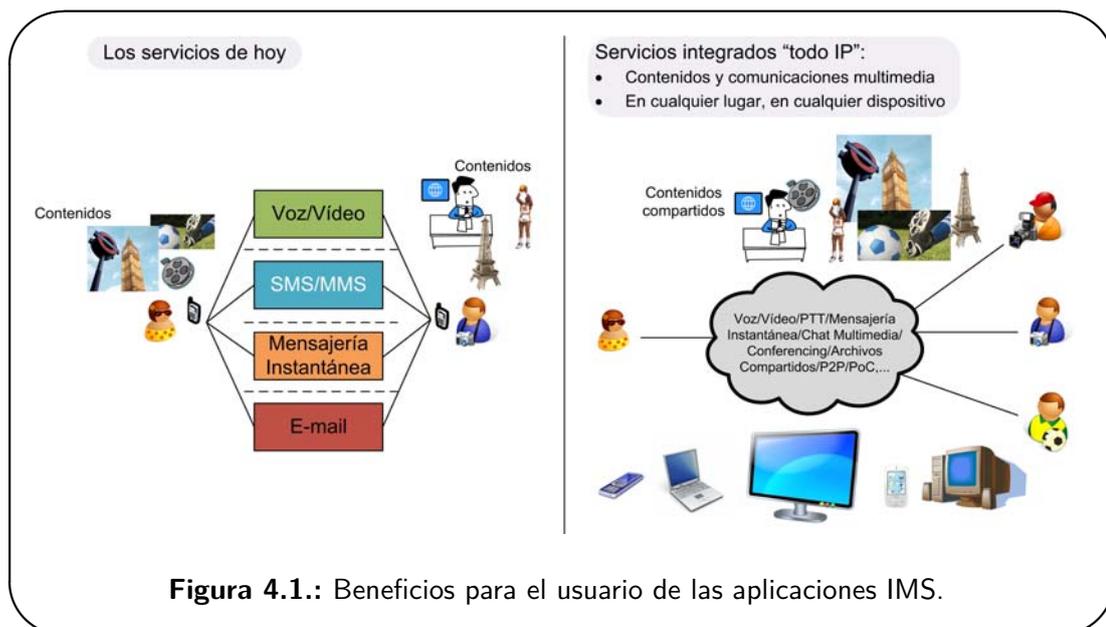


4. CREACIÓN DE UN SERVICIO IMS CON EL SDS

En este capítulo se abordarán las diferencias más significativas entre el estándar *IP Multimedia Subsystem* (IMS) y su entorno de desarrollo más completo, el *Service Development Studio* (SDS). Para ello, el estudio se centrará en el núcleo de la filosofía de IMS y de la Cuarta Generación (4G): los servicios. En un primer lugar se estudiará el proceso de creación de un servicio desde la perspectiva del estándar del *Third Generation Partnership Project* (3GPP) para más tarde acometer una comparación en base al análisis de la creación de un servicio desde la perspectiva del *Service Development Studio* (SDS).



4.1 Visión general de la creación de un servicio IMS

El *IP Multimedia Subsystem* (IMS) permite la introducción y libre integración de los servicios que hoy en día se disponen en una "torre" de diferentes redes independientes entre sí. En la figura 4.1 queda ilustrado esta transición, desde el concepto de n servicios- n redes a la filosofía de n servicios-una red. IMS facilita la integración y la convergencia en el acceso (soporte para múltiples dispositivos y múltiples redes de acceso), basándose en capacidades y servicios estandarizados

4.1. VISIÓN GENERAL DE LA CREACIÓN DE UN SERVICIO IMS

como son la telefonía IP multimedia, la mensajería IMS, *Push-to-Talk* (ya bautizado en castellano como “pulsar y hablar”), presencia, gestión de grupos, etc. con nuevos y diferenciados servicios tales como aplicaciones para la productividad empresarial, aplicaciones para el consumidor, juegos multijugador, contenidos y aplicaciones multimedia, y una infinidad de posibilidades que puedan surgir a partir de una buena idea.

Los servicios de comunicaciones de IMS, o *IMS Communication Services* (CoSe) (en adelante CoSe), proporcionan el soporte de red para aplicaciones universales destinadas al gran público; así como permite la integración de aplicaciones diferenciadas de nuevos operadores a través de capacidades bien definidas entre las cuales se incluyen la gestión de medios, principios de encaminamiento, y servicios suplementarios.

Estas capacidades fundamentales son exactamente idénticas en cualquier red o dispositivo IMS y por tanto están optimizadas en rendimiento para su despliegue masivo en los mercados. En consecuencia, los servicios CoSe son los cimientos naturales para la interoperabilidad global. Asimismo, las capacidades CoSe están expuestas a través de APIs en cada extremo de la comunicación para las comunidades de desarrolladores tanto para dispositivos como servidores de aplicación (ASs). Los servicios CoSe están basados en los estándares actuales y se actualizarán a medida que avanza el proceso de estandarización.

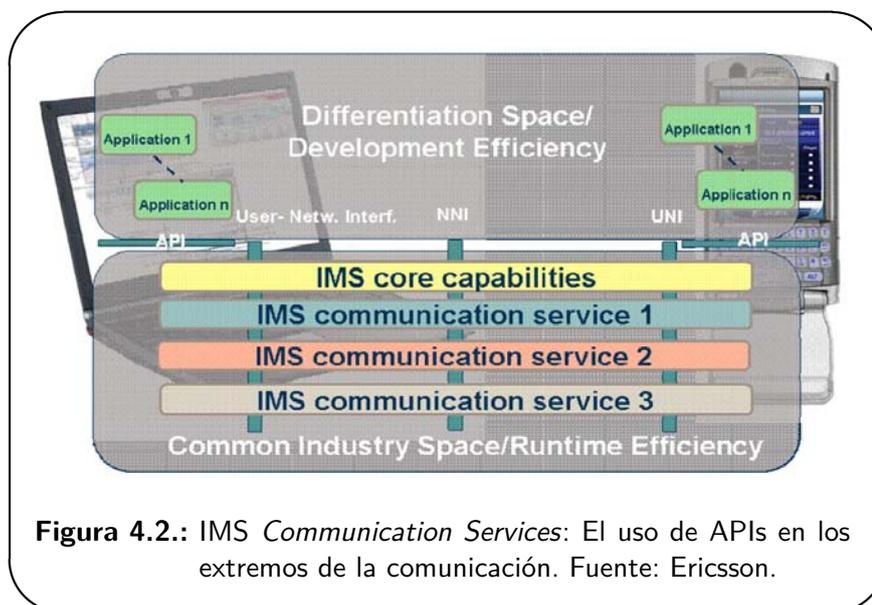


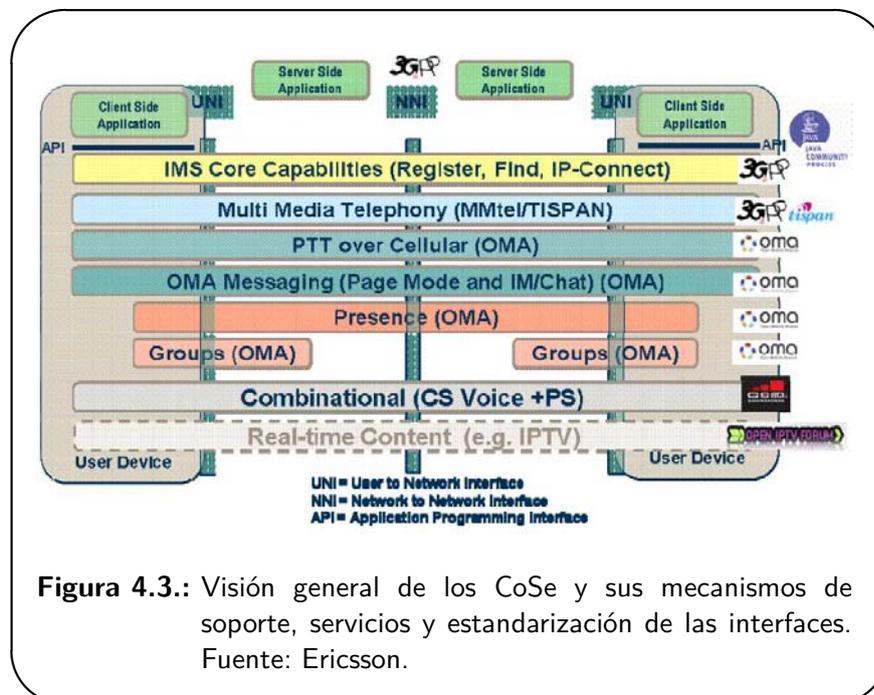
Figura 4.2.: *IMS Communication Services*: El uso de APIs en los extremos de la comunicación. Fuente: Ericsson.

Los CoSe de IMS se extienden cruzando las interfaces denominadas *Network to Network Interface* (NNI) y *User to Network Interface* (UNI). Los servicios multimedia terminan en el dispositivo o en los servidores en la frontera de una red IMS. La red IP actúa como un túnel versátil y estandarizado para los servicios multimedia. Las aplicaciones pueden conectarse a través de APIs tanto en el dispositivo como en la red (en los servidores). Para los operadores, ésta es una forma natural para establecer acuerdos de interconexión sobre “túneles” específicos,

así como para conectarse a Internet y a la industria multimedia.

En la creación de un servicio, la industria de las aplicaciones actuará principalmente en los extremos de la red y sobre UNIs. La combinación de las capacidades CoSe junto aplicaciones únicas y contenidos multimedia crea experiencias de usuario ricas y atractivas. De aquí que la mayoría de las aplicaciones IMS se provean en los extremos de IMS: dispositivos, servidores y *sites* de servicios. De esta manera se conectan de una manera natural las industrias de los medios y de los contenidos.

La industria se encuentra actualmente estandarizando tres servicios de comunicaciones IMS y dos mecanismos de soporte (ver figura 4.3). El *Java Community Process* (JCP) se encuentra de la misma manera normalizando APIs para el cliente/dispositivo, como las APIs de servicios IMS (JSR-281) y las APIs de *enablers* de comunicación IMS (*IMS Communication Enablers* (ICE)) (JSR-325)



- **MMTel CoSe**: se basa en la aplicación de telefonía multimedia (*MultiMedia Telephony*, MMTel), estandarizada de manera común entre el *Telecoms and Internet converged Services and Protocols for Advanced Networks* (TISpan) y el *Third Generation Partnership Project* (3GPP). Está optimizado para el suministro extremo a extremo de datos multimedia en tiempo real entre dos partes conectadas. Además, contiene el control *duplex* en tiempo real de los medios y de los servicios suplementarios definidos en el estándar.
- **PoC CoSe**: basado en el servicio *Push-To-Talk over Cellular* (PoC) estandarizado por la *Open Mobile Alliance* (OMA). El objetivo principal de este servicio es la comunicación en grupos y por ello propone una solución con datos multimedia *half-duplex* en tiempo real, pero con control de “suelo” para la transmisión continua. Cada participante en la conversación está conectado

4.1. VISIÓN GENERAL DE LA CREACIÓN DE UN SERVICIO IMS

a un gestor de grupos central que administra el grupo y la distribución de medios.

- **Mensajería IMS CoSe:** Este servicio está basado en el estándar de mensajería de OMA, el modo de búsqueda (*page mode*) y la mensajería instantánea/chat (IM/Chat). En este servicio la principal misión es la de garantizar la recepción de los mensajes, que ha de ser inmediata si el usuario destino está disponible.
- **Presencia y Gestión de Grupos, *Presence and Group Management (PGM)*:** también de OMA, son unos **mecanismos de soporte** para proporcionar información de disponibilidad y gestión de grupos de usuario.
- **Servicios Combinados (Combinational Services),** para la combinación de voz y vídeo sobre conmutación de circuitos (*Circuit Switched (CS)*) y aplicaciones IP móviles *Peer-to-Peer (P2P)*, tal y como promueve la *GSM Association (GSMA)*, y empieza a estandarizar el 3GPP.
- **Servicios IPTV:** Gestionados por el Open IPTV Forum, se está trabajando hacia una especificación *e-2-e* que permita a cualquier usuario el acceso a servicios *Internet Protocol Television (IPTV)* personalizados y enriquecidos, y desde cualquier dispositivo; siguiendo las especificaciones del Open IPTV Forum.

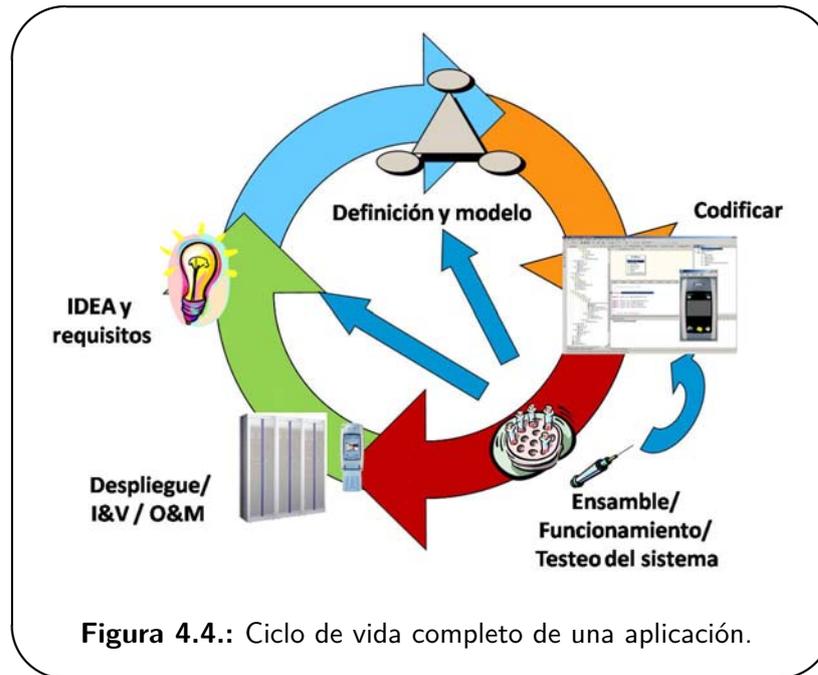
El éxito de IMS, como viene siendo tradicional en las Telecomunicaciones, dependerá de la disponibilidad de dispositivos debidamente adaptados. El principal escollo para el despliegue de nuevas tecnologías es normalmente la falta de disponibilidad de terminales móviles que las soporten.

El 3GPP, como organismo de estandarización, no plantea soluciones a problemas comerciales de la industria. Ericsson, la compañía que ha desarrollado el entorno de desarrollo de aplicaciones IMS estudiado en este proyecto, tomó varias medidas clave para liderar el crecimiento de IMS y sus servicios:

- Liderar la estandarización de la API en Java para terminales (JSR-281 y JSR-325) para que todos los terminales Java futuros posibilitem IMS.
- Ofrecer un *framework* para las APIs IMS (basadas en la pre-JSR-281/325) como una *IMS Client Platform (ICP) descargable* para terminales basados en los sistemas operativos Symbian y Windows — extendiendo gradualmente el soporte para otros SOs como Windows Mobile, Linux (Linux, Linux Mobile, Android,...), etc.
- Proporcionar capacidades IMS a aquellos terminales actuales con soporte Java pero sin soporte SIP: la *IMS JME Client Utility (IJCU)*.
- Integrar la funcionalidad de la API IMS a los terminales con plataformas Ericsson.

Además, Ericsson colabora de manera muy significativa en la especificación de la API Java de servicios IMS (JSR-281) y la API de *enablers* de comunicaciones (JSR-325), dentro del *Java Community Process (JCP)*.

Todas estas razones son por lo tanto suficientes para estudiar la herramienta de



desarrollo SDS en profusión, tal y como se hizo en el capítulo anterior. Sin embargo, se analizará en la siguiente sección la peculiaridad existente en la creación de nuevos servicios.

4.2 Creación de un servicio con el SDS, paso a paso

Los proveedores de servicios, así como empresas independientes de desarrollo software pueden hacer uso del SDS para explorar la viabilidad y aspectos de aceptación del usuario final acerca de los nuevos servicios IMS. Lo primero que hay que tener claro es que las aplicaciones que se desarrollen y testen con el SDS pueden desplegarse en un entorno IMS real y comercial, incluyendo dispositivos y servidores.

El proceso de desarrollo de servicios no consiste solo en la creación de estos, no se centra solo en la programación de una aplicación determinada en un entorno de desarrollo concreto. El proceso de desarrollo de servicios consiste en la gestión de todo el ciclo de vida de las aplicaciones, desde la idea hasta su despliegue y puesta en funcionamiento. La figura 4.4 ilustra este proceso [29].

4.2.1 Primera fase: programar, testar, y simular extremo a extremo en PCs con emuladores IMS SDS

La figura 4.5 ilustra una visión de alto nivel de la primera fase de la creación de un servicio: el trabajo con el entorno de desarrollo. Sobre un PC con una instalación del SDS, se procedería al desarrollo software de la aplicación que, en caso de éxito,

4.2. CREACIÓN DE UN SERVICIO CON EL SDS, PASO A PASO

proporcionaría un servicio en la red IMS real.

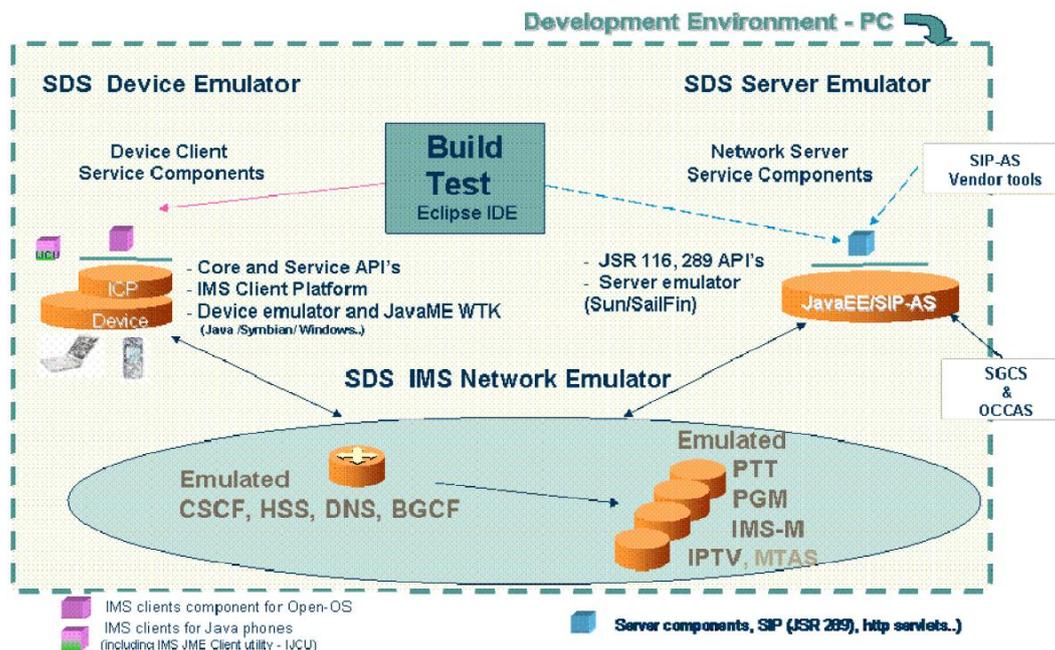


Figura 4.5.: Primera fase: programar, testar, y simular extremo a extremo en PCs con simuladores de IMS SDS. Fuente: Ericsson.

Una aplicación en IMS puede ser cliente a cliente o cliente servidor. En cualquier caso será necesario como mínimo cubrir una de las dos partes siguientes:

1. Lado servidor

- a) Crear un proyecto SIP/Web dinámico
- b) Crear un SIP *servlet*
- c) Codificar
- d) “Provisionar” el CSCF
- e) Establecer el servidor de entorno de ejecución por defecto (*Default Runtime Server*)
- f) Desplegar el proyecto sobre el servidor de aplicaciones SailFin
- g) Testar la aplicación
- h) Realizar baterías de pruebas con el ATF
- i) Depuración y subsanación de errores
- j) Ajustes necesarios para pasar a la siguiente fase de creación de un servicio

2. Lado cliente

- a) Crear un proyecto de ICP
- b) Añadir las capacidades ICP necesarias

- c) Codificar
- d) Instalar el cliente en Windows
- e) Lanzar el entorno de ejecución
- f) Configurar el perfil de usuario en la ICP
- g) Reiniciar la ICP
- h) Desplegar y ejecutar la aplicación
- i) Depuración y subsanación de errores
- j) Ajustes necesarios para pasar a la siguiente fase de creación de un servicio

En esta primera fase donde el papel importante lo juega el simulador del núcleo de IMS de Ericsson existe otra variante: emplear dispositivos reales que se conectan vía la red móvil de paquetes a los PCs con el SDS donde se alojen los servidores de aplicación. Esta variante, que queda ilustrada con la figura 4.6 precisa de unos pasos de configuración en los dispositivos móviles que se describen en la sección 4.2.2, con la salvedad de que habrá que configurar los PCs con el SDS dotándolas de una dirección IP pública para ser accedidos en el core IP desde los *Gateway GPRS Support Node* (GGSN) del operador móvil que da acceso IP en movilidad.

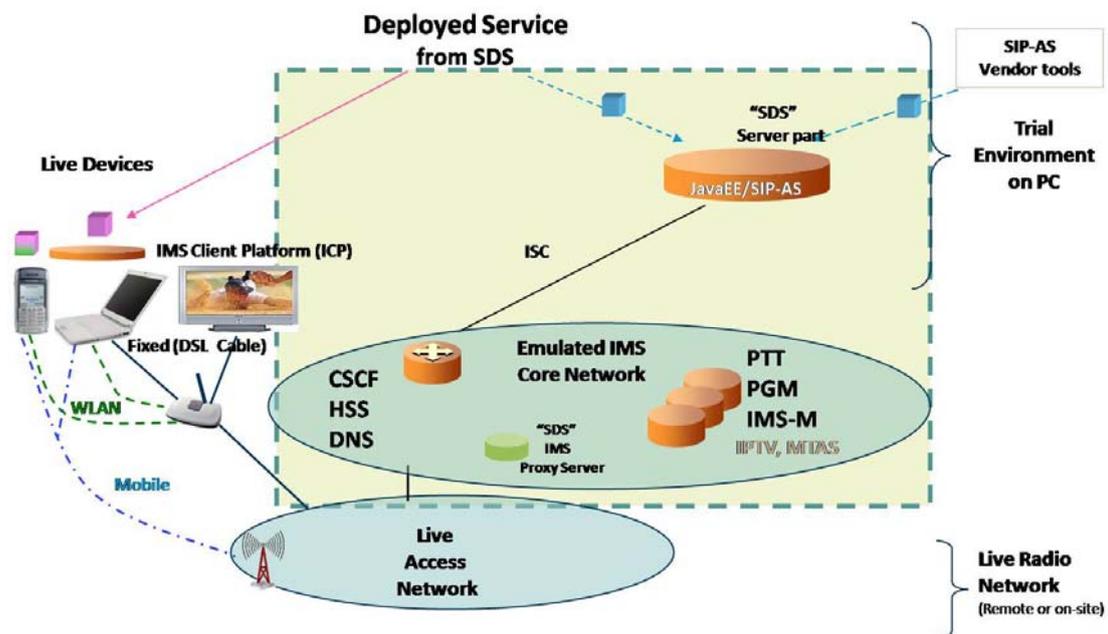


Figura 4.6.: Primera fase (variante con terminales reales): programar, testar, y simular extremo a extremo en PCs con simuladores de IMS SDS pero haciendo uso de dispositivos y redes de acceso reales para acceder al núcleo IMS simulado. Fuente: Ericsson.

4.2. CREACIÓN DE UN SERVICIO CON EL SDS, PASO A PASO

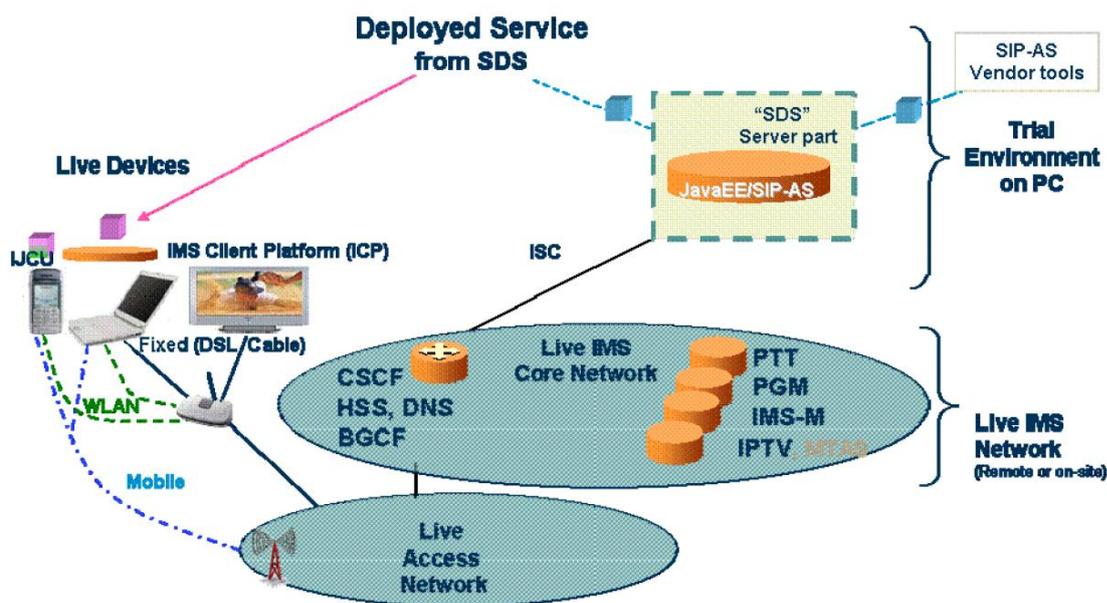


Figura 4.7.: Segunda fase: test y pruebas de usuario extremo a extremo con dispositivos y redes reales. Fuente: Ericsson.

4.2.2 Segunda fase: test y pruebas de usuario extremo a extremo con dispositivos y redes reales

La siguiente fase consiste en el testeo y pruebas extremo a extremo alojando los servicios en los servidores de aplicación integrados en el IDE SDS/Eclipse pero accediendo a ellos a través dispositivos reales y una red con funciones IMS reales implementadas en equipos dedicados.

En este proyecto se han llevado a cabo pruebas sobre terminales Sony Ericsson reales, en concreto los modelos M600 (figura 4.8) y P1i (figura 4.9) ambos, obviamente con sistema operativo Symbian y plataforma UIQ. Los pasos previos para testar los servicios son:

1. Instalar la *IMS Client Platform (ICP)* en los terminales a través de su correspondiente *suite* de comunicación con el PC.
2. Instalar la aplicación cliente del servicio sobre la ICP tal y como se vio en el capítulo 3 generando el fichero .sis y descargándolo en el terminal.
3. Configurar en los terminales la red de acceso a través de la cual se llevará a cabo la conexión al entorno real de IMS.

Una vez realizados los pasos previos es preciso disponer de un entorno real de IMS para llevar a cabo la señalización de los servicios que se quieren testar. En este proyecto se han analizado las dos alternativas posibles: disponer de un entorno real local o remoto. En este sentido, en la siguiente sección 4.2.3 y en el apéndice B se trata con profusión la solución de Ericsson en materia de *core* real y local de IMS. Por ello, esta sección se dedica a la solución de la compañía finlandesa **Octopus**



Figura 4.8.: Terminal de pruebas Sony Ericsson M600 utilizado en el proyecto.



Figura 4.9.: Terminal de pruebas Sony Ericsson P1i utilizado en el proyecto.

4.2. CREACIÓN DE UN SERVICIO CON EL SDS, PASO A PASO

Network, que proporciona un entorno real remoto para pruebas y ensayos de aplicaciones y servicios basados en IMS.

■ La solución de testeo remoto de Octopus Network

Esta plataforma pone a disposición de sus clientes conexiones a sus laboratorios para el testeo en equipos reales y certificados de aplicaciones y servicios IMS. Octopus juega con la baza de disminuir el *Time To Market* (TTM) y garantizar la calidad avalada por

- Un desarrollo y testeo eficiente en costes, ya que esta solución es más económica que la de adquirir un entorno real.
- Proporciona una perspectiva más realista que el entorno simulado.
- Pone a disposición de sus clientes un gran elenco de expertos en la red IMS y en el desarrollo de servicio.
- Octopus agrupa a más de 60 compañías (*partners*), investigadores en el centro finlandés de desarrollo en las Comunicaciones Móviles en Oulu, y clientes de reconocida experiencia, lo que aumenta su prestigio y publicidad. Los participantes más activos son Nokia, DNA, TeliaSonera, OAMK, Innovación de Oulu y la Ciudad de Oulu. Los clientes de Octopus han creado decenas de soluciones y servicios móviles bajo el entorno de Octopus [30].
- Proporciona tutorías y *training* por lo que es una solución atractiva para PyMEs.

Por otro lado, Octopus proporciona las siguientes soluciones:

- Testeo interoperable en un entorno real de un operador con despliegue IMS a través de acceso móvil e inalámbrico.
- Plataforma de pruebas en varias tecnologías.
- Inicio de sesión, autenticación y verificación
- Interoperabilidad y *handover* entre diferentes redes.
- Consultoría tecnológica.
- Testeo de aplicaciones:
 - Trazas del tráfico y distintos registros (*logs*).
 - *Enablers* tecnológicos avanzados: Presencia, PTT,...
 - Pruebas de “usabilidad”, grupos de usuarios reales, metodología *Living Lab*.
 - Multiacceso móvil e inalámbrico (3G, HSPA, WLAN, mWiMAX).

Arquitectura de la solución de Octopus Octopus es una plataforma avanzada de movilidad, donde se investigan, desarrollan y testan tecnologías móviles inalámbricas, así como aplicaciones y servicios. El núcleo del servicio que ofrece Octopus es un entorno cerrado de un operador — concebido para el desarrollo y

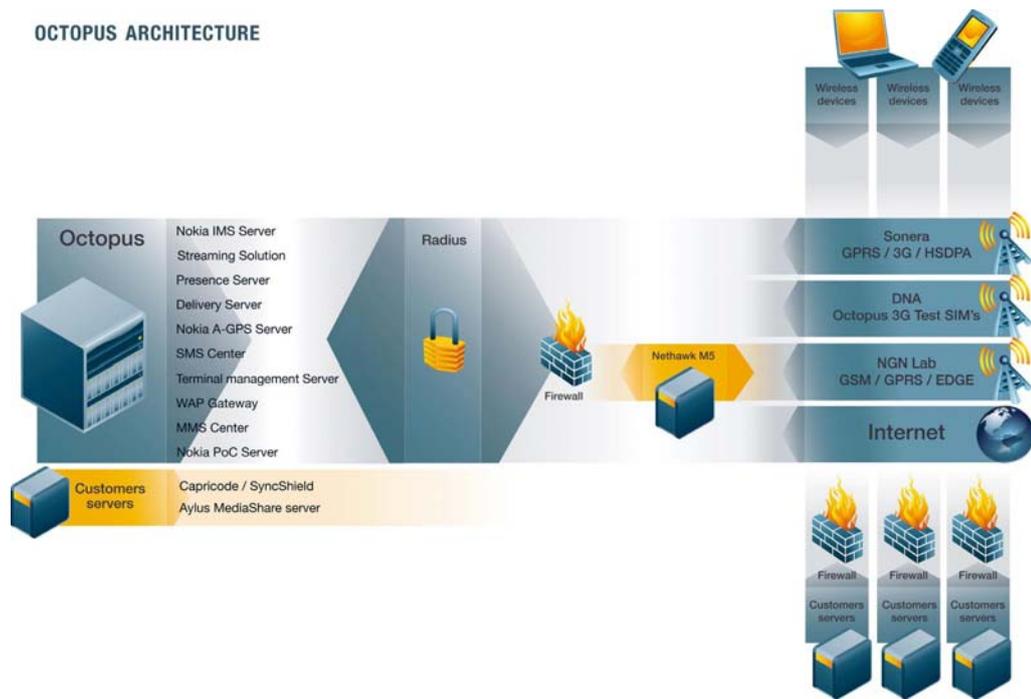


Figura 4.10.: Arquitectura de la solución de Octopus para el testeo remoto de aplicaciones y servicios IMS. Fuente: Octopus Network.

testeo de aplicaciones móviles — que funciona con un acceso internacional desde distintas redes (GPRS, 3G, HSPA, B3G, WLAN). La plataforma de ensayos, testeo y pruebas proporciona a los clientes de esta solución, como se observa en la figura 4.10, un núcleo de red de IMS completo, accesible desde varias redes distintas. En este sentido, el cliente puede elegir el medio de acceder a la plataforma, según las necesidades de su aplicación o la planificación de sus proyectos. Un cliente podría optar, por ejemplo, por hacer uso de WLAN, o por otro lado adquirir una tarjeta SIM de TeliaSonera, con un contrato específico de 3G con la operadora para acceder al *core* IMS de Octopus a través de *roaming* 3G.

Asimismo, el entorno es completamente interoperable, y garantiza la seguridad por medio de *cortafuegos* y autenticación a través de servidores RADIUS (observar la figura 4.10) y túneles *Virtual Private Network* (VPN) con IPsec desde los laboratorios del cliente hasta los de Octopus.

4.2.3 Tercera fase: despliegue en servidores comerciales

Una vez codificado el servicio IMS, testado en un *core* real y abordada la depuración sólo quedaría desplegar comercialmente el servicio. En este último punto, la última capa de IMS, la de aplicación, se transportaría desde el PC donde ha sido desarrollada y testada hasta servidores reales comerciales como el *Oracle Communications Converged Application Server* (OCCAS) (BEA Weblogic Server), con una mayor capacidad y escalabilidad, así como una mayor versatilidad de

4.2. CREACIÓN DE UN SERVICIO CON EL SDS, PASO A PASO

configuración (figura 4.11).

En este apartado, y a caballo entre la segunda fase de creación de un servicio y la presente; se tiene de nuevo una solución de Ericsson que integra en una sola infraestructura *hardware* acceso “agnóstico” a IMS, núcleo IMS, y servidores de aplicación comerciales IMS. La inclusión del **IMS Common System** de Ericsson, así como su estudio pormenorizado en el apéndice B se debe a la falta de despliegue de IMS que hoy en día padecen la mayoría de los *cores* de red de los operadores. Al no existir una puerta abierta para que desarrolladores y otros terceros desplieguen comercialmente nuevas aplicaciones y servicios de IMS, algunos laboratorios y empresas de desarrollo *software* se decantan por adquirir un entorno real local para el testeo y puesta a punto de nuevas aplicaciones y servicios, que quedarían totalmente preparadas para su despliegue en tiempo mínimo cuando los núcleos de los operadores soporten la arquitectura de IMS. O, incluso, lanzar estos servicios sobre redes actuales beneficiándose de protocolos de IMS (SIP, DIAMETER,...) pero soportando deficiencias de la red actual (*best-effort* frente a QoS, ausencia de integración de servicios, pérdida de control de la cadena de valor por parte del operador, etc.).

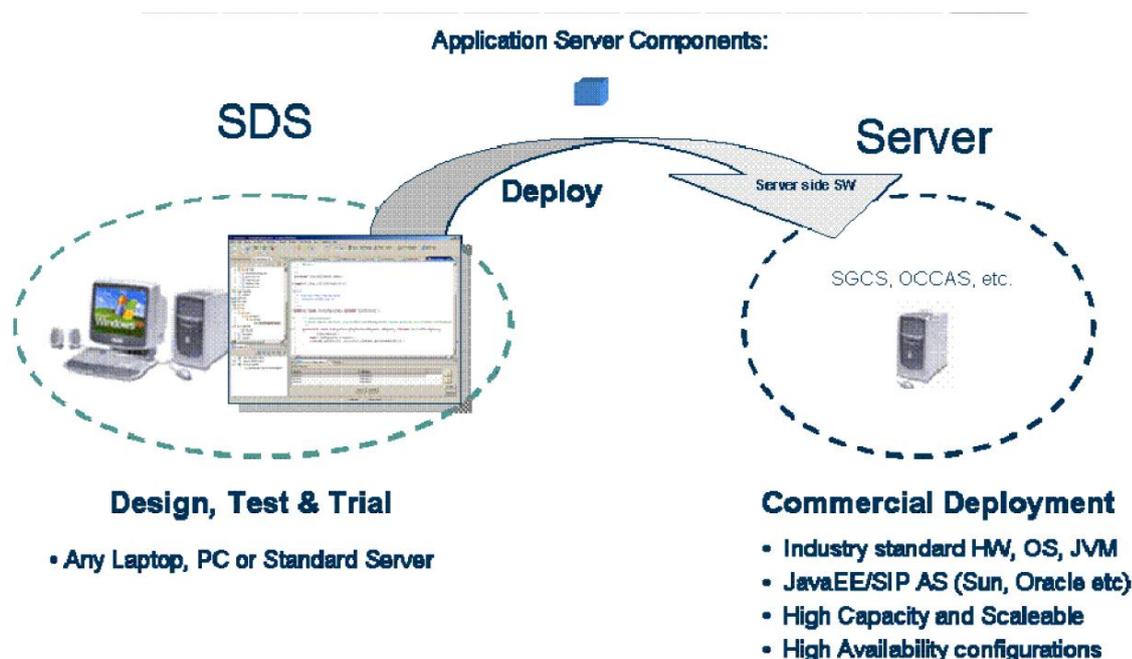


Figura 4.11.: Tercera fase: despliegue en servidores comerciales. Fuente: Ericsson.

■ Un entorno real local: El IMS *Laboratory System* de Ericsson

El IMS *Laboratory System* de Ericsson es la solución personalizada del IMS **Common System** de la compañía sueca. Éste, a su vez, es parte de la unidad de negocio *Ericsson IMS*, como bien es posible observar en la figura 4.12. En esta se ubica el entorno de desarrollo *Service Development Studio* (SDS) estudiado en el capítulo 3 como un producto de creación del nivel de aplicación dentro de *Ericsson*

IMS. En esta figura queda refrendado el papel de Ericsson en su compromiso con el IMS, puesto que hasta la fecha ningún otro fabricante ha desarrollado una solución tan completa ni compacta para IMS.

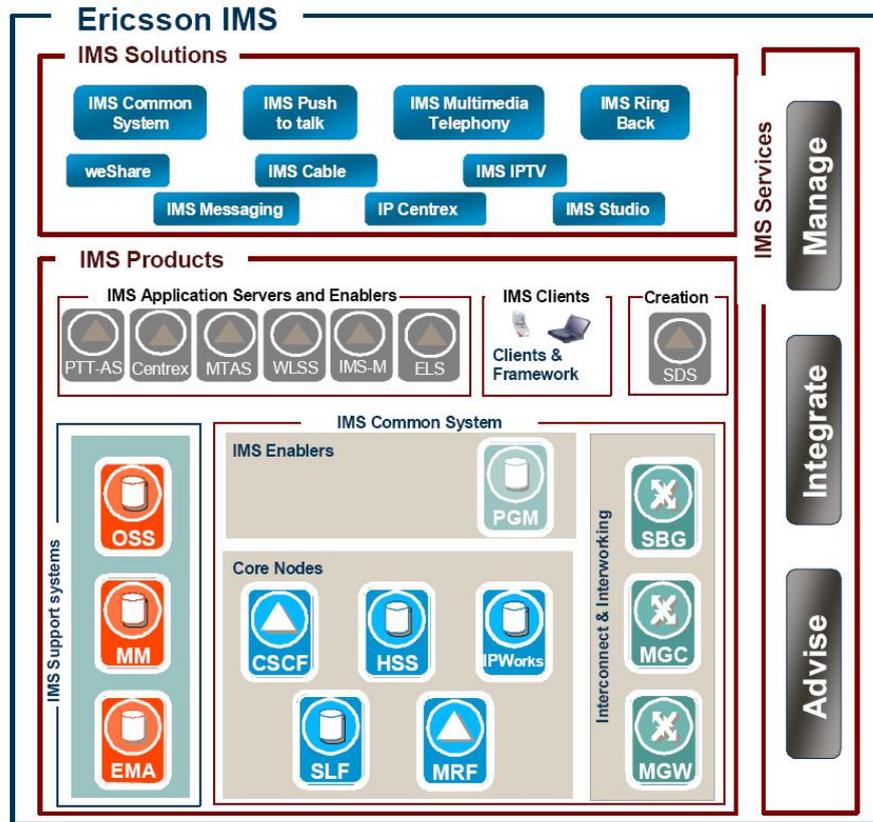


Figura 4.12.: Esquema de las soluciones comerciales del área **Ericsson IMS**, estructuradas en soluciones, productos y servicios. Fuente: Ericsson.

El entorno real de IMS, **IMS Laboratory System** persigue instalar un *core* de IMS que refleje la red de un operador comercial, de tal modo que se permita el acceso al núcleo de IMS como si el operador la tuviera desplegada. Con esto se persigue:

- Facilitar el testeo de nuevas aplicaciones IMS.
- Catalizar y acelerar al cliente que contrate el **IMS Laboratory System** el proceso de pruebas, demos y presentaciones de las aplicaciones IMS.

Asimismo, el **IMS Laboratory System** ha de incluir una integración con la red de conmutación de circuitos del operador, para asegurar la convergencia. Este hecho parte de la suposición de que el potencial cliente requiere una integración en el acceso, y por tanto este supuesto conduce a la provisión de pasarelas (*Media Gateway Control Function* (**MGCF**)/*Media Gate Way* (**MGW**)) en la arquitectura del **IMS Laboratory System**. Esta integración se completa con la adición de un módulo *Session Border Controller* (**SBC**) (Acme 5.1) para ensayos y pruebas con la red de acceso fijo.

4.2. CREACIÓN DE UN SERVICIO CON EL SDS, PASO A PASO

Todo el *hardware* se monta sobre el bastidor Ericsson BYB 501 [4.13](#) en tres compartimentos distintos:

- Armario para *Telecom Server Platform/Network Server Platform* (TSP/NSP) con un *rack* o rejilla de 400 mm de profundidad para montar el *hardware* de las plataformas TSP/NSP.
- Armario para *Integrated Site* (IS): de 400 mm de profundidad para montar los sub-*racks* de IS.
- Cabina de datos en el BYB 501 de 800 mm de profundidad de cara a montar los elementos de la red de SUN y HP, el Acme SBC así como la conectividad entre la plataforma y en el exterior con el módulo *switch/router*



Figura 4.13.: El bastidor BYB 501 que aloja los elementos del *core* real de Ericsson para testeos y pruebas.

Arquitectura del IMS *Laboratory System* de Ericsson En la figura [4.14](#) se ilustra la arquitectura del entorno real de Ericsson, con el núcleo IMS, los *enablers* y las pasarelas para movilidad y convergencia. El PLMN, realmente, puede ser cualquier operador móvil que opere en el país donde se lleve a cabo la instalación de este entorno real.

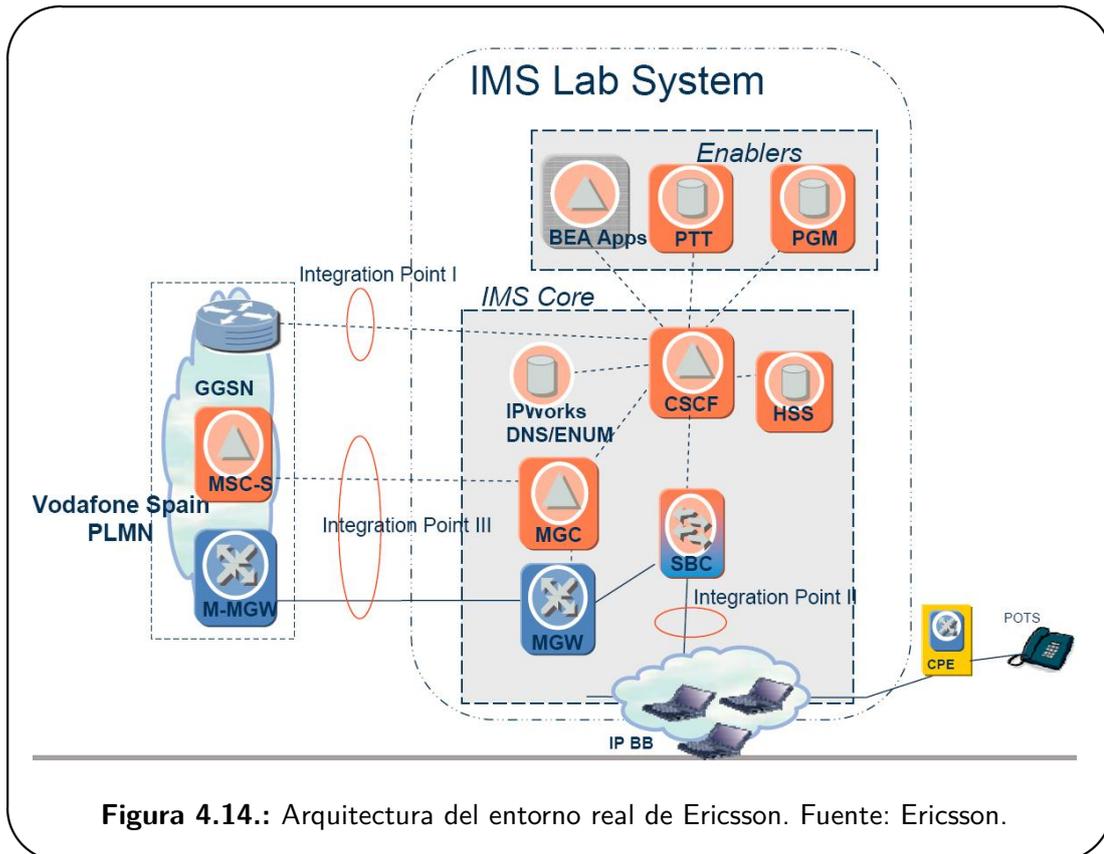


Figura 4.14.: Arquitectura del entorno real de Ericsson. Fuente: Ericsson.

Core

- Un CSCF y un HSS:
 - *hardware*: Un $\frac{1}{4}$ OPTI TSP
 - *software*: IMS Common System 4.1
 - La colocación del *core* IMS (HSS+CSCF) está propuesta de este modo bajo la suposición de una solución más económica y compacta.
 - Es preciso tener en cuenta que de este modo el *core* es una “caja negra” que limita la visibilidad y control de errores sobre algunas interfaces del núcleo, como podría ser el control sobre Diameter.
- Un DNS instalado en una máquina SUN NETRA 240, con *software* IPWorks 5.0.
- Un Redback SE400 con 60 puertos Fast-Ethernet y 2 puertos Gigabit Ethernet.
- Un armario para *Telecom Server Platform/Network Server Platform* (TSP/N-SP) .
- Un armario para datos.

Capa de aplicación

- Un *Presence and Group Management* (PGM)

4.2. CREACIÓN DE UN SERVICIO CON EL SDS, PASO A PASO

- *hardware*: Un $\frac{1}{4}$ OPTI-4 TSP: un armario TSP/NSP
- *software*: PGM 4.1
- Un *Oracle Communications Converged Application Server (OCCAS)* Oracle (BEA) Web Logic Sip Server para instalar las aplicaciones desarrolladas, por ejemplo con el *Service Development Studio (SDS)* de Ericsson.
 - *hardware*:
 - HP PROLIANT DL360 G4p , SCSI
 - Procesador Xeon 3,6 GHz
 - 2º Procesador Intel Xeon 3,6 GHz
 - *software*:
 - Linux Red Hat Enterprise ES 3.0
 - BEA WebLogic SIP Server 2.2
 - MySQL Databe 5.0.24a
- Un *Push-To-Talk (PTT)*
 - *hardware*: Un $\frac{1}{4}$ OPTI-4 TSP: un armario TSP/NSP
 - *software*: Servidor PTT 4.1

Puntos para la integración y convergencia de redes de acceso: IMS Multi-Access (IMA)

- Un punto para la **integración GPRS/UMTS**: interfaz Gi con el operador de telefonía y datos en movilidad¹.
- **Integración Internet/Acceso fijo**:
 - Un A-SBC (Net Session Director 5.1) de ACME.
 - La integración de los servicios entre el núcleo de red IMS y una red IP externa se lleva a cabo a través de un *Session Border Controller (SBC)*.
- **Integración PLMN SS7/TDM**:
 - *software* MGC 5.1 y MGW 1.2, ambos instalados en el mismo equipo
hardware: *Integrated Site (IS)* R1.2.
 - Un armario *Integrated Site (IS)*.
 - La integración con el dominio de conmutación de circuitos móvil (PLMN) en una red externa se lleva a cabo a través de un **IS**.

¹Se asume contrato de colaboración entre el cliente que instala la solución de Ericsson en materia de núcleo de red y el operador que proporciona la red de acceso a susodicho núcleo.

4.3 Diferencias entre el estándar y el SDS: Limitaciones y problemas

El *Service Development Studio* (SDS) es un entorno de desarrollo para aplicaciones y servicios de IMS que simula el núcleo de red de IMS para que esta simulación baste a la hora de asegurar el cumplimiento (al menos en unos mínimos) con el estándar.

Sin embargo, la complejidad de las especificaciones de IMS, hacen casi imposible un soporte de las capacidades IMS total del lado del entorno *software* de desarrollo, al menos en las primeras versiones de este *Integrated Development Environment* (IDE).

Durante la elaboración de este proyecto, se confeccionó un cuaderno de notas sobre el que se iban anotando cualquier limitación y problema que se hallara con la herramienta de Ericsson. Las principales conclusiones fueron:

4.3.1 Limitaciones

- El **CSCF** está como un nodo único y sus tres funciones (P-CSCF, S-CSCF, I-CSCF) están limitadas por este hecho. Su configuración se limita prácticamente a dirección IP y puerto, no dejando opción a muchas otras de las funcionalidades que se describen en el estándar.
 - **Posibles mejoras:** Ampliación de los parámetros, desarrollo de una interfaz gráfica web para la configuración de estos parámetros en remoto.
- Ausencia de un nodo independiente para el **HSS** sobre el que implementar interfaz DIAMETER. El hecho de que el HSS aparezca como una mera GUI reflejando campos de una base de datos interna limita el despliegue real con el paso de mensajes sobre la interfaz Sh (mensajes DIAMETER para autenticación, descarga del perfil XML, etc.)
 - **Posibles mejoras:** Independencia del HSS con dirección IP y puerto propios y soporte para DIAMETER.
- Falta absoluta de control sobre interfaces DIAMETER y COPS: El SDS no soporta mensajes COPS sobre la interfaz Go para políticas de tráfico y QoS ni mensajes DIAMETER con el HSS o el *Policy Decision Function* (PDF).
 - **Posibles mejoras:** Al igual que con los *servlets* SIP, podría crearse una estructura con *wizards* y asistentes de código que implementaran el modelo cliente-servidor de DIAMETER y COPS con una interfaz sencilla, pero soportando elementos importantes como seguridad y QoS.
- Ausencia del MRFC/MRFP limitando los servicios que pudieran implementarse relativos a contestador, servicios originados en la red con inclusión de medios, etc.
 - **Posibles mejoras:** El SDS podría venir integrado con un reproductor/-codificador de medios con su correspondiente soporte en clases de Java y descripción SDP.
- A día de hoy, el SDS sólo soporta el SIP AS, no teniendo soporte para CAMEL

ni OSA.

- **Posibles mejoras:** Aunque no sea uno de los objetivos del SDS, podrían integrarse en el mismo IDE el IM-SSF y el OSA-SCS de terceros como Rhino (IM-SSF) o Parlay (OSA).
- Aparte de la implementación de los elementos del núcleo de red, el SDS no recoge plataformas adicionales definidas por el 3GPP como el *Voice Call Continuity* (VCC)² que es un elemento esencial en el modelo de convergencia fijo móvil.
 - **Posibles mejoras:** Contactos con empresas como Huawei [31], comprometidas con el desarrollo de VCC y su inclusión en IMS.

4.3.2 Principales problemas

- **Problemas con el emulador del cliente:** Mediante los distintos ensayos durante el desarrollo del proyecto se vieron serias limitaciones referentes a la inestabilidad del emulador, básicamente comprobando cómo una aplicación se ejecuta perfectamente en Windows pero no es posible hacer lo mismo emulador. Dos problemas concretos fueron:
 - En el SDK de Symbian/UIQ aparece que se pueden emular distintos terminales, entre los que se encuentra el M600i y el P990. El único terminal que permite configurar correctamente los parámetros de IMS Settings (permiten establecer la conexión con la red IMS) es el M600i.
 - No permite simular varios clientes en la misma máquina. Esta problemática se plantea porque en determinados servicios existen diversos posibles usuarios que pueden usarlo tomando diferentes roles, y para comprobar el correcto funcionamiento del servicio deben probarse dichos roles al mismo tiempo. Para el caso de dos clientes se deben conectar dos ordenadores con un cable de red cruzado y lanzar los elementos del *core* y ASs en uno de ellos; y en el caso de que se deseara añadir algún cliente más a la prueba se tendría que utilizar un *switch*.
- Estado actual de desarrollo de las **APIs**. La tecnología IMS ofrece una gran potencialidad que de momento no está al alcance de los desarrolladores que utilizan esta herramienta debido a que estas características no se encuentran implementadas en las APIs (en concreto las APIs de la *IMS Client Platform* (ICP)): APIs de QoS, APIs de DIAMETER, *streaming*, (en este sentido, la utilidad IPTV de la versión 4.1 FD1 del SDS no funciona), etc.
- Soporte de *IVideoPlayer*, *IVideoRecorder* para lanzar aplicación de reproduc-

²El *Voice Call Continuity* (VCC) (3GPP TS 23.206), en un futuro *Multimedia Session Continuity* (MMSC) y *IMS Service Continuity* (IMScont) (3GPP TS 23.206) consiste, *grosso modo* en un conjunto de especificaciones para describir cómo una llamada de voz (sesión multimedia a partir de la *release 8*) se mantiene mientras un teléfono móvil se mueve entre un dominio de conmutación de circuitos y un dominio de conmutación de paquetes. El ejemplo más claro es el de una llamada originada en GSM que se mantiene continua, y con la misma calidad, al entrar en una zona de cobertura WiFi.

ción de video en el terminal al tener un buffer de entrada de video en la versión 4.0 de la herramienta no estaban implementadas estas clases. En la versión 4.1 FD1 el paquete **com.ericsson.icp.multimedia** está bastante completo en lo referente a audio, pero sigue sin soportar vídeo. Como solución se ha propuesto utilizar otra aplicación programada sobre un tercer componente para manejar el flujo de video en el terminal. Es decir, se estaría tratando por un lado la parte de IMS, con su señalización, lógica de perfiles, etc, hasta que el buffer llega al terminal; y a partir de ese momento se utilizaría otro aplicación totalmente independiente de IMS para manejar dicho flujo y lanzar un reproductor apropiado (en este caso el mismo VLC que sirve para IPTV). De todas formas, por parte del soporte oficial de la herramienta se ofrece esta posibilidad como alternativa al problema y haciendo usos de características de IPTV, pero no como solución probada.

- El SDS no es **multiplataforma**. Además de su falta de compatibilidad con Linux, MAC y otras plataformas, el SDS es particularmente inestable con Windows Vista, por lo que hay que recurrir a herramientas de máquinas virtuales como VMWare o Virtual PC 2007 de Microsoft.
- Problemas con la instalación de aplicaciones en dispositivo real: Existe una opción en la herramienta para instalar aplicaciones en terminales reales (Install in Symbian Device). Esta opción recomienda el soporte de Ericsson no utilizarla porque no funciona correctamente. A la hora de instalar las aplicaciones se ha de utilizar el PC Suite del dispositivo, o hacerlo tratando al teléfono como una unidad externa, pero siempre al margen de la herramienta.
- La compatibilidad entre versiones se ha mejorado con la introducción de un traductor de versiones (*legacy applications*). Sin embargo, la tarea de conversión desde versiones 3.1 a posteriores es especialmente ardua, ya que del lado del cliente la ICP constaba de tres capas o **adapters**(*Platform, Service y Session*) y ahora cuenta con cuatro (*Platform, Profile, Service, Session*).

