

P-IMAP Draft Overview

(<http://www.ietf.org/internet-drafts/draft-maes-lemonade-p-imap-00.txt>)

Stéphane H. Maes – stephane.maes@oracle.com

Jean Sini – jean.sini@oracle.com

Oracle Corporation

PIMAP: Goals and Motivation

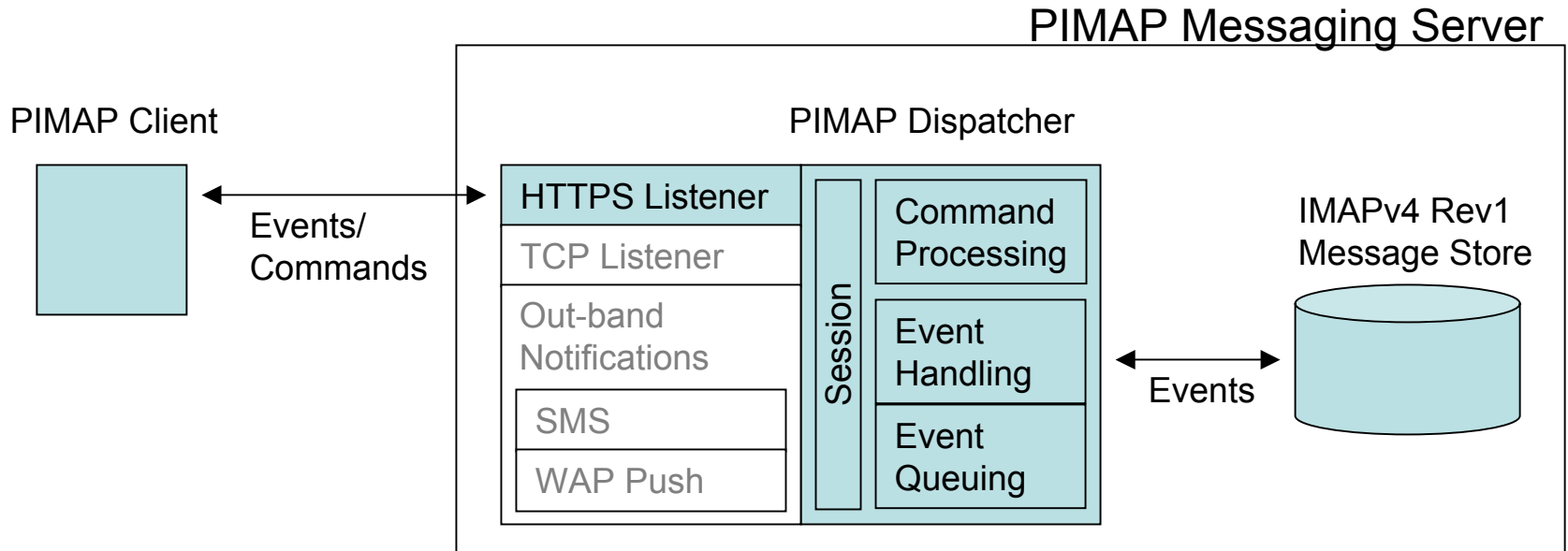
1. Support the mobile e-mail:
 1. secure, quasi real-time 2-way propagation of messaging events between messaging servers and mobile devices, allowing both stores to remain synchronized.
 2. Mobile e-mail usage
 3. Bandwidth optimization
2. Do so in a light-weight fashion:
 1. Not as much overhead as a session-based SyncML approach.
 2. Leveraging compression to ensure bandwidth conservation.
 3. Group commands to reduce round-trips.
3. Take into account the variety of available network bearers in order to adjust the message flows accordingly.
4. Extend message delivery capabilities to obviate the need for SMTP.
5. Directly interfaces and interoperates with IMAPv4 Rev1 Message Store
6. Support event-based PIM synchronization

PIMAP: Protocol Highlights

1. Extension commands allowing clients to:
 1. Set notification mode (in-band, out-band) and device address – XSETPIMAPPREF/XGETPIMAPPREF
 2. Set view and notification filters - XFILTER
 3. Send out messages - XDELIVER
 4. Exchange compression capabilities
2. Combine oft-used groups of commands into macros.
3. Support for various bindings to underlying protocols, for instance:
 1. PIMAP over HTTPS [Mandatory]
 2. PIMAP over TCP.
 3. Any other network optimizations can be used
4. Support for multiple notifications mechanisms:
 1. In-band notifications (e.g. within HTTP or TCP bindings).
 2. Out-band notifications (e.g. SMS, WAP Push, ...).

PIMAP: Flows

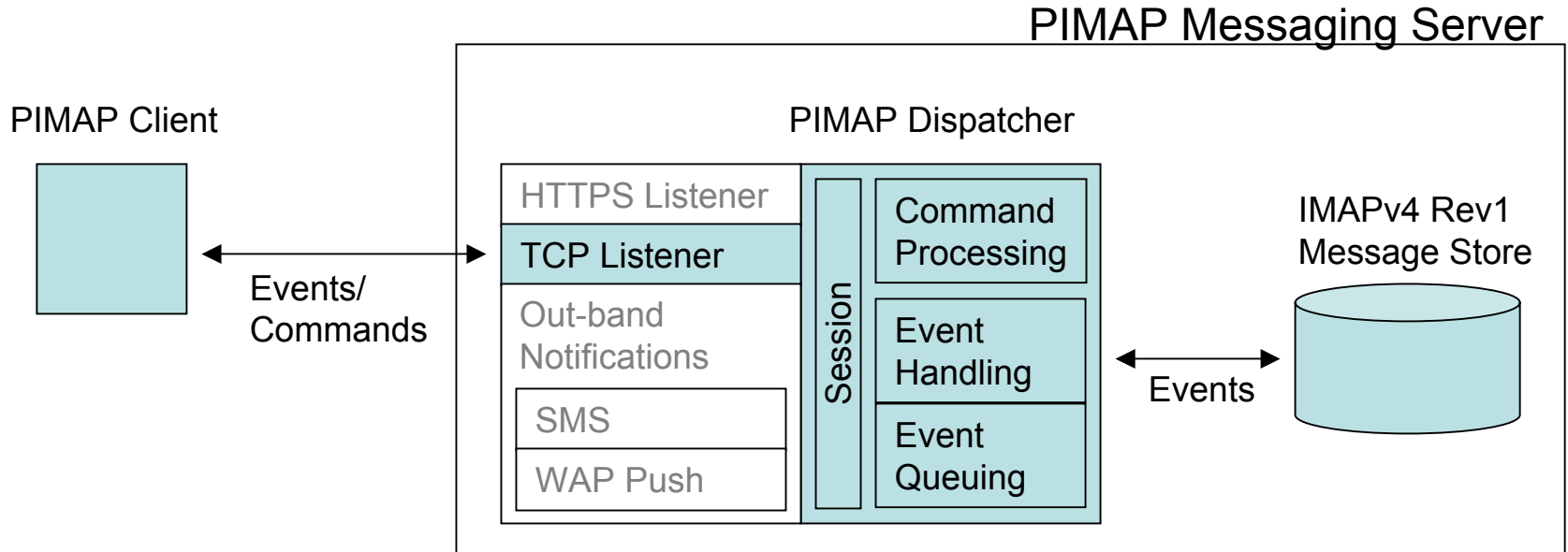
1. Keep-Alive HTTP binding, in-band notifications



1. Client establishes and authenticates a PIMAP session over a long-lived HTTPS request. It performs an IMAP state comparison for subscribed folders.
2. It uses this request to propagate client-originated events (send, delete, etc.)
3. Server uses long-lived response to notify of server originated events.
 1. Server receives notifications from the Message Store in different ways:
 1. Message Store has notification rules capabilities and can actively notify the PIMAP dispatcher
 2. PIMAP dispatcher opens an IDLE session to the Message Store in behalf of the user
4. Client reacts to notifications if needed (e.g. fetches body of new message)
5. If request/response ends, server maintains session and queues events.
6. Client re-establishes HTTPS request within session lifetime and retrieves events without the need for a full state comparison.

PIMAP: Flows

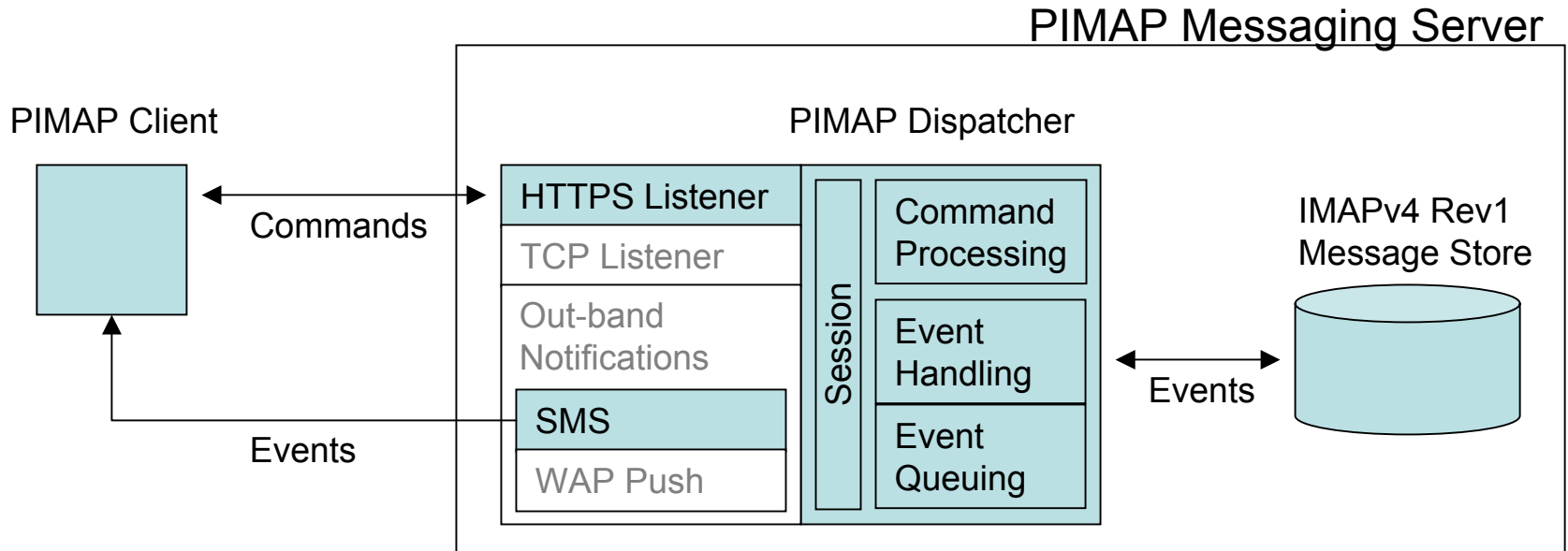
2. TCP binding, in-band notifications



1. Client establishes and authenticates a PIMAP session over a TCP connection.
2. It performs an IMAP state comparison for subscribed folders.
3. Client-originated events (send, delete, etc.) are propagated.
4. Server uses connection to notify of server originated events.
 1. Server receives notifications from the Message Store in different ways:
 1. Message Store has notification rules capabilities and can actively notify the PIMAP dispatcher.
 2. PIMAP dispatcher opens an IDLE session to the Message Store in behalf of the user.
5. Client reacts to notifications if needed (e.g. fetches body of new message).
6. Notifications may be missed when the client suddenly drops connection.
 1. In this case, the server sends a RESYNC untagged response whenever the client reconnects.

PIMAP: Flows

3. HTTPS binding, out-band notifications (e.g. SMS)



1. Client establishes and authenticates a PIMAP session over HTTPS.
2. It performs an IMAP state comparison for subscribed folders.
3. Client-originated events (send, delete, etc.) are propagated as requests.
4. Server uses SMS to notify of server originated events.
5. Client reacts to notifications if needed (e.g. fetches body of new message) over additional requests. The server maintains (cookie-based) a long-lived session and queues server events, reducing the need for full state comparisons.
6. If notifications are lost (out of coverage, etc.) the client retrieves pending events when one finally reaches its destination.
7. It is also possible that the client connects to server without even receiving the SMS. In this case, the server pushes pending events to the client in-band.

PIMAP: Protocol Revision History

1. Planned updates for Release 01
 1. Sections 1.1, 1.3, 2.2.1, 2.2.2, and 2.2.3
 1. Added diagrams to better explain P-IMAP concepts
 2. Section 1.4
 1. Point 1 - changed term definition to Compression
 2. Added points 5 and 6 regarding Attachment Handling
 3. Section 3.1.4
 1. Updated minimal P-IMAP server requirements
 4. Section 3.1.5
 1. Fixed the title – P-IMAP Session/Login
 2. Added examples for “First Login” and “Login after Logout” cases
 5. Added Section 3.1.7
 1. RESYNC untagged response to solve problems with missed notifications
 6. Section 3.2.2
 1. XSETPREF and XGETPREF becomes XSETPIMAPPREF and XGETPIMAPPREF
 2. Reduced the number of preference parameters
 7. Section 3.2.3
 1. Added a Days Before Today filter
 8. Removed section 4
 9. References
 1. Added references to IMAP-DISC and RFC 2180
 2. Removed references to MIMAP, NSMS
 10. Appendix B
 1. added example of outband notification and explained client's responsibilities
 11. Appendix C
 1. Removed completely, as attachment conversion is described in XCONVERT command and ways of retrieving it are discussed in RFC 2683
2. Release 00
 1. Initial release published on Feb. 8th 2004