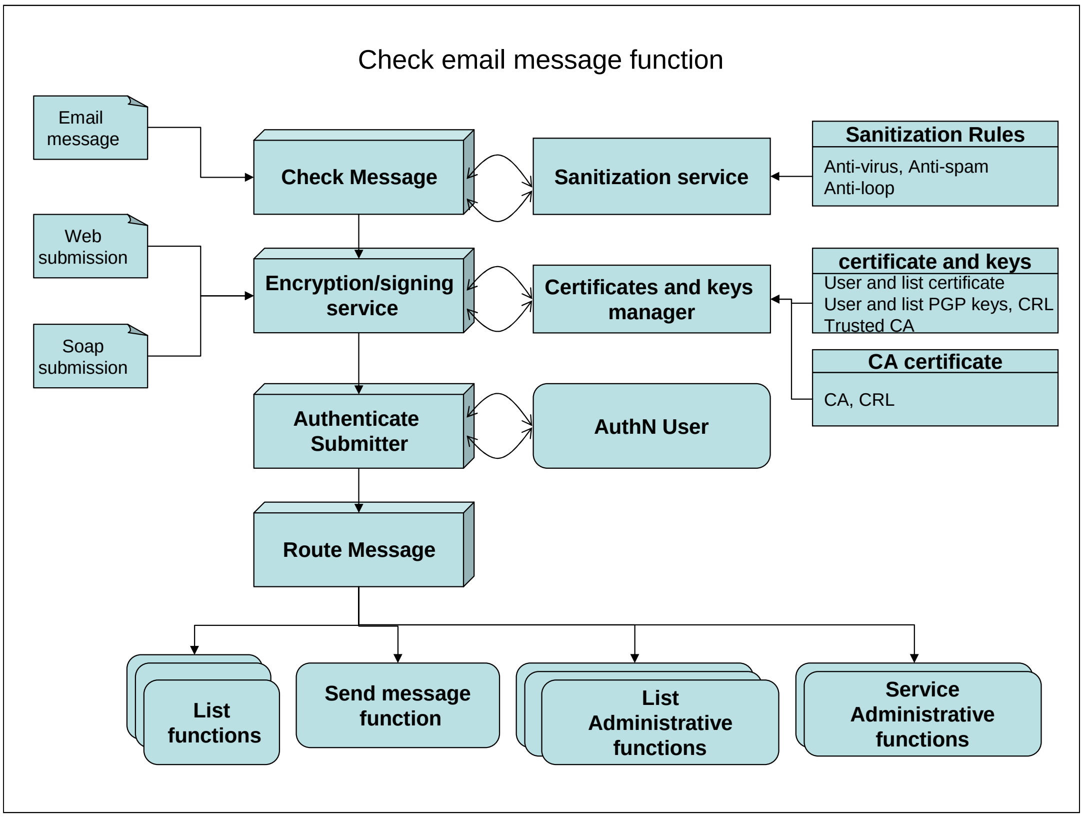


In most case operation requested by WEB or SOAP do not cross check message (except when composing a message to a list via the web interface.)

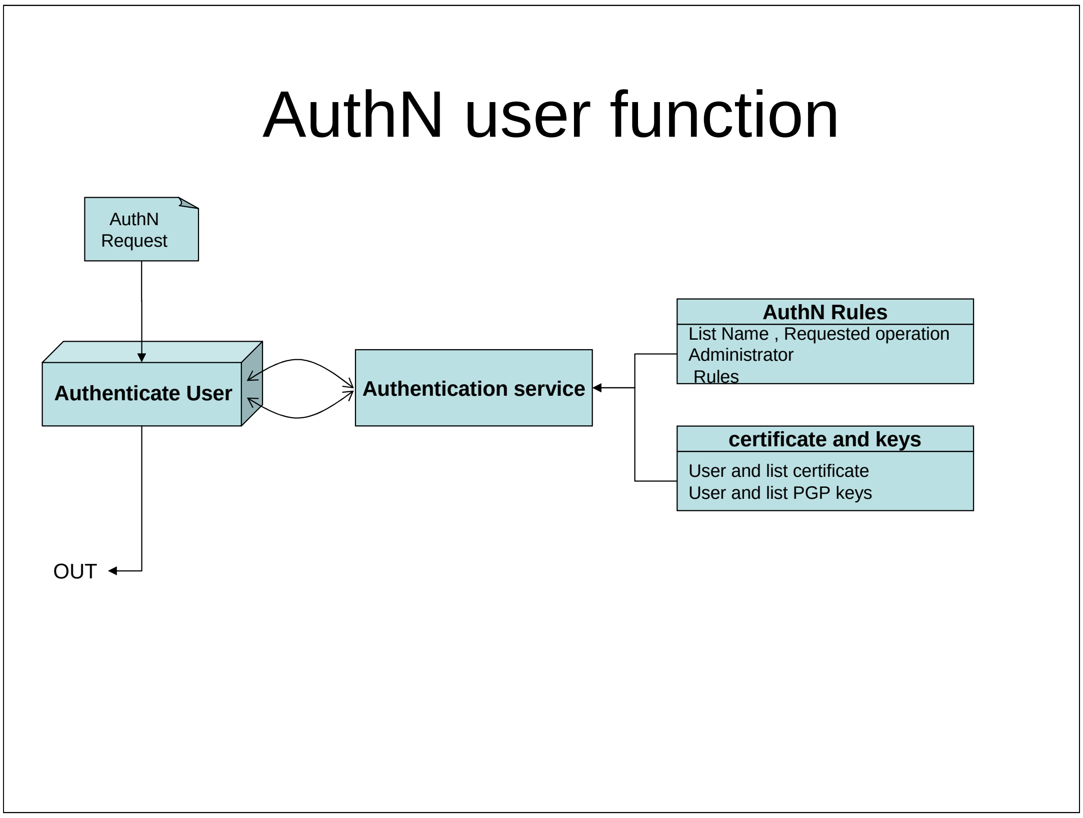
Check Message is the first operation. The meaning of this service is different : AuthN/Z of course is applied later. Now it's just to remove junk, spam and loops.

Encryption/signing service verify message signature and decrypte encrypted message

The authentication Service now point to AuthN/Z function

Join list function is renamed list functions because it was too specific to one of the list operation.

Service administration functions are related to operation which require listmaster privileges.



AuthN user function

AuthN Request

Authenticate User

Authentication service

AuthN Rules

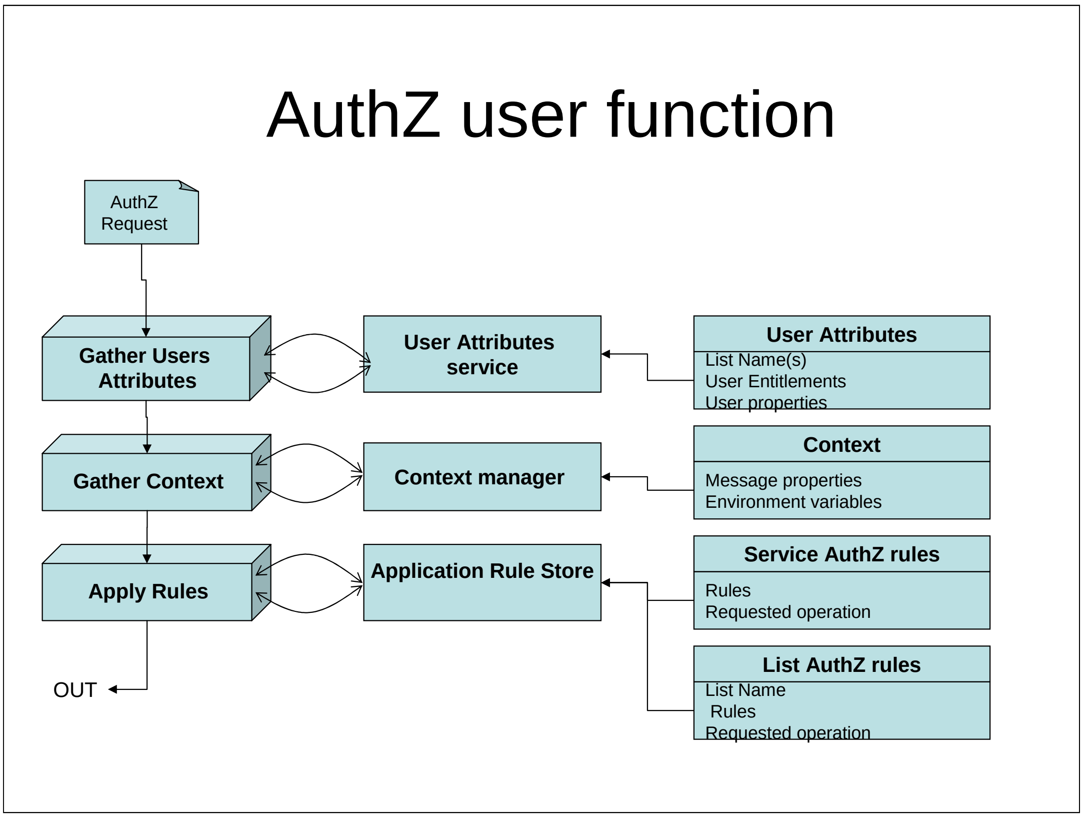
List Name , Requested operation
Administrator
Rules

certificate and keys

User and list certificate
User and list PGP keys

OUT

AuthN service is defined with auth.conf file : select auth method (passwd, smime, pgp) and backend (ldap, internal password). The AuthN Rules depend on the requested operation (removing a list need privilege but also strong authentication)



AuthZ user function

AuthZ Request

Gather Users Attributes

User Attributes service

User Attributes
List Name(s)
User Entitlements
User properties

Gather Context

Context manager

Context
Message properties
Environment variables

Apply Rules

Application Rule Store

Service AuthZ rules
Rules
Requested operation

List AuthZ rules
List Name
 Rules
Requested operation

OUT

Application rules renamed service rule (There is no rule in Sympa itself but each site can chose their rules)

We replace User privileges by user attributes which applies to both user role (is_owner, is_subscriber, …) and User properties such as LDAP attributes (student, gender, …)

The user attributes store is not only a storage but can also provide attributes from external data source such as LDAP directories so we rename attribute store into attribute service

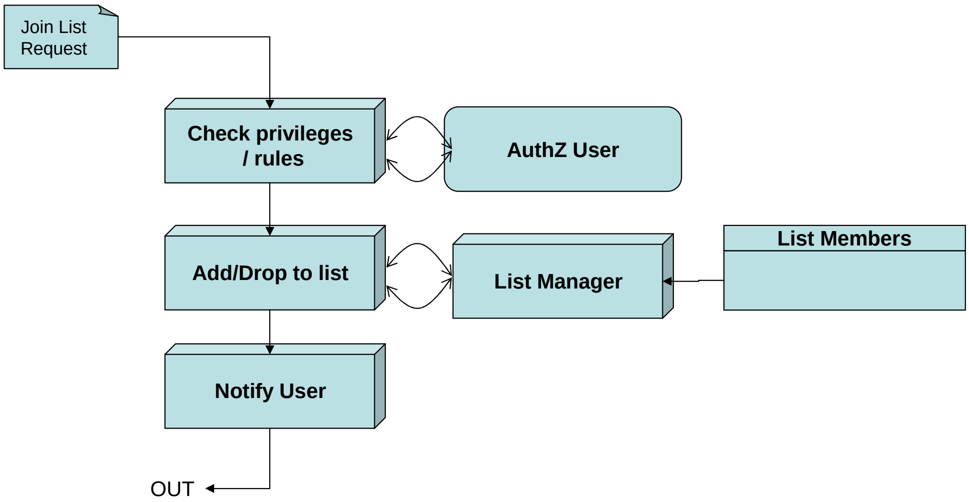
New layer : Gather context : The rule needs not only user attributes but also some other context information.mainly message properties when the operation is distribution message. That way Sympa can test if a message contain attachement, the message size and so on and use this message properties when applying rules.

Application rules is renamed AuthZ rules, it match Sympa’s authorization scenario concept

Apply Rules is almost equivalent to Sympa subroutine “request_action”

AuthZ Rules are split for application

Join / leave function



Join List Request

Check privileges / rules

AuthZ User

Add/Drop to list

List Manager

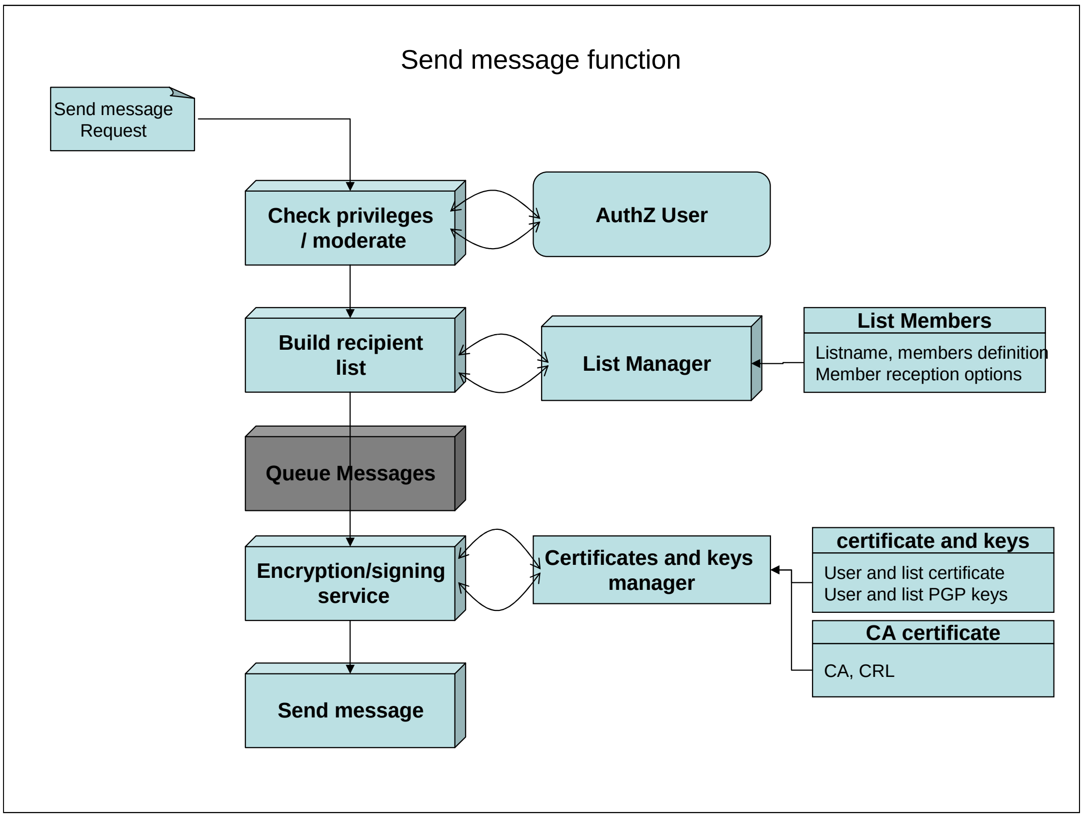
List Members

Notify User

OUT

As we understand your model Gather rule is no more needed because it’s defined in the authZ function

Notify user is controled by list rules, but those rules are stored in scenario (notify or not is returned by authZ function named request action). Quite strange behavior but so pratical in Sympa ☺



First apply authZ function

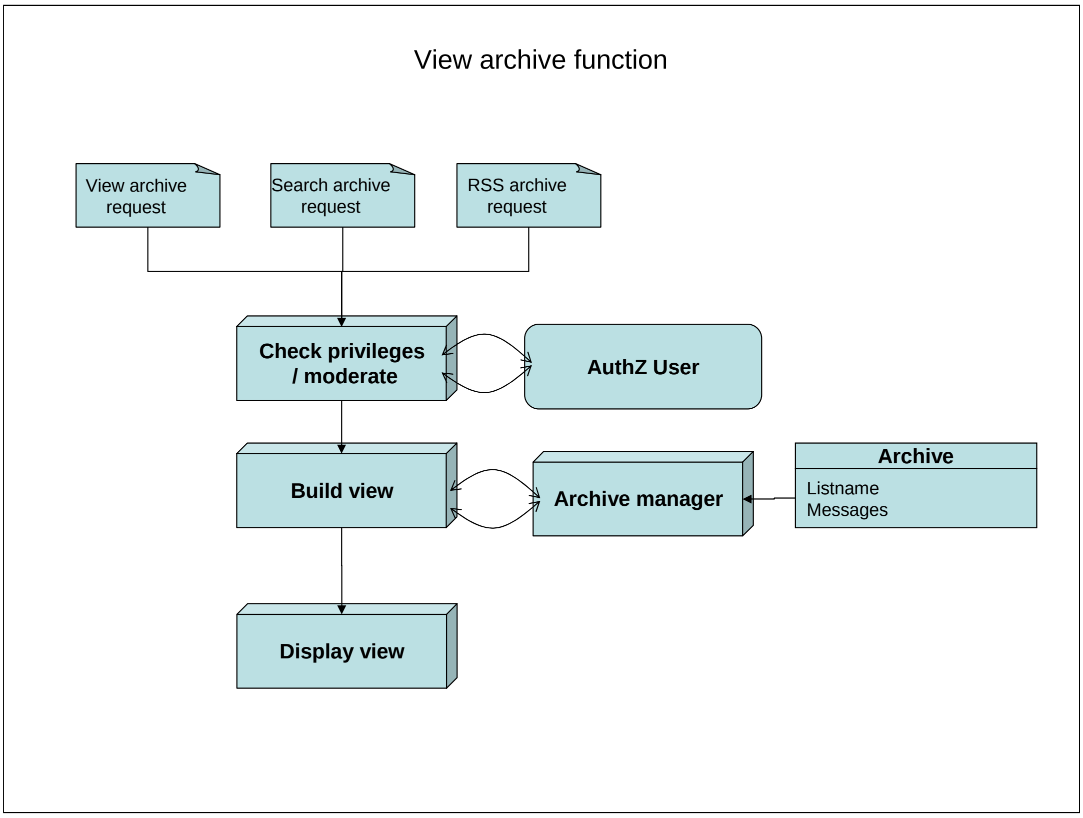
Build email renamed as “Build recipient list” ; this service exploits the members reception options (nomail, digest, summary, html,…).

List members can be either enumerated or refer to external data sources (LDAP, RDBMS, local or remote lists)

In Sympa, Moderation is handled by the AuthZ user function

Queue message : we do not understand this service

Encryption and signing service is optional : needed if message to distribute is encrypted.



Sympa web Archives include a search engine.

Now, archive can be published using RSS

First apply authZ function as for any other sympa function. A pretty example : list visibility function that control if a list can be viewed by someone is under control of AuthZ

From our understanding of your model, Select content is not usefull because authZ is not limited to a control of list members ; we use the complete authZ module

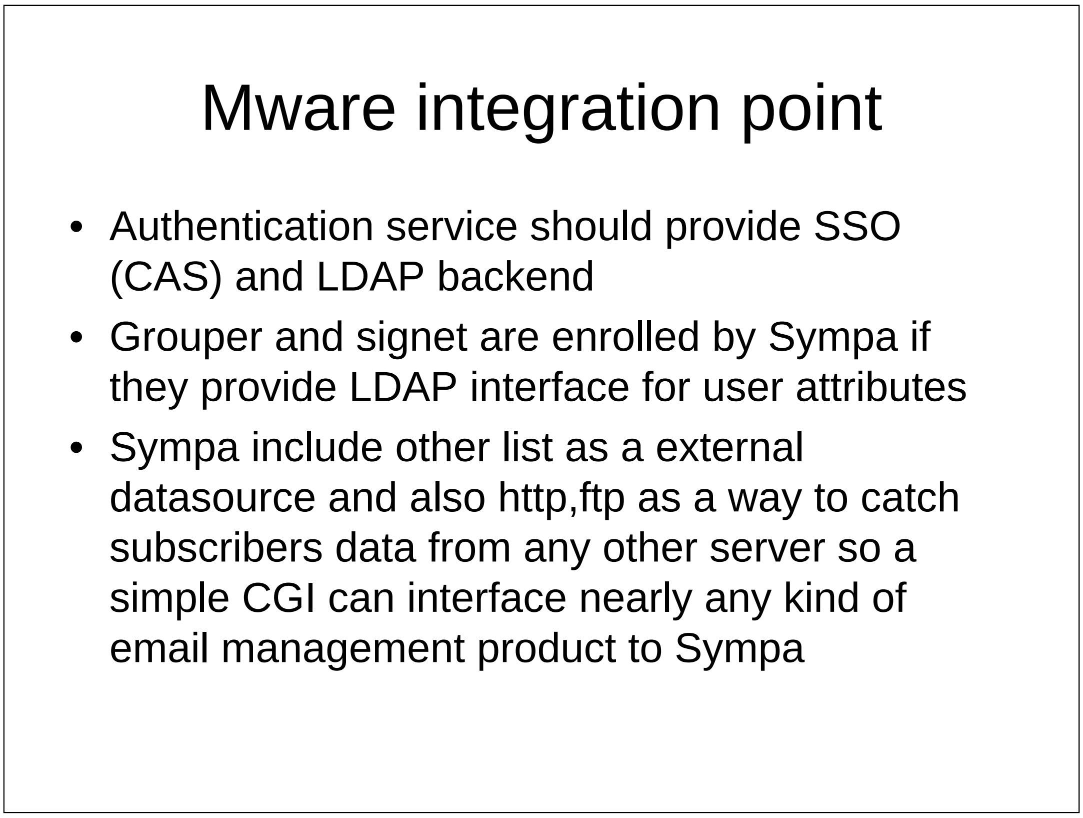
If requested by user the Display view service can be replace by a “resent message service”

List management function

- The model is suitable for Sympa except :
  - Assign Roles, define Rules, define Source are part of the edit_list function
  - Many other features in “change list”
- In addition Sympa provides :
  - moderation managment (message, subscribtion, documents)
  - Bounces management
  - Service messages customization
  - Archive administrative management
  - Remove and rename list
  - ….

End user Manage Preferences function

- No Change Bio function
- What is Publish list function
- No change set/display function (skins are requested by some users)
- AuthN prefs are limited to cookies life time and change password
- No other comment except that AuthZ is always requested.



Mware integration point

- Authentication service should provide SSO (CAS) and LDAP backend
- Grouper and signet are enrolled by Sympa if they provide LDAP interface for user attributes
- Sympa include other list as a external datasource and also http,ftp as a way to catch subscribers data from any other server so a simple CGI can interface nearly any kind of email management product to Sympa