

# Optimierung der Dienstentwicklung und -bereitstellung in UMTS/IMS-Mobilfunknetzen

Armin Lehmann, Ulrich Trick, Fachhochschule Frankfurt/M. - University of Applied Sciences, Kleiststraße 3, 60318 Frankfurt/M., E-Mail: lehmann@e-technik.org, trick@e-technik.org;  
Steffen Oehler, Detecon International GmbH, Oberkasseler Straße 2, 53227 Bonn,  
E-Mail: Steffen.Oehler@detecon.com

## Kurzfassung

Die heutigen Lösungen zur Bereitstellung und Entwicklung neuer Dienste, speziell von Mehrwertdiensten, in zukünftigen, auf dem NGN-Konzept basierenden Netzen werden eingehend beleuchtet. Hinsichtlich der Dienstentwicklung und -bereitstellung wird das IP Multimedia Subsystem (IMS) näher erläutert, da dieses die nötigen Schnittstellen und Basisdienste bereitstellt. Anhand eines Beispielszenarios wird aufgezeigt, welche der heutigen Lösungen sich für einen spezialisierten Dienst wie z.B. termingesteuerte Video-Konferenz in einem UMTS-Mobilfunknetz eignen.

## 1 Einleitung und Anforderungen

Die zukünftigen Netze, speziell auch die UMTS-Mobilfunknetze, erfordern eine detaillierte Analyse zur Bereitstellung und Entwicklung neuer Dienste. Hierbei eröffnen sich neue Fragestellungen betreffend der Anforderungen und Lösungen der Dienstplattformen für Provider. Welche heutigen Lösungen zur Dienstentwicklung und -bereitstellung in Next Generation Networks (NGN) sind relevant? Wie können diese Dienstplattformen beurteilt werden? Welche Kriterien sind anzulegen? Welche Möglichkeiten eröffnet das IP Multimedia Subsystem (IMS) in Bezug auf Dienstentwicklung und -bereitstellung? Diese Fragestellungen werden im Folgenden erläutert.

Durch die zukünftigen Netze und deren Dienstplattformen wird eine Bereitstellung von sogenannten Mehrwertdiensten leichter möglich. Ein Mehrwertdienst, engl. Value Added Service (VAS), ist ein spezieller Telekommunikationsdienst, der über die Funktionalität eines reinen Basisdienstes hinausgeht. Telefon- und Telefax-Dienst sind z.B. solche Basisdienste. Zu den Mehrwertdiensten zählen beispielsweise Click-to-Dial, hier wird eine Rufnummer aus dem Webbrowser oder E-Mail Client heraus gewählt, und Voice-Mail, wo bei einem Anruf in Abwesenheit dem Empfänger eine Sprachnachricht per E-Mail zugestellt wird. Diese Dienste erfordern Funktionen, die das Kernnetz nicht alleine erfüllen kann, daher werden weitere Netzelemente wie z.B. Application Server (AS) und Media Server (MS) benötigt. Die AS stellen die Dienste zur Verfügung und bedienen sich bei Bedarf für komplexere Dienste weiterer AS oder MS. Der Media Server stellt hierbei die Funktionalität zur

Verarbeitung der Nutzdaten z.B. bei Video-Konferenzen bereit.

Für die Dienstebereitstellung und -entwicklung auf den Application Servern existieren verschiedene Lösungen, die später noch erläutert werden. Alle diese Techniken haben gemeinsam, dass sie auf dem NGN-Konzept basieren. Eine standardisierte Umsetzung des NGN-Gedankens ist das NGN/IMS-Konzept. Hier lässt sich schon die Bedeutung des IMS für Dienste erkennen. Im folgenden Abschnitt werden die Gemeinsamkeiten, aber auch Differenzen zwischen NGN und NGN/IMS herausgestellt.

## 2 Übersicht zu NGN und NGN/IMS

Anhand der Bilder 1 und 2 werden die Gemeinsamkeiten und Unterschiede der Begriffe NGN und NGN/IMS herausgearbeitet. Bild 1 zeigt die NGN-Architektur in einer Strata-Struktur [1]. Darin werden die wesentlichen NGN-Subnetze und Netzelemente Schichten bzw. Strata zugewiesen. Der Begriff Schicht bzw. Layer ist vom OSI-Referenzmodell [2] (Layer 1 bis 7) her geläufig, wird hier aber allgemeiner im Sinne eines Stratum benutzt, wobei ein Stratum normalerweise zwei oder mehr OSI-Schichten (Layer) umfasst. Hier wird zwischen drei Strata unterschieden:

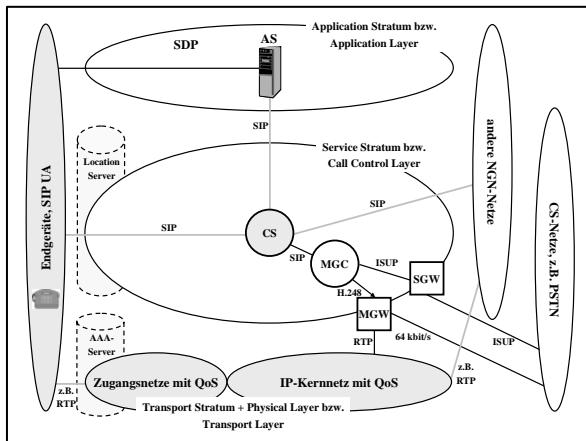
- dem Transport Stratum mit unterlegtem Physical Layer [1] bzw. dem Transport Layer [3] zur Modellierung der Zugangsnetze und des IP-Kernnetzes inkl. des Nutzdatentransports (z.B. RTP (Real-Time Transport Protocol)),
- dem Service Stratum [1] bzw. Call Control Layer [3] für die Funktionen und Netzelemente für die Session-Steuerung (CS (Call Server), MGC (Me-

dia Gateway Controller), z.B. mittels SIP (Session Initiation Protocol), ISUP (ISDN User Part) und H.248) sowie

- dem Application Stratum [1] bzw. Application Layer [3] für die Service Delivery Platform (SDP) mit den Application Servern zur Bereitstellung von Mehrwertdiensten.

Dies zeigt deutlich die im Hinblick auf Offenheit, Flexibilität und Skalierbarkeit vom NGN-Konzept forcierte Trennung des Nutzdatentransports von der Verbindungssteuerung sowie der Funktionen für die Basisdienste von denen für die Mehrwertdienste [4].

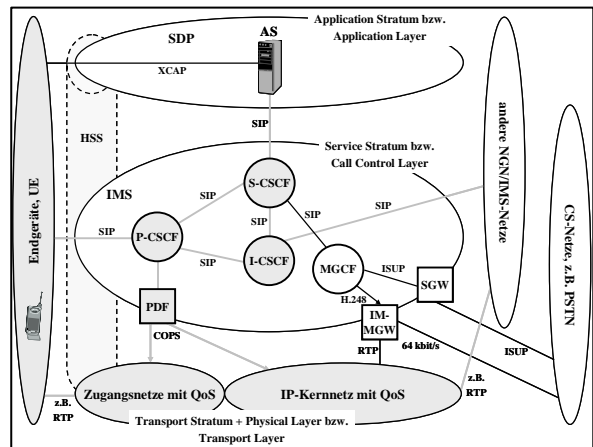
Bild 1 enthält im Hinblick auf das NGN-Kennzeichen „Unbeschränkter Nutzerzugang zu verschiedenen Netzen und Diensteanbietern“ mindestens zwei AAA-Server (Authentication, Authorization, Accounting), einen für den Nutzerzugang und einen für die Session-Nutzerdaten (Location Server). Der Zugriff auf die Service Delivery Platform (SDP) ist in Bild 1 vereinfachend als nicht beschränkt dargestellt. Alle bei einer rein SIP-basierten Kommunikation beteiligten Subnetze, Netzelemente und Schnittstellen sind in Bild 1 grau hinterlegt bzw. markiert, um einen einfachen Vergleich mit einem NGN auf IMS-Basis (NGN/IMS) zu ermöglichen [4].



**Bild 1 NGN-Architektur in einer Strata- bzw. Layer-Struktur**

Die mit Bild 1 für NGN eingeführte Strata- bzw. Layer-Struktur wird in Bild 2 auf ein Netz mit IMS angewandt. Dabei wurden im Vergleich zu Bild 1 vor allem für das Service Stratum und die AAA-Server Konkretisierungen anhand der IMS-Netzelemente (P-CSCF (Proxy-Call Session Control Function) und PDF (Policy Decision Function), I-CSCF (Interrogating-CSCF) und S-CSCF (Serving-CSCF) sowie HSS (Home Subscriber Server)) vorgenommen. Im Ergebnis wird nochmals deutlich, dass es sich auch hier, bei einem Netz mit IMS, um die Realisierung des NGN-Konzepts handelt. Letztendlich geht es beim IMS um nichts anderes als die vollständige und umfassende

Spezifikation des Service Stratum bzw. Call Control Layers eines NGN [4].



**Bild 2 Einordnung des IMS im Gesamtnetz**

Mit dem IMS wurde von 3GPP (3rd Generation Partnership Project) [5] ein IP Multimedia Overlay-Netz definiert, das für ein NGN nicht nur die Session-Steuerung, sondern auch die QoS-Bereitstellung (Quality of Service) sowie alle erforderlichen Funktionen für die Sicherheit, die Authentifizierung, die Zugriffssteuerung und die Erfassung von Abrechnungsdaten (AAA, mittels HSS) spezifiziert. Dabei wird auf Internet-Protokolle wie SIP, RTP und Diameter zurückgegriffen. Nicht zum IMS gehört gemäß Bild 2 der eigentliche Nutzdatentransport. Streng genommen gilt dies auch für das Application Stratum mit den Application Servern der Service Delivery Platform. Allerdings werden die AS in manchen Dokumenten (z.B. [6; 7]) auch mit zum IMS gezählt [4]. Ein IMS kombiniert Internet-Standards (IP, SIP u.a.) für kostengünstige Voice bzw. Multimedia over IP-Kommunikation mit den Konzepten (ZGS Nr.7 (Zentrales Zeichengabesystem), QoS, Sicherheit, Entgeltfassung) der heutigen ISDN- bzw. GSM-Telekommunikationsnetze [8]. Da SIP prinzipiell nicht zwischen Telefonie und multimedialen Diensten unterscheidet, ist die Einführung neuer Dienste vergleichsweise einfach. Zudem bietet die IP-Transportplattform eine hervorragende Basis für kombinierte Sprach-/Datendienste bzw. konvergente Dienste. Dabei ist jedoch hervorzuheben, dass das IMS mit Ausnahme der Basisdienste keine Dienste, sondern nur Service Enabler standardisiert [3]. Für die Dienstebereitstellung kommt vorzugsweise eine separate SDP zum Einsatz [4].

Im nun folgenden Abschnitt werden die verschiedenen Techniken, die bei der Realisierung einer Service Delivery Platform zum Einsatz kommen könnten, detaillierter erläutert.

### 3 Heutige Lösungen der Dienstentwicklung und -bereitstellung

Die heutigen Lösungen für die Dienstentwicklung und -bereitstellung liegen in Form einer sogenannten Service Delivery Platform vor. Zunächst werden die verschiedenen Lösungen kurz genannt, bevor diese dann genauer beleuchtet werden. Grundlegend existieren zwei Arten von Lösungen. Diese werden anhand ihrer Programmierschnittstellen unterschieden. Das sogenannte Low Level API (Application Programming Interface) kennzeichnet Techniken, die direkt auf Servern ausgeführt werden. Dagegen setzt das High Level API auf Middleware auf. Als Middleware wird eine Software-Schicht, die zwischen dem Netz und den Anwendungen liegt, bezeichnet. Die Aufgabe dieser Schicht ist es, die Eigenheiten und die Komplexität des verwendeten Netzes und der Anwendungen so zu abstrahieren, dass eine reibungslose standardisierte Interaktion stattfinden kann. Anhand dieser Definitionen lassen sich die heutigen Lösungen zur Dienstentwicklung und -bereitstellung wie folgt unterteilen.

#### *Low Level APIs*

- SIP Servlets
- CPL (Call Processing Language)
- SIP-CGI (Common Gateway Interface)
- JAIN (Java APIs for Integrated Networks)

#### *High Level APIs*

- Parlay/OSA (Open Service Access), Parlay X
- CSE (Customised Application for Mobile Network Enhanced Logic (CAMEL) Service Environment)
- Web Services
- OSE (Open Mobile Alliance Service Environment)

#### **SIP Servlets**

Servlets sind Java-Applikationen, die auf Servern laufen. SIP Servlets sind ähnlich den HTTP (Hypertext Transfer Protocol) Servlets, welche um ein API zur SIP-Kommunikation erweitert wurden. Standardisiert wurden die SIP Servlets von der Java Community [9] in den Java Specification Requests JSR 116 [10] und JSR 289 [11]. Der SIP Application Server stellt für die Servlets die Ausführungsumgebung zur Verfügung. Diese Ausführungsumgebung enthält u. a. den SIP-Protokoll-Stack, aber auch Timer für zeitgesteuerte Ereignisse. Empfangene SIP-Nachrichten werden anhand von Filtern an das auszuführende SIP Servlet weitergeleitet. Innerhalb eines SIP-Dialogs werden alle SIP-Kommunikationselemente durch ein und dasselbe Servlet bearbeitet.

Die Vorteile der SIP Servlets sind, dass sie in Threads und nicht in Prozessen, wie dies z.B. bei CPL oder CGI der Fall ist, laufen. Daher ist die Ausführungsgeschwindigkeit höher. Außerdem sind Servlets persistent, bis sie gezielt entfernt (destroy) werden. Servlets bieten eine hohe Sicherheit, da sie nur innerhalb des SIP-Serverprozesses laufen und deshalb nur über den Server selbst erreicht werden können. Sie sind darüber hinaus plattformunabhängig und aufgrund der Java-Basierung beliebig erweiterbar.

#### **CPL**

CPL ist eine XML-basierte (Extensible Markup Language) Programmiersprache, die zur Beschreibung von Multimedia-Diensten konzipiert wurde. Diese Programmiersprache wurde durch die IETF (Internet Engineering Task Force) im RFC 3880 (Request for Comments) [12] spezifiziert. CPL zeichnet sich durch seine Unabhängigkeit vom Betriebssystem und Signalisierungsprotokoll aus [13]. Durch Einschränkungen, die im definierten Befehlssatz begründet sind, können nur die dort bekannten Aktionen ausgeführt werden. Durch diese Eigenschaften bietet CPL ein hohes Maß an Sicherheit. Die Vorteile von CPL sind die Unabhängigkeit vom Betriebssystem und Signalisierungsprotokoll sowie eine hohe Sicherheit, begründet durch den eingeschränkten Befehlssatz. Der größte Nachteil ist auch durch den eingeschränkten Befehlssatz begründet, da z.B. keine externen bzw. dem Befehlssatz unbekannt Programme aufgerufen werden können.

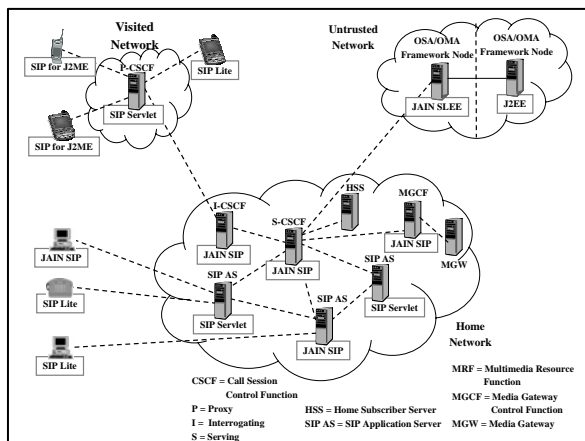
#### **SIP-CGI**

Aufgrund der Ähnlichkeit von SIP und HTTP eignet sich CGI gut für SIP-basierte Dienste [13]. SIP-CGI wurde im RFC 3050 [14] von der IETF spezifiziert. SIP-CGI ist eine sprachneutrale Schnittstelle, die Interaktionen auf SIP AS mit Programmen bzw. Skripten ermöglicht. Vorteil hierbei ist, dass prinzipiell alle Programmiersprachen wie z.B. Skriptsprachen (PHP (Personal Hypertext Pre-Processor), Python, Linux-Shellscript, Perl (Practical Extracion and Reporting Language)), Java, C oder C++ [15] einsetzbar sind, solange diese durch den Rechner, auf dem der SIP AS läuft, ausgeführt werden können. Eine weitere Eigenschaft von SIP-CGI ist, dass alle Daten der SIP-Nachricht an die ausführenden Programme bzw. Skripte weitergegeben werden. Außerdem haben normalerweise CGI-Skripte die gleichen Zugriffsrechte auf Ressourcen des Servers wie andere Serversoftware. Daher sollten nur Entwickler bzw. Administratoren des Providers selbst zur Diensteeerstellung eingesetzt werden, um Sicherheitsaspekte einzuhalten. Ein weiterer großer Nachteil von CGI ist die relativ langsame Ausführungsgeschwindigkeit. Beim Aufruf eines auf SIP-CGI basierenden Dienstes wird vom SIP AS ein Prozess des CGI-Programms gestartet und am Ende des Requests wieder beendet.

## JAIN

JAIN wird von der Java Community standardisiert [16]. Mit JAIN wird das Ziel verfolgt, ein Framework für hochportable und konvergente Dienste zur Verfügung zu stellen. Daher beinhaltet dieses Framework eine Anzahl von sogenannten Network APIs wie z.B. JAIN SIP oder JAIN MGCP (Media Gateway Control Protocol). Diese werden wiederum als Low Level APIs bezeichnet, da man direkten Zugriff auf alle Einzelheiten der Protokolle hat. Das JAIN-Konzept besteht aus drei Schichten, einer Netzwerkschicht, die Schnittstellen zu Festnetzen, Mobilfunknetzen und den IP-basierten Netzen beinhaltet, einer Signalisierungsschicht, die das Call Control beherbergt und einer Service-Schicht, in der die Dienstlogik und Ausführungsumgebung JSLEE (JAIN Service Logic Execution Environment) untergebracht ist. Die JSLEE spezifiziert eine universelle Ablaufumgebung für Telekommunikations-Services. Sie ermöglicht eine skalierbare, asynchrone Ereignisverarbeitung. JSLEE setzt sich aus drei Kernbereichen zusammen, dem Management zur Verwaltung von z.B. der Ablaufkontrolle und Profilen, dem Framework mit einem Ereignis-Router und dem Component Model, welches die Verwendung der Service Building Blocks (SBB) regelt [17]. In den SBBs sind die einzelnen Dienste realisiert.

Der breite Einsatzbereich der verschiedenen Java-Technologien (z.B. SIP Servlets und JAIN) in NGN/IMS ist in Bild 3 [18] zu sehen.

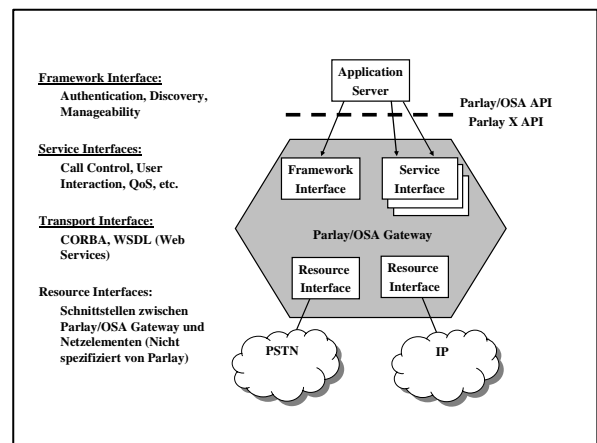


**Bild 3 Java-Plattformen und NGN/IMS**

### Parlay/OSA und Parlay X

Die Standards zu Parlay/OSA und Parlay X werden von ETSI (European Telecommunications Standards Institute), 3GPP (3rd Generation Partnership Project) und der Parlay Group entwickelt. Diese sind in UML (Unified Modelling Language) spezifiziert. Allgemein

wurden hier verschiedene Standard-Schnittstellen (APIs) definiert. Es sind also keine Application Server definiert worden, sondern eher Gateways. Mittels dieser Schnittstellen können 3rd-party Application Server sicher an ein NGN/IMS angebunden werden. Funktionen wie Discovery, Authentisierung, Registrierung und Zugriffssteuerung bietet OSA/Parlay selbst. Das OSA/Parlay Gateway konvertiert Protokolle wie z.B. SIP oder ISUP mittels der Service Capability Server (SCS) in das sogenannte OSA-API. Über das OSA-API können dann Application Server mittels eines sogenannten Transport Interfaces (z.B. Internet Inter-ORB Protocol (IIOP) bei CORBA (Common Object Request Broker Architecture)) angesprochen werden. Bild 4 zeigt eine Übersicht des Gateways [19].



**Bild 4 Parlay Gateway**

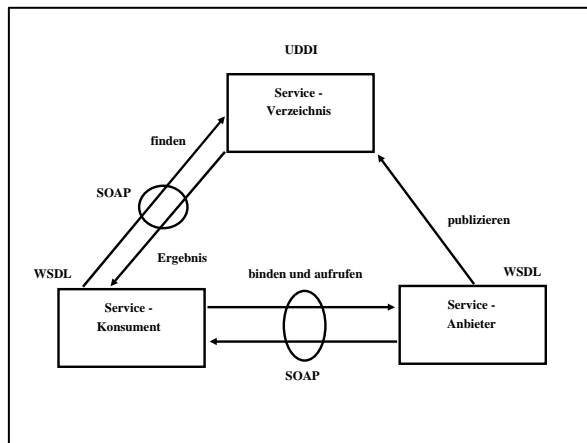
Parlay X basiert auf der Web Services-Technologie. Bei Parlay X handelt es sich um ein Programmiermodell, welches hauptsächlich Funktionen für synchrone Kommunikation bereitstellt. Für 3rd-party Access eignet sich Parlay X hervorragend aufgrund seiner Basisierung auf Web Services, welche später erklärt werden. Parlay X und Parlay/OSA können in den Gateways kombiniert zum Einsatz kommen.

### CSE

Das CSE basiert auf CAMEL-IN-Funktionalitäten (Intelligent Network). Mit seiner Hilfe wird die Nutzung vorhandener CAMEL-basierter IN-Dienste ermöglicht. Standardisiert wurde das CSE durch ETSI und 3GPP. Um die im CSE bereitgestellten Dienste nutzen zu können, bedarf es einer Protokollkonvertierung. Diese findet mittels der IM-SSF (IP Multimedia Service Switching Function) im NGN/IMS statt. Hier werden jeweils die Protokolle ISC (SIP) und CAP (CAMEL Application Part) ineinander konvertiert [6].

### Web Services

Web Services sind Software-Anwendungen, die sich an der SOA (Service Oriented Architecture) orientieren. Die Web Services verwenden standardisierte Schnittstellen (WSDL (Web Service Description Language)) und Protokolle (SOAP (Simple Object Access Protocol) oder XML-RPC (Extensible Markup Language Remote Procedure Call)). Sie sind unabhängig von der Programmiersprache, Plattform und dem Transportprotokoll (z.B. SIP oder HTTP). Basierend auf XML-Nachrichten (XML-RPC, SOAP) vereinen Web Services verteilte und objektorientierte Programmierstandards. Besonders hervorzuheben ist, dass Web Services untereinander kompositionsfähig sind. D.h., ein Web Service kann mit einem anderen zusammenarbeiten, um einen gewünschten Dienst zu ermöglichen. Aufgrund des nicht zu vernachlässigenden Overheads, der durch die eingesetzten Protokolle wie z.B. SOAP entsteht, leidet die Performance (Ausführungsgeschwindigkeit). Bild 5 zeigt den engen Zusammenhang zwischen SOA und Web Services, wobei das SOA-Konzept nicht an Web Services gebunden ist.



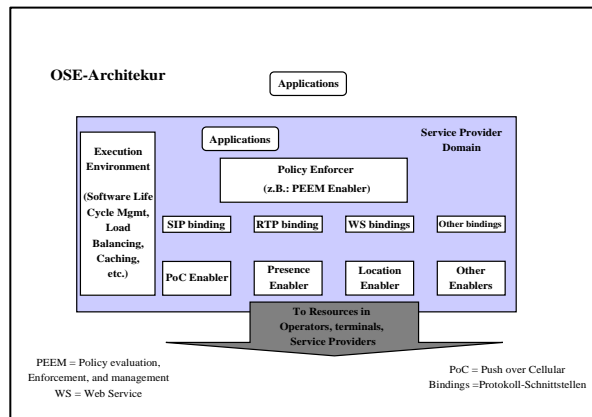
**Bild 5 SOA Find-Bind-Execute-Paradigma für Web Services**

### OSE

Die Open Mobile Alliance (OMA) liefert offene Spezifikationen zur Erstellung inter-operabler Dienste, die über Landesgrenzen, Betreiber und feste und mobile Terminals hinweg funktionieren. Vor dem Hintergrund der Bedürfnisse der Anwender und des wachsenden Marktes für Datendienste setzen sich die Mitglieder der Open Mobile Alliance für die Annahme neuer und verbesserter Informations-, Kommunikations- und Unterhaltungsdienste ein.

Der Open Mobile Alliance gehören wichtige Akteure und Unternehmen aus den zentralen Bereichen der Wertschöpfungskette im Wireless-Segment an [20]. Die OMA definiert im OSE sogenannte Enabler und Services. Es wird spezifiziert, wie die Enabler zusammenarbeiten und wie diese ihre Ressourcen (z.B.

Netzressourcen) bereitstellen. Diese Technologie ähnelt Parlay/OSA bzw. JAIN in großen Bereichen. Die in OSE spezifizierte SDP ist in Bild 6 dargestellt.



**Bild 6 OSE-Architektur**

### Gegenüberstellung der genannten Techniken

Anhand verschiedener Kriterien wie z.B. Sicherheit, Skalierbarkeit oder Performance wurden die genannten Techniken verglichen. Eine Übersicht zu den Kriterien wie Portabilität (Unabhängigkeit von Betriebssystem und Hardware), Sicherheit, Skalierbarkeit, Erweiterungsfähigkeit, Performance, Anzahl der Programmierer für die jeweiligen Programmiersprachen, Kosten pro System und benötigtem Entwicklungsaufwand für einen typischen Dienst liefert Tabelle 1.

**Tabelle 1 Vergleich der heutigen Lösungen zur Dienstentwicklung und -bereitstellung**

	Parlay	OSE	Web Services	SIP Servlets	CPL
Portabel (OS, Hardware)	ja	ja	ja	ja	ja
Sicherheit	hoch	hoch	hoch	gut	hoch
skalierbar	gut	gut	gut	gut	schlecht
erweiterbar	sehr gut	sehr gut	sehr gut	sehr gut	schlecht
Performance	schnell	gut	langsam	sehr schnell	langsam
Anzahl der Programmierer	viele ,außer für CORBA	sprachabhängig	sehr viele	sehr viele (benötigen aber SIP-Kenntnisse)	wenige
Kosten pro System	sehr hoch	hoch bis sehr hoch	mittel	mittel	gering
Entwicklungsaufwand	ca. 3 Monate	ca. 3 Monate	einige Wochen	einige Wochen	einige Tage

Mittels dieser Kenntnisse über die verschiedenen Techniken können nun Empfehlungen für Service

Provider, die ihren Kunden die geforderten Dienste möglichst effizient anbieten wollen, gegeben werden.

## 4 Beispielszenario

Es zeigt sich anhand der vorhergehenden Abschnitte, dass sich für Service Provider völlig neue Möglichkeiten der Dienstentwicklung und -bereitstellung ergeben. Zukünftige Dienste können genau an die Nutzer angepasst und binnen kürzester Zeit zur Verfügung gestellt werden. Welche Techniken sich für welchen Diensteanbieter bzw. Nutzer am besten eignen, wurde im Rahmen des mit der Detecon International GmbH realisierten Forschungsprojekts „Services in NGN“ anhand spezifischer Szenarien für Service Provider und deren Kunden umfassend aufgezeigt.

Ein konkretes Beispiel soll hier die Ergebnisse des Forschungsprojekts auszugsweise repräsentieren.

Ein UMTS-Mobilfunknetzbetreiber möchte Business-Kunden bedienen. Business-Kunden fordern spezialisierte Anwendungen wie z.B. termingesteuerte Video-Konferenzen oder sichere Datenübertragungen. Die Anzahl der geforderten Dienste wird daher verhältnismäßig gering sein. Da es sich um spezialisierte Dienste handelt, werden diese ein hohes Maß an Komplexität beinhalten. Um solche komplexen Dienste bereitzustellen, werden diese aus Kombinationen verschiedener Dienste und deren Leistungsmerkmalen zusammengesetzt.

Die hieraus resultierenden Anforderungen an die zur Dienstentwicklung und -bereitstellung verwendete Technik werden im Folgenden aufgezählt:

- beschränkte Anzahl an Diensten
- spezialisierte komplexe Dienste
- kombinierte Dienste
- hohe Performance
- gute Skalierbarkeit
- Unabhängigkeit von Betriebssystem und Hardware
- hohe Sicherheit
- angemessene Kosten für das System
- schnelle und leichte Entwicklung der Dienste
- einfache Erweiterungsmöglichkeit der Dienste
- leichte Bedienbarkeit der Dienste
- hohe Verfügbarkeit der Dienste

Aufgrund dieser Anforderungen kommen nur ganz bestimmte Lösungen in Betracht. Die bereits erläuterten Techniken SIP Servlets, JAIN, Parlay/OSA, OSE und Web Services eignen sich wegen ihrer Eigenschaften wie z.B. gute Skalierbarkeit, hohe Sicherheit und einfache Erweiterungsmöglichkeit der Dienste (siehe Tabelle 1) alle. Entsprechend dem explizit geforderten Dienst müssen hier jedoch weitere Kriterien zur Wahl des geeigneten Systems getroffen werden. Am Beispiel der bereits erwähnten Dienste (terminge-

steuerte Video-Konferenz und sichere Datenübertragung) wird dies verdeutlicht.

Für die „termingesteuerte Video-Konferenz“ wäre z.B. eine Kombination aus Web Services und SIP Servlets bzw. JAIN zu empfehlen. Die Web Services können hierbei hervorragend den Teildienst der Terminsteuerung übernehmen. Mittels eines Timers werden die Web Services z.B. dazu veranlasst einen Click-to-Dial auszuführen. Dies kann für beliebig viele Teilnehmer geschehen. Die Teilnehmerliste kann hierbei z.B. einer Datenbank entnommen werden, was auch durch die Web Services realisiert werden kann. Die Video-Konferenz selbst wird dann durch SIP Servlets oder die JAIN-Technologie realisiert. Dieses Beispiel verdeutlicht, dass mittels Kombination verschiedener Technologien, die besonderen Eigenschaften wie z.B. hohe Performance bei SIP Servlets oder kombinierte Dienste (Click-to-Dial und Datenbankabfrage) bei Web Services besonders stark genutzt werden können.

Für eine sichere Datenübertragung eignet sich z.B. Parlay/OSA besonders gut. Mit Hilfe der Gateways und der Funktionen wie z.B. Authentisierung, Registrierung lässt sich dieser Dienst vortrefflich umsetzen.

Wie hier gezeigt, können mittels einer Analyse der speziell geforderten Dienste und derer Anforderungen an den Provider treffende Empfehlungen für die zu verwendende Technik ausgesprochen werden.

## 5 Literatur

- [1] Trick, Ulrich; Weber, Frank; El Bouarfati, Soulaïmane: Modellierung heterogener Telekommunikationsnetze. ITG-Fachbericht 187 Mobilfunk, S. 41-49, Juni 2005
- [2] X.200: Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. ITU-T, July 1994
- [3] Magedanz, Thomas: IP Multimedia Subsystem (IMS) als Dienstplattform für Next Generation Networks. Vermittlungstechnisches Kolloquium Wien, 22.06.2006
- [4] Trick, Ulrich; Weber, Frank: SIP, TCP/IP und Telekommunikationsnetze. Oldenbourg, April 2007
- [5] <http://www.3gpp.org/>
- [6] TS 23.002: Network architecture (Release 5). 3GPP, September 2003
- [7] TS 123.228: IP Multimedia Subsystem (IMS); Stage 2 (3GPP Version 7.5.0 Release 7) ETSI, September 2006
- [8] Magedanz, Thomas: IMS als Dienstplattform für Netzbetreiber und MVNOs. Detecon Management Report 04/2005
- [9] <http://jcp.org/aboutJava/communityprocess/background.html>

- [10] Kristensen, Anders: SIP Servlet API Version 1.0. JSR 116, February 2003
- [11] <http://www.jcp.org/en/jsr/detail?id=289>
- [12] Lennox, J.; Wu, X.; Schulzrinne H.: RFC 3880 - Call Processing Language (CPL): A Language for User Control of Internet Telephony Services. IETF, October 2004
- [13] Magedanz, Thomas: Application Servers for Telecommunications. TU Berlin
- [14] Lennox, J.; Schulzrinne, H.; Rosenberg, J.: RFC 3050 – Common Gateway Interface for SIP. IETF, January 2001
- [15] Müller, Günter; Eymann, Torsten; Kreutzer, Michael: Telematik- und Kommunikationssysteme in der vernetzten Wirtschaft. Oldenbourg, 2003
- [16] [http://java.sun.com/products/jain/api\\_specs.html](http://java.sun.com/products/jain/api_specs.html)
- [17] Maretzke, Michael; Haiges, Sven; Bröcker, Christoph: Ereignisorientierte Komponenten mit JAIN SLEE. Java Spektrum Ausgabe 05/2005
- [18] Ranganathan, M.: JAIN-SIP: Architecture, Implementation, Testing. NIST, May 2002
- [19] LeClerc, Marc: Parlay/OSA and Parlay-X and their role in creating new revenue streams for Operator. Boston Member Meeting, October 2005
- [20] Fraher, Bobby: Open Mobile Alliance beendet 2006 mit bedeutenden Meilensteinen. OMA, December 2006

ITG-Fachbericht 202 Mobilfunk, S. 101-107, Mai 2007