Mobile Services (ST 2010)

Chapter 5: Mobile Web

Axel Küpper

Service-centric Networking
Deutsche Telekom Laboratories, TU Berlin
5 Mobile Web

5.1 WAP Architecture and Protocols
5.2 From HTML to XHTML-MP
5.3 Content Adaptation
5.4 Push Services
5.1 WAP Architecture and Protocols

Overview

WAP Forum and Open Mobile Alliance

- WAP Forum formed to address the standardization and interoperability issues that emerged.
- Group consisting of representatives from various world wide organizations:
  - Terminal (Phone) Manufacturers
  - Network Operators
  - Systems Developers (Microbrowsers & Operating Systems)
  - Application Developers (WAP Applications)
- World’s Experts in:
  - Hardware, Software, Data Networks, Security & Future Internet visionaries
- Adopted by OpenMobileAlliance (OMA)
  - OMA - http://www.openmobilealliance.org/

Wireless Application Protocol (WAP) 1.0

- Wireless Application Environment
  - Wireless Markup Language (WML)
  - WML Script
  - Wireless Telephony Application
  - ...
- Wireless Protocols
  - Wireless Session Protocol
  - Wireless Transport Layer Security Protocol
  - Wireless Transaction Protocol
  - Wireless Datagram Protocol
- WAP 1.0 suffered from a bad user experience and less compatibility with existing Web standards

WAP2.0

- Standardized since 2000 and supported by all mobile devices since 2003
- Main elements: XHTML-MP, HTTP-MP, TCP-MP
5.1 WAP Architecture and Protocols
The "Fixed Web"

Fixed Web

- Architecture for a very flexible and powerful provisioning of content and other functions in a world-wide distributed system
- Initially designed for fixed access only

Web standards

- **Standard naming model**: all servers and content are named with Uniform Resource Locators
- **Content typing**: all content on the Web is given a specific type thereby allowing web browsers to process it correctly
- **Standard content formats**: all web browsers support standard content formats like markup languages (HTML), scripting languages (JavaScript), and a large number of other formats
- **Standard protocols**: networking protocols allow interworking between any web browser and any server
- Most commonly used protocol is the HyperText Transfer Protocol (HTTP), operated on top of the TCP/IP protocol suite
5.1 WAP Architecture and Protocols

The "Mobile Web"

WAP Model

- Optimizations and extensions of the original Web model for matching the characteristics of the mobile environment
- Wherever possible, existing standards have been adopted or have been used as a starting point for the WAP technology
- **Push**: content is not explicitly requested (pull), but delivered on the occurrence of predefined events
- **Wireless Telephony Application**: integration of telephone functions, e.g. call control functions
- **Standard naming model**: Web standard URLs and additional standard URIs for identifying local resources in the device
- **Standard typing**: WAP content of a specific type is consistent with WWW typing
- **Standard content formats**: based on Web technology and includes display markup, calendar information, electronic business card objects, and scripting language
- **Standard protocols**: pull (traditional Web access) and push protocols
5.1 WAP Architecture and Protocols

Proxies

- WAP utilizes proxy technology to optimize and enhance the connection between the wireless domain and the Web.
- WAP proxy allows content and applications to be hosted on standard Web servers and to be developed using proven Web technologies.
- **Protocol Gateway**: translates protocols from a wireless protocol stack (e.g., WAP 1.x) to the Web protocols.

**Content Encoders and Decoders**: translate WAP content into a compact format that allows for better utilization of the underlying link due to its reduced size.

**User Agent Profile Management**: describes capabilities of the respective mobile device and personal preferences.

**Caching proxy**: improves perceived performance and network utilization by maintaining a cache of frequently accessed resources.
5.1 WAP Architecture and Protocols

Supporting Servers

Supporting Services
- WAP Architecture includes supporting, which provide auxiliary functions to devices, proxies, and servers.
- Supporting services are specific in function, but of general use to a wide variety of applications.

Examples
- **PKI Portal**: allows devices to initiate the creation of new public key certificates.
- **UAProf Server**: allows applications to retrieve the UE capabilities and personal profiles of user agents and individual users.
- **Provisioning Server**: supports the automatic configuration of devices, for example, w.r.t. network parameters, with minimal or no user interaction.
5.1 WAP Architecture and Protocols

WAP Scenario

- WAP clients communicate with Web servers through a number of different proxies or directly.
- WAP clients support a proxy selection mechanism for utilizing the most appropriate proxy for a given service or to connect directly to that service if necessary.

Proxies may be located in a number of places, including...
- Mobile network operators in order to provide feature enhancements coupled to the mobile network (e.g., telephony, localization, and provisioning).
- Independent service providers to optimize the communication between device and application server (e.g., protocol translation and cookie caching).
- Secure network to provide a secure channel between UE and the secure network.
### 5.1 WAP Architecture and Protocols

#### Protocol Stacks (I)

**WAP 1.x Gateway**
- Protocol stack of the original WAP architecture
- WAP Gateway converts between datagram-based protocol and connection-oriented protocols

**WAP HTTP Proxy with Profiled TCP/HTTP**
- Widely used in the Internet for ordinary web access, multimedia data, etc.
- Like the WAP gateway, proxy is located between wireline and wireless networks to enhance performance by using the wireless profiles of TCP and HTTP
- Wireless profiled versions are interoperable with TCP and HTTP
5.1 WAP Architecture and Protocols

Protocol Stacks (II)

WAP Proxy Support for TLS Tunneling
- Proxy is used for establishing a connection-oriented tunnel to the web server
- Allows TLS to provide end-to-end security between UE and origin server
- Widely used for enabling end-to-end security in E-Commerce scenarios

Direct Access
- UE directly accesses a Web server via the Internet
- Enables bearer level security (e.g., IPSec)
- Wireless optimizations as defined by the wireless profiles for TCP and HTTP are not available
5.2 From HTML to XHTML-MP
Different Appearances of the Same Content
5.2 From HTML to XHTML-MP

**HTML**

HyperText Markup Language (HTML)

- Predominant markup language for web pages
- Enables the creation of structured documents by marking text such as headings, paragraphs, lists, links, quotes, and other items by HTML elements
- Provides elements for including images and objects as well as for the creation of interactive forms
- Allows for the loading and execution of scripts (JavaScript) that affect the behavior of HTML web pages
- May include Cascading Style Sheets (CSSs) to define the appearance and layout of text and other material
- Originally created by Tim Berners-Lee as an application of SGML
- Meanwhile standardized by W3C

**Example**

```html
<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

**Most common form of an HTML element**

```html
<tag attribute1="value1" attribute2="value2">content to be rendered</tag>
```

**W3C**

- World Wide Web Consortium
- Maintains Web standards and specifications
- [http://www.w3.org/](http://www.w3.org/)
5.2 From HTML to XHTML-MP

XHTML

eXtensible HyperText Markup Language (XHTML)

- Intersection of HTML and XML, a more restrictive subset of SGML
- Stricter than HTML – all tags (including empty elements) must be closed and in lowercase
- Because XHTML documents need to be well-formed they can be parsed using standard XML parsers (unlike HTML, which requires dedicated parsers)

Tags and attributes must be in lowercase

```html
<p ID="p1">XHTML Tutorial paragraph 1</p>
```

Tags must be closed

```html
<p>XHTML Tutorial paragraph 1
<br/>
</p>
```

Elements must be properly nested

```html
<p><b>XHTML Tutorial paragraph 1</b></p>
```

Attributes cannot be minimized

```html
<select multiple>
  <option>XHTML Tutorial Part 1</option>
  <option selected>XHTML Tutorial Part 2</option>
  <option>XHTML Tutorial Part 3</option>
</select>
```

Attribute values must be enclosed within quotes

```html
<p id="p1">XHTML Tutorial paragraph 1</p>
```
5.2 From HTML to XHTML-MP

XHTML Basic

- Cut-down version of XHTML suitable for devices with limited processing and display capabilities
- Looks like (X)HTML, but without many features that are hard to support on mobile devices
  - Frames
  - Nested tables
  - Internal Cascading Style Sheets (CSSs)
  - Complex forms and scripting
- Defined by W3C

<table>
<thead>
<tr>
<th>Module</th>
<th>Elements included</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>&lt;body&gt;, &lt;head&gt;, &lt;html&gt;, &lt;title&gt;</td>
</tr>
<tr>
<td>Text</td>
<td>&lt;abbr&gt;, &lt;acronym&gt;, &lt;address&gt;, &lt;blockquote&gt;, &lt;br&gt;, &lt;cite&gt;, &lt;code&gt;, &lt;dfn&gt;, &lt;div&gt;, &lt;em&gt;, &lt;h1&gt;, &lt;h2&gt;, &lt;h3&gt;, &lt;h4&gt;, &lt;h5&gt;, &lt;h6&gt;, &lt;kbd&gt;, &lt;p&gt;, &lt;pre&gt;, &lt;q&gt;, &lt;samp&gt;, &lt;span&gt;, &lt;strong&gt;, &lt;var&gt;</td>
</tr>
<tr>
<td>Hypertext</td>
<td>&lt;a&gt;</td>
</tr>
<tr>
<td>List</td>
<td>&lt;dl&gt;, &lt;dt&gt;, &lt;dd&gt;, &lt;ol&gt;, &lt;ul&gt;, &lt;li&gt;</td>
</tr>
<tr>
<td>Basic Forms</td>
<td>&lt;form&gt;, &lt;input&gt;, &lt;label&gt;, &lt;select&gt;, &lt;option&gt;, &lt;textarea&gt;</td>
</tr>
<tr>
<td>Basic Tables</td>
<td>&lt;caption&gt;, &lt;table&gt;, &lt;td&gt;, &lt;tr&gt;, &lt;th&gt;</td>
</tr>
<tr>
<td>Image</td>
<td>&lt;img&gt;</td>
</tr>
<tr>
<td>Object</td>
<td>&lt;object&gt;, &lt;param&gt;</td>
</tr>
<tr>
<td>Meta Information</td>
<td>&lt;meta&gt;</td>
</tr>
<tr>
<td>Link</td>
<td>&lt;link&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>&lt;base&gt;</td>
</tr>
</tbody>
</table>
5.2 From HTML to XHTML-MP

XHTML Mobile Profile (XHTML MP)

- Extends XHTML Basic to bring enhanced functions, including additional presentation elements and support for internal style sheets
- Part of WAP 2.0 and standardized by OMA (Open Mobile Alliance)

<table>
<thead>
<tr>
<th>Module</th>
<th>Elements included</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>&lt;body&gt;, &lt;head&gt;, &lt;html&gt;, &lt;title&gt;</td>
</tr>
<tr>
<td>Text</td>
<td>&lt;abbr&gt;, &lt;acronym&gt;, &lt;address&gt;, &lt;blockquote&gt;, &lt;br&gt;, &lt;cite&gt;, &lt;code&gt;, &lt;dfn&gt;, &lt;div&gt;, &lt;em&gt;, &lt;h1&gt;, &lt;h2&gt;, &lt;h3&gt;, &lt;h4&gt;, &lt;h5&gt;, &lt;h6&gt;, &lt;kbd&gt;, &lt;p&gt;, &lt;pre&gt;, &lt;q&gt;, &lt;samp&gt;, &lt;span&gt;, &lt;strong&gt;, &lt;var&gt;</td>
</tr>
<tr>
<td>Presentation</td>
<td>&lt;b&gt;, &lt;big&gt;, &lt;hr&gt;, &lt;i&gt;, &lt;small&gt;</td>
</tr>
<tr>
<td>Style Sheet</td>
<td>&lt;style&gt; element and style attribute</td>
</tr>
<tr>
<td>Hypertext</td>
<td>&lt;a&gt;</td>
</tr>
<tr>
<td>List</td>
<td>&lt;dl&gt;, &lt;dt&gt;, &lt;dd&gt;, &lt;ol&gt;, &lt;ul&gt;, &lt;li&gt;</td>
</tr>
<tr>
<td>Basic Forms</td>
<td>&lt;form&gt;, &lt;input&gt;, &lt;label&gt;, &lt;select&gt;, &lt;option&gt;, &lt;textarea&gt;, &lt;fieldset&gt;, &lt;optgroup&gt;</td>
</tr>
<tr>
<td>Basic Tables</td>
<td>&lt;caption&gt;, &lt;table&gt;, &lt;td&gt;, &lt;tr&gt;, &lt;th&gt;</td>
</tr>
<tr>
<td>Image</td>
<td>&lt;img&gt;</td>
</tr>
<tr>
<td>Object</td>
<td>&lt;object&gt;, &lt;param&gt;</td>
</tr>
<tr>
<td>Meta Information</td>
<td>&lt;meta&gt;</td>
</tr>
<tr>
<td>Link</td>
<td>&lt;link&gt;</td>
</tr>
<tr>
<td>Base</td>
<td>&lt;base&gt;</td>
</tr>
<tr>
<td>Legacy</td>
<td>&lt;start&gt; attribute on &lt;ol&gt;, value attribute on &lt;li&gt;</td>
</tr>
</tbody>
</table>

Extensions compared to XHTML Basic
5.2 From HTML to XHTML-MP

Evolution of Wireless Markup Languages

© by David Höffer, 2007
5.2 From HTML to XHTML-MP
Cascading Style Sheets (I)

Cascading Style Sheets (CSS)
- Way to define the visual appearance of different (X)HTML elements
- Maybe used to specify the size, color, and position of standard (X)HTML elements like <body>, <p>, <td> etc. as well as sections that may be defined using a name or id
- Preferred way of formatting well-structured (X)HTML code, because it facilitates having standard design throughout the website

CSS1
- Published by W3C in December 1996

CSS2
- Published by W3C in May 1998
- Additional features: absolute, relative, fixed positions, new font properties,...

CSS3
- Still subject to standardization
- Additional features: multi-column layout, box radius, multiple backgrounds, border images,...
- Will look the same across all borders

WAP CSS (WCSS)
- Used in XHTML MP
- Subset of CSS2 plus some WAP-specific extensions

(c) Hicksdesign, http://www.hicksdesign.co.uk/boxmodel/
5.2 From HTML to XHTML-MP
Cascading Style Sheets (II)

Multiple CSS sources

- CSS style information can be either attached as a separate document or embedded in the HTML document.
- Multiple style sheets can be imported.
- Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.

Priority scheme for CSS sources (from highest to lowest priority):

- Author styles (style information provided by the web page author), in the form of
  - Inline styles, inside the HTML document, style information on a single element, specified using the "style" attribute.
  - Embedded style, blocks of CSS information inside the HTML document itself
  - External style sheets, i.e. a separate CSS-file referenced from the document
- User style
  - Local CSS-file specified by the user using options in the web browser, and acting as an override, to be applied to all documents.
- User agent style
  - Default style sheet applied by the user agent, e.g. the browser's default presentation of elements
5.2 From HTML to XHTML-MP
Cascading Style Sheets (III)

Without CSS:

An Example

Item 1
Item 2

With CSS:

An Example

Item 1
Item 2

example.html:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM/DTD XHTML Mobile 1.0//EN"
 "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Example"</title>
<link rel="stylesheet" href="mobile.css" type=text/css" />
</head>

<body>
<h1>An Example</h1>
<p>Item 1</p>
<p>Item 2</p>
</body>
</html>
```

mobile.css:

```css
/* Mobile Style Sheet */

h1 {
  font-size:xlarge;
  color:#4040ff;
  text-align:center;
  text-decoration:underline
}

p {
  border:1px #6060ff solid;
  background:#f0f0ff;
  text-align:center;
  font-size:medium;
  padding:4px
}

a {
  color:black
}

body {
  background:#c0c0ff
}
```
5.3 Content Adaption
Strategies for Adapting a "Fixed Web Site"

Going Mobile with Your Web Site
- Different options for going mobile with a web site initially designed for the Fixed Web
- Do nothing: just leave the site as it is and let the user's browser render it
- Remove formatting: simplify the site so that it loads faster and uses less bandwidth, but leave the design the same
- CSS-based design: use a different CSS file for mobile visitors and define mobile optimized formatting of various page elements through this CSS
- Create a new site: develop a version of your site tailored for mobile browsers and people on the move
5.3 Content Adaption

Device Variety

Problem Statement

- No device homogeneity – mobile devices significantly vary in a number of features

  - Hardware
    - Screen size
    - Number of colors
    - Type of keypad
    - Number of softkeys
    - ...

  - Software and operating system
    - Type of browser
    - Java support
    - WML, HTML, XHTML support
    - CSS support
    - JavaScript/Ajax support
    - ...

- How to design a mobile web site that provides all users a best possible experience depending on the type of device they use
5.3 Content Adaption

Adaption versus Lowest Common Denominator

Adaptation/Multiserving
- Provides the content tailored to the capabilities of the respective device
- Techniques used for adaptation
  - Detection
  - Redirection
  - Setting correct MIME types
  - Changing links
  - Removing/scaling graphics

Lowest Common Denominator (LCD)
- Find the lowest common denominator of the capabilities of all target devices one wants to address with the content
- Works reasonable well for all devices
- Devices with better capabilities than LCD will see a version of the content that may not be very beautiful

How to Adapt?
- Design alternative CSS
  - Controls the display of elements and images
  - Easiest method
- Create multiple versions of pages
  - Re-direct the user to a device-specific version (alteration)
  - Used to get the most control over what is shown to each device
  - Most complex method
- Automatic adaptation
  - Create content in one format and use a tool to generate device-specific versions
  - Most elegant method
5.3 Content Adaption

Progressive Enhancement and Device Plans

Progressive Enhancement

- Method of using web techniques in a layered fashion to allow anyone with any web browser to access the content, regardless of its capabilities
- Start with encoding the content with a markup language that is all devices can manage (lowest common denominator)
- Next step: add basic styling techniques
- Continue to add layers until the best possible user experience is achieved

Device plans

- Useful method: create a device plan at the very start, which encompasses all devices that should be supported
- Class with the most limited capabilities represents the layer to start progressive enhancement

<table>
<thead>
<tr>
<th>Class</th>
<th>Markup</th>
<th>CSS</th>
<th>JavaScript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>XHTML, XHTML-MP, HTML5</td>
<td>CSS2, CSS3</td>
<td>Great, includes DHTML, Ajax</td>
</tr>
<tr>
<td>Class B</td>
<td>XHTML, XHTML-MP</td>
<td>CSS2 (Decent)</td>
<td>Limited, some DHTML</td>
</tr>
<tr>
<td>Class C</td>
<td>XHTML, XHTML-MP</td>
<td>CSS2 (Limited)</td>
<td>Limited</td>
</tr>
<tr>
<td>Class D</td>
<td>XHTML-MP</td>
<td>CSS2 (Basic)</td>
<td>None</td>
</tr>
<tr>
<td>Class E</td>
<td>XHTML-MP, WML</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
5.3 Content Adaption

HTTP Content Negotiation and User Agent Detection

HTTP Content Negotiation

- HTTP header includes content negotiation mechanisms where the browser indicates what type of information it can accept
- Server decides what type of information to return based on the information received from the browser
- Accept header of HTTP fixes character set, content codings, and natural language

User Agent Detection

- In addition to ACCEPT headers, clients send a User-Agent header to identify themselves
- Contains information about the browser, the operating system making the request, and hardware information

<table>
<thead>
<tr>
<th>Accept header example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept: audio/*; q=0.2, audio/midi</td>
<td>I prefer audio/midi, but send me any audio type if midi is not available.</td>
</tr>
<tr>
<td>Accept-Charset: utf-8; q=0.9, iso-8859-1; q=0.6</td>
<td>I prefer utf-8 character set, but you can send me also iso-8859-1 character set.</td>
</tr>
<tr>
<td>Accept-Encoding: identity, gzip, *q=0</td>
<td>I can handle content without encoding and content encoded with GNU zip. Do not send other content-codings.</td>
</tr>
<tr>
<td>Accept-Language: en, fi; q=0.6</td>
<td>I prefer English content, but send me Finnish content if English is not available.</td>
</tr>
</tbody>
</table>

(C) Content Adaptation, MediaLab, TeliaSonera
5.3 Content Adaption
CC/PP and UAProf

Device Capabilities and Preference Profile

- Growing number of devices connected to the Internet requires to tailor content to the capabilities of the respective device
- CC/PP standardized by W3C is a standard way for publishing configuration details and capabilities to servers
- Extensible framework for communicating the delivery context (screen size, audio capabilities, bandwidth, etc.) from a device to a web server, resulting in the delivery of web content that is usable on a given device

User Agent Profile (UAProf) by OMA

- Concrete CC/PP vocabulary dedicated to mobile phone descriptions
- Content servers, gateways, and proxies can use this information and optimize the content for the device and the user
- Supported attributes:
  - Hardware platform – screen size, audio capabilities, color capabilities, ...
  - Software platform – operating system, mime types, character sets, encoders, ...
  - Network characteristics – GSM/GPRS capability, security support, Bluetooth, ...
  - Browser characteristics – browser info, HTML/XHTML, Java, JavaScript, frames, tables, ...
  - Push characteristics – push content types, push message size, ...
5.3 Content Adaption

UAProf Scenario

- UAProf files tend to grow large
- Common practice: only the URL of the device profile is transmitted from the mobile device to the content server
- Content server fetches the profile from a device profile repository and may store it in its own database for later use
- WAP gateways and HTTP proxies must support UAProf header forwarding
5.3 Content Adaption
UAProf Example

User Agent Profile for the Nokia 6230 (excerpt):

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:ID="Nokia6230">
    <prf:component>
      <rdf:Description rdf:ID="HardwarePlatform">
        <prf:BitsPerPixel>16</prf:BitsPerPixel>
        <prf:ColorCapable>Yes</prf:ColorCapable>
        <prf:ImageCapable>Yes</prf:ImageCapable>
        <prf:Keyboard>PhoneKeypad</prf:Keyboard>
        <prf:Model>6230</prf:Model>
        <prf:NumberOfSoftKeys>3</prf:NumberOfSoftKeys>
        <prf:PixelAspectRatio>1x1</prf:PixelAspectRatio>
        <prf:ScreenSize>128x128</prf:ScreenSize>
        <prf:ScreenSizeChar>18x5</prf:ScreenSizeChar>
        <prf:StandardFontProportional>Yes</prf:StandardFontProportional>
        <prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
        <prf:TextInputCapable>Yes</prf:TextInputCapable>
        <prf:Vendor>Nokia</prf:Vendor>
        <prf:VoiceInputCapable>Yes</prf:VoiceInputCapable>
      </rdf:Description>
    </prf:component>
    <prf:component>
      <rdf:Description rdf:ID="SoftwarePlatform">
        <prf:AcceptDownloadableSoftware>Yes</prf:AcceptDownloadableSoftware>
        <prf:AudioInputEncoder>
          <rdf:Bag>
            <li>AMR</li>
          </rdf:Bag>
        </prf:AudioInputEncoder>
      </rdf:Description>
    </prf:component>
  </rdf:Description>
</rdf:RDF>
```
5.3 Content Adaption

WURFL

Wireless Universal Resource File (WURFL)

- XML configuration file which contains information and features of many mobile devices
- Open source and intended for developers working with WAP and Wireless
- Includes (and corrects) information from UAPref repositories maintained by the manufacturers
- Device type is derived from the USER AGENT header sent by the browser
- WURFL API is available in many programming languages, including Java, PHP, .Net, Ruby, and Python

[Image of Tera-WURFL Explorer]

[Image of WURFL API example]

http://www.tera-wurfl.com/explore/
http://wurfl.sourceforge.net/
5.4 Push Services

Pull versus Push

Pull model
- Client requests a service or information from a server which then responds in transmitting information to the client
- Example: browsing the Web, where the client enters a URL (the requests) and the server answers by sending a Web page (the response)

Push model
- Also based on the client/server model
- No explicit request from the client before the server transmits its content

OMA Push Framework
- Introduces a means to transmit information to a device without a user request
- Enables server-to-client content delivery via ...
  - ... various communication methods (protocols)
  - ... interconnection types (point-to-point, point-to-multipoint)
  - ... transport bearers (underlying protocols and bearer networks)
5.4 Push Services

Push Components

Push Proxy Gateway (PPG)
- Proxy for push operations initiated by external applications
- Gateway for converting external application and interface operations for delivery via Push Over-the-Air (OtA)
- Push client registration
- Push client address translation
- Push content transformation
- Store-and-forward support for push

Push Initiator (PI)
- Triggers the delivery of push content
- Transmits the push content and delivery instructions to a PPG

Push Client
- Push software (daemon) at the mobile device permanently listening for incoming push messages
- Receives content and delivers it to the respective client application

Multicast/Broadcast Push Delivery via Push-OTA Protocol

Unicast Push Delivery via Push-OTA Protocol

Push Request via Push Access Protocol (PAP)

Content provider as Push Initiator
5.4 Push Services

Delivery of a Push Message

1. PI sends a request for delivery of a new or replacement of a prior message
   - Indicates one or several target users and a client application to whom the message is delivered
   - PI may select various options for message delivery
2. PPG sends a confirmation or a reject
3. PPG delivers push message to the push client via Push-OTA protocol
4. Push client confirms delivery
5. PI is notified about the successful delivery of the push message
6. Push client passes the content to the client application (browser, multimedia messaging client, instant messaging client, email client,...)
5.4 Push Services

Delivery of a Push Message from the PI to the PPG

- Push message is sent as an HTTP POST to the Push Proxy Gateway
- POST is a multipart XML document, with the first part being the PAP (Push Access Protocol) Section and the second part being either a Service Indication or a Service Load.

**POST**

- Contains at a minimum the URL being posted to (this is not standard across different PPG vendors), and the content type

**PAP**

- PAP XML contains at the minimum, a `<pap>` element, a `<push-message>` element, and an `<address>` element

**Service Indication**

- PUSH Service Indication (SI) contains at a minimum an `<si>` element and a `<indication>` element.
5.4 Push Services

Delivery of a Push Message from the PPG to the Client

PPG Delivery to Mobile Station

- Once a PUSH Message is received from the Push Initiator, the PPG has two options for delivery:

Connectionless PUSH

- PPG has a local SMS queuing mechanism running locally, which is connected to the operator's SMSC
- Since a WAP PUSH Message can be larger than a single SMS message allows for, the message may be broken up into multiple SMS messages

Connection-oriented PUSH

- Gateway is aware of the handsets IP Address
- If the gateway is unable to determine the IP Address of the handset, or is unable to connect to the device, the push notification will be encoded and sent as an SMS.
- Problem: devices while registered to the network, may not have a valid PDP context

Other PUSH Attributes

- Push notifications can be confirmed or unconfirmed.
- Push notifications can be set to expire if not delivered before a certain time.
5.4 Push Services

End-to-End Push Framework

- Client device
  - Native (Embedded) Push-Enabled Services
    - Browser (wmLua)
    - MMS Client (mms.ua)
    - IM Client (wv.ua)
    - Email Client (email.ua)
    - Data Sync Client (syncml.ua)
    - DM Client (syncml.dm)
    - etc...
  - Extended Push-Enabled Services
    - Web Runtime Environment (Widget user agent)
- OMA Push Client
- Push-OTA (Over-the-Air) Protocol
  - OTA-PTM/CBS
  - OTA-WSP/SMS (WAP1 Push)
  - OTA-HTTP/TCP (WAP2 Push)
  - OTA-WSP/(WDP/MTP) (WAP1 Push)
  - OTA-SIP (SIP Push) (SIP MESSAGE or SIP INVITE+MSRP)
  - OTA-PTM/MBMS
- Access control white list
- OMA Push Proxy Gateway
- Push Access Protocol (PAP)
- Push Initiator / Push-Enabled Services
  - Web Server, MMSC, IM Server, Email Server, Sync Server, DM Server, etc.
- Service Personalization and Contextualization Interfaces
  - UAPProf Server
  - DPE Server
  - Location Server
  - Group Mgmt Server
  - Sub Profile Server
- OMA Push Architecture Entities
  - Push-Enabled Applications/Services
  - Interface defined by OMA Push
  - Interface not defined by OMA Push

Web Runtime Environment (Widget user agent)