Spécifications de Dyna2.

Auteurs	Pierre-Olivier TERRISSE Pierre-Olivier.Terrisse@univ-nantes.fr Laurence GUITTET Laurence.Guittet@univ-nantes.fr Nicolas COSTES Nicolas.Costes@univ-nantes.fr

1. Contexte

Entièrement codé en langage Java, Dyna est le méta-annuaire de l'Université de Nantes. Il a pour but d'agréger des informations d'annuaire à partir de données externes, de les stocker dans un référentiel unique, de permettre leur consultation et leur modification par des utilisateurs authentifiés, et enfin de les propager dans des annuaires externes qui sont généralement de type LDAP. Techniquement, Dyna s'exécute dans un moteur de servlets, Tomcat, lui-même relié au serveur web Apache. La base de données référentielle est un moteur MySQL. Afin de comprendre de manière plus approfondie le fonctionnement de Dyna 1, se reporter au document “introduction à Dyna”. Le projet, démarré fin 2001, avait pour but de mettre en place un annuaire LDAP. A l'origine, il avait simplement pour but d'agréger les informations issus de la base de données de gestion du Personnel, Harpege, afin de les stocker dans un référentiel consultable par le web, et de les propager dans un annuaire LDAP de contact. Puis, des besoins liées à l'authentification sont apparus. Dyna a permis de gérer des mots de passe saisis de manière sécurisée par HTTPS. Parallèlement, il a fallu changer de source de données pour passer sur KSUP, base de données relationnelle permettant d'animer le site institutionnel de l'Université. Dans ce cadre, Dyna a été développé afin d'abstraire la source de données et de définir un format d'échange XML, la DTD Dyna. Ensuite, le projet Gromel a rendu nécessaire la gestion des attributs spécifiques à la messagerie. Les comptes des utilisateurs sont devenus modulaires, avec des dossiers optionnels. En aval, les réplicateurs se sont multipliés :

- Annuaire LDAP au format SUPANN du CRU (supann.univ-nantes.fr) ;
- Annuaires LDAP de POLYTECH'Nantes identifiant étudiants et personnels ;
- Annuaire de contact sans mot de passe en plus de l'annuaire d'authentification (ldap.univ-nantes.fr).

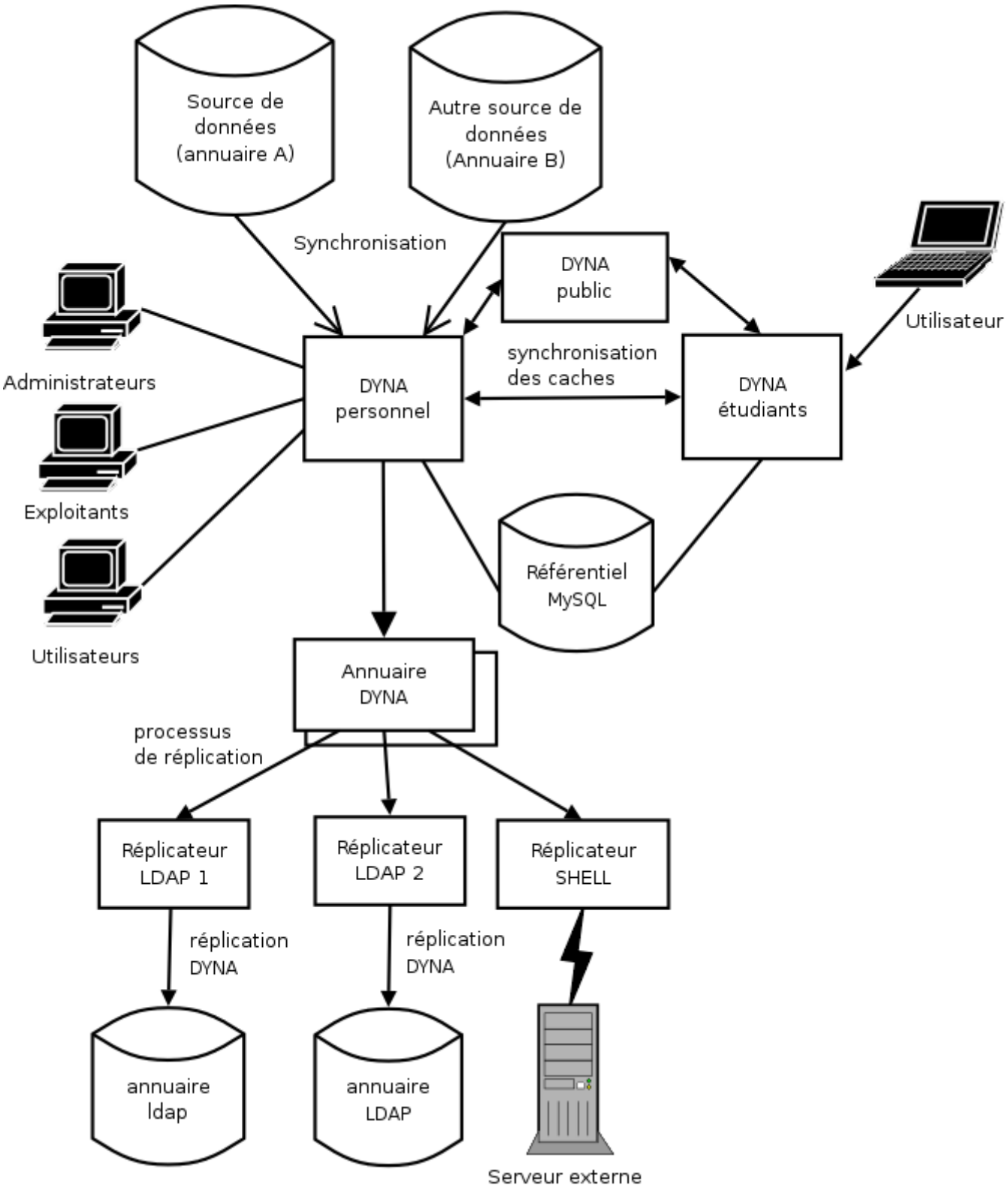
En 2005, les sources de données se sont diversifiées afin de répondre à des projets nouveaux :

- Gromel des étudiants ;
- Authentification et synchronisation d'annuaire avec KSUP.

Les projets en cours amèneront de nouvelles fonctionnalités :

- Espace de stockage ;
- Projet lié à UVPL (service d'identification des étudiants avec leur profils, et d'authentification) ;
- Multiplication des annuaires LDAP de composantes et d'organismes extérieurs tels que le CNAM.

Dyna gère à présent plus de 400 groupes liés à la messagerie et environ 43000 comptes de personnes (personnels, étudiants). Son code source contient 33700 lignes de java directement liées au méta-annuaire et 32200 lignes codant les services génériques (cache, pool de connexions, etc), soit un total de près de 66000 lignes. Le schéma suivant représente Dyna tel qu'il est paramétré actuellement.



2. Limites de l'architecture actuelle

2.1 La présentation et la logique applicative s'exécutent dans la même JVM

Les classes de Dyna se répartissent en deux parties :

- les classes “métier”, sérialisées dans la base de données : par exemple des personnes, des groupes, etc.
- les classes “requête web”, chargées de l'interaction avec les utilisateurs et produisant du code HTML. Sauf exception, elles n'interagissent pas directement avec la base de données.

La principale difficulté rencontrée avec Dyna est que ces deux types de classes s'exécutent dans la même machine virtuelle Java. Cette architecture présente l'avantage de la simplicité. Mais cela nécessite un nouveau déploiement pour toute modification dans les pages. De plus, la production de code HTML par des programmes en Java (les “requêtes web”) conduit rapidement à un code abondant (quoique facile à lire) ; il n'est pas possible d'utiliser un outil de type DreamWeaver pour modifier la présentation des pages. Dans ce domaine, il est possible d'améliorer la situation en utilisant un moteur de “templates” tel que Velocity. Mais chacune des quelque 260 pages devant être transformée et intégrée, le travail est considérable. Avec un moteur de templates, il serait possible de créer des “skins” de Dyna, afin de respecter une charte graphique pouvant évoluer dans le temps. Un autre inconvénient de cette architecture est de nécessiter le déploiement d'exactement autant de moteurs Dyna que l'on souhaite de serveurs web d'accès au service. Ainsi, à l'Université de Nantes, nous avons trois serveurs Dyna : pour le personnel, pour les étudiants, et pour les accès externes. Mais seulement un seul moteur Dyna sert réellement pour les synchronisations. Séparer le serveur web du moteur de méta-annuaire permettrait de clarifier et de sécuriser l'architecture, en garantissant l'unicité du moteur sans préjudice au déploiement de multiple serveurs web. Mais pour cela, il faut disposer d'une interface claire et stable entre les deux, et d'un protocole de communication.

2.2 La réplication est séquentielle

La réplication des entrées ajoutées, modifiée ou supprimée est assurée de manière asynchrone par un ordonnanceur qui traite les demandes de manière séquentielle et par ordre de priorité d'abord, puis par ordre d'arrivée. Ainsi, la réplication d'un annuaire entier peut durer des heures sans charger le serveur Dyna à plus de 5 pour cent. Un annuaire LDAP lent voire défaillant peut perturber tout le mécanisme de réplication. De plus, la réplication des entrées dépendantes (exemple : une personne change d'organisation qui est membre d'un groupe : il faut régénérer celui-ci) se fait dans la foulée ce qui peut créer des goulets d'étranglement. Afin d'améliorer les performances de la réplication, il faudrait : - paralléliser la réplication en gérant les dépendances et en distribuant la charge sur plusieurs tâches ; - gérer le réordonnancement des entrées dépendantes, de manière à fluidifier le mécanisme. - améliorer la tolérance aux annuaires LDAP défaillants, par exemple en stockant de manière temporaire une référence aux entrées à répliquer dans le référentiel. - améliorer la journalisation et mettre en place un mécanisme de remontées d'alarmes lorsqu'un annuaire LDAP à répliquer est défaillant.

2.3 La reconstruction dynamique des groupes pourrait être optimisée

Dans Dyna 1, lorsqu'une personne est ajoutée ou modifiée dans le référentiel, tout groupe contenant son organisation ou l'une de ses organisations est reconstruit dynamiquement. Cette opération peut être coûteuse dans le cas de grands groupes et elle est parfois inutile. On pourrait optimiser Dyna en

gérant l'historique de changement d'organisation de manière à ne recalculer le contenu des groupes qu'à bon escient.

2.4 La mise en place de nouveaux dossiers associés aux entrées est lourde

Dans Dyna, chaque personne peut optionnellement posséder un ou plusieurs dossiers additionnels tels que la messagerie, les permissions, l'import en provenance de bases de données externes. Les groupes peuvent également posséder un dossier de messagerie. Mais tout ceci s'est mis en place au fil de l'eau, en fonction des exigences des projets et une remise à plat des dossiers associés aux entrées serait utile de manière à clarifier et épurer le code, et à faciliter l'ajout de nouveaux types de dossiers. Il faudrait en particulier ajouter une couche d'abstraction permettant de traiter les dossiers associés aux entrées de manière générique.

2.5 Les classes de tests unitaires ne sont pas systématiquement développées

Chaque classe, chaque fonctionnalité, devraient posséder une classe de test unitaire JUNIT, de manière à vérifier le fonctionnement de l'ensemble des mécanismes le plus tôt possible. Malgré l'inconvénient d'une inflation importante de la taille du code, le gain en terme de fiabilité est considérable. Actuellement, il existe de nombreuses classes de test, mais la totalité des fonctionnalités n'est pas couverte, à cause du travail imposant que cela demande.

2.6 Il y a des branches mortes à supprimer

Les fonctionnalités suivantes ne servent plus ou sont trop peu utilisées dans Dyna1 pour que leur reprise dans Dyna2 soit justifiée : - les annuaires privés servaient de carnet d'adresse personnel dans NOCC, lequel a été démantelé au profit de Squirrelmail dans le projet Gromel ; - les dénominations permettaient de classifier les structures avec un niveau minimum et un niveau maximum. - il était possible d'associer des entrées à la fiche d'une personne, pour indiquer qu'un utilisateur avait plusieurs comptes. En pratique, ce n'est pas utilisé dans Dyna1.

3. Proposition d'une nouvelle architecture

3.1. Interfaces

Suite aux problèmes évoqués au §2.1, nous proposons une architecture quatre tiers :

1. client navigateur (comme actuellement),
2. serveur web applicatif : Servlet java mais on pourrait éventuellement utiliser un autre langage

comme PHP,
3. moteur de méta-annuaire codé en Java s'exécutant dans une JVM dédiée (éventuellement sur un serveur virtuel dédié),
4. base de donnée s'exécutant sur un serveur virtuel dédié (comme actuellement).

Les interfaces sont les suivantes :

1. entre 1 et 2 : HTTP/HTTPS, comme actuellement.
2. entre 2 et 3 : nouvelle interface ! Le choix est large : on pourrait utiliser RMI, les EJB, Corba, SOAP, ou bien XML-RPC.
3. entre 3 et 4 : Mysql sur l'API JDBC, comme actuellement. Voir éventuellement l'utilisation d'une sur-couche de sérialisation d'objets telle que Hibernate.

Le choix préconisé est XML-RPC, en utilisant le moteur fourni par Apache. Voir le site http://ws.apache.org/xmlrpc/ XML-RPC permet d'exposer sur le réseau les méthodes publiques d'un objet quelconque. Il s'appuie sur un moteur de servlets tel que Tomcat ou bien sur un serveur web dédié, au choix. Il est possible de sécuriser le service par un chiffrement TLS. Le données sont encapsulées dans du XML et s'échangent par HTTP, de manière totalement transparente, sans avoir à analyser du code XML. Le principal avantage de XML-RPC est sa simplicité ; son inconvénient est de ne pas être orienté-objets (on a simplement des appels de procédures sur un objet exporté) et de ne permettre l'échange que de types prédéfinis (entiers, booléens, chaînes de caractères...). Il faut donc spécifier une API stable et claire permettant l'échange entre les serveurs web et les autres bases d'une part, et le moteur Dyna 2 d'autre part.

Sécurisation des flux XML-RPC : il existe trois possibilités que l'on peut employer de manière indépendante :

- TLS (mais il n'est pas sûr qu'il soit possible de baser l'authentification sur un certificat avec cette API car l'implémentation du serveur HTTPS est spécifique : cette fonctionnalité est plutôt proposée dans le but de chiffrer les échanges en XML) ;
- mode “paranoïaque” : il s'agit en fait de la restriction de la connexion au serveur XML-RPC basée sur le nom ou l'adresse IP du client ;
- authentification par identifiant et mot de passe.

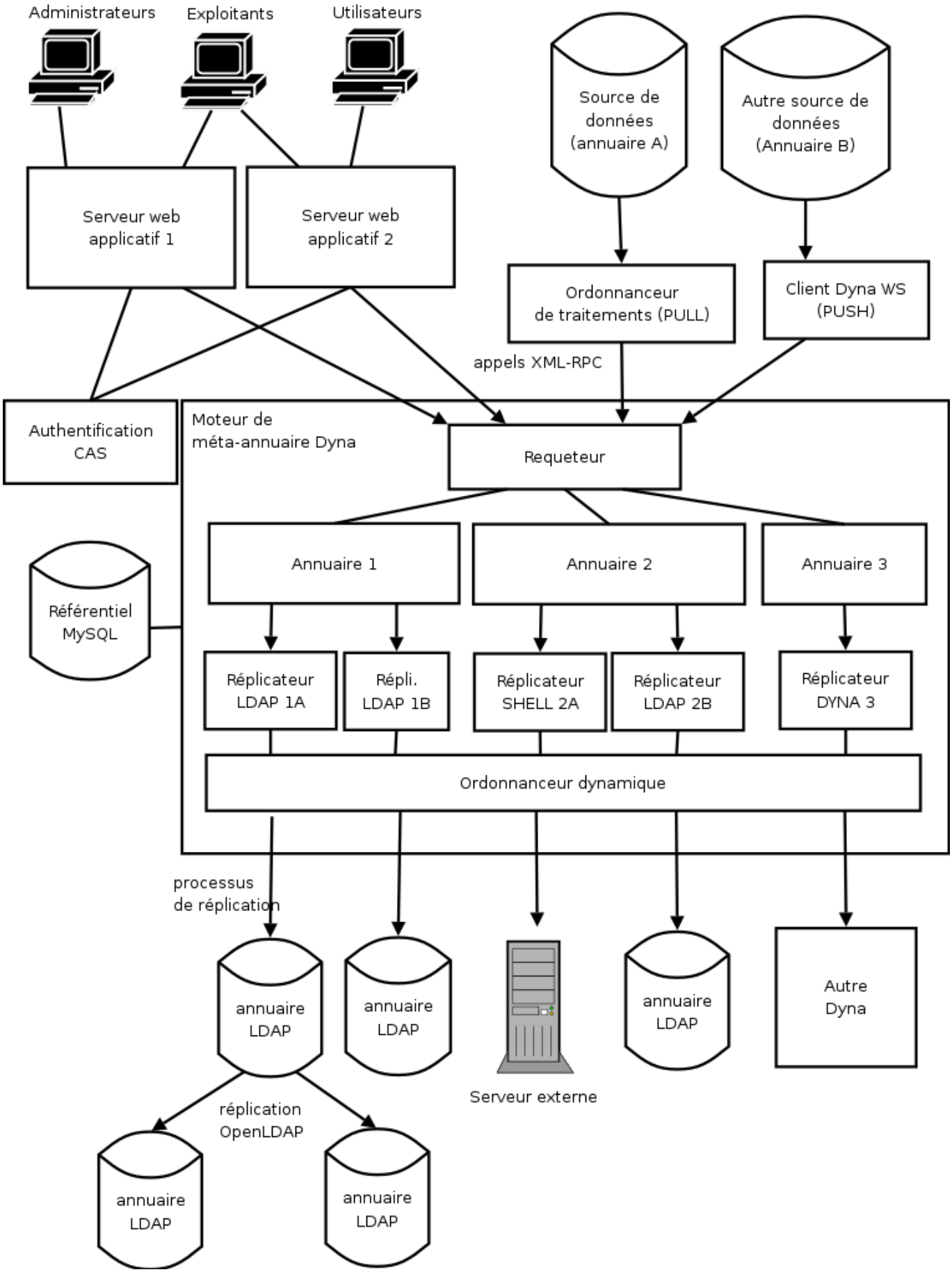
Ceci afin d'éviter qu'un client XML-RPC “pirate” puisse se faire passer pour un véritable client Dyna.

Choix d'une API LDAP. Comme le moteur Dyna est écrit en java, on a le choix entre les trois APIs suivantes :

1. LDAPJDK de Netscape. Cette API présente l'avantage d'être celle qui est la plus simple à utiliser. Elle est actuellement utilisée par Dyna1. Mais elle présente les inconvénients suivants :
    - problème d'encodage de caractère notemment avec les mots de passe, ce qui empêche de l'utiliser de manière exclusive ;
    - elle n'est plus maintenue depuis la dispartition de Netscape.
2. JLDAP de Novell. Cette API est un peu plus complexe que celle de Netscape, mais elle présente plus de fonctionnalités (LDAP asynchrone, manipulation d'entrées aux formats LDIF et DSML...) et gère très bien l'encodage de caractères accentués ;
3. JNDI de Sun. Cette API est multi-standard : elle gère aussi bien l'accès à des annuaires de type LDAP que DNS, NDS, Corba, etc. L'inconvénient est sa complexité, et la taille du code qui en résultera si on fait ce choix. La société Kosmos qui a fait le choix de cette API rencontre des problème d'encodage des mots de passe contenant des caractères accentués.

Le meilleur compromis complexité / fonctionnalités nous semble être l'API JLDAP de Novell. Il faut cependant assumer le coût de la migration.

3.2. Flux dans Dyna2


Administrateurs
Exploitants
Utilisateurs
Source de données (annuaire A)
Autre source de données (Annuaire B)
Serveur web applicatif 1
Serveur web applicatif 2
Ordonnanceur de traitements (PULL)
Client Dyna WS (PUSH)
appels XML-RPC
Authentification CAS
Moteur de méta-annuaire Dyna
Requeteur
Annuaire 1
Annuaire 2
Annuaire 3
Référentiel MySQL
Réplicateur LDAP 1A
Répli. LDAP 1B
Réplicateur SHELL 2A
Réplicateur LDAP 2B
Réplicateur DYNA 3
Ordonnanceur dynamique
processus de réplication
annuaire LDAP
annuaire LDAP
Serveur externe
annuaire LDAP
Autre Dyna
réplication OpenLDAP
annuaire LDAP
annuaire LDAP

3.3 Classes d'objets manipulés par Dyna

3.3.1. Un requêteur

Un requêteur est un ensemble d'objets exposés par XML-RPC. Il réceptionne les demandes des clients (interrogation et modifications). Il oriente les demandes vers les différents annuaires. Le requêteur contient un handler par catégorie d'actions que l'on peut effectuer sur Dyna.

3.3.2. Un annuaire

Un annuaire est un conteneur d'entrées. Il contient un ensemble d'entrées caractérisant une population homogène, par exemple les étudiants. Il possède des règles de gestion implémentées par des filtres. Un annuaire peut être en lecture seule, interdisant tout ajout, modification ou suppression de ses entrées. Un annuaire contenant des entrées ne peut pas être supprimé.

Il faut être administrateur de Dyna pour ajouter ou supprimer un annuaire, ou en modifier la configuration

3.3.3. Un filtre d'annuaire

Un filtre d'annuaire est un règle de gestion que l'on impose aux entrées appartenant à un annuaire. Un annuaire peut posséder un nombre quelconque de filtres, qui s'exécutent dans un ordre précis.

Chaque filtre possède des méthodes à implémenter :

- méthodes exécutées pour initialiser les entrées à leur création, ou avant leur modification ;
- méthodes exécutées afin de contrôler les entrées avant leur création ou leur modification. Ces méthodes peuvent interrompre le processus de mise à jour du référentiel et de propagation dans les réplicateurs si on détecte des valeurs inacceptables ;
- méthodes exécutées après la modification du référentiel (création, modification, suppression d'une entrée) mais avant la réplication.

Remarque : dans Dyna1, il n'y a qu'un seul filtre par annuaire. Autoriser le passage successif de plusieurs filtres permettra de rendre ceux-ci plus simples.
Dyna possède des filtres standards qui s'appliquent en principe à tous les annuaires.

Les filtres standards actuels ou à développer sont les suivants :

Nom du filtre	Description	Dans Dyna1
FiltreAnnuaireDefaut	Ce filtre effectue les opérations par défaut sur une entrée d'annuaire : - toute entrée a un nom - toute personne physique a un prénom - formatage du nom et du prénom : less noms s'écrivent en majuscules, et les prénoms en minuscules avec les initiales en majuscules ; dans les prénoms les signes diacritiques (accents, cédilles, etc.) sont respectés ; les majuscules accentuées ne sont pas autorisées - toute personne a un UID ; à défaut, celui-ci est automatiquement généré ; - cohérence des dates de départ et d'arrivée (on ne peut pas partir avant d'être arrivé) ; - validité des adresses de messagerie ; - état de l'entrée conforme avec les états autorisés par l'annuaire auquel elle appartient.	Oui. Ce filtre devra être décomposé en filtres plus simples, à raison d'un filtre par traitement.
FiltreAuth	Filtre garantissant que toute personne a un dossier d'authentification. Si la personne n'a pas de dossier d'authentification, celui-ci est crée avec la lettre du compte au format PDF qui l'accompagne.	Oui.

Il est possible de développer des filtres spécifiques et de les intégrer dans le moteur de Dyna. Exemple de filtre spécifique : tout étudiant doit dès sa création posséder un dossier de messagerie dont l'adresse principale est Prenom.Nom@domaine, le domaine étant un paramètre de l'annuaire des étudiants. C'est ce filtre qui gère la problématique des Prenom.Nom en doublon.

Les filtres spécifiques évitent d'écrire du code personnalisé dans les classes des entrées d'annuaire. Ils permettent d'isoler les fragments personnalisés du code, en implémentant la logique qui est spécifique à une population donnée.

Les filtres spécifiques actuels ou à développer sont les suivants :

Nom du filtre	Description	Dans Dyna1
FiltreAnnuPublicUnivNantes	Filtre utilisé pour l'annuaire du personnel de l'Université : - initialisation de l'adresse de messagerie de la personne ; - création éventuelle du dossier de messagerie, avec un dossier de messagerie, un serveur de messagerie en fonction de la première lettre du nom de la personne et un quota qui est fonction de sa catégorie. Initialement la BAL est désactivée ; - ajout des alias fonctionnels si la personne est responsable ou secrétaire d'une organisation ; - détermination de la nécessité ou non de faire signer par le responsable la lettre de validation du compte ; - vérification que la date de départ de la personne est bien indiquée si la personne appartient à certaines catégorie comme CDD UNIVERSITE, VACATAIRE, etc.	Oui. Filtre à décomposer en filtres simples.

Nom du filtre	Description	Dans Dyna1
FiltreAnnuaireEtudiantsUnivNantes	Filtre utilisé pour l'annuaire des étudiants de l'Université : - initialisation de l'adresse de messagerie de la personne ; - activation de la BAL ; - création éventuelle du dossier de messagerie, avec un dossier de messagerie, un serveur de messagerie en fonction de la première lettre du nom de la personne et un quota. - création d'alias automatiques en fonction de la composante auquel appartient l'étudiant (Polytech) ;	Oui. Filtre à décomposer en filtres simples.
FiltreMailPrioritaire	Ce filtre garantit que l'adresse de messagerie de la personne n'est attribué à personne d'autre. Il est utilisé par Géode via le client Dyna (car c'est Géode qui impose l'adresse de messagerie), pour les étudiants.	Oui.
FiltreEtatNormal	Ce filtre garantit que les personnes sont à l'état NORMAL. On utilise ce filtre pour les étudiants qui viennent de Géode via le client Dyna, afin de parer au problème des étudiants partis une année et revenus l'année suivante.	Oui.

3.3.4. Une entrée

Une entrée est un objet contenu dans un annuaire. Une entrée appartient à tout moment à un annuaire unique, mais celui-ci peut changer au cours du temps. Dyna doit à tout changement dans son référentiel vérifier l'appartenance de l'entrée concernée à un annuaire. Cette notion ne change pas par rapport à Dyna 1 : elle est le reflet de l'entrée gérée par un annuaire LDAP.
Une entrée possède :

- une fiche principale, contenant les informations obligatoirement présentes pour cette entrée. Par exemple, le nom pour une personne.
- optionnellement un ou plusieurs dossiers additionnels, permettant de stocker des informations supplémentaires que n'ont pas forcément les autres entrées. Par exemple, le dossier de messagerie qui contient les alias.

Une entrée est un objet qui peut éventuellement être répliquée par un réplicateur. Tous les réplicateurs ne savent pas répliquer toutes les entrées mais potentiellement toute entrée peut être répliquée.

Les entrées de Dyna se répartissent en trois catégories : les personnes, les structures et les groupes.

Une entrée se caractérise par un état, parmi les suivants :

- état normal (par défaut) ;
- état logiquement supprimé : l'entrée devient invisible en consultation normale ; elle est totalement inaccessible pour un utilisateur ne disposant pas de permission d'exploitant ou d'administrateur. Cet état marque le fait que l'entrée pourrait être supprimée de l'annuaire

dans certaines conditions ;
- état liste rouge : les conditions de visibilité sont les mêmes que celles de l'état logiquement supprimé, mais la sémantique est différente. Il n'est pas question de supprimer l'entrée, mais on ne souhaite pas que celle-ci soit visible. Il importe notamment d'éviter de répliquer les entrées en liste rouge dans les annuaires de contact, accessibles par tout un chacun ;
- état en attente de validation : cet état caractérise les entrées importées automatiquement et pour lesquelle il existe un doute sur leur validité. Par exemple, en cas de doublon sur le nom et le prénom. Les conditions de visibilité sont les mêmes que celles de l'état logiquement supprimé. Un exploitant doit décider si une entrée en attente de validation est valide, auquel cas il doit la passer à l'état normal.

Touts les annuaires n'autorisent pas nécessairement chacun de ces quatre états : ceci se trouve paramétré au niveau de l'annuaire lui-même.

Une entrée est un objet contrôlable : elle implémente l'interface DynaControlable car on soit savoir qui peut lire et modifier une entrée d'annuaire.

Une entrée peut de plus détenir une adresse de messagerie, et un dossier de messagerie : auquel cas, elle implémente l'interface EntreeMail.

3.3.5. Une personne

Une personne représente un compte individuel.

3.3.5.1. Personnes physiques, entrées génériques

Les personnes peuvent être des personnes physiques ou des entrées génériques, qui sont utilisés pour des comptes collectifs. Une entrée générique n'a pas de civilité ni de prénom : elle représente soit un compte de redirection, soit un compte collectif, par exemple pour une équipe de travail, une association, etc.

3.3.5.2. Identifiant unique des personnes

Outre son numéro unique qui l'identifie parmi les entrées de Dyna, elle possède un identifiant alphanumérique unique à travers tous les annuaires (uid). Cet identifiant est généralement la chaîne de caractères permettant à un utilisateur de d'identifier (login).

3.3.5.3. Attributs

Les personnes possèdent des attributs que l'on retrouve dans tous les annuaires, comme le nom et le numéro de téléphone. Ce sont les attributs obligatoires. Les personnes peuvent en outre posséder des attributs supplémentaires en fonction de leur annuaire d'appartenance : ce sont les attributs supplémentaires. Par exemple, pour les étudiants on gère les formations auxquelles ils sont inscrits, attribut que ne possèdent pas les personnels de l'Université qui en revanche appartiennent à une catégorie (titulaire, contractuel, retraité, etc.). Pour chaque attribut, obligatoire ou supplémentaire, on gère les paramètres suivants :

- nom court : utilisé pour les échanges en XML ;
- nom complet : utilisé pour l'affichage ;
- visibilité : indique si l'attribut est invisible (c'est à dire pas utilisé), visible seulement dans l'écran de détail de la personne ou bien visible dans le tableau de recherche des personnes et dans l'écran de détail ;
- confidentialité : indique si l'attibut peut être vu par tout le monde, ou bien seulement par les exploitant ayant le droit de modifier la fiche de la personne, ou bien encore par les administrateurs de Dyna seulement ;
- modification : indique si l'attribut peut être modifié par les exploitants et la personne elle-même, ou par les exploitants habilités seulement, ou bien encore par les administrateurs de Dyna seulement ;
- syntaxe : conditionne les valeurs autorisées ;
- valeurs_permises : ensemble des valeurs autorisées suivant la syntaxe (voir le tableau suivant) ;
- types_personnes : indique si l'attribut peut être détenu par les personnes physiques seulement, par les entrées génériques seulement, ou bien par les personnes de tous types ;
- ordre : ordre d'apparition de l'attribut dans le tableau ainsi que dans l'écran de détail d'une personne ;
- alignement : alignement de l'attribut dans le tableau : à gauche, centré ou à droite ;
- limitation_annuaires_supportés (booléen) : si vrai, alors cet attribut n'est supporté que par certains annuaires, si faux, il est supporté par tous les annuaires ;
- listes_annuaires_supportes : liste des annuaires supportés ;
- nom_ldap : nom de l'attribut dans un annuaire LDAP ;
- ldap_fige (booléen) : si vrai, alors on ne peut pas changer son nom_ldap.

3.3.5.3.1. Tableau des syntaxes des attributs

Le tableau suivant indique la liste des syntaxes qu'il est possible d'affecter à un attribut. Lors de toute modification de la fiche d'une personne, Dyna vérifiera que chaque attribut respecte bien sa syntaxe. Certaines syntaxes sont de plus pourvues de contraintes à faire respecter.

Syntaxe	**Signification**	**Contraintes**
Texte	texte libre	nombre minimum et maximum de caractères
Téléphone	numéro de téléphone	aucune
Mail	adresse de messagerie	aucune (on pourrait ajouter un domaine ?)
Date	date du calendrier	dates minimum et maximum
Discret	prend une valeur possible parmi un ensemble prédéfini	ensemble fermé de valeurs possibles
Organisation	prend une valeur parmi les organisations	annuaires où trouver ces organisations
Site	prend une valeur parmi les sites	annuaires où trouver ces sites
Activité	prend une valeur parmi les activités	annuaires où trouver ces activités
Groupe	prend une valeur parmi les groupes auxquels appartient la personne	aucune

3.3.5.3.2. Les attributs obligatoires

Les attributs obligatoires que l'on gère pour une personne sont les suivants :

Attribut	**Nom court**	**Commentaire**
nom	nom	Commun à tous les éléments de Dyna. Toujours stocké en majuscules.
nom2	nom2	Autre nom sous lequel la personne est connue Toujours stocké en majuscules.
prénom	prenom	Toujours stocké avec les initiales en majuscules. Sans objet pour les entrées génériques.
prénom2	prenom2	Autre prénom sous lequel la personne est connue. Toujours stocké avec les initiales en majuscules. Sans objet pour les entrées génériques.
civilité	civilite	Mme, Mlle, ou M. Sans objet pour les entrées génériques.
groupe principal	gr_princ	Groupe principal de la personne. Cet attribut peut être vide.
mail	mail	Adresse principale de la messagerie. Ce champ peut être éventuellement utilisé pour stocker des e-mails secondaires, en dehors de alias.
téléphone	tel	Numéro(s) de téléphone de la personne, séparés par des caractères tiret lorsqu'ily a plusieurs numéros.
fax	fax	Numéro(s) de télécopie de la personne, séparés par des caractères tiret lorsqu'ily a plusieurs numéros.
bureau	bureau	Bureau de la personne (texte libre).
responsable	resp	Identité du responsable explicite de la personne, en dehors du cas où le responsable est celui de l'organisation principale de la personne.
organisation principale	org_princ	Organisation dans laquelle la personne travaille principalement.
organisations secondaires	orgs_secs	Organisations dans laquelle la personne travaille (ou est administrativement rattachée) en plus de l'organisation principale.
site principal	site_princ	Site sur lequel la personne travaille principalement.
sites secondaires	sites_secs	Sites sur laquelle la personne travaille en plus du site principal.
activité principale	ac_princ	Activité (projets, rôles, missions...) principale de la personne.
activités secondaires	acs_secs	Activités dévolues à la personne en plus de l'activité principale.
date d'arrivée	dte_arr	Date d'arrivée de la personne : celle-ci ne sera pas visible dans l'annuaire (sauf pour les exploitants et les administrateurs) avant cette date.
date de départ	dte_dep	Date de départ de la personne : celle-ci ne sera pas visible dans l'annuaire (sauf pour les exploitants et les administrateurs) après cette date. De plus ce champ est utilisé pour exécuter les mécanisme de désactivation automatique de la BAL après un certain délai après le départ d'une personne, en fonction de sa catégorie
informations complémentaires	infos_cpl	texte libre que les exploitants Dyna peuvent écrire afin de commenter une fiche personne.

Remarque concernant le responsable d'une personne : toute personne doit avoir, pour autant que cela soit possible, un responsable hiérarchique. L'algorithme pour le trouver est le suivant :

1. on prend le responsable explicite de la personne si celui-ci est positionné dans la fiche ;
2. à défaut on prend le responsable de l'organisation principale à laquelle appartient la personne ; si celle-ci est elle-même responsable de son organisation principale, on prend le responsable de l'organisation de niveau supérieur ;
3. à defaut on prend le responsable de l'organisation de niveau supérieur à l'organisation principale de la personne, et on remonte ainsi la hiérarchie jusqu'à l'organisation de niveau 1 ;
4. à défaut la personne n'a pas de responsable connu.

Cet algorithme est perfectible dans la mesure où le prend pas en compte ni les organisations secondaires, ni les responsables adjoints des organisations.

3.3.5.3.3. Les attributs supplémentaires

Pour chaque annuaire, on peut ajouter jusqu'à 12 attributs supplémentaires. Physiquement, dans la base de données, ces attributs sont stockés comme du texte, bien qu'ils respectent chacun une syntaxe parmi celles définies dans le tableau ci-dessus.

3.3.6. Une structure

Les structures sont représentées chacune par un arbre dont les noeuds et les feuilles contiennent des personnes et des groupes. Il n'existe aucune limite dans la profondeur de l'arbre. On a trois sortes de structures : les organisations, les sites et les activités. Pour chaque personne et chaque sorte de structure, on définit la structure principale, et éventuellement une ou plusieurs structures secondaires.
Une structure possède les propriétés suivantes :

- comme tout élément de Dyna, un numéro unique ;
- le nom de la structure, en utilisant les acronymes (exemple : le CRI) ;
- le nom de la structure, en développant les acronymes (exemple : le Centre de Ressources Informatiques) ;
- le numéro de la structure mère, avec 0 pour les structures de premier niveau ;
- le nom complet de la structure, qui est calculé automatiquement par Dyna par concaténation des noms des stuctures mères en partant du premier niveau jusqu'à la stucture elle-meme, séparés par des /. Exemple : SERVICE GENERAL/SIG.

3.3.6.1. Une organisation

Les organisations sont le reflet de l'organigramme de l'Université avec les composantes comme premier niveau et les départements (ou directions pour le Service Général de la Présidence) au second niveau. Une organisation possède les propriétés suivantes, en plus de celles qui sont communes aux structures :

- un nom court, qui est utilisé dans certaines circonstance comme lors de la génération automatiques d'adresses de messageries fonctionnelles ;
- le numéro unique de la personne qui est reponsable de cette organisation ;
- le numéro unique de la personne qui s'occupe du secretariat de cette organisation ;
- un nombre quelquonque de secrétaires parmi les personnes de l'annuaire ;

- un responsable unique. En l'absence de responsable, on utilisera le responsable de l'organisation de niveau supérieur, si l'organisation est de niveau supérieur ou égal à deux ;
- un nombre quelconque de responsables adjoints. Une organisation ne peut pas avoir de responsables adjoints sans avoir un responsable principal ;
- le numéro de téléphone du standard de l'organisation ;
- le numéro de télécopie de l'organisation ;
- l'adresse du site web de l'organisation ;
- le numéro unique du site par défaut auquel appartiendront les personnes membres de cette organisation.

Les groupes peuvent posséder un organisation. Les organisations ont éventuellement un dossier externe, dans la mesure où on effectue une synchronisation sur la base Harpege. Les notions de secrétaire, de responsable et de responsable adjoint servent notemment à générer automatiquement des mails fonctionnels, dans les dossiers de messagerie des organisations :

- dir.(nom court de l'organisation)@univ-nantes.fr alias direction.(nom court de l'organisation)@univ-nantes.fr
- sec.(nom court de l'organisation)@univ-nantes.fr alias secretariat.(nom court de l'organisation)@univ-nantes.fr

3.3.6.2. Un site

Les sites ont une signification géographique : à chaque site on attribue une adresse postale. Les sites de premier niveau sont généralement des campus ; au second niveau, on peut avoir des bâtiments, et si on le souhaite on peut pousser plus loin cette décomposition, par exemple en étages, bureaux, etc. Un site possède les propriétés suivantes, en plus de celles qui sont communes aux structures :

- un nom court ;
- une adresse postale ;
- un complément d'adresse (boite postale, etc) ;
- un code postal ;
- une ville ;
- le numéro de téléphone du standard ;
- le numéro de télécopie du site ;
- l'adresse du site web du site.

3.3.6.3. Une activité

Les activités permettent de classer les personnes en fonction de leurs rôles, projets, fonctions, etc. au sein de l'Université, c'est à dire les informations de nature hiérarchique que les organisations et les sites ne permettent pas de représenter.
Une activité ne possède pas de propriétés autres que celles qui sont communes aux structures.

3.3.7. Un groupe

Les groupes sont des ensembles informels de personnes définis soit explicitement, soit par leur appartenant directe ou non à une structure, soit par leur appartenance directe ou non à un autre groupe. Il faut veiller à ce qu'il n'existe pas de cycle dans la définition des groupes (exemple : un

groupe qui se contiendrait lui-même, ou deux groupes qui se contiendraient mutuellement). Une personne peut donc appartenir à un nombre arbitraire de groupes. Le contenu des groupes est recalculé dynamiquement lors de toute modification d'une entrée risquant de l'impacter : création de la fiche d'une personne, affectation à une structure, changement du contenu d'un groupe lui-même membre d'autres groupes, changement dans l'arborescence des structures, etc.

3.3.7.1. Interaction entre l'état d'une personne et son appartenance à des groupes

Directement ou non, un groupe contient toujours in-fine des personnes. Celles-ci peuvent être automatiquement exclues de leurs groupes si leur état de leur permet pas d'y appartenir :

- état en attente de validation ;
- état logiquement supprimé.

Ainsi, lorsqu'on supprime logiquement une personne, celle-ci est automatiquement exclue de tous les groupe auxquels elle appartenait. Cette opération peut être effectué manuellement, ou bien automatiquement lorsque la date de départ est atteinte. De même, lorsqu'une personne appartient à une organisation elle-même membre d'un groupe, et qu'elle est en attente de validation, il suffit qu'un exploitant valide sa fiche (qui passera à l'état normal) pour que cette personne soit automatiquement intégrée dans ce groupe.

3.3.7.2. Les groupes principaux

On a la possibilité d'indiquer un groupe principal parmi ceux auxquels appartient la personne. Ceci peut se répercuter sur un annuaire LDAP et être utile par exemple pour définir des permissions sur un serveur de fichiers. Le groupe principal est une information individuellement affectée à une personne. L'appartenance à un groupe donné peut déterminer automatiquement le groupe principal d'une personne, avec un notion de priorité afin de résoudre les conflits, ce qui autorise les affectations de groupes principaux à des personnes par cohortes plutôt que individuellement. On attribue une priorité arbitrairement égale à 100 l'affectation manuelle d'une personne à un groupe principal, ce qui permet de forcer de manière rétroactive l'appartenance aux groupes principaux.

De même, on peut indiquer qu'un groupe ne peut pas être le groupe principal d'une personne.

3.3.7.3. Les groupes mono et multi-annuaire

Un groupe est multi-annuaires s'il peut contenir des entrées appartenant à différents annuaires, mono-annuaire s'il ne peut contenir que des entrées appartenant au même annuaire que lui.

3.3.7.4. Autres fonctionnalités

Idée à creuser (proposée par Nathalie Vincent) : la notion de groupe basé sur des attributs autres que des structures. Par exemple, un groupe basé sur l'appartenance d'un étudiant à une formation afin d'attribuer des permissions sur un serveur. La ou les formations auxquelles est inscrit un étudiant fait partie des attributs optionnels de l'annuaire des étudiants.

3.3.8. Un conteneur et un objet contrôlable

On définit comme un conteneur un objet de Dyna qui peut en contenir d'autres. Les annuaires, les groupes et les structures sont des conteneurs. On peut s'appuyer sur cette notion afin d'attribuer des permissions à des utilisateurs. On définit comme un objet contrôlable un objet de Dyna sur lequel on peut affecter des permissions à des utilisateurs. Le contrôle peut avoir lieu pour visualiser un objet ou pour le modifier. Un objet contrôlable doit également être en mesure d'indiquer la liste des personnes autorisées à le modifier. Toute entrée est contrôlable, ainsi qu'un annuaire.

Progammatiquement, les conteneurs et les objets contrôlables sont des interfaces.

3.3.9. Un dossier additionnel

Un dossier additionnel est un ensemble d'informations qui se rattachent à une entrée. Une entrée peut ou non posséder un dossier additionnel selon sa catégorie :

Type de dossier additionnel	Personnes	Structures	Groupes
Authentification	X		
Permissions	X		
Externe	X	organisations seulement	
Messagerie	X		X
Système	X		X

Chaque entrée ne peut posséder au maximum qu'un seul dossier additionnel par type différent. Un dossier ne peut pas être partagé par plusieurs entrées. Par contre, au cours de sa vie certains types de dossiers peuvent être transférés d'une entrée à une autre. C'est à partir de l'entrée elle-même qu'on crée, retrouve, modifie et supprime les dossiers additionnels. Charge est laissée à l'entrée d'interdire de créer un dossier additionnel qu'elle ne peut pas détenir, ou de créer plusieurs dossiers additionnels de même type sur une seule entrée.
Lorsqu'une entrée est supprimée physiquement, l'ensemble de ses dossiers additionnels sont supprimés également.

3.3.9.1. Dossier d'authentification

Seule une personne peut posséder un dossier d'authentification.
Le dossier d'authentification contient les éléments nécessaires à la vérification du couple identifiant - mot de passe d'une personne. Ce dossier contient le mot de passe chiffré de toutes les manières (SSHA, MD5, Crypt, etc) demandées par les réplicateurs dépendant de l'annuaire de rattachement de la personne, et sachant gérer les mots de passe (implémentant l'interface ReplicateurAuth). De cette façon, il est possible de régénérer à la demande un annuaire répliqué, LDAP ou autre.
Le dossier d'authentification gère le lien avec la politique de compte de l'annuaire auquel appartient la personne associée à ce dossier. Il permet de gerer les notions d'expiration de compte (date au-delà de laquelle la personne ne peut plus se connecter) et de mot de passe (date au delà de laquelle la personne doit changer son mot de passe).

3.3.9.2. Dossier de permissions

Seule une personne peut posséder un dossier de permissions. Dans Dyna2, le dossier de permissions est séparé du dossier d'authentification.
A toute personne, on peut attribuer des permissions qui peuvent être soit globales (exemple : la permission d'administrateur de Dyna), soit liées à un objet contrôlable ou un ensemble d'objet contrôlables (exemple : la permission d'administrer l'annuaire du personnel de l'Universdité, pour les personnes appartenant à l'UFR Histoire ou à l'UFR de Sciences Economiques). NB : les entrées sont des objets contrôlables, de même que les annuaires.
Les permissions que l'on peut attribuer à une personne sont les suivantes :

Permission	**Donne le droit de :**	**Lié aux entrées**
ADMIN	administrer totalement Dyna	Non
EXPLOIT_ANNUAIRE	exploiter l'annuaire de Dyna pour un sous-ensemble des entrées appartenant à un ou plusieurs annuaires, avec des possibilités restreintes : - suppression logique des personne et non physique ; - pas de création de structures.	Oui.
EXPLOIT_AVANCE_ANNUAIRE	exploiter l'annuaire de Dyna pour un sous-ensemble des entrées appartenant à un ou plusieurs annuaires. Les limitations liées à EXPLOIT_ANNUAIRES sont levées.	Oui.

Remarque concernant la colonne “lié aux entrées” : ce point devra être précisé car on n'apporte aucune précision concernant l'appartenance d'une entrée à un annuaire. En effet, une structure peut contenir des entrées appartenant à des annuaires autres le sien.

3.3.9.3. Dossier externe

Lorsqu'une entrée provient d'une base externe par synchronisation, elle possède un dossier externe permettant d'assurer sa traçabilité. On sait ainsi, par exemple, qu'un étudiant provient de GEODE et un personne d'HARPEGE, avec son identifiant externe. Dans certains cas, il s'avère qu'une personne peut provenir de plusieurs sources différentes et il serait intéressant de les conserver toutes, ce qui n'est actuellement pas possible avec Dyna1. Outre les personnes, les organisations peuvent posséder un dossier externe.
Un dossier externe se compose d'un nombre arbitraire de références externes, chacune liée à une base de données externe. Pour chaque référence externe, on indique :

- l'entrée concernée (sa signature sous la forme d'une lettre et d'un numéro unique) ;
- la provenance ;
- la référence sous forme d'une chaine de caractères ;
- la date de création du dossier externe ;
- la date de dernière synchronisation.

On doit pouvoir retrouver les dossiers externes liés à une entrée, et un dossier externe à partir du couple unique provenance - référence.

3.3.9.4. Dossier de messagerie

Dyna1 connaît déjà le dossier de messagerie afin de stocker des informations relatives à la messagerie électronique. Ce sont :

- le répertoire de messagerie,
- le serveur de messagerie,
- les alias
- les adresses de redirection
- le fait que le compte soit en redirection pure, c'est à dire sans délivrance locale des message. Il faut alors que le dossier de messagerie possède au moins une adresse de redirection ;
- la valeur du quota en octets
- l'activation ou non de la boîte aux lettres. Une BAL inactive ne interdit à l'utilisateur de relever son courrier, et d'envoyer des messages.

On reporte ces mêmes fonctionnalités sur Dyna2.

3.3.9.5. Dossier système

Le dossier système contient les données nécessaires à une personne pour se connecter à un serveur de fichiers. Ces informations seront propagées à travers un réplicateur LDAP.
Les informations manipulées par le dossier système sont les suivantes :

- état du compte (actif ou inactif) ;
- serveur de fichiers ;
- répertoire d'accueil ;
- quota.

3.3.10. Un réplicateur

Un réplicateur appartient à un annuaire unique. Il sait comment propager toute mise à jour en aval de Dyna : par exemple, dans un annuaire LDAP, ou bien en exécutant un script, ou bien encore en invoquant le requêteur d'un autre serveur Dyna. Il effectue la journalisation des actions qui lui sont demandées avec une verbosité que l'administrateur peut configurer.

3.3.10.1. Les différents types de réplicateurs

Le tableau suivant présente les différents types de réplicateurs présents dans Dyna2.

Type	Utilité	Commentaires
LDAP	Répliquer les entrées dans un annuaire LDAP. On utilisera des sous-classes de ce réplicateurs en fonction des besoins, du schéma et du DIT utilisés par le serveur LDAP distant : - annuaires de la messagerie ; - annuaires système ; - annuaires de contact ; - annuaires propre aux différentes composantes (Polytech a déjà ses propres réplicateurs LDAP)	déjà dans Dyna1, mais à faire évoluer.
SHELL	Exécuter un shell sur le serveur Dyna (ou un autre serveur via une commande SSH), comportant en paramètre des informations sur l'entrée à répliquer .	déjà dans Dyna1.
DYNA (XML-RPC)	Répliquer les entrées dans un autre serveur Dyna via un client XML-RPC.	à développer dans Dyna2.

3.3.10.2. Propriétés communes aux réplicateurs

3.3.10.2.1. Types d'entrées répliquées

Tout réplicateur ne sait pas forcément répliquer toute entrée quelle que soit son type. Par exemple, certains réplicateurs peuvent savoir répliquer les groupes, d'autres non. De plus, l'administrateur de Dyna peut volontairement limiter la réplication à certains types d'entrées. Par exemple, répliquer seulement les personnes vers tel annuaire LDAP, et pas les groupes ni les organisations.

3.3.10.2.2. Validité d'un réplicateur

Un réplicateur est valide lorsque sa configuration a été vérifiée. Lorsqu'un réplicateur est marqué comme invalide, il n'accepte aucune entrée.

3.3.10.2.3. Réplicateur actif / inactif

L'administrateur de Dyna peut rendre un réplicateur inactif. afin d'éviter qu'il propage des entrées.

3.3.10.2.4. Testabilité d'un réplicateur

Un autre point important est la possibilité de tester le service correspondant à un réplicateur. Tous les réplicateurs ne sont pas nécessairement testables (pour un réplicateur de type SHELL cela n'a pas de sens), mais tous les réplicateurs de type LDAP le sont. Le résultat d'un test est de type feu vert - feu orange - feu rouge suivant que le service correspondant soit opérationnel, dégradé ou disponible. Un réplicateur peut éventuellement proposer différents tests, plus ou moins poussés. Ce sera notemment le cas des réplicateurs LDAP. Chaque réplicateur devra donc indiquer dans sa classe la liste des classes de tests possibles.

3.3.10.2.5. Réplicateurs de mots de passe

Certains réplicateurs savent travailler avec les mots de passe des utilisateurs. C'est le cas des annuaires LDAP, comme cela peut être le cas d'autres types de réplicateurs. C'est également le cas des réplicateurs de type XML-RPC puisque Dyna sait gérer les mots de passe. Afin de signaler qu'une classe donnée de réplicateurs sait traiter les mots de passe, on implémente l'interface ReplicateurAuth. Celle-ci impose les méthodes suivantes :

- verifierPwd(String uid,String pwd); (vérifier le mot de passe d'un utilisateur)
- modifierPwd(String uid,String pwd); (modifier le mot de passe d'un utilisateur)
- getNumAlgoChiffrement(); (quel est l'algorithme de chiffrement à utiliser pour les mots de passe)
- doitEncoderPwdSamba(); (faut-il chiffrer les mots de passe selon les algorithme nbPassword et lmPassword de Samba ?)

Remarque : la réplication les mots de passe doit pouvoir être déactivée par l'administrateur pour un réplicateur donné, même si celui-ci sait gérer les mots de passe. Par exemple, un annuaire LDAP de contact ne doit pas détenir de mots de passe même s'il en est intrinsèquement capable.

Le protocole LDAP (RFC 2251) spécifie différentes manières de chiffrer les mots de passe. Il existe cinq algorithmes :

- Crypt (déconseillé mais disponible) ;
- MD5 ;
- SMD5 (MD5 salé) ;
- SHA ;
- SSHA (SHA salé).

Ce sont des algorithmes de chiffrement à sens unique : à partir du texte en clair on gérère le cryptogramme, le contraire est impossible. Dans le cas des algorithmes salés, on détermine le sel aléatoirement, puis on le concatène au cryptogramme. Le salage améliore la résistance aux attaques par dictionnaire, puisque pour un message en clair donné, il existe un très grand nombre de cryptogrammes possibles, contre un seul pour les algorithmes sans salage. Pour vérifier un mot de passe, on effectue un nouveau chiffrement (éventuellement avec le sel fourni), et on vérifie l'identité des cryptogrammes.

A cette liste il faut ajouter les algorithmes de chiffrement pour Samba : le ntPassword et lmPassword. Ceux-ci ne sont pas salés et nécessitent de passer par une commande externe pour les obtenir : mkntpwd.

L'administrateur de Dyna a la possibilité de choisir pour chaque réplicateur LDAP lequel de ces cinq algorithmes de chiffrement sera utilisé, plus une sixième possibilité déconseillée : l'absence de chiffrement. Lorsqu'on change le mot de passe d'un utilisateur, celui-ci appartient à un annuaire possédant certain nombre de réplicateurs LDAP, donc nécessitant de chiffrer le mot de passe de différentes manières. Le dossier d'authentification de l'utilisateur contiendra donc autant de cryptogrammes de son mot de passe que de manières de le chiffrer demandés par les réplicateurs LDAP auquel son annuaire appartient. Le stockage chiffré des mots de passe présente l'avantage de pouvoir régénérer toute fiche, et globalement tout annuaire LDAP, en cas de perte des données sur le serveur LDAP, à partir des données contenues dans le référentiel.

3.3.10.2.6. Lancement global d'un réplicateur

L'administrateur de Dyna doit poivoir lancer globalement la réplication des entrées appartenant à un annuaire auquel appartient ce réplicateur :

- soit toutes les entrées concernées par ce réplicateur (cette fonctionnalité existe déjà dans Dyna2) ;
- soit les entrées concernant ce réplicateur et appartenant directement ou non à une organisation donnée (nouvelle fonctionnalité).

3.3.10.3. Propriétés spécifiques à un réplicateur LDAP

Les paramètres de configuration d'un réplicateur de type LDAP sont décrits dans les paragraphes suivants.

3.3.10.3.1. Paramètres du serveur

Tout réplicateur LDAP doit connaître le serveur LDAP vers lequel il doit propager les entrées :

- nom DNS du serveur (ou adresse IP) ;
- port LDAP (par défaut, 389 pour LDAP, 636 pour LDAPS) ;
- LDAP ou LDAPS, autrement chiffrement TLS ou non. Si au moins un réplicateur LDAP possède la propriété TLS, il faut intégrer un certificat X.509 à la JVM de Dyna (dans le keystore).
- compte LDAP à utiliser : le DN et le mot de passe. Ce compte doit posséder la permission de modifier l'ensemble de l'annuaire LDAP à partir de la base du DIT fournie en paramètre du réplicateur. A étudier : chiffrer ce mot de passe dans la base de données de Dyna. Actuellement, ce n'est pas fait dans Dyna1 ;
- base du DIT ;
- éventuellement la liste des réplicas connus, afin de pouvoir tester la réplication.

3.3.10.3.2. Attributs LDAP

Pour un réplicateur LDAP, il est nécessaire d'établir une correspondance entre les attributs des personnes définis pour l'annuaire auquel appartient cette personne, et les attributs LDAP qui seront propagés. Pour chaque attribut, il existe trois possibilités :

- on ne réplique pas cet attribut dans ce réplicateur ;
- on réplique cet attribut avec son nom et sa syntaxe par défaut, correspondant aux cas les plus courants ;
- on réplique cet attribut sous un nom spécifique, avec un syntaxe éventuellement spécifique. Charge alors au réplicateur LDAP de générer les noms et valeurs de cette attribut d'une manière appropriée.

3.3.10.3.3. Le DIT et le DN

Chaque annuaire LDAP possède sa propre façon de “ranger” ses entrées dans un arbre : le DIT (Directory Information Tree). La position d'une entrée LDAP dans le DIT est indiquée par le DN (Distinguished Name), qui est une chaîne de caractères constituée de couples <attribut>=<valeur> dont chacun constitue le RDN (Relative Distinguished Name), unique pour le noeud correspondant du DIT. L'élément de base servant à construire un DIT est généralement l'unité organisationnelle (OrganizationalUnit, ou OU) : il s'agit d'une classe LDAP spécialement conçue pour gérer l'arborescence des entrées. Dans ce domaine, il existe deux manières de construire un DIT :

- DIT “arborescent” : les unités organisationnelles s'emboîtent d'une manière calquée sur les organisations ;
- DIT “plat” : les unités organisationnelles sont fixes : on a généralement une OU pour les personnes et une OU pour les groupes. C'est notamment le cas des annuaires de type SupAnn (standard édicté par le CRU).

Dyna2 doit savoir gérer les deux types de DIT. Chaque réplicateur LDAP devra donc posséder une méthode getDN(AnnuEntree e) permettant de positionner l'entrée e dans son DIT. Dans la classe de base des réplicateurs LDAP, on calcule le DN simplement à partir du nom de la racine du DIT, qui est un paramètre du réplicateur, et de l'organisation principale dans le cas d'un DIT arborescent. Dans les classes dérivées, on peut personnaliser le DN des entrées, par exemple afin d'insérer des unités organisationnelles spécifiques.

3.3.10.3.4. Les classes LDAP

Pour chaque type d'entrée, on doit indiquer au réplicateur LDAP la liste des classes à employer. Il fait indiquer : - la classe structurelle des entrées (personnes, groupes, organisations, etc) ; - les classes abstraites dont la classe structurelle hérite ; - les éventuelles classes auxiliaires, permettant d'implémenter des attributs supplémentaires. Ces classes doivent être connues du réplicateur, qui doit savoir quels attributs générer sur les entrées en fonction de la classe. Si une classe inconnue est spécifiée, le réplicateur sera marqué comme invalide. Le tableau suivant donne la liste des classes LDAP que Dyna devra être capable de répliquer.

Type d'entrée	Classes LDAP
Personne	utiliseInetOrgPerson utiliseOrganizationalPerson utilisePosixAccount utiliseShadowAccount utiliseSambaAccount utiliseSambaSamAccount utiliseQmailUser
Groupes	utilisePosixGroup utiliseGroupOfUniqueNames utiliseGroupOfNames utiliseSambaGroupMapping
Organisation	organizationalUnit

Actuellement il n'est pas prévu qu'un réplicateur LDAP sache répliquer d'autres types d'entrées.

3.3.10.3.6. Interrogation directe d'un annuaire LDAP

Un exploitant de Dyna doit pouvoir vérifier qu'un réplicateur propage correctement les informations concernant une entrée d'annuaire sur laquelle il a accès. Pour cela, sur l'écran de consultation et/ou de modification de la fiche, il doit parvenir à un menu lui permettant de choisir quel réplicateur il souhaite interroger parmi ceux qui sont valides pour cette entrée. Cette fonctionnalité existe déjà sur Dyna1.
Si la liste des réplicas (slurpd) connus pour cet annuaire LDAP est renseigné dans Dyna, celui-ci doit pouvoir également vérifier que serveur LDAP propage bien les modifications qu'on lui envoie pour cette entrée. Ceci est une nouvelle fonctionnalité de Dyna2.

3.3.10.3.7. Tests LDAP plus poussés

Dyna1 implémente déjà un test simple, consistant à ouvrir puis fermer immédiatement une connexion LDAP vers l'annuaire testé. Il serait possible d'effecteur des tests plus poussés :

- comptage du nombre d'entrée. La configuration du réplicateur doit alors permettre d'indiquer le nombre minimum d'entrées que l'on doit trouver pour que le test soit considéré comme bon ;
- comparaison globale entre le maître et ses réplicas (slurpd) connus, si ceux-ci ont été renseignés pour ce réplicateur ;
- test des index. La configuration du réplicateur doit alors permettre d'indiquer la liste des attributs indexés, d'une manière unique ou non ;

3.3.10.3.8. Le réplicateur LDAP SUPANN

Le CRU a formé un groupe de travail, SUPANN, afin d'homogénéiser les annuaires LDAP utilisés dans l'enseignement supérieur. Il s'agit d'un ensemble de recommandations pour une visibilité externe LDAP des annuaires d'établissements, et un modèle de DIT et de schéma permettant une intégration de différents produits comme ceux de l'AMUE.
L'Université a besoin d'offrir une vue de son référentiel conforme au standard SUPANN. Pour cela, Dyna propose une classe de réplicateur LDAP spécialisée pour les classes LDAP et le schéma de SUPANN : *ReplicateurLDAPSupAnn.*
Il suffit qu'un annuaire possède un réplicateur de cette classe pour que ses entrées soient répliquées dans une base LDAP conformément au standard SUPANN. Lorsque SUPANN, actuellement en version 1, évoluera vers sa version 2, il faudra mettre à jour la classe ReplicateurLDAPSupAnn.

3.3.10.3.9. Le pool de connexions à l'annuaire LDAP

Un réplicateur de type LDAP possède son propre pool de connexion au serveur distant, de manière à limiter le nombre de connexions et de déconnexions LDAP, très coûteuses en performances. Lors de chaque demande de réplication, le réplicateur s'alloue une connexion prélevée sur le pool, et la restitue après l'opération. Le pool vérifie de manière périodique l'état des connexions de manière à abandonner celles qui ne fonctionnent plus (suite par exemple à la défaillance temporaire d'un serveur LDAP), et l'adéquation entre la charge demandée par les réplicateurs et le nombre de connexions disponibles (les connexions surnuméraires sont fermées de manière à économiser la mémoire). Ce pool de connexion LDAP utilise un code commun au pool de connexion au SGBD relationnel de Dyna, avec une classe spécifique aux connexion vers un serveur LDAP.

3.3.10.5. Propriétés spécifiques à un réplicateur SHELL

Un réplicateur SHELL exécute un script lors de tout ajout, modification ou suppression d'une entrée. Ce script peut être un programme ssh permettant d'exécuter une action sur une autre machine, en utilisant l'authentification par certificats SSH de manière à éviter la saisie de mots de passe. Lors de l''exécution, le réplicateur appelle le script en ajoutant les options :

- -a : ajout d'une entrée ;
- -m : modification d'une entrée ;
- -s : suppression d'une entrée.

Puis le réplicateur indique le type d'entrée répliquée :

- -p : une personne ;
- -g : un groupe ;
- -o : une organisation.

On pourra ajouter d'autres types d'entrées en cas de besoin (sites, activités, etc). L'information suivante sur la ligne de commande est le numéro unique de l'entrée : -num <numéro unique de l'entrée> Les autres informations indiquées sur la ligne de commande dépendent de l'entrée. Pour les personnes : -<nom court de l'attribut> <valeur de l'attribut> ... Dans le cadre du projet GROMEL les réplicateurs SHELL sont utilisés pour créer les répertoires nécessaires aux comptes de messagerie des utilisateurs. Pour cela, on se connecte en ssh sur les différents serveurs de messagerie.

3.3.10.6. Propriétés spécifiques à un réplicateur XML-RPC

On indique :

- le nom du serveur Dyna situé en aval, vers lequel on propage les demandes de réplication ;
- le port utilisé pour XML-RPC ;
- l'éventuel identifiant et mot de passe dans le cas où on est en mode XML-RPC paranoïaque.
- si on est en mode de sécurisation par TLS ou non

3.3.10.7. Réplicateurs spéciaux

Lorsque les besoins sont spécifiques et que les réplicateurs standards de Dyna ne suffisent pas, il est possible à l'administrateur de Dyna de déclarer un réplicateur spécial. Il doit alors fournir le nom de la classe de son réplicateur, sachant que cette classe doit implémenter les méthodes abstraites communes à tous les réplicateurs (ajouter, modifier, supprimer une entrée, savoir s'il faut répliquer une entrée ou non, tester le réplicateur). Un réplicateur spécial peut appartenir à n'importer quel type : LDAP, SHELL ou DYNA, voire implémenter un nouveau type indépendant de ceux-ci. En fonction du type, il peut y avoir des méthodes supplémentaires à implémenter dans la classe. Une contrainte importante est de devoir intégrer la classe du réplicateur spécial dans la JVM du moteur Dyna, et de redémarrer celui-ci lors de toute modification.

3.3.10.8. Statistiques sur les réplicateurs

Il faut également tenir à jour des statistiques concernant : - les erreurs de réplication, afin d'être en mesure de remonter des alertes ; - les performances de la réplication, afin de pouvoir diagnostiquer des lenteurs lors de la propagation des entrées : temps minimum, moyen, maximum. Ceci est un nouveau développement pour Dyna2.

3.3.11. L'ordonnanceur de réplication

L'ordonnanceur de réplication est unique. Il est chargé de prendre en charge des demandes de mise à jour par les différents annuaires, et doit savoir quand les différentes réplicateurs de cet annuaire doivent entrer en action. Il possède des files d'attente gérant la priorité des réplicateurs. Il sait gérer la reprise sur erreur en cas de problème avec l'un des réplicateurs.

3.3.11.1. Le ré-ordonnancement dynamique

Toute demande de réplication doit être atomique, c'est à dire n'impacter qu'une seule entrée d'un seul système d'annuaire en aval. Comme il existe des dépendances entre les entrées (exemple : le changement d'organisation d'une personne peut entraîner des changement de groupes), il faut ordonnancer à nouveau les demandes de réplication dépendantes, et ce pour tous les réplicateurs concernés. Ces nouvelles demandes de réplication pourront être pourvues d'une priorité différente, la constitution des groupes étant généralement moins urgente que la réplication des entrées individuelles. Cette ré-ordonnancement dynamique est une nouveauté de Dyna2 : Dyna1 effectue la réplication d'une seule passe quelle que soient les dépendances entre les entrées. Cela présentera l'avantage d'une meilleure fluidité dans la propagation des modifications vers les annuaires LDAP.

3.3.11.2. La remontées d'alertes

Une nouveauté de Dyna2 sera la mise en place de remontées d'alerte en cas d'erreurs de réplication, afin d'éviter que l'administrateur de Dyna soit obligé de consulter régulièrement les journaux de réplication pour vérifier le bon fonctionnement des réplicateurs. Ces alertes pourront être transmises :

- par mail ;
- via une plate forme d'administration système telle que Cacti.

3.3.12. Une politique de comptes

Une politique de comptes est un objet associé à un annuaire gérant le cycle de vie des comptes des utilisateurs. Elle permet de spécifier :

- quels contrôles sont à effectuer sur un nouveau mot de passe demandé par un utilisateur. Ces contrôles, chacun implémenté par une classe - un contrôleur de mots de passe - sont indiqués dans la configuration de la politique de compte ;
- le nombre de jours que l'on attend après le départ d'un utilisateur, avant l'expiration de son compte ;
- le nombre de jours que l'on attend après le changement du mot de passe, avant d'imposer un nouveau changement ;
- le délai dont dispose l'utilisateur pour changer son mot de passe après que celui-ci ait été

changé par un exploitant ou un administrateur, avant blocage de son compte.

Il existe un politique de comptes par défaut, et chaque annuaire peut posséder sa propre politique de comptes qui remplace alors la politique de comptes par défaut.

Il faut natuellement être administrateur de Dyna pour gérer les politiques de comptes.

3.3.12.1. Un contrôleur de mots de passe

Un contrôleur de mots de passe est un objet chargé de vérifier que le nouveau mot de passe que souhaite utiliser une personne est acceptable. Dyna prédéfinit un certain nombre de ces contrôleurs. Une politique de comptes peut posséder un nombre arbitraire de contrôleurs de mots de passe, qui seront exécutés dans l'ordre. Une mot de passe ne sera accepté que si tous les contrôleurs donnent leur feu vert. Dans le cas contraire, l'utilisateur verra s'afficher un message d'erreur provenant du premier contrôleur ayant refusé le mot de passe. Ce message devra expliquer clairement pourquoi le mot de passe est refusé, invitant la personne à en choisir un nouveau qui soit plus résistant aux attaques. Lorsqu'une politique de comptes ne spécifie aucun contrôleur, tous les mots de passe sont acceptés.

Le tableau suivant énumère les contrôleurs standards proposés par Dyna, sachant qu'on peut aisément en développer de nouveaux pour des besoins plus spécifiques.

Nom du contrôleur	Classe	Description
Contrôleur simple	ControlePwdSimple	vérifie que le mot passe comporte un nombre minimum de caractères, dont un nombre minimum de caractères non alphabétiques. L'administrateur indique ces nombre minimaux de caractères.
Contrôleur par dictionnaire	ControlePwdDico	vérifie que le mot de passe n'appartient pas à un dictionnaire de mots de passe triviaux
Contrôleur de non réutilisation	ControlePwdNonReutil	vérifie que le mot de passe n'est pas réutilisé avant un certain délai que l'on peut paramétrer.

3.3.13. Un objet de Dyna

Les entrées d'annuaire, les annuaires et les réplicateurs sont des objets Dyna. Ceux-ci présentent des caractéristiques communes :

- ils sont identifiés par un numéro unique pour leur classe. Ce numéro leur est attribué par le référentiel ;
- ils sont identifiés de manière unique à travers tous les objets Dyna par leur signature. La signature se compose d'une ou plusieurs lettres majuscules identifiant la classe de l'objet, suivie du caractère '_' et du numéro de cet objet. Par exemple : P_15236 identifie une personne.
- ils sont susceptibles d'être placés dans le cache de Dyna : ceci permet d'éviter de trop solliciter la base de données.

3.3.14. Les interfaces Java de Dyna

Les objets de Dyna peuvent implementer des interfaces afin de signaler qu'ils comportent des caractéristiques supplémentaires.

3.3.14.1. Un service de Dyna

Un service de Dyna est un objet qui a une existence durable en mémoire, qui doit être démarré au démarrage de Dyna et terminé lors de l'arrêt de Dyna. Le service doit également être capable de dire s'il est déjà démarré ou non.
Un service est une interface du nom de DynaService.

3.3.14.2. Un élément contrôlable

Un élément contrôlable est un élément sur lequel Dyna exercera un contrôle d'accès, c'est à dire vérifiera que l'utilisateur a bien le droit de visualiser ou de modifier cet objet.
Les verbes associés à cette interface sont les suivants :

- public boolean peutEtreVuPar(Personne p) : renvoie vrai si la personne p peut visualiser l'objet ;
- public boolean peutEtreModifiePar(Personne p) : renvoie vrai si la personne p peut modifier l'objet ;
- public IntSet lnPersonnesHabilitees() : renvoie la liste des numéros uniques des personnes habilitées à modifier l'objet.

On pourrait ajouter une méthode pour obtenir la liste des personnes habilitées à visualiser un objet mais en pratique ce n'est pas utile.

3.4. La base de données référentielle

Dyna a besoin d'une base de données référentielle, de type SQL, afin de stocker ses données de manière pérenne. Le choix se porte sur MySQL pour des raisons de performances, de stabilité et de simplicité d'utilisation, sachant que le code SQL étant portable, il sera possible de migrer vers un autre moteur relationnel moyennant un effort limité.

3.4.1. Bases de données à créer

Nous avons besoin de deux databases (bases de données indépendantes selon la terminologie MySQL) :

- dyna : tous les objets de Dyna y sont sérialisés indépendamment de l'annuaire ;
- journal : tous les événements stockés par le journal de Dyna.

Nous n'avons besoin que d'un seul compte MySQL : dyna. Seul le moteur de méta-annuaire accède directement à la base de données référentielle.

3.4.2. Le pool de connexions

Afin d'optimiser les performances du moteur MySQL, on utilise un pool de connexions. Celui-ci permet la réutilisation de connexions. Le pool conserve en mémoire un certain nombre de connexions

ouvertes. Régulièrement, ces connexion sont testées et renouvelées si elles s'avèrent défectueuses, suite à un problème passager avec le serveur de base de données par exemple. On teste également l'adéquation entre la charge demandée au pool et le nombre de connexions disponibles. Lorsqu'un objet de Dyna a besoin d'une connexion, au lieu de l'ouvrir et de la fermer lui-même sur la base de données (très coûteux en performances), il demande une connexion au pool, en mode exclusif ou non. Le mode exclusif est préférable pour les opérations modifiant les données. Après usage, il restitue la connexion au pool. Cette technique permet de réduire très fortement la charge du serveur de bases de données.

3.5. Le serveur web Dyna

Le serveur web Dyna aura l'exclusivité de l'interaction entre les utilisateurs et le moteur de méta-annuaire. Il communiquera avec le moteur Dyna au travers de l'interface XML-RPC. Il n'est pas indispensable que le serveur web Dyna soit écrit en java comme le moteur Dyna dans la mesure où l'API XML-RPC est disponible dans différents langages, dont PHP, Perl, etc. On choisit le langage PHP, fonctionnant avec le moteur HTTP Apache (Apache1 ou Apache2 : à voir). Cette séparation aura l'avantage de garantir qu'aucun accès au référentiel ne sera directement effectué hors du contrôle du moteur de méta-annuaire. L'inconvénient est que cela ajoute un étape nécessitant une conversion en XML et la passage par un service HTTP (voire HTTPS), ce qui est forcément coûteux en termes de performances.

3.6. La synchronisation des bases de données en amont

On aura deux types de synchronisation :

3.6.1. La synchronisation de type PULL

Dans ce schéma, Dyna vient chercher lui-même les informations dont il a besoin en se connectant sur des bases de données en amont. Pour cela, il est nécessaire de passer par un mécanisme de traitements régulés par un ordonnanceur, afin de gérer leur ordre de passage, la journalisation, la remontée de problèmes éventuels, etc. Dans la plupart des cas, ce type de synchronisation se traduit par un traitement s'exécutant toutes les nuits. L'ordonnanceur de traitements convertit les ordres de mise à jour des entrées en XML, puis il s'adresse au requêteur pour les propager. Il n'est pas indispensable qu'il s'exécute dans la même machine virtuelle Java que le requêteur.

L'avantage de ce type de synchronisation est sa simplicité. Il est facile de vérifier que tout se passe bien en consultant le journal des traitements. L'inconvénient majeur est la nécessité d'attendre la prochaine session de traitements avant qu'une information soit propagée. Actuellement la synchronisation de la base de données de personnel de l'Université HARPEGE utilise ce mécanisme, pour les personnels ainsi que pour les organisations.

3.6.2. La synchronisation de type PUSH

Dans ce cas, ce sont les bases de données externes qui sollicitent Dyna à chaque fois qu'ils détectent qu'une entrée doit être ajoutée, modifié ou supprimée. Dyna accueille ces requêtes grâce à son

requêteur, pouvant être le même objet que celui utilisé par les client web. L'avantage de ce type de synchronisation est sa réactivité, puisque les informations sont mises à jour dans le référentiel puis dans les annuaires LDAP immédiatement. Ceci est indispensable pour des attributs tels que les mots de passe. Cela se paie par une certaine complexité des communications, notamment par la nécessité de développements spécifiques sur des serveur extérieurs à Dyna. Il faut également mettre en place un mécanisme de reprise automatique sur erreur, afin d'éviter les incohérences d'annuaires dues à un problème de communication momentané entre les bases de données en amont et Dyna.
Actuellement la synchronisation de la base de données des étudiants de l'Université GEODE utilise ce mécanisme, à travers un client Dyna échangeant des données dans un format ad-hoc. Dans Dyna2 il faudra remplacer ce mécanisme par des appels au requêteur Dyna XML-RPC.

4. Services fournis par Dyna

Les services seront fournis par le requêteur. Celui-ci aura pour but d'assurer l'interface entre le ou les serveurs web de Dyna et le moteur de méta-annuaire. Pour cela, il devra implémenter des méthodes accessibles par XML-RPC, qui se classent parmi les catégories suivantes : Concernant les éléments de Dyna en général :

- recherche d'un élément à partir de sa signature.

Concernant les entrées d'annuaire :

- recherche d'une entrée à partir de sa classe et de son numéro unique (1 par type d'entrée) ;
- recherche d'entrées dans un annuaire à partir de différents critères (1 par type d'entrée) ;
- ajout, modification, suppression d'entrées (1 par type d'entrée). Ceci entraîne : l'exécution des filtres, la modification de la base de données référentielle, et l'exécution de tous les réplicateurs via l'ordonnanceur de réplication ;
- soumission d'une entrée au format XML de Dyna ;
- liste des dossiers additionnels existants liés à une entrée donnée ;
- liste des dossiers additionnels qu'il est possible de créer pour une entrée donnée ;
- récupération d'un dossier additionnel donné pour une entrée donnée.

Concernant les dossiers additionnels en général :

- liste des dossiers additionnels appartenant à une entrée donnée ;
- vérification qu'une entrée possède un dossier additionnel d'un type donné.

Concernant l'authentification :

- vérification qu'un nouveau mot de passe est acceptable suivant la politique de comptes s'appliquant à une personne donnée ;
- modification du mot de passe pour une personne, avec impact dans tous les réplicateurs sachant manipuler des mots de passe ;
- vérification du mot de passe d'un utilisateur en utilisant le réplicateur le plus prioritaire sachant vérifier un couple identifiant - mot de passe ;
- faire expirer le compte ou le mot de passe d'une personne maintenant ou à une date donnée ;
- vérification que le mot de passe d'une personne n'est pas expiré ;
- vérification que le compte d'une personne n'est pas expiré ;

Concernant l'autorisation :

- vérification qu'un utilisateur donné a le droit de voir un objet Dyna donné ;
- vérification qu'un utilisateur donné a le droit de créer, modifier ou supprimer un objet Dyna donné ;
- liste des utilisateurs ayant le droit de lire / modifier un objet Dyna donné ;
- affectation ou révocation d'une autorisation donnée sur un objet Dyna donné, à un utilisateur.

Concernant les dossiers externes :

- liste des dossiers externes d'une entrée donnée ;
- ajout / modification / suppression d'un dossier de externe pour une entrée donnée, et pour un type donné de base externe (p. ex. HARPEGE) ;
- recherche d'un dossier externe en fonction de différents critères de ce dossier (p. ex. la référence externe).

Concernant les dossier additionnels de messagerie :

- ajout / modification / suppression du dossier de messagerie pour une entrée donnée ;
- recherche d'un dossier de messagerie en fonction de différents critère de ce dossier (p. ex. un alias).

Concernant les dossier additionnels système :

- ajout / modification / suppression du dossier système pour une entrée donnée ;
- recherche d'un dossier système en fonction de différents critères de ce dossier (p. ex. un répertoire d'accueil).

Concernant les annuaires :

- ajout, modification, suppression et recherche d'un annuaire ;
- informations globales concernant un annuaire : nombre d'entrées de chaque classe, etc ;
- consultation de la liste ordonnée des attributs pouvant être affectés à une personne appartenant à une annuaire donné.
- reconstruction globale d'un annuaire pour un ou plusieurs réplicateurs, entrée par entrée.
- reconstruction d'un annuaire pour les entrées dépendant d'une organisation donnée et pour un ou plusieurs réplicateurs, entrée par entrée.

Concernant les filtres :

- ajout, suppression, modification de l'ordre d'exécution d'un filtre pour un annuaire donné ;
- consultation de la liste ordonnée des filtres affectés à un annuaire donné.

Concernant les réplicateurs :

- ajout, modification, suppression et recherche d'un réplicateur ;
- liste de tous les réplicateurs dépendant d'un annuaire par ordre de priorité ;
- exécution d'une action (ajout, modification, suppression) sur une entrée par un réplicateur.

Concernant les traitements :

- ajout, modification, suppression et recherche d'un traitement ;
- exécution directe d'un traitement.

Concernant les permissions :

- affectation, retrait d'une permission sur une entrée (ou un ensemble d'entrées) à un utilisateur ;
- recherche des utilisateurs possédant une permission donnée sur une entrée donnée ;
- vérification qu'un utilisateur possède ou non une permission données sur une entrée donnée.

Concernant la journalisation :

- ajout d'une ligne dans le journal ;
- recherche dans le journal concernant un utilisateur, une plage de dates, etc ;
- consultation du journal concernant un élément.

4.1. L'accès aux objets de Dyna

Un objet de Dyna est stocké dans le référentiel. Il est identifié par une signature unique sous la forme d'un code sur une ou plusieurs lettres représentant sa classe, et d'un numéro unique pour cette classe. Les entrées sont des objets Dyna, tout comme les annuaires, les réplicateurs, etc. On doit donc pouvoir, à travers l'interface XML-RPC, accéder à un objet Dyna :

- à partir de sa signature,
- à partir se son numéro unique, connaissant déjà sa classe,
- à partir de différents attributs, et en construisant une requête SQL sur le référentiel renvoyant une liste ordonnée de numéros d'objets Dyna.

Certaines classes présentent d'autres attributs uniques permettant une recherche : les personnes possèdent un identifiant unique (uid), les groupes un nom unique et les structures un nom unique branche leur branche (ex : X/Y/Z identifie de manière unique une structure). Pour chaque objet, on a ces méthodes :

- recherche en fonction de son numéro unique,
- recherche en fonction de son identifiant unique,
- recherche en fonction de différents critères (requête pouvant ramener plusieurs objets),
- ajout,
- modification,
- suppression.

4.2. L'autorisation

Dyna2 permettra de gérer l'autorisation de tout utilisateur sur tout objet Dyna. Une autorisation pourra être :

- soit directe : autorisation sur un objet unique ;
- soit indirecte : autorisation sur un conteneur, par exemple une structure, impliquant l'autorisation récursive sur tout le contenu.

On pourra ainsi autoriser un utilisateur à modifier tout un annuaire, ou bien seulement sur une organisation donnée. La possibilité de modifier les autorisations sera bien-sûr réservée aux administrateurs de Dyna.

4.3. L'ordonnanceur de réplication

Toute entrée d'annuaire crée, modifiée ou supprimée doit ensuite être propagée par les réplicateurs dépendant de l'annuaire auquel elle appartient. Chaque réplicateur sait dans quelles conditions il doit traiter une entrée donnée, toutes les entrées d'un annuaire n'étant pas nécessairement concernées par tous les réplicateurs de cet annuaire. L'ordonnanceur de réplication prend en charge les demandes de réplication d'entrées, et gère leur exécution de manière asynchrone par les réplicateurs. Il possède une file d'attente par priorité. La réplication d'une entrée peut entraîner d'autres demandes de réplication : par exemple, lorsqu'une personne change d'organisation, cela peut avoir une influence sur les groupes auxquels elle appartient ou n'appartient plus, et qu'il faut répliquer également. L'ordonnanceur gère l'indisponibilité des annuaires LDAP correspondant aux réplicateurs. Si un annuaire LDAP est momentanément indisponible, la demande de réplication sera soumise à nouveau quelques minutes plus tard, jusqu'à ce que l'opération réussisse ou que le nombre maximum de tentatives soit atteint. Le GUI doit permettre à l'administrateur de visualiser rapidement les réplicateurs produisant des erreurs. Il faudrait prévoir également un mécanisme de remontées d'alertes.

4.4. L'ordonnanceur de traitements

Dans Dyna il existe un nombre important de tâches à effectuer à intervalles réguliers, tant de manière interne, par exemple la limitation de la taille du journal, que de manière visible par les utilisateurs, par exemple gérer l'expiration des boîtes aux lettres de la messagerie. La plupart des traitements sont développés de manière spécifique pour un besoin précis de l'Université. Sans ordonnateur de traitements, le service cron du serveur Dyna s'encombre de nombreux programmes s'exécutant à une heure données, avec le risque de chevauchement. L'ordonnanceur de traitements permet de limiter le nombre de programmes s'exécutant à travers le cron à un seul : l'ordonnanceur lui-même.

L'ordonnanceur de traitements gère :

- l'ordre d'exécution des traitements en fonction de leur priorité ;
- le compte-rendu de leur bonne ou mauvaise exécution. On pourrait éventuellement gérer les remontées d'alertes, par exemple par mail, en cas d'erreurs (ce que Dyna1 ne fait pas) ;
- la dépendance entre les traitements : pour qu'un traitement B puisse s'exécuter, il faut que le traitement A ne soit pas en erreur ;

Un traitement géré par l'ordonnanceur est une classe comparable à une servlet, bénéficiant de tout l'environnement Dyna nécessaire. Elle implémente une méthode executer() renvoyant un code de retour permettant de savoir si le traitement s'est bien déroulé.

L'annexe 7.2 spécifie la liste des traitements actuels et futurs. De même que l'ordonnanceur de réplication peut être autonome, il est possible de séparer l'ordonnanceur de traitements du moteur Dyna2.

4.5. L'envoyeur de messages SMTP

L'envoyeur de messages SMTP est un ordonnanceur permettant d'envoyer des messages de manière

asynchrone, en s'affranchissant de la lourde API JavaMail et de la configuration du service de messagerie. Ce mécanisme fonctionnant de manière satisfaisante dans Dyna1, on peut le reprendre dans Dyna2.

4.6. Le cache des objets Dyna

Afin d'assurer de bonnes performances, il est préférable de garder en mémoire les objets Dyna qui ont été trouvés dans la base de données au lieu de les charger d'une manière répétitive. La gestion du cache utilise le fait que les objets Dyna sont identifiés de manière unique par un code de classe sur une ou plusieurs lettres, et un numéro unique pour leur classe. Lorsqu'un objet est demandé, il est chargé en mémoire depuis la base de données s'il n'est pas présent dans le cache. Il faut éviter l'accumulation d'objets dans le cache. Périodiquement, le cache est purgé des objets anciens. De plus, le JVM est autorisée récupérer de la mémoire sur le cache à travers le mécanisme “SoftReference”. Pour plus de précisions sur cette fonctionnalité de Java, voir la page http://java.sun.com/j2se/1.4.2/docs/api/java/lang/ref/SoftReference.html Ceci permet d'assurer que jamais le système ne se trouvera à court de mémoire à cause du cache. Le cache Dyna n'est pas un cache en écriture et ne gère pas la modification asynchrone de la base de données. Ceci n'est pas indispensable, dans la mesure où un annuaire est une base de données où les lectures sont beaucoup plus nombreuses que les écritures.

4.7. La journalisation

La journalisation est un service permettant de tracer tout l'historique des actions effectuées sur des objets Dyna. Elle permet de savoir, pour un objet donné :

- quelle action a été effectuée : création, modification, suppression ;
- si c'est une modification : quels sont les attributs modifiés, quelles sont les anciennes et les nouvelles valeurs ;
- qui a effectué l'opération : soit c'est un utilisateur authentifié, et on indique son numéro unique, soit c'est un traitement, et on indique son nom ;
- quand l'opération a été effectuée ;
- informations complémentaires.

Ces informations sont stockées dans une base de données distincte du référentiel, les contraintes en termes de taille, de sauvegardes et de restauration étant différentes. Le GUI doit permettre d'effectuer des recherches dans le journal, en fonction des différents critères énumérés ci-dessus, et d'afficher les lignes de journal correspondantes.

5. Services fournis par le serveur web de Dyna

Par analogie avec le service wwsympa du serveur Sympa, sous appellerons ce service wwdyna.

5.1. Utilisateurs du site web Dyna

On a quatre sortes d'utilisateurs se connectant à Dyna :

- les utilisateurs en consultation simple, sans authentification sauf s'ils viennent d'un réseau extérieur à l'Université ;
- les utilisateurs qui veulent changer leur mot de passe, et qui doivent s'authentifier auparavant ;
- les exploitants de Dyna pour une ou plusieurs composantes, habilités pour un ou plusieurs annuaires ;
- les administrateurs qui ont tous les droits.

Les deux premiers constituent une population importante, mais ils n'utilisent qu'une faible proportion des pages de Dyna. On ne peut pas leur imposer un navigateur particulier. Inversement, les deux derniers sont peu nombreux et connus, mais ils ont besoin d'une interface riche et ils doivent afficher un grand nombre de pages différentes. On peut leur imposer un navigateur particulier, par exemple Firefox.

5.2. L'authentification sur le site web Dyna

Dyna2 devra, comme Dyna1, permettre l'authentification de trois manières, au choix de l'utilisateur :

- par LDAP : auquel cas, la configuration de Dyna indique un annuaire possédant au moins un réplicateur de type LDAP et sachant gérer l'authentification. Pour vérifier le couple identifiant - mot de passe fourni par l'utilisateur, Dyna choisit le réplicateur le plus prioritaire satisfaisant ces conditions. La vérification du couple identifiant - mot de passe échoit alors au moteur dyna ;
- par CAS (Centralized Authentication Service) : Dyna demande à CAS d'authentifier un utilisateur lors de la connexion. Il signale également la déconnexion à CAS ;
- par certificats X.509 : Dyna laisse à Apache le soin de vérifier l'authenticité du certificat que lui présente l'utilisateur via HTTPS, par rapport au certificat de l'autorité de certification du CRI. Puis il extrait l'identifiant unique du sujet du certificat.

Pour cela, wwdyna devra disposer d'un mécanisme d'authentification modulaire. L'administrateur décide quels sont les modules d'authentification qui seront disponibles pour les utilisateurs. Ces modules seront configurés individuellement dans un fichier.
Lorsqu'un utilisateur voudra s'authentifier, les modules d'authentification s'exécuteront à tour de rôle, et le premier ayant accepté l'utilisateur arrêtera le processus qur un succès. Si aucun module d'authentification ne donne son accord, l'authentification échoue. Succès ou échec, toute tentative d'authentification doit être journalisée dans un fichier texte avec les informations suivantes :

- date et heure ;
- identifiant de l'utilisateur (mais pas le mot de passe) ;
- modules d'authentification essayés (seulement en mode de journalisation verbeuse) ;
- module d'authentification ayant donné son accord, en cas de succès, ou notification de l'échec dans le cas contraire.

On envisage d'utiliser une librairie de journalisation comme log4php (http://logging.apache.org/log4php/).

5.3. La navigation dans le référentiel

Le service wwdyna devra permettre à tout utilisateur de l'Université de rechercher une entrée selon différents critères : nom, appartenance à un groupe pour une personne, à une organisation, un site, une activité, etc. Le client devra fournir au navigateur un code xhtml valide, avec des feuilles de style CSS2. Celles-ci permettront à la demande de modifier un paramètre simple de la présentation, comme une couleur ou l'espacement entre deux paragraphes, sans modifier le code en PHP. Il faudra tester le bon fonctionnement sur Firefox et Internet Explorer. Pour cela, l'utilisateur doit avoir la possibilité de choisir interactivement un critère “discret” autrement que sous forme de texte, comme avec Dyna1. Exemple : on veut la liste des personnes appartenant à un groupe donné. Dans l'écran de recherche sur les personnes, on clique sur le bouton “groupe” afin de choisir le groupe auxquelles doivent appartenir les personnes sélectionnées. Dyna1 possède pour cela un ensemble de classes appelées “sélecteurs”, permettant de choisir une entrée interactivement. Il faudra effectuer le portage dans l'environnement PHP de wwdyna. L'interface wwdyna devra permettre d'agir, c'est à dire créer, modifier et supprimer suivant les permissions de l'utilisateur, sur chacun des objets Dyna : annuaires, réplicateurs, entrées de chaque catégorie. Pour cela, il disposera de l'API d'accès au référentiel, basée sur XML-RPC. Il disposera donc d'une interface d'un niveau plus élevé que les applications classiques PHP - MySQL qui accèdent directement à une base de données relationnelle. Ceci garantit que toute modification d'une entrée d'annuaire sera propagée dans les réplicateurs d'une manière strictement identique que l'on agisse depuis l'interface wwdyna ou depuis un autre client distant. Pour chaque objet Dyna visualisé ou modifié, wwdyna devra s'assurer auprès du serveur Dyna que l'utilisateur a bien la permission d'accéder à cet objet en lecture ou en modification. Le code qui sera présenté au navigateur client sera formaté en XHTML , au lieu de HTML comme dans Dyna1. Évaluer la possibilité d'introduire les technologies suivantes :

- On Demand Javascript (chargement de fonctions JavaScript à la demande) ;
- AJAX (Asynchronous JAvascript XML : échange dynamique de fragments de code XML de manière à améliorer l'interactivité des pages).

Types de requêtes :

- recherche sur les personnes selon différents critères. Selon de type d'entrée recherchée, on doit pouvoir effectuer la recherche sur différents critère (pas nécessairement tous les attributs). Si le nombre de réponses est important, Dyna2 doit paginer le tableau des résultats ;
- ajout, modification, suppression d'une personne ;
- recherche sur les groupes ;
- ajout, modification, suppression d'un groupe ;
- recherche sur les structures (3 requêtes, une par type de structure) ;
- ajout, modification, suppression d'une structure ;
- recherche générique sur toutes les entrées en fonction de critères communs : le nom, l'état, l'adresse e-mail et l'annuaire d'appartenance ;
- partant de l'écran de détail la fiche principale d'une entrée (personne, groupe ou structure) : accès à tous les dossiers additionnels existants ou qu'il est possible d'ajouter compte tenu du type d'entrée ;
- …

A COMPLETER

6. Services fournis par le client Java Dyna

Le client Java Dyna permet de synchroniser des entrées en mode PUSH. Dyna2 devra prévoir le portage du client Dyna1 actuel, utilisé par Geode, sur l'API basée sur XML-RPC.

7. Migration vers Dyna2

7.1. Migration de la base de données

Il faut naturellement récupérer les données de Dyna1 dans Dyna2, et pour cela disposer d'un programme assurant la migration des données. Ce programme lira la base Dyna1 afin de remplir la base de Dyna2.
Les changements entre Dyna1 et Dyna2 étant substantiels, l'écriture de ce programme de migration et son test représentent une partie non négligeable de la charge de travail du projet.

7.2. Migration des serveurs

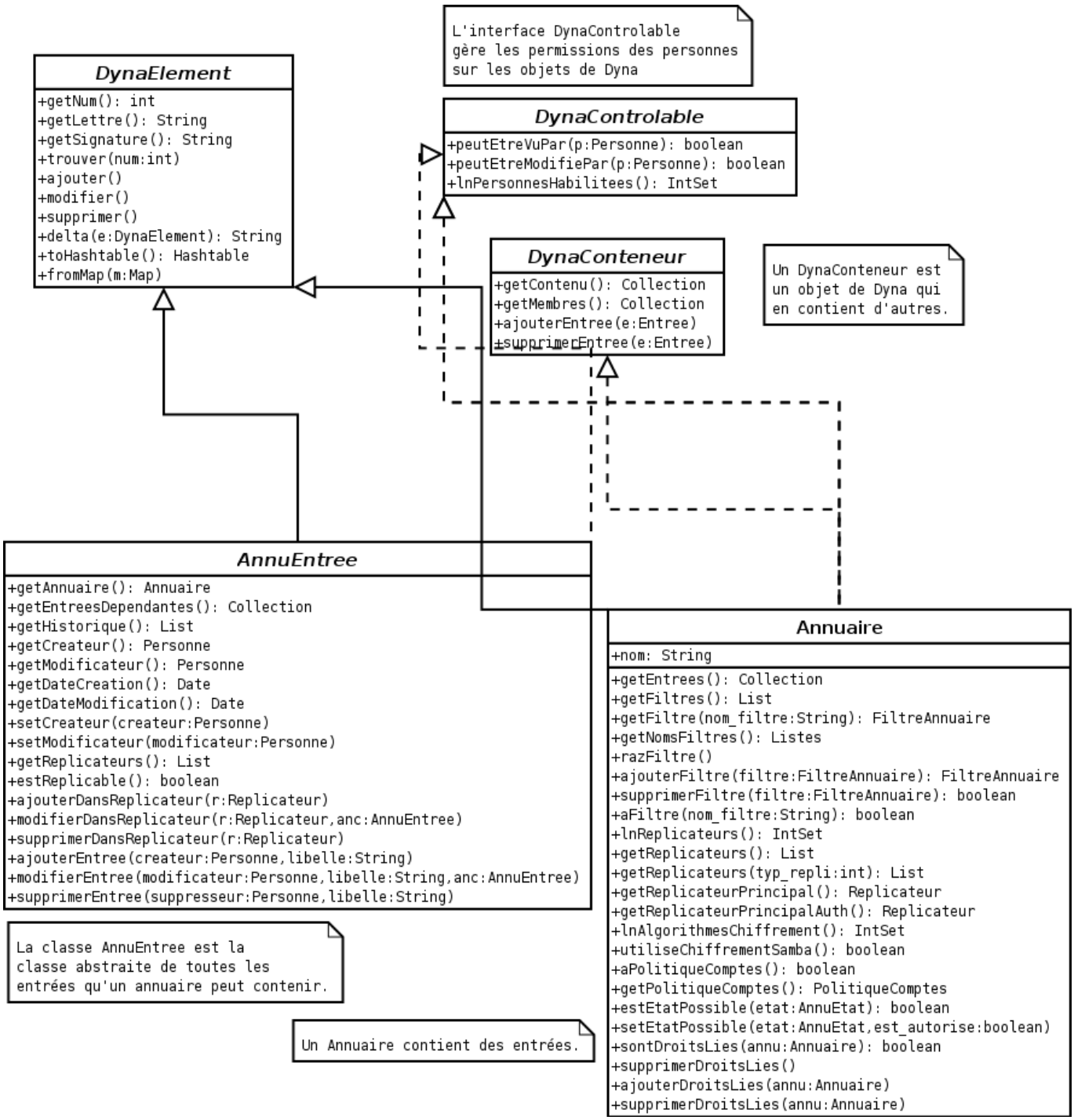
SRU.

8. Annexes

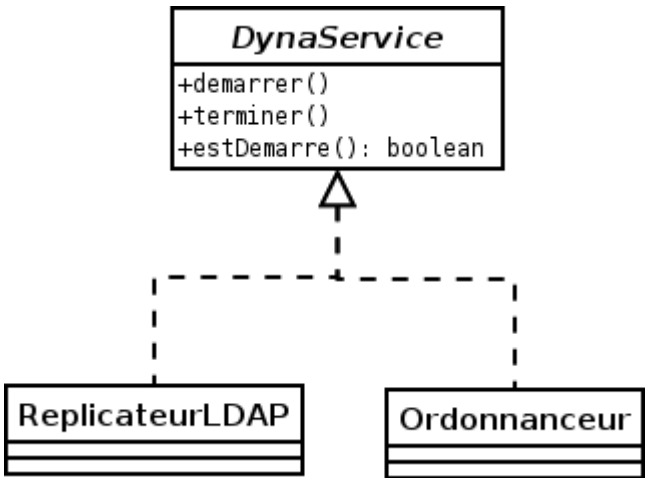
8.1. Diagrammes UML des classes et interfaces

Les paragraphes suivants spécifient les classes de Dyna2 à travers des diagrammes UML.

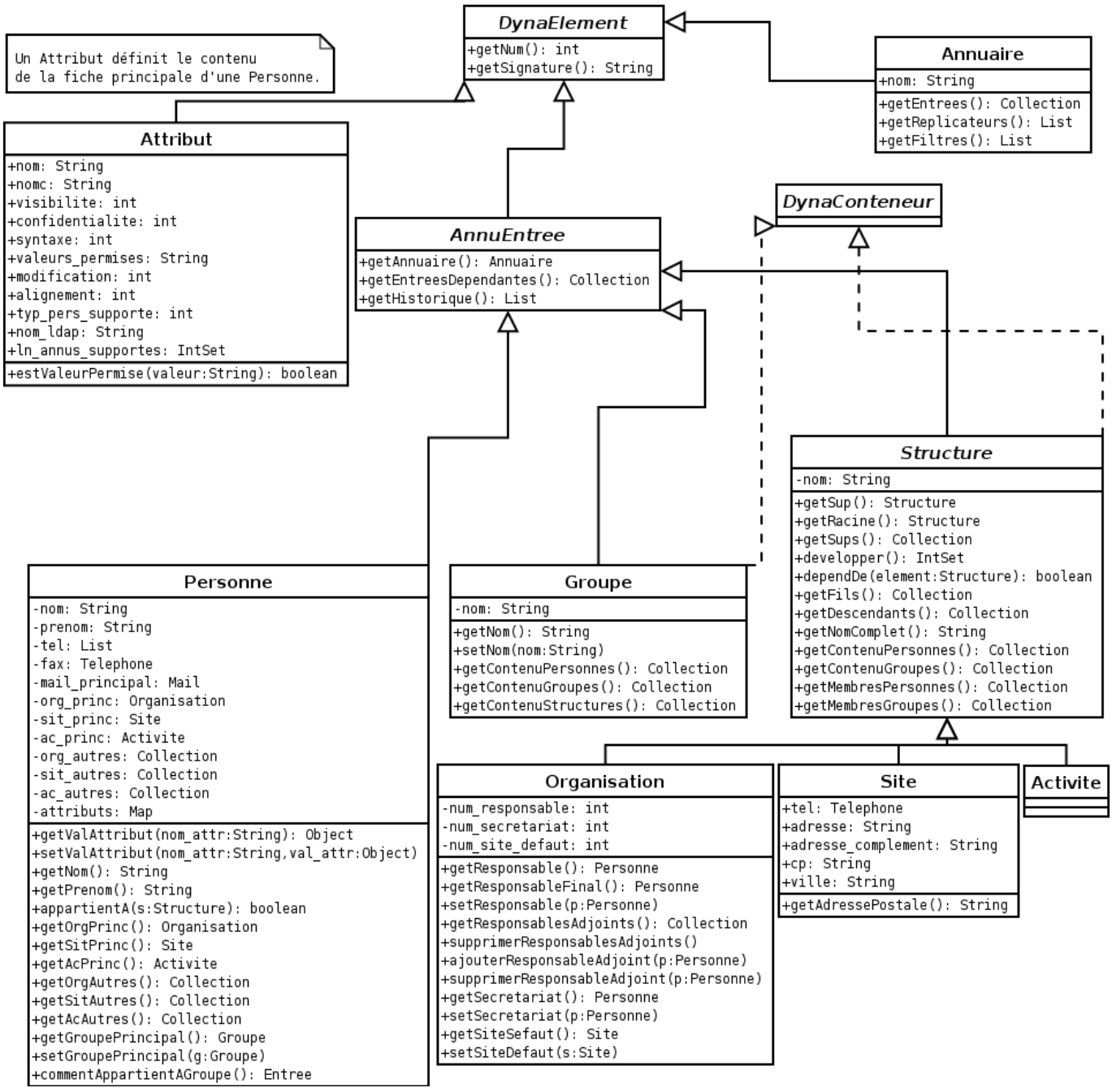
8.1.1. Sommet de la hiérarchie



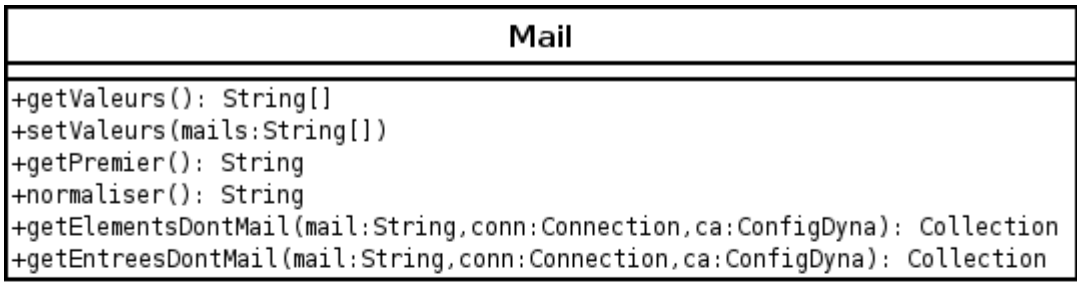
8.1.2. Interface DynaService

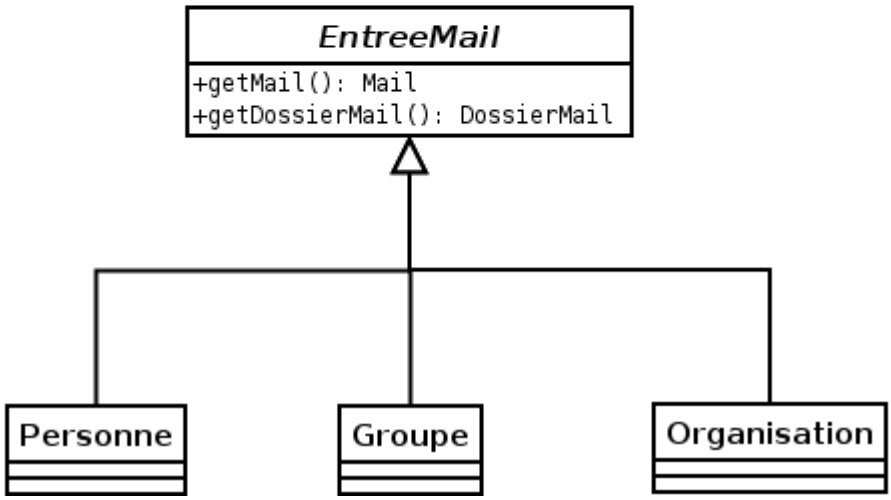


8.1.3. Classes des éléments d'annuaire

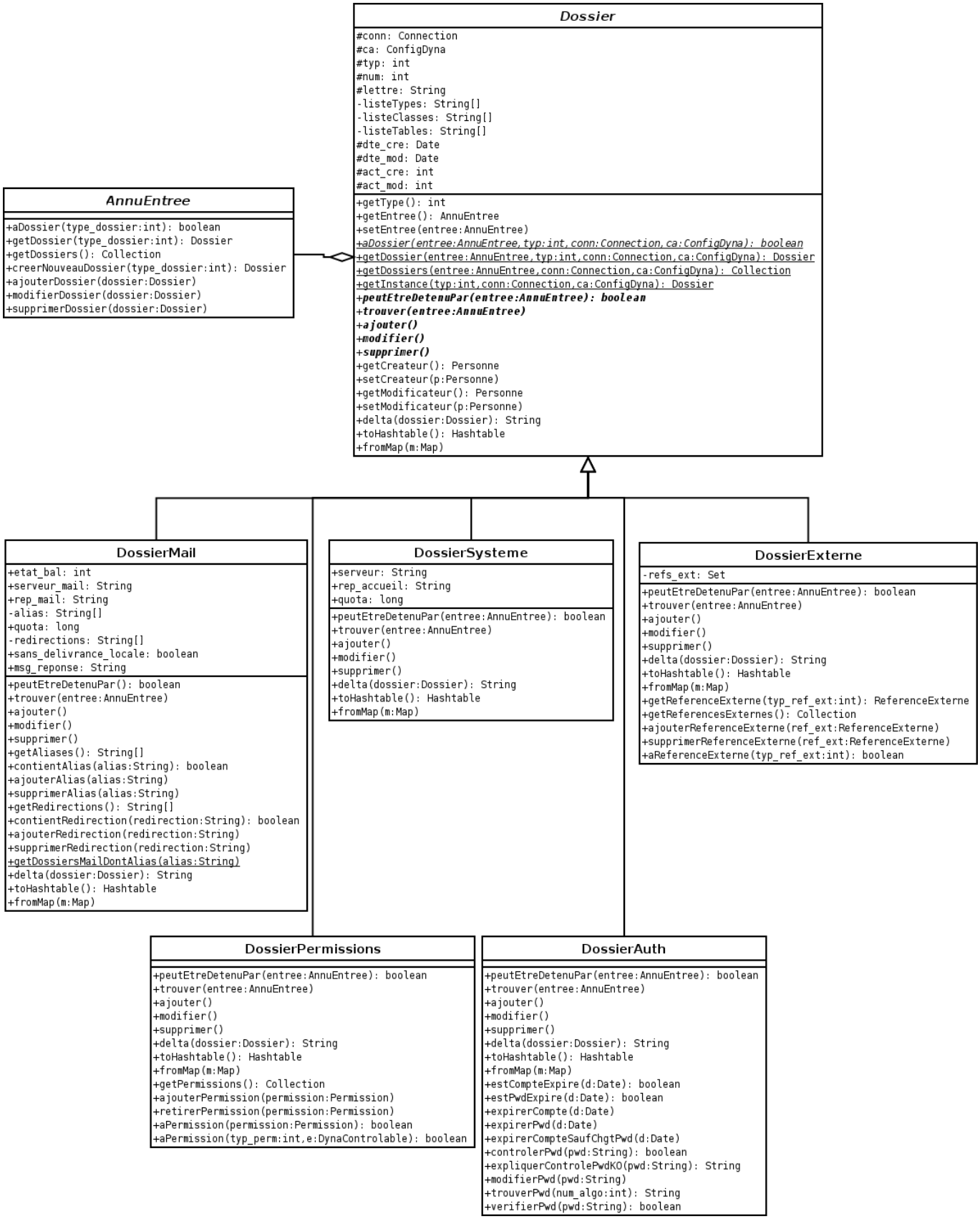


8.1.3. Classe mail et interface EntreeMail

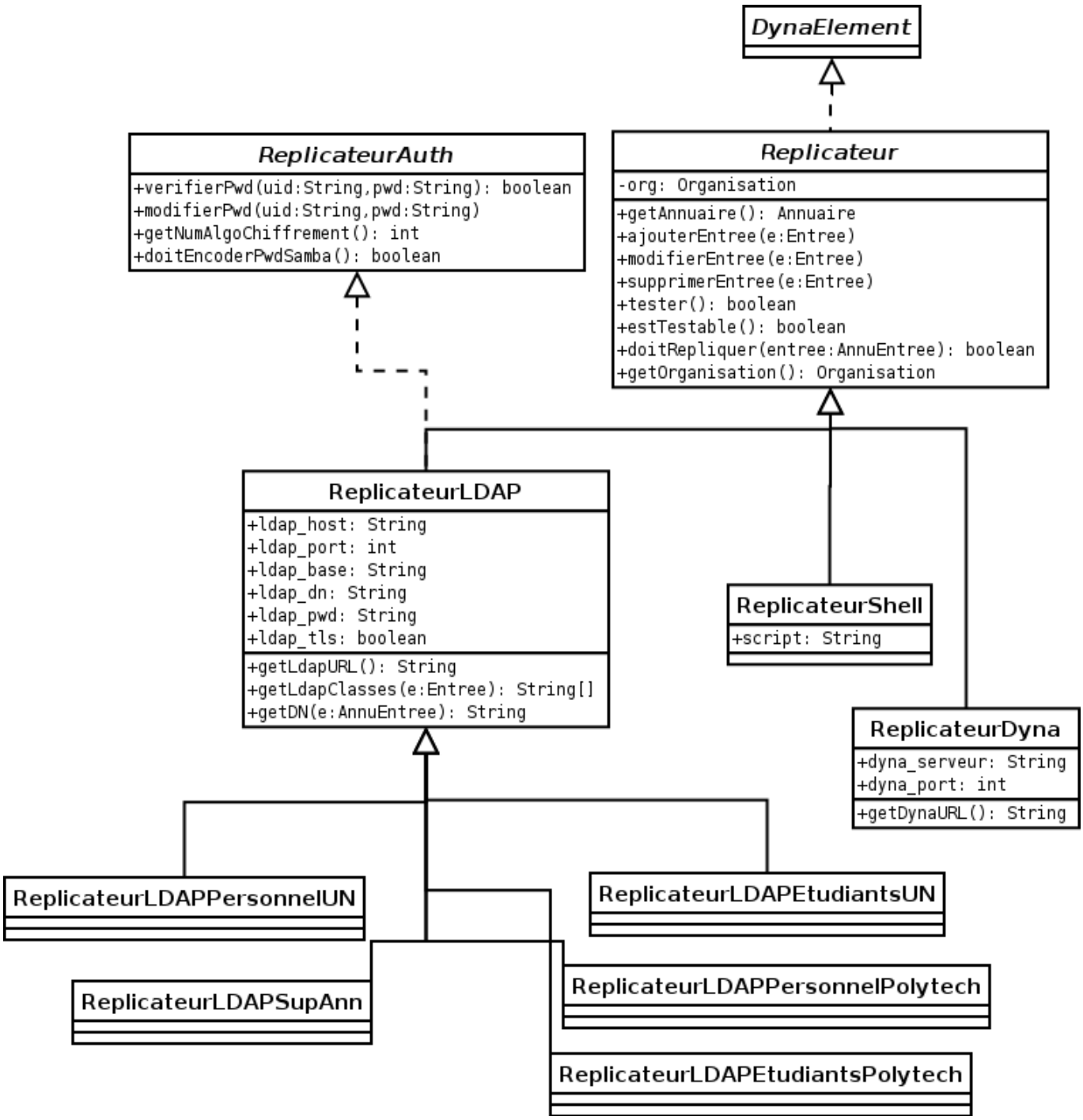




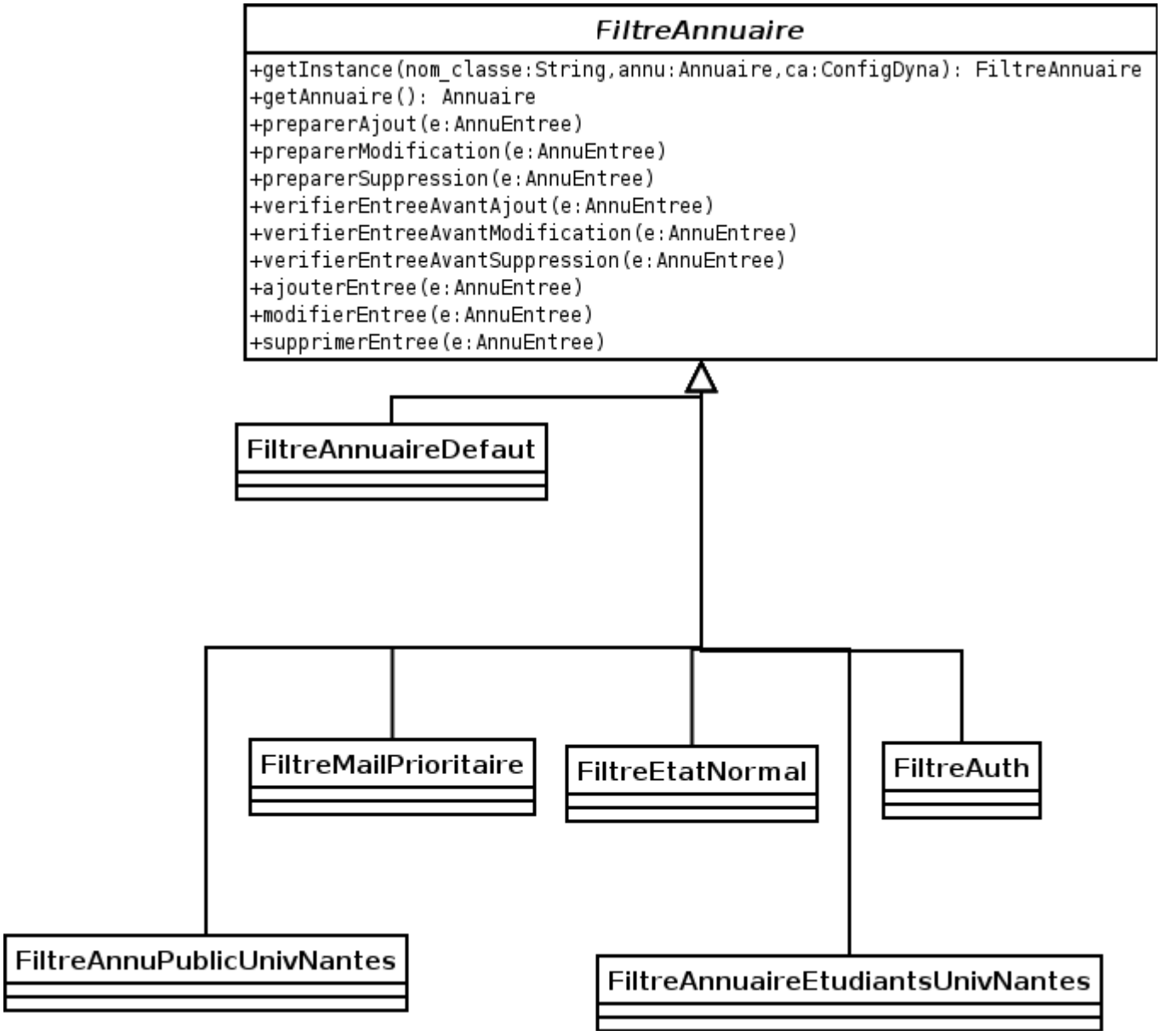
8.1.4. Classes des dossiers additionnels



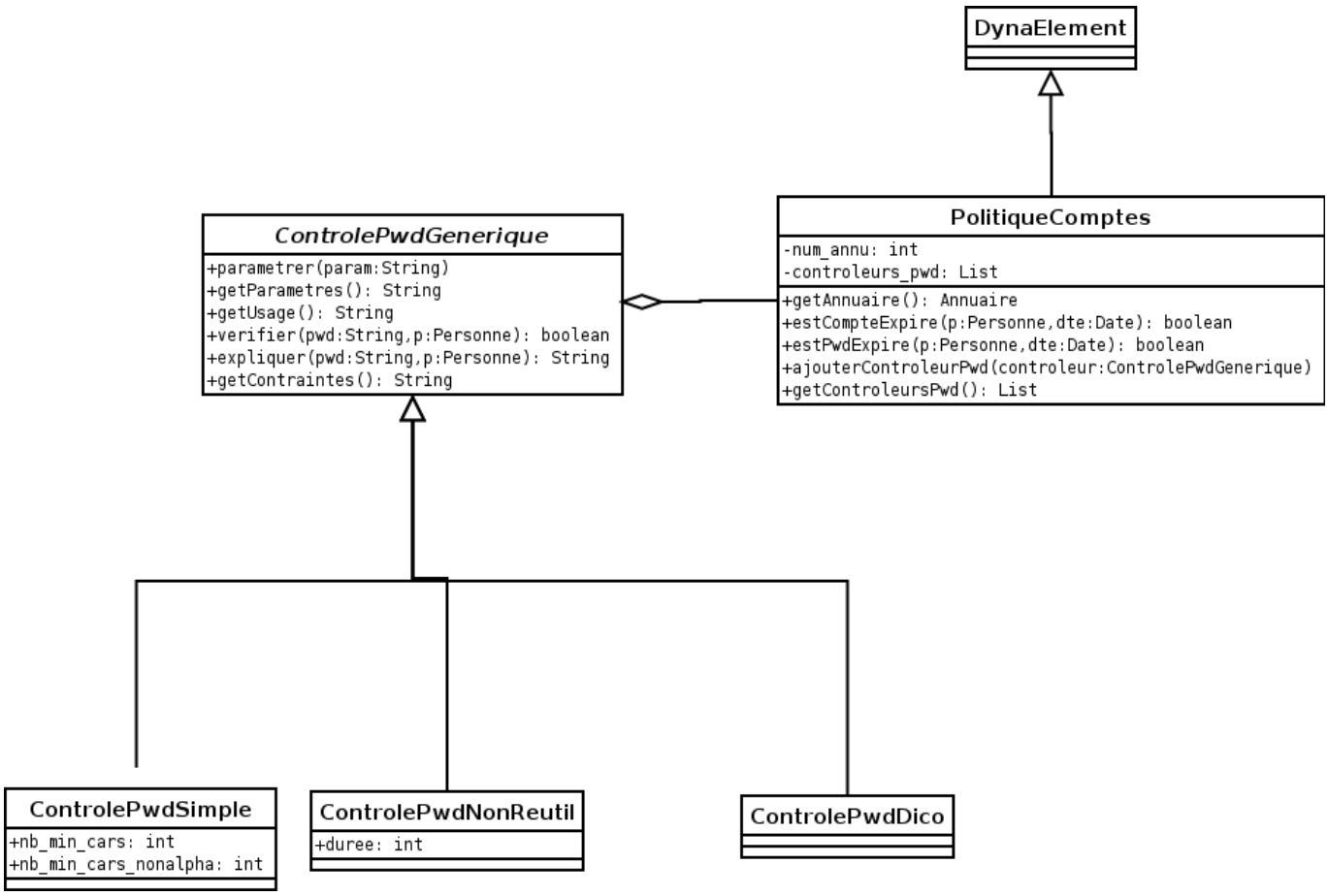
8.1.5. Classes des réplicateurs



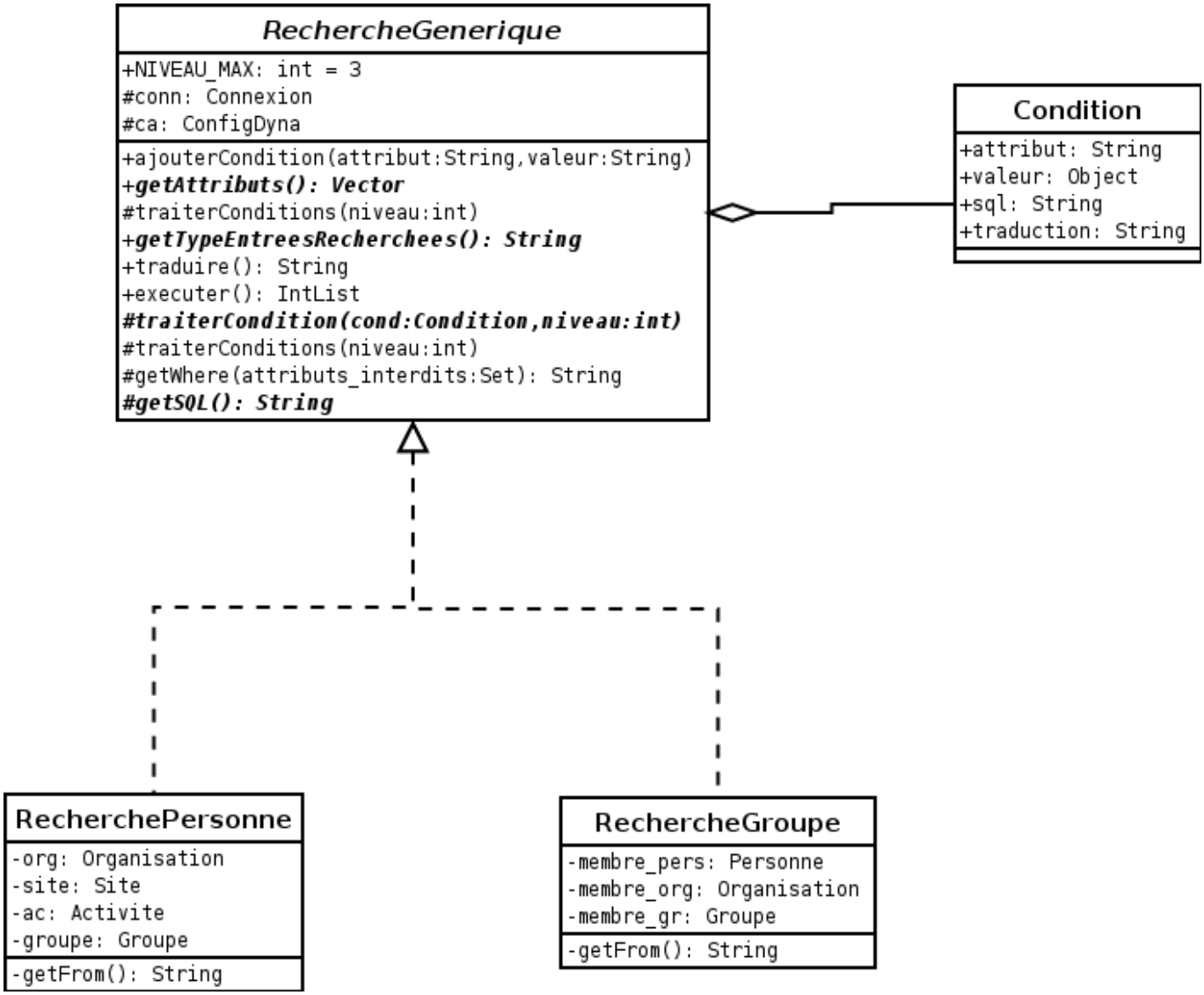
8.1.6. Classes des filtres d'annuaire



8.1.7. Classes des politiques de comptes et des contrôleurs de mots de passe



8.1.8. Classes de recherche dans le référentiel



8.2. Liste des traitements

Les tableaux suivants décrivent les traitements à prendre en compte dans Dyna2.

8.2.1. Traitements présents dans Dyna1

Nom	**Description**	**Exécution**	**A reporter sur Dyna2**
TraitementPersonnesParties	Ce traitement supprime logiquement toutes les personnes qui sont déjà parties et qui ne sont pas déjà logiquement supprimées.	Quotidienne	Oui
TraitementDroitsPersonnesSupprimees	Suppression des droits pour les personnes logiquement supprimées	Quotidienne	Oui
TraitementLimitationTailleJournal	Limitation de la taille du journal	Quotidienne	Oui

Nom	**Description**	**Exécution**	**A reporter sur Dyna2**
ArchivageAppelsFermes	Archivage des groupes liés aux appels anciens	Quotidienne	Non (lié au centre d'appels)
TraitementValidationGroupes	Ce traitement permet de valider tous les groupes.	Quotidienne	Oui
TraitementComptesExpires	Traitement des comptes expirés : la BAL éventuelle passe INACTIVE.	Quotidienne	Oui
TraitementMaintienBAL	Envoi des mails afin de maintenir ou non la BAL pour les retraités, les associations et les doctorants, et traitement de ces BALs.	Quotidienne	Oui
TraitementRappelsPwdCpt	Envoi de mail aux utilisateurs dont le compte et/ou le mot de passe expire	Quotidienne	Oui
TraitementRappels	Ce traitement envoie des messages pour rappeler aux différents intervenant la liste des appels et des services qu'ils ont dans leur portefeuille.	Quotidienne	Non (lié au centre d'appels)
TraitementMajHisto	Mise à jour de l'historique des consultations et des modifications de l'annuaire. A supprimer ? Les informations contenues dans le journal peuvent suffire.	Quotidienne	Oui
TraitementKSUPPersonnelUnivNantes2	Import dans un fichier XML des données de KSUP pour le personnel de l'Université. On ne s'intéresse qu'aux attributs suivants : téléphone, télécopie, bureau.	Quotidienne	Oui
TraitementImport2 (ksup.xml)	Import du fichier XML en provenance de KSUP (personnel de l'Université). Ce traitement est dépendant de TraitementKSUPPersonnelUnivNantes2	Quotidienne	Oui
TraitementKSUPResponsablesStructures	Import des responsables de structures dans KSUP	Quotidienne	Oui
TraitementRappelQuotas	Envoi d'un message aux utilisateurs sont le quota est utilisé à plus de 80%	Quotidienne	Oui
TraitementHarpegeStructures	Import dans un fichier XML des données de HARPEGE pour les structures de l'Université.	Quotidienne	Oui
TraitementHarpegePersonnelUnivNantes	Import dans un fichier XML des données de HARPEGE pour le personnel de l'Université.	Quotidienne	Oui
TraitementImport2 (harpege_structures.xml)	Import du fichier contenant les structures de l'Université importées depuis HARPEGE. Ce traitement est dépendant de TraitementHarpegeStructures	Quotidienne	Oui

Nom	Description	Exécution	A reporter sur Dyna2
TraitementImport2 (harpege.xml)	Import du fichier contenant les personnels de l'Université importées depuis HARPEGE. Ce traitement est dépendant de TraitementHarpegePersonnelUnivNantes	Quotidienne	Oui
TraitementSympaEtu	Mise à jour de la base mysql de sympa pour les étudiants de l'Université.	Quotidienne	Oui
TraitementSympaPers	Mise à jour de la base mysql de sympa pour le personnel de l'Université.	Quotidienne	Oui

8.2.2. Nouveaux traitements pour Dyna2

Nom	Description	Exécution
TraitementInfosPersonnesParties	Information des exploitants de Dyna sur les personnes dont la date de départ approche. Demandé par Frédéric Lussori et Nicolas Costes	Quotidienne
TraitementStatsReplicateur	Informations des administrateurs de Dyna concernant les statistiques des réplicateurs : nombre total de demandes de réplications, nombre de demandes de réplication ayant échoué, etc.	?

8.3. Algorithme de réplication

Ce chapitre détaille l'algorithme à utiliser lors de la réplication d'entrées LDAP. En effet, il convient d'éviter d'être naïf en croyant que si une entrée est à ajouter, alors elle n'existe pas dans l'annuaire LDAP correspondant au réplicateur, et que si elle est à modifier ou à supprimer, cette entrée existe déjà et qu'elle est située à l'emplacement attendu du DIT. Rien n'est moins sûr, car Dyna peut prendre en charge un annuaire LDAP déjà existant ou bien des entrées ont pu être crées, modifiées ou supprimées à l'insu de Dyna.

8.3.1. Algorithme préliminaire : calcul du DN

```
String getDN(AnnuEntree e) :
   Si e est une organisation :
      Si le DIT est hiérarchique :
         Si le niveau de l'organisation est 1 :
            Résultat : "ou="+nom de l'organisation+","+base du DIT
         Sinon :
            Résultat : "ou="+nom de l'organisation+","+getDN(organisation de
niveau supérieur)
         Fin si
      Sinon :
         Résultat : "ou="+nom de l'organisation+","+base des organisations
dans le DIT
      Fin si
   Sinon si e est une personne :
```

```
      Si le DIT est hiérarchique pour les personnes et que la personne
appartient à au moins une organisation :
         Récupérer une organisation "org" parmi celles auxquelles appartient
la personnes et que ce réplicateur prend en charge,
            l'organisation principale étant prioritaire
         Résultat : "uid="+uid de la personne+","+getDN(org)
      Sinon :
         Résultat : "uid="+uid de la personne+","+base des personnes dans le
DIT
      Fin si
   Sinon si e est un groupe :
      Si le DIT est hiérarchique pour les groupes et que le groupe
appartient à une organisation :
         Résultat : "cn="+nom du groupe +","+getDN(organisation auquel
appartient le groupe)
      Sinon :
         Résultat : "cn="+nom du groupe+","+base des groupes dans le DIT
      Fin si
   Fin si
Fin.
```

Remarque : ceci est le code par défaut, les réplicateurs spécifiques pouvant surcharger cette méthode.

8.3.2. Réplication d'une entrée

```
void ajouterOuModifier(AnnuEntree e) :
   Calculer DN = getDN(e)
   Récupérer dans LDAP DN1 = l'entrée qui a ce DN, par un search en mode
scope=base
   Récupérer dans LDAP DN2 = l'entrée qui a un ou plusieurs attributs
déterminant pour l'entrée e
      (uid pour les personnes, cn pour les groupes...)
   Si DN1 existe et ((DN2 existe et DN1=DN2) ou (DN2 n'existe pas)) :
      On modifie l'entrée e :
         Supprimer les attributs de l'entrée correspondant à DN1 qui
n'appartiennent pas aux attributs calculés pour l'entrée e
            (ces attributs risquent d'entrer en conflit avec la classes de
l'entrée e)
         Ajouter ou modifier les autres attributs
   Sinon si DN1 n'existe pas et que DN2 n'existe pas :
      On ajoute l'entrée e :
         On vérifie d'abord que le DIT peut accueillir cette entrée :
            Calculer DN3 : DN de l'entrée de niveau supérieur à l'entrée e
            Si DN3 existe :
               OK, passer à l'étape suivante
            Sinon si le DIT est hiérarchique pour le type d'entrée de e :
               Organisation org = organisation à laquelle appartient
l'entrée et que le réplicateur prend en charge
```

```
               ajouterOuModifier(org)
            Sinon :
               Réplication impossible pour l'entrée e
            Fin si
         Fin de la vérification
      Ajouter l'entrée e dans LDAP
   Sinon si DN1 n'existe pas et DN2 existe :
      On renomme l'entrée e :
         Si l'entrée e est une organisation et que le DIT est hiérarchique
pour au moins un type d'entrée :
            Créer l'entrée DN1
            ajouterOuModifier sur toutes les sous-organisations directes de
e
            ajouterOuModifier sur toutes les personnes dont au moins une
organisation (principale ou secondaire) est directement e
            ajouterOuModifier sur tous les groupes appartenant directement à
e
            Supprimer l'entrée DN2
         Sinon (entrée non hiérarchique) :
            renommage LDAP de l'entrée e
            Si e est une personne :
               Pour tout groupe g contenant directement ou non cette
personne, et concerné par ce réplicateur :
                   ajouterOuSupprimer(g)
               Fin pour
             Fin si
         Fin si
   Sinon si DN1 et DN2 existent et DN1 != DN2
      Supprimer DN2
      Vérifier que DN2 a bien été supprimé (attention à la boucle infinie !)
      ajouterOuModifier(e)
   Fin si
Fin

void supprimer(AnnuEntree e) :
   Calculer DN = getDN(e)
   Récupérer dans LDAP DN1 = l'entrée qui a ce DN, par un search en mode
scope=base
   Récupérer dans LDAP DN2 = l'entrée qui a un ou plusieurs attributs
déterminant pour l'entrée e
   Si DN1 existe :
      Supprimer l'entrée LDAP DN1
   Fin si
   Si DN2 existe :
      Supprimer l'entrée LDAP DN2
   Fin si
Fin
```