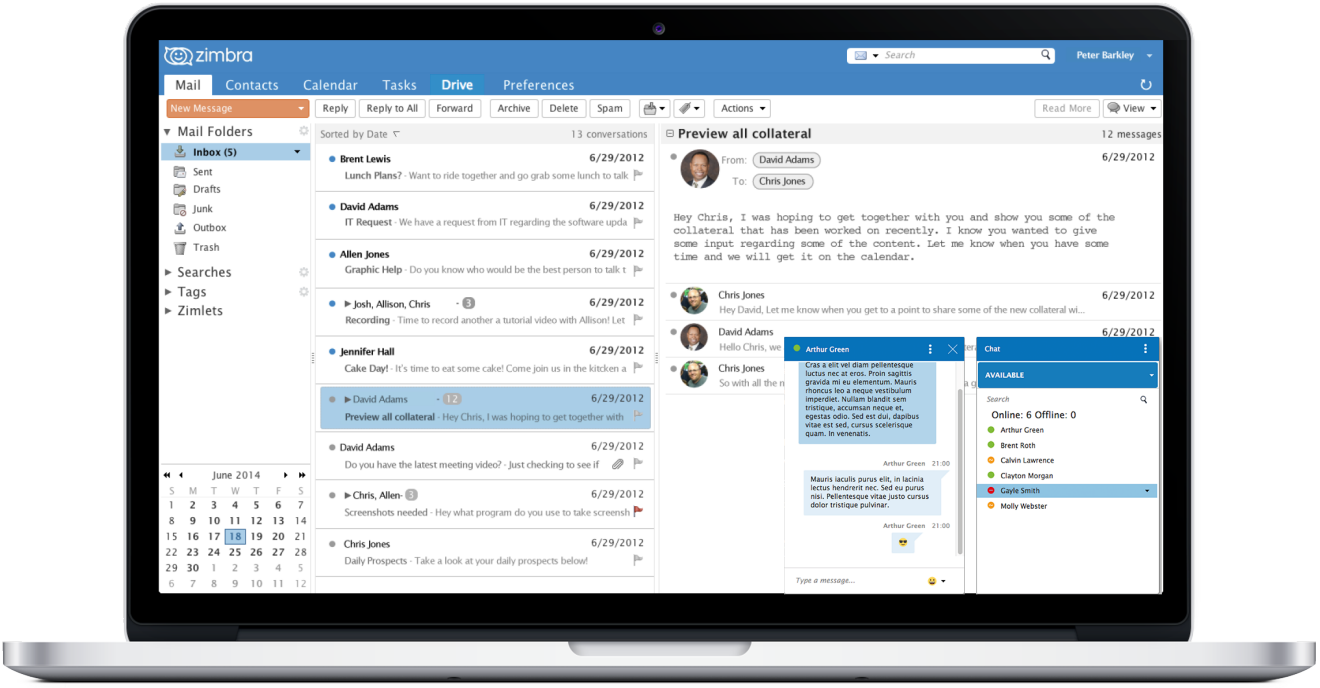
Zimbra Collaboration Technical Overview

Table of Contents

Zimbra Overview

Zimbra Collaboration is the world's leading open source messaging and collaboration solution, trusted by more than 5,000 Enterprises, Public Sector organizations and Service Providers, providing mailbox services for hundreds of millions of end users, in over 140 countries.

Zimbra includes complete email, contacts, calendar, file-sharing and tasks, and can be accessed from the Zimbra Web client via any device. Featuring the most innovative cross-platform web application today, compatibility with all devices and desktops, Zimbra boosts end-user productivity at dramatically lower costs compared to other legacy solutions. Zimbra provides an open platform designed for virtualization and portability across private and public clouds, making it simpler to manage and more cost effective to scale. Zimbra is the world leader in Open Source based, next-generation messaging and collaboration software.



The Zimbra Collaboration architecture is built with well-known open source technologies and standards-based protocols. The architecture consists of client interfaces and server components that can run as a single node configuration or be deployed across multiple servers for high availability and increased scalability.

The architecture includes the following core advantages:

Core Advantage	Components/Description
Open source integrations	Linux®, Jetty, Postfix, MariaDB, OpenLDAP®
Industry-standard open protocols	SMTP, LMTP, SOAP, XML, IMAP, POP
Modern technology Design	HTML5, Javascript, XML, and Java
Scalability	Each Zimbra mailbox server includes its own mailbox accounts and associated message store and indexes. The Zimbra platform scales vertically (by adding more system resources) and horizontally (by adding more servers)

Core Advantage	Components/Description
Browser-based client interface	Easy, intuitive access to Zimbra Collaboration features, using a standard web platform.
Browser-based Administration Console	

New Features with Zimbra

Next Generation Modules

NG modules include Real-time Backup, Synchronization of Shared Resources, HSM on Amazon S3 and a new simplified Delegated Admin.

Realtime Backup & Restore

Zimbra's new realtime backup and restore engine backs up every single item and event on your server with split second precision. It's designed to avoid data loss using atomic and ever-consistent algorithms, while still saving disk space *up to 50%* thanks to an intelligent deduplication and compression system. It offers six restore features, from a single-item restore to complete disaster recovery. All of the restore modes are transparent to the end-user and are 100% OS, architecture and version independent.

Hierarchical Storage Management (HSM)

Featuring advanced Zimbra store management and HSM, you can now manage multiple volumes and HSM policies through your Zimbra Administration Console. Save additional valuable storage space, easily expand your Zimbra server by adding new volumes at will and improve your server performances by splitting the I/O load onto different storage media and use different tiers of storage equipment.

Mobile Sync

What's new about Zimbra's mobile sync functionality? Users can now sync their shared Zimbra items along with all of their other Zimbra data. Using Exchange ActiveSync protocol, which is natively supported by the vast majority of mobile devices, your data will always be at your fingertips, without any middleware or dedicated client. With version 8.8.9, there is another new secuity feature added, which is ABQ (Allow, Block, Quaratine)

Simplified Delegated Admin

In minutes, you can now grant Delegated Admin rights to users, allowing them to perform management tasks such as setting quotas, COS and user limits for your domains. Keep track of your Delegated Admins' actions and your domain status with new reporting features, straight from your Zimbra Administration Console thanks to the dedicated Administration Zimlet.

Other Features

Two-Factor Authentication

Zimbra includes now two-factor authentication, making your mailbox more secure by providing a physical layer to produce a successful and secure login.

SSL SNI for HTTPS

Zimbra SSL SNI allows the server to present multiple certificates on the same IP address and TCP port number, so multiple hostnames can be served over HTTPS from the same IP address.

Postscreen for Zimbra MTA boosts Email Security

Keeping spambots away, Zimbra Postscreen leaves more SMTP server processes available for legitimate clients, and delays the onset of server overload conditions.

Zimbra Collaboration Packaging System

Apply security patches and Zimbra updates easier than ever before. Simply use your OS update commands, and all of your Zimbra packages will be updated as well. This is still a work in progress and full controls will be available in a future release.

S/MIME Digital Signatures and Encryption

Zimbra (Network Edition) provides cryptographic security services for email: authentication, message integrity, non-repudiation of origin (using digital signatures) and privacy and data security (using encryption). It's a server-based solution that does not require Java on the client machine. User encrypted certificate and private key are stored on the server, and the server performs all the cryptographic operations.

Exchange Web Services

If your users are using an enterprise email client like Microsoft Outlook, Zimbra provides Exchange Web Services, which allows your users to connect to their entire mailbox on a Mac. Microsoft Outlook 2011 and Microsoft Outlook 2016 are both supported.

Outlook Synchronization ZCO

If some of your users prefer to work with enterprise email client like Microsoft Outlook, Zimbra provides a MAPI Connector which allows your users see their entire mailbox in Microsoft Windows using Microsoft Outlook 2010 and above.

Archiving and Discovery

Zimbra Archiving & Discovery is an optional feature that enables you archive messages that were delivered to or sent by Zimbra and to search across mailboxes. It's specially designed for legal and audit purposes.

Reconfigured IMAP Service

Zimbra 8.8 has a new refactored IMAP service to run separately from all other Zimbra services. IMAP will no longer affect SLAs! Zimbra 8.8 also has improved SIEVE RFC compliance, offering administrators greater email filtering.

Zimbra Chat

Now available in all versions of Zimbra! Peer-to-peer chat using XMPP. Know your company's chat data is secure and private while your users enjoy searching chat history, emoticons, etc.

Zimbra Talk V2

Zimbra Talk V2 brings secure, high-quality chat and videoconferencing right into the Zimbra Web Client. Users can chat and videoconference 1:1 and in groups, share files, share their screen and so much more. Zimbra Talk V2 is licensed on a per-user basis, and is available in Network Edition only.

Zimbra Chat and Zimbra Talk V2 are not compatible and interoperable, meaning that the two products cannot freely coexist on any Zimbra NE infrastructure. The Zimbra Chat zimlet is uninstalled during the installation process of the Zimbra Talk Zimlet package. However, Zimbra Talk includes all «basic» IM features provided by Zimbra Chat which will be automatically enabled for all users who don't have access to the «advanced» Zimbra Talk V2 features.

Zimbra Drive

Zimbra Drive offers an integrated file sync and share functionality which needs to be built on an external ownCloud/Nextcloud platform. Zimbra Drive provides seamless synchronization and sharing of files between your users, wherever they are and on any device.

Zimbra Docs

Users can now share and collaborate in documents, spreadsheets and presentations realtime within Zimbra. This feature is in the Zimbra Briefcase, and it is an integration with LibreOffice.

Zimbra Multi-Tenancy Value Addition

Zimbra is a multi-tenant platform that can natively host hundreds or thousands of domains with an array of service offerings. Within the Zimbra platform, a set of common service capabilities is called a "Class of Service" (COS), and COS settings can be used to automate users provisioning requirements and establish service levels. For example, one COS can be targeted at “Basic” users with basic services (Webmail, POP, and SMTP), while another COS can provide “Normal” or “Mobile” services (Zimbra Connector for Outlook, over-the-air ActiveSync for mobile devices, sharing calendaring, Documents, Briefcase, etc.).

Zimbra features and preferences can be set in the account profile for individual users or by the Class of Service (COS) for multiple users/groups when accounts are created. These settings can be modified at any time and are easily controlled via the Zimbra Admin interface. When deploying a

Zimbra email infrastructure, multiple Class of Services can be defined and users will inherit the functions, features and branding associated with the COS to which they provisioned. When a user is provisioned into a specific COS they will automatic inherit the features and settings as defined in the COS, however it is possible to override COS attributes at a per user account level as part of the provisioning process.

Technical Overview

Zimbra is an innovative messaging and collaboration application that offers the following state-of-the-art solutions that are accessed through the browser based web client.

- Intuitive message management, search, tagging, and sharing.
- Personal, external, and shared calendar.
- Personal and shared Address Books and Distribution Lists.
- Personal and Shared Task lists.

Zimbra Components

Zimbra architecture includes open-source integrations using industry standard protocols. The third-party software listed in Third-Party Software is bundled with Zimbra software and installed as part of the installation process. These components have been tested and configured to work with the software.

Table 1. Third-Party Software

3rd-Party Component	Description
Jetty	Web application server that runs Zimbra software.
Postfix	Open source mail transfer agent (MTA) that routes mail messages to the appropriate Zimbra server
Open LDAP software	Open source implementation of the Lightweight Directory Access Protocol (LDAP) that stores Zimbra system configuration, the Zimbra Global Address List, and provides user authentication. Zimbra can also work with GAL and authentication services provided by external LDAP directories such as Active Directory
MariaDB	Database software
Lucene	Open source full-featured text and search engine
	Third-party source that converts certain attachment file types to HTML
Anti-virus/anti-spam	Open source components that include: • ClamAV, an anti-virus scanner that protects against malicious files • SpamAssassin, a mail filter that attempts to identify spam • Amavisd-new interfaces between the MTA and one or more content checkers

3rd-Party Component	Description
Apache JSieve	Manages filters for email
LibreOffice	High fidelity document preview

Zimbra Application Packages

Zimbra Collaboration provides the application packages listed in Application Packages.

Table 2. Application Packages

Package	Description
Zimbra Core	The libraries, utilities, monitoring tools, and basic configuration files. `zmconfigd` is contained in the zimbra-core and is automatically enabled to run on all systems.
Zimbra Store	The components for the mailbox server (including Jetty). The Zimbra mailbox server includes the following components: • **Data store** — A MariaDB database. • **Message store** — Location of all email messages and file attachments. • **Index store** — Index and search technology is provided through Lucene. Index files are maintained for each mailbox. • **Web application services** — The Jetty web application server runs web applications (webapps) on any store server. It provides one or more web application services.
Zimbra LDAP	Zimbra Collaboration uses the OpenLDAP® software, which is an open source LDAP directory server. User authentication, the Zimbra Global Address List, and configuration attributes are services provided through OpenLDAP. Note that the Zimbra GAL and authentication services can be provided by an external LDAP Directory such as Active Directory.
Zimbra MTA	Postfix is the open source mail transfer agent (MTA) that receives email via SMTP and routes each message to the appropriate Zimbra mailbox server using Local Mail Transfer Protocol (LMTP). The Zimbra MTA also includes the anti-virus and anti-spam components.

Package	Description
Zimbra Proxy	Zimbra Proxy is a high-performance reverse proxy service for passing IMAP[S]/POP[S]/HTTP[S] client requests to other internal ZCS services.This package is normally installed on the MTA server(s) or on its own independent server(s). When the zimbra-proxy package is installed, the proxy feature is enabled by default. Installing the Zimbra Proxy is highly recommended, and required if using a separate web application server.
Zimbra Memcached	Memcached is automatically selected when the zimbra-proxy is installed. At least one server must run zimbra-memcached when the proxy is in use. You can use a single memcached server with one or more Zimbra proxies. zimbra-memcached is required if using a separate web application server.
Zimbra SNMP (Optional)	If you choose to install zimbra-SNMP for monitoring, this package should be installed on every Zimbra server.
Zimbra Logger (Optional)	If used, this is installed on one mailbox server, and must be installed at the same time as the mailbox server.The Zimbra Logger installs tools for syslog aggregation and reporting. If you do not install Logger, the **server statistics** section of the Administration Console will not display.
Zimbra Spell (Optional)	Aspell is the open source spell checker used on the Zimbra Web Client. When Zimbra-Spell is installed, the Zimbra-Apache package is also installed.
Zimbra Apache	This package is installed automatically when Zimbra Spell or Zimbra Convertd is installed.
Zimbra Convertd	This package is installed on the zimbra-store server. Only one Zimbra-convertd package needs to be present in the Zimbra Collaboration environment. The default is to install one zimbra-convertd on each zimbra-store server. When Zimbra-Convertd is installed, the Zimbra-Apache package is also installed.
Zimbra Archiving (Optional)	Archiving and Discovery offers the ability to store and search all messages delivered to, or sent by the Zimbra Collaboration Server. This package includes the cross mailbox search function which can be used for both live and archive mailbox searches. Note: Using Archiving and Discovery can trigger additional mailbox license usage. To find out more about Zimbra Archiving and Discovery, contact Zimbra sales.

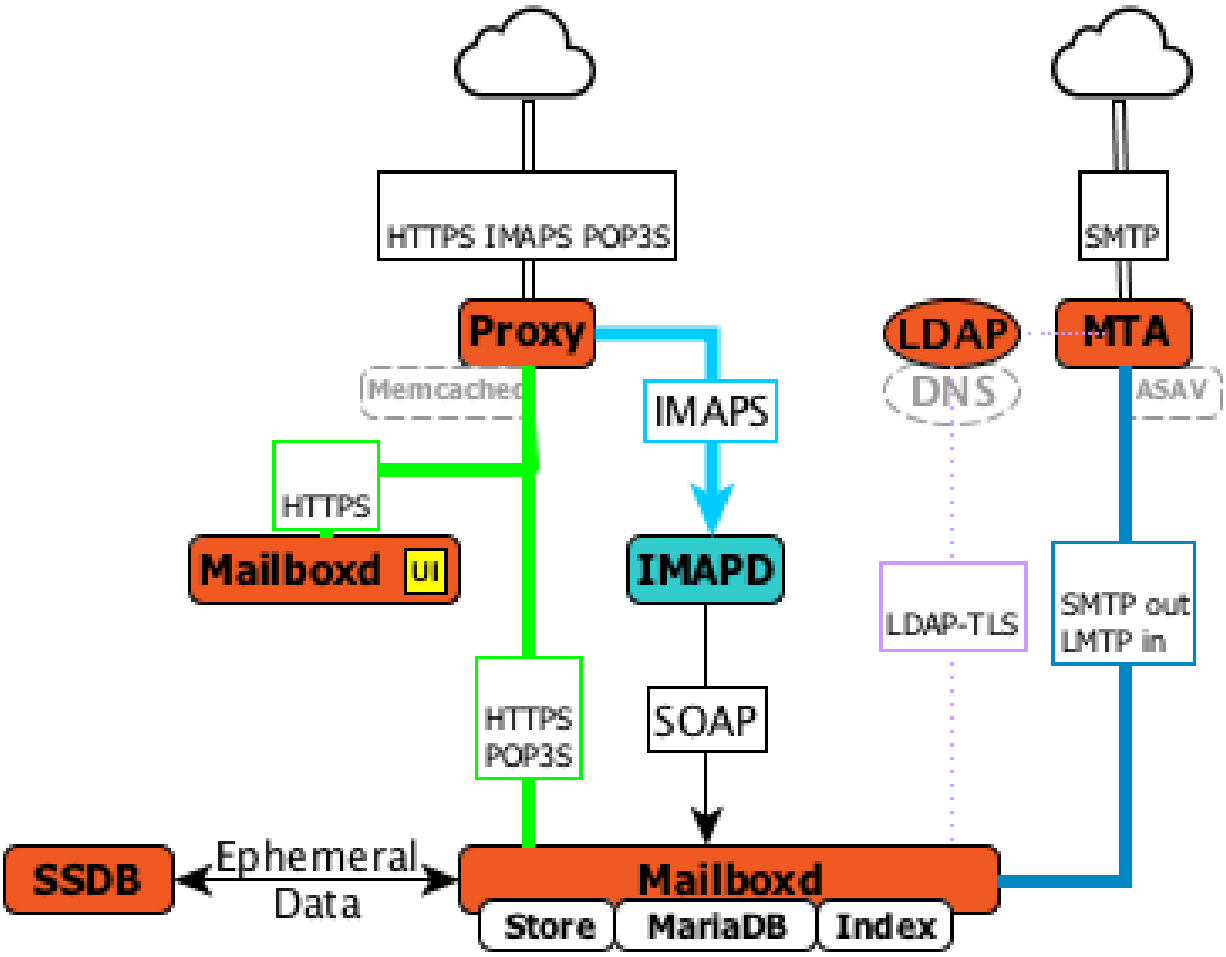
Supported Operating Systems

Zimbra Collaboration Network Edition v8.6.x, v8.7.x and later is supported on the following operating systems:

- Red Hat Enterprise Linux 7
- CentOS Linux 7
- Red Hat Enterprise Linux 6, patch level 4 or later is required
- CentOS Linux 6, patch level 4 or later is required
- Oracle Linux 7.2
- Oracle Linux 6.6
- Ubuntu 16.04 LTS
- Ubuntu 14.04 LTS

Zimbra Components Overview

Zimbra is designed to provide an end-to-end mail solution that is scalable and highly reliable. There are 4 major services in Zimbra described below. You can also have all of the services running on a single virtual server or they can be installed separately on multiple servers. If you have less than 1,000 users, it is very common to have only a single virtual server running all of the Zimbra processes.



In the Zimbra architecture, you can mix and match these services on a server basis. For example, you may have a server that is running the proxy and MTA services, and you may have another server on the backend running the mailboxd and the LDAP services. You can also have different virtual servers for each environment. For example, you might have a virtual server that is running a proxy, a virtual server that is running the MTA, a virtual server that is running LDAP, and a virtual server that is running the mailbox service.

LDAP

This is the heart of the Zimbra architecture based on the open source project OpenLDAP®. Every implementation of Zimbra must have the Zimbra LDAP instance.

The LDAP service holds all of the configuration information needed to run the Zimbra environment. There is account information in the LDAP database that includes the username, password, and all other attributes associated with that account, including the mailbox server that the account resides on, the preferences for each user, etc. Zimbra can also integrate with other directories for wider network requirements or external lookups.

LDAP also stores the domain information. With each email domain that you create in Zimbra, there is specific configuration information included, such as how does authentication occur for this domain; where does the global address list reside for this domain; and Class of Service information, where you can group users by features.

LDAP can also look up email delivery addresses both from internal and external LDAP servers as well. Zimbra supports the proxying of user login and Global Address List (GAL) access to an existing enterprise directory such as Microsoft Active Directory or other LDAP-compliant directories. The ideal configuration seems to be store Zimbra specific configuration data within the Zimbra managed, embedded OpenLDAP and store independent enterprise configuration data within the existing enterprise directory. To provide scalability and redundancy, the Master LDAP server can be horizontally scaled by deploying multiple replica servers or be configured in a multi-master replication mode.

MTA

The MTA service is responsible for receiving email from the internet and delivering it to mailboxes in the Zimbra environment. It also delivers email sent by Zimbra users out-bound or to other internal users. It serves in the Zimbra architecture as a relay point for archiving.

Internally, Local Mail Transfer Protocol (LMTP) is used to route the emails to the appropriate Zimbra mailbox server. The Zimbra MTA server includes the following programs:

- Postfix MTA, for mail routing, mail relay, and attachment blocking
- ClamAV - Anti-Virus engine
- SpamAssassin - Spam filters
- Amavis - interface between Postfix and ClamAV/SpamAssassin

In the Zimbra configuration, mail transfer and delivery are distinct functions. Postfix primarily acts as a Mail Transfer Agent (MTA) and the Zimbra mail server acts as a Mail Delivery agent (MDA).

Most SME/SMB or larger enterprises will require a 3rd party AS/AV solution that is more enterprise grade or carrier grade. The Zimbra AS/AV is turned off or only some of the features are used. SpamAssassin is an open source project and does not have the fine grained administration features for better control and flexibility.

As the user base grows and domains become popular, the environments is more susceptible to spammers. An enterprise or carrier grade solution becomes mandatory.

Mailboxd

The mailboxd process is where all the hard work is done. It controls everything from presenting the web client to users, so they see their mailbox data, to responding to other mail client requests for POP and IMAP and delivering the mail to those environments. It is responsible for storing messages on disk and providing indexing for those messages. It also maintains the MariaDB database that has the information for calendar, contacts, and tasks.

One of the differences post the Zimbra 8.5 architecture is that we split out the mailboxd process. You now have the option of running static content separately from dynamic content. There is a mailboxd user interface node option in addition to the traditional mailboxd process, which includes the message store, the database information, and the indexing information. This is optional. You do not have to split out these two components.

Proxy

The Zimbra Proxy is a high performance POP/IMAP/HTTP proxy server that allows end users to access their Zimbra account using end clients such as Chrome/Firefox/IE/Safari, Microsoft Outlook (Windows and Mac), Mozilla Thunderbird, or other POP/IMAP end client software.

Traditionally, we separate what is exposed to the internet and what is behind a firewall. The proxy server and the MTA server traditionally live in what is known as the DMZ or demilitarized zone, which is a security zone that is exposed to the internet. The proxy server listens for requests from the client and then translates across different ports, communicating with the mailboxd servers on the backend. This provides a layer of security on the backend. The proxy service is listening on the traditional protocols of HTTPS, IMAPS, and POP3S, which are secure ports 443, 995, and 993. It translates the incoming requests to different ports: the mailboxd process in not listening on port 443, it is listening on port 8443; it is not listening on port 993, it is listening on 7993; it is not listening on port 995, it is listening on port 7995. This becomes a layer of security, with the proxy service out front and the mailboxd processes separate.

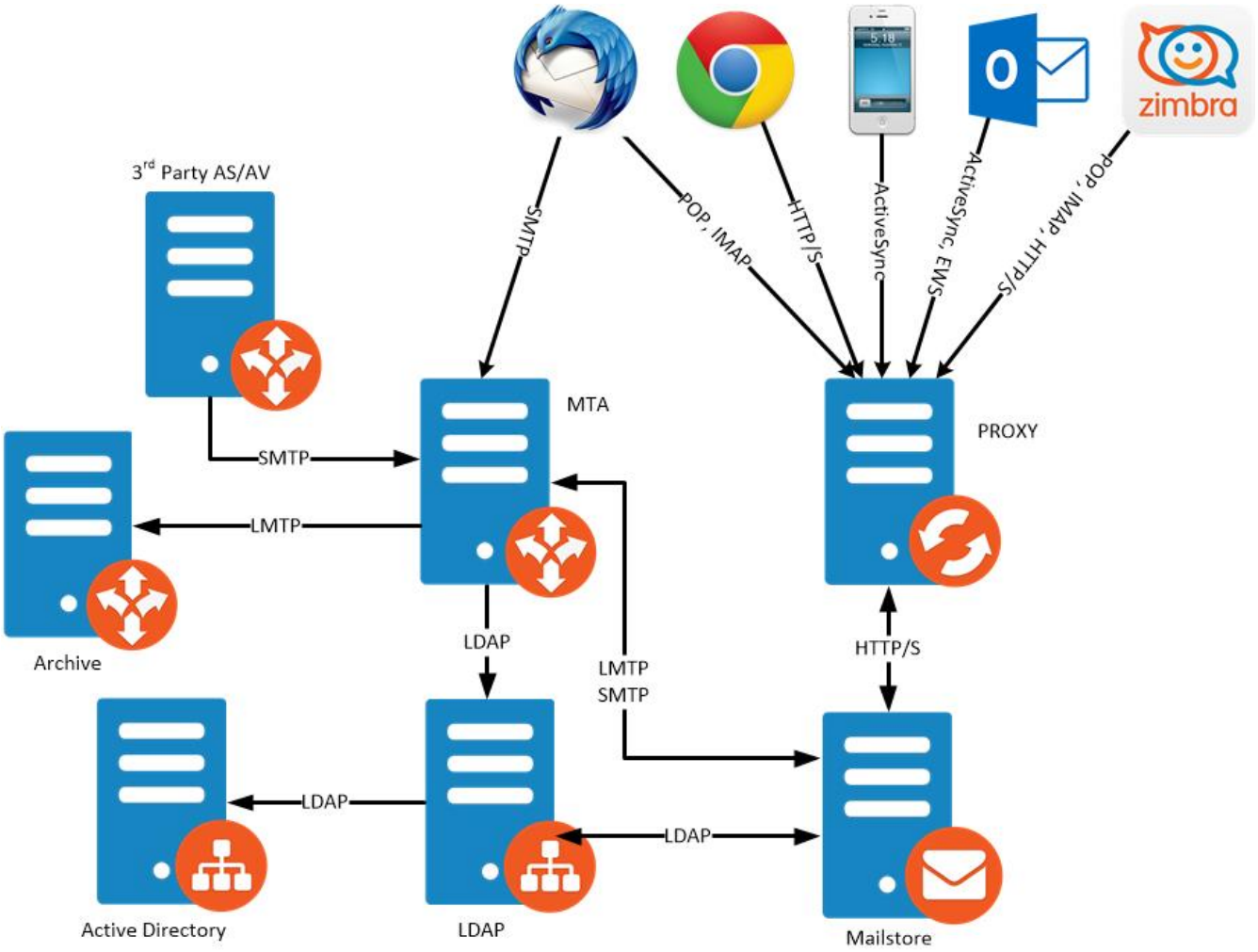
Proxying allows users to enter *imap.example.com* as their IMAP server, rather than remembering the actual mailbox server the user has been provisioned on. Encapsulation provides a layer of security and the proxy does a lookup to determine which backend mailbox server a user's mailbox lives on and transparently proxies the connection from user's client to the correct mailbox server.

In addition to IMAP/POP3 proxying, the Zimbra proxy package based on NGINX is also able to reverse proxy HTTP requests to the right backend server. Using an Nginx based reverse proxy for HTTP helps to hide names of backend mailbox servers from end users. For example, users can always use their web browser to visit the proxy server at https://mail.example.com. The connection from users' whose mailboxes live on mbs1.example.com is proxied to mbs1.example.com by the proxy running on the mail.example.com server. Clients such as REST and CalDAV clients, Zimbra Connector for Outlook, and Zimbra Mobile Sync devices are all supported by the Zimbra Proxy.

Client Access

Zimbra features compatibility with Microsoft Outlook (both Windows and Mac), Apple Desktop applications, and all other standards based POP/IMAP/iCal/CalDAV/CardDAV clients. Our broad desktop compatibility gives end-users freedom of choice and administrators the ability to protect their desktop investments because mixed PC, Mac, and Linux desktop deployments can all talk to the same Zimbra Server. The mail flow diagram below shows some of the more common methods of access and indicates the protocols used to interact with the Zimbra Mailstore.

The Zimbra Connector for Outlook (ZCO) provides real time two-way synchronization of mail, contacts, tasks, and calendar between Outlook and the ZCS server. Outlook for Mac works similarly using the EWS (Exchange Web Services) interface with the ZCS server. Standards-Based clients such as Mozilla Thunderbird, Sunbird, and Eudora can be used with Zimbra to access email and even calendar data because the Zimbra Server uses an all standards-based approach and supports POP, IMAP, iCal, CalDAV, RSS, etc.



Mobile Access

Zimbra Mobile for smartphones enables two-way, over-the-air synchronization of mail, contacts, calendar and tasks data between the mobile device and the Zimbra Server. It features push email, which sends messages in real time to your device when it arrives on the Zimbra Server. Supported devices must be ActiveSync compatible, for example Apple iPhones, Android smartphones from Samsung, HTC, etc.

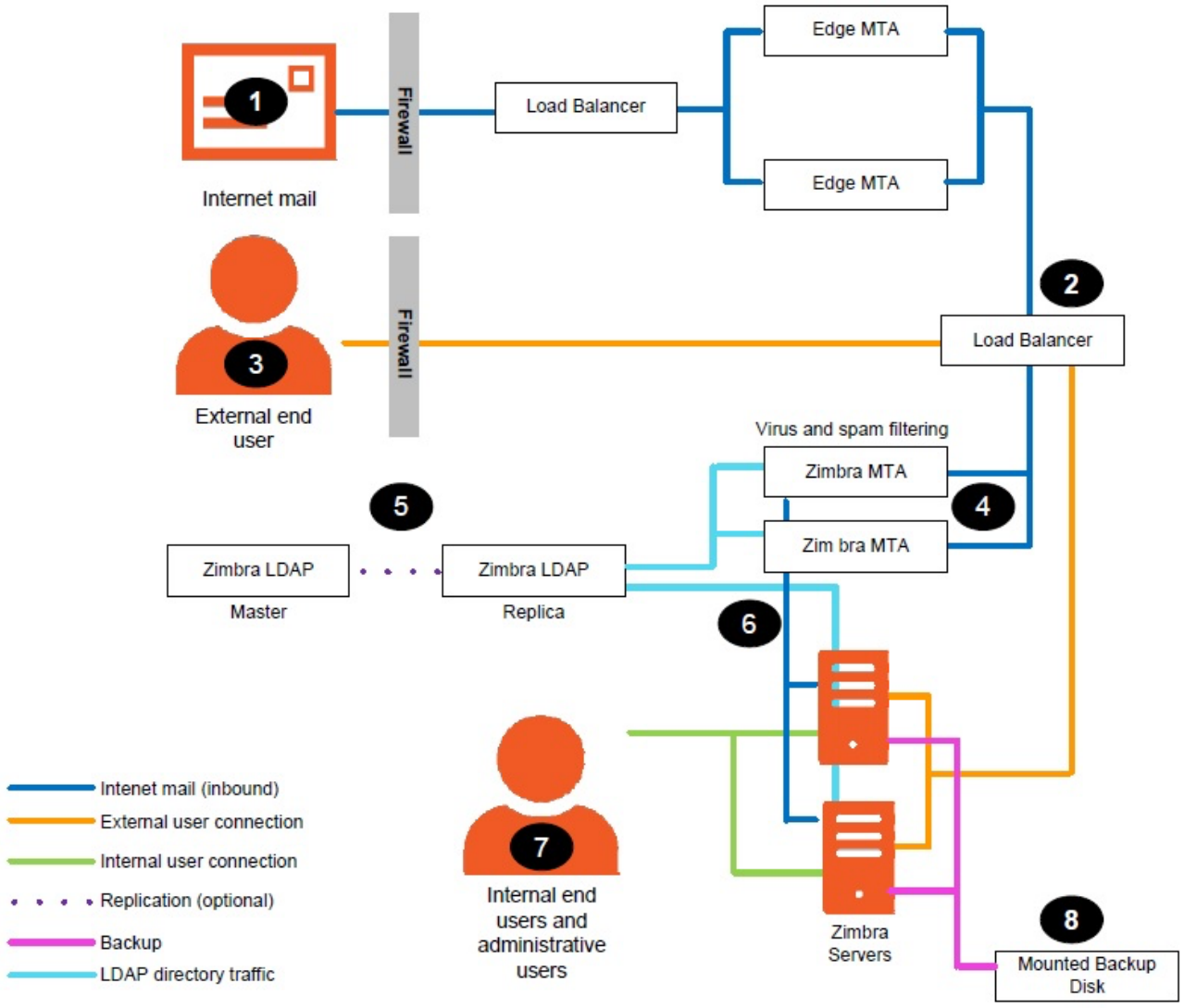
Zimbra's mobile access is further enhanced via its Mobile Web Client.

- Mobile Web Browsers - All devices with HTML capable browsers, have real time access to the Zimbra Server using our Mobile Web Client. Zimbra's Mobile Web Client allows users access to their email, contacts, and calendar. This provides on-the-go access to the Zimbra experience to virtually all end-users.
- Responsive Design – Zimbra web client adapts itself to the device being used. Tablets have a new layout based on the Sencha framework and the smaller mobile browsers use an xHTML format.

Mail Flow in a Multi-Server Configuration

The configuration for each deployment is dependent on numerous variables such as the number of mailboxes, mailbox quotas, performance requirements, existing network infrastructure, IT policies, security methodologies, spam filtering requirements, and more. In general, deployments share

common characteristics for incoming traffic and user connectivity, as depicted in the following diagram. Alternate methods for configuring numerous points within the network are also possible.



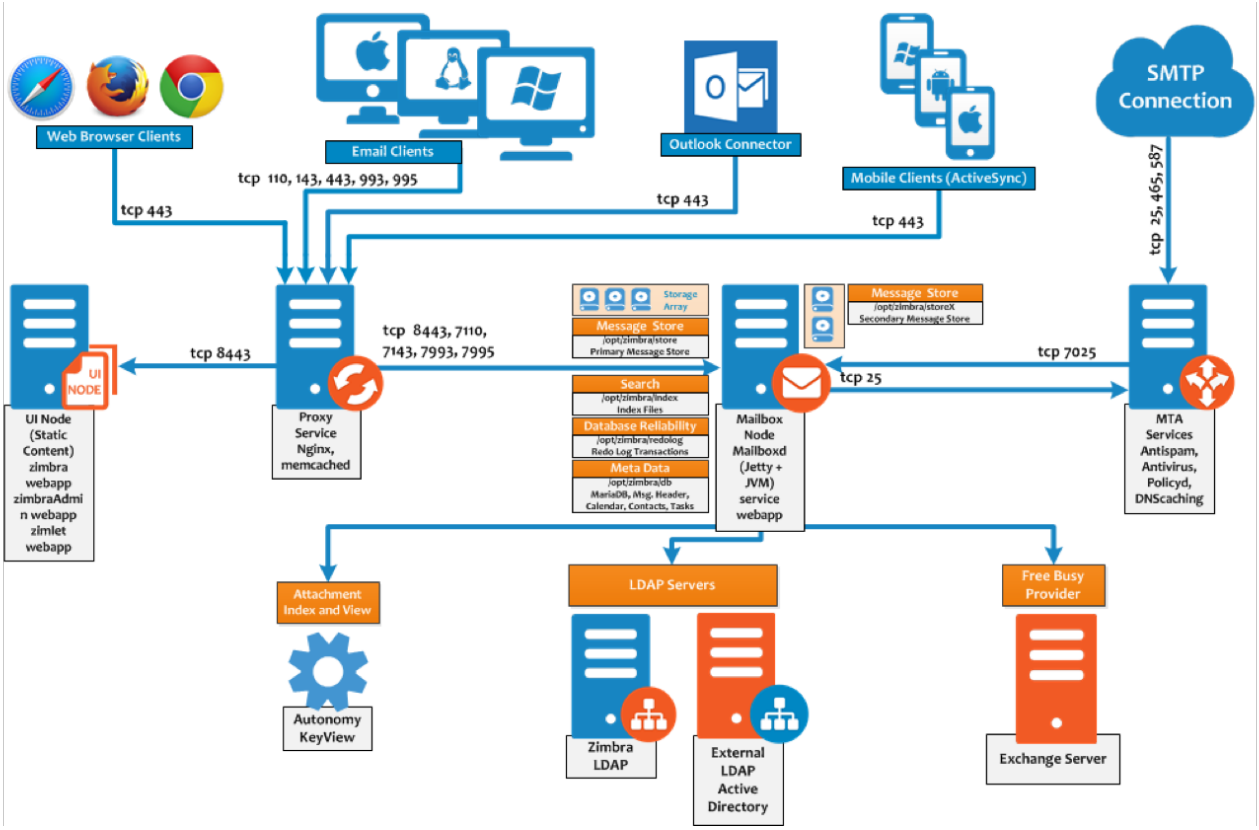
The numbered sequences are described below:

1. Inbound Internet mail goes through a firewall and load balancing to the edge MTA for spam filtering.
2. The filtered mail then goes through a second load balancer.
3. An external user connecting to the messaging server also goes through a firewall to the second load balancer.
4. The inbound Internet mail goes to any of the Zimbra Collaboration MTA servers and goes through spam and virus filtering.
5. The designated Zimbra Collaboration MTA server looks up the addressee's directory information from the Zimbra Collaboration LDAP replica server.
6. After obtaining the user's information from the Zimbra Collaboration LDPA server, the MTA server sends the mail to the appropriate Zimbra Collaboration server.
7. Internal end-user connections are made directly to any Zimbra Collaboration server that then obtains the user's directory information from Zimbra Collaboration LDAP and redirects the user, as needed.

8. The backups from the Zimbra Collaboration servers can be processed to a mounted disk.

Zimbra Architecture

This diagram represents the Zimbra architecture.



The mailbox server is running Jetty, the Java Virtual Machine, and this is the service node for the mailboxd architecture. On the backend, it is handling everything from storing the messages on disk to all of the other functions shown.

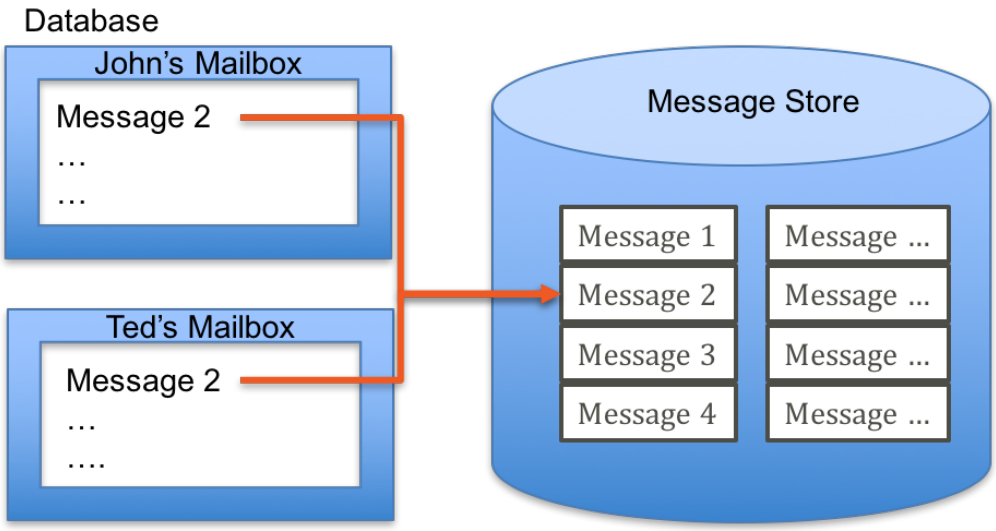
Zimbra was built around SOAP requests: the browser clients issue a SOAP request to the mailboxd process to retrieve information and that information is delivered back to the browser client.

Message and File Store

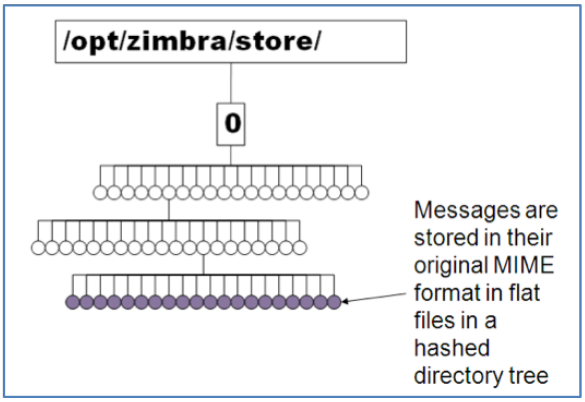
Zimbra stores messages on disk. Messages are stored in the **RFC 822** format. If you have access as a system administrator, you can go to /opt/zimbra/store directory and drill down to look at the actual messages stored in MIME format on disk. A benefit of this storage architecture is that LINUX does single-instance storage: if I send a message to 3 different people, and we all share the same mailbox server, then there is only one instance of that mail item in the store. LINUX maintains multiple points to that mail item.

The data store is a MySQL database where internal mailbox IDs are linked with user accounts. The data store maps the mailbox IDs to users' OpenLDAP accounts. This database contains each user's set of tag definitions, folders, calendar schedules and contacts, as well as the status of each mail message - read, unread, tags associated to message, and folder the message resides in. Index and search technology is provided through Lucene and index files are maintained for each individual mailbox.

The other enhancement in the Zimbra 8 series is that the storage system can be a different type of file system, such as a global file system like Scality or EMC. Many Zimbra customers implement Zimbra over object store, so the message blobs are stored on a global array across multiple spindles and multiple pieces of hardware. It provides more redundancy and availability for the message store.



Directory structure of the Message Store



The Zimbra message (or blob) store is built on the underlying Unix/Linux file system. The mapping is one file per message, we actually write the RFC822 MIME message representation directly to a file. This has a number of benefits:

- Once written, messages are immutable in Zimbra, and file systems are very efficient at storing and retrieving immutable unstructured/semi-structured blocks of data.
- Garbage collection is provided natively by the operating system, which offers both greater efficiency and lower administrative overhead.
- Independent utilities, tools, and scripts can easily recognize and process the Zimbra message representation on disk.
- Native operating system capabilities such as indexing/search (e.g., Mac Spotlight), compression, and security.
- Multi-level caching. Zimbra caches data itself, but Zimbra also benefits from the underlying operating system caching. Since enterprise messaging systems are generally gated on read and

write performance, effective caching is one of the critical foundations of a scalable messaging implementation.

For each instance of the Zimbra mailbox server (one instance per machine), we store one copy of each message including all its attachments, even if that message is destined for multiple mailboxes. This obviously can lead to substantial savings in disk if large attachments are often sent to multiple recipients on a server (some messaging servers only provide single copy per mailbox or single copy per storage group, which can lead to significant redundancy in more expensive managed storage with no improvement in availability/fault tolerance). For operating systems that support them, we utilize hard links to the single copy to efficiently ensure that a file is garbage collected after all users have deleted it from their mailbox metadata.

Metadata SQL Store

While messages themselves are immutable, the meta-data associated with a message goes through frequent state changes as it is used. Message meta-data includes such items as

- What folder is the message in?
- What tags does the message have?
- Is the message read or unread?
- What conversation is the message part of?

We are storing the header information of every message. When you go into the web client and open your mailbox, you see the header information (who the message is from, when it was received, etc.). That is all meta data stored in the MariaDB database. When you click on the message to display it, Zimbra reads the message from the storage system.

Zimbra includes this MariaDB database for managing mailbox meta-data, and it is shipped within each Zimbra distribution. We choose to use a relational database for Zimbra meta-data because:

- We sought to leverage the hardening and performance investment of smart engineers before us. For efficient searching and reliably updating highly structured meta-data, it is hard to beat relational databases.
- Caching - Unlike immutable messages, Zimbra's meta-data is both frequently read and written. Relational databases provide very efficient caching for both reads and writes. Update caching efficiency comes out of combining multiple updates, even those spanning transactions, into a single disk write (this optimization, of course, leverages the database's sequentially-written transaction log to ensure that no changes are lost in the event of an outage).
- The SQL interface provides the Zimbra Community with investment protection in that the ZCS mailbox server could be very easily ported to other relational databases from the MySQL implementation that is currently included in ZCS.
- Database Reliability: Every transaction that occurs in Zimbra is stored in the Zimbra redo log. That includes everything from receiving a message in your inbox, reading that message, moving that message to another folder, deleting the message, adding a calendar appointment, adding a new address book, etc. All of these things are transactions, and all transactions are stored in the Zimbra redo log, which is handled by the mailboxd process.

Search

When a message comes into Zimbra, that message is indexed with the Lucene indexing process. This provides the basis for the very fast search in Zimbra, including the ability to search across multiple criteria.

Search is fundamental to the Zimbra. Zimbra users do need to rely on complicated folder hierarchies to organize their emails because Zimbra users can search almost anything instantaneously across any folders. This enables usability advantages as users can archive emails into a single folder. All the meta-data associated with a message (folder, read/unread, tags, etc.) is available as search criteria, as are indexes into the content of the messages and the content of any attachments to the messages. Zimbra provides a simple, yet rich grammar for specifying searches, and then offers a graphical search builder for constructing advanced searches without having to learn that grammar.

Zimbra search is conducted on the server side. The advantages of server-side search are access to full mailboxes (including archived content), reliance on faster disk/CPU for increased performance (even factoring in the network latency), and the elimination of a substantial amount of redundant computation associated with indexing attachments sent to multiple recipients.

Zimbra leverages Apache Lucene to manage the index that enables fast searching, and Zimbra uses third-party software from Verity to extract text from attachments for indexing within Lucene.

Lucene Index

Lucene is a high-performance, full-text search engine from Apache. Lucene works by generating a full “segment” index for all words/tokens in a particular message, and optionally, its attachments. This segment is then merged into the receiving users’ existing index (one index per user). The index itself is represented in flat files. Search simply traverses these optimized file structures, often in parallel.

Search is very fast—users perceive it to be nearly instantaneous. The pre-processing required to construct an index, on the other hand, grows linearly with the size of the text. Attachments tend to be the overriding factor in this overhead. Attachment indexing is a function of user’s class of service, and so can be turned on or off based on server scaling requirements. Message text indexing is more essential to the user experience (as well as significantly lower-overhead due to the smaller datasets)—it is a big part of the reason why many Zimbra users become comfortable with dropping their complex, time-intensive folder hierarchies in favor of a single “Stash” folder, because powerful search frees them from the worry that they will not be able to find what they are looking for.

We have found this Lucene index to typically be about 20% of the size of the text being indexed. Simply by compressing messages and their attachments prior to storage, we can make up the space required to store the Lucene index. As a user’s index grows larger, the savings increase, since there is greater reuse of tokens. Garbage collection is done on the indices only after expunging messages (emptying the trash), so that trashed messages can still be found (many Zimbra users choose to use their trash as a secondary archive folder, since no message data need be expunged from the trash until the user’s quota is reached). Should a more catastrophic failure lead to an index corruption;

there is also an administrative interface for regenerating Lucene indices (which are, of course, idempotent).

Other Services and Access Methods

- **Attachment Index & View**: With the Zimbra Network Edition, and you have attachment indexing and view capabilities turned on, and that process is being provided by the mailboxd process. Convertd is the pricess wiht the attachment index and viewing capability. If I send you an email with a PowerPoint presentation, the Autonomy Keyview process adds the attachment to your search index, so you can perform an email search that also searches the attachment.
- **Free/Busy Providers**: The last process that is being handled by the mailboxd process is the integration with free/busy providers. If you are migrating from MIcrosoft Exchange to the Zimbra platform, and you have some users on the legacy system and others on the new Zimbra system, and you want them to be able to update calendar information in both directions, that service would be provided by the free/busy provider within the mailboxd architecture.
- **Lite Browser Clients**: It is possible to use Zimbra in HTML mode, where the information is displayed in HTML. The AJAX client, built on the AJAX framework, provides drag-and-drop functionality and other things that users are accustomed to in the browser.
- **Desktop Clients**: We have the Zimbra Desktop. We also have Microsoft Outlook, which connects to the Zimbra environment over POP or IMAP. You can also install the Zimbra Connector for Outlook. This is part of the Network Edition of Zimbra. The Zimbra Connector for Outlook provides the integrated functionality of calendar, contacts, and tasks from within Outlook, which makes Outlook think that it is talking to an Exchange server. When Outlook communicates with an Exchange server, it is using the MAPI protocol. The MAPI protocol is a Microsoft API. Zimbra does not listen on that API, so we have to translate those requests from MAPI commands to SOAP commands over HTTPS.

  We also have the ability for other clients like Thunderbird, the Apple mail client, or the Apple calendar to communicate with the Zimbra mailboxd process. We can do this over CardDAV or CalDAV or POP or IMAP.

- Zimlets: Zimbra provides the ability to extend the web browser interface to integrate services from external web service providers. One Zimlet example is the Yahoo Finance Zimlet, which you can use to present a stock ticker in Zimbra that updates every 20 minutes. That information is coming from an external web service. From a security stand point, it is not recommended to present information in a client from across domains. The mailboxd process has something called the Zimlet web service proxy. When the browser running the Zimlet needs to update stock quotes, the request is made to the mailboxd process, the mailboxd process goes to the Internet and pulls that information from the external web service, and then delivers it back to the browser client. You can extend this to any external web service. If you are a manufacturing organization and you need to see order information displayed in email clients dynamically, you can write a Zimlet that goes out and pulls that information from your ordering management system and presents the information in the browser client without the user having to log into a separate system.

  We also have the ability to create Zimlet JSP tags that communicate information back to the browser via the mailboxd interface. The other point to make is that when users of the desktop

clients and browser clients send email, they connect to the mailbox server, they compose email in the client, and then they click send. The mailboxd process forwards that information to the postfix MTA service where antispam and antivirus is performed, and then it is forwarded to the endpoint.

High Availability and Scalability

High Availability

Single points of failure should be avoided wherever possible, at all levels: network, server, power supply, cabling, disk drives and software elements. This should apply to ancillary systems (e.g. DNS and External Authentication Servers) as well as to the email servers themselves.

LDAP Master

The LDAP Master will have its database stored on the servers' local disks. Redundancy is provided via the deployment of LDAP Replica servers. The replicas contain a full copy the master database and in the event of failure of the master server the replica can be promoted to master status and the service can continue. Adding replicas also facilitates the distribution of load on the LDAP services; balancing the load across the master and replicas is configured internally within the ZCS configuration. Zimbra also supports a Multi-Master mode where there can be more than one master LDAP server eliminating the need for replica servers.

Mailstore

Zimbra is designed to be deployed in a virtual environment. VMware, KVM are the popular VI environments supported. Zimbra works best with VMware vSphere and can leverage vSphere HA for the mailstore high-availability. Active-Active configuration is not supported today. Zimbra can also work with traditional OS clustering (RedHat) but there is no application level failover supported and this works best for a hardware failure.

MTA and Proxy

Both the MTA and Proxy servers are stateless services and can run in an Active-Active mode behind a load-balancer.

Scalability

Zimbra was designed from the ground up to scale to meet the needs of businesses with 100,000s of users and Service Providers with millions and even tens of millions of users. The Zimbra architecture inherits from distributed systems expertise that was gleaned building messaging systems that today host more than 100 million mailboxes world-wide and Java/Web systems that have thousands of production server CPUs within single large-scale deployments.

All medium and large Zimbra deployments are horizontally partitioned across servers (and the attached storage) by end-user mailboxes. An end-user's mailbox includes his or her messages, calendar(s), address book(s), documents, and so on, which are all collocated for very efficient context switching and search across applications. So ZCS servers are inherently stateful - each serves as the primary home for a subset of the aggregate mailboxes. This requires that each ZCS server have the smarts to reroute a protocol request (via XML/SOAP/HTTP/S, IMAP/S, POP/S, RSS, iCal, etc.) to the appropriate primary server in the event that an in-bound load balancer makes the wrong decision.

Automated replication and failover is also essential for large-scale deployments. For example, LDAP configuration data (which includes end-user mailbox home locations) can be fully replicated/partitioned across as many replica servers as are required to meet performance and availability requirements. LDAP replicas may be collocated with ZCS mailbox servers, MTAs, or "vertically partitioned" to dedicated servers.

Mailbox data, on the other hand, can be transparently replicated within the underlying storage system (such as by using RAID or mirroring) for availability only. (It simply does not make sense to replicate mailboxes for scalability, given how frequently state data is updated, and the overhead of ensuring transactional consistency between multiple mailbox copies.) Clustering technology is used to automatically failover from the primary server to a preconfigured secondary (or tertiary) server, which then assumes the role of primary for that mailbox. (I/O fencing and associated technologies ensure that the former primary no longer has "write" access to the mailbox in order to avoid "split-brained syndrome").

ZCS Meta-data optimization & partitioning; Meta-data for a mailbox is generally all of the data required for navigating to the appropriate message or meeting. Zimbra meta-data includes ZCS's very efficient, Lucene based index into all the text contained in every message, meeting, contact, document, attachment, and so on. Zimbra meta-data also includes the structured meta-data that captures folders, tags, dates, read/unread status, etc. Zimbra uses an off-the-shelf SQL database for optimizing structured meta-data queries and updates.

Meta-data is horizontally partitioned by user into the appropriate ZCS mailbox server, but the key additional insight is that this meta-data can also be partitioned from the target data (message body, meeting, document) to ensure very efficient search, UI painting, and so on. Separating the meta-data and target-data makes it far more cost-effective to keep the meta-data on fast disk, allowing sophisticated search and navigation to be nearly instantaneous even across multi-gigabyte mailboxes. Latency in access to the message body itself (which could, for example, reside in an HSM system) is not nearly as problematic to the user experience as latency or inefficiency in accessing the meta-data. Partitioned meta-data also allows potentially expensive operations such as compliance-related cross-mailbox discovery to be handled efficiently (via simply composing the appropriate horizontally partitioned search results).

Proposed Project Approach

This section details how Zimbra professional services handle a project to ensure a smooth transition from the existing legacy platform to full Zimbra Collaboration Suite. This section serves as a guideline only and the approach needs to be modified and made suitable as per the customers requirements. Here are some of the processes and phases mentioned.

Assessment and Solution Design

Site assessment and auditing means identifying the current infrastructure and business goals of the customer. This includes getting information about the current platform topology like hardware/storage/network capabilities, service SLA's, getting information on the current issues, etc.

- Review the current environment and examine all the business and technical issues.
    - Get more information on the server, storage and network infrastructure
- Formulate problem definitions and document the objectives.

Solution design and service planning will include development of the Baseline Architecture, Low-Level Design ("LLD"), hardware Bill-of-Materials ("BOM") and development of customizations and documentation for all of the above.

- Develop an understanding of the customer's existing architecture and technical challenges or issues.
- Document the project requirements, phases and communication plans with the customer's product and technical staff
- Create a project definition including roles, dependencies, schedule estimates, communication plans, etc.
- Create an integrated application architecture that is both flexible and scalable.
- Present a solution design methodology that is agile.

Zimbra Deployment and Configuration

This phase will involve the actual deployment of all the different Zimbra components like LDAP, MTA, Mailbox services based on solution design.

- Execute the project according to the documented and approved requirements and specifications.
- After installation, configure the platform as per the design
- Integrate with existing gateway services, anti-spam, antivirus and other 3rd party dependencies etc.
- Configuration of the dependencies and the Zimbra platform as per the requirements

A good resource for deployment can be found here - Performance Tuning

Testing

Based on the requirements, a functional test case document needs to be ready for this phase. An UAT or functional testing ensures that there are no bugs/issues with the platform for regular day-to-day use and the customer has all their challenges addressed before the environment is made production ready. Testing may include the use of in-house or third party tools and the level of testing always depends on the customer requirements.

- Perform testing and quality assurance via functional, performance, and end-to-end testing.
- Manage or assist with any pilot/user acceptance testing.

Migration

Migration is most critical part of any mail service. Once all the required functional testing has been performed, its time to migrate the data from old legacy platform to Zimbra platform.

Zimbra professional services can help with the design, planning, scripting, and/or testing of migration of all legacy data (email, calendars, contacts, filters, etc) to the new Zimbra platform. Zimbra services can help design the best-suited migration approach for the customer based on inputs on the current setup and migration plans and also help develop the migration methods based on the migration plan, which may include migration via all-at-once/big-bang or incremental, split-domain methods. Zimbra services will also develop the necessary scripts to carry out migration and run the tools for a subset of users. Customer will run remaining subset when they gain enough knowledge and understanding.

Zimbra Professional Services is not a bundled offering and is a separate paid service offering.

Audit and Operations

The goal of this phase is to deliver an operational support model that will streamline production operations, provide proactive issue detection, and provide overall service excellence. Once the environment is live, there must be a process of regular auditing of the environment and its configurations to ensure that the system has been tuned for optimum performance. All fo the Zimbra statistical information is collected by the "stat" process and logged. This data can be analyzed further and decisions can be made on how to improve the system performance.

Refer to this ZCS Best Practices reference guide for more information.

Appendix A: ZCS Firewall Ports

Inside Firewall Ports

Port	Function
22	SSH for Remote Server logs/Management
25	SMTP to MTA from internal AS/AV service
389	MTA to LDAP Authentication
514	(r)syslogd from ZCS servers to Logger (TCP/UDP)
7025	LMTP from MTA's to Mailbox Servers
7071	Zimbra Administration
7072	Zimbra Route Lookups
7110	Pop Proxy to Mailbox Servers
7143	IMAP Proxy to Mailbox Servers
7993	IMAP Proxy to Mailbox Servers (SSL)
7995	POP Proxy to Mailbox Servers (SSL)
8080	Proxy to Mailbox Server
8443	Proxy to Mailbox Server if using SSL at Proxy
8443	Mailbox Server to Zimbra Docs (both ways for HTTPS backend)
9091	incoming traffic from all mailbox and proxy servers to Docs Server Extension
9071	Zimbra Administration if using Admin Web Interface through Proxy

Outside Firewall Ports

Port	Function
25	SMTP to Internal AS/AV from external mail servers
80	Connect to Zimbra Mail (if SSL offloaded by Load Balancer)
110	POP to Proxy server (if SSL offloaded)
143	IMAP to Proxy Server (if SSL offloaded)
465	SMTPS to Internal
587	Submission Email using TLS over SMTP
443	Connect to Zimbra Mail over HTTPS
993	IMAP SSL
995	POP SSL
5222	XMPP (TLS) - Direct to Mailbox required

Appendix B: Zimbra Talk V2

Zimbra Talk V2 is the newest addition to the Zimbra product line-up. Zimbra Talk V2 will be available in version 8.8.8 onwards. This is enterprise-level messaging and videoconferencing built inside the Zimbra web client. It includes one-on-one and group messaging and videoconferencing, along with full screen and file sharing capabilities.

Zimbra Talk V2 inherits some of the features from the previous Zimbra Chat, but the vast majority of the features are new. Some of the inherited features from Chat are (basic features):

- **1-to-1 text messaging**
- **Buddy List Management**
- **Chat history**
- **Emojis**
- **Status or presence management**
- **User preferences**

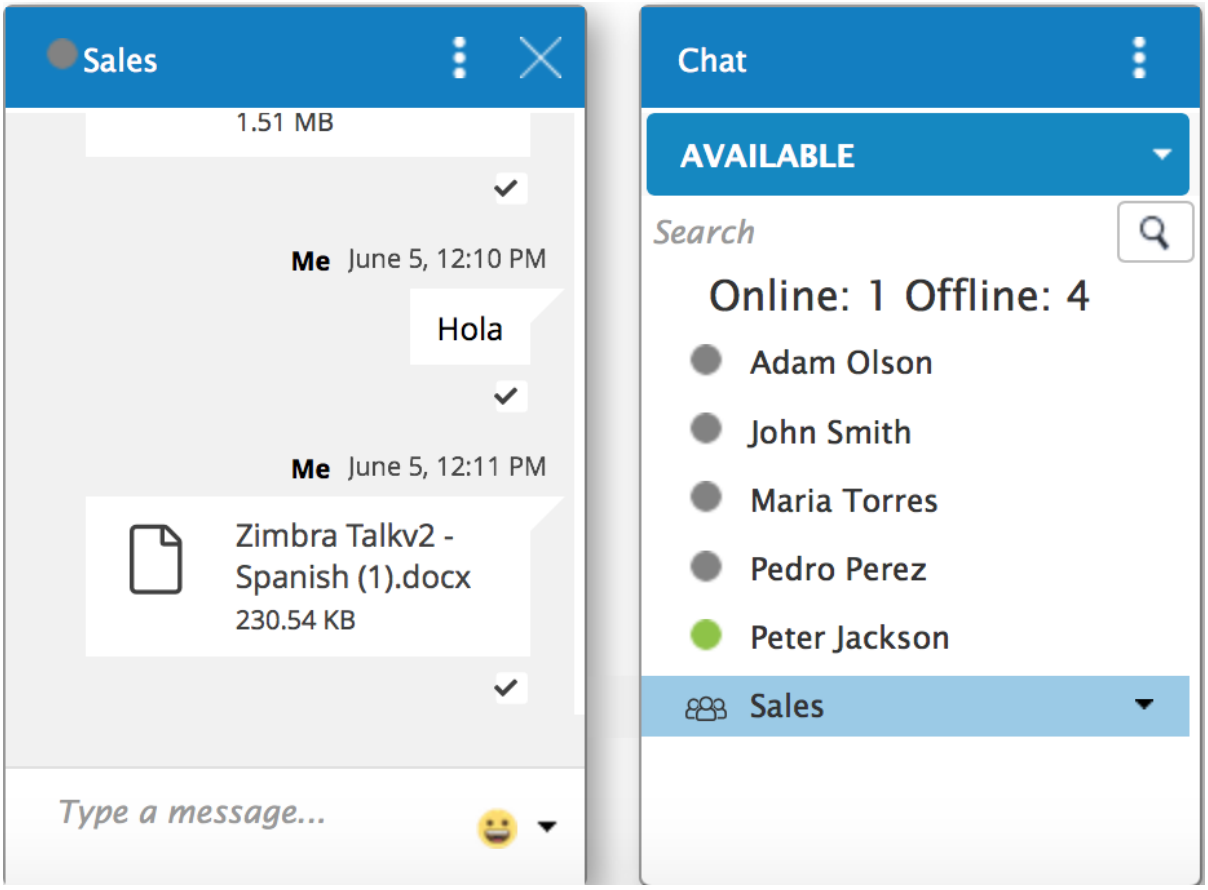
Zimbra Talk V2 provides advanced features. These features are:

- **Message Delivery and Read Awareness.** Check marks show delivery or read status
- **Group Messaging.** Secure and private messaging for small groups (up to 5 people)
- **Videoconferencing.** WebRTC based video, out-of-the-box without any plug-in
- **Corporate Communications.** Spaces and Channels for private or organization-wide communications
- **File Sharing**
- **Screen Sharing**

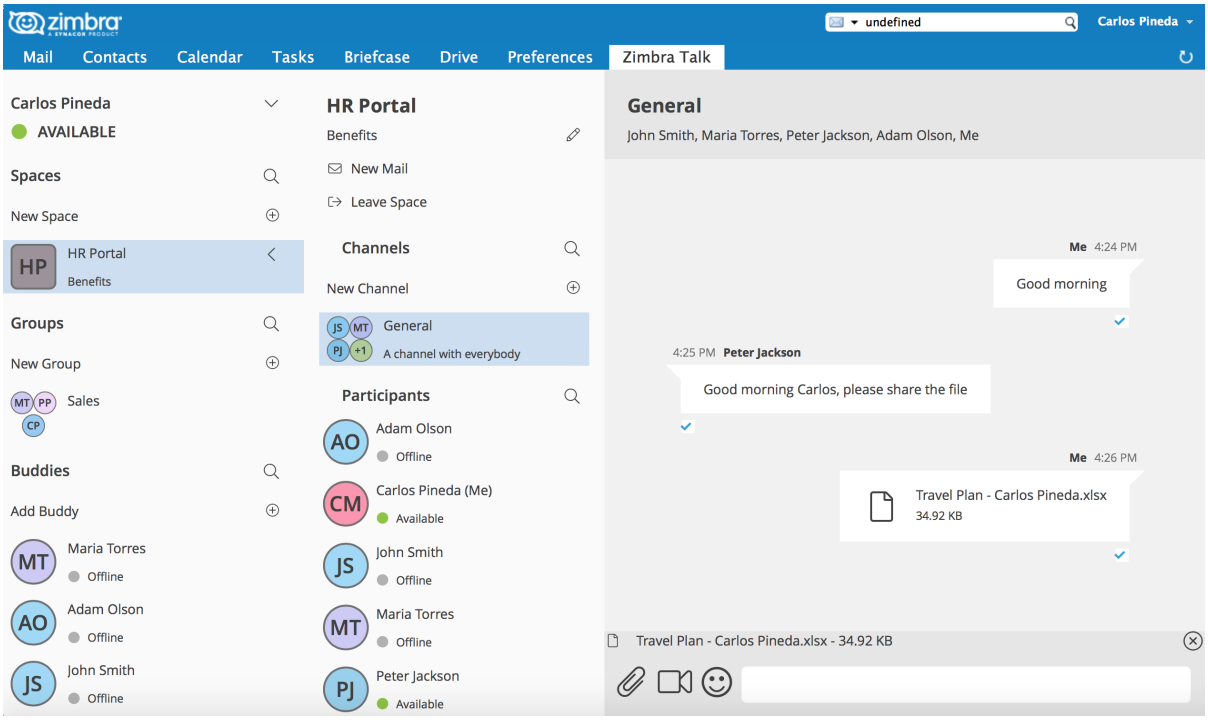
Beginning in Zimbra Collaboration Network Edition 8.8.8, the Zimbra Talk Zimlet will provide the basic chat features, these features will be available for all of your users. In order to enable the advanced chat and videoconferencing features per user, a purcharse of Zimbra Talk V2 license is needed.

Zimbra Talk V2 has two user interfaces: **Chat Panel and Zimbra Talk tab**.

The **Chat Panel** is available to all users - basic and advanced - for text messaging. Advanced users will have the additional options of videoconferencing and file sharing in the panel.



The **Zimbra Talk** tab is only available for advanced Zimbra Talk users. This tab offers all of the corporate instant messaging features such as Spaces and Channels.



Advanced Zimbra Talk users will have three options to interact with other users: Groups, Spaces, and Channel.

Groups

Groups are the basic way of communicating with multiple people at the same time (up to 5 total).

Those are non-persistent entities that are not tied to any specific space: any user can create a group inviting people from their Buddy List and any group member can invite more people in the same way. When all users leave a group, the group itself ceases to exist.

Groups Features

- A user in a Group can add more users to the Group itself up to the allowed limit
- A user in a Group can chat with all of the others. Messages sent in a Group are viewed by all members of that Group
- A user in a Group can send files to all of the others. Files sent in a Group are available to all members of that Group
- A user in a Group can start a videoconference with all of the others. Group videoconferences can be joined at any time by all members of the Group

Spaces

Spaces are a themed container that can hold any number of Channels. A Space is a community portal where people gather to discuss different topics in dedicated areas (named Channels).

Spaces Features

- Each space has a unique name and topic. The name cannot be changed after creating the space but the Topic can be edited
- Users in a space can send an email to all members of that space
- Members can leave a space at any time
- Members can create new channels and invite new people to the space

Channels

Channels are topic-defined areas inside of a same space. Those can contain any number of users, and unlike groups, users are able to autonomously join any Channel in a Space they are in instead of being invited to it by a member.

Channels Features

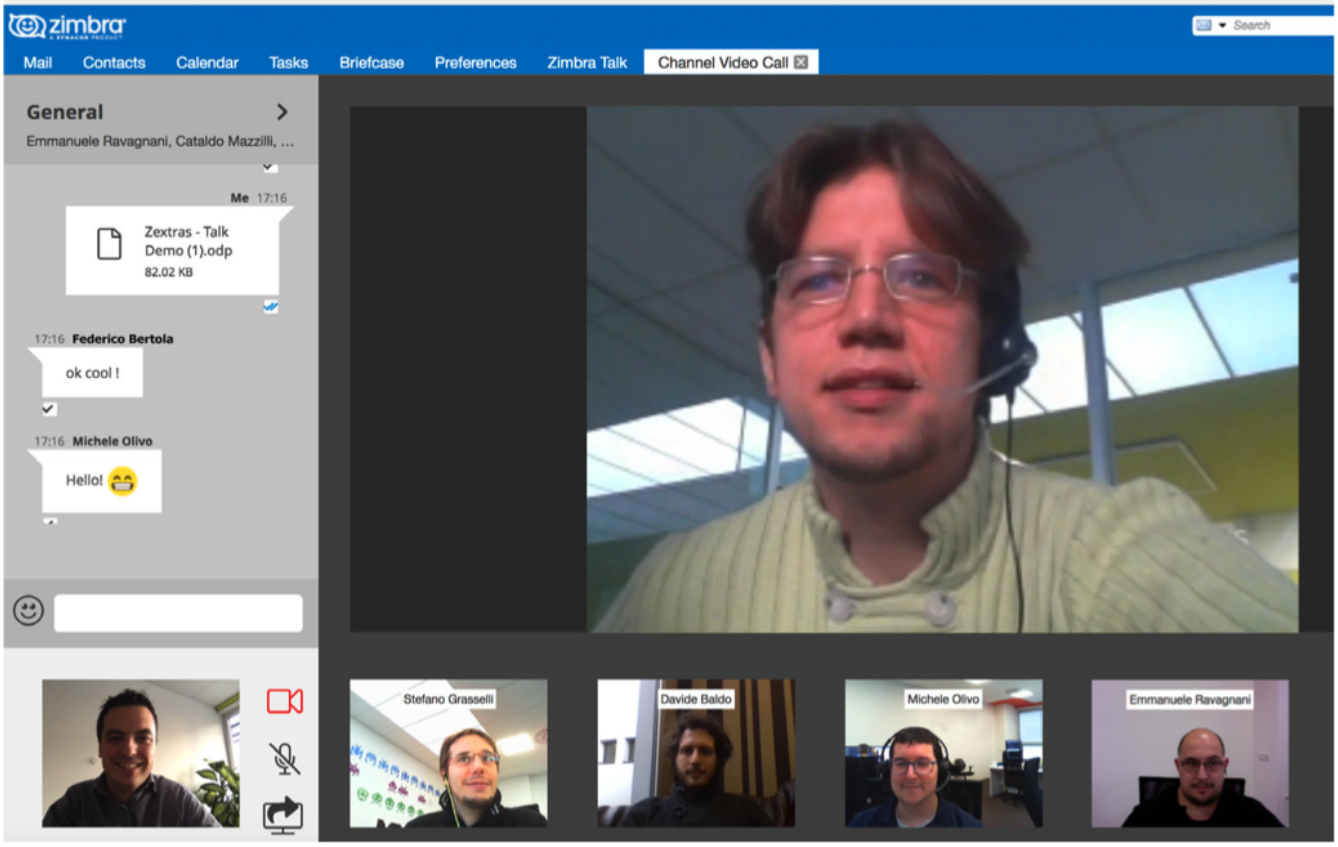
- A user in a Channel can chat with all of the others
- A user in a Channel can send files to all of the others
- A user in a Channel can start a videoconference with all of the others
- Channel videoconferences can be joined at any time by all members of the Channel

Videoconferencing

Videoconferencing features are available in both Groups and Channels, allowing multiple people to communicate in real-time using a webcam and a headset as well as allowing them to share their screen with all other attendees.

This feature is based on the WebRTC protocol, a peer-to-peer auto-adaptive technology that allows

clients to communicate directly without overloading the server and whose call quality is automatically tweaked based on the available bandwidth - with the maximum quality being Full HD for both video and audio. The first time a Videoconference is started, users will need to grant their browser access permissions to their camera and microphone.



Zimbra Docs

Zimbra Docs is based on a heavily customized LibreOffice online package allowing for collaborative editing of documents, spreadsheets and presentations straight from the Zimbra WebClient.

Components

Zimbra Docs Server

The Zimbra Docs server is the heart of the service. The service hosts each document opened through a LibreOffice engine and responds to the client via an image upon every keystroke and change in the document.

This component must be installed on one or more dedicated nodes running Ubuntu 16.04 LTS.

Zimbra Docs Extension

The extension is the key component which coordinates everything. Its main tasks are:

- Select which Docs server the next document will be opened on.
- Redirect the client when it needs to open a document.
- Read and write documents to and from system storage on behalf of the Docs server.
- Connect to each Docs server via an administrative websocket and keep track of the availability and the resource usage of each.
- Orchestrate concurrent user connections to the same document in the same server. (document sharing)

The Zimbra Docs Extension is contained within the NG modules package together with the NG Core.

Zimbra Docs Zimlet

A Zimbra Docs Zimlet handles the integration with briefcase and with email attachments. It is a thin web client which connects to a native server instance via websocket, renders a document and only sends changes to the client in order to keep the fidelity of the document on par with a desktop client while at the same time reducing the bandwidth to the bare minimum.

Documents in preview and attachments are shown in read-only mode with a simplified interface, while edit mode has a full interface.

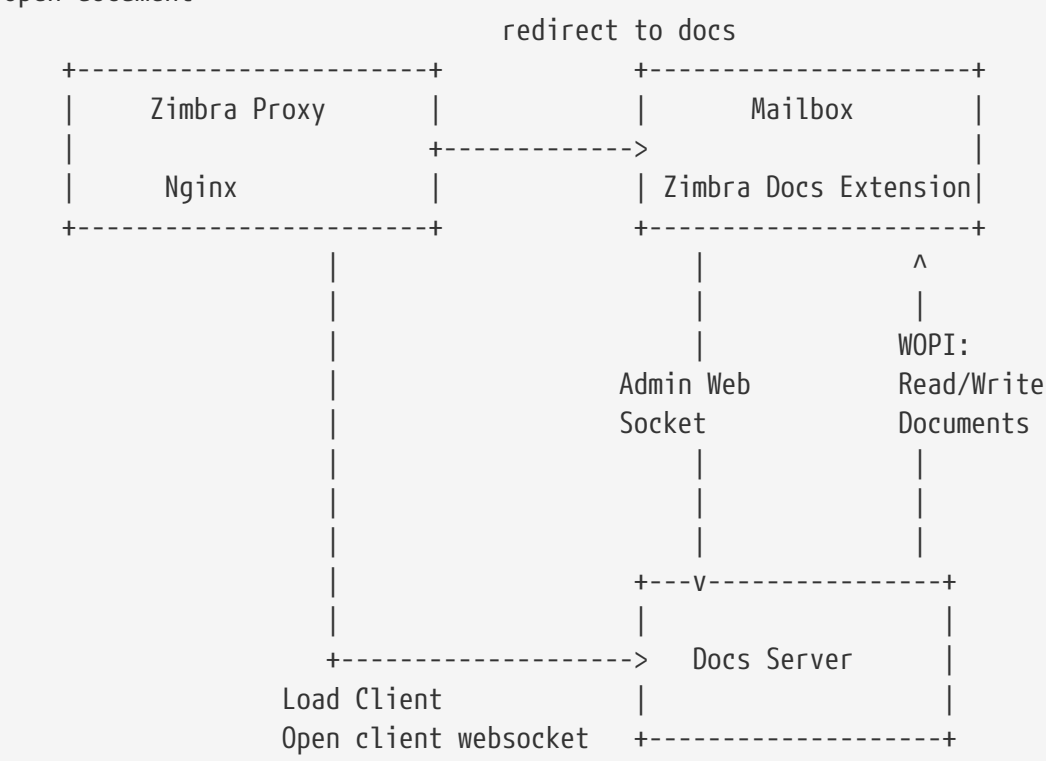
Its main tasks are:

- Change the "create" button in the ZWC to the related Docs feature.
- Change the preview feature to use Docs.
- Allow the preview of documents.

Document Management Flow

This is what happens "behind the scenes" when a user creates a new document:

1. The Zimlet prompts the Extension to create a new empty document
2. The Extension creates the document and returns the document's ID to the client
3. The Zimlet opens a new Zimbra tab containing an iframe pointing towards '/service/extension/wopi-proxy'
4. The extension receives the request from the client, creates a new token for the needed document, and replies with a new url
5. The new url points toward `/docs/[docs-node-id]/[token]`, which will be proxied by nginx to the specific Docs Server node
6. The Docs Server will respond with the web application in Javascript
7. The web application opens a websocket connection, going through the nginx
8. Docs Server receives the websocket connection along with a token, sends a `read wopi` command towards the mailbox url indicated in the parameters (the url is validates against allowed nodes)
9. The Extension validates the token and replies with information and content
10. The Docs Server node parses the document, renders it and sends it back to the client.
11. The document is fully opened and editable.



Networking and ports

All mailbox servers will need to be able to directly communicate with the Docs Server over port 8443 (HTTPS Backend), which must be open on both ends.

The Docs Server communicates with the Extension through port 9091, so incoming traffic from all mailbox and proxy servers to that port must be allowed. The Docs Server component must also be able to directly communicate with the master LDAP server as well as with all Proxy servers.

License

 Synacor, Inc., 2017

Synacor, Inc., 2017
40 La Riviere Drive, Suite 300
Buffalo, New York 14202

https://www.synacor.com