The Top Ten Ways to Fix Email
Zimbra, Inc.[1]

Are you satisfied with your email application?
Or does your inbox look something like this?

As an email administrator, are you
happy with how much time you spend per
mailbox on basic "care and feeding"?

Is Email Broken? Web browsing and email are *the* two killer applications of the Web. Given the ubiquity of email, it is perhaps surprising that we users are not a happier lot. The frustrations oft associated with the email experience stand in contrast to the relative satisfaction of web browsing and web administration. How come?

Email has changed dramatically since the advent of the World-wide Web:

- The number of messages per day is up by an order of magnitude or more;
- The amount of storage required for our mailboxes is up by two orders of magnitude or more;
- Email has grown from the original one-to-one communication model to also include one-to-many (as mailing lists have displaced bulletin boards);
- Email applications are often responsible for managing calendars, group scheduling, contacts, tasks, public folders, and so on;
- Email applications also often manage shared documents (think "content management-lite") and even ad hoc document-oriented workflow among users;

[1] Zimbra is a trademark of Zimbra, Inc. All other trademarks belong to their respective owners.

- Email applications are expected to trap ever more sophisticated and ever higher volumes of spam and viruses;
- Email platforms are growing into unified messaging platforms by incorporating support for fax, voicemail, and instant messaging (including integrated anti-spam and anti-virus); and finally,
- Email applications are now also being asked to implement retention and discovery policies (such as for compliance with Sarbanes Oxley).

Indeed, email has changed sufficiently that we're no longer quite sure what to call it: Enterprise messaging? Groupware? Collaboration?

While email use has changed dramatically with the growth of the Internet, the underlying messaging systems we are counting on to keep pace with this change were generally designed more than a decade ago. Given the amount of time IT-intensive employees spend on email, it is ironic that innovation has reached consumer mail (e.g., gigabyte mailboxes for Google and Yahoo! users) ahead of enterprise mail!

As an industry, we can do better:

- We must make our messaging applications substantially more powerful and efficient: since employees can often average a couple hours per day dealing with emails, marked improvements in efficiency will have a significant impact on the bottom line.
- We must lower the total cost of ownership of deploying and managing enterprise collaboration solutions in the face of growing requirements: archiving and compliance; performance and scalability; fault tolerance; storage management/attachments; integration; and security?

Fixing Email Top Ten. In the coming years, we expect that email will undergo as dramatic a transformation as it has since its inception over three decades ago. What follows is our top ten proposals for fixing email:

1. *Self-Organizing Mailboxes (a.k.a. downsizing the "folder" habit)* – Too much of our email time is spent organizing (moving, copying) our correspondences between potentially hundreds of nested folders. The rich, instantaneous search capabilities of the Google Desktop or Apple SpotLight enable you to archive (or "stash") email even in a single archive folder with the confidence that you can almost instantaneously find whatever you are looking for without any of the overhead of managed folders. Comprehensive, automatic indexing ensures that subsequent search is far faster (and less error prone) than traversing folder trees. Search should be sufficiently rich in that just about any syntactic criteria can be fair game — message content, attachment types, attachment content, web URL content, sender, receiver, sending or receiving domain, dates, tags/labels, and so on. These rich searches can be saved in order to provide virtual folders for managing "views"

into your inbox or archive. (Newly arrived messages that conform to the search criteria, of course, appear in the matching virtual folders.)

Fixing Messaging and Collaboration for System Administrators

- Complete Solution
  - Native archiving capabilities
  - Native discovery/cross mailbox search
  - Native hierarchical storage management (HSM)
  - Native replication/disaster recovery (WAN optimized)
- Server and Storage Consolidation
  - Substantial reduction in managed storage — One copy of email & attachments per server (rather than one per user or one per storage group)
  - Substantial increase in the number of users per CPU — Multi-level caching and optimization of underlying open source software (Linux File System, MySQL, Postfix, etc.)
- Lower-Overhead Management
  - Easy installation and configuration
  - On-line automated backup
  - On-line, intelligent restore (single mailbox, point-in-time, etc.)
  - Fast mailbox move
  - Rich, secure, zero administration client
- Compatibility with existing infrastructure
  - Client — Outlook, Mobile, IMAP, POP, iCalendar, etc.
  - Directory — Active Directory/LDAP integration, provisioning support
  - Messaging Server — Co-existence and migration tools (including Microsoft Exchange)
  - Web services — Bi-directional integration with enterprise applications
- Security
  - Web security model — Single sign-on, no VPN required
  - Secured attachment opening and HTML rendering
  - SpamAssassin and ClamAV included
  - Compatibility with existing anti-spam/anti-virus (via Postfix & amavisd-new)
- Open Solution
  - Open source
  - Open formats — One MIME message per file means Zimbra storage is not opaque to sys. admin. or operating system utilities

2. *Active Messaging*™ – Web browsers provide the universal "pull" interface for accessing applications on the Internet. As email has become the universal "push" interface, Internet applications ought to also be better integrated with collaboration. Such integration can go well beyond recognizing URLs: Imagine
   - Mousing over a tracking number to locate your FedEx shipment;
   - Mousing over the string "next Friday" to see what's in your calendar;

- Right-clicking on “next Friday” to set up an appointment for lunch in your calendar;
- Right-clicking on a phone number to make a Skype call;
- Mousing over a purchase order (PO) number to see the associated company and dollar value;
- Right-clicking on a PO to approve or reject it; and so on.

With Active Messaging, email content can be easily linked (via web services) to back-office applications such that a “mouse over” shows customer case tracking details and a mouse click allows you to take actions such as an escalation. External links such as to trade status, shipping status, on-time departures, weather for a particular zip code, and so on can just as easily be “activated” via simple administrative configuration. The net result is “mash ups” in email!

3. *Extensible, “Integratable” Collaboration* – Under the above scenarios, email is the “client” of the integrated service, but email applications should just as easily be able to provide services for existing enterprise applications to tap into. In this way, your enterprise applications can securely send email, send instant messages, submit tasks, check on presence, and even manage a document workflow over email, all via XML/web service bindings into the email system. Such extensibility can also be leveraged, for example, to implement specialized security policies (such as for externally-bound email, inter-departmental mail, and so on).

4. *Self-Organizing Conversations* – Too much of our emailing time is spent reassembling the multiple threads (the replies and forwards) that comprise a conversation. Suppose a message arrives from a customer, which you first acknowledge, then forward internally for assistance, and then file away. When another message arrives from that customer, shouldn’t your email take care of presenting the new message in context with the other relevant communications so that the information you need for decision making is always at your fingertips? Or more simply, how often do you respond to a message without realizing that there has already been relevant additional discussion?

5. *Efficient Context Switching* – When a meeting proposal arrives, why do I have to switch to my calendar view and then back to my inbox? Shouldn’t I be able to take a “quick look” at my schedule and make a decision without leaving the email application? When my boss sends me his new cell-phone number, shouldn’t I be able to add that to my contacts without leaving the email? And why when I travel can’t my collaboration client keep track of the fact that my weekly staff call is still in my home time zone? Today’s collaboration solutions require that we switch top-level contexts to do even simple tasks, and then navigate back by hand.

Fixing Messaging and Collaboration for End-users

- Self-Organizing Mailboxes
  - Powerful, fast search (including messages *and* attachments)
  - Saved searches across folders
  - Conversation management across folders
  - Tags for automatic and hand categorization of messages
- Active Messaging™
  - Content linked to other applications (for mouse-over and click)
  - Intranet — ERP, CRM, Support, Finance, HR, VoIP phone, etc.
  - Internet — Google Maps, Skype, Travel, Package Tracking, etc.
  - Messaging application — Access across email, contacts, calendar, etc. without switching context
- Any Place, Any Machine
  - Zero footprint, rich AJAX Web client (cross browser/OS)
  - Security sans VPN
  - Secure, read-only access to attachments without special-purpose client software
- Freedom of Client Choice
  - PC — Outlook (Online, Offline, Cached Mode), Apple Mail and iCal, Eudora, Evolution, Thunderbird/Sunbird, etc.
  - Mobile — Wireless devices "over the air" synchronization sans special client software: Blackberry, Palm, Nokia, Motorola, Good, PocketPC, etc.

6. *Better Storage Management* — Why is it that I can get 1Gb of storage for my Google or Yahoo! consumer mail for free, but my enterprise email quota is still measured in Mbs. Disk may well be cheap, but managed storage (e.g., SAN, NAS) is far from free. Yet, most existing collaboration solutions redundantly store attachments, either within every mailbox or at least within every storage group. Simply by increasing the sharing of attachments, you can reduce your managed storage overhead dramatically. Moreover, administrative operations associated with managed storage must be simplified. Garbage collection, for example, should be fully automated. Restoration from a backup should able to be for an individual mailbox, at a particular point in time, and into a live system (no "bare metal" restoration required). Finally, typical I/O-bound email systems perform two or three disk reads for every write. While disk writes tend to be mandatory (for persistence), advanced caching can reduce the read load substantially, thereby improving the user experience and enabling server consolidation. We have now demonstrated multi-level caching (both at the application level and at the underlying systems level) improving throughput on a particular class of hardware by 2X over industry leading collaboration servers.

7. *Unified Search, Retention, and Archiving* — Companies are increasingly obligated to retain all messages for a specific period of time in order to

comply with Sarbanes-Oxley, corporate retention policies, and so on. Why not kill two birds with one stone by using the same infrastructure for both individual users and the greater corporation to insure that all messaging content complies with such policies? Why purchase and deploy two different solutions and the associated redundant storage and management overhead? If all of my emails are being retained anyway, why not allow users to have access to the same powerful search tools (and avoid the overhead of restoring snapshots of inadvertently deleted email). Why shouldn't your collaboration software be smart about hierarchical storage management

8. *Integrated Web collaboration technology* – RSS and Wiki have emerged as simple, versatile tools for managing changing content on both intranets and the Internet, while the emerging iCalendar and CalDav standards promise to bring some of the interoperability of Internet email to calendaring. Finally, instant messaging and presence have helped to speed time-critical collaboration. To date, enterprise messaging servers have often served in the role of *de facto* platforms for collaboration, since email attachments and public folders provide the means for sharing (and now auto-indexing and searching) documents. Why not simply implement these web collaboration standards within the next-generation of collaboration products, rather than relying upon heavier-weight, proprietary alternatives?: for example, RSS feeds for calendar updates and Wiki overlays for group folders.

9. *More Server-Centric Architecture* – Most existing messaging products are client-centric, in that they have grown up from the PC-client to a shared backend server designed to support those clients. Numerous factors are pushing more collaboration functionality and the message "store of record" to the server-side:
   - Archiving of all messages for compliance or discovery
   - Implementation of retention policies that require all messages over a certain age to be discarded
   - Multi-client support (PCs, Mac's, Linux desktops)
   - Multi-device support (Blackberry, Treo, PocketPC, phones, etc.)
   - Performance – Since email performance is disk I/O bound, users derive better performance by spinning more and faster disks on the server.
   - Faster client/server synchronization

10. *Security* – Email security is still catching up to that of the Web. Frist, we need single sign-on (SSO) in order to secure implement mash ups/Active Messaging that incorporate web services. Second, we need technologies like SMIME that authenticate senders to receivers, and which will provide a stronger foundation from which to combat spam. Next, there is simply no excuse today for email solutions that pass

client/server communications in clear text. Similarly, it is high time that instant messaging (IM) traffic be encrypted on the wire, and not stored in the 3rd-party servers. IM infrastructure must, moreover, be subject to the same archival (for compliance) and anti-virus infrastructure in place for traditional email. Finally, collaboration solutions should support a mode of operation in which potentially dangerous attachments are opened and managed on secured servers, often running less-vulnerable operating systems. While such attachments are opened on the server, they are still rendered to the client in HTML, thereby preventing the spread of a virus; protecting the security of the document (by keeping it from being cached on the disk of a non-secured machine); and, incidentally, allowing read-only access without having to install expensive client software.

Charge. Innovation is the key. Too often our rhetoric in promoting open source alternatives is focused on reduced licensing costs, which is more a recipe for customers soliciting further discounts from the incumbents. Instead, we should leverage the strength of our open source communities (whether Zimbra or the alternatives) by innovating in order to make email/collaboration more efficient, to make email administration less painful, and to drive down total cost of ownership in enterprise messaging. As a community, we have many more bright ideas than any single company can muster, but only through innovation can open source achieve its potential.

Did we miss your email pet peeve(s)? Then please come join the fixing email dialog at www.zimbra.com. Comments specific to this whitepaper can be sent to scott.dietzen@zimbra.com. Thank you.

