

VERS DES SERVICES WEB ADAPTES : COMMENT INTEGRER LE CONTEXTE DANS LES DIFFERENTES ARCHITECTURES DE SERVICES WEB ?

Bouchra SOUKKARIEH, Dana KUKHUN, Florence SEDES

{sokarieh,kukhun,sedes}@irit.fr

IRIT, Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse Cedex 9, France

Mots clefs :

Services Web, adaptation, contexte, client/serveur, Peer-to-Peer, CC/PP, CSCP.

Keywords:

Web Services, adaptation, context, client/server, Peer-to-Peer, CC/PP, CSCP.

Résumé

Avec l'émergence de la mobilité et la croissance du nombre des utilisateurs et du volume des données, l'accessibilité aux ressources par les Services Web est devenue difficile. C'est pourquoi ces services, basés sur des architectures client/ serveur, s'orientent vers l'architecture Peer-to-Peer qui fournit à l'utilisateur un accès direct à ces ressources.

Cet accès n'intègre pas jusqu'à présent le processus d'adaptation au contexte. Les chercheurs dans ce domaine travaillent donc à intégrer le contexte comme un paramètre essentiel et un mécanisme d'adapter la recherche d'information dans les services web non seulement sur le profil et la requête de l'utilisateur mais aussi sur son contexte.

Dans cet article, nous mettons l'accent sur le manque de prise en compte du contexte (de l'utilisateur et du Service Web) et proposons de l'intégrer dans les deux architectures de Services Web (client/serveur et Peer-to-Peer) afin de restituer à l'utilisateur la liste de services la plus adaptée à son contexte.

1 Introduction

Avec l'émergence des nouvelles technologies de terminaux mobiles (PDA, téléphone mobile, etc.), l'utilisateur peut travailler et effectuer plusieurs missions en se déplaçant en toute liberté. Cette mobilité exige de nouvelles contraintes du processus d'adaptation qui doit prendre en compte non seulement le profil de l'utilisateur (centres d'intérêts, caractéristiques statiques, etc.) mais aussi son contexte (dispositif, environnement, localisation, etc.).

En effet, les chercheurs dans le domaine de l'adaptation travaillent à mettre l'accent sur le contexte comme un paramètre essentiel et un mécanisme associé à la recherche d'information qui est basé, de plus en plus, sur les Services Web. L'adaptation et la prise en compte du contexte permettent de délivrer à l'utilisateur des services pertinents non seulement pour sa requête mais aussi pour son contexte d'utilisation.

Ce processus d'adaptation doit être pris en compte dans les Services Web, à l'origine basée sur l'architecture client/serveur. Cette architecture donne aux clients un accès centralisé aux données où le serveur devient au cœur du système et contrôle le passage des données entre les différents clients. Mais à cause de la croissance du nombre d'utilisateurs et du volume des données, le serveur est, de plus en plus, chargé ce qui menace la stabilité et la facilité d'accès aux ressources d'information. Donc nous avons besoin d'une architecture Peer-to-Peer qui repose sur un principe d'égalité entre les nœuds où chaque utilisateur (peer) joue le rôle d'un serveur et d'un client en même temps.

Dans le cadre de ce travail, nous nous intéressons donc aux possibilités d'introduire l'adaptation au contexte dans les différentes architectures de Services Web (client/serveur, Peer-to-Peer) afin de restituer à l'utilisateur nomade la liste des Services Web adaptés non seulement à son profil mais aussi à son contexte. Cet article est organisé de la manière suivante : nous présentons, dans la section 2, un état de l'art sur les différentes architectures des Services Web, la notion du contexte et des travaux connexes concernant l'adaptation des SW. Dans la section 3, nous détaillerons notre contribution qui vise à étendre l'architecture client/serveur et Peer-to-Peer des SW afin prendre en compte le contexte, dans la section 4, nous traduirons notre contribution dans une application de Services Web, la Bourse.

2 Etat de l'art

2.1 Les Services Web

Un Service Web (SW) est une application mise à disposition sur l'Internet par un fournisseur de service. Cette application permet l'interopérabilité entre les utilisateurs à travers le Web. Cette interopérabilité est assurée en utilisant des standards et protocoles ouverts [8] [3]. Citons:

- WSDL (Web Service Description Language) qui décrit les interfaces des SW [12],
- UDDI (Universal Description, Discovery and integration) qui contient les descriptions de SW [14] et,
- SOAP (Simple Object Access Protocol) qui invoque les descriptions de SW [11].

Le cycle de vie des Services Web est présenté dans la Figure 1: le fournisseur de service construit le service et publie sa description dans l'annuaire. Le client lance une requête à l'annuaire en cherchant un service. L'annuaire découvre le service demandé et le restitue à l'utilisateur. Une fois le service trouvé, le client aura une interaction directe avec le service [8].

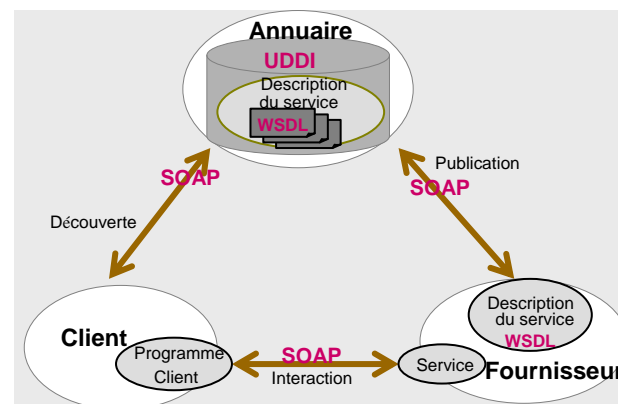


Figure 1. Architecture classique de SW

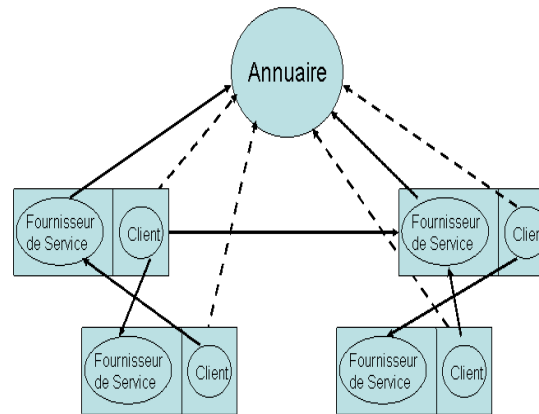


Figure 2. Service Web basé sur l'architecture Peer-to-Peer hybride

La structure principale des Services Web est basée sur l'architecture client/serveur qui donne aux clients un accès centralisé aux données. Dans cette architecture, le serveur est le cœur du système. Il contrôle le passage des données entre les différents clients. Mais à cause de la croissance du nombre des utilisateurs et du volume des données, le serveur devient, de plus en plus, chargé ce qui menace la stabilité et la facilité d'accès aux ressources d'information. Alors, les applications s'orientent vers l'architecture Peer-to-Peer qui assure un accès direct entre les différents clients (cf. Figure 2). Cette architecture repose sur un principe d'égalité entre les nœuds où chaque utilisateur (peer) joue le rôle d'un serveur et d'un client en même temps. Cette architecture a été utilisée dans les Services Web afin de s'adapter à la mobilité simultanée des services et des clients.

Dans cette architecture, les "peers" sont tous connectés de manière centralisée avec un seul annuaire : chaque peer enregistre ses services dans l'annuaire et l'utilise pour chercher d'autres services disponibles. Ensuite, quand un service est trouvé, le peer peut entrer directement dans une connexion décentralisée (Peer-to-Peer) avec le peer qui fournit ce service.

L'interaction dans cette architecture hybride prend deux directions ; la première concerne l'interaction entre un peer et l'annuaire où le peer peut jouer le rôle d'un fournisseur qui publie ces services à l'annuaire et/ou le rôle d'un client qui cherche un service (cf. Figure 2), la deuxième concerne l'interaction entre peers, où le peer joue le rôle d'un client (il utilise le service des autres peers) et le rôle d'un fournisseur de service (il fournit un service aux autres peers) [4].

2.2 L'adaptation au Contexte dans les Services Web

Dans cette section, nous commençons par définir le contexte puis nous présentons un rapide tour d'horizon de quelques travaux qui s'intéressent aux possibilités d'introduire de l'adaptation au contexte dans les Services Web.

Le contexte est largement utilisé et défini comme un ensemble des circonstances qui accompagnent un événement. À partir de cette idée, on peut parler du contexte dans lequel les interactions entre un individu et un système ont lieu.

Les chercheurs dans le domaine de l'adaptation n'ont pas encore abouti à une définition générique de la notion de contexte. Mais parmi les différentes définitions qui existent, la définition de Day est largement acceptée : le contexte couvre toutes les informations pouvant être utilisées pour caractériser la situation d'une entité. Une

entité est une personne, un lieu ou un objet qui peut être pertinent pour l'interaction entre l'utilisateur et l'application, y compris l'utilisateur et l'application eux-mêmes [2].

Plusieurs travaux ont pris en compte le contexte dans les Services Web :

- Les travaux de [1] s'intéressent en aval à l'impact du contexte sur l'application. Cette équipe a développé une plateforme (SECAS) qui rend les applications adaptables aux différents contextes d'utilisation sur leurs trois composantes : données, services et présentation. Une stratégie d'adaptation a été établie dans le domaine médical,

- Les travaux de [9] présentent un système d'information dans le domaine de tourisme (CATIS) en permettant l'adaptation aux dispositifs mobiles en termes de contenu et de présentation. Ils prennent en compte les préférences de l'utilisateur, sa localisation, le type de terminal, puis ils calculent la direction et la vitesse de l'utilisateur par sa localisation précédente,

- Les travaux de [6] présentent une plateforme qui facilite le développement et le déploiement de Services Web adaptables au contexte d'utilisation. Ils permettent l'adaptation aux dispositifs mobiles concernant l'adaptation à la localisation de l'utilisateur.

Nous constatons que les travaux étudiés ci-dessus présentent des services programmés afin de répondre de façon adéquate à des besoins particuliers des utilisateurs. Aucun de ces travaux ne permet de restituer à l'utilisateur la liste de services adaptés à son contexte. C'est dans cette direction que notre travail s'inscrit.

3 Contribution

Dans cette section, nous présentons notre proposition qui vise à intégrer l'idée de l'adaptation au contexte, d'une part, dans l'architecture client/serveur de Services Web et d'autre part, dans l'architecture Peer-to-Peer de Services Web.

3.1 L'intégration du contexte dans l'architecture client-serveur des Services Web

Notre proposition (Figure 3) est basée sur l'architecture classique client/serveur de SW en gardant les trois entités (fournisseur, annuaire et client) ainsi que les trois standards (WSAL, SOAP et UDDI). Notre contribution vise à ajouter à cette architecture une couche d'adaptation contenant différentes composantes dédiées à l'adaptation au contexte. Cette couche est centralisée dans l'annuaire. Les différentes étapes pour intégrer le contexte dans l'architecture classique de SW et les rôles et fonctionnalités de ces composantes sont les suivantes :

- 1) Les fournisseurs de SW décrivent leurs services à l'aide de métadonnées. Ces métadonnées, qui sont enregistrées dans un document XML [13], décrivent des critères d'adaptation des services.
- 2) Un composant de Gestionnaire de Profil Service (GSPS) extrait ces métadonnées et les stocke dans la base de données (Profil Services (PS)).
- 3) La description WSDL, fournie par le fournisseur, est transmise à l'annuaire comme dans l'architecture classique.
- 4) Le client lance sa requête à l'annuaire (au format SOAP).
- 5a) + 5b) Le Gestionnaire de Contexte GC capte le contexte.
- 6) Le GC stocke le contexte, en utilisant les métadonnées, dans une base de données sous format de documents XML.
- 7) Le résultat de la requête (la liste des services répondant aux besoins du client) est transféré de l'annuaire au GC.
- 8) Le GC demande au GSPS les profils de services répondant aux besoins de l'utilisateur.
- 9) Le GC prend le contexte de l'utilisateur stocké, puis il compare les deux profils (l'un de service et l'un de l'utilisateur).
- 10) Le GC envoie à l'utilisateur la liste des services adaptés à son contexte.
- 11) La communication entre le client et le fournisseur se fait ensuite de manière traditionnelle par l'intermédiaire de SOAP.
- 12) Quand la communication entre le client et le fournisseur commence, le GC envoie le contexte de l'utilisateur au fournisseur qui rend le service choisi adaptable au contexte de l'utilisateur.

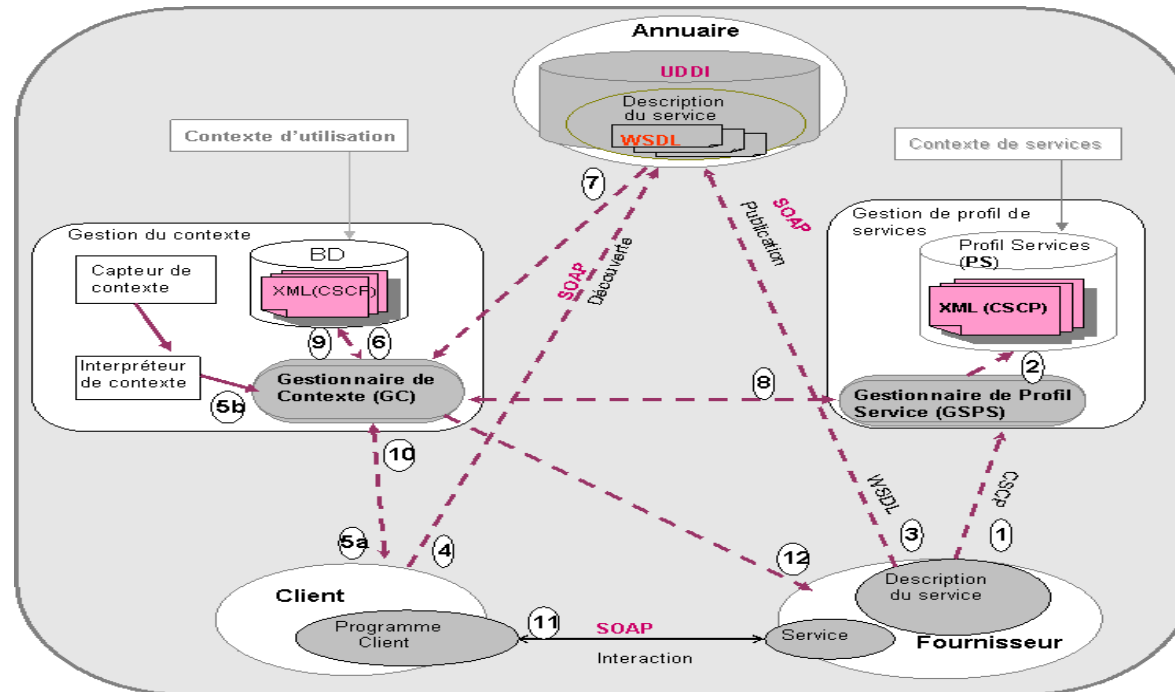


Figure 3. Architecture client/serveur de Service Web intégrant le contexte de l'utilisateur

Le contexte de service est exprimé directement du fournisseur de service. Dans notre architecture, le contexte de l'utilisateur est récupéré d'une manière explicite et implicite. Quand l'utilisateur lance sa requête, il peut exprimer ses caractéristiques statiques et ses préférences explicitement. Les autres caractéristiques (localisation, network, etc.) sont récupérées implicitement par le capteur du contexte comme nous allons montrer dans le paragraphe suivant.

3.1.1 Gestion de Contexte

Captage du contexte de l'utilisateur à l'aide de senseurs physiques : ces senseurs génèrent des données brutes qui ne sont pas directement exploitables par l'application. Dans notre architecture nous définissons l'entité « capteur de contexte » pour représenter un système de capture du contexte.

Interprétation du contexte : cette étape vise à transformer les données contextuelles brutes en paramètres plus faciles à utiliser par l'application.

Stockage du contexte : l'élément Gestionnaire de Contexte (GC) stocke les informations contextuelles (implicite et explicite), sous format XML, dans une base de données. Cet élément permet de :

- recevoir la liste de Services Web adaptés à la requête d'utilisateur de l'annuaire pour la filtrer selon les informations contextuelles,
- demander le profil de chaque Service web adapté dans la liste pour le comparer avec les informations contextuelles,
- envoyer à l'utilisateur la liste de Services la plus adaptée à son besoin et à son contexte, et
- vérifier si le contexte d'utilisation est changé. Dans ce cas : le GC doit prendre en compte ces modifications pour mettre à jour la liste de Services retournée.

3.1.2 Gestion de profil de services

Dans ce composant, GSPS est responsable d'extraire les profils de Services Web, de stocker ces profils dans une base de données PS, et de restituer à GC les profils de services répondant aux besoins de l'utilisateur.

3.1.3 Représentation de profil du contexte

Dans notre architecture, nous nous basons sur les travaux de [7] auxquels nous ajoutons des paramètres concernant la session et le réseau. Donc, nous définissons le contexte par *quatre axes pour couvrir tous les paramètres du contexte* :

Les caractéristiques de l'utilisateur composées des *caractéristiques statiques et évolutives*. *Les caractéristiques statiques* comprennent l'identité de l'utilisateur (nom, prénom, etc.), les données démographiques (âge, genre, adresse, etc.), etc. *Les caractéristiques évolutives* comprennent l'environnement de l'utilisateur (localisation, temps, etc.) et ses préférences.

Les caractéristiques de dispositif présentent d'une part, le profil de matériel (type de dispositif, taille de l'écran, etc.), et d'autre part, le profil de logiciel (système d'exploitation, version, vendeur, etc.).

Les caractéristiques de réseau qui présentent des informations sur le type de réseau, ses caractéristiques, etc.

Les caractéristiques de session présentant la connexion par l'utilisateur au système (la durée, la date, l'état de connexion (en cours, interrompue ou suspendue),...).

Pour restituer à l'utilisateur la liste de services les plus pertinents à son contexte, nous devons prendre en compte non seulement le contexte d'utilisateur mais aussi le contexte de services. Pour représenter les informations contextuelles, la majorité des chercheurs, comme [7] a utilisé le format CC/PP [10]. Bien que CC/PP soit décomposable, uniforme et extensible, mais nous ne pouvons pas l'utiliser pour représenter notre contexte car il manque, dans ce format, de fonctionnalités de **structuration**. Sa hiérarchie stricte en **deux niveaux** n'est pas appropriée à la capture de structures de profils complexes [5].

C'est pour cela, nous utilisons le format CSCP qui surmonte les défauts de CC/PP concernant la structure en fournissant une structure multi-niveaux et qui permet une représentation de tous les types d'information contextuelle. CSCP fournit des mécanismes qui attachent des conditions et des priorités aux attributs.

La représentation de notre contexte (utilisateur et services) selon ce format (Figure 4) est composée de quatre éléments : le profil d'utilisateur, le profil de session, le profil terminal et le profil de réseau.

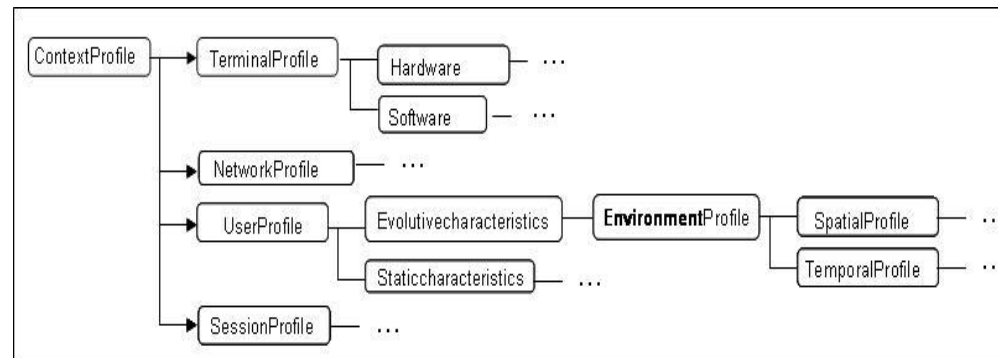


Figure 4 Représentation de contexte

Afin de trouver la liste de services adaptés au contexte de l'utilisateur, nous allons effectuer un processus de "matching" entre le contexte de l'utilisateur et le contexte de chaque service en comparant les attributs de leurs métadonnées. La liste des Services Web retournée à l'utilisateur sera ordonnée

selon le degré de similarité calculé dans le matching où le Service Web adapté en tête de la liste est le service le plus adapté à la demande de l'utilisateur.

3.2 L'intégration du contexte dans l'Architecture Peer-to-Peer Hybride de Services Web

Dans le paragraphe précédent, nous avons intégré le contexte dans l'architecture client/serveur de service Web pour assurer une meilleure adaptation. Mais comme nous l'avons mentionné, les Services Web s'orientent vers l'architecture Peer-to-Peer hybride à cause de la croissance du nombre d'utilisateurs et du volume des données. Et donc, nous devons intégrer le contexte pas seulement dans l'architecture client/serveur de SW mais également dans l'architecture Peer-to-Peer hybride de Services Web.

Nous illustrons, dans la Figure 5, la prise en compte du contexte dans l'architecture Peer-to-Peer de Services Web ; nous allons conserver les différents peers de Figure 2. Notre contribution réside dans l'ajout, à chaque peer, différents composants dédiés à l'adaptation au contexte tout en gardant l'état de l'annuaire comme il est présenté dans l'architecture principale des Services Web Peer-to-Peer.

Dans la figure 5, nous montrons l'interaction entre 3 peers ; Peer 1 et Peer 2 jouent le rôle des clients (P1-Client, P2-Client) et des fournisseurs des services (P1-Fournisseur, P2-Fournisseur) et Peer 3 joue le rôle de l'annuaire. Ces étapes sont présentées dans la Figure 5 par des numéros :

1) Le fournisseur des Services Web (P2-Fournisseur) décrit ses services à l'aide de métadonnées, extraites par le GSPS, qui décrivent les critères d'adaptation de Services Web et les stocke dans la base de données SP.

2) La description WSDL, sans les métadonnées, est ensuite transmise à l'annuaire (Peer3).

3) Le client (P1-Client) lance sa requête à l'annuaire et au même temps, le GC de Peer 1 capte le contexte de l'utilisateur et stocke les métadonnées décrivant ce contexte dans une base de données sous format de document XML.

4) Le résultat de la requête (la liste des services répondant à la requête de Peer 1) est transféré de l'annuaire (Peer3) vers le GC de Peer 1.

5) Le GC de Peer 1 demande au GSPS de Peer 2 les profils de services répondant à la requête de l'utilisateur (Peer 1).

6) Après la restitution du profil du service depuis le GSPS de Peer2, le GC de Peer 1 prend le contexte de l'utilisateur stocké, puis compare les deux profils (profils de services de Peer 2 et de celui de l'utilisateur de Peer 1).

7) Le GC de Peer 1 envoie à l'utilisateur la liste des services adaptés à son contexte.

8) En fonction des résultats restitués par CG de Peer1, l'utilisateur (Peer 1) établit une connexion Peer-to-Peer directe avec le fournisseur de service situé en Peer2.

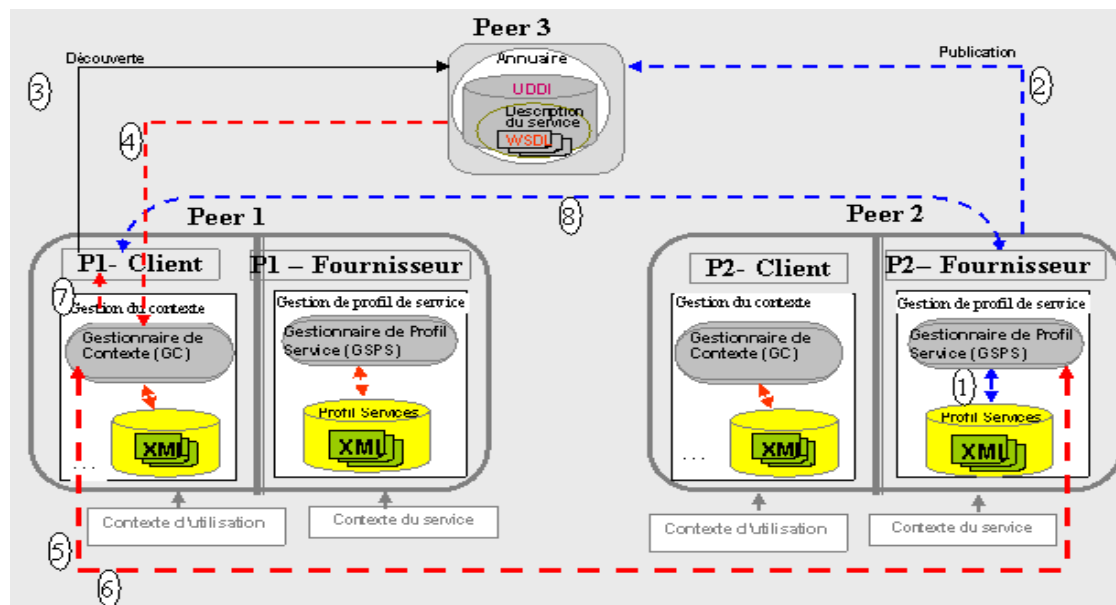


Figure 5. Architecture Peer-to-Peer de Services Web contenant le contexte de l'utilisateur

Nous allons utiliser les méthodes présentées dans la section 3.1 afin de réaliser le processus de matching et la représentation du contexte dans cette architecture.

4 L'application de Notre Proposition dans Un Service Web

Notre vision de l'adaptation peut être intégrée dans un Service Web dynamique basé sur l'architecture client/serveur ou Peer-to-Peer. Dans cette partie, nous allons prendre le domaine de la Bourse comme application de nos travaux selon l'architecture Peer-to-Peer hybride dans le but de valider notre proposition.

Dans cette application (Service Web basé sur la Bourse), nous offrons au participant la possibilité de :

- jouer le rôle d'un fournisseur de service, où il va enregistrer sa demande de vente de ses actions dans l'annuaire de ce service. L'annuaire présente ici la base de données du site de l'entreprise qui prend en charge la médiation entre les clients et la bourse (Dans l'architecture Peer-to-Peer, les fournisseurs vont utiliser l'annuaire pour enregistrer leurs demandes de manière centralisée). Et / ou,

- jouer le rôle d'un client, où il peut lancer sa demande à l'annuaire afin de recevoir la liste des fournisseurs potentiels (Dans l'architecture Peer-to-Peer, les clients vont utiliser l'annuaire pour lancer leurs demandes et recevoir la liste des fournisseurs potentiels. Puis, ils vont contacter directement les fournisseurs potentiels d'une manière décentralisée).

Nous envisageons d'appliquer la phase d'auto-adaptation dans ce scénario. Cette adaptation permet de faciliter le processus de recherche conduit par le client pour trouver les fournisseurs de service les plus adéquats.

Dans ce cas, l'annuaire va restituer aux clients une liste de résultats qui prend en compte le contexte du client et le contexte du fournisseur pour réaliser un meilleur achat. Ce contexte traduit les préférences du participant (le domaine dans lequel le client aimerait investir, la marge de profit qu'il préfère, ...).

5 Conclusion

Dans cet article, nous avons intégré l'idée de l'adaptation au contexte, d'une part, dans l'architecture client/serveur de Services Web et d'autre part, dans l'architecture Peer-to-Peer de Services Web.

Notre proposition vise à permettre à l'utilisateur d'atteindre la liste de services les plus pertinents à sa requête et à son contexte. Enfin, nous avons proposé la Bourse comme une application dans lequel nous pouvons intégrer notre contribution.

Dans nos travaux futurs, nous envisageons d'implémenter notre proposition d'architecture supportant l'intégration de contexte dans les deux architectures des Services Web.

6 Bibliographie

- [1] CHAARI T., EJIGU D., LAFOREST F. et SCUTURICI V. M., Modeling and Using Context in Adapting Applications to Pervasive Environments, IEEE International Conference on Pervasive Services, June 2006, Lyon, France.
- [2] DEY A. K. et ABOWD G. D., Towards a Better Understanding of Context and Context-Awareness, CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness, April 2000, The Hague, The Netherlands.
- [3] FAUVET M. C., Modèles et Systèmes pour Applications Emergentes (suite) Chapitre 6, Systemes d'Information accessibles via des Services Web, November 14, 2005.
- [4] GEHLEM G. et PHAM L., Mobile Web Services for peer-to-peer Application, IEEE, 8/04/2004.
- [5] HELD A., BOUCHHOLZ S. et SHILL A., Modeling of Context for Pervasive Computing Application, In Proceedings of the 16th world Multiconference on Systemics, Cybernetics and Informatics (SCI), July 2002, Orlando, FL, USA.
- [6] KEIDL M. et KEMPER A., Towards Context-Aware Adaptable Web Services, Proceedings of the 13th World Wide Web Conference (WWW), New York, USA, Mai, 2004.
- [7] LOPEZ-VEMASCO C., VILLANOVA-OLIVER M., GENSEL J. et MARTIN H., Services Web adaptés aux utilisateurs, Deuxièmes Journées Francophones : Mobilité et Ubiquité 2005, 31 mai - 3 juin, 2005, pp.133-136.
- [8] MELLITI T., Interopérabilité des Services Web complexes. Application aux systems multiagents, Thèse de doctorat, Université Paris IX Dauphine, 8 décembre 2004.
- [9] PASHTAN A., HEUSSER A. et SCHEUERMANN P., Personal Service Areas for Mobile Web Application, IEEE Internet Computing, November-December 2004.
- [10] W3C, Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation 15 January 2004. La dernière version: <http://www.w3.org/TR/CCPP-struct-vocab/> (Janvier 2007).
- [11] W3C, Recommendation W3C 24 Juin 2003. La dernière version : <http://www.w3.org/TR/soap12-part0/> (Décembre 2006).
- [12] W3C, Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001. La dernière version: <http://www.w3.org/TR/wSDL>. (Décembre 2006).
- [13] W3C, XML Base W3C Recommendation 27 Juin 2001. La dernière version : <http://www.w3.org/TR/xmlbase> (Décembre 2006).
- [14] W3SCHOOLS, http://www.w3schools.com/wSDL/wSDL_uddi.asp (Décembre 2006). ARGYRIS C., *Savoir pour agir : surmonter les obstacles à l'apprentissage organisationnel*, InterEditions, 1995.