



Xythos WebFile Server Architecture
***A Technical Guide to the Core Technology, Components, and
Design of the Xythos WebFile Server Platform***

© Copyright Xythos Software, Inc. 2006
All Rights Reserved

TABLE OF CONTENTS

Overview	1
The Technology Behind the Server	1
Required Components	2
Web Server	2
Application Server	3
Database	3
File System (optional)	3
Load Balancer (optional)	4
LDAP Server (optional)	4
Hardware Requirements	4
Xythos WebFile Server Storage	5
Document stores and File Storage	5
External Storage Directory Structure	6
Single File Instancing	6
Global Schema	6
Backup and Recovery	6
Xythos WebFile Server Sizing and Scaling	8
Sizing	8
Database Sizing	8
Scaling	9
Application Tier Scaling	9
Storage Tier Scaling	10
WebFile Server Clustering and Fail-Over	10
Logical Architecture	10
Xythos Applications	11
Digital Locker	11
Enterprise Document Manager	11
Custom Applications	12
Java API	12
• File System Security Manager	12
• Event Listeners	12
• Portlet Platform	12
• File Content Search	12
• Virus Scan	12
• Storage Filter	12
• Workflow Engine	13
Web Services API (WebDAV)	13
Conclusion	13

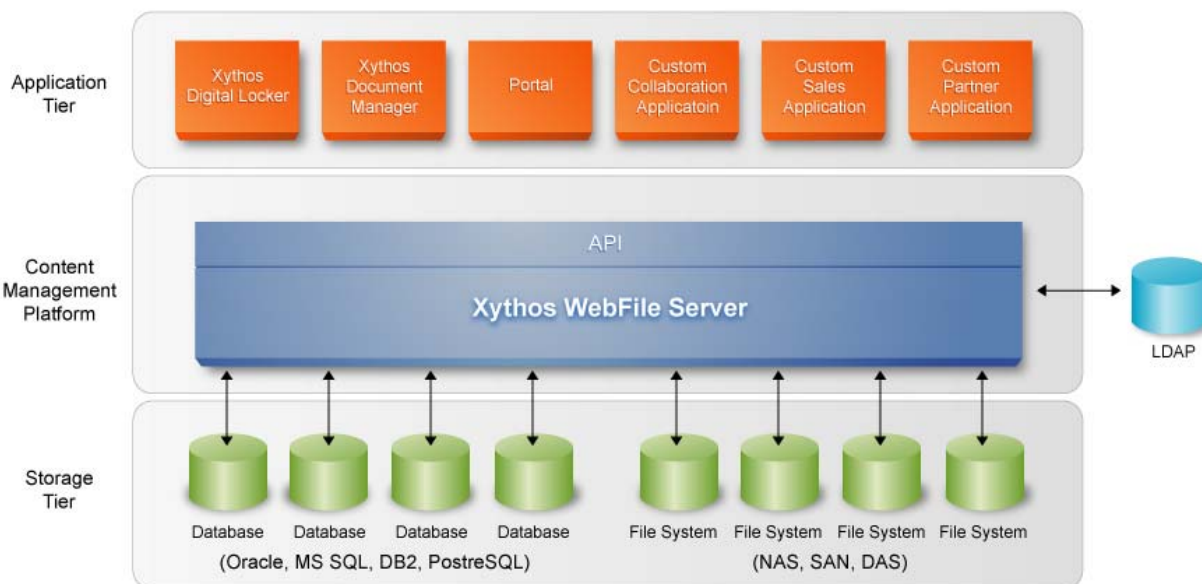
Overview

Xythos WebFile Server (WebFile Server) is the core of the WebFile Development Suite and the foundation for the Enterprise Document Manager Suite and Digital Locker Suite. WebFile Server was designed to support large-scale deployments of enterprise applications for thousands of users from a single instance, and provide a set of useful and immediately deployable services to application developers. Using the WebDAV protocol and documented APIs, developers can integrate a standard set of library services such as version control, file-level and folder-level security, comments, check-in/check-out, flexible metadata and full-file logging to address the collaborative requirements of their applications.

WebFile Server offers users secure access to store, modify and share files from any web-enabled device. Because WebFile Server is a full-featured WebDAV (level 2) server, users can manage their files via any WebDAV-enabled client application. Using WebDAV and other web standards, WebFile Server helps enterprise customers easily find and share information from any Internet location while improving file system security.

The Technology Behind the Server

WebFile Server was designed to easily scale to meet the most demanding performance requirements. Scalability is achieved through a loosely coupled, distributed system in which each component can scale independently. It can operate with all of its components on one machine or each component can scale independently on separate machines. Installation with large numbers of users or amounts of data can be accommodated by increasing the number of each respective storage, database and application server components. WebFile Server performance has been optimized for HTTP and WebDAV so that it can efficiently support file access and storage to both local and remote locations.



WebFile Server was designed to integrate into a company's existing network infrastructure, business processes and application through its standards-based architecture. WebFile Server uses the following protocols and technology:

- **WebDAV** - Web-based Distributed Authoring and Versioning (WebDAV) is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers. WebDAV extends search, share and access traits of HTTP to include editing and versioning abilities to all enterprise data and files. All devices with an Internet browser can utilize WebDAV on supported platforms.
- **HTTP** - Hypertext Transfer Protocol (HTTP) is the primary method used to transfer or convey information on the Internet. HTTP is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems. All devices with an Internet browser can manage files with HTTP. WebFile Server uses HTTP in conjunction with WebDAV to safely connect all users to a centralized file system.
- **Xythos API** – The Xythos application programming interface (Xythos API) is the interface that the Xythos WebFile Server provides in order to allow requests for service to be made to it by other programs in order to allow data to be exchanged between them. WebFile Server is flexible enough to enable software developers and IT professionals to use the powerful Xythos API to align with their specific needs, such as integrating into existing systems that use proprietary APIs. Developers and OEMs can customize the core WebFile Server product to bring power and intelligence to their own services and offerings by programming to the Xythos API.
- **Java™** – Java™ is an object-oriented programming language designed to be platform independent. Java is a derivative of C++ but was designed with a simpler syntax, a more robust runtime environment and simplified memory management. WebFile Server is written in the Java™ coding language which allows for the implementation of the server on the platform that best suits a company's specific IT infrastructure. Whether there is a preference to use Intel-based hardware running Windows, Linux, or Solaris or highly performing server class hardware from Sun, HP, Compaq, or IBM, WebFile Server accommodates an enterprise company's need.

Required Components

Web Server

The web server component of WebFile Server fields incoming HTTP requests. WebFile Server is delivered with a bundled Tomcat servlet container that also functions as the web server. However, it is also possible to install WebFile Server with a separate web server such as the Apache Web Server or Microsoft's Internet Information Server.

The web server functionality may be split among several machines by making use of an HTTP load balancer, by using round-robin IP addressing, or by using other methods native to the web server software or operating system. The web server software running on these machines needs to be able to communicate requests to the servlet container or application server.

Application Server

The Application Server (or Java Servlet Container) component of WebFile Server runs the servlet engine and APIs as well as hosts the WebFile Server servlets and JSPs. The Application Server component allows communication through the WebDAV extensions to the HTTP specification as well as all other headers and methods supported under the HTTP 1.1 and WebDAV protocols.

As previously stated, WebFile Server is delivered with a bundled Tomcat servlet container. However, it is also possible to install WebFile Server with a separate application server or an alternate servlet container. WebFile Server is J2EE-compatible and therefore can be deployed using J2EE-1.3 compatible application servers and some J2EE-1.2 compatible application servers. WebFile Server provides the functionality for creating an Enterprise Archive (EAR) file which is a Java Archive (JAR) file that contains a J2EE application. WebFile Server also provides the functionality for creating several Web Archive (WAR) files which are a package of web modules that collectively perform as a J2EE application. WebFile Server can be installed with a J2EE-compatible application server by using either the EAR file or the WAR files. Additionally, any Java Servlet 2.3 compatible servlet container may be installed with WebFile Server to fulfill its application server/servlet container requirements.

Since the WebFile Server does not store state information in the application server, the system can be scaled by adding multiple, independent application servers that run together or by clustering application servers within the same WebFile Server installation.

Database

To store persistent configuration information and to store all file metadata, such as permissions, versions, properties, comments and logging information, Xythos WebFile Server uses a relational database. Supported databases include DB2, Oracle, PostgreSQL (open source), and MS SQL Server. The WebFile Server can also be scaled across multiple, independent databases and can also be run in any high availability configuration. Therefore, there is no limitation on the volume of files and associated metadata that can be managed by the WebFile Server system.

File System (optional)

Optionally, WebFile Server can store all file content directly onto a storage device, rather than in the chosen database. In other words, a file server may host WebFile Server files as ordinary files on a file system. Some commonly adopted WebFile Server storage solutions include:

- A NAS solution where the storage is simply exposed as another network device, which can be "mounted" on multiple application servers.
- A SAN solution where the SAN holds all the files and exposes the SAN storage to all of the application servers as a file system.
- RAIDed disk array where NFS is used to connect to multiple application servers.

The WebFile Server was designed to scale its storage capacity by simply adding storage devices. Therefore, there is no limitation on the amount of space and the number of storage devices that WebFile Server can manage. WebFile Server can easily scale to store any file size, file type and any number of files.

The storage server component of a document store can be any type of file system that presents itself as a directory to each of the WebFile Server application servers. Xythos recommends using a NAS, SAN, or RAID type technology for use as its storage server. The storage server can be defined as a file system using the External Storage feature of WebFile Server.

Load Balancer (optional)

If there is more than one Application Server in the system, a load balancer should be used in front of WebFile Server. To improve performance, the WebFile Server makes extensive use of caches. Therefore, it is recommended that the load balancer be configured to use “sticky sessions” in order for requests to be sent to the same application server for the duration of a user’s session. Note, however, that this is not a requirement because each application server is stateless and sessions are maintained in the database. If the user is bounced from one application server to another, the user will still remain on the system.

LDAP Server (optional)

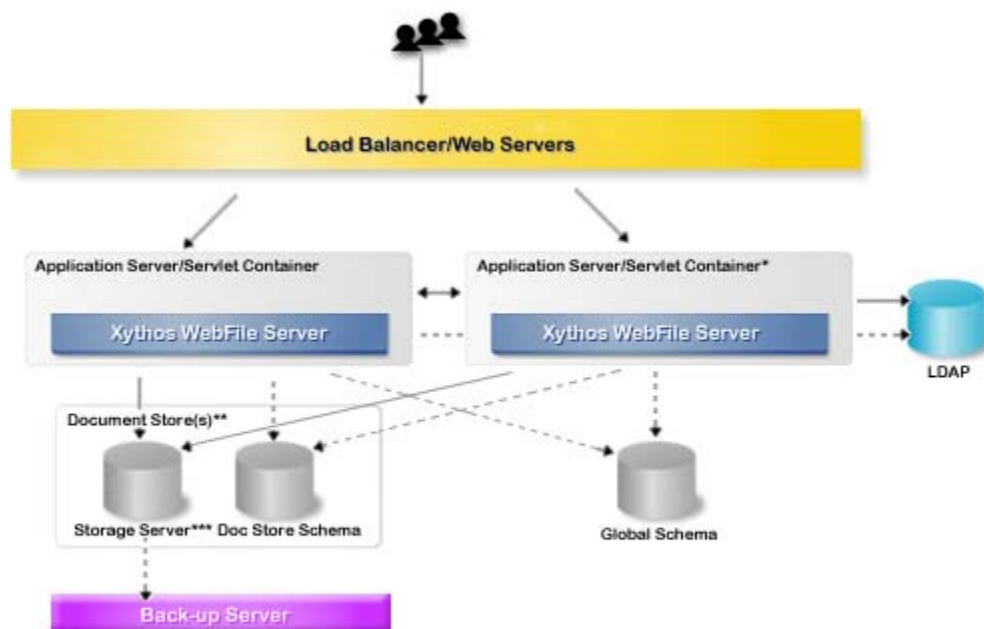
Although WebFile Server does provide its own user model, a directory server may be used in order to provide a single place across multiple systems, including WebFile Server, for the management of user accounts, group membership, and logon credentials. Centralized account management and a single secure user credential eliminates the problem of requiring users to remember multiple passwords, as well as requiring IT managers to handle multiple system accounts and passwords for one set of users. It also provides an easy and secure way to manage user and group access rights. Xythos WebFile Server helps address this goal by integrating with any directory server that conforms to the LDAP specification v2 or v3.

Hardware Requirements

The Xythos WebFile Server can be run on virtually any hardware platform. Because of its standards-based design and browser-based interfaces, the only factor influencing hardware requirements is the software selected to run each of the required WebFile Server components described above. This flexible design supports all mainstream operating systems including Solaris, AIX, Linux, Windows, and Mac OS X.

Although virtually any hardware platform may be chosen, hardware requirements should be roughly based on the number of concurrent users. A concurrent user is defined as a user who is uploading/downloading a file at a particular moment. Once that file has been uploaded/downloaded and the user is reading and/or updating the file, the load is no longer incurred on the server, as the user is not maintaining any connection to the server. Therefore, ideally, in a high availability environment, it is preferred to have two servers running in a load-balanced scheme. For several thousand users, running each server with one CPU and approximately one half gigabyte of memory should suffice. However, for more than several thousand users, another CPU and additional memory may be required.

Xythos WebFile Server Storage



*The Application Server/Servlet Containers can be scaled up to as many as needed.
 **This solution stores metadata in the database and the file contents are stored on the storage server. The Document Stores can be scaled up to as many as needed.
 ***Each Application Server must have a mount/share to the storage server(s). The Storage server can be scaled up to as many servers as needed.

Document Stores and File Storage

WebFile Server files are stored in document stores. A document store is a database construct used to store file metadata and, optionally, file bodies. Each document store is, therefore, composed of two components: a database to store file metadata (file name, size, ownership, permissions, etc.), and a file server(s) to store the file content. Every database machine in a WebFile Server topology that is intended for document storage must have at least one document store associated with it which must be created and configured during installation of each WebFile Server.

Document stores are a means for partitioning database storage in order to make that storage more manageable. Therefore, if a particular document store is becoming too large, it is possible to create a new document store in order to make the database table's size more reasonable and ultimately, more manageable. It is even possible to associate this document store with another database instance in order to transfer some processing load to another server. However, it is important to note that the document store does not segregate the content and control access. This is governed by the WebFile Server virtual server construct, as well as file and directory permissions. WebFile Server was designed this way so that the entire set of documents stored in any document store can be accessed by any user (given the correct permission set).

External Storage Directory Structure

If the database is not being used for the document store storage server, external storage must be defined within the system. Specifically, external storage locations are defined with the administrative pages of WebFile Server. Each external storage location defines a root of a directory structure containing WebFile Server file content. WebFile Server maintains its own proprietary directory structure on each of the external storage locations. Any new file uploaded to the system is placed in a directory corresponding to the date. Additionally, the WebFile Server never overwrites any file in external storage. If a file is edited, the system will not overwrite the file on external storage. Instead, it will delete the old file and create a new one in a directory corresponding to the edit date.

For example, a WebFile Server file called “/users/jdoe/test.doc” is physically stored at “/xythosdata/2005/2005-01/2005-01-12/4901/4201.204”. In this case, /xythosdata is a mounted directory, on each application server running the WebFile Server, that points to the storage server. In the WebFile Server, /xythosdata is called an External Storage Location.

Single File Instancing

Even if there is more than one copy of a file within the system, WebFile Server will not store more than one copy of that file on external storage. This is known as single file instancing. The document store architecture allows for single file instancing because multiple files in the database can point to the same file on disk. Single file instancing dramatically reduces storage requirements by eliminating duplicate data which is often estimated to be 30% of an organization’s overall stored content.

Global Schema

WebFile Server stores system-wide parameters in a database called the Xythos Global Schema. User and session information is also stored here. There is only one Xythos global schema per WebFile Server and it must be accessible to all machines running the WebFile Server code.

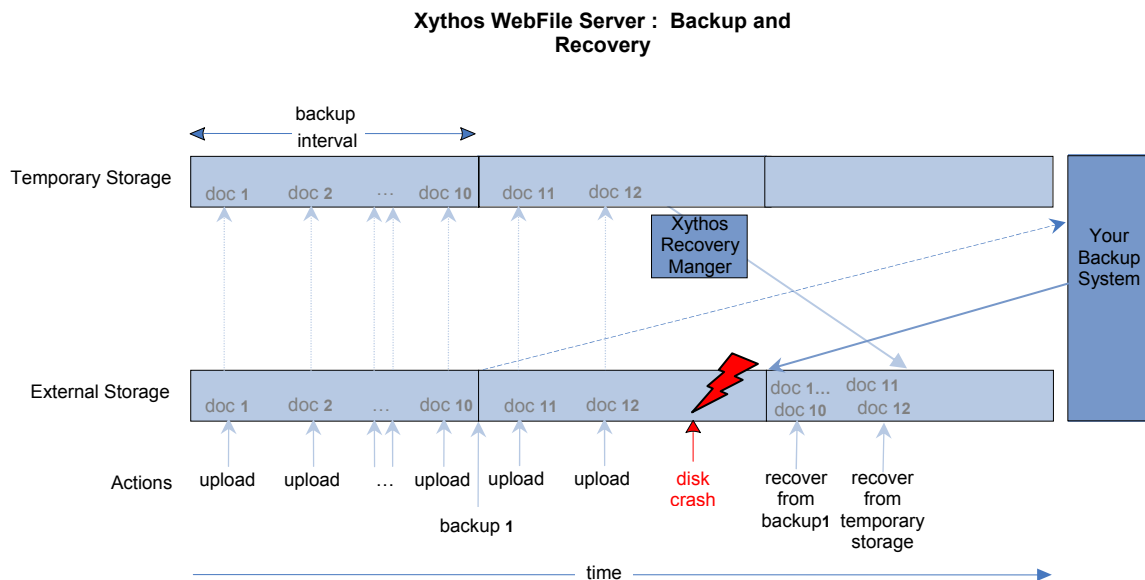
Backup and Recovery

Document backup and recovery depends on the storage server strategy in use, specifically whether documents are stored in the database or external storage. Documents stored in the database can be recovered by the standard database recovery procedures. Documents stored in external storage are protected by the standard file system backup procedures. To complement the file system backup, WebFile Server can be configured to provide backup and recovery of documents added or changed after a regularly scheduled file system backup.

There are two storage locations available per document store when configuring the system’s back-up strategy: primary storage and temporary storage. Using the Document Store database, External Storage, and both storage locations, WebFile Server files may be stored in a variety of file storage configurations.

When temporary storage is configured, documents added to the primary disk are also added to the temporary disk. If the primary and temporary storage are physically different disks and a crash occurs, the WebFile Server Recovery Manager uses the temporary storage to restore documents that had been added to the system since the backup, to the primary storage

location. Temporary storage is cleaned on a time schedule corresponding to the file system backup schedule.



Because the system was intended to be configured in this manner, Xythos recommends configuring primary storage and temporary storage on different disks as external storage. In addition to the document recovery described earlier, reads and writes to the file system are much more efficient than reads and writes to a database. Additionally, file system tools, such as search or virus scanning, can be used with documents stored in a file system.

There are also features within WebFile Server for recovery of a single document. Each directory has a designated spot where documents are moved before they get deleted completely from the system. This designated spot is known as the directory's trash. Additionally, the system can track deletions from the system. In this manner document names and storage locations in the backups are marked when a document is deleted. The Xythos Delete History feature can then be used to retrieve the deleted document based on the location of the document within the backup.

Xythos WebFile Server Sizing and Scaling

Sizing

The table below provides recommendations for system sizing based on the number of users.

Total Number of Licensed Users	*Xythos Deployment Architecture	
	Xythos WebFile Server /App Server	Database
< 1,000	1 machine: 1-2 Processors 500MHz+ 1 GB RAM	1 machine: 1-2 Processors 500MHz+ 1 GB RAM
1,000 – 3,000	2 machines, each: 1-2 Processors 900MHz+ 1 GB RAM	1 machine: 2 Processors 900MHz+ 2 GB RAM
5,000 – 10,000	2-3 machines, each: 2 Processors 900MHz+ 1 GB RAM	1-2 machines: 2 processors+ 900MHz+ 2 GB RAM
10,000 – 20,000	3-4 machines, each: 2 Processors 900MHz+ 2 GB RAM+	1-2 machines: 4 total Processors (1 X 4 or 2 X 2) 900MHz+ 2 GB RAM
20,000 – 40,000	4+ machines, each: 2 Processors 900MHz+ 1 GB RAM (more ram if less boxes)	1-2 machines: 4 total Processors (1 X 4 or 2 X 2) 900MHz+ 2 GB RAM+
40,000 +	6+ machines, each: 2 Processors 900MHz+ 1 GB RAM (more RAM if less boxes)	1-2 machines: 4+ total Processors (1 X 4 or 2 X 2) 2) Proc, 900MHz+, 4 GB RAM+

*Each of the above cases assumes the solution includes a storage server. The storage server machine(s) is dedicated to storing file content and it must have enough space to store all of the data. The storage server machine must also present itself to each application server as a mounted drive or network share.

Database Sizing

The number of document stores depends on the anticipated growth of the database on which the document store resides. This growth depends on the amount of files and corresponding metadata per file. In order to estimate the database size, the following assumptions are made:

- Each file will average three file versions
- Each file will average three custom properties
- Each file will average ten log entries

Given these assumptions and some base averages for file version, custom property, and log entry sizing, each file will require approximately 4588 bytes, in order to store metadata information. This number would need to be adjusted accordingly if the estimates in the assumptions are increased or decreased.

It is always possible to add document stores "on the fly" within WebFile Server. Adding a document store in this manner would require setup of the corresponding database schema by running the WebFile Server installation utility to setup the database objects.

Scaling

WebFile Server was designed to easily scale to meet performance requirements. As stated previously, scalability is achieved through a loosely coupled distributed system in which each component can scale independently.



Application Tier Scaling

A single WebFile Server instance can support numerous application servers at the application tier. If a load-balancing switch or router is used, the solution can have load-balancing as well as fail-over capabilities.

For systems involving frequent access and/or upload of files by a large number of users, clearly bandwidth is a concern. If the WebFile Server site is serviced by a sufficiently wide pipe, administrators may want to scale up the number of web servers and application servers

accessing the data. Effectively, scaling adds machines to the application server tier to handle the load. The number of application server machines is unrestricted.

Storage Tier Scaling

The storage tier consists of one or more document stores. Each document store consists of a database to store file metadata, and one or more external storage locations to store file content. Thus, the storage tier can scale in two dimensions: by increasing the number of document store databases and by increasing the number of storage servers.

Additional document stores can be added to the system to provide load balancing for the database(s). Each top level directory (root directory) in the WebFile Server is associated with a particular document store and all files stored within that directory are stored in the database and file system(s) configured for that document store. This allows administrators to distribute the management of all WebFile Server content across multiple databases and storage devices. For example, data within /users/ can be managed by one database and storage server, while all shared data in /company/ can be managed by different database and storage server. Top level directories can also be moved from one document store to another.

Additional storage servers can be added to any of the document stores in order to provide more storage capacity. Once disk space starts to run low on a particular external storage location, administrators can add another external storage location to the document store. The system will continue to read and write to the old storage locations for existing files, but new content will be written to the new storage location. External storage locations can be moved and renamed as computing resources change.

WebFile Server Clustering and Fail-Over

Application server fail-over can be accomplished by deploying the WebFile Server solution with multiple application servers in a load-balanced configuration.

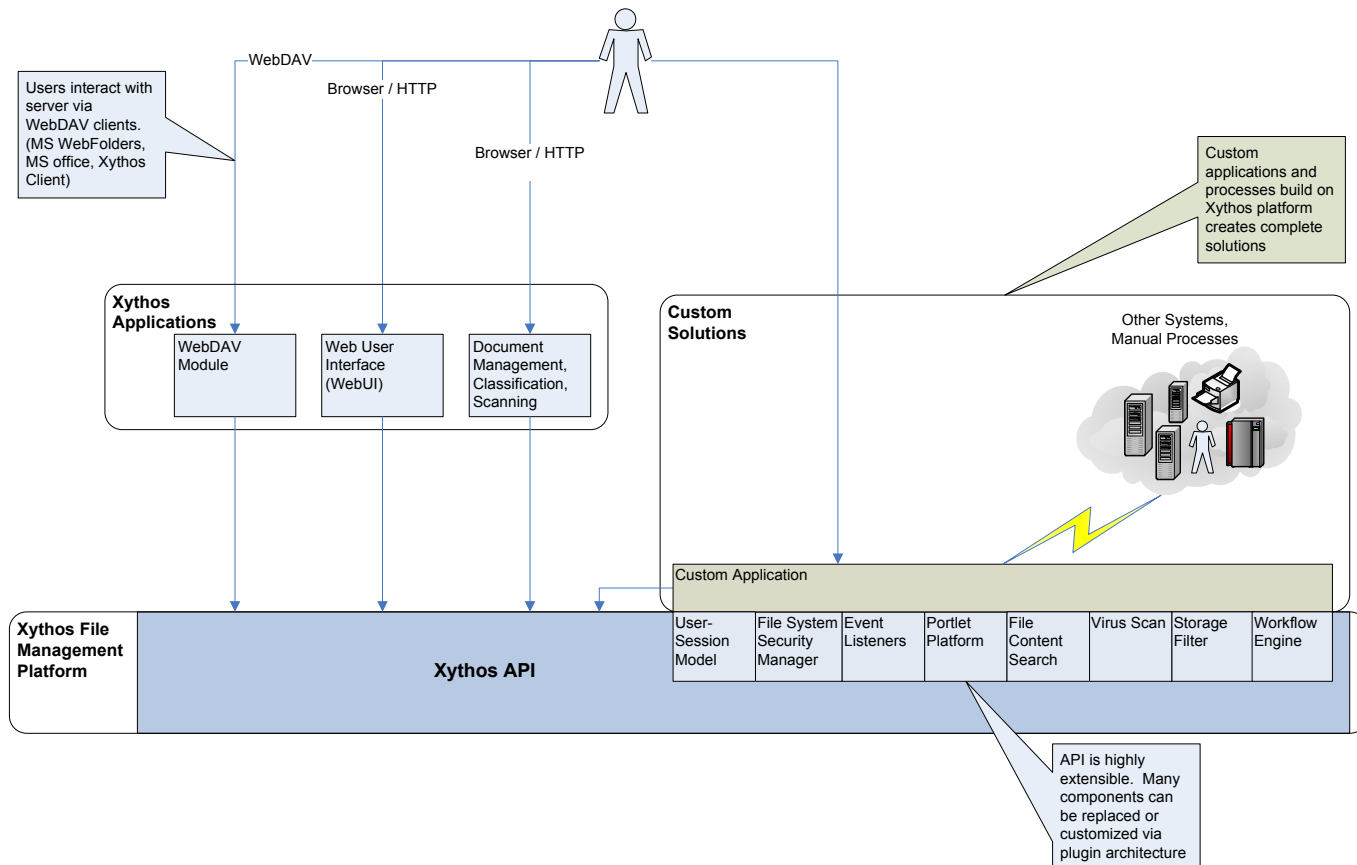
WebFile Server relies on existing database clustering technologies to provide fail-over/clustering capabilities. For example, both Oracle and MSSQL Server provide application clustering capabilities. If using either of these as the WebFile Server database, configuration is done at the database tier level and communication is facilitated through WebFile Server to the clustered environment.

For the storage server(s), fail-over would be provided by the storage technology deployed with the solution. This may be accomplished by deploying SAN or NAS solutions (e.g. a Network Appliance box), or by simply using RAID technology to achieve disk mirroring.

Logical Architecture

The Xythos WebFile Server has both “out of the box” applications as well as a file management platform that can be extended and built upon to create new applications.

Xythos WFS: Logical Architecture



Xythos Applications

Xythos has built two applications on top of the WebFile Server file management platform.

Digital Locker

The Digital Locker application allows users to manage and share their files and directories through the user's preferred web browser. The Digital Locker provides an easy-to-use interface, exposing all WebDAV functionality as well as additional file management features. This includes bookmarks, search, contacts and groups, locks, logging, access control, versioning, tickets, comments, and custom properties.

Enterprise Document Manager

The Enterprise Document Manager has all of the features and functionality of the Digital Locker with the addition of a Classification module and a Workflow module. Classification gives users the tools to effectively manage the metadata associated with different document types. Workflow allows for simple routing, reviews and approvals of individual documents.

Both the Digital Locker and the Enterprise Document Manager are fully customizable.

Custom Applications

Java API

The WebFile Server contains an extensive, open, documented API written in Java, which exposes the full functionality of the server. This API can be used to build entirely new applications, or to customize the behavior of the WebFile Server. The File Management Platform has a plug-in architecture where numerous components within the server can be customized or replaced completely

- **User-Session Model**
The user-session model plug-in is used by WebFile Server to find users and groups, authenticate users, and maintain sessions. This is often customized to integrate the WebFile Server with other authentication systems besides LDAP (Kerberos, CAS), or to implement single sign-on (Netegrity Siteminder, HP Select Access).
- **File System Security Manager**
The File System Security Manager plug-in is used by WebFile Server to determine what actions are allowed on the file system. WebFile Server consults the security manager to determine if a file/directory is readable, writable, and so forth.
- **Event Listeners**
Every action which takes place in WebFile Server (file reads, writes, deletes, copies, etc) fires an event which can be processed to perform custom logic.
- **Portlet Platform**
The WebFile Server has a JSR 168 Portlet Platform which can be extended to build custom portlets. The platform integrates with the WebFile Server security model so developers can concentrate on building user interface components. Additionally, WebFile Server contains three portlets: Directory View, Bookmarks, and File Search. The source code for these portlets, as well as the underlying portlet platform is provided with WebFile Server.
- **File Content Search**
The File Content Search plug-in is used by WebFile Server to find files based on content searches. This can be customized to integrate with the search filters of your choice, such as Verity. Xythos has integrated with the Lucene search engine and uses the Verity search filters.
- **Virus Scan**
WebFile Server has the capability to scan every file which is written to the system with the virus scanner of your choice. A simple class must be written to interpret the results of your virus scanner.
- **Storage Filter**
When WebFile Server writes files to, and reads files from the storage server the bytes are passed through a filter. In high security environments, this plug-in can be used to encrypt files before they are written to disk, and to decrypt files when they are read from disk.

- **Workflow Engine**

The Workflow Engine can be used to create additional functionality above and beyond the simple document approval and routing processes available as part of the Enterprise Document Management solution.

Web Services API (WebDAV)

The WebDAV module works with any WebDAV client and allows users to manage their files with powerful GUIs such as Microsoft's Web Folders and Xythos Drive.

Conclusion

In today's computing environment, unstructured data is pervasive. Most business processes require the sharing and management of unstructured information that is often accessed by users residing in different locations, using different browsers on different operating systems. As a result, an increasing number of enterprise applications must interact with a variety of unstructured information and support collaborative document and file management capabilities.

Along with the Enterprise Document Manager and Digital Locker applications, the Xythos WebFile Server offers all the tools a distributed organization needs to safely capture, manage, store and share unstructured data over the Internet. Because it was designed to work together with the applications and processes that users and developers already know and use, Xythos WebFile Server accelerates user adoption, reduces support requirements and provides the open-standards platform necessary for the further development and integration of file management functionality into existing enterprise applications.