This document is a description of Sympa project direction. The goal is to fix priorities and to publish it in order to collect users comments. The initial version (November 2005) describe development project from Sympa version 5.1. We will try to keep it up to date but no garanties.

Of course, Sympa forge server include a bug report and feature request system that are used too.

We kindly suggest you to comment this document using the wiki feature.

Performances

Performances become a limitation because many sites now create thousands of lists. Many universities create lists automatically (through families) using data coming from the information system. This performances issue concerns mainly the **web interface**.

web interface with many lists

We all ready have listed some problems :

- startup delay
- response time
- memory usage

In version 5.2 startup is all ready optimazed using binary version of list config file (so initial loading of configs is much faster) and the number of *which* calls is limited when operation is related to a list.

We still need to fasten the response time for all other request. We will explore thge following directions :

- sharing memory storage of each scenario (currently duplicated for each list)
- optimize and limits number of SQL request
- do_log subroutine calls might be expansive
- List::new calls might expansive (exemple : testing the date of config files)

List families

List family instanciation is currently only available via the command line, as a bulk process. You provide a set of data and a family definition and sympa.pl instanciates the family to create all the lists.

We have planned to extend the family features and allow list creation from the web interface and also from the mail interface (to create one-time lists).

task manager

- spool managment needs to be optimized

perl memory leaks

- not sure, comments welcome !

Scalability

First, it could be very usefull to separate differents daemons on separate servers. This can be done already but there are some limitation because everything is not in Sympa database (configs, archives and spools).

Second : a daemon could need load balancing

lock files

In order to ease repartition of Sympa service on multiple server, lock files should be done using a mysql table.

Using memcached

mem cached is a free software tool that allows storing data in RAM and share access to those data using a inet or unix socket. may be this tool can be used in order to solve sharing of data between different daemons.

moving config files and spools to the database

This would make the application more complex and difficult to manage using your favorite editor file !

Authentication architecture

Some improvements of security and usuability are expected.

password storage (done in version 5.5)

Currently password are stored in the database using reversible encryption. This make *password reminder* possible but of course there is a security issue. We plan to replace password encryption with hashing (MD5 or SHA1) in version 5.5. So reminder feature will be replaced by *reset password* in a similar way as other web application.

1. wwsympa create a *one time ticket*, store it in a dedicated table (one_time_ticket_table) and send it to the user
2. if the user goes to the webinterface, this ticket is *burned*, the user is authenticated and user is prompted to change/choose password (this way a user never had to use a password he didn't choose).

The method using a “one time ticket” could be used else where in Sympa (in many cases sympa send some URL to users, it is usefull that thoses URL are considered as a way for authentication).

This dev is started and will change all authentication forms.

auth.conf

- ~~auth.conf is currently a global file. We will make it possible to define different auth.conf for each virtual robot.~~(done)
- Some want to use https only for encryption of login form. This need to be introduced in auth.conf

export password in LDAP

If Sympa could export password in a LDAP directory, it would be possible to use this LDAP directory for some other application (single sign on such as CAS).

SSO without email patch

Done : ~~We received a big contribution from John-Paul Robinson from University of Alabama at Birmingham. The goal of this patch is to change the Shibboleth-based authentication so that Sympa does not use email addresses as user identifier. We will introduce this contribution in main Sympa version quickly.~~

n-tier architecture

~~This section concerns the SOAP server. It is needed for some remote application acting as soap client of Sympa server to make some authentication assertion about the end user. Exemple : a CMS with its own authentication method can request services to Sympa as proxy of the end user. That can be done if Sympa SOAP server trusts the remote application and accept a set of assertions from this SOAP client. Sympa would use incoming variables in authorization scenario.~~ (done) A Sympa server could also request services from another Sympa server. Imagine that “which” action returns could tell the user about his subscriptions on another mailing list server.

DKIM

DKIM is a emerging standard for message authentication based on signature. Sympa will check DKIM signature to the benefit of authorization scenario. It is also needed to verify that message modification made by Sympa (headers and body) doesn't alter original DKIM signature. In some case Sympa could sign itself messages when authentication of the original sender is good enough.

Alternate email addresses

Sympa allows you to declare an alternative_email_attribute LDAP attribute in your auth.conf file. When authenticating a user via LDAP this information is retrieved. We currently have a VERY limited used for this data :

- Stored in a HTTP cookie
- Used to propose email unification (subscriptions with adresse A are changed to address B)

What should be done in the near future to extend this future

Customizing mail content per subscriber

A lot of work as been done last year in order to allow parsing of message footer and header using all variables of Sympa templates including the subscriber email. This would make possible to add a message footer with personnalized unsubscribtion instruction or to use VERP for beter bounce management. This need to bypass the Sympa internal bulk mailer that save some many bandwith.. Now everything is ready to start this new developement We plans to allow both current mail grouping and mail parsing depending on the need (example use parse for unique return path only for subscribers which get bounces or every 10 messages send to the list etc)

User preference

In addition to session attribute, user preferences and attributes could be introduced :

- subscriber attributes defined by list owner. The managment of subscriber attributes could be performed by the list owner or by subscribers themself. Attributes could be hidden to other subscribers or not. Attributes could be used in scenario ?
- List owner need to tag some boor users in order that any mail from this person is submitted to list editor before distribution.

Shared document repository

Some more feature in the shared repository document :

- get a complete folder as a ziped file

- display a header and a footer of a folder
- encoding file name. There is big bug related to file name containing some 8bits chars.
- move, remove, rename a folder or a file : needed if you have to reorganize a folder.
- display folder content recursively (something that look like ls -lR unix command)
- provide a basic search engine (using filenames, not contents)
- Some wants to restrict access to some part of the shared, somes want to open more access on sub-folder. The solution we have choosen is somithink like Unix access heritage from directory to subdirectory but it is difficult to understand (and to explain). We plans to introduce a “inherited access” attribute to a folder. With his attribut any subfolder of that folder will have exactly the same access control rules.
- replace htmlarea wysiwyg html editor because it is not any more supported.

Internals : a work has been started to rewrite an object oriented version of this code. The work needs to be finilized…

Bounces management

Extending bounces notifications (see Pour une meilleurs remontée des bounces, in French)

Logs and OPT/IN traceability

subscription traceability

Many spammers claim their “service” is based on opt-in but that's false. Sometime users claim they never subscribe to a legitimate mailing list, but in fact they did ! Subscribtion traceability will provide information about the users subscription, on the web interface (initial subscribtion message, confirmation). This is ready (nice work from Floriane Fiquet) and it will integrated in a very near future.

Internals

- add a particular session attribute that allows debug mode
- add a particular attribute that display variable available for TT2 parsing so no more documentation is needed for that.
- ~~replace CVS by subversion to manage sympa sources~~
- check point during distribution processus so it could be possible to stop sympa.pl during a hugue message distribution without duplicate any message
- currently listmaster is listowner of any list. In some case listmaster must have more restriction then listowner.
- archive should include new message as soon as possible preferably just when the distribution processus start. Message should be copied to the outgoing spool before the distribution processus start.

- ~~test PAPI authentication method~~
- ~~utf8 encoding : charset encoding is depending on server file system, .po catalogs and client. We need to fix some bugs but also to consider global organization for encoding usage.~~
- ~~full virtual robot : different virtual robots should be able to manage different lists using the same name. The work is almost finished in main the CVS branch.~~

Documentation

The reference manual needs to be completed regarding some topics. Other documentation documents would be of interest (quick start, developpers doc, list owner documentation, online documentation).

List members management

This chapter should explain the difference between subscription and inclusion. It should also explain how include2 works.

Internationalization

This chapter should explain how Sympa manages translations and how new tra

nslations can be added. It would also cope with common problems with encodings (mail, web, archives,..)

List management

A new chapter should deal about list management (rename, close, import, export).

Owner / Moderator howto

WWSympa should provide online help for owners / moderators, describing their role, tools.

Miscellaneous

List management

- List import and export features for listmasters

- Import subscribers via upload of a csv file
- Import archives via upload of archive from other MLM software format

SOAP related

- allow scenario testing : submitting a message via the soap interface could be a good API for thoses who need a milter in order to reject message during smtp session.
- soap_extensions

Other

- For a message distributed to multiple list, block multiple distributionto a same subscriber
- Prefered skins per user.
- CSS : fix some display bugs
- Display last login date and ip when login
- Display multiple message in a single web page in archive
- Check compatibility with IPV6, for example allow access condition based on IPV6 address in scenario
- Attributs utilisateurs par liste (French only)
- External plugins in archived.pl
- Handling multiple similar notifications