

# Adapting Email Functionality for Mobile Terminals

Jon-Finngard Moe<sup>1</sup>, Eivind Sivertsen<sup>1</sup>, and Do Van Thanh<sup>2</sup>

<sup>1</sup>Norwegian University of Science and Technology  
7491 Trondheim Norway

{jonfinng, eivindsi}@stud.ntnu.no

<sup>2</sup>Telenor R&D

1331 Fornebu, Norway

Thanh-van.Do@telenor.com

**Abstract.** There is an increasing need for mobile users to access email services anytime and anywhere. The solutions proposed so far, like Web/WAP [6] mail and VPN [7] tunneling are not ideal for mobile clients with limited resources. These solutions do not offer any new concepts to how the emails are transferred or the mail format itself. In this paper we present an XML Web Service [1] solution, which opens for new ways of providing email functionality to mobile as well as stationary systems. The paper starts with a brief summary of how email systems of today work. Ultimately, our solution based on XML Web Services and the proposed XMMAP (XML Mobile eMail Access Protocol) is explained thoroughly.

## 1. Introduction

So far, all attempts to find new killer applications for mobile phones after voice communication and SMS (Short Message Service) have not been very successful. However, there is strong indication that mobile email access may be a candidate. Indeed, for mobile professional users that have used data applications in their work for many years, it is valuable to be able to access their mails when being on the move.

Unfortunately, so far there is no satisfactory solution because of several issues. First, small displays and limited navigation capability are not suitable for displaying a large number of emails. Emails may also be unnecessary large in size due to excessive presentation formatting and large attachments.

Another difficulty is standardization of the user interface. The plurality and heterogeneity of mobile terminal types makes this a complex task.

## 2. Special Requirements for Mobile Terminals

Mobile terminals have many limitations compared to standard PCs that put new requirements on the application and service design for mobile terminals:

- Lower available bandwidth and transfer rate.
- Smaller screen size, less number of colors and resolution.
- Smaller Keyboard size and reduced control equipment.
- Lower processing capability.
- Less support for application protocols.
- Less storage space on device.

### 3. Current Email Infrastructure

Current email communication systems consist of two main components, email Server and email Client. They are both compositions of several elements that are making use of multiple communication protocols, as well as internal service elements. Examples of protocols are SMTP [2] (Simple Mail Transfer Protocol), IMAP [3] (Internet Message Access Protocol), POP [4] (Post Office Protocol).

An SMTP mail object contains an envelope and content. In order to send and/or relay mail, we must follow the protocol described in RFC2821 [2].

Figure 1 shows that a minimum of 11 message transfers is needed between the client and server in order to send a single email when using SMTP. Two extra transfers are introduced for every additional recipient. When using a mobile terminal with slow network connection, each message adds to the delay. It is therefore desirable to keep the number of message transfers to an absolute minimum.

SMTP - mailsending procedure (SMTP Envelope)

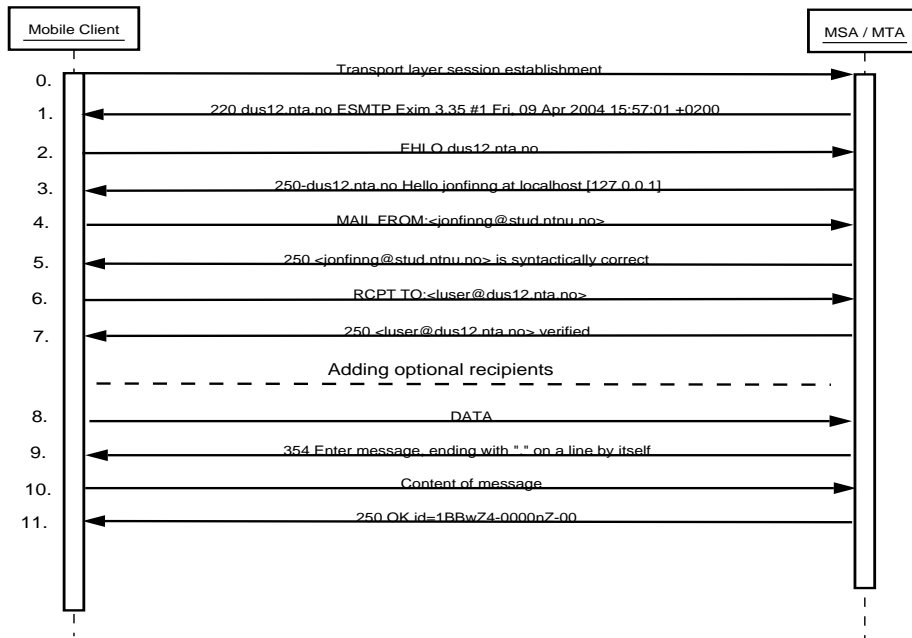


Figure 1 SMTP message transfer

### 3.3 IMF and MIME

IMF and MIME [5] are standards used for representing the actual content of the email. IMF defines which headers are mandatory, and how a standard message should be organized. MIME is an extension to IMF, which defines how complex emails should be represented. These are typically emails that have file-attachments; multiple alternative representations formats or even enclosed sub-email-messages.

An email usually contains a lot of information enclosed in IMF headers. Most of this information is not relevant to the user, such as the list of mail servers the message has visited on its way to its destination, and most "X-headers"<sup>1</sup>. This information is usually not presented in user-agents and introduces a significant amount of overhead<sup>2</sup>.

MIME representations may become very complex, and may take time to process on terminals with limited processing capabilities. To view all parts a MIME encoded message, the end clients must have support for displaying the presentation formats and attachments sent. In cases when the clients lack this support, MIME body parts will end up as unnecessary overhead. Again, adaptations must be done for mobile terminals.

## 4. XML Web Service for Mobile Email Access

Mail servers normally reside inside the corporate network, protected from the Internet by a firewall. To enable email access from mobile phones, the XML Web Service concept is found most suitable due to the ubiquity of the World Wide Web and the ability to traverse firewalls using SOAP [8] (Simple Object Access Protocol). SOAP can run over HTTP and use port 80, which is open on many firewalls facing the Internet.

The most straightforward solution is to expose the whole SMTP and IMAP/POP as Web Services. Each SMTP and IMAP/POP command is mapped to a Web Service method. In effect, each SMTP or IMAP/POP command is encapsulated in a SOAP message and transported to the WS client.

Advantage in this solution compared to Web/WAP solutions is that there are no overhead data specifying the appearance of the content. Compared to VPN, this solution is more robust due to the use of SOAP. SOAP has a more relaxed way to handle sessions. This involves that sessions may survive even if the link is goes down for a period of time.

The disadvantage is the numerous functional requirements that are put on the mobile phone. To access and retrieve mails, the WS client must be capable of understanding *all* the SMTP/IMAP/POP commands and communicating properly with the email Server. It must therefore be equipped both with a MUA (Mail User Agent) with full SMTP support as well as an IMAP/POP client. This functionality is put on top of the SOAP engine.

---

<sup>1</sup> "X-headers" are custom headers for providing extra information about the email. Typically added by clients, virus and spam-scanners.

<sup>2</sup> In an example email we were able to strip the header size of an email from 2351 to 323 bytes removing unnecessary header entries.

Other disadvantages are the high number of interactions and also the high amount of downloaded data that are needed. Each SOAP message adds overhead themselves through their headers, making messages exchanges excessively verbose. This is definitely not suitable in a wireless network with limited bandwidth.

## 5. XMAP Mail Web Service

To overcome the limitations and insufficiencies of the straightforward email XML Web service presented in the previous section and to solve the major problems related to email access from mobile terminals, we introduce a protocol called XML Mobile eMail Access Protocol (XMAP).

As mentioned, a minimum of 11 SMTP messages is required between the client and server just in order to send an email. This is highly undesirable when using a mobile terminal. Additionally, every message transfer introduces unnecessary overhead from underlying protocols. On top of this we have the considerations regarding overhead from unnecessary headers and presentation information.

The major benefit when using XMAP is that it both combines and simplifies the functionality and information given by both the MTA (SMTP), Access Agent (IMAP/POP) and the representation format (SMTP-IMF/XMTP[9]). This is achieved by mapping the information into an XML format and tying different parts of the format to specific requests and responses to SOAP – functions.

Like XMTP, XMAP utilizes the strength of XML to simplify parsing of the received information. The terminal can use an XML-parser for retrieving the information from the XMAP-message as well as constructing new XML documents. This makes implementation of an XMAP-compatible mail client a very simple task compared to a full-scale email client.

XMAP introduces a new concept of messaging. While traditional protocols rely on frequent message transfers with well-defined operations, XMAP is more flexible, making it possible to do most operations in *one* message exchange. It is also possible to define some custom options within the scope of the format.

### 5.1 XMAP Data Format

XMAP is in its basic form a representation of an entire mail account, spanning everything from login credentials to flags in a specific message. This makes it possible to use sub-parts of the XMAP-format for representing different parts of a mail account for different purposes. This is especially useful when utilizing XMAP for invoking functions on the mail server. When invoking a function, only the relevant subparts of the formats are sent, thus avoiding unnecessary overhead. Here follows a standard email message represented in XMAP:

```
<Message
  xmlns='http://www.finngard.org/2004/03/xmap/'
  xmlns:web='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  web:about='mid:1078406317002232@lycos-europe.com'
  <BoxNumber mailbox='INBOX'>12</BoxNumber>
```

```

<Headers>
  <From>johndoe@telenor.com</From>
  <To>&lt;finngard@finngard.org&gt;</To>
  <Subject>XMMAP, suitable for PDA?</Subject>
  <Date>Fri, 2 Mar 2004 12:23:12 -0700</Date>
</Headers>
<Flags protocol='imap'>
  <Seen>1</Seen>
  <Answered>0</Answered>
</Flags>
<Body charset='ISO-8859-1'>
  Hi! Do you suggest using the XMMAP format for PDAs?
</Body>
<Attachments>
  <Attachment
    content-type='application/x-ms-word'
    encoding='base64'>
    <AttachmentNumber>1</AttachmentNumber>
    <Filename>pda_description.doc</Filename>
    <Size>1345</Size>
    <Content>9j/4AAQSkZJRgABAQEAYABgAAD.. </Content>
  </Attachment>
</Attachments>
</Message>

```

XMMAP maps perfectly into a standard SMTP-IMF message, with or without MIME extensions, and vice versa. In addition to this, an XMMAP-message may supply information given by both POP3 and IMAP mail access protocols.

In order to represent an entire account, some additional elements are needed. These are also defined within the XMMAP data format.<sup>3</sup>

## 5.2 XMMAP Message Transfer

The XMMAP data format is useful for representing an account in a minimalist way, and may also be well suited for storage purposes. On the other hand, the data format lacks the coupling to specific functions related to mobile mail access. This coupling is achieved by defining SOAP – methods which receives and responds with messages in XMMAP format. The following set of methods is implemented so far:

SOAP/XMMAP Methods	
LoginMobileTermXMMAP	setFlagXMMAP
LoginProfileXMMAP	getMessagesAsXMMAP
GetHeadersAsXMMAP	sendMailXMMAP
GetMailboxesXMMAP	deleteXMMAP
GetNewHeadersAsXMMAP	logoutXMMAP

**Table 1.** SOAP/XMMAP methods

<sup>3</sup> Due to space limitations, a complete description of each element in XMMAP is not given in this paper.

**Request message example:**

```
<Account>
  <UserName>jon</UserName>
  <PassWord>secret</PassWord>
  <Host>imap.ntnu.no</Host>
  <Protocol>IMAPS</Protocol>
  <Port>443</Port>
</Account>
```

**Response message example:**

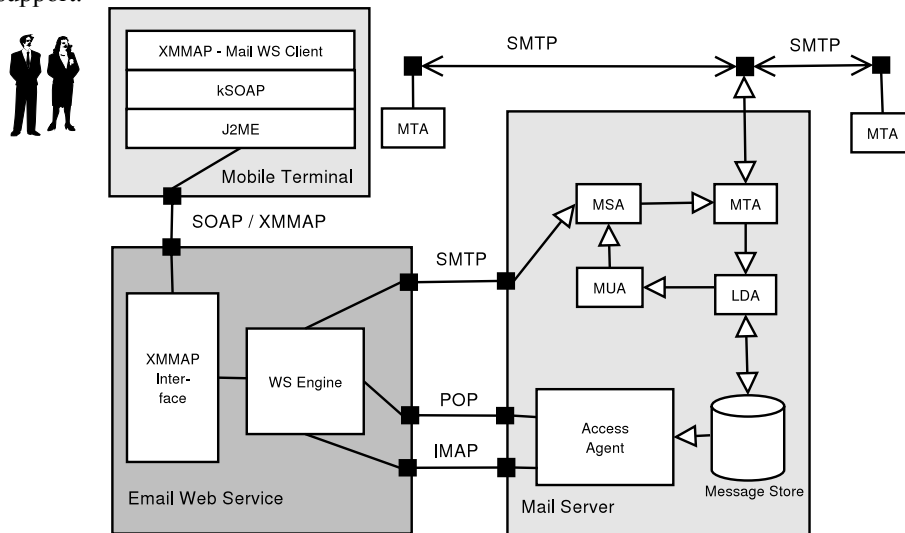
```
<Mailboxes>
  <Mailbox>
    <BoxName>INBOX</BoxName>
    <Unread>2</Unread>
    <Total>23</Total>
  </MailBox>
</Mailboxes>
```

The methods listed in Table 1 are the most important methods that can be implemented by coupling SOAP and XMMAP for email access on mobile terminals. All these methods have well defined requests and responses, all in XMMAP format. Some of the methods have additional SOAP parameters for system specific purposes.

However, the XMMAP format is flexible, and can be extended by new elements in any part of the format if necessary. There is no fixed order in how the current messages are sent after the user has logged in. When defining new methods, this principle should be followed in order to keep every message independent from both previous and succeeding messages.

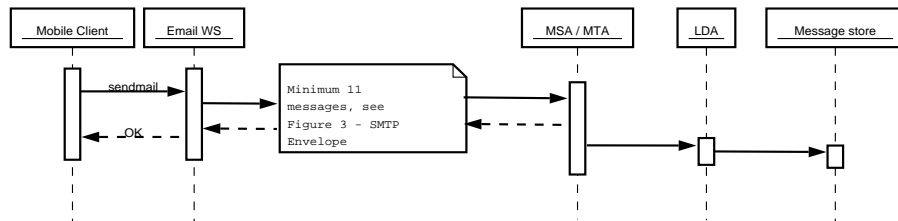
**5.3 XMMAP Web Service Architecture**

As shown in Figure 2, the mobile terminal needs support for SOAP and J2ME[10], and must have a Mail User Agent (MUA) capable of XML document creation and parsing. The MUA is used for sending and receiving XMMAP messages. No support for any other mail protocols is needed. This makes the client implementation much simpler compared to other solutions that require full SMTP and Access Agent support.



**Figure 2** Web Service Architecture

Figure 3 shows how the message exchanges are performed when sending a mail using the proposed architecture and XMMAP.



**Figure 3.** Email messaging using XMAP

#### 5.4 Internal Web Service Functionality

As mentioned earlier, the XML Web Service will adapt the mail messages for mobile telephones and PDA's. In addition to removing unnecessary email headers, alternative representations of the same content (e.g. both plain text and HTML version of the message body) are reduced to one. Attachments are by default kept back until the user specifically asks to get them. These actions keep the amount of data transmitted on a minimum, which is important when having low bandwidth. A summary of the internal functionality offered of the XML Web Service (see Figure 2):

- SMTP-IMF to XMAP gateway and vice versa.
- IMAP/POP to XMAP gateway and vice versa.
- User authentication.
- User profile management.
- Session management for all interfaces and related active connections.
- Content adaptation. This includes everything from tag and attachment stripping to picture resizing.
- Optional: Local caching of messages and attachments.
- Provide necessary interfaces.

## 6. Conclusion

In this paper, we propose a solution that enables mail access from mobile terminals based on XML Web Service concept and a protocol called XMAP (XML Mobile eMail Access Protocol). Such a solution has several advantages as follows:

- Our solution gives fewer message transfers between client and server, and less interaction gives better performance for low-bandwidth devices.
- Since only the message related information is delivered to the client, messages can be stored on the device. This cannot be done as easy and well arranged when accessing email accounts via e.g. webmail.
- The content is automatically adapted to fit the terminal by using information sent by the client software about display dimensions and color support etc. Unnecessary information is stripped away by the web service before replying to the client.

- Session timeout is a problem when working with services requiring authentication. Especially when accessing the Internet via a GPRS (General Packet Radio Service) network, the GPRS/Internet gateways tend to have short timeout periods. Reestablishment after a timeout often implies assignment of a different IP address, making all session on higher layers invalid. This problem is solved when using SOAP sessions, ensuring session mobility.
- Firewalls are no longer an issue, since all mail traffic can pass through SOAP messages.<sup>4</sup>

The combined use of an XML web service and XMMAP makes a simple and flexible email solution for mobile users. XMMAP allows for a minimal client footprint and uses less bandwidth than other solutions.

## 7. References

1. XML Web Service. <http://www.w3.org/2002/ws/>
2. SMTP Simple Mail Transfer Protocol <http://www.ietf.org/rfc/rfc2822.txt>
3. IMAP Internet Mail Access Protocol <http://www.ietf.org/rfc/rfc2060.txt>
4. POP Post Office Protocol <http://www.ietf.org/rfc/rfc1939.txt>
5. MIME Multipurpose Internet Mail Extensions <http://www.ietf.org/rfc/rfc1521.txt>
6. WAP Specifications <http://www.wapforum.org/what/technical.htm>
7. VPN Virtual Private Networks Standards <http://www.vpnc.org/vpn-standards.html>
8. SOAP Specification. <http://www.w3.org/TR/soap/>
9. XMTP, XML MIME Transformation Protocol. <http://www.openhealth.org/documents/xmtp.htm>
10. J2ME Java 2 Micro Edition <http://java.sun.com/j2me>
11. Sivertsen, E., Moe, J-F., Søvik, A: Master Thesis, Institute of Telematics, Norwegian University of Science and Technology: "E-mail Access Using XML Web Services" (2004). <http://xmmmap.nta.no/report.pdf>

---

<sup>4</sup> This is of course only true if the corporate firewall allows global WWW access (when using HTTP as transport protocol).