aplicativos que funcionarão de acordo com o ambiente no qual foram lançados interagindo com outros elementos da mesma camada e utilizando recursos de rede disponibilizados pela camada inferior. A camada de recursos permite a interacção da camada de serviços com seus recursos e dispositivos ligados à rede.

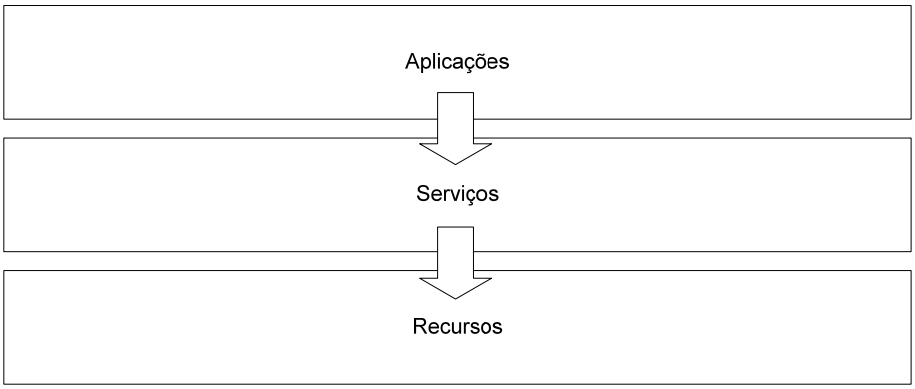

Figura 3.8 – Arquitectura lógica de referência.

3.6.2. Processo de implementação

A exposição de 'enablers' para terceiros cria a necessidade de utilizar interfaces abertas usando tecnologias já estabelecidas e aceites. Como se trata de um ambiente experimental transitório a nível da rede, dos serviços e processos, é importante a implementação faseada de modo a provocar o mínimo de disrupção na rede IMS já funcional aparentando aos seus utilizadores que apenas estão-lhes sendo fornecidos novos serviços, para além disso necessitará de ter uma restrição controlada dos serviços a disponibilizar e das alterações necessárias ao sistema actual para minimizar as potenciais vulnerabilidades do novo sistema. Assim numa fase inicial estender-se-á apenas o conceito de fornecimento de serviços ao utilizador final para incluir o acesso a estes através da Internet para além da rede de acesso típica. Deste modo os serviços básicos disponíveis ao assinante de um dispositivo móvel, voz, videoconferência, SMS e MMS, são disponibilizados 'On-line'. Será assim disponibilizada uma ou mais API's para os serviços existentes sendo preferível a utilização de uma API para cada serviço pois reduz a complexidade da implementação de cada um dos serviços facilitando a sua depuração e funcionamento ao mesmo tempo que a execução de cada serviço numa API separada permitirá uma melhor gestão de recursos e funcionalidades para além da disponibilização personalizada de cada um deles ao cliente juntamente com facturação apropriada e remetendo assim a autenticação do utilizador que será comum a todos para uma única API independente.

O acesso às API's a partir do exterior deverá ser permitido através da disponibilização de um servidor a partir do qual poderão ser invocados métodos. A utilização de serviços 'web' RESTful com mensagens SOAP será ideal pois permite uma melhor separação dos recursos em URI's e utiliza o protocolo HTTP, que formando a base da Internet e para o qual existe um grande suporte não necessita de alterações profundas e não irá causar problemas criados por 'firewalls' e NAT's por onde poderão ter que passar as mensagens, nos pedidos HTTP serão anexadas mensagens SOAP que sendo baseadas em texto utilizando XML preenchem as necessidades de transporte de tipos de dados, invocação dos métodos, segurança e descrição das interfaces. Assim cada API terá um URI específico e os serviços de autenticação poderão utilizar as funcionalidades de segurança permitidas pelo protocolo HTTPS e pelas mensagens SOAP. Esse servidor que actuará principalmente

como uma porta de entrada (‘gateway’) para a rede interna e conectar-se-á aos AS’s que disponibilizam os serviços. Haverá um certificador de SLA que poderá ser implementado em separado ou no servidor para que averigúe e aplique as regras de utilização para cada utilizador ou grupos de utilizadores. De modo a permitir a escalabilidade desta configuração a arquitectura de exposição é representada por 3 entidades lógicas, uma ‘gateway’ de exposição, um certificador de SLA’s e um fornecedor de serviços para permitir, desse modo, a adição de recursos, físicos ou lógicos, indo ao encontro das necessidades resultantes do aumento de utilizadores, da utilização de cada API e de novas API’s que se julguem importantes adicionar nesta fase da exposição de serviços.

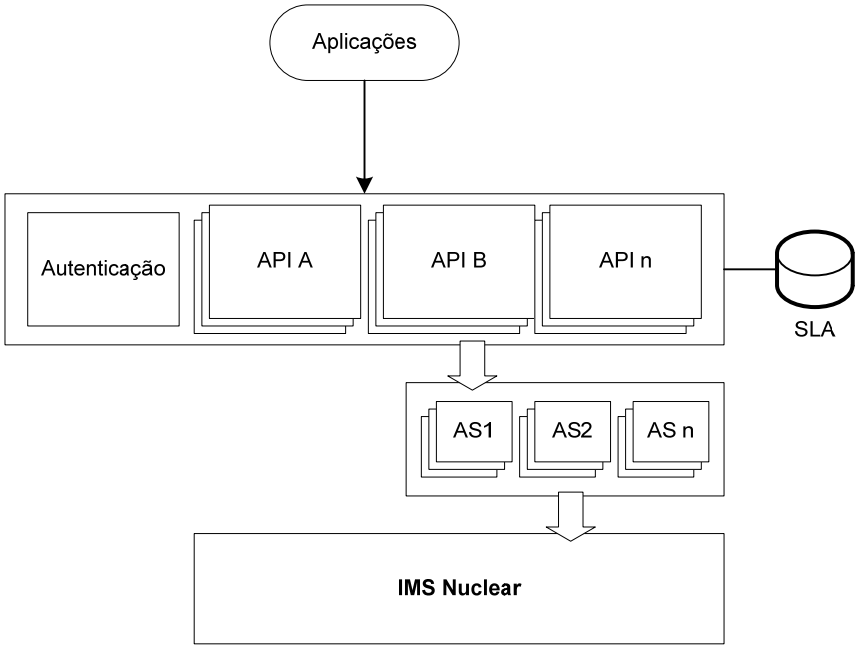

Figura 3.9 – Arquitectura de exposição.

Como se pode observar na figura 3.9, o ‘gateway’ de exposição conterá 1 ou mais servidores, um deles será predefinido para a API de autenticação que durante o processo atribuirá um dos servidores ao utilizador para a duração do registo e cada servidor será responsável pelo balanceamento de carga do tráfego para os vários AS’s presentes no fornecedor de serviços. O certificador de SLA’s poderá estar contido em cada servidor do ‘gateway’ de exposição ou será uma entidade lógica que conterá um elemento de

policiamento para todos os servidores da ‘gateway’ de exposição ou um para cada uma e obterá os dados de autenticação e SLA relativos ao utilizador durante a sua autenticação através de um AS específico para o efeito que obterá a informação necessária do HSS ou através de uma base de dados dedicada para o efeito e a armazenará pelo menos enquanto o registo do utilizador estiver activo.

Para que programadores e empresas externos possam utilizar e desenvolver aplicações que utilizem as API’s o lançamento de um SDK em uma ou mais linguagens de programação facilitará a implementação de aplicações e aumentará a adesão aos serviços. Deste modo já será possível a um utilizador criar aplicações para páginas de Internet e dispositivos que utilizam apenas as API’s dos serviços necessários permitindo à operadora personalizar os serviços oferecidos a cada cliente taxando-o de acordo com isso e melhorando a gestão dos recursos disponíveis.

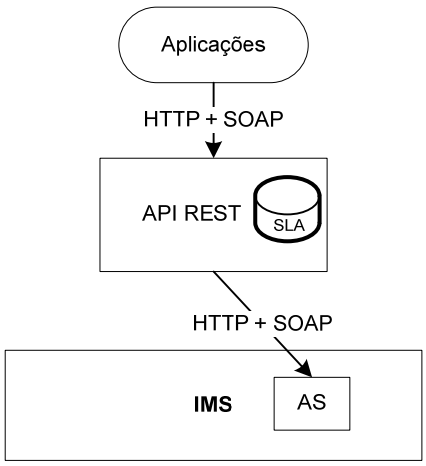

Figura 3.10 – Diagrama do servidor implementado.

Para esta fase inicial foi realizado um servidor com o seu conceito básico como se poderá observar na figura 3.10. Os requisitos apresentados foram a implementação de um ‘gateway’ de serviços num servidor ‘web’ Apache Tomcat fornecendo uma interface ‘web’ baseada em REST, com capacidade de fornecer limitações no serviço para cada cliente, um género de SLA simples. O serviço a expor foi o ‘click-to-call’, permitindo com que a partir de um cliente ‘web’ se possa iniciar uma chamada entre dois dispositivos com a limitação

de número máximo de chamadas a executar por um período de tempo para cada cliente. A troca de informação é feita com mensagens SOAP, o 'schema' XML já foi definido pelo ponto de contacto e o 'gateway' poderá ler e bloquear mas não deverá alterar as mensagens.

Para o seu desenvolvimento foi utilizada a linguagem Java com o IDE 'NetBeans' devido à sua estabilidade, integração com o Apache Tomcat e 'plugin' de suporte ao REST permitindo o desenvolvimento e teste da aplicação de uma maneira ágil e rápida.

Após esta fase inicial iniciar-se-á a implementação de um ambiente OSE e integrá-lo na rede IMS. Como alguns serviços são comuns entre o OSE e o IMS foi definido pela OMA o 'enabler' 'Utilization of IMS in OMA', ou 'IMSinOMA', com o objectivo de definir que funções e capacidades estandardizadas pelo IMS poderão ser reutilizadas e como é que os 'enablers' OMA as poderão utilizar evitando a integração vertical. OMA implementa o conceito de neutralidade às tecnologias de rede, desse modo apenas define a utilização de duas categorias de interfaces para comunicação com o exterior do ambiente OSE, a interface I0 e a interface I2. A interface I0 é utilizada para a interacção entre 'enablers' OMA e para a interacção com a camada de aplicações, neste caso estando sujeita a políticas, I0+P, como foi discutido no capítulo sobre OSE. A interface I2 compreende todas as interfaces IMS que um 'enabler' OMA poderá utilizar: ISC, Sh, Ut, Ro, Rf, Gm e Mb permitindo assim uma correlação entre as interfaces IMS e as OSE sem que OSE tenha que redefinir ou replicar as definições das interfaces utilizadas do IMS no âmbito das suas especificações tendo assim uma interface englobante e independente seguindo o principio da neutralidade de OMA. Como se pode ver na figura seguinte, figura 3.11, a camada de aplicações poderá interagir com algumas interfaces nucleares do IMS, com os serviços IMS fornecidos por 'enablers' IMS utilizando Parlay/OSA, SIP ou outro protocolo suportado e com a plataforma OMA OSE através da sua interface I0+P. Ao nível das interacções entre a camada de serviços e recursos da rede os serviços IMS interagem com os recursos IMS enquanto que o ambiente OMA OSE interage com os mesmos recursos mas limitado pelas interfaces permitidas no âmbito da interface I2. [107] [66]

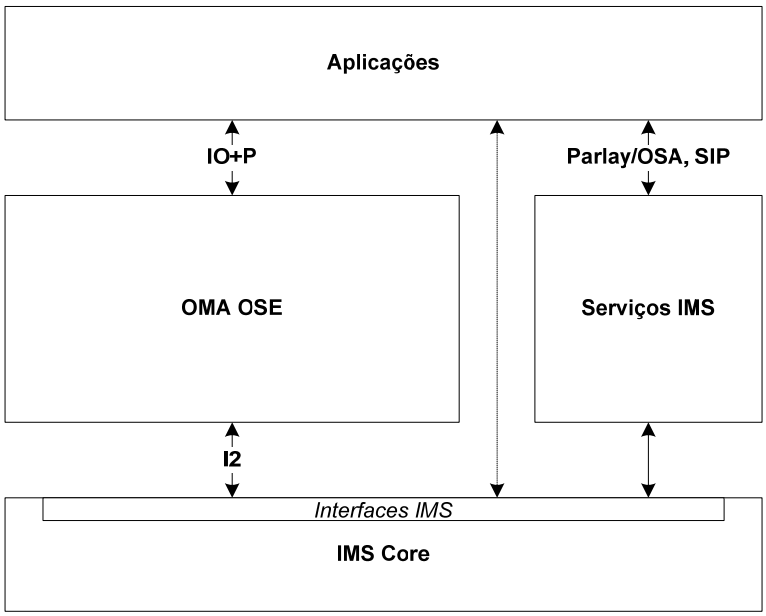

Figura 3.11 – OMA OSE e 3GPP IMS. [107] [66]

Ao nível da camada dos serviços existe uma separação dos que fazem parte do IMS e dos 'enablers' OMA no ambiente OSE pois existe um variado número de diferentes implementações de serviços IMS e 'enablers' OMA, incluindo algumas para as mesmas funcionalidades, deste modo poderão ser ambos aproveitados para uma mesma aplicação, para além disso os 'enablers' OMA deverão utilizar as capacidades comuns internas (ligações seguras, autenticação e autorização) e serviços do IMS (gestão de sessões, acesso a dados de utilizadores, subscrição de eventos e notificações, etc.) sempre que estes estiverem disponíveis. Deste modo conseguir-se-á uma plataforma conjunta onde será possível a orquestração ou coreografia de serviços entre o OSE e os serviços IMS cumprindo os princípios de reutilização e rápida criação de serviços. Nesta plataforma conjunta a interacção entre 'enablers' OMA e serviços IMS é assegurada pela interface I2 utilizando a rede nuclear IMS e entre diferentes 'enablers' OMA poderão se verificar 2 situações, uma onde é utilizada a interface I2 pelo desconhecimento da I0 ou para descobrir a I0. Ao nível de políticas o ambiente OSE conterá um 'Policy Evaluation, Enforcement and Management' (PEEM) fornecendo serviços de gestão, policiamento e aplicação de politicas para além da orquestração de serviços. Nesta fase os serviços expostos transitarão para um 'enabler' de serviços 'web' que os disponibilizará através do PEEM, assim uma aplicação 'web' poderá aceder a serviços expostos enviando pedidos à

interface ‘web’ implementada no PEEM permitindo com que serviços, antes acessíveis apenas aos dispositivos de rede dos assinantes, possam ser acedidos pela Internet disponibilizando a cada utilizador um conjunto de serviços de acordo com a sua assinatura tal como se tratasse de um acesso por um terminal. [107] [66] [64]

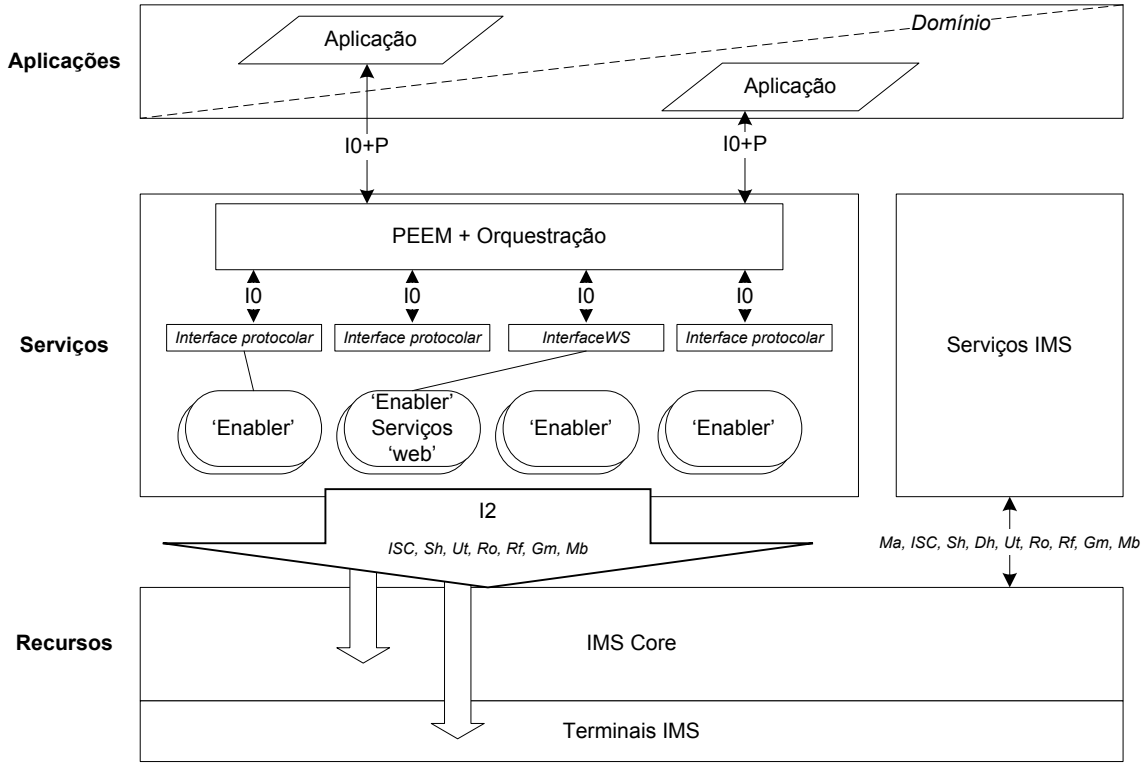

Figura 3.12 – OSE em IMS. [66] [107]

De modo a optimizar o funcionamento da estrutura será importante, mas não essencial, aumentar um pouco mais o nível de integração entre o OSE e o IMS estendendo funcionalidades do SCIM do IMS para o interior do ambiente de execução OSE utilizando um ‘enabler’ com funcionalidades SCIM (‘enabler’ SCIM), esse ‘enabler’ poderá actuar a só ou em conjunto com o SCIM AS da camada de serviços do IMS para a execução de micro-orquestração de recursos IMS aumentando funcionalidades e assim possibilidades de construção de serviços podendo até libertar o PEEM, ao nível de orquestração, de alguns procedimentos ficando exclusivamente dedicado à macro-orquestração. [66]

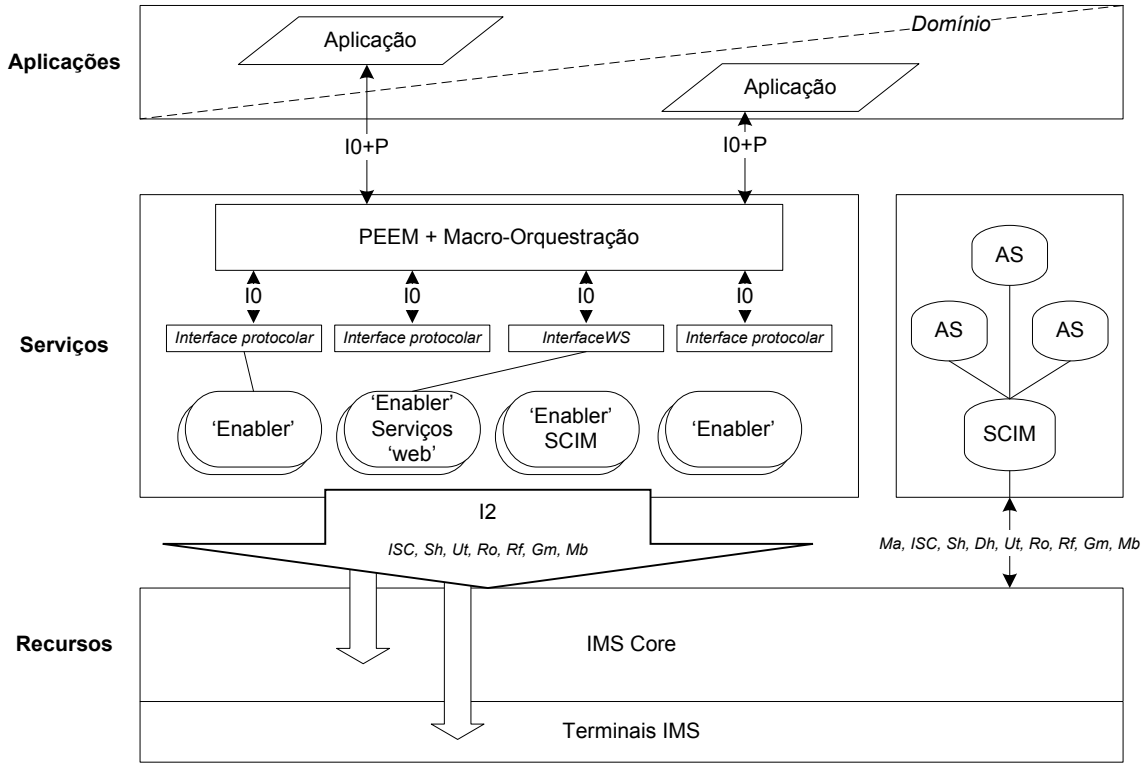

Figura 3.13 – OSE em IMS com 'enabler' SCIM e SCIM AS. [66] [107]

3.6.3. Consequências

Possibilitar a utilização de ‘enablers’ OMA numa rede IMS pela implementação de um ambiente OSE irá aumentar o nível de estandardização na industria levando a que mais fabricantes decidam investir em equipamentos direccionados que por sua vez impulsionará a implementação de novas redes e a redução de custos por economia de escala, sendo que no final se reduzirão os custos OPEX e CAPEX. Para os programadores será mais atractivo desenvolver para esta plataforma pois as especificações de enablers ‘OMA’ são independentes das tecnologias utilizadas permitindo com que novos ‘enablers’ e aplicações tenham mais tempo de vida e um maior espectro de aplicação o que beneficiará os utilizadores não só pelo aumento de serviços e opções mas também pela potencialidade de utilizarem os mesmos serviços em diferentes redes de um modo transparente. Também para as empresas com redes IMS a acessibilidade da rede IMS pela interface I2 permitirá suplantar os desafios técnicos criados pela especificidade de algumas implementações IMS

permitindo com que beneficiem da implementação de ‘enablers’ OMA. Com a exposição de serviços e a implementação de um ambiente aberto onde serviços podem ser criados e articulados segundo as necessidades personalizadas e na hora juntamente com o controlo do ciclo de vida destas explorará as potencialidades da arquitectura IMS e capacitará as empresas de telecomunicações de uma enorme versatilidade na construção de soluções personalizadas às necessidades dos clientes atingindo o objectivo principal que levou a esta enorme transição de redes, estruturas e modos de fazer negócio, o aumento do ARPU criando mais lucros. [107] [66]

4. Conclusão

Ao longo desta dissertação foi abordada a arquitectura IMS, os protocolos que a compõem, suas entidades, seus aspectos de fornecimento de qualidade-de-serviço, funcionalidades de segurança e sua estrutura de facturação. Foram abordados também os conceitos de SOA de um modo abrangente, do que se trata e seu paradigma, a definição mais genérica de um SDP que poderá reunir maior consenso tal como a sua arquitectura geral e a concretização de ambos no ambiente OSE. Finalmente a análise de API's de exposição de serviços disponibilizadas por algumas empresas de telecomunicações no âmbito da abertura dos recursos das suas redes para terceiros com o objectivo de tornar mais abrangente os seus modelos de negócio para o aumento da facturação e como se poderá aplicar a arquitectura OSE definida por OMA numa rede IMS para que desse modo se possam maximizar as capacidades e potencialidades da rede para o desenvolvimento e fornecimento de serviços disponibilizando-os não só para a rede IMS mas também expondo-os de modo análogo ao que foi verificado nas API's de exposição já existentes. A importância desta união é extrema pois só com a utilização de uma plataforma para a gestão, desenvolvimento e acesso de serviços é que a rede IMS se torna viável, sendo desenvolvida principalmente como uma arquitectura de sinalização em redes IP permitindo a interoperabilidade com as redes tradicionais de comutação de circuitos e tendo-se tornado mais tarde agnóstica à rede de acesso, mas deixou uma lacuna na sua camada de serviços onde necessita que se assegurem muitas das necessidades para o rápido desenvolvimento, reutilização e entrega destes que é colmatada por OSE integrado com o IMS através do 'enabler' 'IMSinOMA'.

Na situação de convergência de redes unida à necessidade das operadoras de aumentar a facturação devido à estagnação da actual facturação média por assinante nos serviços de voz e dados enfrentam-se grandes desafios e dificuldades para a criação de novas arquitecturas que estendam o conceito e alarguem o âmbito do que pode constituir um serviço, a sua a criação e seu fornecimento, e simultaneamente coexistam e interoperem com as redes tradicionais enquanto que possibilitem uma suave transição evolutiva, a arquitectura IMS resolve muitos desses desafios mas não sem suscitar algumas dúvidas

acerca da sua viabilidade como plataforma de sustentação para os novos ambientes baseados em serviços de disponibilização universal e apesar do IMS fornecer serviços através de servidores de aplicações essa capacidade apresenta-se mais como ponto ou uma interface de ligação a possíveis serviços. Torna-se então pertinente a explicitação de um modo resolver ou minimizar os restantes problemas dissipando muitas das dúvidas que põem em causa a viabilidade do IMS não como IMS singular mas o IMS agregado a outra plataforma. Mas nada vem sem as suas limitações, embora a reunião do IMS com o OSE contribua para o desenvolvimento do mercado móvel de telecomunicações e para a redução dos custos e riscos associados com o desenvolvimento e lançamento de aplicações e serviços devido principalmente à sua terceirização criando novos mercados e melhorias económicas para algumas empresas que os utilizem, podem-se prever situações em que não haja 'enablers' estandardizados por OMA adequados para facilitar a criação de uma nova classe de serviços o que significa que ou ter-se-á que perder tempo a especificar um novo 'enabler' ou esperar-se-á até que OMA o especifique, de qualquer caso perder-se-á tempo valioso que era suposto não se perder e, caso se siga pela primeira opção, criar-se-ão sistemas proprietários que também se estão a tentar evitar exactamente pela implementação desta plataforma. Também embora as especificações sejam bastante abrangentes não definindo tecnologias em específico existe a problemática de que são as tecnologias que fazem depender muitas das características essenciais do OSE sendo que como cada bloco lógico ou funcional é constituído por um número variável de sub-blocos, cada um com as suas especificidades, a implementação e consequente funcionalidade e praticabilidade do sistema final ainda terá o factor de imprevisibilidade associado a si, para além disso como a existência de limitações é intrínseca a qualquer especificação ou definição, será espectável que a entidade que as estabeleça se guie por um meio termo onde seja previsto o máximo necessário de casos com o mínimo possível de pontos.

Finalmente será de esperar que se critique a abordagem puramente teórica e conceptual destas arquitecturas, seus protocolos, entidades, blocos lógicos e suas interligações que aqui se apresenta pois não é apresentada uma implementação física e funcional do sistema final, caso que se encontra fora do âmbito da dissertação e cuja realização seria base para muitos mais capítulos devido à complexidade e extensão de sub-entidades, sub-blocos e sub-arquitecturas que suportarão todas as sub-funcionalidades às quais o sistema deveria ser agnóstico.

Especulando para o que poderá vir, a rede IMS está bastante preparada para a adição de novas entidades de interoperação tal como o OSE, ao aparecimento de novos protocolos de comunicação motivado pela maturação das redes ópticas, ou por qualquer outro caso, ou à evolução dos protocolos já existentes tanto no IMS como no OSE sendo que as suas especificações permitem com que novas interfaces ou 'enablers' possam ser criados sendo daí que vêm parte das vantagens destas plataformas. A nível de serviços e OSE será também de prever que o conceito de "serviços" evolua permitindo a criação de novas extensões, arquitecturas, ou até a inclusão do conceito de serviços inteligentes para além da evolução do OSE para a criação de uma arquitectura meta-OSE que alargue ainda mais as suas potencialidades.

Anexos

A. ‘WebServices’

i. Definição

Os serviços ‘web’, do inglês ‘Web Services’ (WS) representam a evolução dos conceitos de comunicação que têm conduzido a Internet durante anos, são aplicações modulares com uma lógica desenvolvida através de um conjunto de standards de Internet tendo uma interface acessível pela rede posicionando-se entre o código da aplicação em si e o consumidor das funcionalidades. Um WS não é suposto ser uma aplicação mas sim disponibilizar funções de aplicações num modo funcional ‘programa-a-programa’ através de uma dada rede, agindo como uma camada de abstracção que separa a plataforma de acesso dos detalhes específicos da linguagem utilizada e do modo como é executada a aplicação. Pode-se dizer então que os WS’s são uma arquitectura que serve de plataforma para a troca de mensagens de um modo agnóstico às tecnologias, ou linguagens de programação, utilizadas em cada uma das entidades intervenientes no processo, utilizando para isso uma combinação de protocolos de Internet estandardizados. A estandardização dessa arquitectura permite então com que possa haver uma grande interoperabilidade através de diferentes plataformas que seguem as suas directrizes. [73] [74] [75]

Os WS’s seguem uma arquitectura que se pode apresentar com dois níveis de abstracção, um ao nível das entidades envolvidas e outro ao nível das camadas protocolares utilizadas. Ao nível das entidades lógicas as envolvidas existe o fornecedor, o cliente e o registo dos serviços. O fornecedor implementa um ou mais serviços e disponibiliza-os na rede publicando as suas descrições no registo de serviços, o cliente procura o serviço que necessita no registo dos serviços e utiliza-o estabelecendo uma ligação com o seu fornecedor enviando posteriormente uma mensagem com o pedido e o registo dos serviços que funciona como um repositório dos serviços disponíveis e suas localizações fornecendo funcionalidades de publicação e procura de serviços como pode ser observado na imagem seguinte. [73] [74] [75]

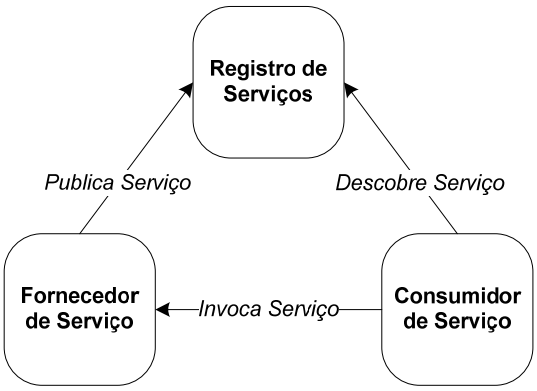

Figura A.1 – Entidades lógicas dos serviços 'web'. [75]

Ao nível das camadas protocolares os WS's são constituídos por 5 camadas como se pode observar na figura seguinte.

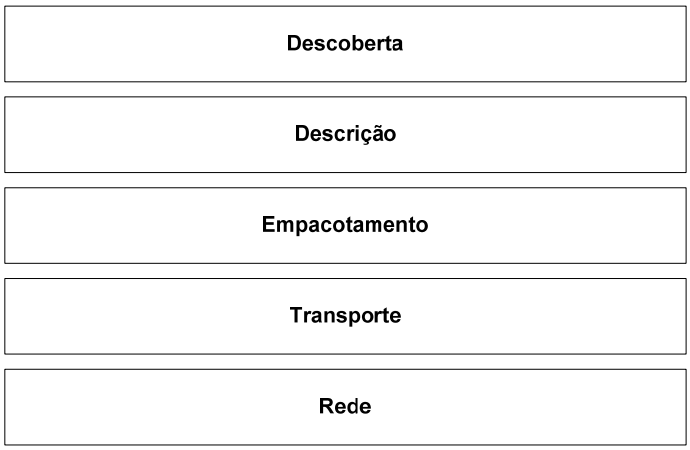

Figura A.2 – Pilha protocolar dos WS's. [75]

A camada 'Descoberta' refere-se ao mecanismo que permite aos consumidores dos serviços obter as descrições e localizações dos serviços existentes, tipicamente este papel é executado pelo 'Universal Description Discovery & Integration' (UDDI) que foi criado inicialmente por um conjunto de empresas, Microsoft, IBM e Ariba contendo um registo com a meta-informação de todos os WS's e os descritores WSDL de cada serviço publicado. Num UDDI a informação é organizada em 3 categorias: páginas brancas, amarelas e verdes. As páginas brancas contém informação geral tal como nome, descrição e endereço da companhia a que se refere o serviço, as amarelas contém uma classificação

geral da empresa ou serviço ao nível da industria ou produto e as verdes é que contém a informação técnica acerca do WS, um apontador para a sua especificação e um endereço para a invocação do serviço. A camada 'Descrição' refere-se à necessidade de descrever de um modo exacto e estandardizado as decisões tecnológicas e os métodos de um serviço tal como as suas especificidades. Existem várias soluções mas a mais utilizada é o 'Web Service Description Language' (WSDL). A camada 'Empacotamento' refere-se às tecnologias utilizadas para "serializar" a informação das mensagens a serem enviadas de modo a que todos os intervenientes possam compreender a informação trocada, tipicamente é utilizado o 'Simple Object Access Protocol' (SOAP) ou o XML-RPC. A camada 'Transporte' diz respeito aos protocolos de transporte utilizados para o envio da informação entre os intervenientes. Os WS podem ser desenvolvidos usando quase qualquer protocolo de transporte existente e a sua escolha baseia-se principalmente nas restrições impostas pelo ambiente em que o serviço deverá ser executado. O protocolo mais comum é o HTTP mas outros tais como, TCP, SMTP, Jabber, BEEP, FTP, JMS também podem ser utilizados. E a camada 'Rede' diz respeito às capacidades de endereçamento e encaminhamento necessárias para a correcta entrega das mensagens, conceptualmente semelhante à camada OSI com o mesmo nome. Nos serviços 'web' não há uma especificação conjunta que os englobe, em vez disso existem várias especificações que resolvem problemas em separado que são referidas em conjunto como 'WS-*'. Assim poderão ser utilizadas apenas as especificações necessárias para cada caso simplificando a sua implementação e aumentando as suas potencialidades derivado de não haver restrições causadas por especificações irrelevantes para a situação a ser resolvida. [74] [75] [66]

ii. WSDL

O 'Web Services Descrioption Language' (WSDL) é baseado no formato XML e fornece um modelo para a descrição de WS's podendo conter informação de todas as funções disponibilizadas, tipos dados, protocolos de transporte utilizados e localização do serviço fornecendo assim uma descrição estandardizada agnóstica à linguagem de programação

que possa ter sido utilizada para o desenvolvimento dos serviços que o WS fornece. Utilizando WSDL um cliente pode automaticamente localizar um determinado serviço de que necessita e saber como invocar cada uma das funções disponibilizadas. A descrição num WSDL abrange o formato das mensagens trocadas, suas especificidades e protocolos utilizados representando um contracto de comunicação entre o cliente e o fornecedor independentemente da plataforma e linguagem de programação. É tipicamente utilizado para descrever serviços que utilizam SOAP. Resumindo, um cliente usando o WSDL de um WS pode localiza-lo, saber que funções disponibiliza e como as poderá invocar. [74] [77]

De um modo geral um ficheiro WSDL é constituído pelos seguintes elementos: ‘<definition>‘ que especifica o nome do serviço e os ‘namespaces’ utilizados no documento, ‘<types>’ onde são declarados os tipos de dados utilizados entre o cliente e o fornecedor tais como ‘float’, ‘string’, ‘<message>’ que descreve uma mensagem unidireccional, seja pedido ou resposta, definindo o seu nome e poderá conter elementos ‘<part>’ referentes a parâmetros ou valores de retorno, o ‘<portType>’ define um porto de ligação entre as entidades referente a um conjunto pedido/resposta contendo elementos ‘<message>’, o ‘<binding>’ que fornece os detalhes de como o porto de ligação definido em ‘<portType>’ será transmitido, tipicamente é utilizado HTTP ou SOAP e o elemento ‘<service>’ que informa o endereço para a invocação do serviço especificado sendo tipicamente um URL. [74]

iii. SOAP

O SOAP é uma plataforma estandardizada e extensível para a troca de mensagens XML de um modo agnóstico à plataforma de transporte e linguagens de programação utilizadas. É principalmente focado na invocação remota de procedimentos, ‘Remote Procedure Call’ (RPC) e foi desenvolvido tendo em mente a simplicidade e expansibilidade fornecendo um mecanismo simples e leve para a troca de informação estruturada entre serviços de um

modo descentralizado num ambiente distribuído, reduzindo a complexidade e custos associados com a integração de aplicações desenvolvidas em diferentes plataformas. A versão 1.2 já não define SOAP como um acrónimo pois esta tecnologia já pode ser interpretada de dois modos: o 'Service Oriented Architecture Protocol' onde uma mensagens representa informação de invocação ou retorno de um serviço e o 'Simple Object Access Protocol' onde uma mensagem representa uma invocação de um método remoto. Como foi dito inicialmente o SOAP pode ser transportado utilizando qualquer protocolo de transporte mas tipicamente é utilizado o HTTP. [76] [74] [78] [77]

Estrutura de uma mensagem SOAP

Uma mensagem SOAP consiste de um envelope SOAP que contém o seu nome local, definições de 'namespace', um cabeçalho SOAP cuja presença é opcional e um corpo SOAP.

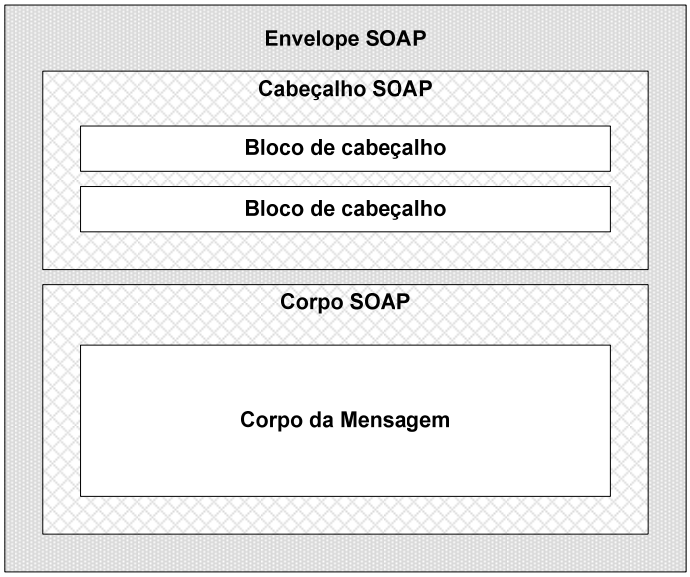

Figura A.3 – Estrutura da mensagem SOAP. [75]

Um cabeçalho SOAP fornece um mecanismo para estender uma mensagem SOAP num modo descentralizado e modular fornecendo blocos de informação relevantes para o modo de como a mensagem será processada tais como configurações de encaminhamento e entrega, autenticação, autorização e contextos transaccionais. Cada cabeçalho conterá o seu

nome local, definições de ‘namespace’ e blocos de cabeçalho SOAP, cada bloco de cabeçalho SOAP conterá um nome de ‘namespace’ e poderá conter atributos relacionados com a entidade a que está destinado o bloco tais como a obrigatoriedade de suporte ao bloco em questão e a permanência o não do bloco na mensagem ao longo do seu percurso para além outras possíveis especificidades. Um corpo SOAP fornece um mecanismo para a transmissão de informação para um destinatário SOAP contendo a mensagem propriamente dita a ser entregue e é constituído pelo seu nome local, definições de ‘namespace’ e elementos filho. Cada elemento filho de um corpo SOAP tem um ‘namespace’ associado e vários atributos relacionados com a informação a ser enviada. Em caso de erro é utilizado um ‘SOAP Fault’ para transportar informações relacionadas com a falha tais como o código da falha e razão da falha, também poderá informar em que nó ocorreu a falha, o seu papel e outros detalhes relacionados com a falha, conterá também um nome local e um nome de ‘namespace’. Um ‘SOAP Fault’ será enviado dentro do corpo de uma mensagem SOAP como único filho e apenas poderá conter uma falha. [78] [75]

iv. REST

O ‘Representation State Transfer’ (REST) foi introduzido pela primeira vez em 2000 por Roy Fielding na sua dissertação de doutoramento e difere de um WS típico pela disponibilização de uma interface uniforme cujas operações para a manipulação de recursos remetem para os métodos HTTP PUT, GET, POST e DELETE funcionando ao nível da camada de aplicação do HTTP sendo cada recurso identificado unicamente por um URI, para além disso a manipulação de recursos é executada através da troca de representações do mesmo. O método PUT é utilizado para a criação de um novo recurso, o GET permite a obtenção do estado actual de um recurso sob a forma de uma representação “serializável”, o POST transfere um novo estado a um recurso já existente e o DELETE apaga um recurso existente. A potencialidade do REST vem do facto de uma mensagem HTTP poder conter informação de carga que nesta situação será tipicamente um formato

baseado em XML tal como SOAP que será manipulado consoante o método HTTP invocado e também porque se trata de uma reutilização de um protocolo que forma a base da Internet de um modo não disruptivo e sem necessitar de precauções exceptionais durante o seu lançamento e utilização. Isto tudo em oposto aos WS's típicos onde a interface expõe um conjunto de operações não estandardizadas invocáveis para o mesmo URI para a manipulação de vários recursos sendo a sua utilização do HTTP apenas ao nível de transporte não aproveitando essa potencialidade, embora possa desse modo utilizar vários protocolos de transporte para além deste também porque funciona de um modo 'stateless' onde o significado de um pedido não será influenciado por pedidos anteriores, podendo também funcional de modo 'stateful' havendo para isso várias técnicas. Para além disso o REST não necessita de um descritor da interface tal como os WS's típicos necessitam do WSDL pois as sua interface é simples e estandardizada necessitando apenas de saber os formatos e detalhes do serviço disponibilizado em determinado URI sendo para isso proposto o 'Web Aplication Description Language' (WADL) para esse efeito facilitando ainda mais a geração automática de código para sua utilização. [83] [81] [80] [77]

v. WADL

O 'Web Application Description Language' (WADL) foi criado para executar o mesmo papel que o WSDL mas para os serviços REST. O WADL descreve as funcionalidades presentes na interface standard do REST e o modo como o cliente deverá manipular o estado do recurso abstraindo os detalhes referentes aos pedidos HTTP e sua construção e análise sintáctica sem esconder a sua interface HTTP uniforme. No início a necessidade de um descritor como o WADL para o REST era discutível mas à medida que foi sendo utilizado as especificações dos seus serviços foram aumentado de complexidade apesar da sua interface simples sendo cada vez mais pertinente a sua existência. Com o WADL um programador com uma ferramenta de desenvolvimento preparada poderá gerar o código necessário para a invocação do serviço que doutro modo teria que criar manualmente

baseando-se nas especificações disponibilizadas do serviço. Mesmo assim o WADL não é tão necessário como o WSDL para os WS baseados em SOAP. [80]

De um modo simplificado, um ficheiro WADL tem como elemento raiz ‘<application>’ que poderá conter zero ou mais elementos ‘<doc>’, um elemento opcional ‘<grammars>’, outro elemento opcional ‘<resources>’ e zero ou mais elementos de ‘<resource_type>’, ‘<method>’, ‘<representation>’ e ‘<fault>’. O elemento ‘<doc>’ poderá conter a definição da linguagem e a descrição do serviço a que se refere o ficheiro. O elemento ‘<grammars>’ contêm definições do formato da informação trocada durante a execução do protocolo descrito no próprio ficheiro. O elemento ‘<resources>’ contém uma definição do URI base e descendentes ‘<resource>’ descrevendo recursos disponibilizados pela aplicação do serviço, cada um identificado por um URI que seguirá um padrão comum. O elemento ‘<resource_type>’ descreve os métodos que perfazem um recurso e seu comportamento. O elemento ‘<method>’ descreve os dados de entrada para o pedido e os dados de saída para a resposta. O elemento ‘<representation>’ descreve o estado de um recurso podendo definir ou referenciar uma definição. E o elemento ‘<fault>’, que é semelhante ao ‘<representation>’ no modo em que define um estado do recurso, define condições de erro podendo haver partilha de códigos de estado do HTTP e desse modo poder especificar melhor várias condições de erro ou fornecer informação equivalente em formato diferente. [82]

E do ponto de vista de expansibilidade, a maior parte dos elementos definidos no WADL são extensíveis utilizando elementos ou atributos de ‘namespaces’ externos. [82]

B. ‘Business Process Execution Language’ (BPEL)

i. Definição

Para a automação dos processos de negócio num mundo de Internet baseado em serviços será necessária uma plataforma que forneça uma capacidade de definição e estandardização desses mesmos processos, para isso foi criada a linguagem para a execução de processos de negócio, ‘Business Process Execution Language’ (BPEL), também apelidada de ‘BPEL for Web Services’ (BPEL4WS). A sua especificação inicial foi escrita em conjunto pelas empresas, IBM, Microsoft e BEA baseando-se conceptualmente em linguagens já existentes, o ‘Web Services Flow Language’ (WSFL) criado pela IBM, o XLANG criado pela Microsoft e o ‘Process Definition for Java’ (PD4J) criado pela BEA, principalmente convergindo o conceito de grafos direccionados do WSFL com a estruturação em blocos do XLANG daí resultando a versão 1.0 em Julho de 2002 e utilizando o PD4J como base para a definição de uma extensão para o Java apelidada de BPELJ, mais tarde foi lançada a versão 1.1 com a contribuições vindas da SAP e da Siebel Systems em Maio de 2003, ao mesmo tempo a especificação foi submetida para a ‘Organization for the Advancement of Structured Information Standards’ (OASIS) formando-se o comité WS-BPEL para a sua estandardização resultando na especificação WS-BPEL2.0 em Abril de 2007, neste caso com a colaboração de dezenas de organizações, algum tempo após foi criado o comité BPEL4People para definir uma extensão ao WS-BPEL permitindo a definição de tarefas humanas como parte de um processo.

O WS-BPEL é então uma linguagem de descrição da lógica de controlo para a orquestração de diferentes sistemas que podem estar isolados ou de diferentes parceiros de negócio permitindo a criação e gestão de processos de negócio e deste modo facilitar e permitir o seu continuo melhoramento e optimização trazendo um valor acrescentado às empresas que expõem funcionalidades pelo meio de serviços ‘web’. Utiliza o XPath para a manipulação de dados, especifica um ‘XML Schema’ com um vocabulário próprio, utiliza SOAP e descritores de serviços ‘web’ WSDL sendo na realidade uma extensão aos

serviços ‘web’ já existentes que, por sua vez, são utilizados como blocos de construção para a construção de um serviço ‘web’ englobante de maior funcionalidade do qual não necessitam de saber que fazem parte. [67] [68] [69] [72] [70]

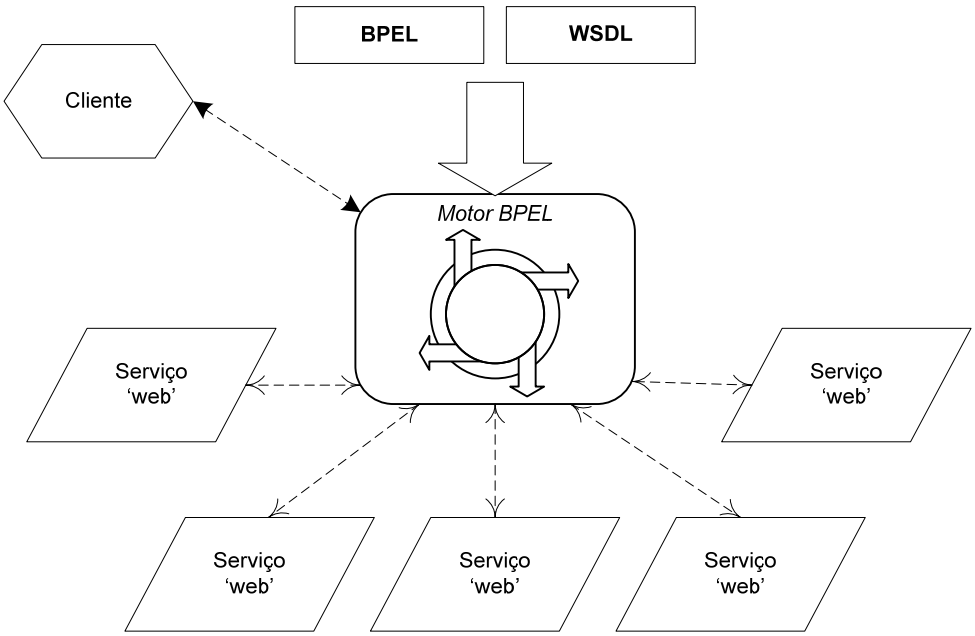

Figura B.1 – Posicionamento de um motor BPEL. [69]

ii. Processos BPEL

Um processo BPEL pode ser executável seguindo o paradigma de orquestração, onde um processo central tem controlo dos serviços ‘web’ envolvidos no processo coordenando as suas execuções sem a necessidade de estes saberem ou não se fazem parte de um processo maior, ou abstracto que segue o paradigma de coreografia que, ao contrário da orquestração, não depende de um processo central mas de um esforço de colaboração onde todos os participantes necessitam de estar conscientes das operações a executar e dos tempos de acção com cada um dos seus parceiros. Um processo executável é composto por um conjunto de serviços ‘web’ que são utilizados segundo o algoritmo definido no

processo e consequente manipulação de dados, mais, define a estrutura das mensagens de entrada e saída, tratamento de excepções e deverá ser executado num motor de processos BPEL. Um processo abstracto define um protocolo comportamental que pode ser observado externamente descrevendo o fluxo de mensagens da interacção de uma entidade com os seus parceiros sendo muito semelhante a um processo executável do ponto de vista do código embora só com verificação de dados em actividades modeladoras da interacção e do fluxo, é utilizado tipicamente para definir o comportamento de um serviço independentemente do processo de negócio a que pertence ou para definir um protocolo de colaboração entre múltiplos parceiros, descrevendo o comportamento de cada um, sendo muito adequado para aplicar regras, normalmente ambíguas, expressas em linguagem natural. [67] [69]

Um processo BPEL pode ser executado de modo síncrono ou assíncrono, no modo síncrono o cliente terá de esperar pela resposta do processo que será enviada após a conclusão deste enquanto que no modo assíncrono o cliente não necessitará de esperar, em vez disso poderá receber uma confirmação do pedido feito e receberá a resposta mais tarde identificada de modo a poder ser correlacionada com o pedido correspondente. O mesmo se passa com as ligações de parceria de um processo BPEL, poderão ser síncronas ou assíncronas pois todos os parceiros, incluindo o próprio processo BPEL, são disponibilizados como serviços 'web'. [67]

iii. Objectos BPEL

Do ponto de vista de programação a linguagem BPEL suporta mais de 50% das 'patterns' de programação P4. Na figura seguinte, figura B.2, apresenta-se a estrutura funcional da linguagem e suas relações. [69]

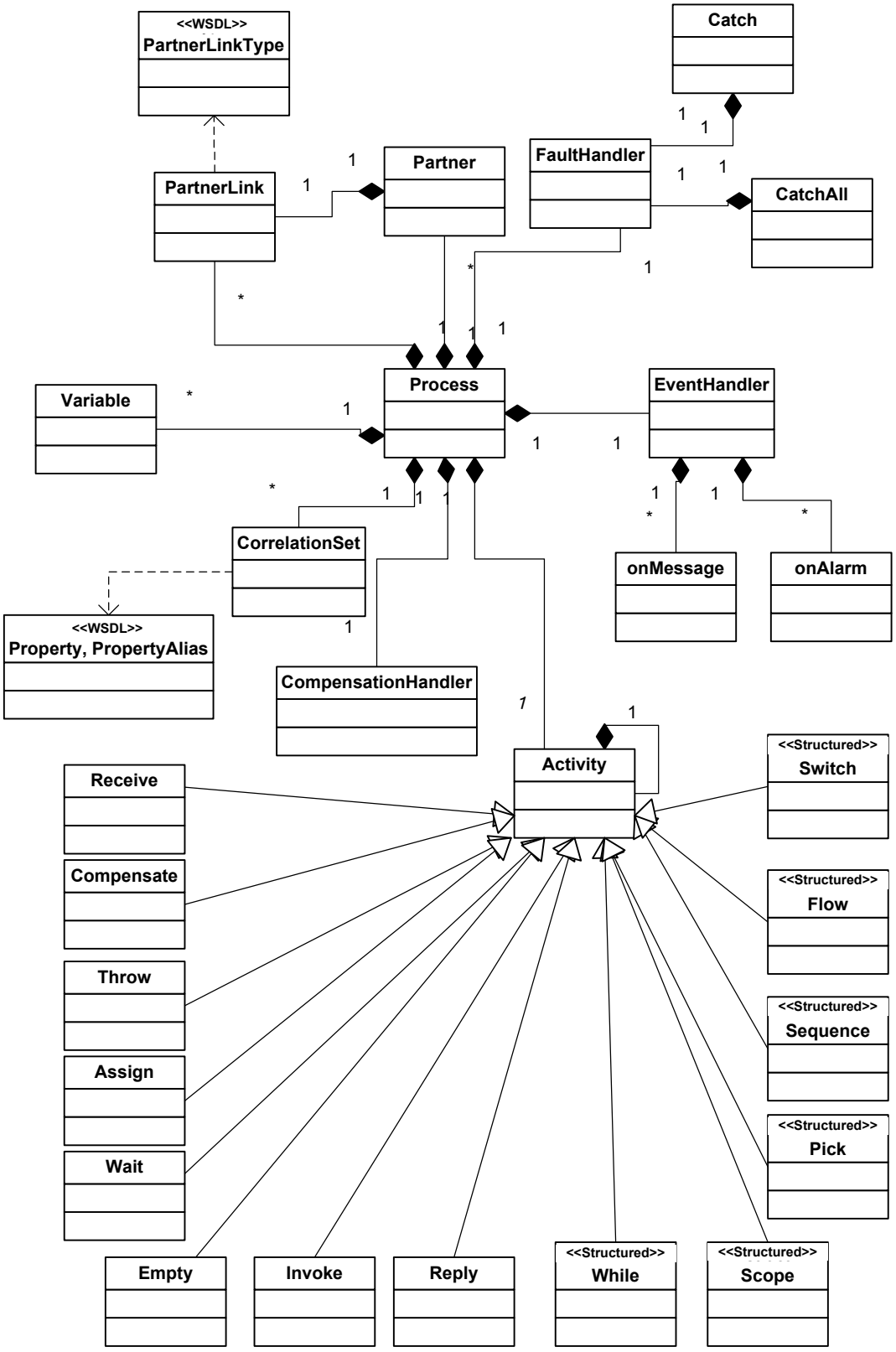

Figura B.2 – Modelo de objectos BPEL. [69]

Os processos BPEL para serem lançados num motor BPEL necessitam de 2 grupos de ficheiros, um grupo de ficheiros WSDL especificando as várias ligações de parceria, operações disponíveis, tipos de portos e mensagens trocadas entre o processo e os vários serviços 'web' utilizados e um grupo de ficheiros BPEL cada um contendo a definição em XML de um processo incluindo as suas actividades principais, ligações de parceria, variáveis, falhas e seus compensadores e eventos. [69]

Existe uma enorme variedade de servidores BPEL comerciais e de código aberto que fornecem um ambiente de execução de processos de negócio. A título de exemplo existe o 'Oracle BPEL Process Manager', o 'Microsoft BizTalk' e o 'FuegoBPM Software Suite' da 'BEA WebLogic' como servidores comerciais e o 'ActiveBPEL', o 'Apache ODE' e o 'jBPM' da JBoss como servidores de código aberto. Para o design e desenvolvimento de processos BPEL existem várias ferramentas de edição gráfica tipicamente adaptadas a um servidor BPEL específico pois algumas empresas definem extensões BPEL às especificações existentes. Por exemplo a Oracle tem o 'Oracle JDeveloper' e um 'plugin' para o Eclipse e a JBoss tem um 'plugin' para o Eclipse.

Referências

[1] **The 3G IP Multimedia Subsystem (IMS) : Merging the Internet and the Cellular Worlds** [Livro] / autor Camarillo Gonzalo e Garcia-Martin Miguel A.. - [s.l.] : John Wiley & Sons, Ltd, 2006. - 2nd Edition. - ISBN: 0-470-01818-6.

[2] **The IP Multimedia Subsystem (IMS) : Session Control and Other Network Operations** [Livro] / autor Russell Travis. - [s.l.] : McGraw-Hill, 2008. - ISBN: 0-07-159464-7.

[3] **Carrier Grade Voice Over IP - Second Edition** [Livro] / autor Collins Daniel. - [s.l.] : McGraw-Hill, 2003. - 2n Edition. - ISBN: 0-07-140634-4.

[4] **Signaling in Telecomunication Networks** [Livro] / autor Bosse John G. van e Devetak Fabrizio U.. - [s.l.] : John Wiley & Sons, Inc., 2007. - 2nd Edition. - ISBN: 0-471-66288-7.

[5] **Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol** [Livro] / autor Sinnreich Henry e Johnson Alan B.. - [s.l.] : Wiley Publishing, Inc., 2006. - 2nd Edition. - ISBN: 0-471-77657-4.

[6] **IMS Multimedia Telephony over Cellular Systems: VoIP Evolution in a Converged Telecommunication World** [Livro] / autor Chakraborty Shyam [et al.]. - [s.l.] : John Wiley & Sons, Ltd, 2007. - ISBN: 978-0-470-05855-8.

[7] **SIP Demystified** [Livro] / autor Camarillo Gonzalo. - [s.l.] : McGraw-Hill, 2002. - ISBN: 0-07-141462-2.

[8] **The IMS IP Multimedia Concepts and Services** [Livro] / autor Poikselkä Miikka [et al.]. - [s.l.] : John Wiley & Sons, Ltd, 2006. - 2nd Edition. - ISBN: 0-470-01906-9.

[9] **Convergence Technologies for 3G Networks: IP, UMTS, EGPRS and ATM** [Livro] / autor Bannister Jeffrey, Mather Paul e Coope Sebastian. - [s.l.] : John Wiley & Sons, Ltd, 2004. - ISBN: 0-470-86091-X.

[10] **Value-Added Services for Next Generation Networks** [Livro] / autor Velde Thierry Van de. - [s.l.] : Auerbach Publications, 2008. - ISBN: 978-0-8493-7318-3.

[11] **Network Convergence: Services, Applications, Transport, and Operations Support** [Livro] / autor Hanrahan Hu. - [s.l.] : John Wiley & Sons, Ltd, 2007. - ISBN: 978-0-470-02441-6.

[12] **Mobile Messaging - Technologies And Services: SMS, EMS and MMS** [Livro] / autor Bodic Gwenaël Le. - [s.l.] : John Wiley & Sons, Ltd, 2005. - 2nd Edition. - ISBN: 0-470-01143-2.

[13] **3GPP TS 23.002 version 8.2.0 Release 8** [Online] / autor ETSI // ETSI Publications Download Area. - Janeiro de 2008. - http://pda.etsi.org/pda/.

[14] **3GPP TS 23.228 version 8.5.0 Release 8** [Online] / autor ETSI // ETSI Publications Download Area. - Janeiro de 2008. - http://pda.etsi.org/pda/.

[15] **3GPP TS 23.167 version 7.9.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Junho de 2008. - http://pda.etsi.org/pda/.

[16] **3GPP TS 29.162 version 7.1.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Março de 2006. - http://pda.etsi.org/pda/.

[17] **3GPP TS 29.163 version 7.11.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Junho de 2008. - http://pda.etsi.org/pda/.

[18] **3GPP TS 29.332 version 7.10.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Abril de 2008. - http://pda.etsi.org/pda/.

[19] **3GPP TS 33.210 version 7.3.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Outubro de 2007. - http://pda.etsi.org/pda/.

[20] **3GPP TS 33.203 version 7.9.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Abril de 2008. - http://pda.etsi.org/pda/.

[21] **3GPP TS 33.102 version 7.1.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Dezembro de 2006. - http://pda.etsi.org/pda/.

[22] **3GPP TS 33.310 version 7.1.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Setembro de 2006. - http://pda.etsi.org/pda/.

[23] **3GPP TS 32.240 version 7.2.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Março de 2007. - http://pda.etsi.org/pda/.

[24] **3GPP TS 32.260 version 7.5.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Julho de 2008. - http://pda.etsi.org/pda/.

[25] **3GPP TS 32.296 version 7.0.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Junho de 2007. - http://pda.etsi.org/pda/.

[26] **3GPP TS 32.299 version 7.7.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Outubro de 2007. - http://pda.etsi.org/pda/.

[27] **3GPP TS 32.295 version 7.0.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Junho de 2007. - http://pda.etsi.org/pda/.

[28] **3GPP TS 23.207 version 7.0.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Junho de 2007. - http://pda.etsi.org/pda/.

[29] **3GPP TS 23.107 version 7.1.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Outubro de 2007. - http://pda.etsi.org/pda/.

[30] **3GPP TS 23.203 version 7.7.0 Release 7** [Online] / autor ETSI // ETSI Publications Download Area. - Junho de 2008. - http://pda.etsi.org/pda/.

[31] **3GPP2 X.S0013-000-B** [Online] / autor 3GPP2 TSG-X // 3GPP2 Specifications. - Dezembro de 2007. - http://www.3gpp2.org/Public_html/Specs/X.S0013-000-B_v1.0_080224.pdf.

[32] **OpenIMSCore** [Online] // OpenIMSCore.org | The Open IMS Core Project. - Outubro de 2008. - http://www.openimscore.org.

[33] **SOA Security** [Livro] / autor Kanneganti Ramarao e Chodavarapu Prasad. - [s.l.] : Manning Publications Co., 2008. - ISBN: 1-932394-68-0.

[34] **SOA Approach to Integration: XML, Web services, ESB, and BPEL in real-world SOA projects** [Livro] / autor Juric Matjaz B. [et al.]. - [s.l.] : Packt Publishing, 2007. - ISBN: 978-1-904811-17-6.

[35] **The New Language of Business: SOA & Web 2.0** [Livro] / autor Carter Sandy. - [s.l.] : IBM Press, 2007. - ISBN: 0-13-195654-X.

[36] **Patterns: Implementing Self-Service in a SOA Environment** [Online] / autor Sadtler Carla [et al.] // IBM Redbooks. - 2008. - 17 de Abril de 2008. - http://www.redbooks.ibm.com/.

[37] **Patterns: Implementing an SOA Using an Enterprise Service Bus** [Online] / autor Keen Martin [et al.] // IBM Redbooks. - 2008. - 18 de Abril de 2008. - http://www.redbooks.ibm.com.

[38] **SOA in Practice** [Livro] / autor Josuttis Nicolai M.. - [s.l.] : O'Reilly Media, Inc, 2007. - ISBN: 0-596-52955-4.

[39] **RFC1889: RTP: A Transport Protocol for Real-Time Applications** [Online] / autor Group Audio-Video Transport Working [et al.] // ietf.org. - Janeiro de 1996. - Maio de 2008. - http://www.ietf.org/rfc/rfc1889.txt.

[40] **SDP & SOA: Progress Report** [Online] / autor Chappell Caroline // Lightreading. - 8 de Novembro de 2006. - Julho de 2008. - http://www.lightreading.com/document.asp?doc_id=110199.

[41] **The SDP Alliance** [Online]. - Outubro de 2008. - http://www.thesdpalliance.com/.

[42] **Service Delivery Platform: A sua operadora ainda vai ter um... e logo.** [Online] / autor Vuono Evandro // teleco.com.br. - Julho de 2008. - http://www.teleco.com.br/hp/hp_artigos006.asp .

[43] **Parlay X Web Services Specifications** [Online] // Parlay :: Parlay/OSA Specifications. - Outubro de 2008. - http://www.parlay.org/en/specifications/pxws.asp.

[44] **Web21C Developer Center** [Online] // Web21C SDK. - Outubro de 2008. - http://web21c.bt.com/.

[45] **Orange** [Online] // Orange Partner. - Outubro de 2008. - http://www.orangepartner.com/.

[46] **Deutche Telecom**[Online] // Developer Portal. - Outubro de 2008. - http://developer.telekom.de.

[47] **Open Movilforum** [Online] // Open movilforum | La comunidade abierta de movilforum. - Outubro de 2008. - http://open.movilforum.com/.

[48] **Ericsson Service Delivery Platform: The fast, cost-eficient and reliable way to create, introduce and manage richer communication services and media services** [Online]. - Julho de 2008. - http://archive.ericsson.net/service/internet/picov/get?DocNo=28701-FGB101265.

[49] **Parlay X Web Services Specifications** [Online] // Parlay :: Parlay/OSA Specifications. - Outubro de 2008. - http://www.parlay.org/en/specifications/pxws.asp.

[50] **OMA Service Environment: Approved Version 1.0.4 – 01 Feb 2007** [Online] / autor OMA. - Julho de 2008. - http://www.openmobilealliance.org/.

[51] **HP Service Delivery Platform: A white paper from HP** [Online]. - Julho de 2008. - http://h71028.www7.hp.com/ERC/downloads/4AA1-6286ENW.pdf.

[52] **A sigla IMS chega ao Brasil com discurso promissor - I** [Online] / autor Fonseca João Carlos. - Maio de 2007. - Julho de 2008. - http://www.telebrasil.org.br/ impressao/artigos.asp?m=486.

[53] **A sigla IMS chega ao Brasil com discurso promissor – II** [Online] / autor Fonseca João Carlos. - Novembro de 2007. - Julho de 2008. - http://www.telebrasil.org.br/impressao/artigos.asp?m=488.

[54] **OpenSOA Evolution** [Online]. - Julho de 2008. - http://www.fokus.fraunhofer.de/.

[55] **SDP & IMS Architectures Mutually Exclusive or complementary?** [Online] / autor Kimbler Krzysztof. - 2006. - Julho de 2008. - http://pl.sun.com/sunnews/events/2007/transition_into_sdp/pdf/krzysztof_kimbler-sdp_and_ims_architectures.pdf.

[56] **Parlay/OSA: From Standards to Reality** [Livro] / autor Unmehopa Musa, Vemuri Kumar e Bennet Andy. - [s.l.] : John Wiley & Sons, Ltd, 2006. - ISBN: 0-470-02595-6.

[57] **Next Generation Inteligent Networks** [Livro] / autor Zuidweg Johan. - [s.l.] : Artech House, 2002. - ISBN: 1-58053-263-2.

[58] **RFC2543: SIP: Session Initiation Protocol** [Online] / autor Handley M. [et al.] // ietf.org. - Março de 1999. - Maio de 2008. - http://www.ietf.org/rfc/rfc2543.txt.

[59] **Ribbit** [Online] // Ribbit Developer Platform. - Outubro de 2008. - http://developer.ribbit.com/.

[60] **RFC4083: Input 3rd-Generation Partnership Project (3GPP) Release 5 Requirements on the Session Initiation Protocol (SIP)** [Online] / autor Garcia-Martin M. e Nokia // ietf.org. - Março de 2005. - Maio de 2008. - http://www.ietf.org/rfc/rfc4083.txt.

[61] **Sipgate** [Online] // sipgate.co.uk - free phone service over your broadband link. - Outubro de 2008. - http://www.sipgate.co.uk/ .

[62] **RFC3261: SIP: Session Initiation Protocol** [Online] / autor Rosenberg J. [et al.] // ietf.org. - Junho de 2002. - Maio de 2008. - http://www.ietf.org/rfc/rfc3261.txt.

[63] **RFC3966: The tel URI for Telephone Numbers** [Online] / autor Schulzrinne H. e University Columbia // ietf.org. - Dezembro de 2004. - Maio de 2008. - http://www.ietf.org/rfc/rfc3966.txt.

[64] **The OpenPEEM as core for service orchestration within the Open IMS Playground at FOKUS** [Online] / autor Magedanz T., Dutkowski S. e Gil-Laich Y. Gerat. - Setembro de 2008. - http://www.icin.biz/files/programmes/Poster-6.pdf.

[65] **The OpenPEEM as core for service orchestration within the Open IMS Playground at FOKUS** [Online] / autor Magedanz T., Dutkowski S. e Gil-Laich Y. Gerat. - Setembro de 2008. - http://www.icin.biz/files/programmes/Poster-6.pdf.

[66] **The Open Mobile Alliance: Delivering Service Enablers for Next-Generation Applications** [Livro] / autor Brenner Michael e Unmehopa Musa. - [s.l.] : John Wiley & Sons, Ltd, 2008. - ISBN: 978-0-470-51918-9.

[67] **Business Process Execution Language for Web Services** [Livro] / autor Juric Matjaz B., Mathew Benny e Sarang Poornachandra. - [s.l.] : Packt Publishing Ltd., 2006. - 2nd Edition. - ISBN: 1-904811-81-7.

[68] **SOA and WS-BPEL** [Livro] / autor Vasiliev Yuli. - [s.l.] : Packt Publishing Ltd., 2007. - ISBN: 978-1-847192-70-7.

[69] **Essencial Business Process Modeling** [Livro] / autor Havey Mike. - [s.l.] : O'Reilly, 2005. - ISBN: 0-596-00843-0.

[70] **Web Services Business Process Execution Language Version 2.0 Primer** [Online]. - Maio de 2007. - Outubro de 2008. - http://docs.oasis-open.org/wsbpel/2.0/Primer/wsbpel-v2.0-Primer.pdf.

[71] **Business Process Execution Language for Web Services Version 1.1** [Online]. - Maio de 2003. - Setembro de 2008. - http://download.boulder.ibm.com/ibmdl/pub/ software/dw/specs/ws-bpel/ws-bpel.pdf.

[72] **Service Oriented Enterprises** [Livro] / autor Khoshafian Setrag. - [s.l.] : Auerbach Publications, 2007. - ISBN: 0-8493-5360-2.

[73] **Web Services: Theory and Practice** [Livro] / autor Gurugé Anura. - [s.l.] : Elsevier Inc., 2004. - ISBN: 1-55558-282-6.

[74] **Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL** [Livro] / autor Cerami Ethan. - [s.l.] : O'Reilly, 2002. - ISBN: 0-596-00224-6.

[75] **Programming Web Services with SOAP** [Livro] / autor Tidwell Doug, Snell James e Kulchenko Pavel. - [s.l.] : O'Reilly, 2001. - ISBN: 0-596-00095-2.

[76] **Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More** [Livro] / autor Weerawarana Sanjiva [et al.]. - [s.l.] : Prentice Hall PTR, 2005. - ISBN: 0-13-148874-0.

[77] **Web Services Architecture** [Online] / autor W3C Working Group. - Fevereiro de 2004. - Setembro de 2008. - http://www.w3.org/TR/ws-arch/wsa.pdf .

[78] **SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)** [Online] / autor W3C. - Abril de 2007. - Setembro de 2008. - http://www.w3.org/TR/soap12-part1/ .

[79] **SOAP Version 1.2 Part 0: Primer (Second Edition)** [Online] / autor W3C. - Abril de 2007. - Setembro de 2008. - http://www.w3.org/TR/soap12-part0/.

[80] **Restful Web Services** [Livro] / autor Richardson Leonard e Ruby Sam. - [s.l.] : O'Reilly, 2007. - ISBN: 0-596-52926-0.

[81] **SOA Using Java Web Services** [Livro] / autor Hansen Mark D.. - [s.l.] : Prentice Hall, 2007. - ISBN: 0-13-044968-7.

[82] **Web Application Description Language (WADL)** [Online] / autor Hadley Marc J. e Inc. Sun Microsystems. - Novembro de 2006. - Setembro de 2008. - https://wadl.dev.java.net/wadl20061109.pdf.

[83] **RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision** [Online] / autor Pautasso Cesare, Zimmermann Olaf e Leymann Frank. - Abril de 2008. - Setembro de 2008. - http://www.jopera.org/files/www2008-restws-pautasso-zimmermann-leymann.pdf.

[84] **RFC3265: Session Initiation Protocol (SIP)-Specific Event Notification** [Online] / autor Roach A. B. e dynamicsoft // ietf.org. - Junho de 2002. - Maio de 2008. - http://www.ietf.org/rfc/rfc3265.txt.

[85] **RFC3903: Session Initiation Protocol (SIP) Extension for Event State Publication** [Online] / autor A. Niemi Ed. e Nokia // ietf.org. - Outubro de 2004. - Maio de 2008. - http://www.ietf.org/rfc/rfc3903.txt.

[86] **RFC3515: The Session Initiation Protocol (SIP) Refer Method** [Online] / autor Sparks R. e dynamicsoft // ietf.org. - Abril de 2003. - Maio de 2008. - http://www.ietf.org/rfc/rfc3515.txt.

[87] **RFC3311: The Session Initiation Protocol (SIP) UPDATE Method** [Online] / autor Rosenberg J. e dynamicsoft // ietf.org. - Setembro de 2002. - Maio de 2008. - http://www.ietf.org/rfc/rfc3311.txt.

[88] **RFC3329: Security Mechanism Agreement for the Session Initiation Protocol (SIP)** [Online] / autor Arkko J. [et al.] // ietf.org. - Janeiro de 2003. - Maio de 2008. - http://www.ietf.org/rfc/rfc3329.txt.

[89] **RFC3310: Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)** [Online] / autor Niemi A. [et al.] // ietf.org. - Setember de 2002. - Maio de 2008. - http://www.ietf.org/rfc/rfc3310.txt.

[90] **RFC3312: Integration of Resource Management and Session Initiation Protocol (SIP)** [Online] / autor G. Camarillo Ed. [et al.] // ietf.org. - Outubro de 2002. - Maio de 2008. - http://www.ietf.org/rfc/rfc3312.txt.

[91] **RFC4032: Update to the Session Initiation Protocol (SIP) Preconditions Framework** [Online] / autor Camarillo G. [et al.] // ietf.org. - Março de 2005. - Maio de 2008. - http://www.ietf.org/rfc/rfc4032.txt.

[92] **RFC3313: Private Session Initiation Protocol (SIP) Extensions for Media Authorization** [Online] / autor W. Marshall Ed. e AT&T // ietf.org. - Janeiro de 2003. - Maio de 2008. - http://www.ietf.org/rfc/rfc3313.txt.

[93] **RFC3320: Signaling Compression (SigComp)** [Online] / autor Price R. [et al.] // ietf.org. - Janeiro de 2003. - Maio de 2008. - http://www.ietf.org/rfc/rfc3320.txt.

[94] **RFC3486: Compressing the Session Initiation Protocol (SIP)** [Online] / autor Camarillo G. e Ericsson // ietf.org. - Fevereiro de 2003. - Maio de 2008. - http://www.ietf.org/rfc/rfc3486.txt.

[95] **RFC3327: Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts** [Online] / autor Willis D. [et al.] // ietf.org. - Dezembro de 2002. - Maio de 2008. - http://www.ietf.org/rfc/rfc3327.txt.

[96] **RFC3608: Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration** [Online] / autor Willis D. [et al.] // ietf.org. - Outubro de 2003. - Maio de 2008. - http://www.ietf.org/rfc/rfc3608.txt.

[97] **RFC4566: SDP: Session Description Protocol** [Online] / autor Handley M. [et al.] // ietf.org. - Julho de 2006. - Maio de 2008. - http://www.ietf.org/rfc/rfc4566.txt.

[98] **RFC4346: The Transport Layer Security (TLS) Protocol Version 1.1** [Online] / autor Dierks T. [et al.] // ietf.org. - Abril de 2006. - Maio de 2008. - http://www.ietf.org/rfc/rfc4346.txt.

[99] **RFC4301: Security Architecture for the Internet Protocol** [Online] / autor Kent S., Seo K. e Technologies BBN // ietf.org. - Dezembro de 2005. - Maio de 2008. - http://www.ietf.org/rfc/rfc4301.txt.

[100] **RFC3852: Cryptographic Message Syntax (CMS)** [Online] / autor Housley R. e Security Vigil // ietf.org. - Julho de 2004. - Maio de 2008. - http://www.ietf.org/rfc/rfc3852.txt.

[101] **RFC4005: Diameter Network Access Server Application** [Online] / autor Calhoun P. [et al.] // ietf.org. - Agosto de 2005. - Junho de 2008. - http://www.ietf.org/rfc/rfc4005.txt.

[102] **RFC3588: Diameter Base Protocol** [Online] / autor Calhoun P. [et al.] // ietf.org. - Setembro de 2003. - Junho de 2008. - http://www.ietf.org/rfc/rfc3588.txt.

[103] **RFC2205: Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification** [Online] / autor Braden R. [et al.] // ietf.org. - Setembro de 1997. - Junho de 2008. - http://www.ietf.org/rfc/rfc2205.txt.

[104] **RFC3550: RTP: A Transport Protocol for Real-Time Applications** [Online] / autor Schulzrinne H. [et al.] // ietf.org. - Julho de 2003. - Junho de 2008. - http://www.ietf.org/rfc/rfc3550.txt.

[105] **RFC3262: Reliability of Provisional Responses in the Session Initiation Protocol (SIP)** [Online] / autor Rosenberg J. e Schulzrinne H. // ietf.org. - Junho de 2002. - Maio de 2008. - http://www.ietf.org/rfc/rfc3262.txt.

[106] **AAA and Network Security for Mobile Access: Radius, Diameter, EAP, PKI and IP Mobility** [Livro] / autor Nakhjiri Madjid e Nakhjiri Mahsa. - [s.l.] : John Wiley & Sons, Ltd, 2005. - ISBN:0-470-01194-7.

[107] **OMA Service Environment: Approved Version 1.0.4 – 01 Feb 2007** [Online] / autor OMA. - Julho de 2008. - http://www.openmobilealliance.org/.