FD – Mailbox

Content

History

Version	Date	Adjustments
A	2006-10-03	First version. (MANDE)

1 Introduction

Mailbox is a MAS software component responsible for handling the access to subscriber mailboxes. This comprises retrieving, storing and printing messages.

2 Function Structure

2.1 Overview

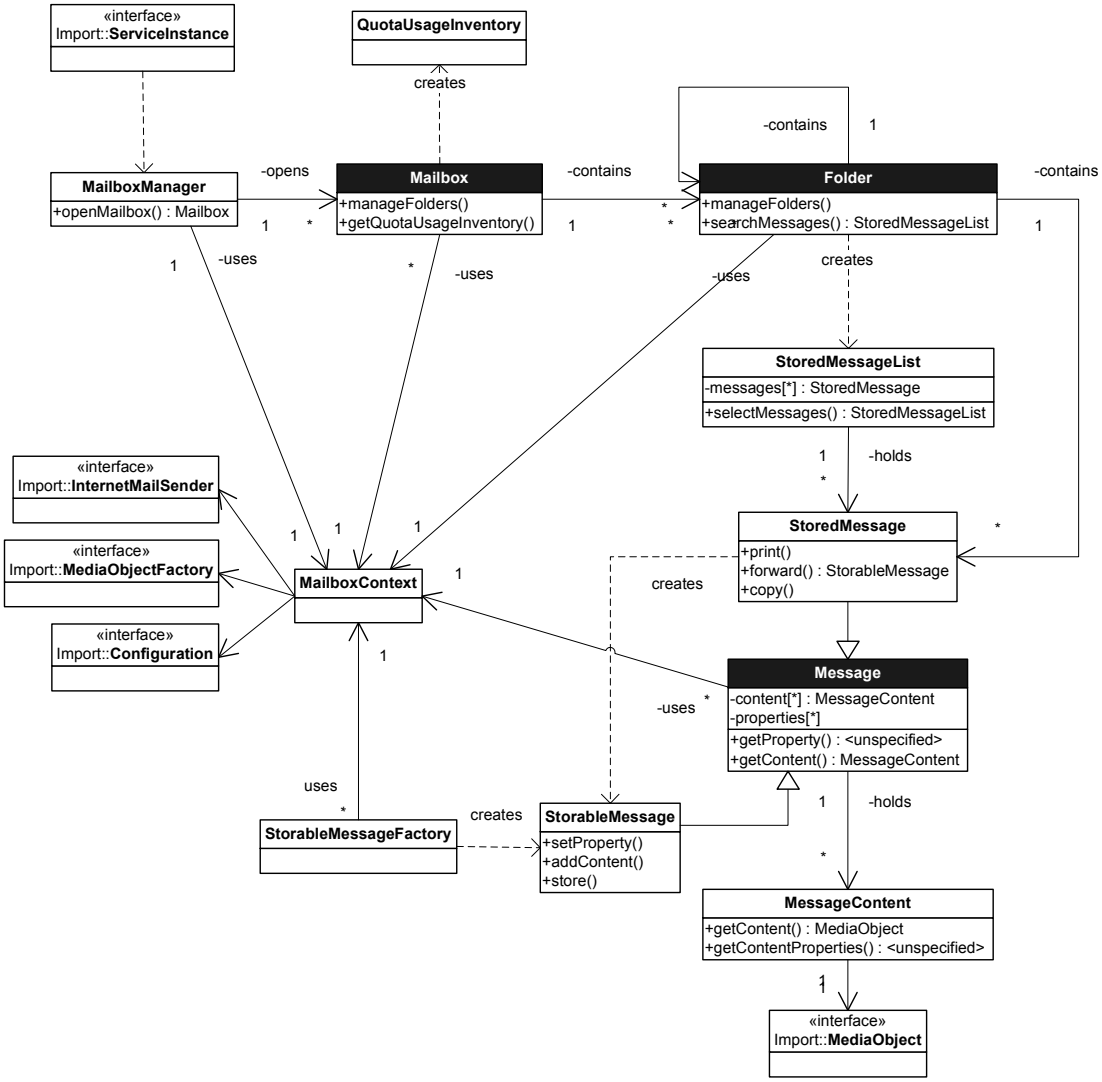

Figure 1 Mailbox component overview

The Mailbox component structure is based on the fundamental business entities **Mailbox**, **Folder** and **Message**. This triad is complemented with classes to accomplish Mailbox and Message management.

2.1.1 MailboxManager

This class uses an IServiceInstance and subscriber account credentials to open and instantiate a Mailbox.

2.1.2 Mailbox

A Mailbox contains folders. The Mailbox has methods to manage folders in the mailbox. Mailbox has also methods to examine the total quota usage (bytes and message count) in the Mailbox.

2.1.3 Folder

A Folder can contain other Folders or Messages. A Folder has methods to manage folders contained in the folder. Folder also provides methods to within the Folder search Messages fulfilling a certain criteria. The messages corresponding to the criteria are put together and ordered in a StoredMessageList.

2.1.4 StoredMessageList

StoredMessageList holds references to a specific set of messages of class StoredMessage. It provides methods to access the messages in the list. StoredMessageList also provides a method to select a subset of messages into a new StoredMessageList.

2.1.5 Message

A Message is an abstract class. It has a number of properties of a basic type and a list of content objects. It provides methods to read the message properties and content.

2.1.6 StoredMessage

The StoredMessage is derived from Message and represents a Message stored in the message storage. It adds methods for copying the message to another Folder, print the message at a specified destination, and forwarding the message by including it in a StorableMessage.

2.1.7 MessageContent

The MessageContent class aggregates a MediaObject and holds content properties in a map. The content properties add semantics, such as description, language and filename, to the MediaObject.

2.1.8 StorableMessage

The StorableMessage is derived from Message and represents a new Message to be stored in the message storage. Along with the read methods it provides methods for setting the message properties and adding content.

2.1.9 StorableMessageFactory

This class is simply a Factory for creating new instances of StorableMessage.

2.1.10 MailboxContext

The MailboxContext provides an environment for class instances in the Mailbox component. MailboxContext has a number of injected references to imported interfaces from other component. It also aggregates an instance of MailboxConfig.

2.1.11 QuotaUsageInventory

Mailbox objects are able to create a snapshot view of the current quota usage in the mailbox. This snapshot view is represented by an instance of the QuotaUsageInventory class.

2.2 JavaMail

The Mailbox component is based on JavaMail in a layered structure. The Mailbox class hierarchy is build upon a JavaMail based abstract layer. In the JavaMail abstract layer abstract methods is used to allocate and consume mail system storage resources. The provider implementation class layer implements the internet standards RFC822, MIME and IMAP.

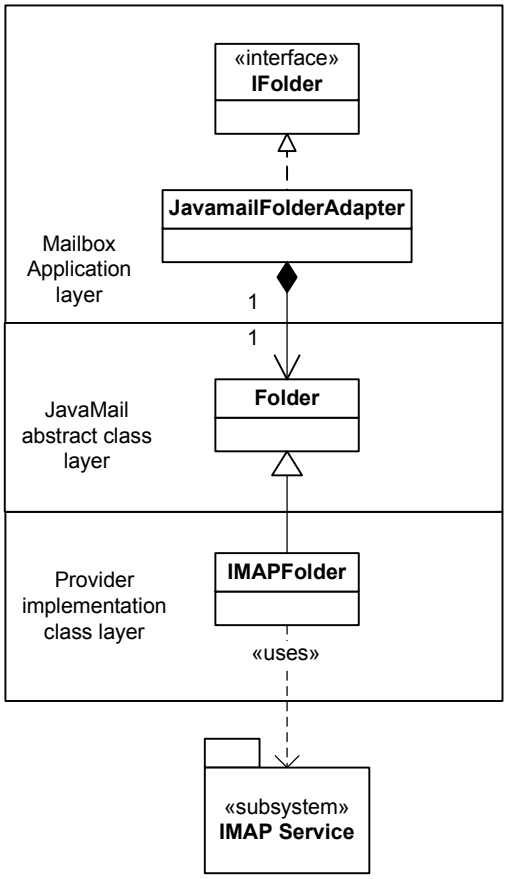

Figure 2 Example of the layered structure based on JavaMail

2.3 Message Retrieval

The main classes used for message retrieval are shown in Figure 3. Functionality that can be separated from the Javamail dependency has been located in superclasses not shown in the figure. See general description of the classes in section 2.1.

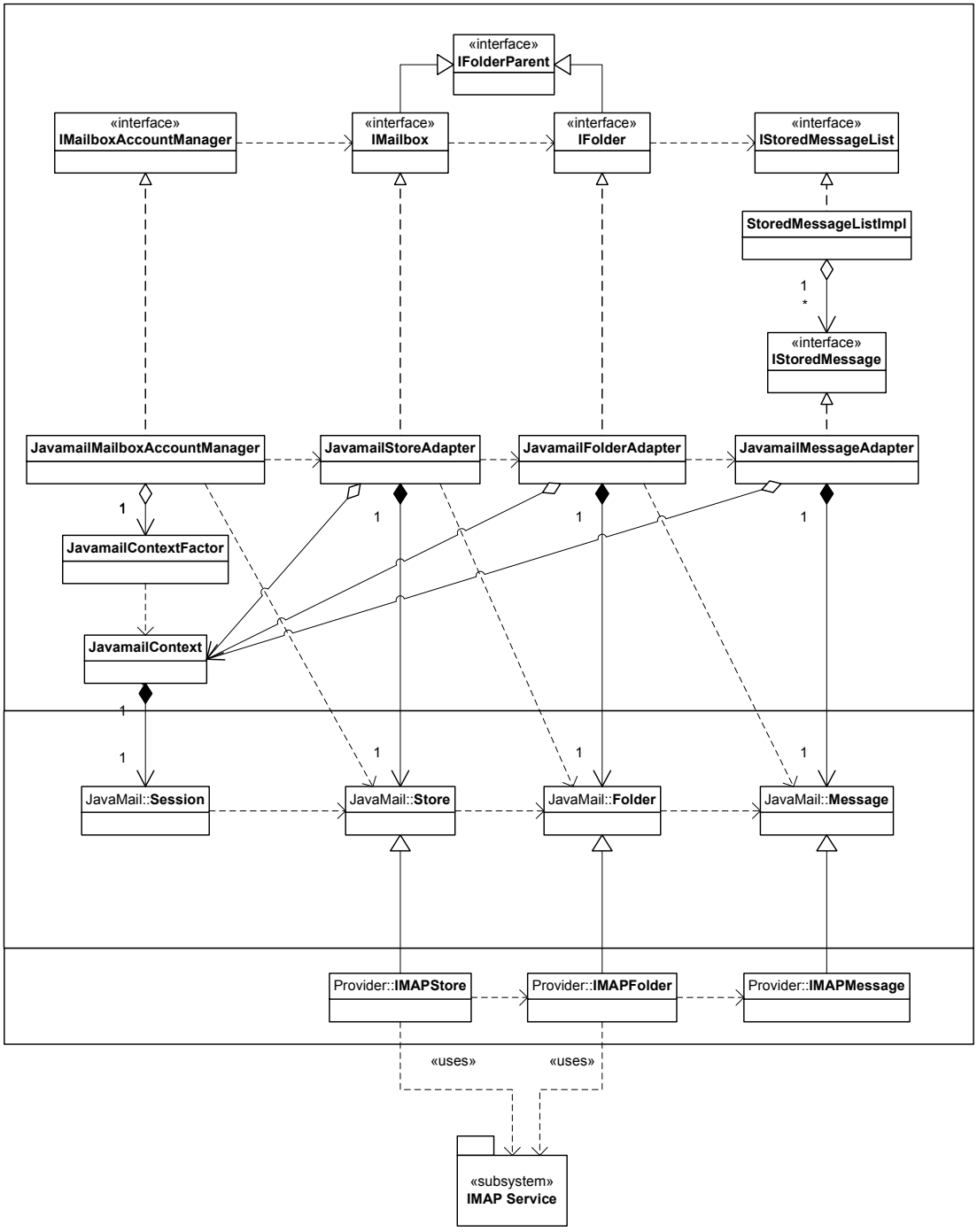

Figure 3 Message retrieval view of class diagram

2.4 Message Content

The main classes used for handling message content are shown in Figure 4. Functionality that can be separated from the Javamail dependency has been

located in superclasses not shown in the figure. See general description of the classes in section 2.1.

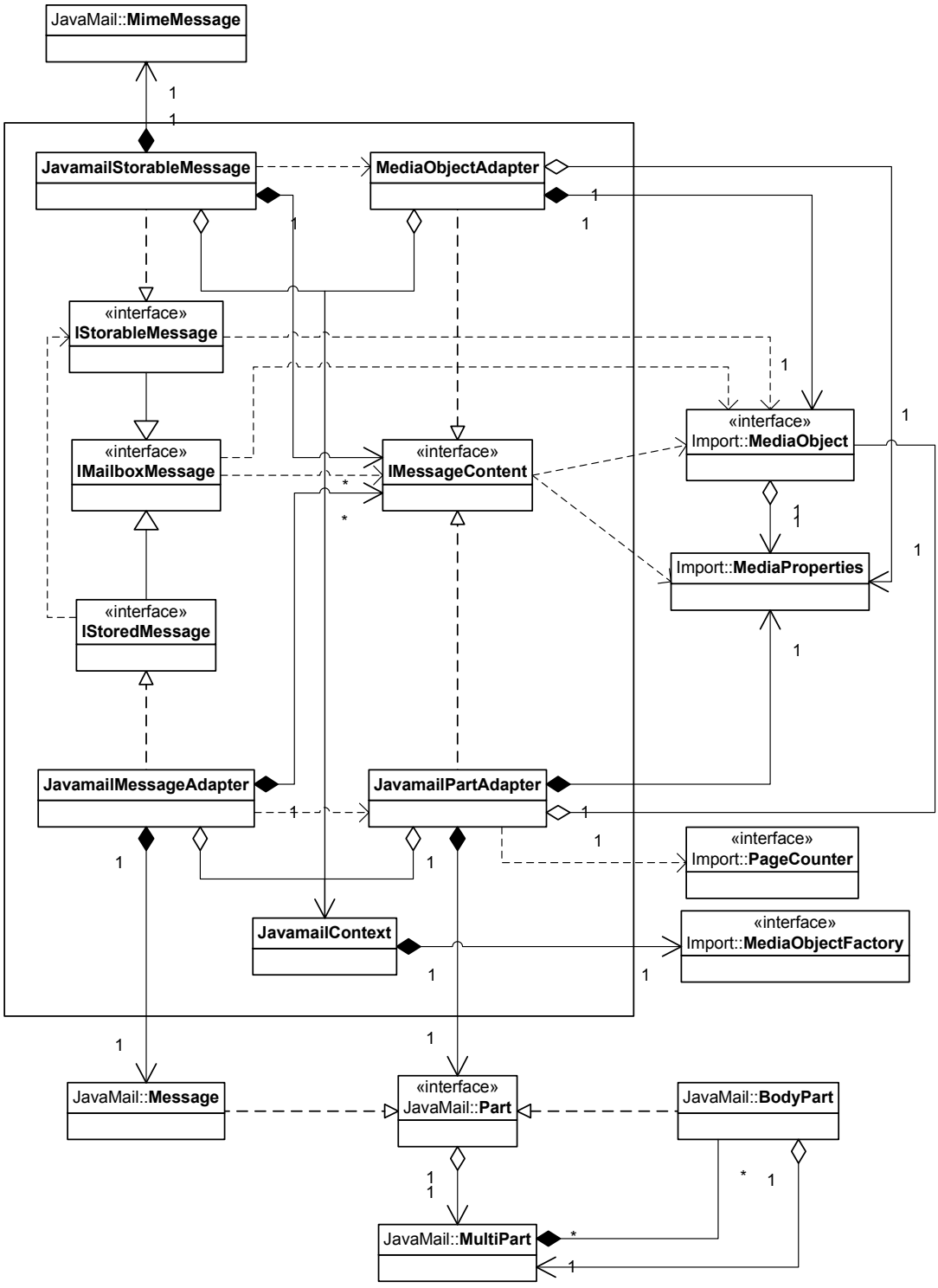

Figure 4 Message content view of class diagram

2.5 Message Deposit

The main classes used for message deposit are shown in Figure 5. Functionality that can be separated from the Javamail dependency has been located in superclasses not shown in the figure. See general description of the classes in section 2.1.

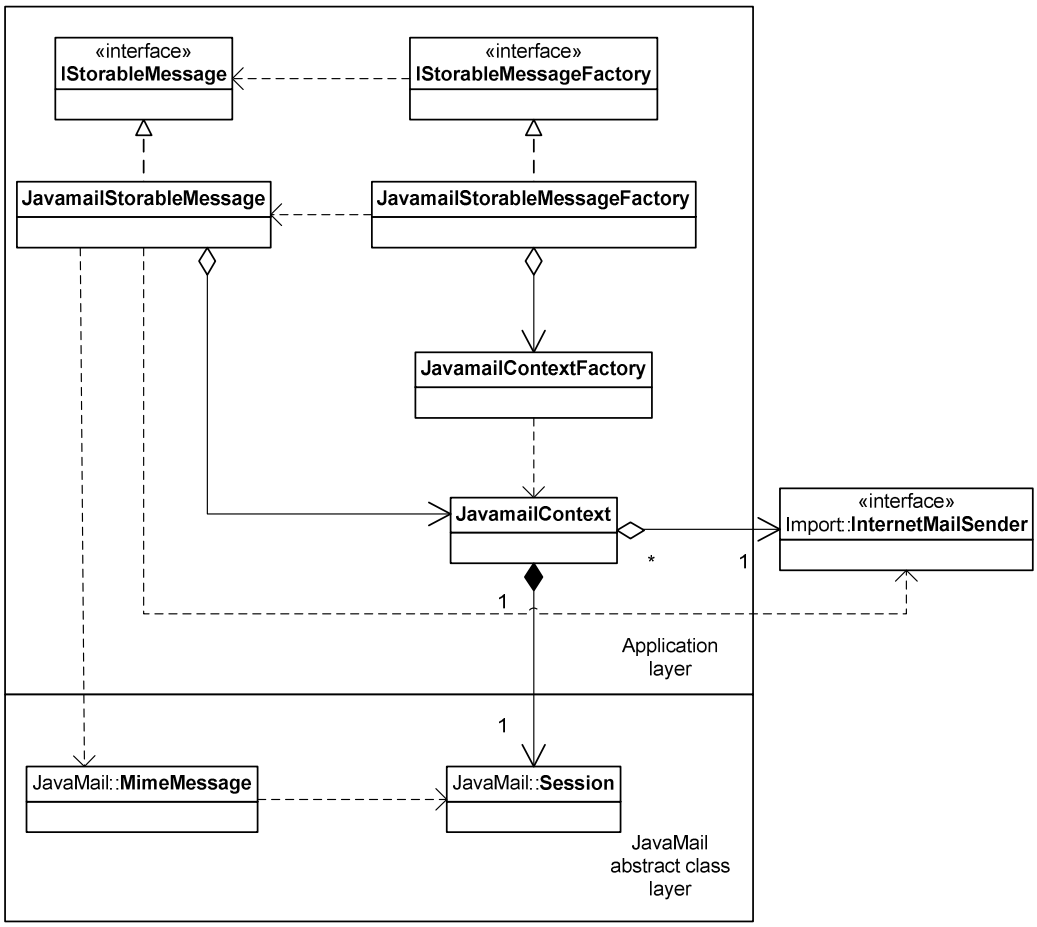

Figure 5 Message deposit view of class diagram

2.6 Mailbox Quota Usage

The main classes used for handling mailbox quota usage are shown in Figure 6. Functionality that can be separated from the Javamail dependency has been located in superclasses not shown in the figure. See general description of the classes in section 2.1.

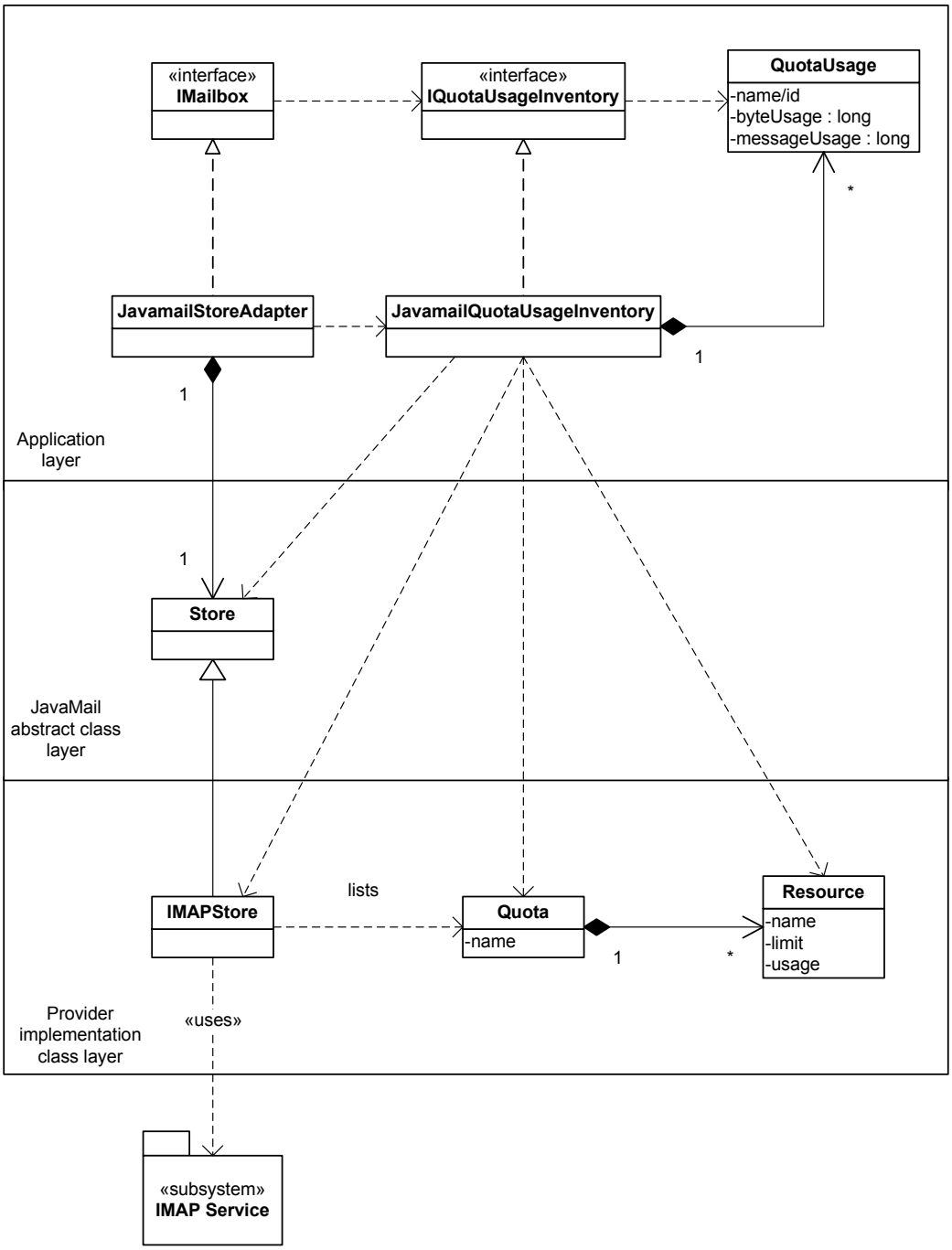

Figure 6 Mailbox quota view of class diagram

3 Function Behavior

3.1 Opening Mailbox

To retrieve messages and manage user mailbox the JavamailMailboxAccountManager needs to aggregate a JavaMail Store object. The Store object is allocated from the MailboxContext object which holds a StoreManager object. The StoreManager is initialized with a JavaMail Session object. This Session object is partly configured from static properties in the ComponentConfig.xml and partly from the Mailbox configuration.

3.2 Retrieving messages and content

Retrieving messages and its content is basically done in four phases.

1. Open user mailbox through allocating (if not already open) the JavaMail Store object representing user mailbox.
2. Fetching messages from the Message Store.
3. Fetching and examining/printing/forwarding the message content.
4. Close user mailbox and return resources.

3.2.1 Message and content cache

Mailbox tries to minimize the amount of IMAP calls and therefore message content is not fetched until first needed, i.e. lazy retrieval. But when requested, the content is fetched at once. The same applies for the message body; it is not fetched until first requested.

To be able to examine the message content the content properties can be retrieved without retrieving the actual content. The JavamailPartAdapter parses the message part and creates a MediaProperties object with the correct properties. When the content is requested, it will be retrieved from the store and cached in the JavamailPartAdapter as a MediaObject.

3.2.2 Page Counter

Counting pages in fax messages is executed by an injected PageCounter. The injected PageCounter object is setup to count the number of occurrences of the string "Fax Image" in all MIME attachments of content-type image/tiff.

3.2.3 Printing stored message

Printing of stored messages is performed by creating a JavamailForwardMessage (see section 3.2.4) from the message to be printed and sending this message to a preferable mail server with the fax recipient as receiver.

3.2.4 Forward stored message

Forwarding of stored messages is performed by creating a JavamailForwardMessage with the message to be printed. A

JavamailForwardMessage is an extension of the JavamailStorableMessage class which appends the original Message as a message/rfc822 content.

3.3 Manage Folders

Managing folders is basically done in three phases.

1. Open user mailbox through allocating (if not already open) the JavaMail Store object representing user mailbox.
2. *Manage folders*
3. Close user mailbox and return resources.

3.4 Quota usage inventory

Retrieving quota usage inventory is basically done in four phases.

1. Open user mailbox through allocating (if not already open) the JavaMail Store object representing user mailbox.
2. Request current Quota Usage from Message Store
3. Compose inventory
4. Close user mailbox and return resources.

3.5 Storing a Message

Storing a message begins with a client call to a JavamailStoreableMessageFactory to get a new JavamailStoreableMessage object. The client then sets the message properties, adds message content and finally calls the method to store the message.

The JavamailStorableMessage creates a new instance of the JavaMail message class MimeMessage. Then the properties and content are transferred from the JavamailStorableMessage to the MimeMessage. The `saveChanges` method is called on the MimeMessage to update the appropriate header fields of this message to be consistent with the message's contents.

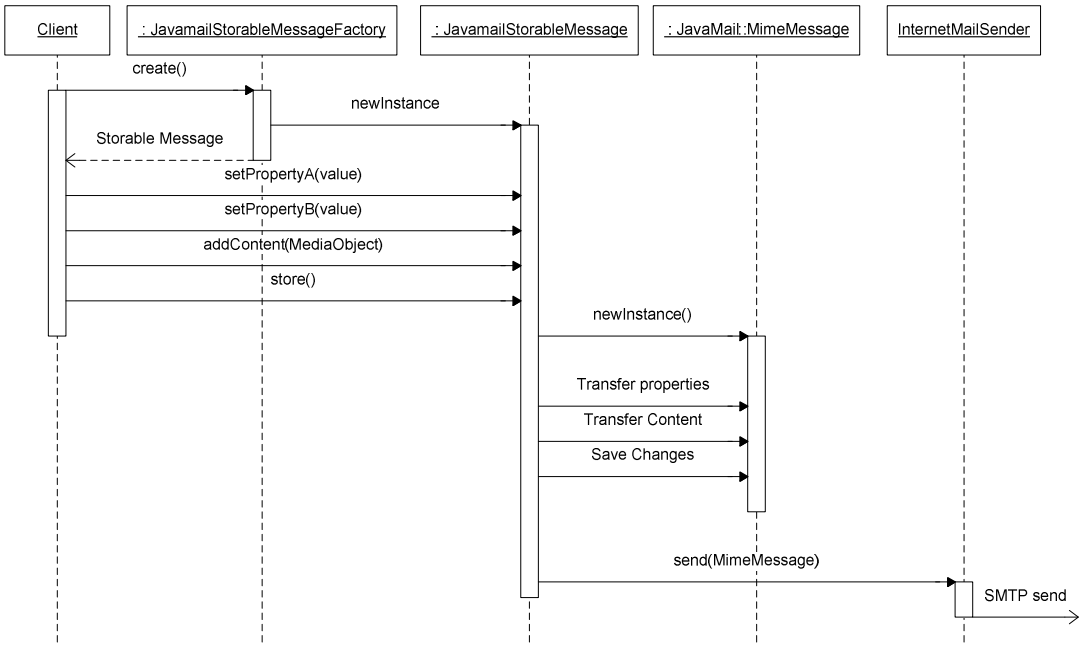

Figure 7 Storing a message

3.6 Additional properties

New message properties may be introduced without modifying the Mailbox implementation. Each additional property has a name that can be mapped to a MIME message header name. If no MIME message header name is configured it is supposed to have the same value as the additional property name.

3.7 Configuration

Mailbox has the below configuration possibilities. The latest configuration is loaded when opening the Mailbox and does not change during the session.

Table 1 Configuration settings

Setting	Description	Default
Additional properties	See section 3.6.	No additional properties.
Connection timeout	Service socket connection timeout value in milliseconds.	5000 ms
IMAP timeout	Socket I/O timeout value in milliseconds.	5000 ms

4 References

[1] FS – Mailbox
5/FS-MAS0001 Uen

[2] JavaMail 2.1
http://java.sun.com/products/javamail/

[3] IWD End User Message Format
3/155 19-HDB 101 02 Uen

5 Terminology

MIME	Multipurpose Internet Mail Extension
SMTP	Simple Mail Transfer Protocol

6 Appendix B: 3PPs

6.1 Third-Party Products and Freeware

3PPName/ Freeware Name	Version of the product/ freeware	Company	Used for	Delivered with the component	ECCN US/EU	Product No. and R-state
JavaMail API	1.3.3	Sun Microsystem, inc	Retrieving e-mail with IMAP, parsing and composing Mime formatted messages.	Yes	5D002	SWF0004 R1C
JavaBeans Activation Framework	1.0.2	Sun Microsystem, inc	Needed by JavaMail.	Yes	5D002	SWF0007 R1A