Mobile Service Delivery Platform

By David Eror

26. 11. 2010

The recent experience of explosive growth of wireless data-access in 3G networks and the expected data-access "storm" in 4G networks has led to demands for change in mobile network architecture. The Evolved Packet Core (EPC), based on a single all-IP core network, is designed to meet the high demands for increased penetration of different services and smart tablet devices.

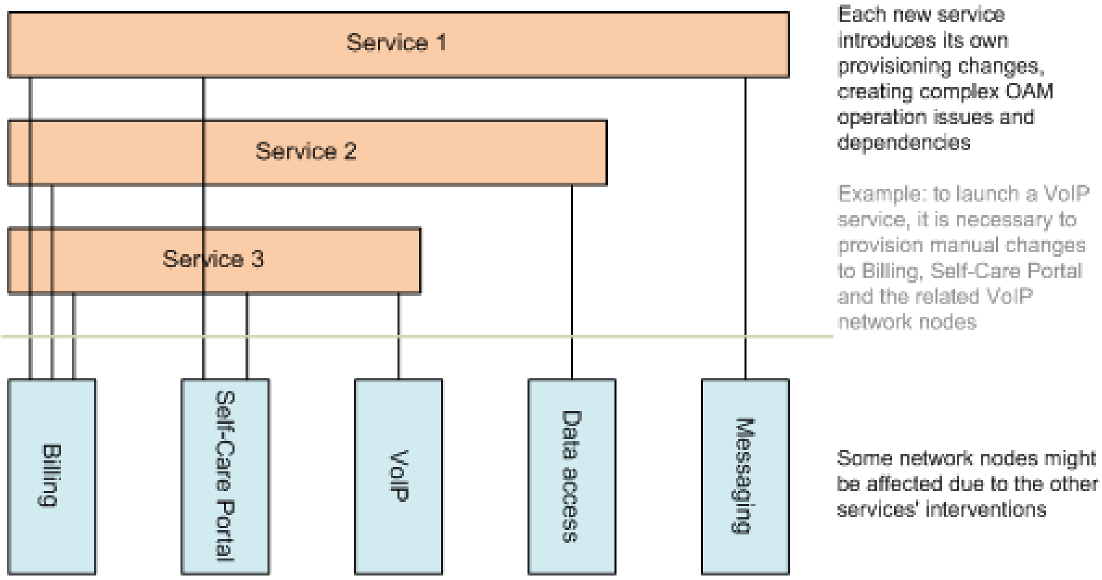
We're talking about one multipurpose middleware for a mobile service provider (SP), to handle the SP's commercial (service) delivery efforts toward the network. An all-IP cloud needs to make more sense, to be manageable from a single platform. Investments in mobile-content differentiation and service delivery, together with real-time charging, are required, because legacy platforms have become obsolete.

The main role of the mobile Service Delivery Platform (SDP) is the creation of a **service layer** that is separate from the core network elements. Unfortunately, this SDP hasn't yet been standardized, despite ongoing efforts of the Telemanagement Forum (TM Forum) industry association. The benefit offered to SPs is the potential to **create and implement these**

new services without the necessity of provisioning the core network elements. The SDP is the platform responsible for the **rapid creation of the services**, implementation and handling, across the heterogeneous networks (fixed, mobile, enterprise). Although the vast majority of the services are mobile-originated, this platform could be common to the complete network, regardless of the access technology.

There is no single definition of what functionality the SDP should include, precisely; therefore, its implementation is quite vendor-dependent. The aim of the introduction of the SDP is minimizing of the service provisioning and implementation issues, thereby increasing efficiency within an SP's internal business process. The criteria that SPs are following in the evaluation process are mostly based on the following:

- Revenue gain - including predefined services (within the SDP)
- Integration – how difficult it is to add an SDP to the existing infrastructure
- Technology – high availability, geographic redundancy, load-sharing etc.



Each new service introduces its own provisioning changes, creating complex OAM operation issues and dependencies

Example: to launch a VoIP service, it is necessary to provision manual changes to Billing, Self-Care Portal and the related VoIP network nodes

Some network nodes might be affected due to the other services' interventions

Figure 1: Service implementation without SDP

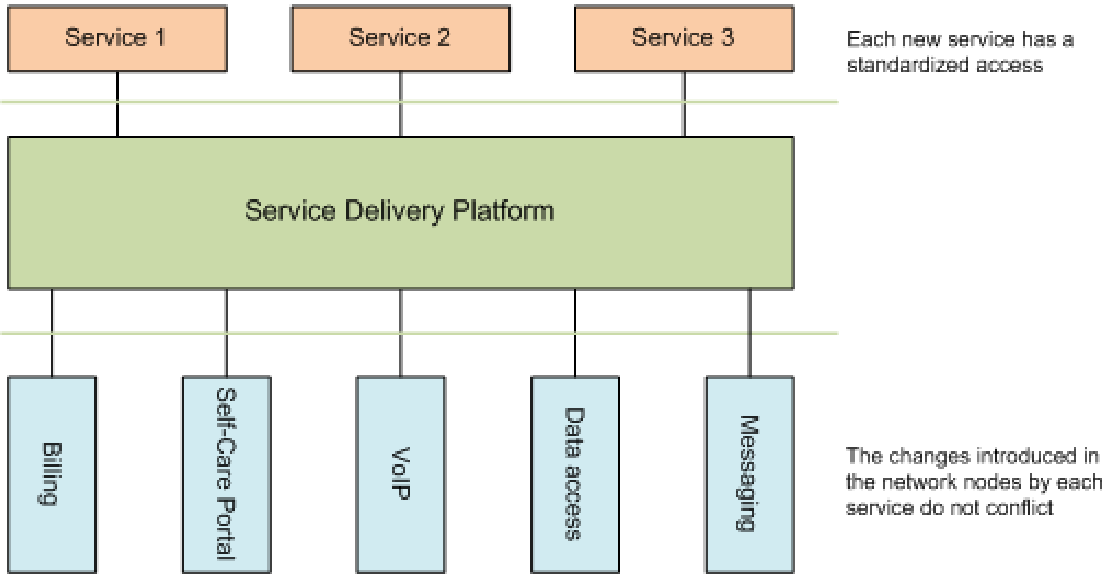
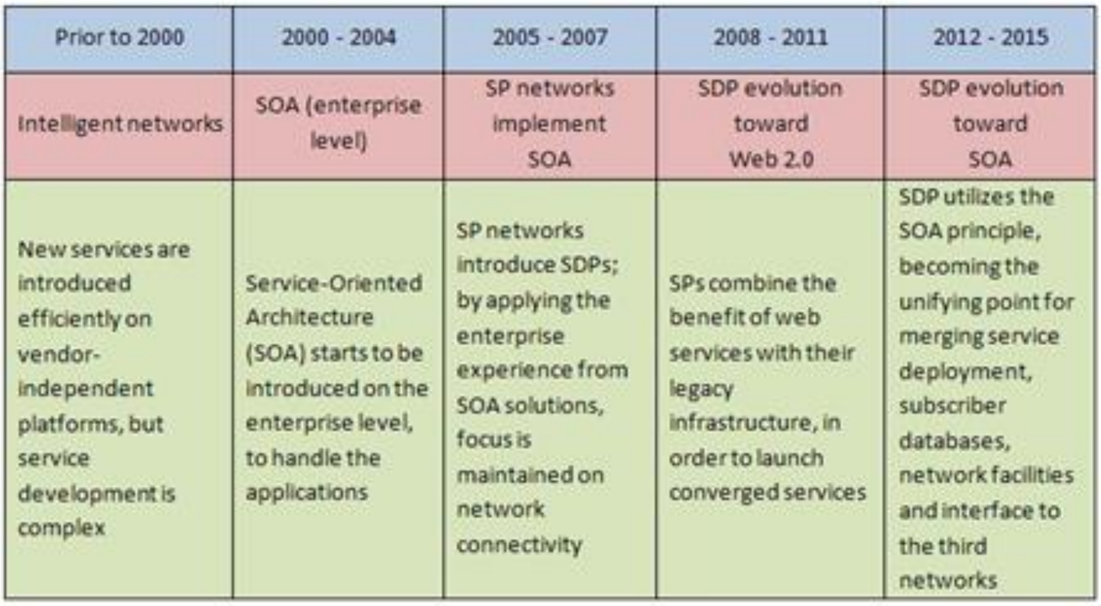

Service 1
Service 2
Service 3
Each new service has a standardized access
Service Delivery Platform
Billing
Self-Care Portal
VoIP
Data access
Messaging
The changes introduced in the network nodes by each service do not conflict

Figure 2: Service implementation after the introduction of SDP

Service Delivery Evolution

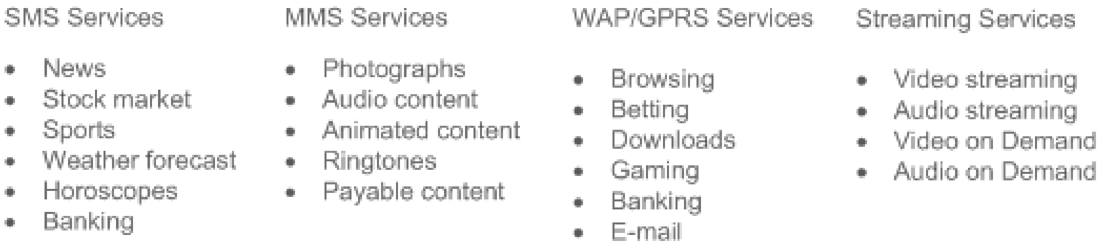
Due to the increased necessity of developing the new programmed interfaces between different network elements, the SDP has become an important network element that has been implemented rapidly by SPs. The recent customer requests are related to the built-in services on ready-to-use platforms, in order to be faster to market. Consequently, the suppliers are granted by the revenue sharing model, instead of the license fee.



Prior to 2000	2000 - 2004	2005 - 2007	2008 - 2011	2012 - 2015
Intelligent networks	SOA (enterprise level)	SP networks implement SOA	SDP evolution toward Web 2.0	SDP evolution toward SOA
New services are introduced efficiently on vendor-independent platforms, but service development is complex	Service-Oriented Architecture (SOA) starts to be introduced on the enterprise level, to handle the applications	SP networks introduce SDPs; by applying the enterprise experience from SOA solutions, focus is maintained on network connectivity	SPs combine the benefit of web services with their legacy infrastructure, in order to launch converged services	SDP utilizes the SOA principle, becoming the unifying point for merging service deployment, subscriber databases, network facilities and interface to the third networks

Figure 3: SDP evolution phases

The main task of suppliers is resolving the business case (built-in applications – see example in Figure 4), followed by integration with Customer Relationship Management (CRM) systems and Operations Support Systems (OSS), and finally platform hardware and software capabilities (high availability, scalability, load sharing…).



SMS Services

- News
- Stock market
- Sports
- Weather forecast
- Horoscopes
- Banking

MMS Services

- Photographs
- Audio content
- Animated content
- Ringtones
- Payable content

WAP/GPRS Services

- Browsing
- Betting
- Downloads
- Gaming
- Banking
- E-mail

Streaming Services

- Video streaming
- Audio streaming
- Video on Demand
- Audio on Demand

Figure 4: Example of built-in SDP services

SDP Reference Architecture

The Service Delivery Platform is often delivered as a bundle in which the vendors might include several other components; consequently, it might be called the "SDP framework." These other components could be soft switches, Session Initiation Protocol (SIP) application servers, Internet Protocol Television (IPTV) middleware, Video on Demand (VOD) servers, Java APIs for Integrated Networks (JAIN), Parlay or Open Services Architecture (OSA) gateways, and they are seen as the directly related functionality, tied to SDP but not residing within the SDP domain.

Application programming interface (API) tools are the standardized capabilities that create an interface for the third-party platforms to introduce new services into the core network elements. Several industry standards have been developed:

- Java APIs for Integrated Networks (JAIN)
- Open Services Architecture (OSA)/Parlay
- XML-based scripts, such as VXML/CCXML

API tools determine the classes and methods that are used by the SP network operators. In addition to APIs, it is possible to use scripting languages in an XML-based format that connect the existing APIs with the network elements. Such an architecture is able to facilitate the use of common resources (databases, service support, core network elements) to be used among several other interfaces, with no dependencies. Real-time charging is the important resource of the service recognition and handling, according to the provisioned user-profile. The OAM interface through the SDP is an additional benefit for the SP support team.

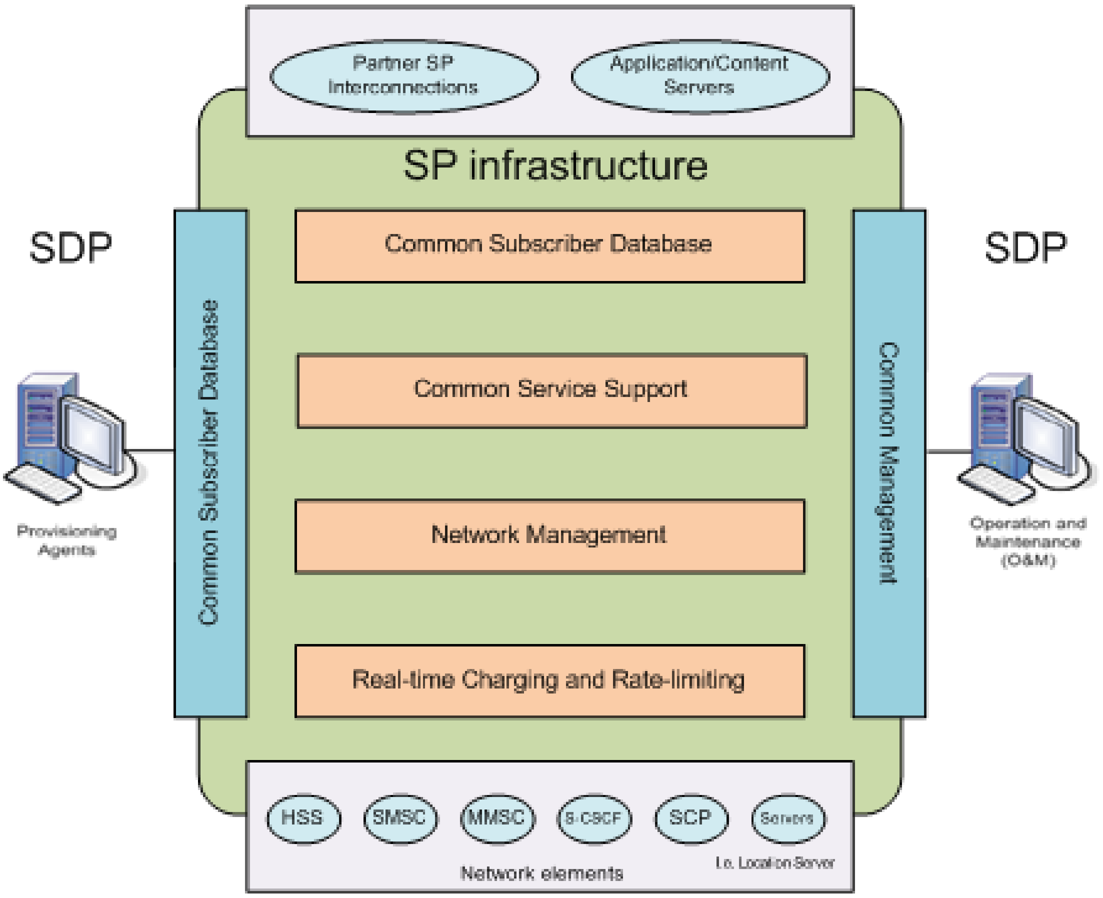

Partner SP Interconnections
Application/Content Servers
SP infrastructure
SDP
SDP
Common Subscriber Database
Common Service Support
Network Management
Real-time Charging and Rate-limiting
Common Subscriber Database
Common Management
Provisioning Agents
Operation and Maintenance (O&M)
HSS
SMSC
MMSC
S-CSCF
SCP
Servers
I.e. Location Server
Network elements

Figure 5: SDP reference architecture. The SDP connects to the common SP infrastructure and enables centralized service-provisioning changes.

JAIN APIs

Java-based APIs on functions and protocol layers (JAIN) define (Figure 5) the JAIN Service Logic Execution Environment (JSLEE) and the JAIN Service Creation Environment (JSCE). Use of JAIN offers the service migration across the next-generation network (NGN), regardless of the network nodes on which it will be implemented. There are different APIs related to the various call-control protocols (JAIN SIP, JAIN ISUP etc.) that are responsible for session handling toward the third-party platforms:

- JAIN SIP API. Related to the SIP that handles VoIP gateways and endpoints to intercommunicate. The JAIN SIP API provides user-agent, proxy and redirect server interfaces.
- JAIN ISUP API. Handles the signaling function supporting the switched voice connection.
- JAIN Operations, Administration, and Maintenance (OAM). Creates, deletes, modifies and monitors the network nodes, etc.

The JAIN Service Creation Environment (JSCE) implements the graphical tool that builds the service; it consists of the modular service building blocks accessed by the secured-access JAIN Service Provider API (JAIN SPA).

The execution and management environment JSLEE manages the services' interaction between the service layer and the network – it deploys the created services.

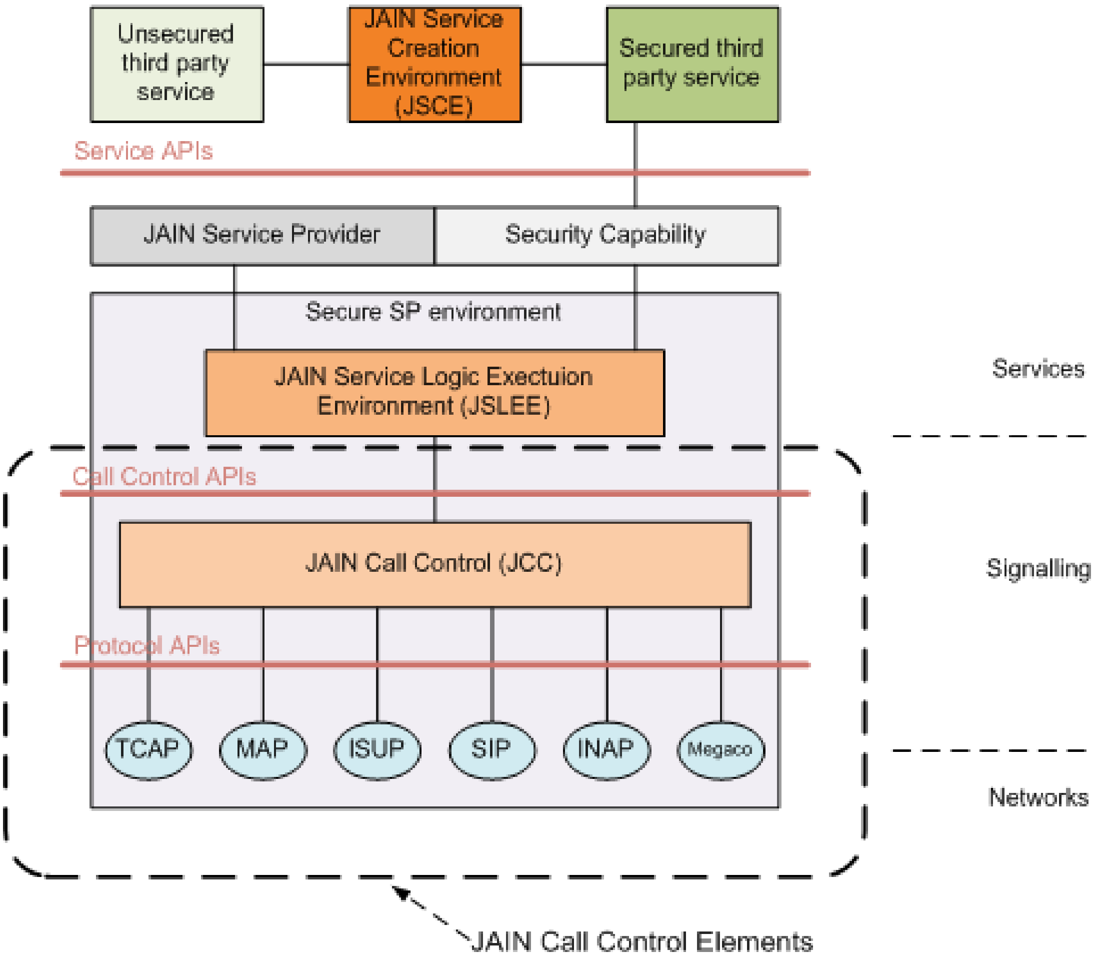


Figure 6: JAIN architecture

OSA/Parlay APIs

Open Services Architecture (OSA) is the 3GPP standardized architecture for the 3G mobile services' deployment. Parlay is the OSA's API part and does not replace the network protocols. OSA/Parlay creates the network-independent APIs, suitable to the mobile, fixed and any future NGN architectures. Parlay's standardization is still ongoing, and it is built on key open standards as Common Object Request Broker Architecture (CORBA), Unified Modeling Language (UML) and web services such as Simple Object Access Protocol (SOAP), eXtensible Markup Language (XML) and Web Services Description Language (WSDL). Service brokering is the Parlay function of service selection and provisioning.

Since Parlay has sometimes been seen as too complex, the web services use and applications employ the more adapted Parlay X API interface, with a more efficient abstraction of the communication capabilities, which is open to a wide Internet development community.

The OSA architecture contains (Figure 7):

- **Applications** such as VPN, conferencing and location-based services …
- A **framework** of the enabling applications to use the Service Capability Features (SCFs), through authentication, registry and discovery methods defined in the OSA interface
- **Service Capability Servers** providing the applications with SCFs, which are abstractions from the related network functionality

The framework of APIs can be grouped into the following functional areas:

- Call control – Enable the setup of the call
- User interaction – Define how applications obtain information from the end user
- Mobility – Enable applications to find the location of the terminal
- Availability management and presence - Obtain information about the user's presence and availability
- Policy management – Set up policies
- Terminal – Determine the capabilities of the end-user's terminal

- Messaging – Allow access to the mailboxes (email, voice messaging, fax)
- Content-based charging – Control how different applications are billed
- General – Introduction, methodology and design paradigm used
- Common data definitions – Generic datatypes used throughout the Parlay specifications
- Frameworks – Define how applications authenticate themselves on the network
- Data session control – Define how applications handle data sessions initiated from the terminals
- Account management – Retrieve accounting information
- Connectivity management – Control quality of service between the endpoints

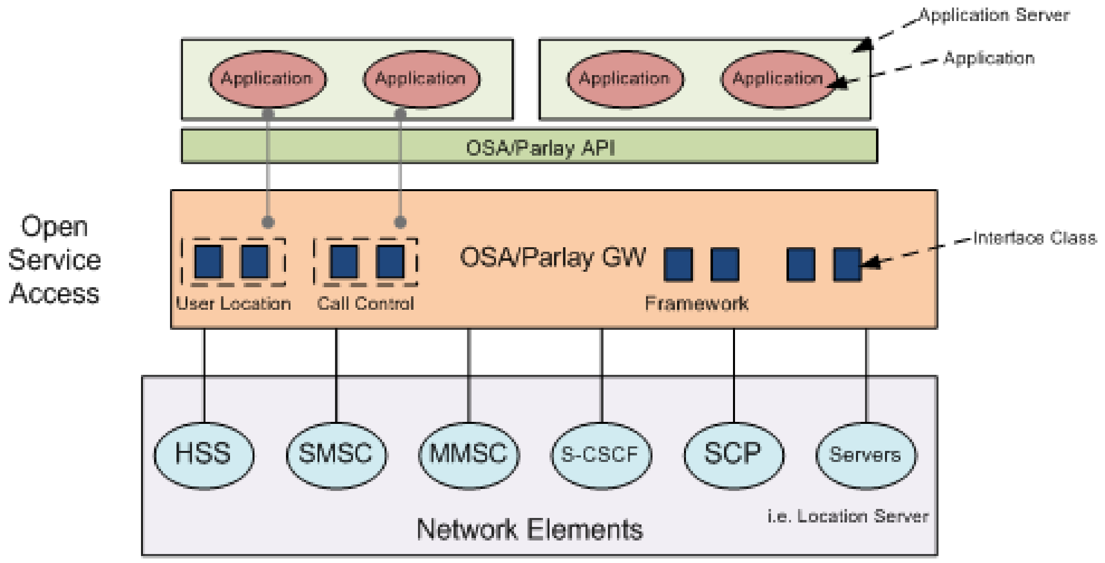


Figure 7 (Source: DEVOTEAM): Parlay architecture

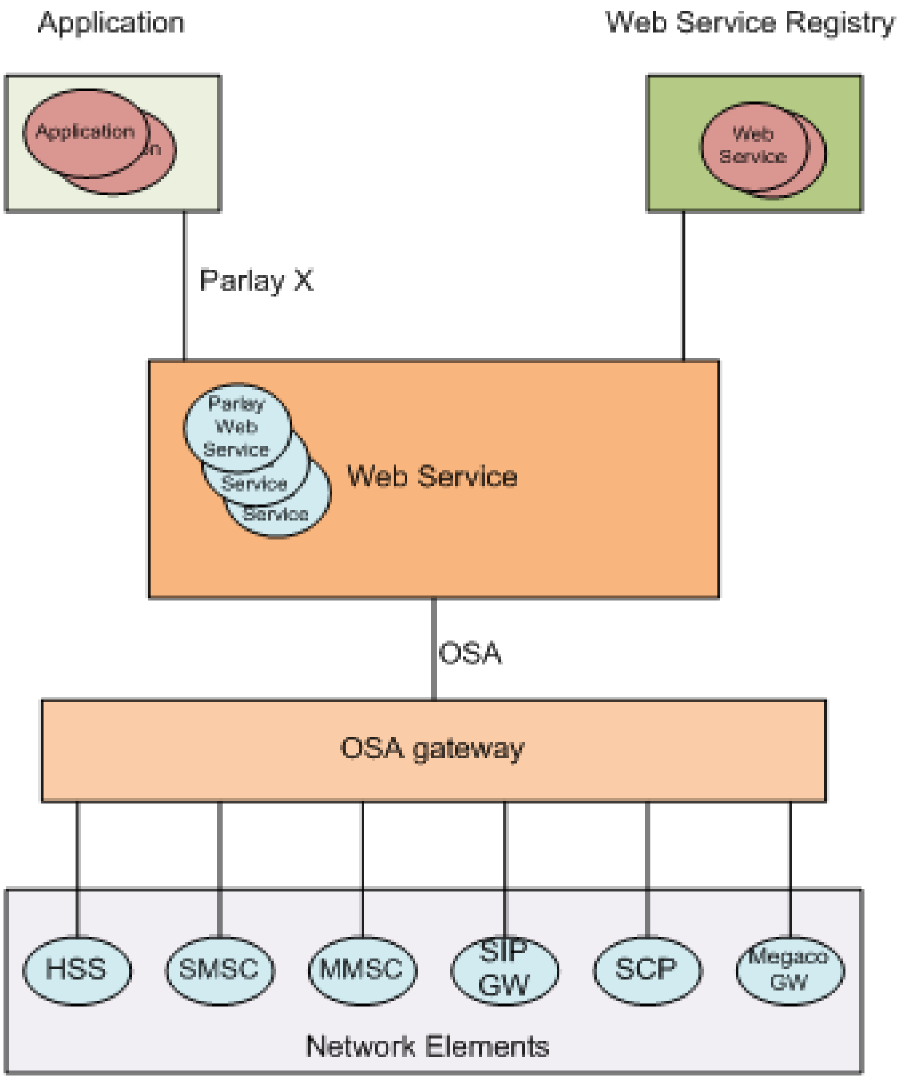


Figure 8 (Source: DEVOTEAM): Parlay X architecture

VXML/CCXML scripts

Voice eXtensible Markup Language (VXML) is a flavor of XML used for the control of the voice browser (web browser as a voice application), such as voice recognition, dual-tone multi-frequency signaling (DTMF) and text-to-speech playback. VXML-based deployment uses media server facilities to be deployed based on HTML protocol; for example, to an Interactive Voice Response (IVR) application. The great advantage of VXML is the low level of expertise required by developers, thus providing for rapid service creation (Figure 9).

For advanced use of the call control with the means of the XML-based scripts, VXML is not completely appropriate and has limited capabilities.

For such a purpose, another flavor of XML is used complementary to VXML: Call Control XML (CCXML). One example would be CCXML-based control of the ringback melody played from the media server, click-to-dial or conference, and controlled by the SDP. CCXML could also be used in non-voice contexts, together with or separately from VXML.

!

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
<meta name="description" content="hello world example"/>
<meta name="author" content="NIL, d.o.o., Slovenia (http://www.nil.si)"/>
<meta name="copyright" content="free for any purpose"/>
<var name="hi" expr="'Hello World!'"/>
<form id="say_hi">
  <block>
    <prompt><value expr="hi"/></prompt>
    <goto next="#say_goodbye"/>
  </block>
</form>
<form id="say_goodbye">
  <block>
    Goodbye!
  </block>
</form>
</vxml>
```

!

FIGURE 9: (Source: WORLD WIDE WEB CONSORTIUM, www.w3.org): An example of the VXML “Hello world.”

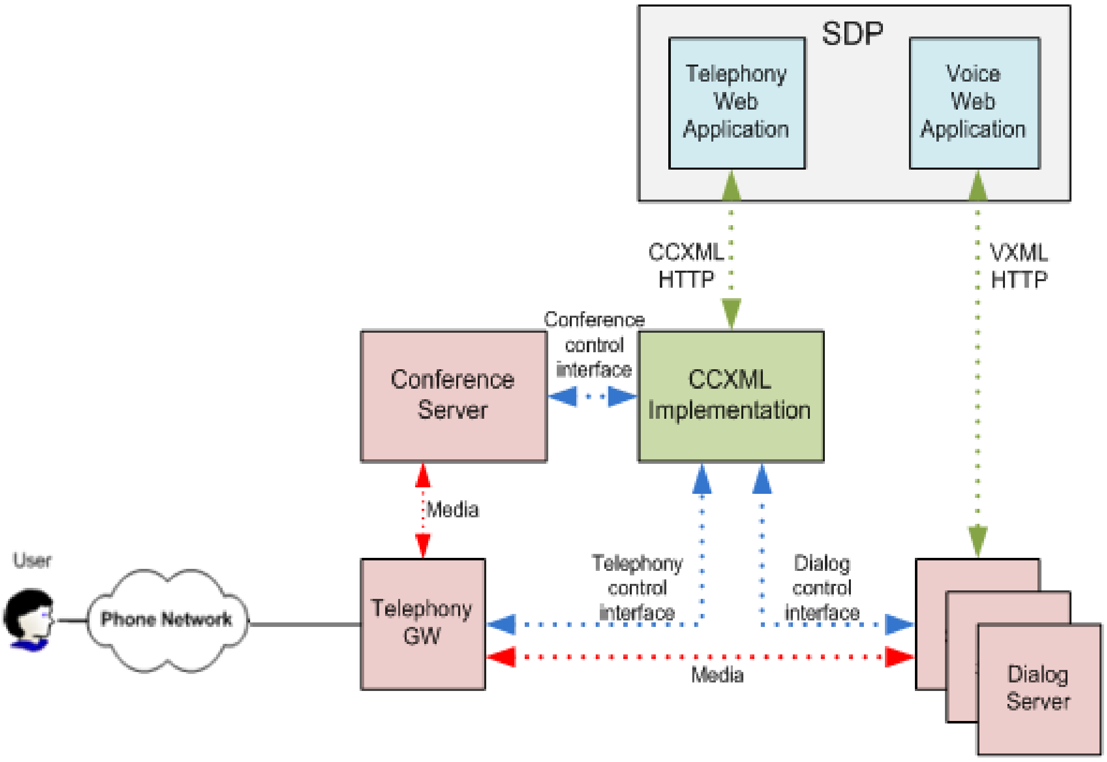


Figure 10 (Source: WORLD WIDE WEB CONSORTIUM, www.w3.org): VXML/CCXML-controlled call setup.

Figure 10 explains the architecture of a telephony implementation of an Interactive Voice Response (IVR) application, consisting of several components:

- Caller from the phone network
- Dialog server (e.g., a VXML implementation of the voice browser facility)
- Conference server used to mix media streams
- SDP-based CCXML implementation that manages the connections between the first two components listed above

The telephony control and dialog control interfaces may be implemented as an API or a protocol.

Conclusion

Despite generally negative market trends, there has been **steady growth in** SDP market share (currently $3 B -

http://www.researchandmarkets.com/reports/1071153/service_delivery_platforms_market_review.htm). Service providers provide the services in the global application stores and resist the recession impact (more than 3 billion downloads reached). While there are more than 4 billion mobile users in the world (2/3 prepaid, 1/6 using tablets), they could create $1,000 B (2/3 of the voice revenue) by the year 2011. For example, U.S. citizens have access to a billion web pages, hundreds of TV stations, several thousand newspapers and hundreds of thousands of mobile applications. In the next five years, the number of tablets is expected to grow more than tenfold. Voice is a decreasing revenue generator within tablet services, since mobile subscribers are transforming themselves from voice users to Internet users, using a wide set of different services (voice is just one service).

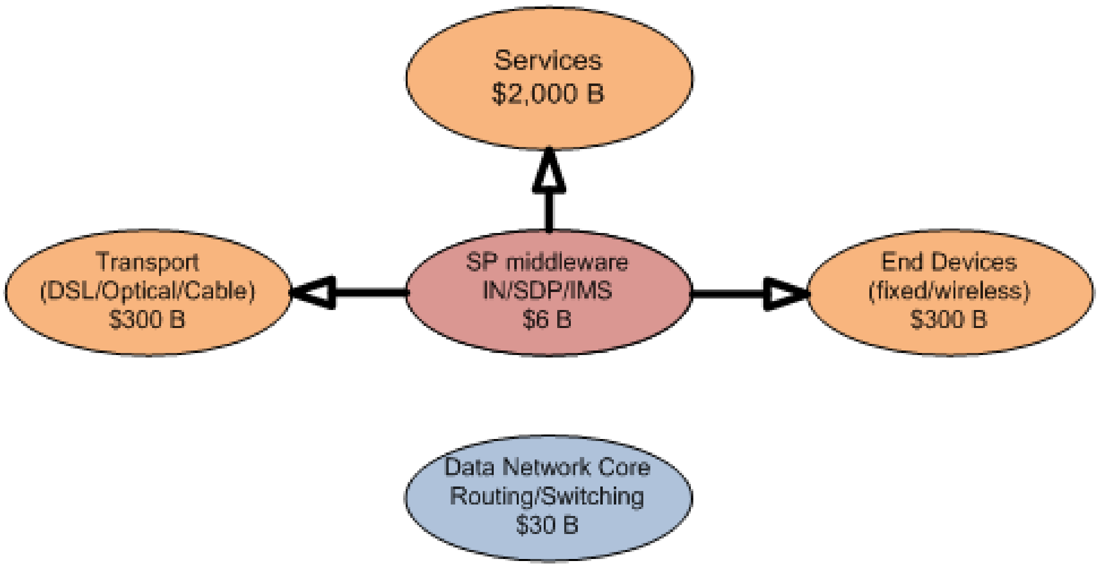


Figure 11 (Source: ALAN QuAYLE): SDP as an indirect revenue generator for suppliers.

NIL – More Than Just a Training Company

NIL Learning delivers the leading-edge Cisco training to IT professionals and companies around the globe. Through field-proven experts — each both active engineer and instructor — NIL Learning enhances the standard learning curriculum with real-life experience and helps clients to maximize their training investment.

NIL Learning is part of NIL, a leading global IT solutions provider. Since 1992, NIL has been at the forefront of advanced contributors to strategic partner Cisco's technologies, learning curriculum and value-added solutions deployed to clients around the globe. Today, NIL has earned the highest certifications offered by Cisco, VMware, EMC, HP, IBM, Microsoft, F5, Jive, MobileIron, RSA, VCE and others. Their portfolio of solutions consists of managed services, professional services and learning services.

NIL is headquartered in Slovenia, with regional offices in Croatia, Serbia, Saudi Arabia, the U.S., Turkey, South Africa, Morocco, Nigeria, Kenya and Botswana.

Why learn at NIL LEARNING?

- All NIL LEARNING instructors are **field-proven experts** - each both active engineer and instructor.
- **75% of NIL LEARNING engineers hold CCSI** certifica- tions, and **18 have already achieved the respected CCIE rank**.
- NIL LEARNING **enhances the standard learning curriculum** with real-life experience and helps clients to maximize their training investment.
- NIL has been a Cisco Training Partner for many years; it became a **Cisco Learning Partner in 1993**, and has been a **Cisco Gold Partner since 1995**.
- NIL was awarded the **Cisco Most Business Relevant Learning Partner in MEA in 2010 and the most innova- tive learning partner in MEA**.

- NIL received the **Innovation Award for its Technology Led Training and its extensive contribution to Cisco learning solutions** at the Cisco EMEAR Learning Partner Summit in 2012.
- NIL received the **Innovation Award for its Technology Led Training and Advanced Engineer Program** at the Cisco Global Learning Partner Summit in 2013.
- NIL LEARNING runs a **centralized training schedule across the whole EMEAR region**.

More Info

Slovenia

T: +386 1 4746 500

E: sales-support@nil.com

Botswana

T: +267 318 1684

E: training@it-iq.bw

Croatia

T: +385 (0)51 583 255

E: info-nilcroatia@nil.com

Kenya

T: +27 (0)11 575 4637

E: mea_sales@nil.com

Morocco

T: +212(0) 660 808 394

E: info-nilmorocco@nil.com

Nigeria

T: +27 (0)11 575 4637

E: mea_sales@nil.com

Saudi Arabia

T: +966 1 465 4641

E: info.nilme@nil.com

Serbia

T: +381 11 2282 818

E: info-nilserbia@nil.co.rs

South Africa

T: +27 (0)11 575 4637

E: mea_sales@nil.com

Turkey

T: +902 123 81 8639

E: info-nilturkey@nil.com

USA

T: +1 612 886 3900

E: info-nilusa@nil.com

www.learning.nil.com