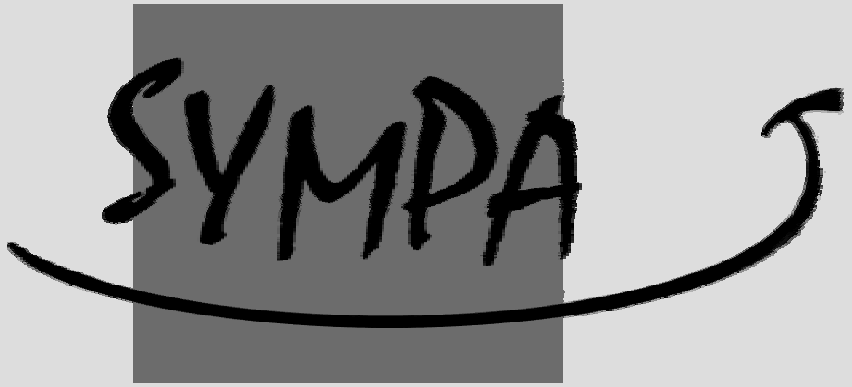
An Open Source, Middleware-enabled Mailing list server



Serge Aumont & Olivier Salaün

September 2004

CRU

CRU brief presentation, why Sympa

Sympa overview

The web document repository

Sympa organization, virtual robots

The template format

The SOAP interface

Dynamic mailing lists

Lists families

Privacy

S/Mime and Sympa

Authentication in Sympa

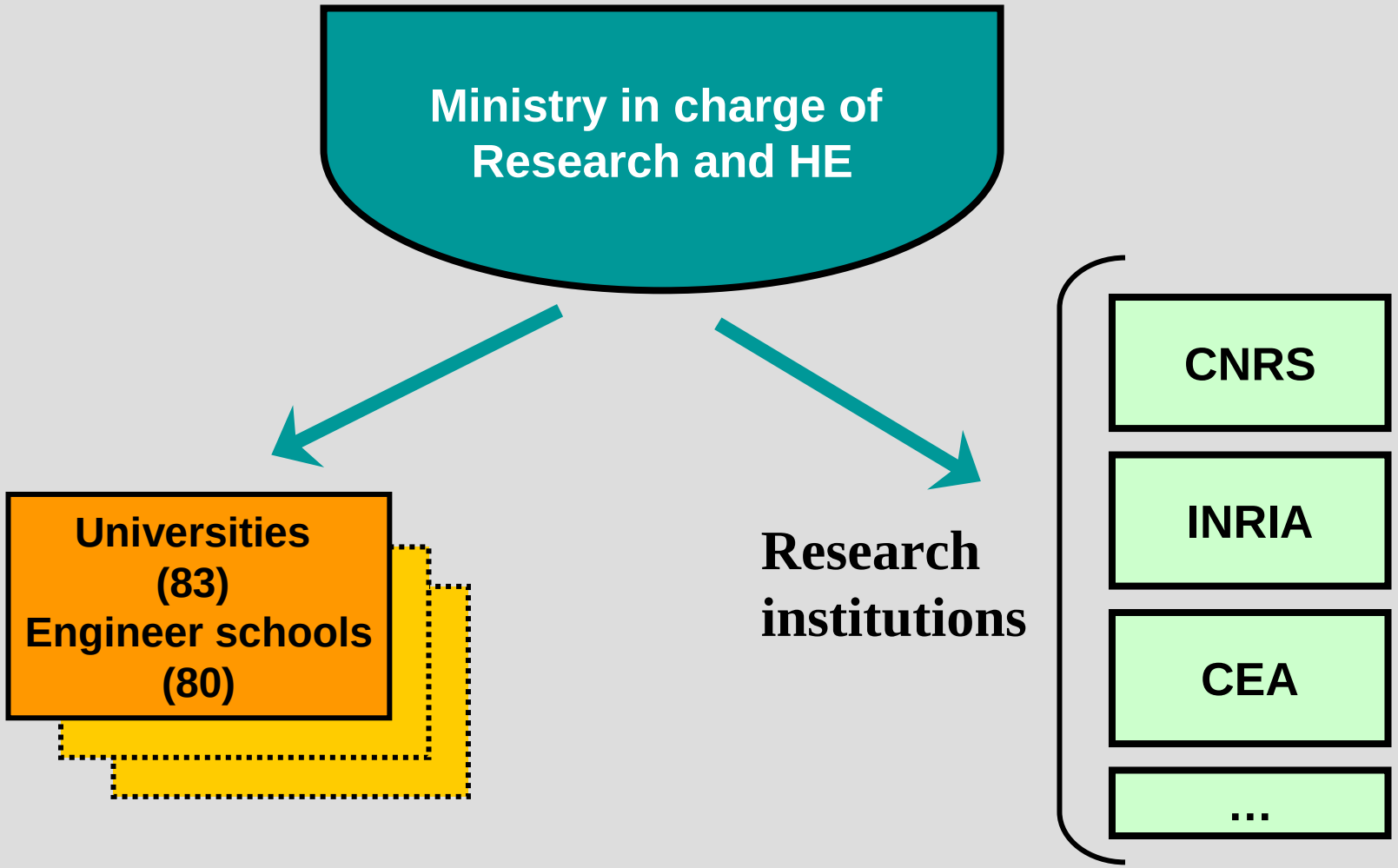
Access control management

Perspective

What is the CRU

- CRU stands for « Comité Réseau des Universités » (network committee for French universities)
- We do NOT operate a national academic network (=> Renater)
- The CRU is responsible for coordinating actions among universities and between universities and the ministry

Institutional view



Working forces in our environment

- Renater is operating the national academic and research network
- UREC, working for CNRS laboratories
- CRU, working for Universities

CRU activities / Training

- Training for universities technical staff :
  - Tutorials on targeted technologies (directories, groupware, video conferencing,…)
  - Yearly camps for security officers and network managers
  - Networking conference, held every two years (> 1000 participants, http://www.jres.org)

CRU activities / Working Groups

- Working groups coordination for topics such as :
  - Directories
  - Security of information systems
  - Authentication and authorization
  - Wireless and mobility
  - Anti-spam
  - ….

CRU activities / Services

- We operate national services including :
  - PKI for universities
  - SourceForge like service for university projects
  - Mailing lists service, mainly for virtual organizations
- We develop Sympa

What is Sympa ?

- Sympa is a mailing list software developed by the CRU
- Sympa is an open source software (GPL licence)
- Sympa has been designed to meet universities needs (middleware enabled)
- Sympa is internationalized (English, Chinese and most European languages)

Sympa genesis

- Sympa inherits experiences from a previous software, designed as a Listserv replacement, to migrate from BITNET
- First developments started in 1997 with members storage in a RDBMS, a built-in bulk e-mailer
- Later Sympa included a web interface, A&A architecture, LDAP support, S/Mime encryption, Virtual hosting, SOAP interface,…
- It is now a quite sophisticated communication tool used by 90% of French universities and others

The design process…

- 2 key elements that help us in the design process :
  - We are running a major ML service ourselves
  - We have good relations with our user community
- Sympa tends to make administration job as light as possible via automation and delegation
- We try to transfer technologies in Sympa when needs arise at the university level (LDAP, S/Mime, Single Sign-on, Shibboleth)

Who uses Sympa

- 4 000 known sites
- Universities and schools
  - 90% of French ones + others (Europe, USA, south America, Asia)
- Government agencies
- Service providers & Open source software firms
- Non-profit organizations
- Private companies

CRU brief presentation, why Sympa

Sympa overview

The web document repository

Sympa organization, virtual robots

The template format

The SOAP interface

Dynamic mailing lists

Lists families

Privacy

S/Mime and Sympa

Authentication in Sympa

Access control management

Perspective

What makes a ML software better than another

- End-user features (doe not make the difference, apart from side groupware tools)
- Ergonomics and documentation
- Scalability (big lists, many lists)
- Security features (spam, loops, virus, abuse, data protection,...)
- Group management tools (moderation, access control, bounces management)
- Ease of administration (list creation, user support, customization)
- **Connections with the information system (group definition, privilege definition, authentication, integration with portals)**

Sympa / End-user features

- Basic features : subscribing, sending a message, unsubscribing,…
- Advanced features :
  - Reception modes (digest, urlize,…) and preferred mail format (text/plain, text/html)
  - List of subscriptions
  - Web archives (search engine, access control, email addresses protection)
  - Shared document repository

Sympa / the web interface

- Portal to the ML service (not just to each list)
- Each web page is adapted to user privileges (private ML are not advertised)
- 3 in 1 :
  - List member (and anonymous) features
  - List management features
  - Service admin. Features
- Flexible user authentication
- Access control (equivalent on the mail interface)

Sympa / Security features

- Message submission is restricted (private, moderated,…)
- Mail commands can be configured to require confirmation or a valid cryptographic signature
- Sympa includes an antivirus plug-in
- Advanced loop detection system
- User email addresses are protected from “spam harvesters”

Sympa / Group management tools

- Group management responsibility is shared between :
  - list owners (members)
  - moderators (contents)
- Add / Review / Remind / Remove members
- Automatic bounces management
- Moderation of :
  - Subscriptions
  - Messages
  - Shared documents
- Customization of the ML behavior, access control and service messages

Sympa / ML service administration

- List creation / deletion / renaming is performed via the web interface. List creation can be moderated
- List setup by list owners is controlled
- Virtual hosting is supported (virtual robots)
- Sympa is highly customizable (templates for web pages and service messages, authorization scenarios).
- Delegation :
  - User support is delegated to list owners
  - A privileged owner manages other owners
  - Each virtual robot has a set of listmasters

Sympa / Scalability & Perf

- Sympa allows high performances via :
  - User data are managed by a RDBMS (MySQL, PostgreSQL, Oracle or Sybase)
  - Message delivery is performed by the MTA (Sendmail, Postfix, Exim or Qmail)
  - Web service is provided by a memory-resident process (using FastCGI)
- Sympa can cope with :
  - Big mailing lists (we known up to 200 000 members)
  - Many mailing lists (we known up to 20 000 ML)

Sympa / Middleware connections

- Group definition can be based on LDAP
- User privileges can be deduced from LDAP filters / Shibboleth attributes
- Multiple authentication back-ends are supported
- A subset of Sympa features is accessible from within another application via a SOAP service

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
Lists families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

The web document repository

- Document sharing space attached to a mailing list
- Provided on sympa web interface
- Benefits from group-based access control (read, write and control privileges)
- Privileges can only be restricted in the document hierarchy (security concerns)
- Quota can be defined

Document repository features

- Objects manipulated :
  - Files (create, upload, edit, remove)
  - Folders
  - Bookmarks
- Access management on each object :
  - Read access
  - Write/edit access
  - Change ownership

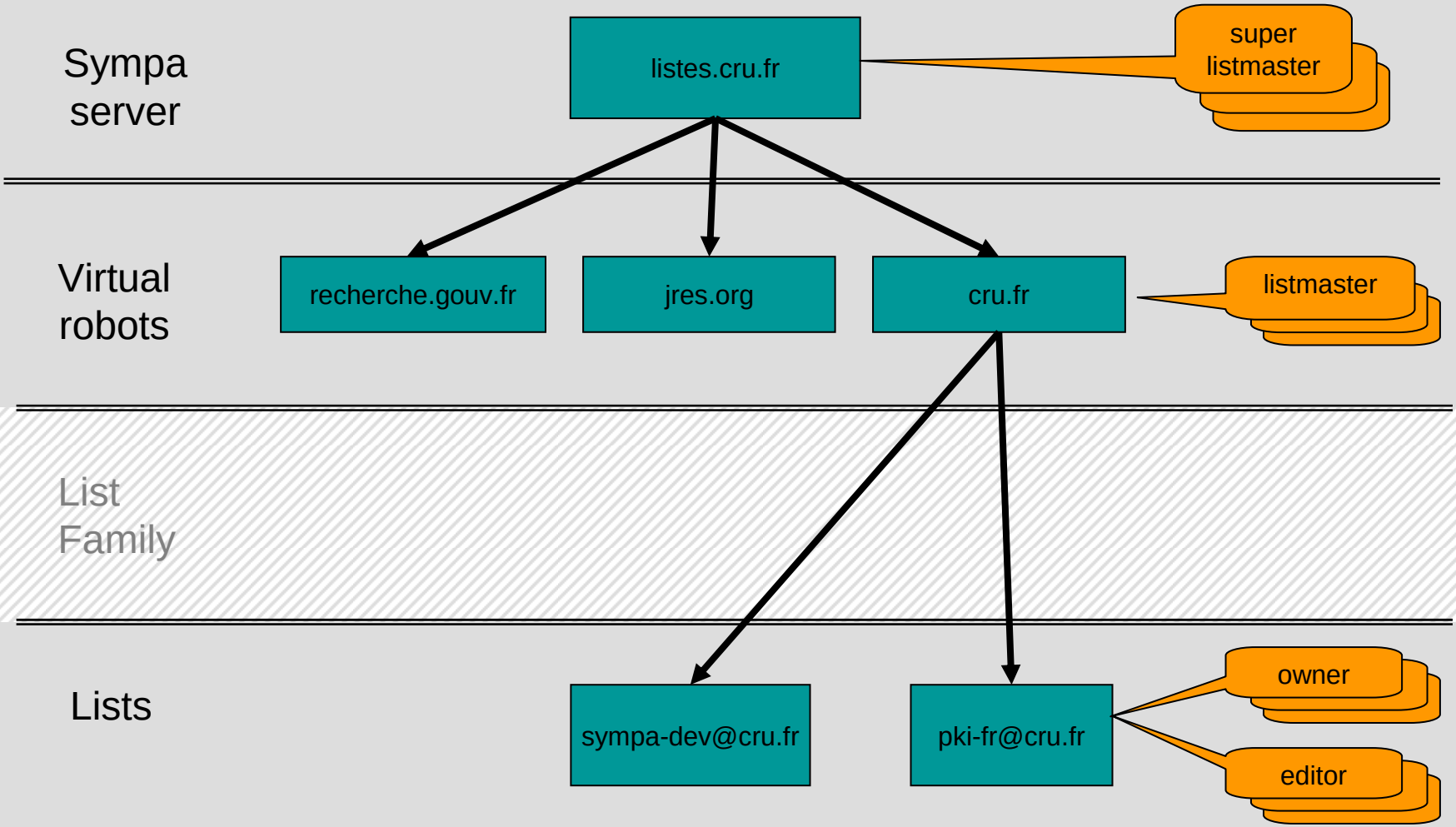
Document repository usage

- On our site, the documents repository is an irreplaceable tool for virtual organizations such as :
  - Technical working groups
  - Groups of researchers / teachers
  - Conference program comitees
  - …
- The goal is NOT to provide an advanced CMS. It is just a light groupware tool !

Demo wwsympa

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
Lists families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

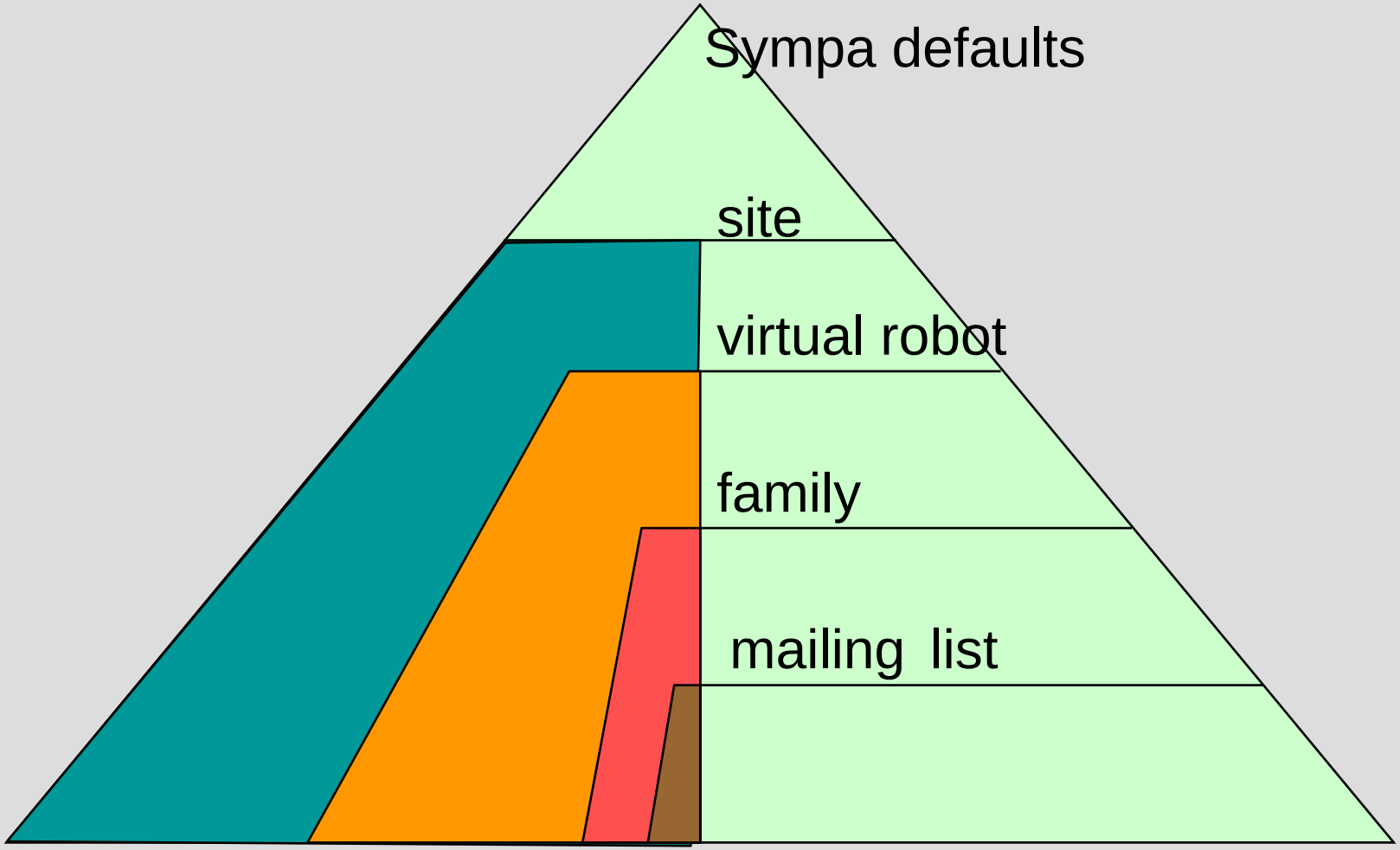
Sympa hierarchical organization



Sympa hierarchical organization

- 3 levels organization levels with associated management roles
- Inheritance mechanism applies to :
  - Configuration files and parameters
  - Authorization files (scenarios)
  - Web templates
  - Mail templates
- Defaults are defined at the higher level when possible
- The higher level is the distribution of Sympa, providing defaults for every customizable piece

Sympa defaults mecanism



Virtual robots

- Virtual hosting feature
- Each robot can be configured separately
- Strict partitioning of list environments
- Each virtual robot is seen as a distinct ML service with its own web and mail interface

Virtual robots deployment

- Virtual robot creation is light :
  - >Can be generalized for small subset of lists
- Requires virtual host definition :
  - On the mail server
  - On the web server
- A single instance of each daemon will serve all virtual robots

Sample virtual robot organization

- /home/sympa/etc/demo.sympa.org/
  - edit_list.conf
  - robot.conf
  - topics.conf
  - scenari/
    - send.default
  - mail_tt2/
    - welcome.tt2
    - list_created.tt2
  - web_tt2/
    - menu.tt2

Sample virtual robot configuration robot.conf

```
http_host  demo.sympa.org
wwsympa_url  http://demo.sympa.org/wws
title A demo  mailing list service
listmaster  bid@cru.fr,dule@cru.fr
create_list  public_listmaster
default_home  lists
lang  us
default_shared_quota  5000
soap_url       http://demo.sympa.org/soap
```

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
Lists families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

Templates in Sympa

- Web interface made of 67 web templates
- Mail service messages (welcome message, help file, digest format,…) made of 34 mail templates
- These templates allow to :
  - Separate the code and the layout
  - Customize the User Interface to meet the site needs
- Using TT2 format (http://www.tt2.org)
  - Open Source Template Toolkit
  - Relation with PO catalogues (GNU)
  - Recent introduction in Sympa…

The TT2 format provides…

- Variable substitution
- Includes
- Conditionals
- Loops
- Support for plug-in objects (I18n, MIME encoding,…)
- Support for complex Perl data types
- Compilation and caching of templates
- Security features
- …

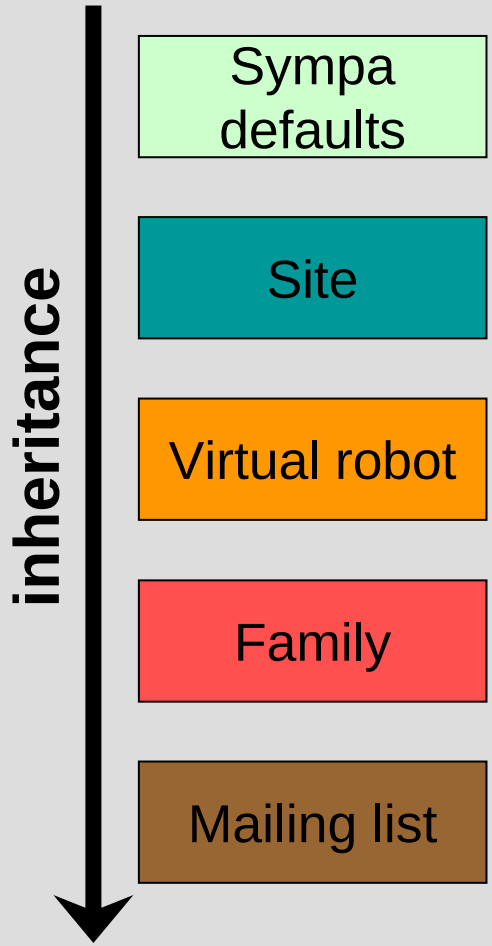
Sample mail template global_remind.tt2

```
Summary of your subscription (using the e-mail [% user.email %]).
If you want to unsubscribe from some list, please save this mail.

Foreach list here is a mailto to use if you want to unsubscribe.

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[% FOREACH l = lists %]
[% l %] mailto:[% conf.sympa %]?subject=sig%20[% l %]%20[% user.email %]
[% END %]
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

[% IF user.password %]

In order to authenticate on the web interface([% conf.wwsympa %])
use your e-mail [% user.email %] and your password [% user.password %]

[% END %]
```

Sample web template list_menu.tt2

```
…….
[% IF is_editor %]
  [% IF mod_total == '0' %]
    No message to moderate
  [% ELSE %]
    Messages to moderate :<B> [% mod_total %]</B>
  [% END %]
  [% IF shared == 'exist' %]
    <BR><BR>
    [% IF mod_total_shared == '0' %]
      No document to moderate
    [% ELSE %]
      <B>Documents to moderate :<B>
      [% mod_total_shared %]</B>
    [% END %]
  [% END %]
[% END %]
…….
```

Templates / defaults mechanism



- Sympa default templates apply if they were not redefined elsewhere
- Default welcome template defined at the site level
- An adapted welcome template is redefined for a family within the site
- List owners are allowed to edit the welcome message for the mailing list they manage

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
Lists families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

Sympa SOAP interface

- Provides access to Sympa services from within another program
- This is the a better solution than allowing other software to access Sympa data because :
  - Data format evolve…
  - Access control and authentication are applied
- Mainly used to include sympa features in an wider framework (Uportal sympa chanel)

Sympa SOAP features

- Currently provides a limited set of features including : login, which, lists, subscribe, signoff, review
- We might extend the set of features (if needed): archives, list creation, add, del,…
- Authentication is based on password or token (CAS proxied credential)
- SAML is considered…

Sympa SOAP clients

- SOAP services are described by a WSDL document
- Most programming languages provide a SOAP library
- Sympa SOAP server has been tested with the following libraries :
  - SOAP::Lite (Perl)
  - Axis (Java)
  - NuSOAP (PHP)
- A sample PHP interface to Sympa demo service : http://demo.sympa.org/sampleClient.php

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
Lists families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

Old style mailing list

- ADD, DEL, SUB, UNSUBSCRIBE
- Using the list as a management tool for groups
- Administration task duplicated or asynchronous update of list member by some scripts
- Often sub-lists are added to other lists
  - Unsubscribe, error management, subcribtion option ?

Dynamic mailing list

Subscribers defined by includes

- file
- other list
- other list on a remote sympa server
- sql query to an external database
- LDAP search

Mixed methods

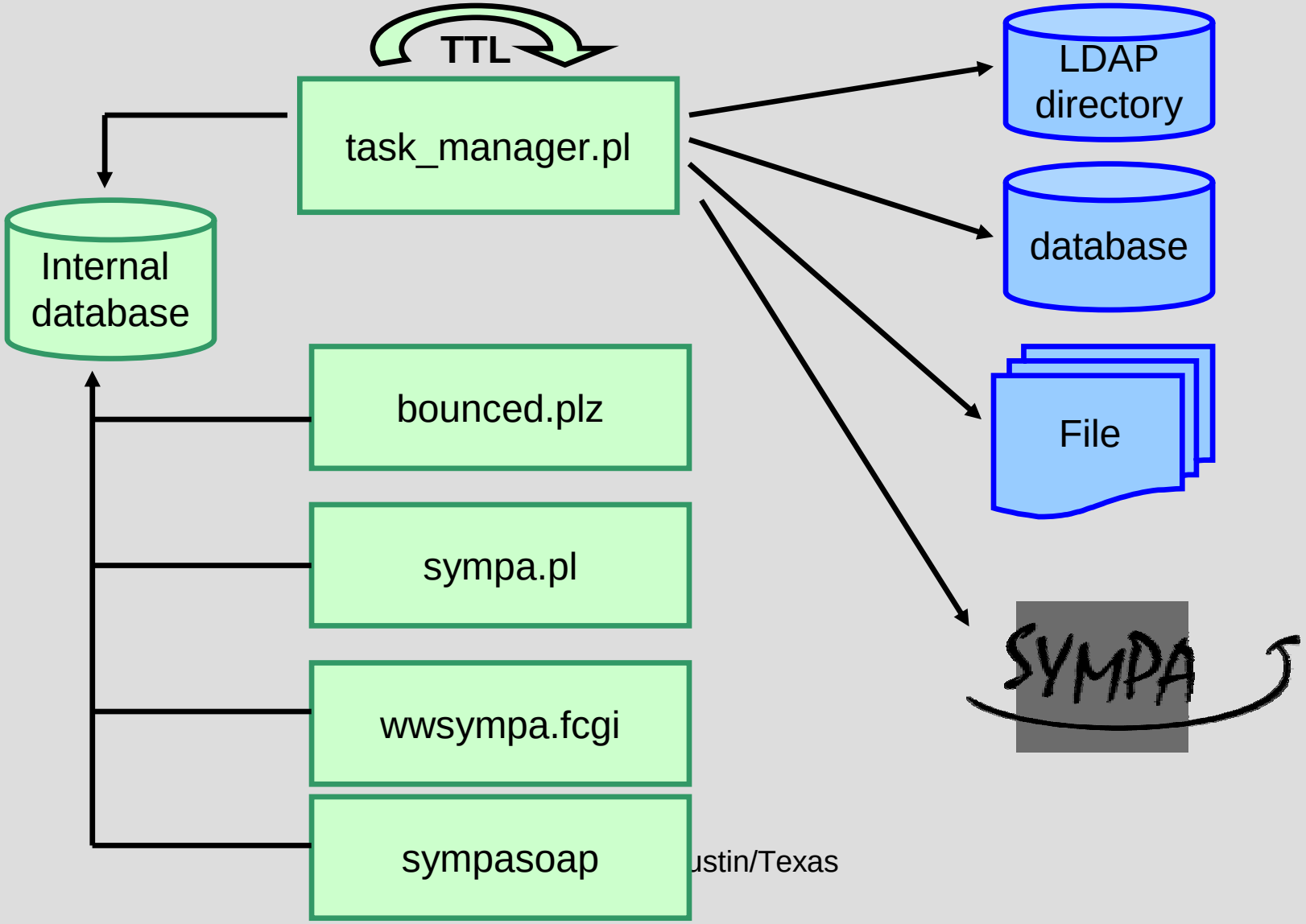
Dynamic mailing lists / perf

Performance issues are important

- Sympa home page : “which”
- Visibility privilege evaluated for each list

1. Sympa use an internal cache (RDBMS)
2. Each data source is defined with a TTL
3. task_manager.pl performs asynchronous update

Dynamic mailing list



Include LDAP

include_ldap_query

host ldap.cru.fr,replica.cru.fr:387

suffix dc=cru, dc=fr

filter (&(student=math) (dc=fr))

attrs mail

select first

timeout 10

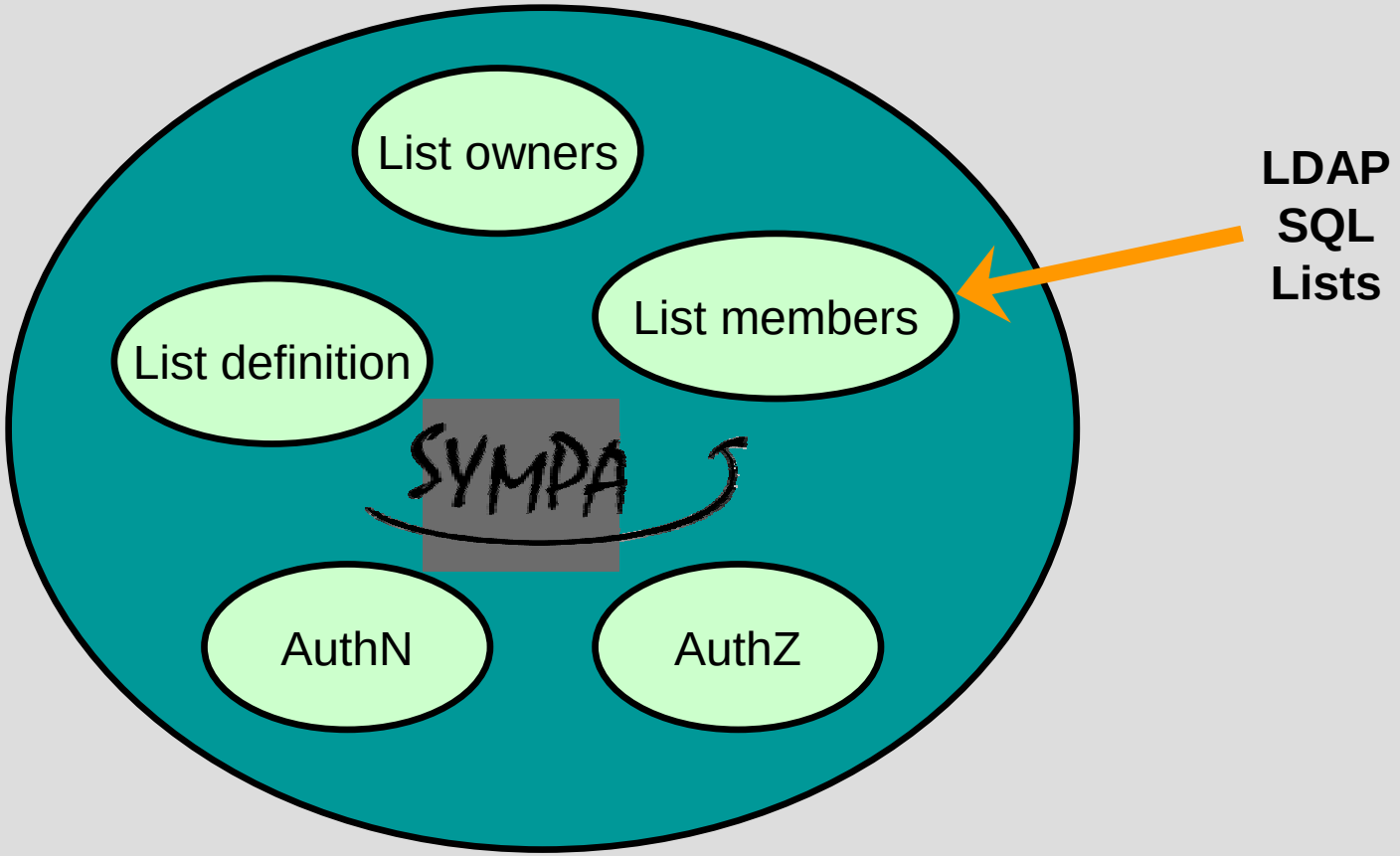
Include LDAP

(contrib from Dalbec ysu.edu)

First LDAP query to select a group

```
include_ldap_2level_query
host ldap.univ.fr
suffix1
ou=Groups,dc=univ,dc=fr
scope1 one
filter1 (&(objectClass=groupOfUniqueNames) (|
    (cn=cri))
attrs1 uniquemember
select1 all
suffix2 [attrs1]
scope2 base
filter2
(objectClass=n2
pers)
attrs2 mail
select2 first
```

For each group member fetch his email adress


List owners
List definition
List members
SYMPA
AuthN
AuthZ
LDAP
SQL
Lists

Dynamic mailing list

- Compatible with LDAP backup server and LDAP/SSL
- A list can use several different external data sources
- Compatible with bounce management and subscription user options
- Widely used in French universities and schools

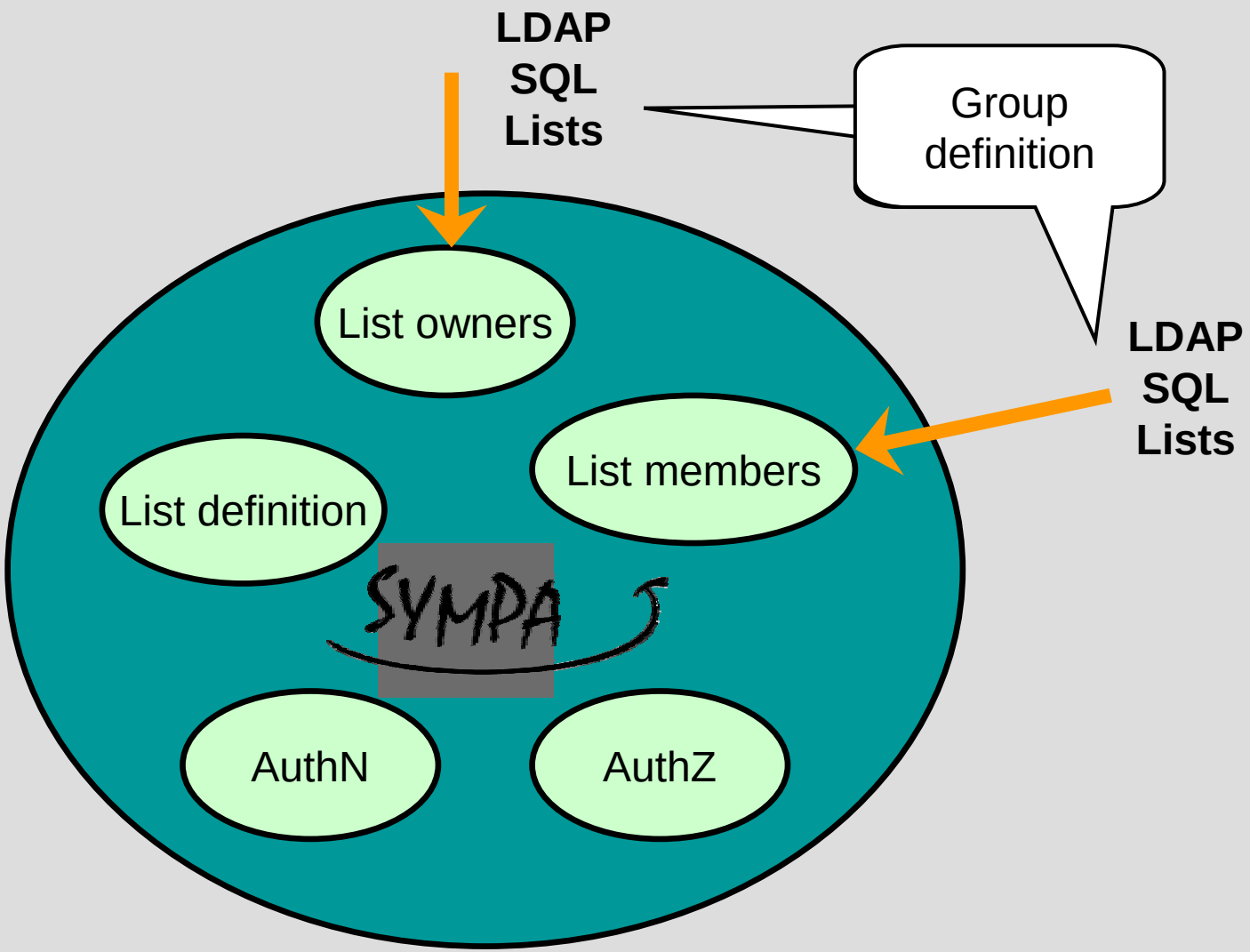
demo

- List configuration overview (data_source)
- review

List owners definition

(recent development)

- List members = any person where department=finance & is a student
- ➢ List owners = any person where department=finance & is from staff.
- List Owners and list editors can be defined in the same way as list members (LDAP,SQL,…)


LDAP
SQL
Lists
Group
definition
LDAP
SQL
Lists
List owners
List members
List definition
SYMPA
AuthN
AuthZ

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
Lists families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

In the past

- List typology is possible (newsletter, private working group, public forum, student group for administrative information)
- How to manage a huge number of lists for each population in a university ?
- Usually listmaster creates these lists based on data coming from the information system by using a home made script and a template
- tamu.edu : 20.000 lists using Sympa

Why lists families

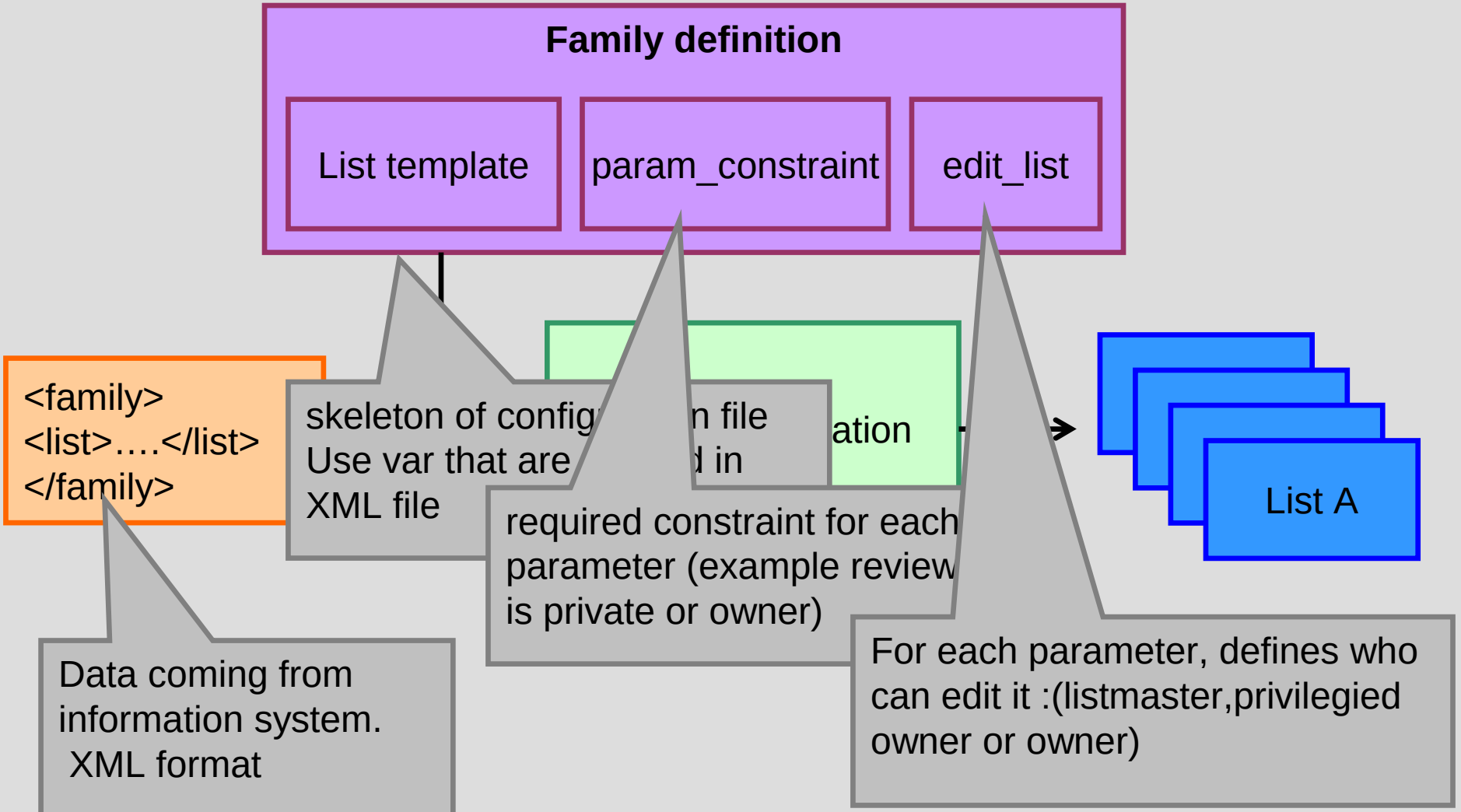
- After the initial creation it’s hard to manage the set of created list.
  - Changing some details of the list template erase the list customization
  - List owners can edit there configuration and change the type of the list out of control from listmaster.
  - How to identify lists to be deleted ?
- We need a management tool for not only for lists but for sets of lists.

Lists families in SYMPA (recent dev)

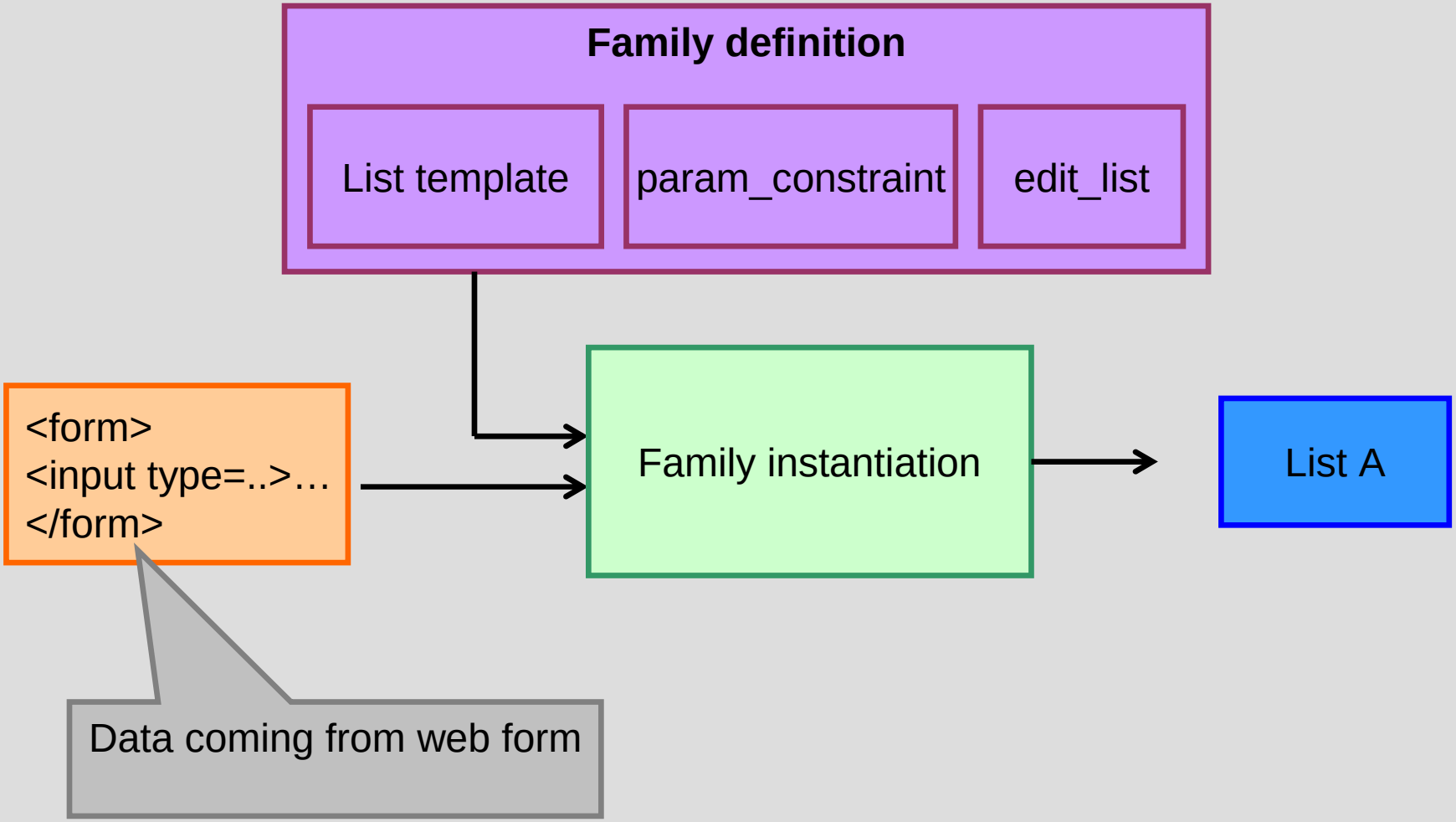
- Lists families allow creation and administration of large number of lists according to a list policy
- list configuration is complex : about 50 parameters.

➢ Choosing a family for a list makes configuration easier (ex : newsletters have a lot of common parameters)

Lists families in Sympa



Lists families in Sympa



XML data and list template

```
<list>
<listname>networking.ipv6</listname>
<section>computer_science<section>
<option>networking</option>
</list>
<list>
<listname>…</listname>
<section>…</section>
…
</list>
```

```
subject students of [section]/[option]

title all students from department [section],
    course [option]
web_archive private

include_ldap_query
Host ldap.foo.edu:389,backup.foo.edu
suffix dc=foo, dc=edu
filter(&(section=[section])(option=[option]))
attrs mail

owner_include
Include_ldap_query
…..
```

param_contraint.conf

```
# very simple example
# parameter edition control
send private_smime, editor_key
web_archive private,owner
shared_doc.d_read private,owner
Shared_doc.d_edit owner
```

Control of list parameters edition

- Roles : owner, privileged owner, listmaster
- List owners need to choose some parameters themselves. Ex : who can subscribe ?
- Super listmaster needs to control some parameters per family and per robot. Ex :
  - Maximum message size
  - Initial list owner (responsible for the list, the privileged owner can delegate list management but can’t transfer completely the list to someone else)
  - ….

edit_list.conf

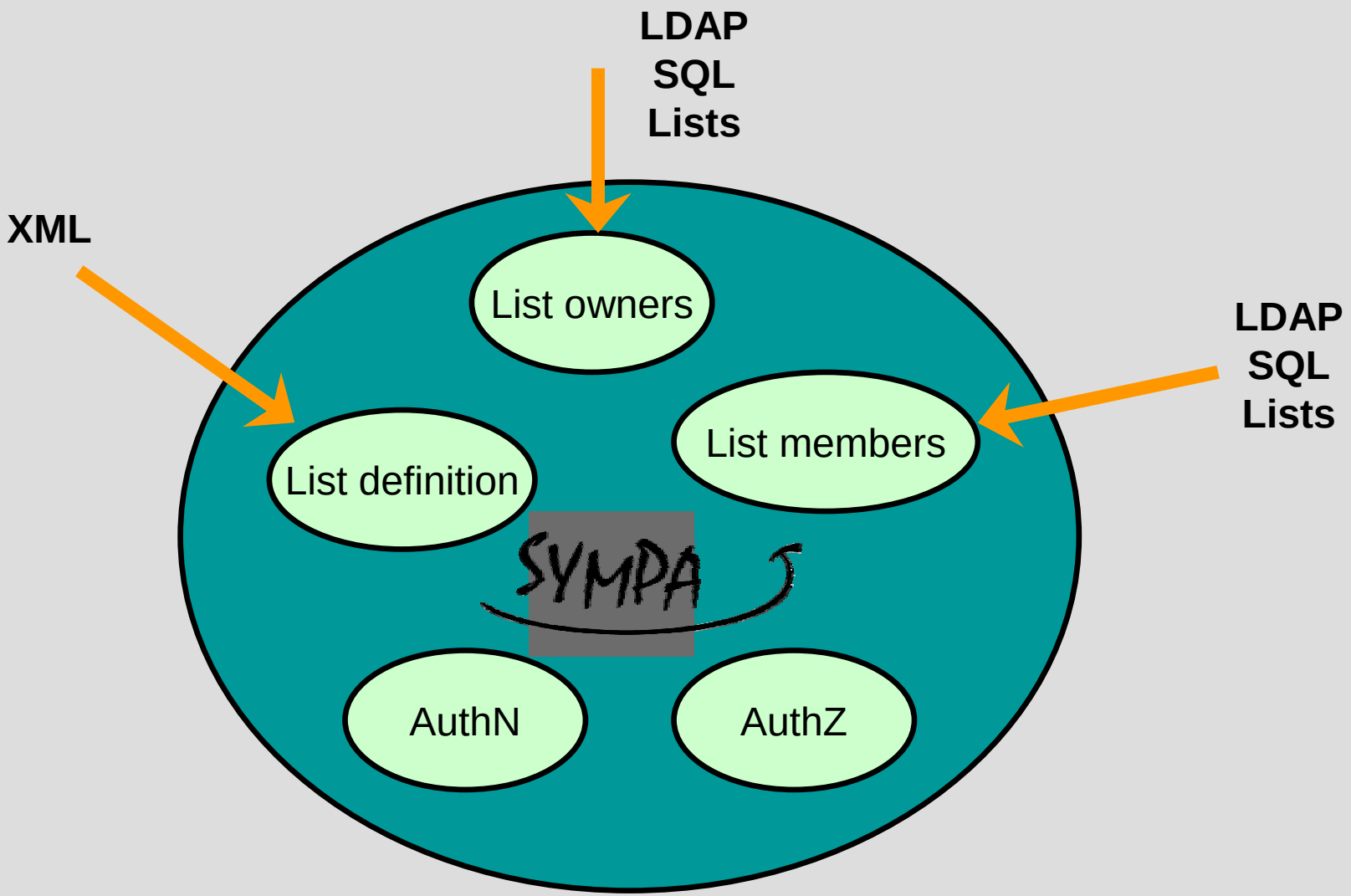
```
user_data_source owner,privileged_owner  hidden
ttl              owner,privileged_owner  hidden
shared_doc       owner,privileged_owner  write
review           owner                   read
review           privileged_owner        write
footer_type      owner,privileged_owner  hidden
owner            owner                   read
owner.profile    owner,privileged_owner  read
owner            privileged_owner        write

default          listmaster              write
```

Lists families for simple list definition

- List depends on family and robot
- Family depends on a particular robot or can be defined for all robots
- Existing default family (no restriction)
- Existing default robot
- A basic list can be defined with only 3 parameters
  - listname
  - Owner email
  - List title
- A sophisticated set of properties can be defined for a list by setting its family

Lists families in Sympa

- Available in current CVS development branch
- Include default families
- Allow multiple families definition at each level
  - robot
  - site
  - distribution.
- Lists family, a tool for :
  - technical implementation of a mailing list service policy.
  - Large number of lists automatically inherited from the information system


LDAP
SQL
Lists
XML
List owners
LDAP
SQL
Lists
List members
List definition
SYMPA
AuthN
AuthZ

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
Lists families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

Privacy/archive

- Spammers use spam harvester
- Googling is a peril

Sympa solution :

- Sympa hides emails using javascript (google indexation is possible) or by a form (all automatic process blocked in archives)
- X-No-Archive header feld
- Users can delete there own posts from archives

Privacy

- Access to member list and archives are controlled
- List option : anonymous mode
- Many method to unsubscribe
- OPT-in traceability (in project)

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
List families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

S/MIME in Sympa : goals

1. Any available MIME features must be useable with Sympa, **including S/MIME**
2. Mailing List supporting S/MIME can help to deploy S/MIME
3. Distribute signed messages without signature alteration (as any other message)
4. When recognizing the sender of a message in order to authorize some operation, Sympa verifies recognize message signature.
5. Distribute encrypted message in a encrypted form to each subscriber

Signed messages

- Distributing a signed message is fairly easy.
- Never add a message footer whatever says the list configuration
- Decode and re-encode a message will probably break the signature

Signed messages for authentication

- In some cases we need a real authentication without email challenge.
- Authentication requirement depends only on list and operation requested
- web and mail interface must provide the same authentication level (if S/MIME signature is required, HTTPS authentication using user certificate too)

Signed messages for authentication

- Sympa recognizes S/MIME signature and uses this verification in the process of authorization
- It checks that the message sender and the message signer are the same
- Because S/MIME signature does not guarantee any header integrity, commands in the subject are applied but not assuming that they are part of valid S/MIME signature
- Message is added to archive including signature, users can’t check it from the web interface, but they can ask to receive the original message.

Distributing encrypted messages

- Each list can hold a certificate and a private key (key usage bits should be set to allow e-mail signing, DN)
- Anyone can send an encrypted message to the list
- Sympa decrypts the message using the list private key, then sends it to each subscriber using his certificate
- The message is stored in an encrypted form in archives

Managing certificates

- Lists certificates and private keys are stored by Sympa in the list directory

```
p12topem.pl --pkcs12 cert.p12 –listname foo --robot cru.fr
```

- Trusted CAs shared with Apache config (internaly use openssl)
- When receiving a signed message Sympa stores the user certificate in a cache in order to encrypt some messages for this user

Managing certificates

- Lists can use 2 different certificates for signature and encryption
- Subscribers can use 2 different certificates for signature and encryption
- During the renew certificate period, Sympa uses both old and new certificate so every user don’t need to switch at the very same time
- List certificate can be loaded from web interface

DEMO

- https authentication
- Load list certificat (IE)
- Post a mail -> request auth
- Post a signed message
- Post a crypted message
- Show archive : resend

Limitation & TODO

- Encryption performance issue for large groups :
  - Only 1 recipient for each message
  - Full encryption algorithm for each subscriber where the symetric encryption could be factorized
  - Is there a real need to encrypt for large groups ?
- Super-listmaster can decrypt any message
- CRL and OCSP are not yet part of OpenSSL s/mime features.
- LDAP search of certificates would be nice
- PGP is on the way : Can be used as a “S/MIME / PGP gateway” (this contrib as been made for a Austrian CERT)

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
List families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

Multiple authentication methods

- For many usage, some list members have no organizational affiliation other than their membership in the list itself
- We want Sympa opened to any users and at the same time connected with user’s home authentication services when available.
- Sympa includes its own traditional authentication service with account creation and password remind.

Multiple authentication methods

- Support multiple authentication services **at the same time**.
- Choose the appropriate authentication server depending on the user email domain if possible
- Provide a coherent authentication for email, http and soap interface.

Authentication method	Interface
Sender confirmation challenge	mail
Password (allocation by email)	web
LDAP AuthN backend	web
SSO: CAS Shibboleth	Web & SOAP Web
User certificate	Mail: S/MIME Web: HTTPS

LDAP authentication

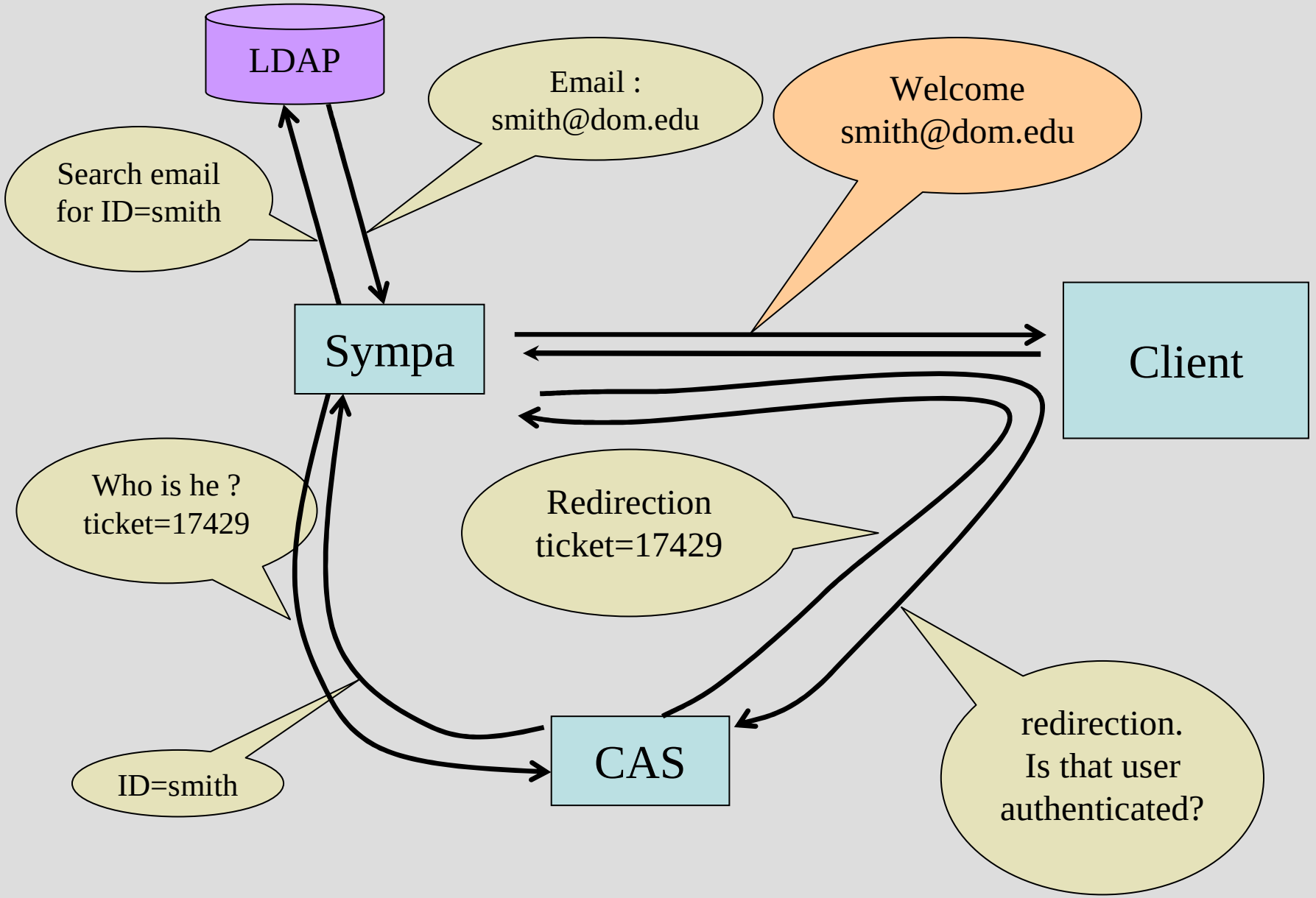
- If a LDAP authentication service is selected

1. Bind anonymously in order to fetch user_id from user id (only if email was provided)
2. Bind with user_id and password to check authentication
3. Bind anonymously to fetch email from user_id (only if user_id was provided)

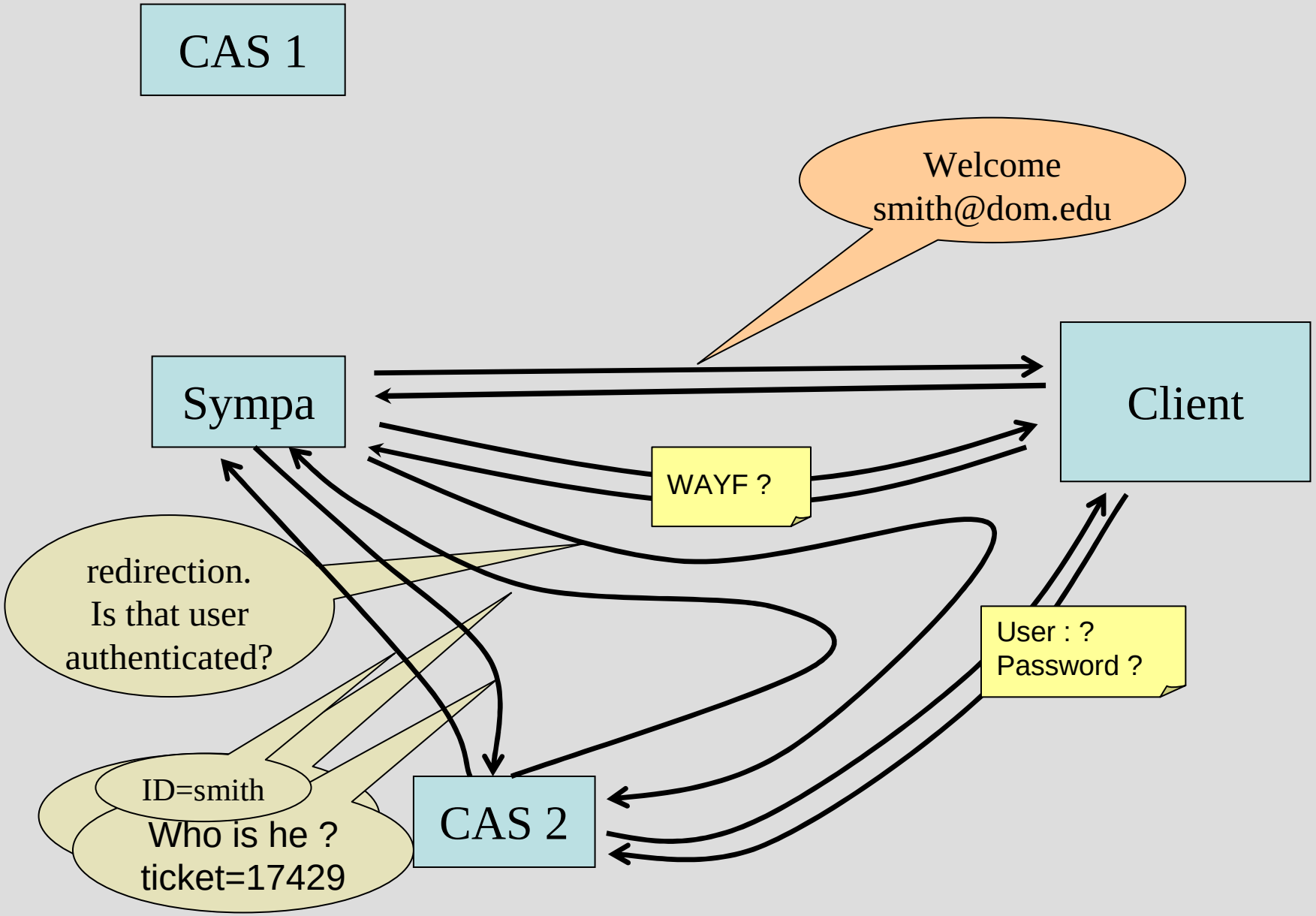
Central Authentication Service

- Yale university web **Single Sign On**
- French academic usage of CAS growing
- Use cookie, **redirections** and a ticket that need to be validated against CAS server
- Support **proxy** credential : needed for Uportal Sympa's channel.
- Not so easy to introduce into Sympa because CAS has not been designed to interoperate with any other authentication system.

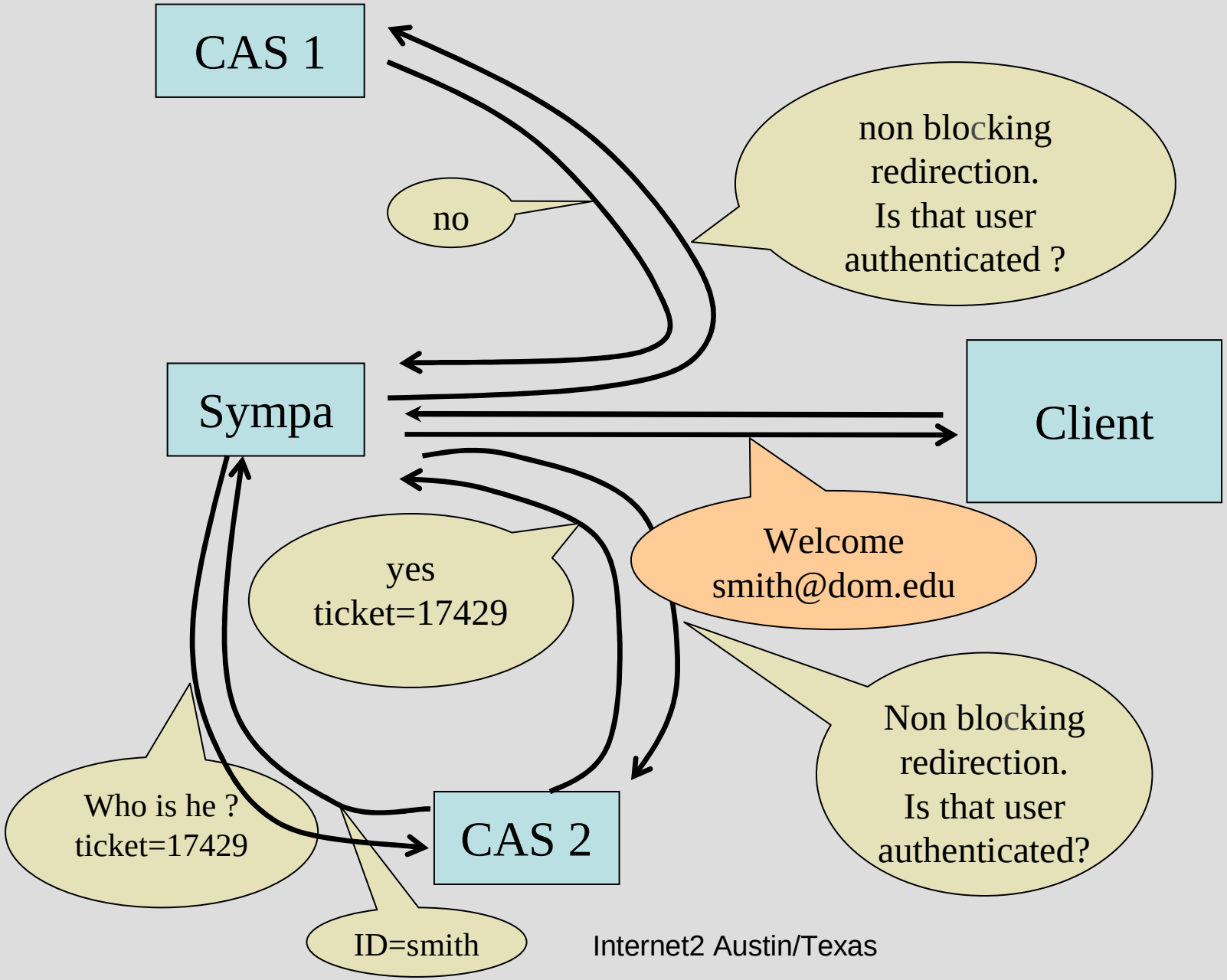
Sympa interaction with one CAS server



Interaction with a chosen CAS server



Interaction with multiple CAS servers

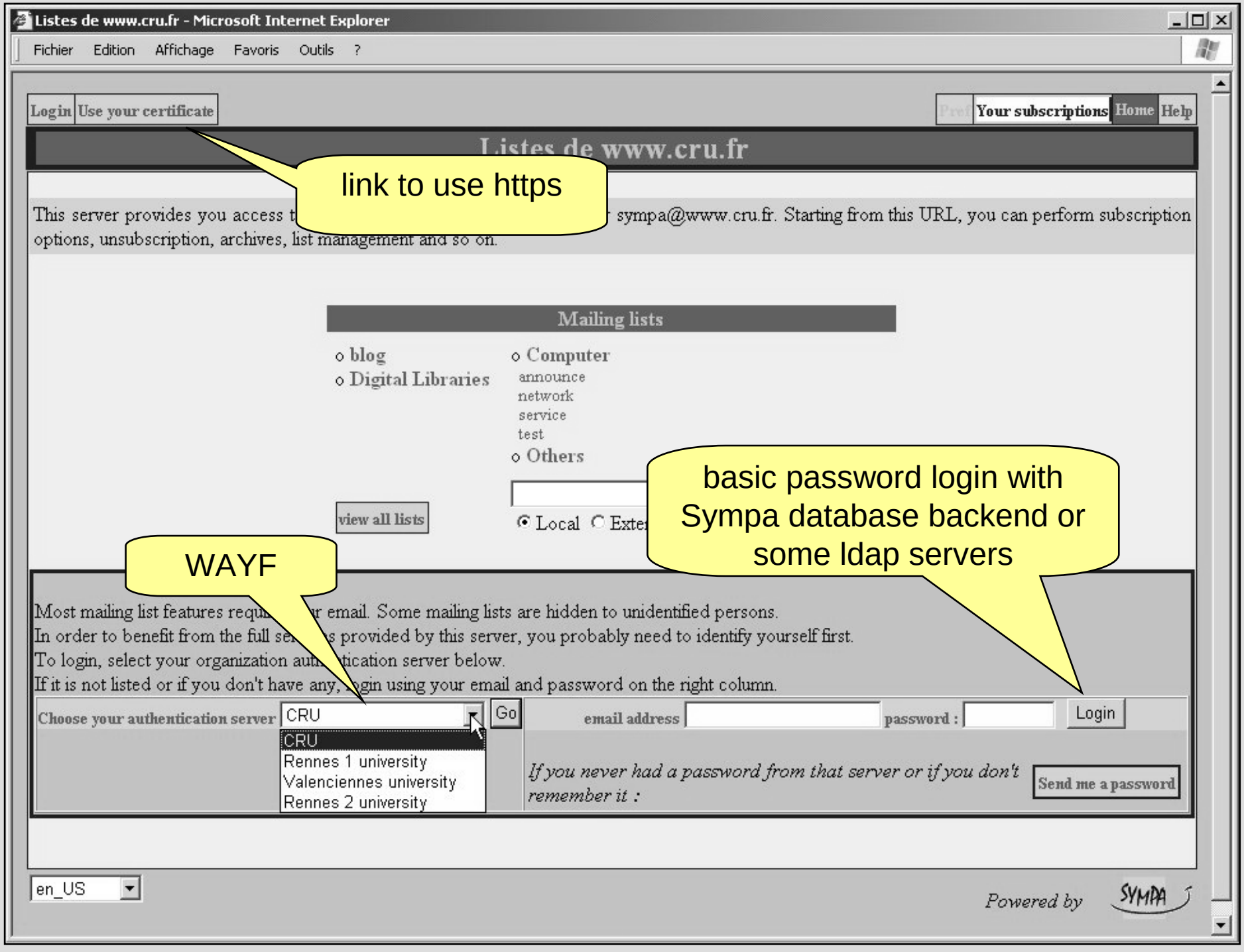


What happens if one CAS server is out of order ?

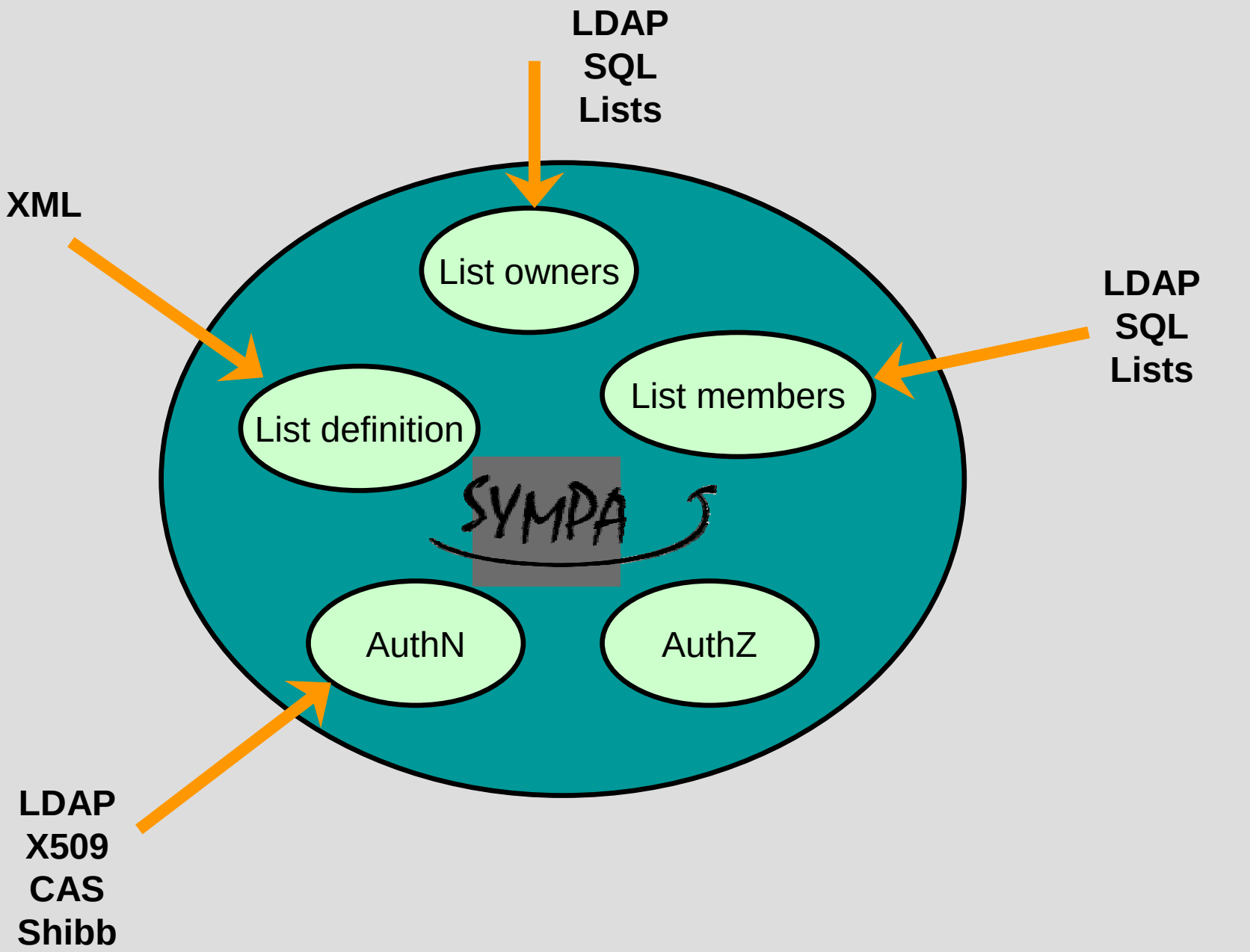
- Any redirection is a dead end
- Choose by configuration for each CAS server if non blocking redirection is enabled
- Ping all CAS servers periodically to detect servers down (todo ?)

What about “CAS logout” ?

- Sympa stores the authentication method used in order to propose appropriate logout button
- Sympa erases its own session cookie and redirects the user to the CAS logout URL
- CAS has some insufficiencies about logout: there is no central logout service

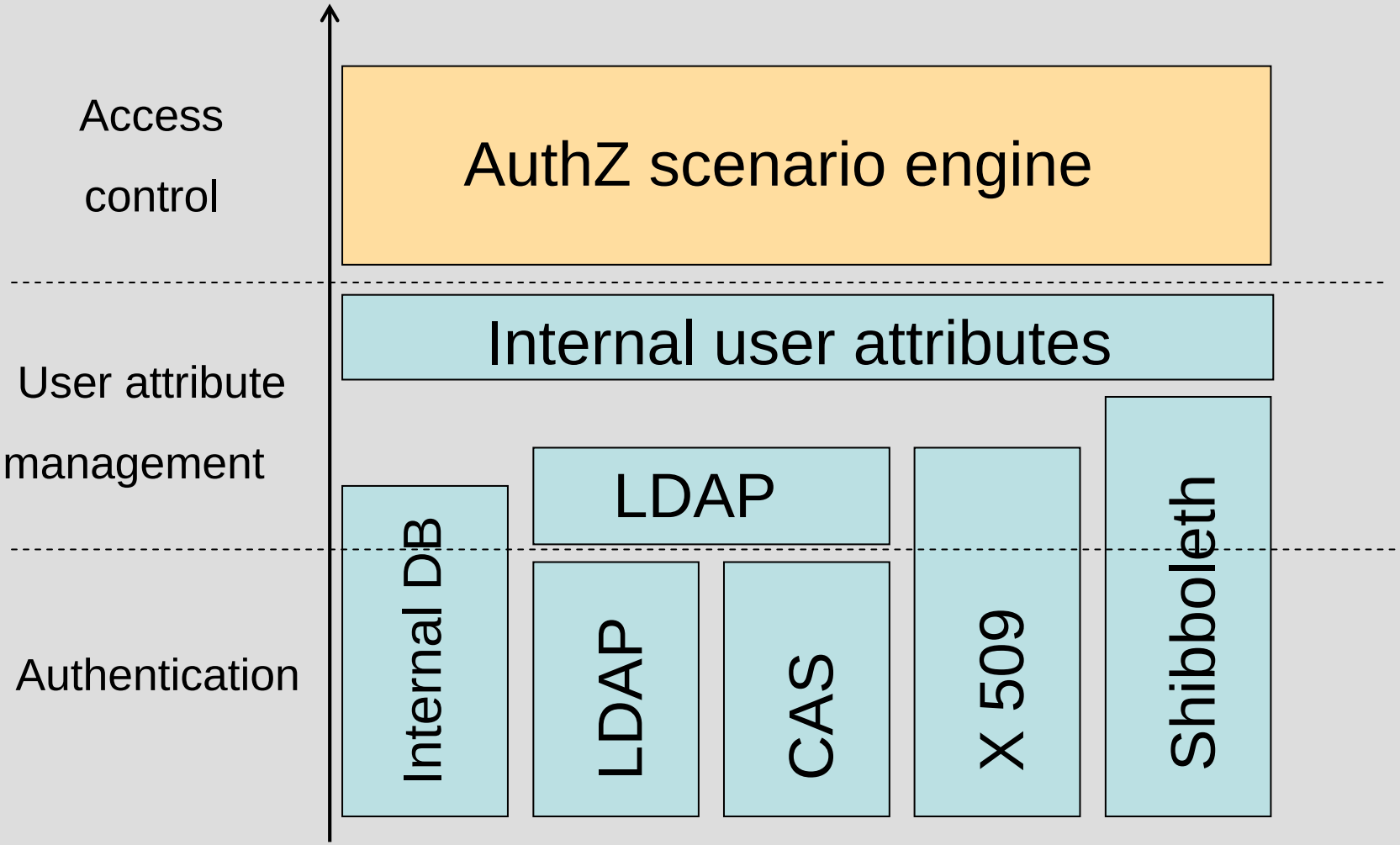

Listes de www.cru.fr - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Login
Use your certificate
Your subscriptions
Home
Help
Listes de www.cru.fr
link to use https
This server provides you access t
sympa@www.cru.fr. Starting from this URL, you can perform subscription options, unsubscription, archives, list management and so on.
Mailing lists
blog
Digital Libraries
Computer
announce
network
service
test
Others
view all lists
Local
basic password login with Sympa database backend or some ldap servers
WAYF
Most mailing list features requ
r email. Some mailing lists are hidden to unidentified persons.
In order to benefit from the full se
s provided by this server, you probably need to identify yourself first.
To login, select your organization aut
tication server below.
If it is not listed or if you don't have any,
gin using your email and password on the right column.
Choose your authentication server
CRU
Go
CRU
Rennes 1 university
Valenciennes university
Rennes 2 university
email address
password :
Login
If you never had a password from that server or if you don't remember it :
Send me a password
en_US
Powered by
SYMPA

```
cas
 base_url                    https://cas.cru.fr:443
 on_blocking_redirection     off
 login_path                  /login
 service_validate_path       /validate
 logout_path                 /logout
 auth_service_name           CRU
 ldap_host                   ldap.cru.fr:392
 ldap_get_email_by_uid_filter(&(uid=[uid])(objectClass=eduPerson))
 ldap_timeout                10
 ldap_suffix                 dc=cru,dc=fr
 ldap_scope                  sub
 ldap_email_attribute        mail

ldap
 host    ldap1.univ-nancy2.fr:392,ldap2.univ-nancy2.fr:392
 timeout                     20
 suffix                      dc=univ-nancy2,dc=fr
 get_dn_by_uid_filter        (uid=[sender])
 get_dn_by_email    (|(mail=[sender])(aliasmail=[sender]))
 alternative_email_attribute  maildrop
 email_attribute              mail
 scope                        sub

user_table
        negative_regexp     (univ\-nancy2)\.fr
```


LDAP
SQL
Lists
XML
List owners
LDAP
SQL
Lists
List members
List definition
SYMPA
AuthN
AuthZ
LDAP
X509
CAS
Shibb

CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
List families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

Authentication/Authorization



Access control in Sympa

- Rule-based access control applied to :
  - List services (subscribe, send, review, visibility,…)
  - Portal services (list creation, topics visibility)
- Separated from the authentication process (can also be applied to anonymous access)
- Configured for each list with defaults
- Authorization is applied the same way on all 3 interfaces (mail, web, soap)
- Extensible behavior using authorization scenarios (distributed with a set of 100)

Authorization scenarios

- Sympa's native ACL separated from the code
- A scenario is evaluated to provide/deny access to a service of Sympa
  - make the web interface highly adapted to the user's profile (inaccessible features are not advertised)
- A scenario is made of ordered rules
- A rule is made of :
  - A condition
  - An authentication method
  - An action (decision)
- The scenario title describes the behavior (useful for list configuration on the web admin interface)

Sample authorization scenarios

- Context :
  - Message distribution
- Expected behavior :
  - Private mailing list

```
is_editor([list->name],[sender])          smtp,smime      -> do_it
is_subscriber([list->name],[sender])      smtp,smime      -> do_it
true()                                    smtp,smime      -> reject
```

Sample authorization scenarios

- Context :
  - Message distribution
- Expected behavior :
  - Private mailing list
  - Confirmation for non subscribers

```
is_editor([list->name],[sender])          smtp,smime      -> do_it
is_subscriber([list->name],[sender])      smtp,smime      -> do_it
true()                                    smtp            -> request_auth
true()                                    md5,smime       -> do_it
```

Sample authorization scenarios

- Context :
  - Message distribution
- Expected behavior :
  - Private mailing list
  - Confirmation for non subscribers
  - Moderate multipart messages

```
is_editor([list->name],[sender])                  smtp,md5,smime        -> do_it
match([msg_header->Content-type],/multipart/) smtp,md5,smime -> editorkey
is_subscriber([list->name],[sender])              smtp,smime        -> do_it
true()                                            smtp              -> request_auth
true()                                            md5,smime         -> do_it
```

Sample authorization scenarios

- Context :
  - View web archives
- Expected behavior :
  - Grant access from the intranet
  - Grant access to authenticated users with local email addresses
  - List members access from anywhere

```
is_editor([list->name],[sender])              smtp,md5,smime      -> do_it
is_subscriber([list->name],[sender])          smtp,md5,smime      -> do_it
match ([remote_host],/utexas\.edu$/)     md5,smime  -> do_it
match([sender],/utexas\.edu$/)           md5,smime    -> do_it
true()                                   md5,smime      -> reject
```

Scenario rules syntax (1)

- Conditions :
  - is_subscriber(),is_owner(),is_editor(),is_listmaster()
  - equal(), match(), search()
  - true()
- Variables :
  - [sender], [user->*attr*], [subscriber->*attr*], [user_attributes->*attr*]
  - [list->*param*], [conf->*param*]
  - [remote_host], [remote_addr], [env->*var*]
  - [msg_header->*field*], [msg_body],[msg_part->type], [msg_part->body], [msg_encrypted], [is_bcc] |

Scenario rules syntax (2)

- Authentication methods :
  - smtp
  - md5
  - smime
  - pgp (soon)
- Actions :
  - do_it [,notify | quiet], reject(<tpl_name>)
  - request_auth, owner, editor, editorkey, listmaster

Scenario protected services

- Sending a message
- Subscribing
- Unsubscribing
- Adding a member
- Removing a member
- Access to a document
- Editing a document
- Review list members
- View web archives
- List visibility
- View list info
- List creation
- Topics visibility
- …

An LDAP-based authorization scenario

- Context :
  - Subscribe privilege in the `feminist-l` students ML
- Expected behavior :
  - Restrict subscription based on the EduPersonGender LDAP attribute

```
filter ('female_students.ldap',,[sender])      smtp,md5,smime      -> do_it
true()                                         smtp,md5,smime      -> owner
```

```
# female_students.ldap
host      ldap.utexas.edu:389, ldap2..utexas.edu:389
suffix    dc=utexas,dc=edu
filter    &((EduPersonAffiliation = student)(mail = [sender])
          (EduPersonGender = female))
scope     sub
```

Shibboleth architecture

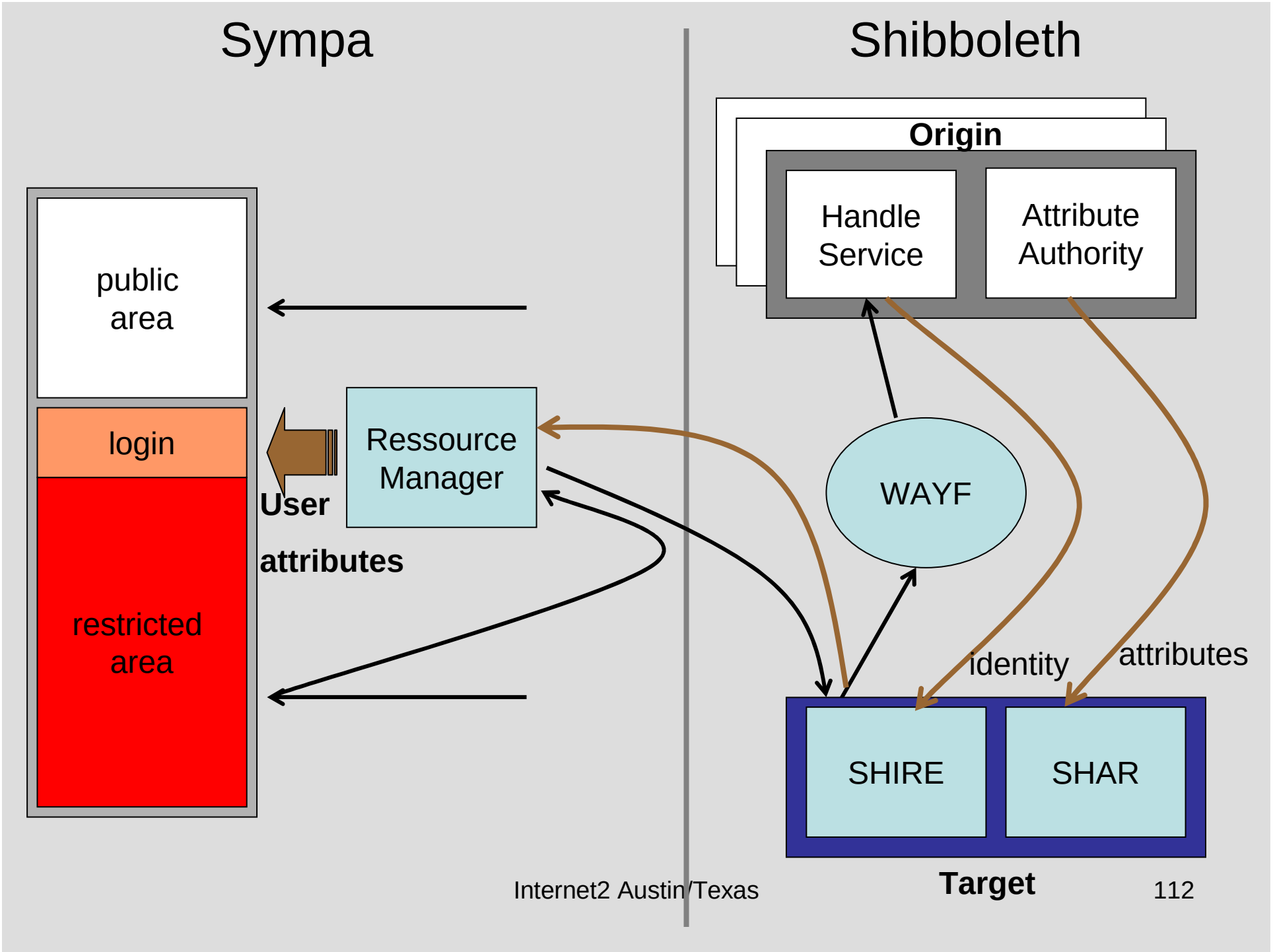
- Developped by Internet2
- Glue between local Single Sign-on servers to provide inter-institutional sharing of web ressources
- Shibboleth architecture made of 3 components :
  - Origin : installed in the user home organisation ; front-end to the local authN system and attributes database
  - Target : installed in front of a web ressource to control its access ; communicates with origin components
  - WAYF (Where Are You From) : the central component shared by a group of organization ; guides users to the origin component at their home org.

Shibboleth and Sympa Usage / Prerequisites

- Usage :
  - Building inter-institutional mailing lists with a strict definition of the targeted population
  - No additional user account on the ML server
- Prerequisites for each institutions:
  - Local SSO + Shibboleth « target » package
  - Common definition of user attributes semantic (study branches, staff categories,…)
  - Sympa server

Sympa and Shibboleth interactions

- Sympa's web interface delegates the authentication procedure to Shibboleth
- Sympa is a ressource protected by Shibboleth (« Target » package)
- Shibboleth provides user attributes to Sympa (email address required)
- Shib user attributes are used by Sympa to:
  - customize the web user interface
  - adapt user privileges


Sympa
Shibboleth
Origin
Handle Service
Attribute Authority
public area
login
restricted area
Ressource Manager
User
attributes
WAYF
identity
attributes
SHIRE
SHAR
Target

Configuration

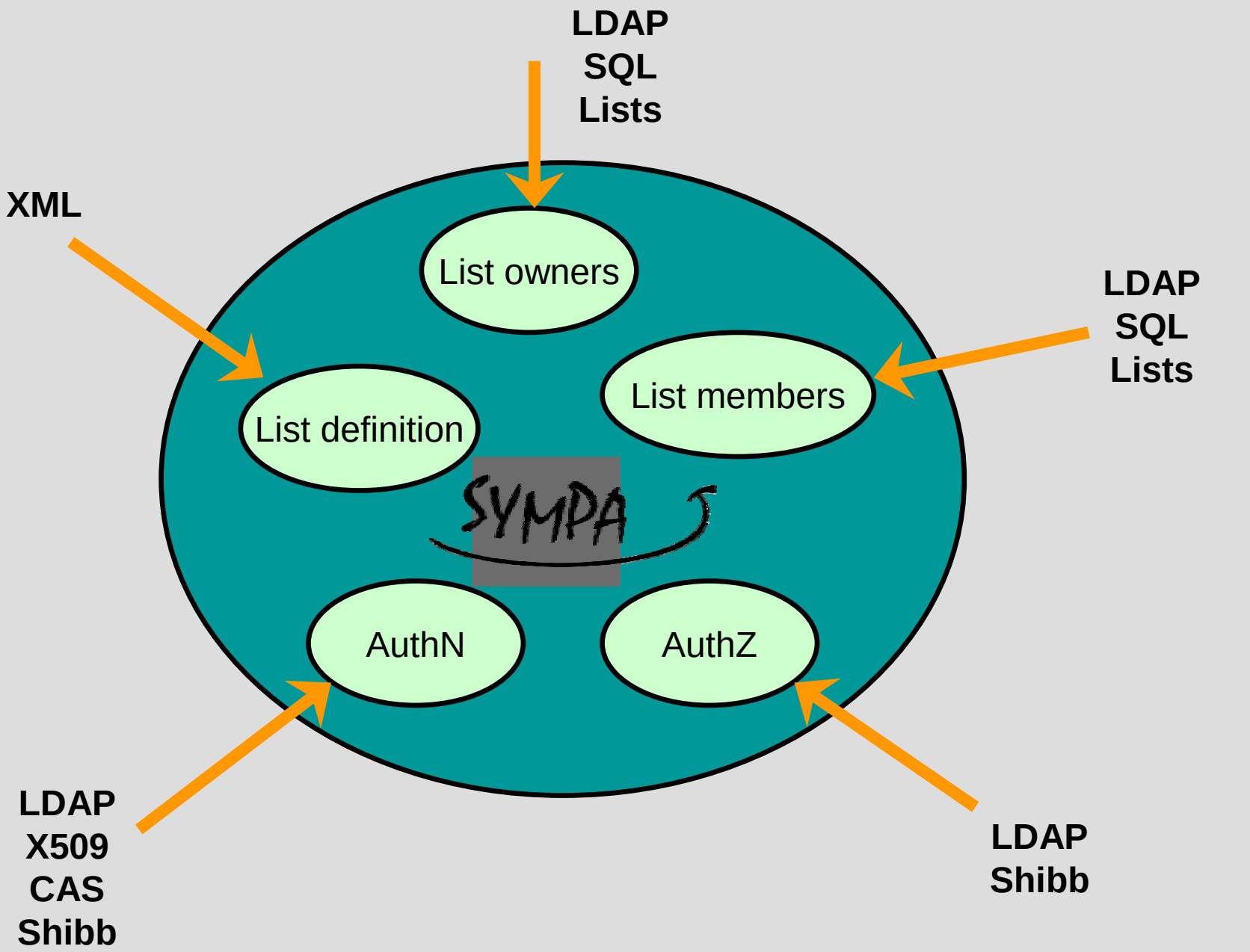
```
# Sympa configuration (auth.conf)
generic_sso
service_name  InQueue Federation
service_id          inqueue
http_header_prefix HTTP_SHIB
email_http_header HTTP_SHIB_EP_AFFILIATION
```

```
# Apache configuration
<Location /wws/sso_login/inqueue>
  AuthType shibboleth
  require affiliation ~ ^member@.+
</Location>
```

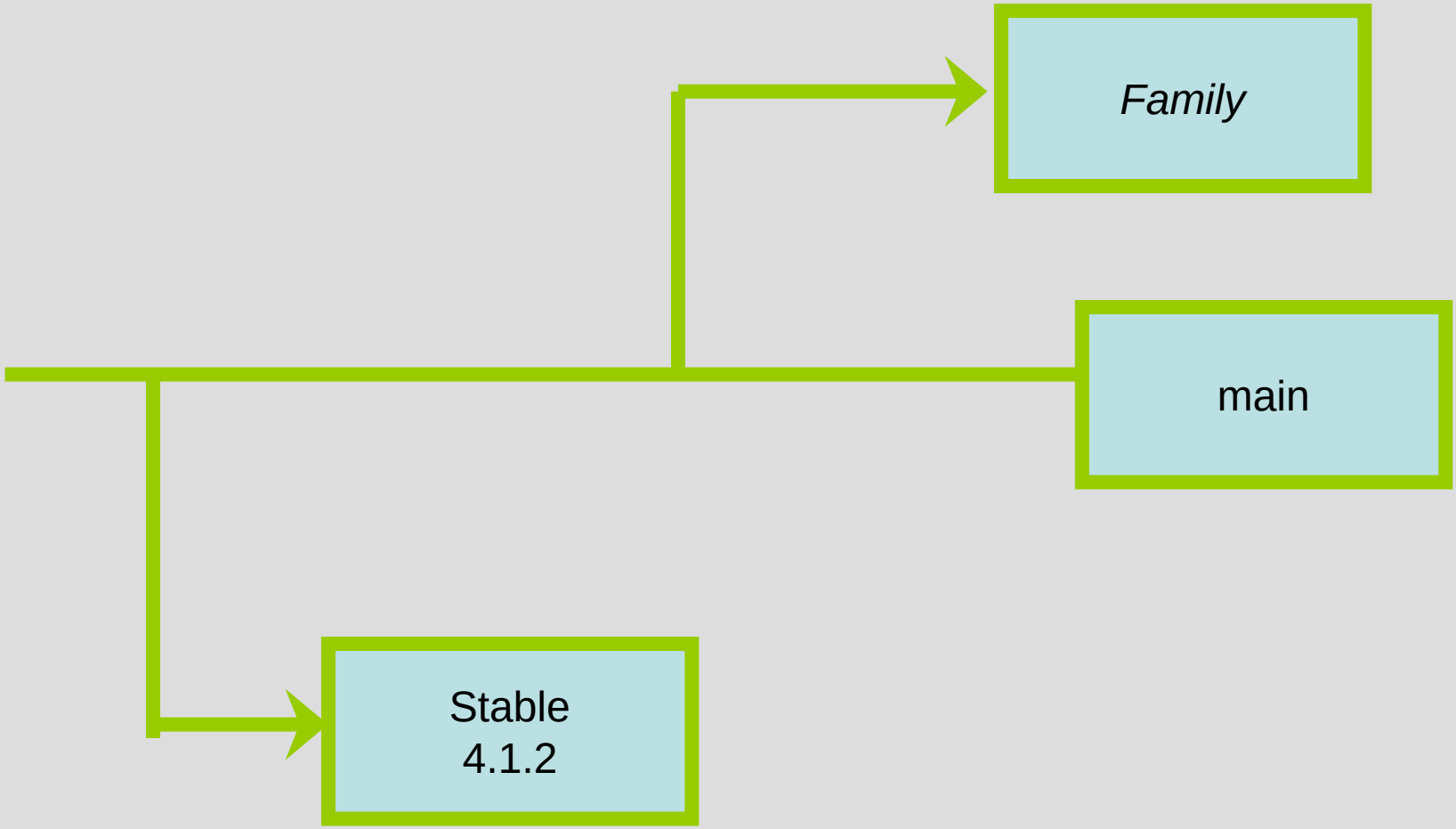
Access control based on Shibboleth user attributes

- Shibboleth user attributes :
  - Inherited via environment variables
  - Stored as session data in Sympa DB
  - Used in the authorization scenario engine
- Scenario sample rule:

```
# check if the user is a geology or archeology student
equal([user_attributes->SHIB_STUDY_BRANCH],'geology')      md5 -> do_it
equal([user_attributes->SHIB_STUDY_BRANCH],'archeology') md5 -> do_it
true()                    smtp,md5,smime   -> reject
```


LDAP
SQL
Lists
XML
List owners
LDAP
SQL
Lists
List members
List definition
SYMPA
AuthN
AuthZ
LDAP
X509
CAS
Shibb
LDAP
Shibb

Versions



CRU brief presentation, why Sympa
Sympa overview
The web document repository
Sympa organization, virtual robots
The template format
The SOAP interface
Dynamic mailing lists
List families
Privacy
S/Mime and Sympa
Authentication in Sympa
Access control management
Perspective

TODO : End user features

- By topic subscription option
- Basic survey/vote module
- « What’s new »
- Central alternate email management (alias)
- New look and feel (skins, CSS, javascript)
- OPT-IN traceability

TODO : More middleware features

- RSS
  - New message, new document, latest list
- Improve SOAP features, (need SAML)
  - All services (except those for list owner) via soap
  - «global my subscription» (require ML server federation)
- A user attribute management layer (auth.conf)
- PGP support (contrib)
- add support for various SSO (on demand)
- POP3 AuthN
- SPF: check SPF headers in scenario
- Automatic list of owners

TODO : miscellaneous

- VERP optimized
- New bulk e-mailer tunning (email that has already generated errors should not be mailed with the same strategy)
- Multithreading to deal with huge spools
- Message distribution recovery
- Statistics
- On line editor of authorization scenario
- Authorization scenario should return a reason

TODO : better quality packaging

- TT2 release
- Fix our poor english in documentation and interface
- Online tutorial
- Internals documentation
- rpm (RH/Fedora) and debian

Project management

- cru.fr and recherche.gouv.fr needs : higher priority
- We need feed back
- We need help to define other priorities
- You are welcome to discuss the design
- Most contrib are integrated in release, consult Sympa authors first is better
- We need beta tester
- Collaborative and professional support

Question ?