

# Plan

- Présentation du logiciel Sympa
- Architecture
- La gestion des hôtes virtuels
- Listes avec inclusion des abonnés
- Les modules d'authentification
  
- **Les Scénarios d'autorisation**
- Le format TT2
- Les familles de listes
- Retour d'expérience de la gendarmerie, listes automatiques
- Le serveur SOAP
- Migration
- Conclusions

# Les scénarios d'autorisation

# Contexte

The screenshot shows the CRU Universalistes web interface. At the top, there is a navigation bar with the CRU logo and the title 'Universalistes'. Below the logo, there are several menu items: 'Création de liste', 'Admin Sympa', 'Liste des listes', 'Accueil', and 'Aide'. The main content area is divided into several sections:

- User Information:** A box on the left shows the user 'david.verdin@cru.fr' as the 'listmaster' with a 'Logout' button.
- Vos préférences:** A section for user preferences.
- Abonnés : 85 (Taux d'erreurs : 0 %):** A section showing the number of subscribers and error rate.
- Propriétaires:** A list of owners: David Verdin, Olivier Salaün, and Serge Aumont, with a 'Contacter le propriétaire' button.
- Options d'abonné:** A section for subscriber options.
- Désabonnement:** A section for unsubscribing.
- Info:** A section for general information.
- Admin:** A section for administrative tasks, with 'Configurer la liste' highlighted. Sub-items include 'Définition de la liste', 'Diffusion/Réception', 'Droits d'accès', 'Les archives', and 'Gestion des'.

The main configuration area is titled 'inscription-formation-sympa@cru.fr' and 'Inscription à la formation Sympa du 7 juin 2007'. It contains three configuration sections, each with a 'code source du scénario' and 'Aide' button:

- Qui a accès aux informations sur la liste (Info) (default):** Set to 'pour tous (default)'.
- Qui peut s'abonner (subscribe):** Set to 'ouvert à tous, notification du propriétaire (open\_notify)'.
- Qui peut ajouter des abonnés (add) (default):** Set to 'ajout par le propriétaire sans authentification (default)'.

At the bottom left, there is a 'Terminé' icon. At the bottom right, there is a 'Site sûr !' icon.

# Contexte

The screenshot shows the CRU Universalistes web interface. At the top, there is a navigation bar with the CRU logo and the title "Universalistes". Below the navigation bar are several menu items: "Création de liste", "Admin Sympa", "Liste des listes", "Accueil", and "Aide".

On the left side, there is a user profile box for "david.verdin@cru.fr" with the role "listmaster" and a "Logout" button. Below this is a section titled "Vos préférences".

Below the preferences, there is a box showing statistics: "Abonnés : 85 (Taux d'erreurs : 0 %)" and "Propriétaires" listing David Verdin, Olivier Salaün, and Serge Aumont. There is a "Contacter le propriétaire" button.

Below that, there is a section for "Options d'abonné" and "Désabonnement". Under "Info", the "Admin" option is highlighted, showing a list of administrative tasks: "Configurer la liste", "Personnaliser", "Gérer les abonnés", "Liste noire", "Gérer les archives", "Gestion des erreurs", and "Journaux".

The main content area displays the configuration for the "inscription-formation-sympa@cru.fr" list, with the subtitle "Inscription à la formation Sympa du 7 Juin 2007". Below this, there is a section titled "info default" with the path "(chemin : /home/sympa/bin/etc/scenari/info.default)".

A yellow box contains the following code snippet:

```
title.gettext for anyone
true() smtp,md5,smime -> do_it
```

Below the code, there is a form field for "nom du scénario" with the value "default" and a "Sauvegarder" button.

At the bottom left, there is a "Terminé" icon and text. At the bottom right, there is a "Site sûr !" icon and text.

Listmaster seul !

# Principe

- Les scénarios déterminent les *droits* à réaliser une action sur une *liste*.
- Ce sont des fichiers localisés dans le répertoire **scenari/**.
- La précedence s'applique :
  - distribution (~sympa/bin/etc/scenari/)
  - installation (~sympa/etc/scenari/)
  - robot (~sympa/etc/dom1.fr/scenari/)
  - liste (~sympa/expl/<dom1.fr/>scenari/)

# Configuration

- Fichier : nom\_action.scenario

Ex : info.default

- Info  $\langle = \rangle$  « qui peut lire le infos ? »
- Default  $\langle = \rangle$  « le scénario par défaut »
- Paramètre de configuration

Dans le fichier « config » de liste :

**info default**

- Les scénarios ne sont pas rechargés dynamiquement.

Mais les fichiers **config**, si.

- Les config.bin contiennent le scénario.

# Quelles actions ?

- **access\_web\_archive** : Qui peut accéder aux archives web ?
- **add** : Qui peut ajouter des abonnés ?
- **automatic\_list\_creation** : Qui peut créer une liste automatique ?
- **create\_list\*** : Qui peut créer une liste ?
- **d\_edit** : Qui peut éditer un document partagé ?
- **d\_read** : Qui peut lire les documents partagés ?
- **del** : Qui peut supprimer des abonnés ?
- **global\_remind\*** : Qui peut lancer un rappel de tous les abonnements ?
- **info** : Qui peut consulter la page de présentation de la liste ?
- **invite** : Qui peut inviter un individu à s'abonner à la liste ?
- **remind** : Qui peut déclencher un processus de rappel des abonnements ?
- **review** : Qui a accès à la liste des abonnés ?
- **send** : Qui peut poster des messages à la liste ?
- **subscribe** : Qui peut s'abonner ?
- **topics\_visibility\*** : Qui peut voir le thème ?
- **unsubscribe** : Qui peut se désabonner ?
- **visibility** : Qui peut voir la liste ?

\* Exceptions : ne concernent pas une liste.

# Structure

- title.gettext <Chaîne>  
Titre internationalisable
- Suite de règles
  - Règle 1
  - Règle 2
  - ...

=> *La première vérifiée est appliquée.*



# Syntaxe des règles

`<condition> <méthode_identification> -> <action_engendrée>`

Condition de  
vérification de la règle

Méthode employée pour  
identifier le demandeur

Attitude adoptée par le serveur  
si la condition est vérifiée

# Exemple de scénario

```
is_subscriber([list->name],[sender])  
is_subscriber('sympa-user@cru.fr',[sender])  
is_editor([sender])  
true()  
reject(reason='remind_owner')
```

```
smtp  
smtp  
smtp  
smtp
```

```
do_it  
do_it  
do_it
```

# Méthodes d'authentification

- **smtp** : header « From: » d'un message
- **md5** : challenge (mail ou login web)
- **smime** : signature du message ou authentification web avec un certificat client

# Attitude du serveur

- **do\_it** : exécute l'action
- **reject** : refuse d'exécuter l'action.
  - (reason = " clé " ) : réponse-type
  - (tt2 = "fichier.tt2") : réponse personnalisée
- **request\_auth** : demande authentification
- **owner, editor, listmaster** : demande confirmation à la personne indiquée

Option **quiet** ou **notify**, suivant l'attitude.

# Conditions

- **true()** : toujours vrai :). Utiliser pour définir le comportement par défaut
- **equal()**, **match()** : test de variables
- **search()** : vérifie les conditions d'un filtre
- **is\_owner()**, **is\_editor()**,  
**is\_subscriber()**, **is\_listmaster()**... :  
teste le rôle d'un utilisateur
- **older()**, **newer()** : compare des dates
- **CustomCondition::<package\_name>( ... )** :  
condition fait-maison !

# Variables

- Forme :  
  [ **nom\_var** ] OU [ **groupe->nom\_var** ]
- Quelles variables ?
  - Chaîne simple ('sympa-user@cru.fr')
  - Liste définie
  - custom\_vars : peuvent être définie au niveau d'une liste
    - Exemple : [custom\_vars->discipline]  
  « discipline = biologie »
  - filtres

# Filtres

- Permettent l'exploitation de filtres élaborés contenant uniquement les variables [**sender**] ou [**listname**] : vérifient l'appartenance d'un individu à un groupe
- Filtres = fichiers contenant la requête (dans **etc/search\_filters**)
  - LDAP (\*.ldap)
  - SQL (\*.sql)

# Exemple de filtre SQL

Fichier `~sympa/etc/search_filters/profsmath.sql`

Sélectionne tous les enseignants d'informatique de l'université Rennes 1 :

`sql_named_filter_query`

`db_type`                    `mysql`

`db_name`                    `people`

`db_host`                    `dbserver.rennes1.fr`

`db_user`                    `sympa`

`db_passwd`                  `pw_sympa_mysqluser`

`statement`                  `SELECT count(*) as c FROM users`  
`WHERE mail=[sender] AND`  
`EmployeeType='PROFESSOR' AND`  
`department='mathematics'`



# Exemple de règle utilisant ce filtre

Appliqué à la liste des étudiants en maths, seuls les étudiants abonnés peuvent effectuer l'action. Leurs enseignants, eux, le peuvent même sans être abonnés (de droit divin) :

```
is_subscriber([list->name],[sender]) smtp do_it
search('profsMath.mysql') smtp do_it
true() smtp reject
```

# Inclusion de scénarios

- Inclusion ponctuelle :

subscribe

*include commonreject*

match(/, /cru\.fr\$/) smtp,smime -> do\_it

true() smtp,smime -> owner

( include include.commonreject)

- Inclusion automatique

include.<action>.header : automatiquement incluse  
dans les scénarios.

# custom\_conditions

- Personnalisation de traitements
  - Package perl dans **etc/custom\_conditions**
  - nom : "quoi.pm" (minuscules !)  
=> condition " CustomCondition::quoi(...)"
  - contient une fonction statique "verify(...)"
- Renvoie 1 : autorise
- Renvoie undef : erreur
- Renvoie autre chose : interdit

# Conclusion

- Scénarios : contexte de liste
- Hautement personnalisables :
  - filtres
  - custom\_conditions
  - custom\_vars
- Envoyez-nous les vôtres !