

# Y a-t-il une place pour opensocial dans l'enseignement supérieur ?

David Verdin  
RENATER  
Centre de Ressources Informatiques  
263 Avenue du General Leclerc  
CS 74205  
Campus de Beaulieu  
35042 Rennes CEDEX

## Résumé

*Monsieur Le Bellec, professeur d'université à Angers est spécialiste des champignons vénéneux. Il a mis en place un excellent catalogue en ligne des espèces recensées. Monsieur Kovacik, chercheur à Varsovie, anime un groupe de chercheurs européens sur les champignons dont Monsieur Le Bellec fait partie. Il se sert pour cela de la plate-forme de groupware déployée en Pologne sur laquelle sont inscrits de nombreux chercheurs européens. Ils décident de permettre à cette communauté de contribuer au catalogue de monsieur Le Bellec. Se posent alors de nombreuses questions parmi lesquelles : Comment gérer les autorisations d'accès aux données déposées par chaque chercheur ? Comment permettre l'utilisation des données du catalogue dans le wiki collaboratif ?*

*À cette problématique, devenue si courante dans les milieux scientifiques, la technologie opensocial propose des réponses. Développée dans le but de permettre l'interaction entre applications hétérogènes et indépendantes, elle offre notamment des mécanismes simples et sécurisés de transfert de données fondés sur le consentement de l'utilisateur*

*L'objectif de cet article est de présenter un tour d'horizon de la technologie opensocial et du protocole Oauth sur laquelle elle s'appuie. Nous explorerons ensuite des pistes d'applications dans le domaine de l'enseignement supérieur et de la recherche, notamment à travers l'exemple des implémentations réalisées par RENATER dans le cadre d'organisations virtuelles trans-fédérations.*

## Mots clés

opensocial, gadget, conteneur, agrégateur, portlet, interopérabilité, portabilité

## 1 Introduction

Sortie des eaux depuis 2007, la spécification opensocial a maintenant un pied bien établi en terre numérique. Adoptée par des acteurs aussi importants que Google, Yahoo ou Myspace[1], elle offre un nouveau mode de personnalisation des applications web par l'emploi de gadgets, petites applications portables d'un réseau social à l'autre.

Au-delà de la mode technologique, opensocial offre de réelles nouveautés en termes d'intégration logicielle. La question, pour les établissements d'enseignement supérieur et de recherche, est de savoir si cette technologie a quelque chose à nous apporter. L'objet de cet article est de proposer une synthèse sur opensocial pour aider le lecteur à se forger son opinion.

Après une présentation générale de la technologie, nous expliquerons les bases de son fonctionnement fondé sur le couple gadget / conteneur. Une troisième section détaillera le modèle de données exploité par opensocial et ses points communs avec Supann. Enfin, l'API serveur d'opensocial sera présentée, ainsi que son application dans le logiciel Sympa.

## 2 Présentation d'opensocial

Opensocial définit un ensemble d'API pour permettre de déployer des applications sociales sur plusieurs sites web. Ces API permettent aux développeurs de créer des applications accédant aux données des utilisateurs sur des réseaux sociaux.

## 2.1 Les problèmes à résoudre

Lorsqu'un nouveau site web est développé, en règle générale son code est spécialisé autour des services que le site va rendre et rarement réutilisable tel quel pour un autre site ; les données manipulées, en particulier, répondent à une sémantique non transposable à un autre contexte.

Avec l'explosion des réseaux sociaux, un intérêt croissant s'est porté vers le développement de fonctionnalités optionnelles, utilisées par une partie seulement des utilisateurs. Intégrer de telles fonctionnalités dans un site web implique que leur code soit compatible avec celui du site où l'on cherche à les intégrer ; en outre, les données des utilisateurs doivent être accessibles. Enfin, une fois la fonctionnalité développée, il est souhaitable de pouvoir appliquer directement le programme dans d'autres sites sans devoir le transformer.

Pour atteindre cet objectif, on disposait déjà de la technologie de portails, notamment dans le monde java. Cela dit, développer un site web pouvant supporter cette technologie ne s'improvise pas : le site est architecturé pour être un portail ou non. Ainsi, on peut développer un portail en suivant la spécification JSR 168[2]. Cette spécification définit les termes d'interopérabilité entre un portail compatible J2EE et une application web prévue pour être incluse dans ce portail, appelée portlet. Cette application doit être obligatoirement développée en Java. Il est donc virtuellement impossible de convertir un site écrit par exemple en PHP en portail compatible avec la JSR 168, puisque les portlets sont exécutés côté serveur. Par ailleurs, les technologies portail ne laissent pas le choix du langage de base dans lequel l'application et les widgets sont écrits.

L'objectif d'opensocial est d'offrir une solution pour contourner ces difficultés.

## 2.2 Solution apportée

Puisqu'il est délicat d'intégrer de nouvelles fonctionnalités côté serveur sans toucher au cœur de l'application, une solution est de faire exécuter les fonctionnalités optionnelles côté client. La documentation opensocial présente une introduction pas à pas d'une telle technologie.[3]

Un site de réseau social est considéré comme un conteneur d'application, nommé **conteneur**. Il va également contrôler le backend de persistance des données. Chaque fonctionnalité est appelée un **gadget**. Ces gadgets sont des fragments XML, contenant optionnellement du HTML. Ces fragments sont inclus dans des Iframes au sein de l'interface utilisateur du conteneur. Le rendu final de la fonctionnalité est assuré par l'appel d'un ensemble de bibliothèques JavaScript. Par ailleurs, il est possible que le gadget fasse référence à un script tiers, éventuellement situé sur un autre serveur, exécuté lors du rendu du gadget. Il en résulte que le contenu des gadgets peut être écrit en n'importe quel langage exploitable sur le web. L'utilisation des fonctions de l'API JavaScript est cependant obligatoire pour accéder aux données du serveur.

Enfin, il est possible à une application complètement extérieure à un réseau social (non implémentée sous forme de gadget) d'accéder aux données du réseau sous certaines conditions qui seront détaillées plus loin.

Tous les échanges entre une application et le réseau social sont sécurisés par l'emploi du protocole OAuth.

## 2.3 Historique

Opensocial est né des efforts conjoints de nombreux opérateurs de réseaux sociaux tels que Google, Yahoo, Myspace et Viadeo. La première version publique (0.5) a été diffusée fin 2007. Elle a été suivie rapidement d'autres versions permettant de combler les failles de sécurité présentes dans les premières versions.

La dernière version de la spécification opensocial (2.0) a été publiée en août 2011. Elle est composée des dix documents suivants formant un total de presque 300 pages.[4]

- ⤴ *OpenSocial Specification* : terminologie générale, conditions générales de conformité, règles pour étendre la spécification ;
- ⤴ *Core Container Spec* : définition générale d'un conteneur, règles de conformité, API JavaScript, configuration, recommandations de fonctionnement ;
- ⤴ *Core Gadget Spec* : définition générale d'un gadget, syntaxe XML, règles de fonctionnement, API Javascript et règle de gestion des données récupérées des conteneurs ;
- ⤴ *Core API Server Spec* : définition des APIS REST et XML/RPC mettant à disposition les données d'un conteneur à d'autres applications web ;
- ⤴ *Core Data Spec* : types de base décrivant les attributs opensocial et représentation des données aux formats JSON et XML ;
- ⤴ *Social Gadget Spec* : compléments sur l'API Javascript des gadgets ;
- ⤴ *Social API Server Spec* : description de l'accès aux données d'un conteneur par l'interface REST ou XML/RPC ;
- ⤴ *Social Data spec* : description des attributs utilisateur opensocial ;
- ⤴ *OpenSocial Templating Spec* : spécification d'un métalangage propre à opensocial permettant d'écrire des gadgets sans

employer le Javascript ;

- ▲ *OpenSocial WAP Extension* : description de la mise en œuvre d'opensocial pour des plates-formes ne supportant pas Javascript et les Iframes.

Un ouvrage couvrant toute la technologie opensocial a été publié en septembre 2011 par l'éditeur O'reilly[5]

### 3 Essence du fonctionnement

Comme nous l'avons évoqué dans la première section, la base d'opensocial est l'intégration de gadgets au format XML dans un conteneur. Cette section donne des détails sur l'implémentation des gadgets, des conteneurs et de leurs interactions.

#### 3.1 Structure d'un gadget

Un gadget est un simple fragment XML qui sera intégré dans un Iframe du conteneur. L'exemple ci-dessous est le « Hello world » des gadgets :

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
  <ModulePrefs title="Hello World!">
    <Require feature="opensocial-2.0" />
  </ModulePrefs>
  <Content type="html">
    <![CDATA[
      Hello, world!
    ]]>
  </Content>
</Module>
```

Une fois intégré dans un Iframe, ce fragment XML affichera simplement le texte « Hello world ! ». L'ensemble des styles appliqués à ce texte sont gérés par le conteneur.

En pratique, à la place de cet affichage simple, les développeurs de gadgets vont appeler des fonctions Javascript qui accéderont aux données du conteneur et généreront le contenu. L'API Javascript n'est pas détaillée dans cet article. Pour avoir une idée des données disponibles, on peut consulter la section 3.

Un point important à noter est que le développeur n'est pas obligé d'inclure le code de son gadget dans le fragment XML. Il peut simplement spécifier l'URL qui renverra le contenu HTML, comme dans l'exemple ci-dessous.

```
<?xml version="1.0" encoding="UTF-8"?>
<Module>
  <ModulePrefs title="Proxied Content">
    <Require feature="opensocial-0.9" />
  </ModulePrefs>
  <Content href="http://www.example.com" view="canvas">
  </Content>
</Module>
```

L'intérêt de cette fonctionnalité est multiple. D'une part, le code du gadget peut être mis à jour simultanément pour toutes ses installations par simple mise à jour d'un serveur partagé. D'autre part, il est possible d'utiliser tout langage sur le serveur distant pour générer le HTML renvoyé. L'inconvénient majeur est d'assujettir le bon fonctionnement d'un gadget à la disponibilité d'un serveur tiers.

#### 3.2 Implémentation des conteneurs

Si une grande latitude est laissée pour écrire des gadgets, en revanche l'implémentation de référence des conteneurs opensocial est le serveur Apache Shindig. Ce logiciel est complètement conforme à la spécification opensocial et permet de déployer, d'après ses auteurs, un conteneur opensocial « en moins d'une heure ».

### 3.3 Relation gadget / serveur tiers : intervention d'Oauth

Il est possible pour un gadget de converser avec un autre serveur que celui du conteneur. Cela permet à une application web de récupérer et centraliser des données d'un utilisateur sur un réseau social, et au gadget, de transférer des données métier vers l'application distante.

Ce mécanisme implique un échange sécurisé entre les deux serveurs. Cette sécurisation est assurée par le biais d'une requête Oauth[6] à deux membres, ou « 2-legged Oauth ». L'expression « n-legged Oauth » définit un type de transfert de données depuis une application jusqu'à son utilisateur à l'aide du nombre de tiers impliqués (en plus de l'utilisateur humain). Dans une requête « 2-legged », par exemple, on peut avoir le client web de l'utilisateur et le serveur web. Dans le cas évoqué dans cette section, les deux tiers sont deux serveurs : le serveur tiers et le serveur du réseau social.

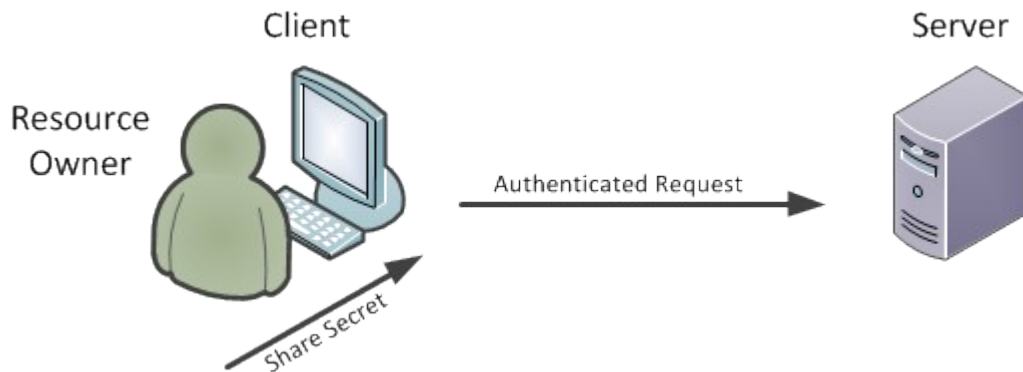


Figure 1: 2-legged Oauth

Ce protocole n'étant employé que dans le cas des gadgets, nous ne le détaillerons pas pour mieux nous concentrer sur le 3-legged Oauth présenté dans la section 4.2.

### 3.4 Brève comparaison avec UPortal

Notre faible expertise d'UPortal nous empêche de trancher autoritairement sur les différences entre ce système et opensocial. Cela dit, certaines différences apparaissent nettement :

- ⤴ si UPortal est entièrement fondé sur la JSR 168 et implique donc le langage java, opensocial permet d'implémenter un gadget dans n'importe quel langage utilisable sur le web ;
- ⤴ l'essentiel du travail d'interprétation du code des gadgets opensocial est effectué côté client, ce qui allège la charge du serveur ;
- ⤴ si UPortal permet d'agrégier des applications au sein d'un système d'information cohérent, opensocial offre la possibilité d'interagir avec des applications développées complètement indépendamment moyennant des aménagements légers.

Au final, les deux technologies permettent de constituer un « mashup » d'applications, mais en se fondant sur des paradigmes radicalement divergents. Cela dit, cette différence de vue est également une richesse, les rendant complémentaires sur certains points. Ainsi, rien n'empêche qu'en implémentant un connecteur opensocial au niveau d'UPortal, on puisse faciliter l'interaction d'un ENT avec des applications externes au système d'information de l'établissement. De cette manière, on pourrait envisager des débuts de connexion avec des réseaux sociaux (gérant par exemple des réseaux d'anciens élèves) ou professionnels (facilitant l'insertion précoce des étudiants dans le milieu professionnel).

À l'heure actuelle (fin 2011), aucune implémentation n'existe pour UPortal permettant d'intégrer la technologie opensocial. Un projet est en cours au sein d'Uportal, mais n'a pas encore abouti.[7]

## 4 Les données utilisateur

### 4.1 Les types principaux de données

Seuls cinq types de données[8] sont directement accessibles à partir de l'API opensocial (que ce soit javascript, REST ou XML RPC) :

- ⤴ Activity streams : relevé d'activité d'une personne (Exemple : « Jean-Yann a ajouté Brigitte à sa liste d'amis »)
- ⤴ Appdata : des données personnalisées créées et exploitées par les widgets. Les noms et les types des données de chaque application cliente étant non prédictibles, une entrée Appdata a deux champs : le nom et la valeur de la variable.
- ⤴ Group : un groupe possédé par un utilisateur. Par exemple : la liste de ses amis.
- ⤴ Message : description d'un message envoyé par un utilisateur à un autre.
- ⤴ Person : la description d'une personne.

Ces cinq données principales sont toutes composées de plusieurs champs. Ils sont soit de type simple (chaîne de caractère, date, entier...) ou de type secondaire, décrivant une adresse postale, une organisation, etc.

L'orientation réseau social apparaît dans le choix des champs décrivant des personnes. On y trouve des descriptions classiques (genre, nom, adresses email) mais également des informations qui s'intégreraient fort mal dans un système d'information d'établissement : préférences sentimentales, humeur, etc.

À ce titre, le partage de données sur les réseaux sociaux dépend des conditions générales du conteneur. Dans iGoogle, par exemple, il est précisé que les gadgets doivent se limiter à exploiter des données raisonnablement éloignées de la vie privée.

### 4.2 Comparaison avec Supann

Supann[9] est prévu pour représenter une personne dans son contexte actuel – ou éventuellement une succession d'états de cette personne – ce qui inclut la description de structures et de groupes. Le contexte des réseaux sociaux est différent : il s'agit de représenter l'ensemble des événements marquants d'une personne, l'organisation de rattachement ayant un rôle moins central. Dès lors, de nombreux champs de la spécification opensocial peuvent être multivalués pour de nombreuses raisons : non distinction des mails institutionnels et personnels, établissement d'un ensemble d'organismes ayant ou ayant eu des liens avec la personne, etc. Les attributs opensocial introduisent un flou là où les attributs Supann cherchent la précision.

De nombreux attributs Supann n'ont pas d'équivalent dans opensocial. Cela dit, dans de nombreux cas, il s'agit d'informations qui sont de toutes façons du ressort exclusif de l'établissement : unité de rattachement, grade, etc. Au contraire, les informations communes aux deux systèmes peuvent très bien être récupérées à partir d'un réseau social : ensemble d'emails de contact, nom, situation maritale, etc.

Une comparaison exhaustive des attributs Supann et opensocial est proposée en annexe de cet article.

## 5 Un cran plus loin : l'API serveur

### 5.1 Le problème à résoudre

Les APIs opensocial permettent d'inclure des gadgets dans un conteneur. Cela dit, tous les sites web ne sont pas, loin s'en faut, prévus pour être des conteneurs opensocial. Et par ailleurs, tous ne se prêtent pas à l'installation d'un gadget sur un conteneur, comme dans le cas du scénario présenté à la section 2.3. Enfin, l'échange de données de type opensocial peut être souhaitable entre deux sites web qui ne sont pas des réseaux sociaux. Pour cela, opensocial propose une API serveur[10], permettant à deux applications d'utiliser les mécanismes opensocial pour s'échanger des données.

### 5.2 Le workflow « 3-legged Oauth »

Quand deux serveurs s'échangent des données, il est important que l'utilisateur donne préalablement son accord. Dans le cas des gadgets, l'installation du gadget dans le conteneur par l'utilisateur tient en général lieu d'accord préalable ; voilà pourquoi on se contente du protocole 2-legged Oauth, où l'utilisateur n'intervient pas.

Dans cette section, on désigne par :

- ⤴ utilisateur : la personne physique utilisant les services,
- ⤴ client : l'application ayant besoin de récupérer des données sur une autre application au nom de l'utilisateur,
- ⤴ fournisseur de ressource : l'application hébergeant les données que le client va chercher à récupérer.

En résumé, dans ce protocole, l'utilisateur demande à l'application cliente d'aller récupérer des données pour lui auprès du fournisseur de ressources. Un jeu d'échanges entre les deux serveurs a alors lieu, qui se conclut par une demande de confirmation, de la part du fournisseur de ressources auprès de l'utilisateur, que l'application cliente a le droit d'accéder à ses données.

Une fois cet accord donné, l'application cliente peut récupérer des données auprès du fournisseur de ressources comme le ferait un gadget auprès d'un conteneur.

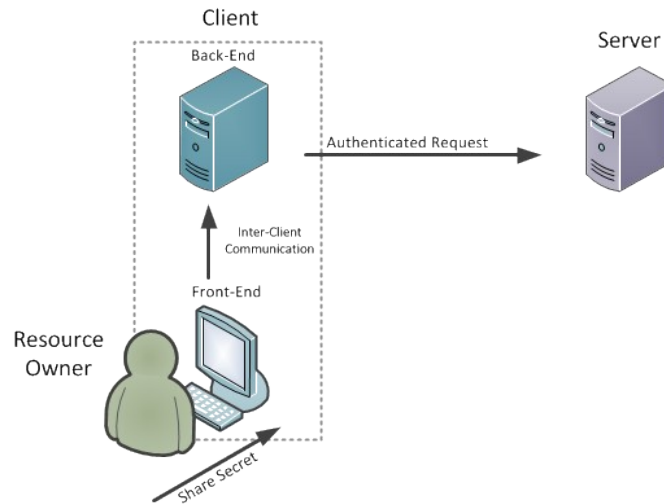


Figure 2: 3-legged OAuth

### 5.3 Implémentation dans Sympa

En suivant le schéma ci-dessus, nous avons implémenté opensocial dans le logiciel Sympa. Sympa est désormais capable d'assurer le rôle de fournisseur de ressources et de client OAuth. Sympa est développé en perl et nous avons pu nous appuyer sur la bibliothèque CPAN OAuth::Lite pour ce développement, seule bibliothèque à fournir des fonctionnalités à la fois pour le côté client et le côté fournisseur de ressources. La difficulté essentielle dans l'implémentation d'opensocial repose sur la partie liée à OAuth. En effet, la partie « strictement opensocial » n'est somme toute que la définition, la récupération et la diffusion d'attributs utilisateur alors qu'OAuth implique la création de jetons permettant de signer les échanges entre les applications, la propagation de ces jetons, leur stockage, leur récupération et le contrôle de leur validité. Si, côté client, tout est déjà développé dans la bibliothèque, la diversité des applications pouvant faire office de fournisseur de ressources rend impossible la constitution d'une bibliothèque générique, en particulier pour les questions de stockage de jeton.

Ce développement a été fait dans le cadre du projet Geant JRA3 [11], où l'objectif était avant tout de se servir de Sympa comme gestionnaire de groupe et comme serveur de listes dans le cadre d'un système de groupware inter-fédérations. Les attributs échangés ne sont en conséquence pas des attributs de base d'opensocial, mais des attributs définis dans une extension d'opensocial, nommées VOOT[12]. Deux attributs sont définis : les groupes d'une personne et les membres d'un groupe.

En suivant les mécanismes de l'API serveur d'opensocial, Sympa peut exporter la listes des listes d'une personne ou la liste des membres d'une liste vers une autre application.

L'intérêt de ce développement est que l'implémentation d'un client opensocial est très simple, du fait du grand nombre de bibliothèques développées. Par conséquent, il sera désormais facile d'exporter les groupes Sympa vers d'autre applications (et inversement d'importer la liste des abonnés à une liste depuis d'autres applications). Ceci facilite la mise en place d'outils de groupware exploitant Sympa comme gestionnaire de groupe.

## 6 Conclusion

Les spécifications opensocial sont à l'origine conçues pour favoriser le développement de gadgets enrichissant et mettant en valeur les réseaux sociaux. Les technologies développées en ce sens présentent l'intérêt de faciliter l'interaction entre applications non prévues pour communiquer entre elles à l'origine. D'ores et déjà, cette technologie est utilisée dans la communauté enseignement supérieur et recherche. Ainsi, la plateforme collaborative danoise « Surfconnect » est entièrement fondée sur opensocial. À l'échelle de RENATER, les évolutions futures d'Universalistes vont s'appuyer sur cette technologie pour améliorer la qualité de l'intégration des briques logicielles utilisées, en particulier pour améliorer l'expérience utilisateur.

Au sein des établissements, nous sommes habitués à évoluer dans des environnements informatiques contrôlés. Cela est rendu possible par des frameworks performants autant que contraignants en terme de développement.

L'évolution actuelle des universités implique un appel de plus en plus fréquent à des ressources extérieures (fournisseurs de documentation électronique, des cours en ligne, de messagerie collaborative, etc.). Pour autant, les outils internes aux établissements, notamment les ENT, ne sont pas obsolètes et remplissent toujours leur rôle. L'intégration d'opensocial dans ces outils permettra d'intégrer de nouvelles fonctionnalités dans nos systèmes d'information en s'appuyant sur des services extérieurs.

Comme l'expérience d'implémentation dans Sympa l'a prouvé, il est facile d'étendre le modèle de données d'opensocial pour construire une large gamme d'attributs. Par ailleurs, la technologie OAuth résout de nombreux problèmes liés à l'interaction entre deux applications non prévues pour être interopérables. Gérés côté client, les échanges de données se font, avec son consentement, dès lors que deux applications (par exemple un ENT d'établissement et un réseau social) ont établi une relation de confiance et accepté que leurs utilisateurs transfèrent des données de l'un à l'autre.

Opensocial est extensible, sûr et facile à implémenter ; rien n'empêche de le considérer comme une technologie générique de transfert de données entre applications, ouvrant à l'enseignement supérieur et à la recherche le potentiel d'offrir tout une gamme de nouveaux services accessibles via le web.

## 7 Annexe : table de correspondance des données opensocial avec les attributs Supann

La table ci-dessous recense les attributs Supann 2009 et propose un équivalent dans les attributs opensocial.

Les attributs opensocial à forme composée (du type Person.displayName) correspondent à des champs d'attributs (dans cet exemple, le champ « displayName » de l'attribut « Person »).

Si la mention (EOE) apparaît à côté du champ, cela signifie que l'attribut opensocial relève de l'extension « Enterprise » (Enterprise OpenSocial Extensions) du modèle de données opensocial.

Il est à remarquer que, pour les attributs opensocial, si le nom de l'attribut se termine par un « s », il peut être multivalué.

Attribut Supann	Équivalent opensocial	Attribut Supann	Équivalent opensocial
cn (common name)	-	supannAliasLogin	-
dc (domainComponent)	-	supannAutreMail	Person.emails
description	-	supannAutreTelephone	Person.phoneNumbers
displayName	Person.displayName	supannCivile	Person.gender (combiné à Person.relationshipStatus)
eduOrgHomePageURI	Person.organizations.webpage	supannCodeEntite	-
eduOrgLegalName	Organization.name	supannCodeEntiteParent	-
eduOrgSuperiorURI	-	supannCodeINE	-
eduOrgWhitePagesURI	-	supannEmpCorps	-
eduPersonAffiliation	Person.employeeType (EOE)	supannEmpld	-
eduPersonNickname	Person.nickname	supannEntiteAffectation	-
eduPersonOrgDN	-	supannEntiteAffectationPrincipale	-
eduPersonOrgUnitDN	-	supannEtablissement	-
eduPersonPrimaryAffiliation	-	supannEtuAnneeInscription	-
eduPersonPrimaryOrgUnitDN	-	supannEtuCursusAnnee	-
eduPersonPrincipalName	-	supannEtuDiplome	-
facsimileTelephoneNumber	Person.phoneNumbers	supannEtuEtape	-
givenName	Person.name.givenName	supannEtuld	-
l (locality)	Person.addresses.locality	supannEtuInscription (attribut composite)	-
labeledURI	Person.urls	supannEtuRegimeInscription	-
mail	Person.emails	supannEtuSecteurDisciplinaire	-
mailForwardingAddress	-	supannEtuTypeDiplome	-
member	-	supannGroupeAdminDN	-
mobile	Person.phoneNumbers	supannGroupeDateFin	-
o (organization)	Organization.name	supannGroupeLecteurDN	-
ou (organizational unit)	-	supannListeRouge	-



owner	-	supannMailPerso	Person.emails
postalAddress	Address	supannOrganisme (obsolète)	Person.organizations
preferredLanguage	-	supannParrainDN	-
sn (surname)	Person.name.familyName	supannRefId	-
supannActivite	-	supannRole (obsolète)	-
supannAffectation (obsolète)	-	supannRoleEntite (attribut composite)	-
		supannRoleGenerique	-
supannTypeEntite	-		
supannTypeEntiteAffectation	-		
telephoneNumber	-		
title	-		
uid	-		
userCertificate	-		
userPassword	-		

## 8 Bibliographie

- [1] Méлина Nemec-Poncik, 2011 : Yahoo, Google et MySpace créent une fondation de soutien à OpenSocial. Le Monde informatique. <http://www.lemondeinformatique.fr/actualites/lire-yahoo-google-et-myspace-creent-une-fondation-de-soutien-a-opensocial-25698.html>
- [2] Martin Scott Nicklous et al., 2003. Spécification JSR 168 : <http://jcp.org/ja/jsr/detail?id=168>
- [3] Dan Holevoet and Arne Roomann-Kurrik, 2009 : Opensocial tutorial, <http://docs.opensocial.org/display/OSD/OpenSocial+Tutorial>
- [4] Opensocial, 2011 : Spécifications opensocial : <http://docs.opensocial.org/display/OSD/Specs>
- [5] Jonathan LeBlanc, 2011 : Programming social applications, O'reilly
- [6] Spécification d'Oauth 1.0 (RFC 5849) : <http://tools.ietf.org/html/rfc5849>
- [7] Liste des projets U-portal : <https://wiki.jasig.org/display/UPC/Mentorable+Projects>
- [8] Spécifications du modèle de données opensocial : <http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-Data.xml>
- [9] Spécification Supann 2009 : <http://www.cru.fr/documentation/supann/index>
- [10] Spécification de l'API serveur d'opensocial : <http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-API-Server.xml>
- [11] Page d'accueil de la Joint research activity 3 : « Identity federations » sur le site de Geant. [http://www.geant.net/Research/Multidomain\\_User\\_Application\\_Research/Pages/IdentityFederations.aspx](http://www.geant.net/Research/Multidomain_User_Application_Research/Pages/IdentityFederations.aspx)
- [12] Andreas Solberg et al. : Spécifications du protocole VOOT : <https://github.com/andreassolberg/voot/wiki/Protocol>