OPEN-XCHANGE™ Whitepaper

keep in touch

Open-Xchange™ Server Integration with eDirectory

eDirectory as an Authentication Source for Open-Xchange Server 5

v1.00

With contributions from:

afOX

IDEALX

CONET

Please direct any recommendations or comments to
documentation@open-xchange.com

Author: Stephan Martin

Contributors: Marcus Klein,
Nicolas Barcet (AFOX),
Farzad Farid (IDEALX),
Dirk Wolter (Conet), Ralf Dobberschütz (Conet)

Editors: Robert Colombara, David Cuthbert, Stephan Martin

Layout: Robert Colombara

Contents

1. Overview

The Open-Xchange Server 5 is an Open Source-based, full featured email and collaboration solution that runs on Linux.

The architecture of Open-Xchange Server 5 is completely based on open standards and open protocols to allow maximum flexibility with regard to integration into existing infrastructures.

This whitepaper describes how an Open-Xchange Server (OX Server) can be set up to use Novell's eDirectory as an authentication source for users and groups as well as providing storage for address books.

This document is meant to be used as a guide, a whitepaper that describes the concepts, and not as a complete How-To. Nevertheless, some configuration files e.g. schema files will be made available in the Open Source Community Wiki and are linked to this document.

Attention:

Deep knowledge about Linux services and administration as well as about directory services in general and especially eDirectory will be necessary to run an integration project like the one described below.
You should definitely know what you are doing and not just follow the documentation blindly. If you don't have confidence that you understand everything written below, it may be a good idea to contact an experienced Open-Xchange partner in your area.

1.1. Contributions

This whitepaper was written by Open-Xchange in cooperation with our partners CoNet in Germany and IDEALX in France and with support from the "Association Francophone pour le developpement d'Open-Xchange" (AFOX).

We would like to thank everybody for their contributions and support.

2. Architecture Overview

This chapter gives a brief overview of the architecture of the Open-Xchange Server and its backend components, as well as some of the different ways to integrate Open-Xchange into existing eDirectory infrastructures.

2.1. Open-Xchange Backends

The Open-Xchange Server does not store any information itself. All the information the server needs is stored by backend services. These services are tailored to fit the kinds of data which need to be stored.

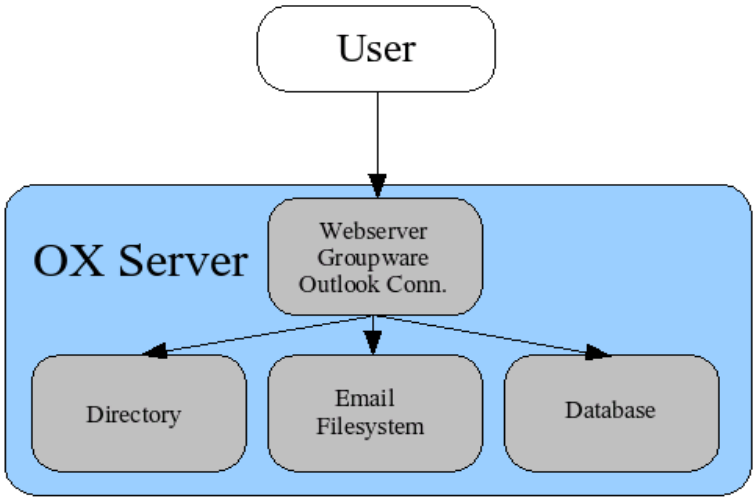

Figure 1: Storage Subsystems inside of the OX Server

2.1.1. Directory Service

Storage of the OX user information needed for authentication is handled by a directory service.

A second usage of the directory service is to store global and private address books. The global address book, which is accessible by everybody, and private address books, which are accessible only by their respective owner, are exported from local client machines to the LDAP server to allow access from standard clients like Mozilla Mail or Outlook Express.

The standard installation of the Open-Xchange Server makes use of the free and stable OpenLDAP server which is included in the underlying Linux distributions.

This paper will focus in detail on the OpenLDAP backend.

2.1.2. Email

Email is stored by an external email service. This service is accessed via the standard protocols IMAP and SMTP. The email components used in the standard setup are the `postfix` Open Source MTA and the Open Source IMAP server, `cyrus-imapd`.

Both services are configured to obtain authentication and mail routing information from the directory service.

Normally, during user creation the administrator creates the mailboxes for the users. This process can be delegated to the web mail front-end, which can be configured to create the standard Open-Xchange folders when a user first logs in.

2.1.3. Database

All groupware data, appointments, tasks, etc. are stored in a relational database. The standard installation of Open-Xchange Server makes use of the free and stable database program, PostgreSQL, which is included in the underlying Linux distributions.

The folder permissions for groupware access are stored in the database. This is the reason why user administration requires changes to the database..

2.2. The Open-Xchange Administration Framework

This chapter describes the standard setup used to administer an Open-Xchange Server integrated with its backend services.

In the standard setup, one administrative front-end is used to administer all users through the Open-Xchange Server. This is done through the XML-RPC interface which allows access to all administrative tasks. On the Open-Xchange Server there is a daemon running which will listen for XML-RPC calls and which coordinates necessary actions with the relevant back-end services to respond to them.

In short this means that while data is stored by backend services, calls to those services are handled by the Open-Xchange server itself rather than directly through the administrative front-end.

The main advantage of this concept is that the Open-Xchange Server acts as an abstraction layer between the administrative front-end and the back-end services. If one backend service is changed, for example, the database component only the relevant interface in the Open-Xchange Server needs to be adapted and the other components don't need to be changed.

3. eDirectory Integration - Concepts

This chapter describes two possible methods of integrating the Open-Xchange Server with Novell's eDirectory.

Common to both methods is the requirement that after integration the Open-Xchange Server will no longer be used as the administrative front-end as was described above.

If a company already uses eDirectory for their identity management it is very likely that they already have a special purpose front-end in place which is not designed to be exchanged with the administrative front-end from a groupware solution.

In effect, this means that the existing directory service front-end needs to be enhanced to:

- write the necessary Open-Xchange information into the directory
- trigger the necessary actions for other back-ends in the Open-Xchange Server

3.1. Simple Integration

The simple method of directory integration works without code changes to the Open-Xchange Server and does not require the coding of special connectors inside eDirectory.

On the other hand, it is necessary to enhance the eDirectory schema to implement this approach.

With this concept there is one administrative front-end which writes changes to user information directly into the directory server. It does not matter which front-end is used. It can be a custom front-end which is already in place, it can be an extension to iManager, or it can be a simple LDAP browser with a template as well.

The Open-Xchange Server will access this information from within the application and read the information. Write access from the Open-Xchange Server to the directory is only necessary for password changes and for changes to exported LDAP address books.

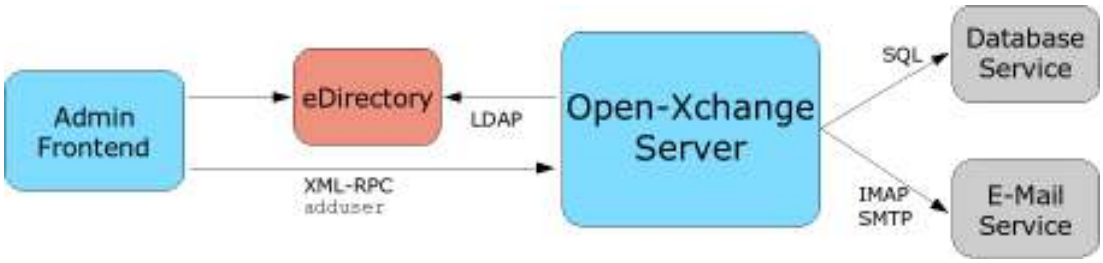

Figure 2: Schema Admin Framework eDirectory – Simple Integration

Using this setup means that changes to the directory and to the back-ends are separated. Not everything will be written through the Open-Xchange Server.

Therefore the existing administrative front-end must be configured to transmit any changes to the Open-Xchange Server, which takes care of propagating the changes to the other back-ends.

This can be achieved in several ways:

- The front-end can send an XML-RPC call to the Open-Xchange Server directly
- The front-end can call the command line tool `adduser` which generates the appropriate XML-RPC call to the Open-Xchange Server
- Another possibility is to run a regular `cron` job to analyze the contents of the directory and to initiate the necessary actions if new users are found.
- Or the IMAP server, `cyrus-imapd`, can be configured in such a way that new mailboxes do not need to be created with the admin interface, but will be created when a user either logs in the first time, or when the user account first receives an email (cyrus options: `autocreatequota`, `createonpost`, `autocreateinboxfolders`, `autosubscribeinboxfolders`)

There is plenty of room and flexibility to design the connection in a way that fits well in the existing environment.

3.2. Advanced Integration

A second, more advanced way to integrate an Open-Xchange Server into an eDirectory environment requires some more effort.

Basically it is the same idea as described above. The main difference is that there is no administrative split for the two channels. The only administrative entry point is by writing changes using eDirectory.

In eDirectory a connector can be implemented that captures changes in the user objects and then triggers the necessary actions on the Open-Xchange Server with XML-RPC calls.

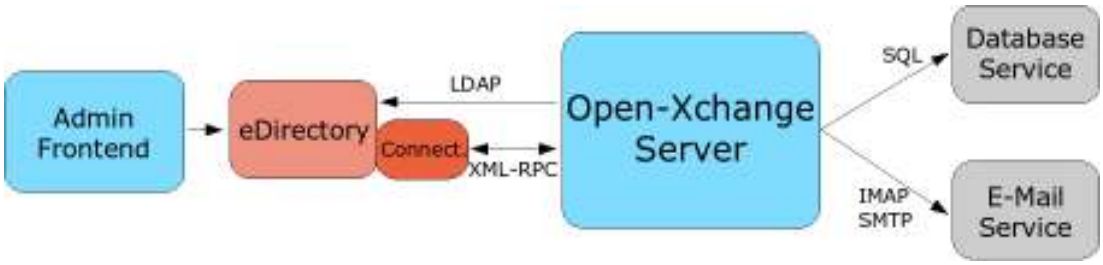

Figure 3: Schema Admin Framework eDirectory – Advanced Integration

4. eDirectory Integration – Adaptations

This chapter describes the necessary steps to achieve the integration described above.

As each implementation of an eDirectory can look different, this is not complete How-To, instead it is a high level description of the necessary tasks.

Example configuration files and schema files can be found in the Open Source community Wiki.

1. Enhance the eDirectory schema:
   Upload the schema files from the Open-Xchange Wiki in the correct order through ICE or a corresponding web front-end in iManager. The given schema files will only be sufficient, if you are running eDirectory on Linux (OES) or if you have added the UNIX schema files for Linux User management (LUM) manually. This task of enhancing the eDirectory schema should only be done by experienced eDirectory administrators.
2. Configure the email back-end services and the Open-Xchange Server to access the new eDirectory LDAP server. This is done in the file `ldap.conf`
3. The LDAP interface to the groupware needs to be adapted to place the private address books in a special subtree. They cannot be placed below the corresponding user object due to a restriction in eDirectory's implementation of the underlying X.500 schema which does not allow a user object to be used as a container.
   Additionally, disabling the export of address entries to LDAP through the Open-Xchange LDAP interface is recommended if no external LDAP address client is in use.
4. The permissions of the eDirectory need to be update:
   - For "anonymous" binds there must be a system user that is able to search all users and groups in the directory tree.
     This user is configured in the file `ldap.conf`
   - Every user needs to be able to read the relevant entries from all other users for different reasons:
     - To select the other users as participants for appointments, tasks, etc...
     - To read the other user objects as address book entries, e.g. for sending email
   - Every user needs read access to all group objects to select users from these groups
   - If a private address book export to LDAP will be used, every user needs read and write access to every object below his private address book container.

5. The mapping of several attributes - the search bases and the search scope for several queries, needs to be updated in the Open-Xchange LDAP interface.

   For example login names are represented in the attribute "cn" in eDirectory in opposite to the attribute "uid" in the standard OpenLDAP deployment.

   These configuration options are set in the file `ldap.properties`. An example of this file is available in the Open-Xchange Wiki as well.

6. The private address book tree needs to be placed in a separate subtree. This is because the X.500 schema of eDirectory does not permit use of the users object as a container.
   Another possibility is to switch export of private address books to LDAP. This saves making a lot of useless entries in the directory server if this feature is not used by the clients.
   Both options can be set in the file `ldap.properties`

5. Administering Users and Groups

User administration through eDirectory is straight forward. In the simple integration, mentioned above, there are three steps involved:

1. User Creation
   The user will be created within eDirectory in the same way as any other eDirectory user was created before the Open-Xchange integration. This can be done with Console One, iManager or any other front-end or integrated software which is currently in use to administer the users in the existing eDirectory.
2. User Object Enhancement
   The users object needs to be enhanced with the Open-Xchange objectclass and some mandatory attributes.
   If any of the attributes needed by Open-Xchange are already available in the directory for some other application, it is possible to change the attribute mapping for Open-Xchange in the file `ldap.properties` so that it is not necessary to store the same information in multiple locations.
   It is recommended that at least the following attributes be added:

   - Objectclass:

   `OXUserObject`

   - Attributes:

   `OXTimeZone (e.g. Europe/Berlin)`

   `smtpServer (e.g. 127.0.0.1)`

   `imapServer (e.g. 127.0.0.1)`

   `mailenabled (e.g. OK)`

   `preferredLanguage (e.g. DE)`

   `userCountry (e.g. DE)`

   `maildomain (e.g. test.tux)`

   `OXTaskDays (e.g. 5)`

   `OXAppointmentDays (e.g. 5)`
3. Groups
   Groups are handled just like normal groups within eDirectory. There is nothing special to take care of. Adding and removing users from groups works as expected.
   (One pitfall: there are many programs under Linux that will have trouble if a user is member of more than 64 groups.)
4. Backend Deployments

- Database
  When the user is created in the directory service, some associations have to be created in the database as well to create standard folders, grant permissions, etc...
  This can be done with the command line tool:
  `adduser –-sql ...`
  `adduser --source_only=groupware . . .`
  Another alternative is to communicate directly with the XML-RPC administration interface and trigger the required actions. The documentation of the XML-RPC interface can be found in the `doc` directory of the admin framework installation.
- IMAP Mailbox
  The user needs a mailbox with several folders where email is physically stored. This mailbox needs to be created before it can be used. There are two ways to achieve this:
  - Creating the mailbox during user creation:
    For this way, the same alternatives exist as described above for the database:
    The command line tool:
    `adduser --source_only=imap ...`
    Communicate directly with the XML-RPC administration interface and trigger the required actions. The documentation of the XML-RPC interface can be found in the doc directory of the admin framework installation.
  - Let the mailbox and folders be created automatically during the user's first usage/login:
    In `imapd.conf` the `autocreatequota` parameter cannot be 0 for allow cyrus to create the mailbox during the user's first login.
    In `webmail.properties` the parameter:
    `user.default.folder.autocreate`
    has to be set to true, to create the system folders through web mail during the user's first login.