

Building a new generation of communication services using Macromedia® Flash® and HP Service Delivery Platform

TABLE OF CONTENTS

1	Overview
3	HP Service Delivery Platform and Service Chaining
4	Flash technology
5	Developing Flash and Flash Lite applications using SDP and Service Chaining
8	Flash application example using SDP services
16	Conclusion
16	References

1. Overview

Purpose

This document is intended to provide Macromedia Flash and Flash Lite™ application developers with an understanding of how to integrate telephony and other communication services into Flash based applications using HP Service Delivery Platform (SDP). It contains a description of the HP and Adobe solution components as well as an example use case from both the user and the Flash developer viewpoints.

Market overview

In today's market, convergence and increased competition are commoditizing traditional communication services. Accordingly, carriers must rapidly bring to market a new generation of communication services that boost customer adoption and new revenue while minimizing costs.

Business and residential users alike want the ability to tap into information and communication services wherever and whenever they need them. And they want easy-to-use multimedia group communication services that leverage rich, personal, and relevant content across networks, over the web, and across different device types. Unfortunately, service enablers that support rich communications such as presence, location, conferencing, and messaging are often trapped in silos of functionality within a carrier's network. Consequently, subscriber services are delivered piecemeal, making them difficult to use and costly to support.

Carriers struggling to implement next-generation service strategies face significant challenges including:

- Unlocking network resources and safely and securely opening up the network to application developers
- Enabling a consistent, easy-to-use user interface across different devices and network types
- Differentiating services to ensure continued revenue growth
- Rapidly and cost-effectively bringing new services to market

To enable carriers to address these challenges and offer a new generation of communication services, HP and Adobe have partnered to integrate Macromedia Flash technology from Adobe with HP Service Delivery Platform. By leveraging ubiquitous Flash technology and HP's telecom products, knowledge, and expertise, carriers can offer services based on their own unique set of network assets and control the delivery and branding of the service end to end.

One of the objectives of the partnership between Adobe and HP is to unleash the Flash development community on the communications marketplace. SDP enables a carrier to safely and securely provide controlled access to network resources, which are then made available to over one million Flash development community members who stand ready to bring a new generation of combined voice and rich Internet applications to market faster and more efficiently.

With SDP and Flash technology, carriers will be able to unlock independent "stovepiped" network resources to create integrated, next-generation communication services and present them through simplified user interfaces that operate consistently across devices and networks. The result will be faster time to market with new services, increased service usage, lower costs, and increased revenue generation.

2. HP Service Delivery Platform and Service Chaining

The Service Delivery Platform is HP's blueprint for developing and deploying standards-based end-user services across multiple network types (fixed, mobile, and broadband) and network generations (2G/2.5G/3G/IMS/MMD).

SDP provides more secure, operator-defined levels of access to network elements, allowing carriers to safely open the network and easily extend service development opportunities to the vast field of content and application developers—including more than one million Flash developers who are eager to share the risks and rewards of introducing new services. Deploying services within this blueprint allows carriers to reach the market faster with new niche services, reduces the risk and complexity of deploying services, simplifies service interaction, and eases service management, resulting in increased revenues and reduced costs.

SDP follows the OMA recommendation for applying a service-oriented architecture (SOA) to the telecom environment and contains four logical subsegments, as depicted in Figure 1. Following is a brief description of these layers.

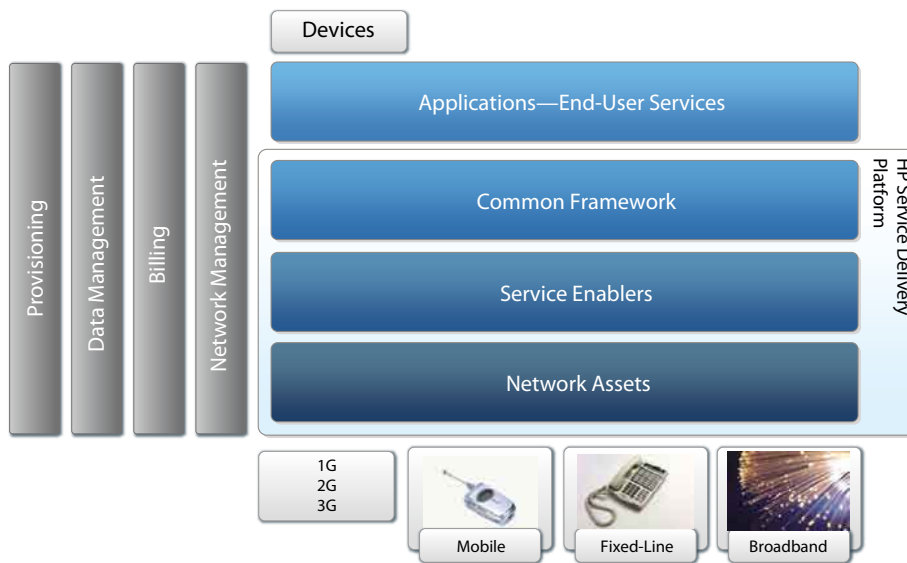


Figure 1: HP Service Delivery Platform

The **Network Assets** layer contains all of the key network resources within the operator environment. It includes core network infrastructure, signaling platforms, service control point, and home location registers.

Service Enablers comprise network services, off-network services, and network gateways. Network and off-network services include the simple services—such as location, presence and availability, messaging, and media streaming—that are required to build sophisticated end-user services. Network gateways abstract core network functions and provide hooks to lower level network assets, in addition to simplifying access to the underlying protocols used to communicate with core network assets.

The **Common Framework** layer is the central control point for many aspects of service delivery including:

- **Third-Party Framework (3PF)**—The Third-Party Framework implements policy control and identity management for in-house services and external service providers and includes the authentication and authorization of access to the Service Enablers and Network Assets layers.
- **Content management**—The content management function provides the ability to ingest, archive, and manage digitized content from internal or external sources.
- **User interaction and presentation**—This component manages how services and content are presented over multiple channels and devices and delivers device management capabilities.

- **Service Chaining**—This application infrastructure uses web services to enable session preservation and collaborative messaging across multiple communication services such as location, presence, audio conferencing, web conferencing, and instant messaging. This chaining of services facilitates the creation of broad new suites of interesting, fun, and collaborative multimedia services.
- **Platform support functions**—These are abstracted interfaces to OSS/BSS systems.

The **Applications** layer addresses the creation, validation, and hosting of end-user services. HP offers worldwide consulting services that can help network operators design service delivery ecosystems that most effectively meet their specific market requirements.

Service development can be performed either internally within the network operator or externally through third parties using the SDP blueprint. SDP uses common service development toolsets that support Macromedia Flex™ and Flash 8 application development environments. They also support Parlay, Parlay X, .NET, Java,™ web services, and Session Initiation Protocol (SIP) capabilities that make it simple for developers to integrate telecom and IT functionality into their applications and core business processes. The Flex and Flash 8 application development environments are the focus of this paper.

3. Flash technology

Flash technology offers a tremendous opportunity for carriers to improve the customer experience of content and applications on the Internet and bring real returns to the business in the form of increased customer satisfaction, improved productivity, stronger branding, and reduced support costs. Flash technology allows the delivery of rich, user-centric application user interfaces, multimedia content and applications, and real-time communications and collaboration solutions. The Flash based components utilized for the joint HP and Adobe solution (and featured in the use case described in Section 5) include Flash Player, Flash Lite, Flash 8, Flex, and Macromedia Breeze.*

Flash Player is the backbone of Flash technology, supported by a layer of tools, servers, components, patterns, and developer programs that provide the on-ramp for designers and developers to build rich content and applications. Flash Player is a high-performance, lightweight (less than 1MB), highly expressive client runtime that delivers powerful and consistent user experiences across major operating systems, browsers, mobile phones, and devices. Today it is installed on over 600 Internet-connected computers and mobile devices. Flash Player also runs on Pocket PC PDAs, such as the HP iPAQ.

Flash Lite is specifically designed for mobile phones and allows the delivery of rich, engaging experiences that increase ease of use and data revenue. It runs on a variety of embedded operating systems, supports integration with a phone's operating system, provides network support for dynamic data in applications, and utilizes hardware codecs when available. Flash Lite is used for a range of purposes on mobile phones, from content to applications to the actual phone user interface itself. It runs on top of the phone operating system and delivers a rich set of capabilities.

Flash 8 is an authoring tool and development environment that allows designers and developers to integrate video, text, audio, and graphics into immersive, rich experiences—from simple animations to complex interactive web applications, such as an online store. Flash 8 includes many features that make it both powerful and easy to use, such as drag-and-drop user interface components and built-in behaviors that add ActionScript to documents and special effects to objects.

Flex is a presentation system that enables programmers to develop applications that combine the responsiveness and richness of the desktop with the reach and deployment characteristics of the web. Flex provides a programming model that is intuitive to both enterprise and web developers and a comprehensive set of components that fosters rapid application development.

Flex applications are stateful and are neither cluttered with page refreshes nor limited to a handful of user interface controls. They can expose rich user interface metaphors such as drag and drop, support smart client-side data manipulation (for example, client-side sorting and filtering), and access a local data store to work in an offline mode.

Breeze is a rich web communication solution that delivers high-impact online communications that can be accessed instantly through Flash Player, providing a powerful, end-to-end solution with ubiquitous reach. Breeze was built on Flash technology to meet the demanding functionality, scalability, and reliability requirements of communication and collaboration applications. It is available as licensed software that can be hosted on premise or on a subscription (ASP) basis.

4. Developing Flash and Flash Lite applications using SDP and Service Chaining

Writing rich Flash and Flash Lite applications using the communications service enablers provided by the SDP platform is a compelling proposition for the Flash developer community. Web services are exposed by the SDP platform and consumed by Flash and Flash Lite applications through Flash 8 web service classes.

Web services bring a standards-based solution to client-server integration. The term *web services* describes a standardized way of integrating web-based applications using the XML, SOAP, WSDL, and UDDI open standards over an Internet protocol backbone. XML is used to tag the data; SOAP, to transfer the data; WSDL, to describe the services available; and UDDI, to list what services are available. Flash 8 natively supports XML.

Service Chaining extends SDP by enabling common communication services to be blended, simplifying and enhancing the end-user experience. Because Service Chaining uses standard web-based protocols and languages, it allows application developers to rapidly include combinations of communication services—such as location, presence, instant messaging (IM), short messaging service (SMS), audio conferencing and Breeze web conferencing—within their applications without time-consuming application-to-application integration. Service Chaining components within SDP leverage web services one step further by facilitating the integration of web-based and traditional telephony services using the existing network infrastructure.

Figure 2 illustrates the SDP and Service Chaining framework and provides a starting point for describing how a Flash application can leverage the capabilities exposed by the SDP environment.

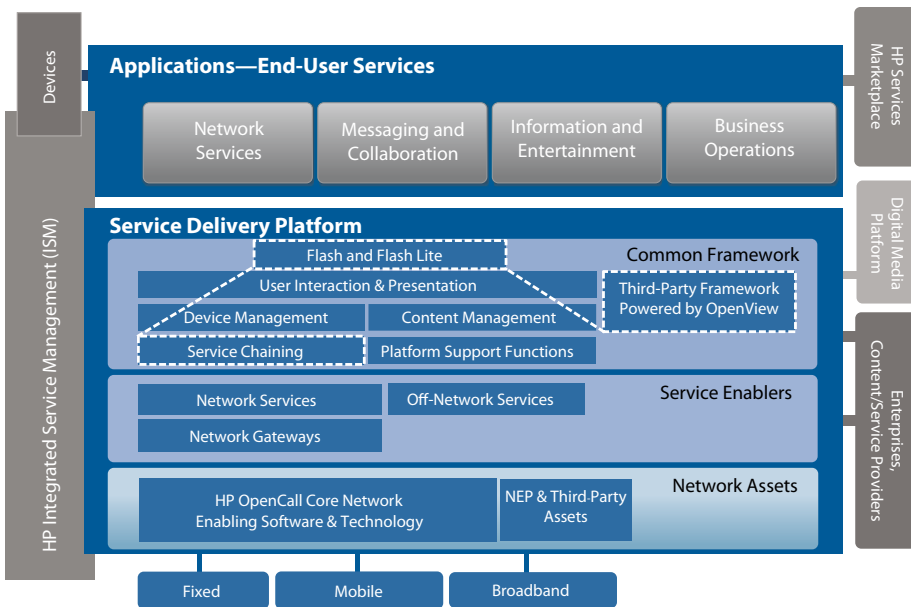


Figure 2: Service Delivery Platform and Service Chaining

Client applications targeting Flash Player may be developed using the Flash 8 IDE or Flex 1.5, depending on the target Flash runtime version. The development paradigm is a SOA approach whereby services are registered in SDP: The Third-Party Framework's UDDI Service Registry and parameters for service invocation are exposed as WSDL XML documents. The Flash 8 IDE exposes the service access points and parameters required for runtime execution to application developers, allowing them to include these services within the Flash application. Example services include those typically found within the carrier environment, such as presence, location, audio conferencing, and messaging (IM, SMS, MMS). This assumes that developers have authorized access to the services that they will bind into the application (policy management is handled by the Third-Party Framework). Flex allows web services to be declared at development time. At runtime, the services are invoked directly through Service Chaining components, the Flex runtime, or some combination of these methods.

Figure 3 provides an expanded view of how Flash and Flash Lite based applications interact with SDP and Service Chaining. The Third-Party Framework includes the UDDI Service Registry, which houses web service definitions and pointers for policy-managed and controlled service enablers in the framework. HP's Service Chaining components are themselves service enablers. There are four main Service Chaining service enablers: Service Controller, ccXML, Group List Management Server (GLMS), and Context Repository. Typical operator-controlled service enablers used in conjunction with Service Chaining are location, audio conferencing, instant messaging, and Breeze web conferencing. SDP includes a portal and the underlying scalable infrastructure that allow service providers to make available any number of service enablers from inside or outside their network. All of these services are registered, policy managed, and controlled through the SDP Third-Party Framework.

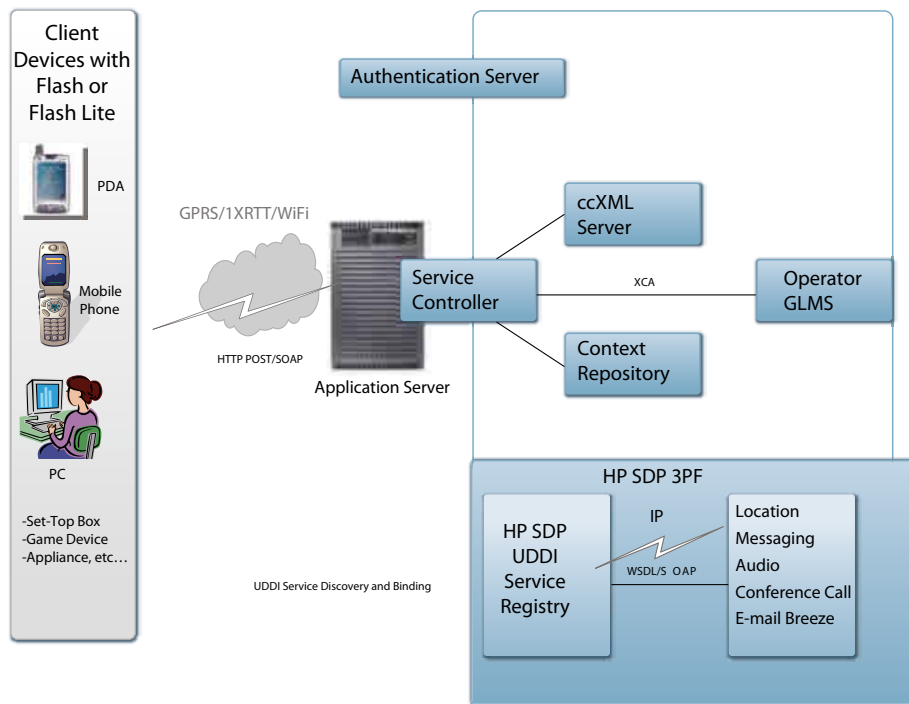


Figure 3: Expanded view of SDP Service Chaining

Flash applications talk to the Service Controller through XML. The Service Controller is responsible for parsing the incoming XML, checking which service to invoke, getting the WSDL from the UDDI repository, calling the web service operation, managing the context, and returning the result code and response XML to the client device.

Figure 4 depicts a more detailed view of the Service Chaining architecture.

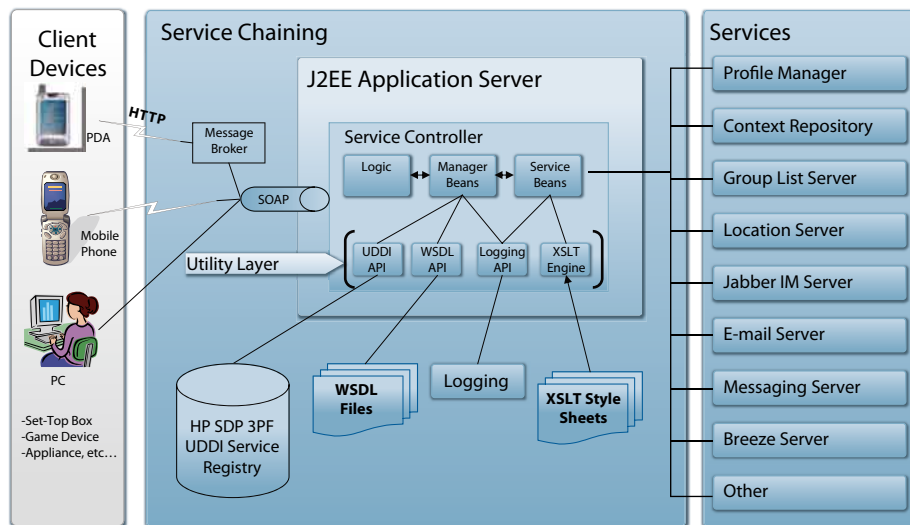


Figure 4: Service Chaining architecture

The Service Controller is a J2EE application through which service requests are executed. Examples of execute methods available through the Service Controller include:

- createMemberList—Retrieves members of a group list from the GLMS.
- getMemberList—Retrieves the member list from the GLMS.
- getLocation—Retrieves location information from the location server registered in the SDP Third-Party Framework.
- getServices—Performs additional processing that might be required to properly format data.
- Authenticate—Handles all authentication requests from the client.
- CleanSession/Logout—Clears the Context Repository and cleans up session data at the end of a client session.
- createConference—Gathers member phone numbers and initiates the conference call.
- setSelectedMembers—Places user-selected member names and associated user profile data into the Context Repository.
- addUserToSession—Adds a new user to an existing communications session.
- startBreeze—Starts a Breeze conference session with the specified members.

These types of services are available in conjunction with all other service enablers exposed within the SDP Third-Party Framework and can be incorporated into the Flash application.

5. Flash application example using SDP services

The following use case and application code examples illustrate the power of Flash and SDP integration and how communication services can be chained together using the Flash development environment to enhance a subscriber's communication capabilities and deliver added value.

Use case

Note: The following illustrates an enterprise business use case of the technology. The same services can easily be used for consumer-based use cases simply by modifying the Flash or Flash Lite GUI interface.

John runs a widget factory. At his warehouse, he notices a case of widgets that was mistakenly left on the loading dock. From the packing slip, he knows that these widgets must get to the customer later in the day. John contacts his operations manager, Paul. Paul uses his HP iPAQ PDA to determine which delivery drivers are closest to the warehouse and can pick up and deliver the widgets to the customer. He launches an instant conference call with all drivers in a specified radius, determines who will deliver the package, and sends out a routing and confirmation e-mail message.

In this case, Paul implements the following sequence of steps:

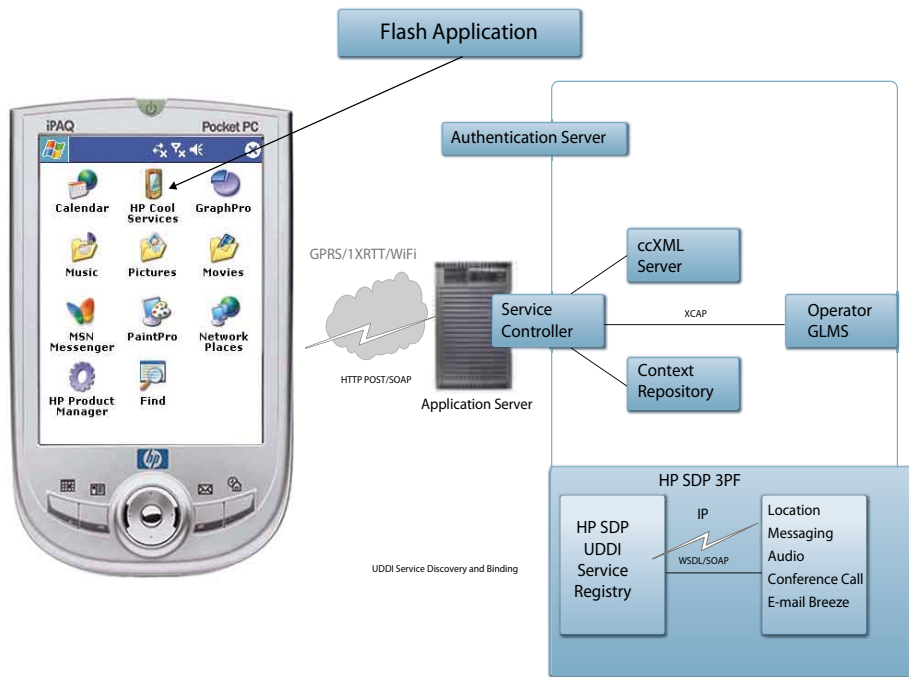
1. Paul clicks his Cool Services application and is automatically authenticated.
2. Paul's various group lists are displayed.
3. From the group lists, Paul selects the Drivers group.
4. A list of the drivers' names is displayed.
5. Paul selects several drivers from the group and clicks a Near Me location lookup service button. He is presented with a map showing the relative location of the selected drivers to the warehouse. Paul sees that four drivers are within a short distance of the warehouse.
6. Paul decides to launch a conference call with the drivers who are located within his Near Me range.
7. Paul launches the conference call service.
8. A conference call is established, and one of the drivers is selected to pick up the widgets.
9. The driver is sent a route map to ensure a successful delivery to the customer.

Application example

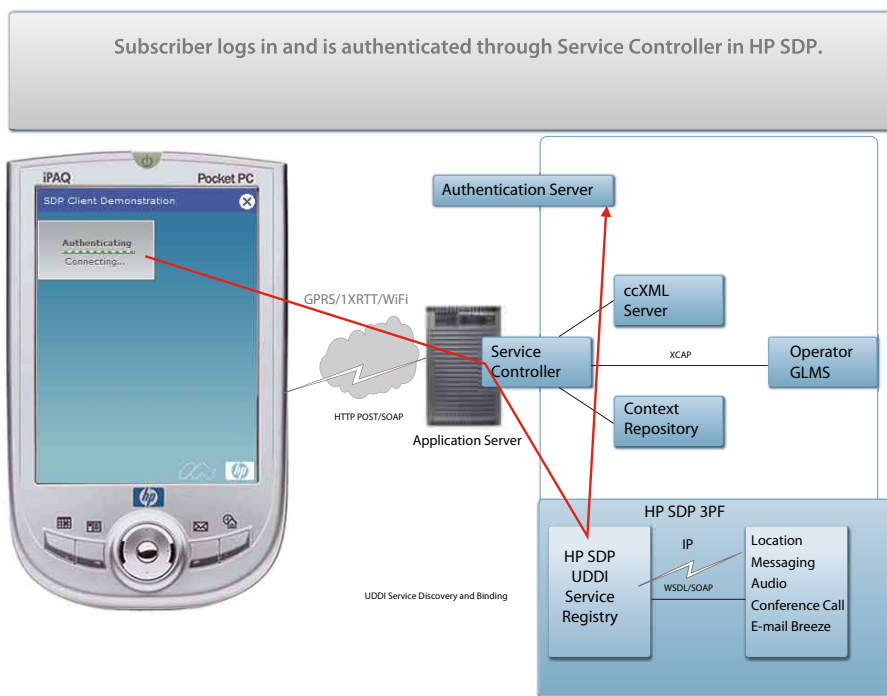
The following diagrams illustrate the application that enables this use case scenario. A more detailed description and examples of the underlying code follow.

The client application in its entirety consists of a single user interface file (such as main.fla) together with several ActionScript objects that handle communication with the network services, as well as the runtime logic of the application on the client side.

Note that the application illustration was developed using the Flash 8 IDE. In a Flex based development scenario, the concepts would be identical, but the development techniques would differ.



1. A Flash based application called HP Cool Services running on the HP iPAQ under Windows® Mobile is created. This application uses the SDP-provided execute methods described in Section 4.



2. When subscribers click the HP Cool Services application, they are logged in to the network and authenticated. No assumptions are made as to the authentication mechanism for the prototype that was developed. It is assumed that service providers will determine the authorization and authentication mechanism that best meets their requirements.

User view:

User logs in and is authenticated on the network with preprovisioned authorization to access and use various services.

Flash developer view:

Flash ActionScript invocation of web service.

```
// Call the execute method and pass it an XML string for authentication.  
service.execute("  
<?xml version='1.0' encoding='UTF-8' ?>  
<SDPSCRequest xmlns='SDP' xmlns:xsi='http://www.w3.org/2001/XMLSchema-  
instance'>  
<Version>01.00</Version>  
<OperationType>Authenticate</OperationType>  
<UserData>  
<UserName>Adobe</UserName>  
<Password>Adobe</Password>  
</UserData>  
</SDPSCRequest>  
");
```

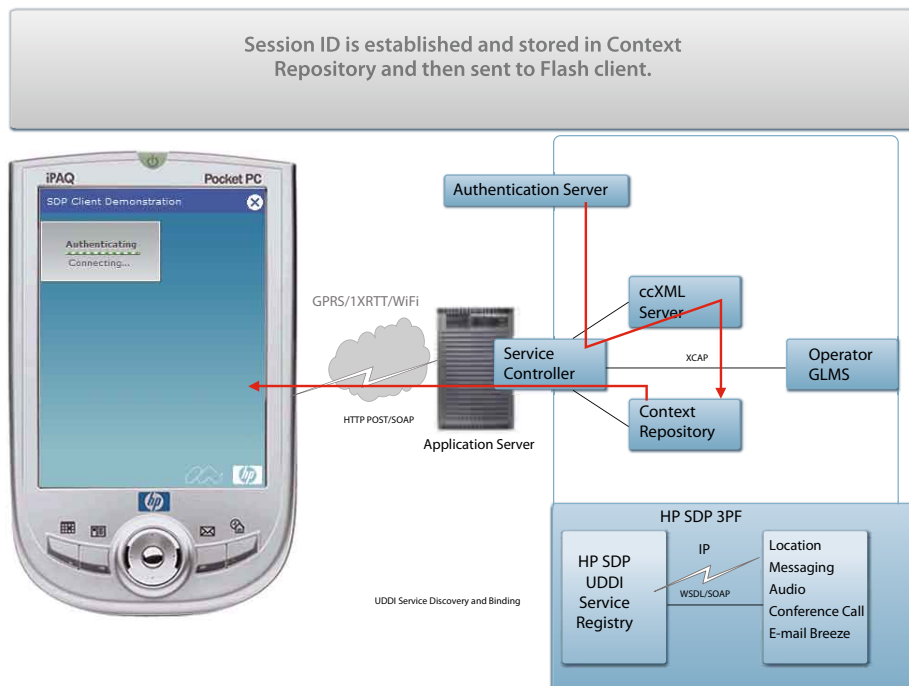
2a. Once the user is authenticated, the Service Controller establishes a session by retrieving a session ID from the ccXML server and forwarding it to the user application on the iPAQ.

User view:

No change in user view.

Flash developer view:

```
// Set the session ID for usage later.  
model.sessionID = XPathAPI.selectSingleNode(x.childNodes[1], "/  
SDPSCResponse/SessionID").firstChild.nodeValue;
```



3. The Service Controller, an SDP Service Chaining component, manages the sequencing of services. The first step is to establish a session ID. ccXML is used because it facilitates the crossover of web-based and circuit-switched voice communication services. The session ID is used as the database key in the SDP Context Repository. This maintains a record of the

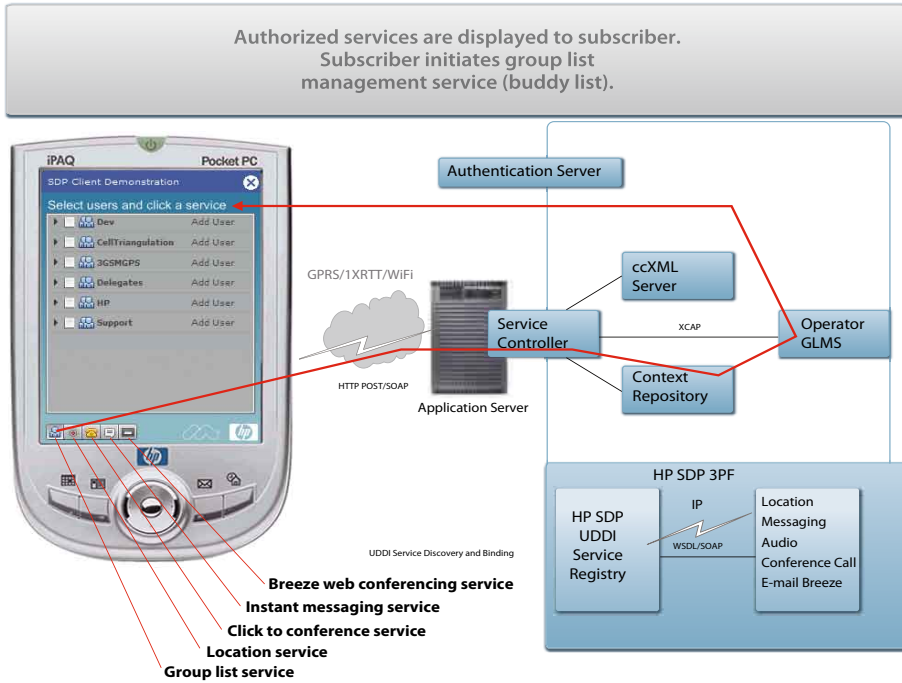
services implemented as well as collects metadata about services initiated during a communication sequence. The session ID is also sent to the Flash client. It is used as an identifier for subsequent user-initiated services during the session and, if required, for later reenactment of the service sequence.

User view:

No change in user view.

Flash developer view:

```
// Set the session ID for usage later.
model.sessionID = XPathAPI.selectSingleNode(x.childNodes[1],
"//SDPSCResponse/SessionID").firstChild.nodeValue;
```



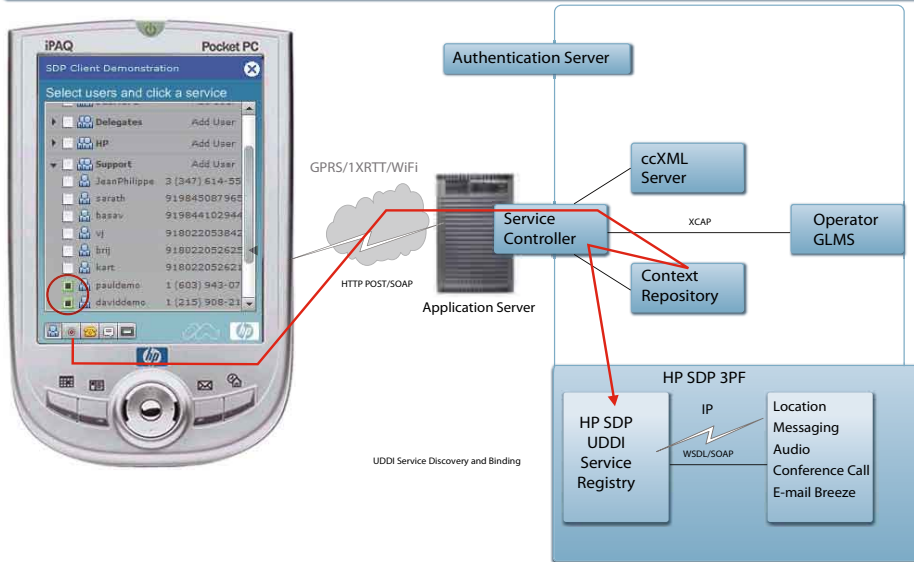
4. Based on a user's authorization, a set of service icons is displayed. The supposition is that the number of service icons will grow or shrink based on the services subscribed to and authorized for a particular user. By default, the group list service (buddy list) is the first service displayed to the user. The user selects the group member to communicate with and the next service. This creates a "service chain." As the user progresses through services, context (collected metadata) from one service to the next is maintained.

Flash developer view:

```
// Call the execute method and pass it an XML string to return all members.
service.execute("
<?xml version='1.0' encoding='UTF-8' ?>
<SDPSCRequest xmlns='SDP' xmlns:xsi='http://www.w3.org/2001/XMLSchema-
instance'>
<Version>01.00</Version>
<OperationType>GetAllMembers</OperationType>
<SessionID>53978</SessionID>
</SDPSCRequest>
");
```

After the results are returned, the XML is converted to an ActionScript object that is used to populate the member list tree.

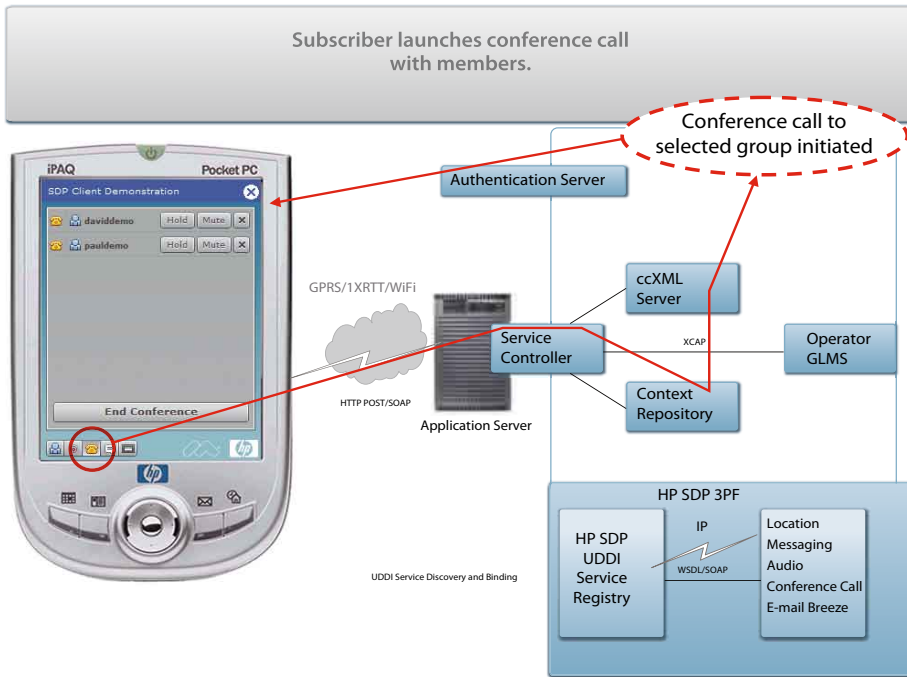
Subscriber selects buddies and initiates location service.
 HP SDP controls connection to authorized location services
 entity (may be a network service delivered from the operator
 or location services provided by a third party).



- When a user selects the location service icon, the Flash application sends the invocation request to the Service Controller along with the session ID that was previously sent to the client application. SDP and Service Chaining facilities route requests for location services through policy-managed and controlled third-party providers of location services. The location service provides a map with appropriate positioning information. Icons representing selected group member locations are positioned on the map and displayed to the user.

Flash developer view:

```
// Call the execute method and pass it an XML string to return the
locations.
service.execute("
<?xml version='1.0' encoding='UTF-8' ?>
<SDPSCRequest xmlns='SDP' xmlns:xsi='http://www.w3.org/2001/XMLSchema-
instance'>
<Version>01.00</Version>
<OperationType>GetLocation</OperationType>
<SessionID>53978</SessionID>
<PayloadSize>1</PayloadSize>
<PayloadEntry>
<MemberID>1</MemberID>
<MemberPhone>4158322000</MemberPhone>
<MemberType>Driver</MemberType>
  <MemberName>Vijay</MemberName>
  <EmailID>vjaiy@hp.com</EmailID>
</PayloadEntry>
</SDPSCRequest>
");
```

7. The user selects the conference call service and initiates a conference call.

Flash developer view:

Formatted for readability.

```
// Call the execute method and pass it an XML string to start a
conference.
service.execute("
<?xml version='1.0' encoding='UTF-8' ?>
<SDPSCRequest xmlns='SDP' xmlns:xsi='http://www.w3.org/2001/XMLSchema-
instance'>
<Version>01.00</Version>
<OperationType>StartCallConference</OperationType>
<SessionID>53978</SessionID>
<PayloadSize>1</PayloadSize>
<PayloadEntry>
<MemberID>1</MemberID>
<MemberPhone>4158322000</MemberPhone>
<MemberType>Driver</MemberType>
  <MemberName>Vijay</MemberName>
  <EmailID>vjaiy@hp.com</EmailID>
</PayloadEntry>
</SDPSCRequest>
");
```

The sequence of steps to implement additional communication services such as instant messaging or initiating a Breeze web conference is essentially the same as that for selecting group members or using previously selected group members and clicking the appropriate service icon.

Flash developer view:

```
// Call the execute method and pass it an XML string to start a Breeze
conference.
service.execute("
<?xml version="1.0" encoding="UTF-8" ?>
<SDPSCRequest xmlns="SDP" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
<Version>01.00</Version>
<OperationType>StartBreezeConference</OperationType>
<SessionID>53978</SessionID>
<PayloadSize>1</PayloadSize>
<PayloadEntry>
<MemberID>1</MemberID>
<MemberPhone>4158322000</MemberPhone>
<MemberType>Driver</MemberType>
  <MemberName>Vijay</MemberName>
  <EmailID>vjaiy@hp.com</EmailID>
</PayloadEntry>
</SDPSCRequest>
");
```

6. Conclusion

The combination of HP's networking expertise and infrastructure innovation with Adobe's more than one million strong Flash developer community presents developers and carriers alike with compelling opportunities.

Flash developers now have a wealth of opportunity to apply their expertise in rich applications and interfaces to the communications industry—creating new experiences that make communications more useful, usable, and desirable to end users.

Carriers now have a path for delivering a next generation of communication services that is more efficient, effective, and profitable. They can create new service combinations and experiences utilizing existing network infrastructure and assets, minimizing new capital expenditures. And they can realize development efficiencies of 25% or more (compared with using other development platforms), allowing them to get new service offerings to market faster. Finally, by combining multiple heterogeneous services in rich homogeneous interfaces that are consistent and portable across platforms and devices, carriers can cultivate a more engaged and loyal user base that will, in turn, generate more revenue.

With HP and Adobe, the next generation in communications is now.

7. References

1. *HP Service Delivery Platform, Built on Industry Standards*—a white paper from HP Network and Service Provider Solutions
2. *HP Service Delivery Platform*—a solution brief from HP
3. *The Flash Platform—Delivering an effective user experience across browsers, operating systems, and devices*—a white paper from Adobe

FOR MORE INFORMATION

For more information, visit
www.adobe.com.

Better by Adobe™

Adobe Systems Incorporated
345 Park Avenue, San Jose, CA 95110-2704 USA
www.adobe.com

Adobe, the Adobe logo, Breeze, Flash, Flash Lite, Flex, Macromedia, and "Better by Adobe" are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. HP is a registered trademark of Hewlett-Packard Company. Windows is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

© 2006 Adobe Systems Incorporated. All rights reserved.
Printed in the USA.

