Protection des données personnelles côté utilisateur dans le e-commerce

Kheira Bekara Dari Bekara

THESE DE DOCTORAT CONJOINT
TELECOM SUDPARIS & L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité : Informatique et Réseaux

Ecole doctorale : Informatique, Télécommunications et Electronique de Paris

Présentée par

Mme Kheira DARI BEKARA

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

Doctorat conjoint Télécom SudParis & Université Pierre et Marie Curie

PROTECTION DES DONNEES PERSONNELLES CÔTE UTILISATEUR DANS LE E-COMMERCE

Soutenue le 18 Décembre 2012 devant le jury composé de :

Frédéric CUPPENS	Professeur HDR, TELECOM Bretagne, Rennes	Rapporteur
Anas Abou El KALAM	Professeur HDR, Université Cadi abi Ayad, ENSA, Maroc	Rapporteur
Jean-Gabriel GANASCIA	Professeur, LIP6, Paris	Examinateur
Samia BOUZEFRANE	Professeur HDR, CNAM, Paris	Examinateur
César VIHO	Professeur HDR, IRISA	Examinateur
Maryline LAURENT	Professeur HDR TELECOM SudParis	Directrice de thèse

Thèse n° 2012TELE0045

A MES BEBES

REMERCIEMENTS

La liste des personnes que je souhaite remercier est longue, et réussir à exprimer toute ma gratitude envers ces personnes avec les mots justes me semble aujourd'hui bien plus délicat que de rédiger une thèse.

Je souhaite remercier et exprimer ma gratitude la plus profonde à ma directrice de thèse Mme Maryline LAURENT. Qui, en dépit de ses énormes responsabilités et un planning démentiel a su trouver le temps de me pour m'encadrer, m'aider, et m'accompagner tout au long de ces années. Sa présence, son ouverture, ses conseils, ses encouragements et sa complémentarité m'ont permis non seulement de mener ce travail à bien, mais également de découvrir et d'aimer le monde de la recherche. Elle a su me passer sa rigueur scientifique et son enthousiasme pour la recherche. Pour cet enrichissement scientifique et personnel, je la remercie sincèrement.

Je la remercie aussi de m'avoir fait remarquer à de nombreuses reprises « as-tu pensé à vérifier ça ? », le « ça » étant souvent très pertinent.

Mes vifs remerciements vont à M. Frédéric CUPPENS et M.Anas ABOU-ALKALAM pour avoir accepté d'être rapporteurs de ma thèse, et pour l'avoir lue avec autant d'attention. Merci pour leurs conseils, remarques et commentaires, qui m'ont permis d'améliorer ce manuscrit tant sur le fond que sur la forme.
Je remercie également à M. Jean-Gabriel GANASCIA et Mme Samia BOUZEFRANE pour avoir accepté de participer à mon jury de thèse, et pour leurs commentaires qui ouvrent de nombreuses pistes de recherche.

Tout mon travail s'est déroulé à Télécom SudParis, (SAMOVAR, LOR, RST). Je tiens à remercier les responsables pour la qualité de l'ambiance qu'ils savent y entretenir. Je remercie notamment Mme Brigitte LAURENT et Mme Françoise ABAD qui m'ont permis de traverser sans encombres toutes les embûches administratives auxquelles j'ai du faire face.

La liste des stagiaires qui ont collaboré avec moi est longue. Je remercie particulièrement Than ha, Diane, Elise, Régis, Yosra , Himani, Sonia, Marco, Amina, Fabien, et Olivia pour les échanges riches noués lors de nos réunions en C104, D105.

Je remercie beaucoup M. Makhlouf Hadji du département RS2M, pour m'avoir patiemment guidé à travers le domaine de la théorie des jeux ; Je le remercie également pour sa disponibilité pour tous mes questionnements.

Je remercie mes colocataires de bureau avec qui j'ai eu la chance de partager mon bureau ont tous contribué à rendre ces années extrêmement agréables à vivre. Je veux remercier en premier lieu Nesrine pour tous les bons moments passés ensemble, pour sa gentillesse

et son aide extrêmement précieuse par moments, merci également à Sondès et Ghada.

Je remercie mes parents, mes frères (notamment Mohammed) et sœurs d'avoir toujours étaient là malgré mon emploi du temps en pointillé.

Je remercie Chakib pour sa présence et son soutien au quotidien, pour sa patience, sa compréhension et sa capacité à me faciliter la vie durant ces années de thèse et pour avoir toujours eu confiance en moi. Un grand merci aussi pour sa précieuse aide dans la dernière ligne droite.

Un regard particulier à mes trois petites merveilles, *Alaa, Ishaq, Adem*. Ils furent mon rayon de soleil dans les mauvais moments et mes supporteurs inconditionnels. Je ne peux rêver de meilleur soutien qu'eux. Je n'ai pas de mots pour exprimer tout l'amour que j'ai pour eux.

ABSTRACT

Informatics and Internet in particular favor largely the collection of data without user permission, their disclosure to third parties and their cross-analysis. The density of the human activities in the digital world thus constitutes a fertile ground for potential invasions of privacy of the users.

Our works examine first the legal context of privacy protection, as well as the diverse computing means intended for the protection of personal data. A need for user centered solutions emerges, giving him/her more control over his/her personal data. In this perspective, we analyze European and French privacy legislation to extract data protection axis. Then we specify the constraints related to these axes, and we introduce them in existing security policy models. Thus we suggest the application of one model for both access control and privacy protection. The access control model should be extended by new privacy related conditions and parameters. To do so, we define the language XPACML (eXtensible Privacy aware Access Control Markup Language) based on XACML and new privacy extensions.

Placed in an E-commerce context, we define a semantic model allowing to represent various electronic transactions contexts, and leading to a dynamic generation of context-aware XPACML policies.

Looking for a vast protection of the personal data, we dedicate the last part of our works to the possible negotiations which can be made between a user and a service provider. Two protocols are proposed. The first one permits the negotiation of the terms and the conditions of data protection policies, while the second permits the negotiation of the requested data themselves..

RESUME

L'informatique et Internet en particulier favorisent grandement la collecte de données à l'insu de l'utilisateur, leur divulgation à des tiers et le croisement des données. La densité des activités humaines dans le monde numérique constitue donc un terrain fertile pour de potentielles atteintes à la vie privée des utilisateurs.
Les présents travaux examinent d'abord le contexte légal de la protection de la vie privée, ainsi que les divers moyens informatiques destinés à la protection des données personnelles. Il en ressort un besoin de solutions centrées utilisateur, lui donnant davantage de contrôle sur ses données personnelles. Dans cette perspective, nous analysons le cadre légal français et européen pour en tirer des axes de protection. Nous spécifions ensuite les contraintes tirées de ces axes, en proposant de les introduire dans les modèles de politiques de sécurité existants. Ainsi, nous suggérons l'application d'un seul modèle pour le contrôle d'accès et la protection de la vie privée. Le modèle de contrôle d'accès doit être étendu par de nouvelles conditions et paramètres d'accès. Pour cela, nous définissons le langage XPACML (eXtensible Privacy aware Access Control Markup Language) conçu sur la base d'extensions apportées au modèle de contrôle d'accès XACML.

Placés dans un contexte E-Commerce, nous avons défini un modèle sémantique permettant de représenter les contextes liés aux différentes transactions électroniques. Ainsi nous avons pu effectuer une génération dynamique des politiques XPACML en fonction du contexte en cours.

A la quête d'une protection étendue des données personnelles, nous avons consacré la dernière partie de nos travaux aux négociations possibles qui peuvent être effectuées entre un utilisateur et un fournisseur de service. Ainsi nous avons proposé deux protocoles. Le premier porte sur la négociation des termes et conditions des politiques de protection des données, alors que le deuxième porte sur la négociation des données à dévoiler elles-mêmes.

TABLE DES MATIERE

Introduction ……13

1 Vie privée et protection des données personnelles 30
Introduction 30
1.1 La sphère privée 30
1.1.1 Nomenclature et définitions 30
1.1.2 Aspects légaux du droit à la sphère privée 32
1.1.3 Protection de la sphère privée au quotidien 37
1.2 Les dimensions techniques de protection de la vie privée sur Internet 39
1.2.1 Gestion des identités 39
1.2.2 Communications IP anonymes 40
1.2.3 Accès anonymes aux services 40
1.2.4 Autorisation préservant la vie privée 40
1.2.5 Gestion des données personnelles 41
1.3 La protection des données personnelles 42
1.3.1 Les six axes de la protection des données personnelles 42
1.3.2 Problématiques spécifiques aux domaines d'intérêt 44
1.4 Technologies de protection des données personnelles 46
1.4.1 Platform for Privacy Preferences 46
1.4.2 Sticky policies 47
1.4.3 Gestion déportée des données sensibles 48
1.4.4 Agents utilisateurs 49
1.5 Discussion 49
1.5.1 Représentation et raisonnement 49
1.5.2 Manque de contrôle 50
1.6 Conclusion 50

2 Langage XPACML et modèle d'architecture pour la protection des données personnelles 56
Introduction 56
2.1 Problématiques et besoins 56
2.2 Langages de politiques de protection des données 57
2.2.1 Platform for Privacy Preferences (P3P) 58
2.2.2 P3P Privacy Policy Exchange Language APPEL et XPREF 63
2.2.3 Implémentations existantes côté utilisateur 66
2.3 Limitations 68
2.4 Le langage de politiques de contrôle d'accès XACML 68
2.4.1 Modèles de contrôle d'accès sensibles à la protection de la confidentialité 68
2.4.2 Principes de XACML 71
2.4.3 Architecture de XACML 77
2.5 Complémentarité des langages de politiques de protection des données et des langages de contrôle d'accès 79

2.6 XPACML 81
2.6.1 Contraintes à considérer dans les politiques de protection des données XPACML …. 81
2.6.2 Principes du langage XPACML 82
2.6.3 Architecture de contrôle d'accès XPACML 91
2.7 Génération des politiques XPACML 99
2.7.1 Définition du schéma de politique 99
2.7.2 Exemples de politiques XPACML 104
2.7.3 Vérification de conformité des fichiers de politiques 106
2.7.4 Comparaison des politiques (implémentation du PPDP) 107
2.8 Prime Life Policy Language (PPL) 108
2.8.1 Principes du langage PPL 109
2.8.2 Architecture de contrôle d'accès PPL 112
2.8.3 Diagramme de flux du moteur de contrôle d'accès PrimeLife 113
2.8.4 Comparaison entre PPL et XPACML 114
2.9 Conclusion 116

3 Prise en compte du contexte situationnel dans la protection des données personnelles 120
Introduction 120
3.1 Problématique 120
3.2 Etat de l'art 121
3.3 Discussion et idée principale 122
3.4 Prérequis en modélisation sémantique et gestion des contextes 124
3.4.1 La modélisation sémantique 124
3.4.2 Gestion de contextes 127
3.5 Approche proposée 129
3.5.1 Modélisation des contextes 130
3.5.2 Politiques de protection des données à base de contextes 137
3.6 Scénario 140
3.6.1 Modélisation des contextes sémantiques 142
3.6.2 Politiques de protection des données à base de contextes 145
3.7 Prototype d'implémentation 146
3.8 Conclusion 148

4 Négociation des politiques de protection des données 152
Introduction 152
4.1 Problématique 153
4.2 Etat de l'art 153
4.3 Limitations de l'état de l'art 156
4.4 Classification des valeurs de tags P3P 156
4.4.1 Classification des valeurs de l'élément Purpose 156
4.4.2 Classification des valeurs de l'élément Recipient 158
4.4.3 Classification des valeurs de l'élément Retention 159
4.5 Protocole de négociation de politiques à base de classification de valeur des tags P3P ……161
4.5.1 Schéma de négociation 161
4.5.2 Création des préférences de l'utilisateur 162

4.5.3 Exemple illustratif ... 163
4.5.4 Limitations de l'approche ... 166
4.6 Protocole de négociation de politiques à base de fonction de risque ... 166
4.6.1 Principe de négociation ... 166
4.6.2 Vocabulaire de négociation et d'expression des règles ... 168
4.6.3 Modèles des préférences et des politiques du SP ... 169
4.6.4 Côté utilisateur ... 170
4.6.5 Côté SP ... 172
4.6.6 Protocole de négociation à deux tours ... 173
4.7 Conclusion ... 175

5 Négociation des données personnelles ... 178
5.1 Problématique ... 178
5.2 Etat de l'art ... 179
5.3 Cadre de travail choisi ... 180
5.4 Les concepts privés et risques de révélation ... 180
5.4.1 Définitions ... 180
5.4.2 Risques de révélation des données personnelles ... 181
5.5 Eléments de la théorie des jeux ... 185
5.5.1 Taxonomie partielle des jeux ... 186
5.5.2 Concept de solution ... 187
5.5.3 Notre cas : Jeu non coopératif à somme non nulle ... 188
5.5.4 Fonctions d'utilité ... 188
5.6 Théorie des jeux appliquée à la négociation des données ... 189
5.6.1 Modélisation de la révélation des données comme un jeu statique ... 190
5.6.2 Formalisation ... 192
5.6.3 Calcul des stratégies optimales de révélation ... 201
5.7 Communication des données de concepts privés selon une révélation optimale 204
5.8 Contraintes complémentaires au protocole de négociation à base de théorie des jeux 205
5.9 Conclusion ... 206

6 Conclusion et perspectives ... 210

TABLE DES FIGURES

Figure 2.1 Mécanisme opérationnel de P3P 50
Figure 2.2 Exemple basique d'une politique P3P 52
Figure 2.3 Exemple d'une politique APPEL des préférences utilisateur 56
Figure 2.4 Configuration des préférences utilisateur (a) présentation de la politique du SP (b) sous Privacy Bird 58
Figure 2.5 Diagramme objet d'une requête XACML 64
Figure 2.6 Évaluation des requêtes dans XACML 65
Figure 2.7 Diagramme objet du langage de politiques XACML 67
Figure 2.8 Architecture de contrôle d'accès XACML 69
Figure 2.9 Contraintes à considérer dans les politiques de protection des données personnelles 73
Figure 2.10 Structure du langage XPACML (modèle de langage de politique) 74
Figure 2.11 Règle XPACML 78
Figure 2.12 Interface de consentement utilisateur 84
Figure 2.13 Architecture générale XPACML 85
Figure 2.14 Choix d'un niveau de protection prédéfini (a) ou à personnaliser (b) 86
Figure 2.15 Diagramme de flux XPACML 89
Figure 2.16 Définition des restrictions sur les valeurs de l'élément Effect 92
Figure 2.17 Définition des restrictions sur les valeurs de l'élément Action 92
Figure 2.18 Introduction des principaux éléments P3P dans le schéma de politique XPACML 93
Figure 2.19 Flux de traitements pour la conception de documents XML [Pol05, Con05] 94
Figure 2.20 Exemple d'une politique législative XPACML 96
Figure 2.21 Exemple de préférences utilisateur 97
Figure 2.22 Constructeur du PPDP 98
Figure 2.23 Comparaison simple 98
Figure 2.24 Exemple des résultats de différences 99
Figure 2.25 Modèle de langage PPL 100
Figure 2.26 Architecture de contrôle d'accès PrimeLife 103
Figure 2.27 Flux de données du moteur PrimeLife XACML 104
Figure 3.1 Modèle conceptuel d'une ontologie 115
Figure 3.2 Gestion du contexte 117
Figure 3.3 Processus de création de politiques XPACML basées sur le contexte 118
Figure 3.4 Concepts de l'ontologie de protection des données 120
Figure 3.5 Sous-graphe DataType de l'ontologie de protection des données 121
Figure 3.6 Sous-graphe ServiceType de l'ontologie Privacy 122
Figure 3.7 Sous-graphe des éléments des règles de protection des données 123
Figure 3.8 La classe Access 124
Figure 3.9 Illustration des relations 125
Figure 3.10 Relation « Generalization » « Sub relation » 126
Figure 3.11 Construction des politiques de contrôle d'accès à base de contexte 127

Figure 3.12 Diagramme de flux de données XPACML avec prise en compte du contexte..... 129
Figure 3.13 Scénario contextuel..... 130
Figure 3.14 Classes et sous classes du contexte « Communication »..... 131
Figure 3.15 Classes et relations liées au contexte « Construction »..... 132
Figure 3.16 Les assertions des différentes instances avec leurs classes..... 133
Figure 3.17 Exemple partiel d'une politique XPACML basée sur le contexte..... 134
Figure 3.18 Exemple partiel d'une requête d'accès XPACML..... 135
Figure 3.19 Architecture de gestion de contextes..... 136
Figure 4.1 Processus de négociation basique..... 144
Figure 4.2 Classification des valeurs du tag Purpose..... 147
Figure 4.3 Classification des valeurs du tag Recipient..... 148
Figure 4.4 Classification des valeurs du tag Retention..... 149
Figure 4.5 Schéma de négociation de politiques - classification de tags P3P..... 151
Figure 4.6 Première interface utilisateur pour créer et mettre à jour ses préférences pour un type de service spécifique..... 152
Figure 4.7 Définition des préférences utilisateur pour la donnée « *address* »..... 152
Figure 4.8 Premières préférences (tableau du bas) vs première politique SP (tableau du haut)..... 153
Figure 4.9 Deuxièmes préférences vs première politique SP..... 154
Figure 4.10 Premières préférences vs deuxième politique SP..... 154
Figure 4.11 Secondes préférences vs deuxième politique SP..... 154
Figure 4.12 Négociation de politiques à base de fonction de risque..... 157
Figure 4.13 Une règle dans les préférences utilisateur sous format XPACML..... 158
Figure 4.14 Sous ensembles des risques de tolérance de l'utilisateur..... 160
Figure 4.15Interface utilisateur pour la spécification des limites acceptables et inacceptables de politiques associées aux données personnelles..... 160
Figure 4.16 Génération de l'offre utilisateur –Tour 1..... 163
Figure 5.1 Concept privé - Carte d'identité..... 169
Figure 5.2 Classe -Concepts privé..... 169
Figure 5.3 Exposition corrélée dépendant de □□ et □□..... 180
Figure 5.4 Qualité de service de l'utilisateur Figure 5.5 Qualité de service de 182
Figure 5.6 Qualité de service de l'utilisateur avec plan tangent aux points : (□,□,□) et (□,□,□)..... 182
Figure 5.7 Qualité de service du SP du service..... 183
Figure 5.8 Préférence de l'utilisateur..... 184
Figure 5.9 Préférence du SP du service..... 185
Figure 5.10 Fonction d'utilité de l'utilisateur - vue 1..... 186
Figure 5.11 Fonction d'utilité de l'utilisateur - vue 2..... 186
Figure 5.12 Fonction d'utilité du SP du service - vue 1..... 187
Figure 5.13 Fonction d'utilité du SP du service - vue 2..... 187

LISTE DES TABLEAUX

Tableau 2.1 Requête XACML au format tabulaire..... 63
Tableau 2.2 Requête XPACML au format tabulaire..... 75
Tableau 2.3 Description des types de sensibilité..... 77
Tableau 2.4 Regroupement des types de sensibilité en des classes de sensibilité selon le niveau de rigueur..... 80
Tableau 2.5 Description des liens entre les éléments de politique XPACML et les axes de protection des données..... 81
Tableau 5.1 k-anonymity avec k=2 et QI={age, race, gender, zip}..... 167
Tableau 5.2 Notations utilisées..... 170
Tableau 5.3 Tailles des concepts privés..... 171
Tableau 5.4 Poids associés au risque de révélation de concepts privés selon le type de service..... 171
Tableau 5.5 Poids du risque des concepts privés selon le type de service..... 172
Tableau 5.6 Poids normalisés du risque des concepts privés selon le type de service 172
Tableau 5.7 Taxonomie des jeux : Domaine d'application des équilibres de Nash.... 175
Tableau 5.8 Description des types de sensibilité associés à un donnée et prenant en compte les concepts privés..... 193

INTRODUCTION

Contexte

La notion de vie privée d'un individu (ainsi que la question de sa protection) est apparue et a évolué dans les sociétés occidentales parallèlement à l'émergence de l'ensemble des libertés individuelles. La *Magna Carta*, la Glorieuse Révolution britannique, les textes des Lumières, la Déclaration d'Indépendance des Etats-Unis d'Amérique ou la Déclaration des Droits de l'Homme et du Citoyen sont quelques une des étapes historiques témoignant de l'importance croissante donnée à l'individu dans la société. On y voir poindre les délicats équilibres et points de compromis qui existent entre les libertés individuelles et le bien collectif. La gestion de ces équilibres occupe toujours une place prépondérante dans l'organisation de la chose publique des nations policées.

Parmi ces libertés individuelles, le droit à la vie privée et à sa protection apparait historiquement assez tôt, mais uniquement dans son principe. L'apparition de la photographie, notamment, sera à la source d'études fondatrices comme l'article « *the right of privacy* » de Samuel D.Warren et Louis D. Brandeis, posant en 1890 les fondements du droit des individus à protéger leur image et, de manière plus générale, leur vie privée [WB90].

La délimitation du droit à la vie privée (et à sa protection) n'est pas un problème trivial, car cette notion dépend de la culture, de l'histoire locale et des sensibilités individuelles. La conscience collective d'un besoin de protection de la vie privée peut notamment être influencée par la mise au grand jour de failles significatives dans cette dernière. En France par exemple, le scandale du projet de fichage SAFARI, dans les années 1970, constitue une des motivations principales de la rédaction de la loi « Informatique et Libertés » [Ré78]. Une fois définie, la protection de la vie privée n'est pas plus simple à assurer. En effet, les usagers maitrisent rarement la formulation spécifique des textes réglementaires. Leur conception de ce qui est protégé ou pas, et dans quelle mesure, peut rester très floue.

Problématique

L'avènement de l'ère numérique, dans le dernier quart du vingtième siècle, a déclenché de nouvelles réflexions sur le sujet, les nouveaux outils introduisant de nouveaux risques. De la même manière que la photographie a permis la propagation des images, Internet et les applications distribuées d'une manière générale permettent le partage, la duplication, le traitement automatisé de nombreux types d'informations. Le public prend assez rapidement conscience des possibilités offertes par les nouveaux développements techniques mais également des risques parallèlement encourus par leurs données personnelles. Néanmoins, il reste un fossé entre le droit formellement exprimé, son

application plus ou moins stricte par les acteurs du monde informatique et sa compréhension par les usagers, souvent rebutés par la forme des textes.

Les exemples du commerce électroniques (auxquels nous nous intéressons) et des réseaux sociaux, applications emblématiques associées à deux phases d'expansion du réseau internet, permettent d'illustrer ces difficultés. Dans le premier cas, l'utilisateur est forcé, pour acquérir un produit ou un service, de transmettre des informations personnelles et potentiellement sensibles, comme son adresse ou son numéro de carte de crédit au fournisseur de service (SP). Dans les réseaux sociaux, les utilisateurs sont incités, afin de profiter au mieux de l'environnement personnalisé, à dévoiler énormément d'informations sur eux-mêmes, sans toujours mesurer le risque associé.

Diverses solutions techniques sont proposées pour améliorer la protection de la vie privée en générale et des données personnelles en particulier (section 1.4 du chapitre 1). Seulement, ces propositions se réfèrent souvent à des politiques de sécurité qui ne sont pas directement reliées aux contraintes légales ou réglementaires pesant sur le contexte d'exécution du traitement.

D'une manière générale, les utilisateurs manquent de moyens pour:

- fournir une représentation des textes préservant leurs données personnelles, et les mettant en relation directe avec les différents contextes d'usage des données. En effet, les contraintes légales à appliquer sur les données personnelles diffèrent selon leur nature, leur destinataire, et leur usage en cours. Ainsi, nous devons disposer de moyens de présentation des données et des cas d'usage, nous permettant de faire un raisonnement sur les données, leur destination et les contraintes qui s'appliquent sur elles. Une partie des contributions que nous allons formuler devront donc être centrées autour de cette question de la *représentation* et du *raisonnement*, qui font majoritairement défaut aux solutions actuelles.

- les impliquer dans le processus de protection de leurs données personnelles, en leur donnant davantage de *contrôle* sur l'usage et la transmission de ces dernières. Par le biais du principe de « *collecte minimale* » qu'elle introduit, la directive européenne 95/46/CE (section 1.1.2.2 du chapitre 1) garantit aux utilisateurs un contrôle fort sur leurs données personnelles, à la fois en leur permettant de définir les données à révéler, et de restreindre l'usage fait de ces données. Cependant, il n'existe pas de solutions techniques leur donnant le contrôle sur l'usage associé à leurs données personnelles et la transmission de ces dernières.

La directive européenne 95/46/CE [The95] apporte des réponses à ces besoins :

- Elle introduit le concept de « *collecte minimale* » visant à contrôler l'usage et la transmission des données personnelles. Les implémentations pratiques de ce principe manquent malheureusement de flexibilité. Elles sont souvent limitées à des cases à cocher avec des questions incompréhensibles, ou à la lecture de politiques de protection des

données et des accords EULAs (End User Licence Agreements) contenant des termes complexes. Ces implémentations sont restreintes à des demandes de consentement et acceptation des termes des politiques. Elles ne fournissent pas une *transparence* sur les pratiques utilisées par le SP, ni de *contrôle* sur le futur usage des données personnelles.

- la *transparence*, est liée à la première problématique autour de la question de la *représentation* ;
- alors que le *contrôle* sur les pratiques d'usage est lié à la problématique du *raisonnement*. Il est exprimé par un pouvoir de contrôle automatique des *conditions d'usage* associées aux données, et/ou la capacité de pouvoir négocier ces conditions avec le SP ;

❖ Le concept de « *collecte minimale* » des données personnelles, défini dans la directive européenne 95/46/CE (section 1.1.2.2), est interprété du côté des usagers comme un concept de « *transmission minimale* » des données. Ce principe est acceptable par l'ensemble des entités impliquées dans le cycle de vie des données personnelles (utilisateur, SP, autorité légale, …etc). En revanche il reste toujours difficile à mettre en place techniquement.

A titre d'exemple, dans plusieurs cas, il est nécessaire que le contrôleur dispose d'un identifiant unique lié à l'utilisateur. Ce besoin est translaté dans la conception du système par une requête de numéro de carte bancaire, un numéro de sécurité sociale, ou d'autres identifiants uniques. Quoique de telles translations semblent correspondre au principe de la minimisation des données personnelles, ceci génère deux problèmes. Premièrement ces identifiants encodent souvent d'autres données personnelles, comme la date de naissance, le sexe, …etc. Plus problématique encore, est le fait qu'au lieu d'avoir juste un identifiant unique, le contrôleur possède un identifiant unique mais d'une personne réelle, ce qui peut être utilisé ultérieurement dans la corrélation des données depuis plusieurs sources par exemple. Des problèmes similaires peuvent se poser dans le cas de paiement ou de la vérification de l'âge.

De ce fait, il y a besoin de s'assurer que les données révélées sont vraiment minimales dans le sens où elles ne fournissent aucune information qui n'est pas explicitement demandée de façon obligatoire par le SP et autorisée par le cadre réglementaire.

Pour aucune de ces problématiques, il n'existe actuellement de solution pleinement satisfaisante, alors que cette étape nous semble fondamentalement nécessaire pour faire de l'informatique un outil respectueux des libertés individuelles.

Approche

Afin de répondre à ces problématiques, nous avons posé dans un premier temps le cadre légal français et européen dédié à la protection de la sphère privée en général, et les données personnelles en particulier.
L'analyse des textes légaux et réglementaires de ces cadres législatifs, nous a permis de classer en six axes (section 1.3.1) les éléments constitutifs des politiques de protection des données personnelles.

Notamment il nous a fallu **présenter ces axes sous forme de contraintes de traitement** à respecter, avant de pouvoir nous appuyer dessus pour définir un langage de politiques de protection des données. Ainsi, nous garantissons une ***représentation* des réglementations** en vigueur, afin de s'adapter dynamiquement à celles-ci.

Ensuite, nous nous sommes donnés comme objectif, de trouver des **solutions techniques permettant d'assurer l'implémentation de ces contraintes selon le principe de *« collecte minimale »*** introduit par la directive européenne 95/46/CE. Plus concrètement, nous proposons des solutions techniques pour l'application de ce concept selon deux dimensions :

- Le *contrôle* d'usage lié aux données personnelles, qui :
  - examine comment les langages de politiques de protection des données peuvent être couplés aux langages de politiques de contrôle d'accès, afin d'assurer l'application des contraintes légales, et la protection des données personnelles en contrôlant l'usage associé à ces dernières. Une telle solution :
    - offre à l'utilisateur un cadre légal, lui permettant de *définir ses préférences* en termes de protection des données ;
    - évite à l'utilisateur, le nécessaire *examen des contrats* proposés par le SP au regard de ses préférences, et ceci en effectuant une analyse automatique des contraintes associées.

    Cette analyse automatique de conformité, est basée sur des règles prédéfinies visant à contrôler l'accès aux données personnelles. L'établissement de ces règles repose sur un schéma d'attributs associé aux données, qui est souvent différent de celui défini et utilisé par le SP. Ainsi, l'augmentation de la complexité de *spécification* et de *maintenance* des politiques de protection de données, se pose comme un résultat logique de cette incompatibilité.

    Par conséquent, il nous parait pertinent d'examiner l'apport des outils du Web Sémantique, en particulier les ontologies pour *unifier* le schéma d'attributs utilisé via un modèle sémantique. Ce modèle d'informations sert également de modèle de base *reliant* directement les données personnelles de l'utilisateur, les règles XPACML les préservant, avec l'usage qu'il en fait dans un contexte sémantique.

- Tent à restreindre techniquement l'usage fait des données personnelles à un ensemble de politiques conclues lors d'un accord commun entre l'utilisateur et le SP.

- La minimisation des données communiquées au SP qui définit:

  - des *opérations de filtrage* à appliquer sur les données à révéler. Ces opérations sont directement liées à la *justification* fournie pour la collecte des données.
    Les données personnelles peuvent être également sujettes à pseudonymisation, ou de chiffrement par des techniques supplémentaires qui ne sont pas abordées dans cette thèse ;

  - un protocole de *négociation* des données permettant de diminuer les informations communiquées au SP, en agissant sur les combinaisons des données personnelles qui lui sont communiquées. Une modification de la combinaison a lieu, quand cette dernière présente un risque sur la vie privée de l'utilisateur.

Organisation du manuscrit

Ce document s'articule autour de trois parties et cinq chapitres.

Dans la partie I, nous examinons les éléments de l'état de l'art qui nous semblent nécessaires pour atteindre nos objectifs. Le chapitre 1 traite de la vie privée en général. Nous y définissons les termes utiles, nous présentons le contexte légal français et européen et nous examinons comment les propositions existantes dans le domaine de l'informatique traitent généralement du problème de la protection de la vie privée des utilisateurs. Nous y précisons que parmi les différents champs de protection de la vie privée, nous nous intéressons particulièrement à la protection des données personnelles, et nous exposons les axes légaux dédiés à cet effet, en particulier l'axe de justification (section 1.3.1 du chapitre 1).

Dans la partie II comprenant les chapitres 2 et 3 nous étudions :

- l'apport des langages de politiques de protection de données pour la représentation et la formalisation des axes légaux sous forme de contraintes de protection des données ;
- l'apport des langages de politiques de contrôle d'accès dans l'application de ces contraintes de protection de données sous forme de politiques de contrôle d'accès aux données ;
- l'apport des ontologies pour minimiser la complexité de spécification et de maintenance des politiques de protection des données, et dans la mise en relation directe des contraintes de protection avec les usages associés.

Le chapitre 2 introduit les principes, les concepts, et les outils spécifiques aux langages de politiques de protection des données, et langages de politiques de contrôle d'accès. Il étudie leur complémentarité, pour enfin définir notre langage de politiques de protection de données basé sur le concept de contrôle d'accès nommé XPACML. XPACML est présenté dans ses principes, et dans son architecture explicitant les flux de données et

détaillant les différents composants constitutifs. Dans cette même architecture, nous avons défini les composants permettant la mise en œuvre de nos contributions décrites précédemment. Nous concluons le chapitre avec les aspects implémentation du langage XPACML, en particulier le schéma de politique représentant les concepts associés aux contraintes de protection de données. Ce schéma de base est mis à disposition de l'utilisateur, pour en édicter ses préférences en termes de protection des données.

Dans le chapitre 3, nous définissons un modèle sémantique explicitant les relations entre les données personnelles, les contraintes de protection, et les différents types de SPs dédiés à la spécification des différents cadres d'usage possibles des données. Ce modèle sémantique représente un contexte sémantique de protection des données, que nous avons mis en place avec un double objectif :

- celui de *définir les liens* potentiels entre les données, les contraintes, et les différents usages possibles ;
- celui d'*unifier le schéma* d'attributs à appliquer sur les données entre l'utilisateur et le SP, dans une perspective de simplification de spécification, et de maintenance de politiques de protection de données.

Dans la partie III comprenant les chapitres 4 et 5, nous nous attachons à donner à l'utilisateur davantage de *contrôle* sur la *collecte* et la *transmission* de ses données personnelles en nous basant sur les contributions précédentes, et ceci selon les deux dimensions mentionnées à la fin de la section précédente.

Le chapitre 4 traite la dimension associée au *contrôle d'usage*. Nous proposons un protocole de *négociation des politiques* de protection de données entre l'utilisateur et le SP, en nous basant sur les risques associés aux différents types d'usage proposés par le SP.

Le chapitre 5, traite la dimension liée à la minimisation des données, en explorant deux voies :

- La première, en proposant un protocole de *négociation des combinaisons de données* personnelles à fournir au SP. Nous examinons dans cette partie l'apport de la théorie des jeux dans la conception de ce protocole en nous basant sur la notion de gain et d'utilité de chacun des acteurs impliqués dans la négociation.
- La deuxième en utilisant l'axe « *justification* » (section 1.3.1 du chapitre 1) pour définir des *opérations de filtrage* sur les données à révéler. Ces opérations de filtrage peuvent :
    - porter sur des données atomiques : comme cette partie est associée à la définition de concepts et vocabulaires dédiés à la protection des données, nous l'avons intégrée à la définition du langage XPACML (section 2.6.2.2 du chapitre 2) où nous traitons l'établissement des règles d'accès sur des données atomiques ;

- porter sur des combinaisons de données : nous l'avons intégré dans le chapitre 5, comme alternative avant le passage au protocole de négociation de données.

Dans la conclusion, nous présentons une synthèse de nos contributions, ainsi qu'un ensemble de perspectives de travail ouvertes par nos travaux, qui posent de nouvelles questions dans le domaine de la protection des données personnelles et de la gestion de la sphère privée.

PREMIERE PARTIE

CHAPITRE 1

VIE ET PROTECTION DES DONNEES PERSONNELLES

Sommaire

1 Vie privée et protection des données personnelles 30
Introduction 30
1.1 La sphère privée 30
1.1.1 Nomenclature et définitions 30
1.1.2 Aspects légaux du droit à la sphère privée 32
1.1.3 Protection de la sphère privée au quotidien 37
1.2 Les dimensions techniques de protection de la vie privée sur Internet 39
1.2.1 Gestion des identités 39
1.2.2 Communications IP anonymes 40
1.2.3 Accès anonymes aux services 40
1.2.4 Autorisation préservant la vie privée 40
1.2.5 Gestion des données personnelles 41
1.3 La protection des données personnelles 42
1.3.1 Les six axes de la protection des données personnelles 42
1.3.2 Problématiques spécifiques aux domaines d'intérêt 44
1.4 Technologies de protection des données personnelles 46
1.4.1 Platform for Privacy Preferences 46
1.4.2 Sticky policies 47
1.4.3 Gestion déportée des données sensibles 48
1.4.4 Agents utilisateurs 49
1.5 Discussion 49
1.5.1 Représentation et raisonnement 49
1.5.2 Manque de contrôle 50
1.6 Conclusion 50

1 Vie privée et protection des données personnelles

Introduction

Dans ce premier chapitre, nous nous intéressons aux notions de vie privée et de protection des données personnelles telles qu'elles sont considérées dans le domaine de l'informatique. Nous étudions la nature de ces concepts, la manière dont ils sont traités dans la réglementation ainsi que les moyens techniques qui leur sont rattachés. L'objectif de cette thèse n'est pas de décrire en détail les textes règlementaires, mais identifier les éléments essentiels à considérer dans les solutions techniques.

1.1 La sphère privée

Nous commençons par nous intéresser à la notion de vie privée en général, en-dehors du cadre informatique. Il nous faut clarifier les termes que nous allons utiliser, avant de nous pencher sur les problèmes qu'ils peuvent poser du point de vue du « législateur » et du point de vue individuel.

1.1.1 Nomenclature et définitions

Les notions liées à la vie privée peuvent difficilement être définies sans s'intéresser aux différentes significations du terme *privacy* en anglais. En effet, si l'on peut le faire correspondre en français à la notion assez générale de « caractère privé » d'une chose, ce mot semble être considéré comme recouvrant un certain nombre de concepts liés. Suivant leur culture et leur point de vue, les auteurs les plus consciencieux prennent soin de préciser ce que désigne pour eux le terme *privacy*, sans l'utiliser indifféremment pour le droit à la vie privée (*right of privacy)* ou la protection de la vie privée (*privacy protection)*. Nous considérerons ici le terme *privacy* comme une traduction imparfaite de l'expression « vie privée », nous autorisant par la suite à définir des termes dérivés.
La mention du concept de *vie privée* éveille chez tout un chacun un ensemble de problématiques liées à notre vie quotidienne ou à notre perception de procédés techniques ou liées à un certain contexte professionnel. Ainsi, la capacité à cacher un certain nombre de choses sur soi au public en général, à des collègues, à des connaissances, relève nécessairement de notre droit à la vie privée. La notion de surveillance des activités d'un individu, l'enregistrement ou le traitement d'informations le concernant, le fait d'entrer en communication avec lui sur la base des résultats d'un tel traitement sont autant d'actions en lien étroit avec la notion de vie privée ou de sphère privée. Le concept semble donc composite et par conséquent difficile à cerner. Néanmoins, certains auteurs ont proposé des

définitions très restreintes dont on peut se demander si elles correspondent vraiment à cette vision naïve et intuitive de la vie privée.

En 1967, Alan Westin définit ainsi le terme privacy [Wes67], adoptant un point de vue de type juridique, confondant en un seul et même mot la chose et le droit à la chose : *"Right of individuals to determine for themselves when, how and to what extent information about them is communicated to other."*
La vie privée est donc essentiellement une question de gestion des flux d'informations se rapportant à une personne physique. Günter Müller adopte un point de vue assez similaire mais plus abstrait [Mü06] en définissant ainsi la privacy : *"Possibility to control the distribution and use of personal data."*
Ces deux points de vue semblent limiter la notion de vie privée à une diffusion maîtrisée d'informations. Cela peut paraître frustrant au premier abord au vu des procédés relativement complexes et détachés de toute considération technique que nous venons de mettre en relation avec la notion de vie privée.
Certains auteurs partent de ce constat pour poser des définitions à vocation plus générale, incluant certaines de ces notions. Ainsi, Sara Baase propose par exemple dans son livre The gift of fire [Baa03] une acception légèrement différente et sans doute plus « parlante », suivant laquelle le mot *privacy* peut signifier indifféremment :

- L'absence d'intrusion ;
- Le contrôle des informations nous concernant[1] ;
- L'absence de surveillance.

Nous incluons donc ici explicitement deux nouveaux processus dans le concept de vie privée, processus qui semblent se rapporter à notre tentative de description intuitive de la vie privée. Il en ressort donc légitimement une certaine satisfaction. Cependant, cette circonscription de la notion de vie privée peut également paraître peu naturelle à cause d'un certain télescopage conceptuel introduit par ces nouvelles notions. En effet, l'intrusion comme la surveillance sont craintes en tant qu'elles sont des menaces pour le contrôle de nos informations. Cette tentative de définition, comme nombre d'autres, amène à penser que les concepts de la vie courante que nous associons instinctivement à la notion de vie privée se réduisent effectivement tous à un problème de contrôle de l'information. Il est aisé en effet de considérer un à un les éléments que nous avions inclus dans la notion de vie privée, pour les exprimer sous la forme d'une manipulation d'informations.
Ceci posé, nous pouvons formaliser quelques définitions liées à la vie privée à partir des éléments déjà recueillis, afin de faire référence par la suite à des concepts clairs et distincts.
Plutôt que d'utiliser le terme de « vie privée » qui, nous l'avons vu, peut s'avérer assez vague à cause même de son écho intuitif, nous prendrons appui sur le concept équivalent de sphère privée, notamment formalisé par Ludivine Crépin et al. [CVJ08].

Définition 1.1 (Sphère privée). La sphère privée d'un individu est l'ensemble des informations se rapportant à lui-même, qu'il considère comme sensibles et donc dignes d'être protégées. Cette sphère est personnelle (l'individu est le propriétaire des informations qu'elle contient), personnalisable (l'individu décide des informations qu'elle contient), dynamique (les informations peuvent y être ajoutées ou en être retirées) et

[1] Dans ce contexte, lesdites informations peuvent être de deux types : ou bien des faits des actions ayant été exécutées) ou bien des données personnelles (comme des identifiants, des informations d'′etat civil ou des caractéristiques physiques).

dépendante du contexte (les informations qu'elle contient peuvent, en nature et en nombre, dépendre du temps, des activités de l'individu ou d'autres paramètres).

Nous disposons, avec le concept de sphère privée, d'un outil aisément manipulable pour traiter de vie privée. La sphère privée encapsule donc toutes les informations, explicitement représentées ou non, qui nous concernent et que nous souhaitons protéger pour quelque motif que ce soit.

Définition 1.2 (Droit à la vie privée). Le droit à la vie privée d'un individu est sa prétention aux caractères personnel, personnalisable, dynamique et contextuel de sa sphère privée ainsi qu'au contrôle de la diffusion, de l'utilisation et de la conservation des informations contenues dans sa sphère privée, quelles que soient la représentation de ces informations et la localisation de cette représentation.
Nous incluons ainsi explicitement dans le droit à la vie privée la notion de propriété des données, fortement liée à celle du contrôle.

Définition 1.3 (Protection de la vie (ou de la sphère) privée). La protection de la vie privée est l'ensemble des mesures techniques visant à assurer le respect du droit à la vie privée.

1.1.2 Aspects légaux du droit à la sphère privée

Ce que recouvre l'expression "droit au respect de la vie privée" est fort difficile à cerner. Ce droit assez flou, prend ses sources dans les droits liés à l'individu et à la propriété privée. Deux points de vue particuliers s'opposent sur la nature philosophique de ce droit. En 1890, dans leur essai « *The right of privacy* », Samuel D. Warren et Louis D. Brandeis identifient le droit à la vie privée comme un droit à part, nécessitant une nouvelle protection juridique appropriée [WB90]. Dans cette étude, le droit à la vie privée, s'appliquant aux personnes physiques, leur permet d'interdire la publication d'informations ou de photographies les concernant. En effet, les auteurs dissocient la vie privée et les données personnelles, des personnes concernées. L'objet de ce droit est ainsi distingué de son titulaire: ce sont les personnes qui sont protégées, et non les données.
Judith J. Thomson au contraire, dans sa recherche des fondements du droit à la vie privée, considère que ce n'est pas un « nouveau » droit, mais un droit dérivé qui procède du droit à la propriété privée, des droits attachés au corps d'un individu et du droit des contrats [Tho75]. Dans ces deux points de vue distincts, nous voyons apparaître d'une part la notion de propriété implicitement appliquée à une information, idée effectivement fondamentale dans la notion de droit à la vie privée, et d'autre part le concept essentiel de consentement de la personne physique dont la vie privée est mise en question.
Dans un contexte plus pragmatique et contemporain, le droit de la vie privée a été intégré dans la plupart des cadres légaux nationaux à partir de la fin du XVIIIe siècle, et adapté à l'évolution de l'environnement technique en matière de gestion de l'information au cours des trente dernières années. Les textes concernés présentent une vision de la sphère privée axée sur des impératifs économiques très contextualisés, et remise à jour au fil de l'évolution des technologies et des menaces.
Dans le cadre de nos travaux, nous nous intéresserons, à titre d'exemple, au cadre légal existant en France. Depuis 2004, il est globalement accepté que le droit français en matière

de protection de la vie privée est compatible avec la législation de la plupart des pays de l'Union Européenne (se situant parmi les plus protectrices, derrière l'Espagne notamment), ou avec celle du Canada, par exemple. La législation fédérale américaine, par contre, est très différente et la présente étude ne saurait en constituer une illustration représentative.
En France, depuis la création du code civil en 1803, le droit des individus à la vie privée est affirmé par son article 9 [Ré03], mais de manière générique et appelant une interprétation appuyée au regard des moyens de traitements mis à disposition par le développement de l'informatique. En effet, les moyens techniques actuels permettent la mise en œuvre de collectes et de traitements automatisés des données personnelles, contexte qui n'était pas envisageable en 1803. Cette adaptation débute en 1978 par une loi nationale [Ré78] et va se poursuivre dans le cadre de l'Union Européenne. L'essentiel du cadre législatif en matière de protection de la vie privée réside dans deux lois nationales [Ré78, Ré04], reprenant deux directives européennes [The95, The02].
Il est à noter que ces textes contiennent tous des limitations au droit à la vie privée, concernant notamment l'instruction des enquêtes judiciaires. Nous ne nous intéresserons pas au cas particulier des restrictions pouvant être apportées par des juges à la protection de la vie privée, pour nous situer plutôt dans le contexte de traitements informatiques ne faisant pas directement suite à un contentieux ou à une mesure de protection de la sécurité nationale.

1.1.2.1 La loi française 78-17 « Informatique et Libertés »

La spécificité des risques induits par les traitements automatisés des informations (et notamment ceux mis en œuvre par les administrations publiques), a motivé la création d'une nouvelle loi en 1978. La France est alors le premier pays européen à inclure dans son droit national des dispositions spécifiques concernant les traitements informatiques de données personnelles. La loi 78-17 du 6 janvier 1978 [Ré78], « relative à l'informatique, aux fichiers et aux libertés » (dite loi « Informatique et Libertés »)[2] parle de traitements automatisés sur des données nominatives. Ce texte pose des principes qui sont depuis rentrés dans la culture française, de par les mentions légales apparaissant systématiquement dans les formulaires de collecte de données.
Les données nominatives sont définies comme celles pouvant être rattachées à une personne physique (que nous appellerons ici « l'intéressé »), de manière directe ou indirecte. Cette précision permet d'englober les données pouvant être liées à une personne après recoupement avec d'autres informations. Le responsable d'un traitement automatisé ou d'une base de données contenant ce type d'informations ne les possède pas, mais est responsable de leur sécurité. Le texte introduit l'obligation fondamentale d'informer l'intéressé sur divers points concernant le traitement et de recueillir son consentement (sauf dans certains cas particuliers prévus par la loi, comme par exemple l'accomplissement d'une mission de service public). L'intéressé jouit également d'un droit d'accès et de rectification des données collectées, exerçable auprès du responsable du traitement.
La loi impose une limitation sur la durée de conservation des données. Celles-ci devront en effet être détruites une fois qu'elles ne sont plus nécessaires au traitement. Cette

2 Dans le cadre de nos travaux, lorsque nous nous référerons à la loi 78-17 [Ré78], nous considérerons qu'il s'agit de la version originalement promulguée, et non pas de la version consolidée (considérablement modifiée par la loi 2004-801 [Ré04]).

disposition, naturellement générale, a appelé par la suite à diverses réglementations spécifiques à certains secteurs d'activité. Les opérateurs de télécommunications, notamment, sont soumis à des directives précises concernant la durée de conservation des informations sur les communications de leurs abonnés.
La transmission à des tiers des données collectées et/ou traitées est également réglementée. Elle est par défaut interdite, sauf si l'intéressé a au préalable donné son consentement pour le transfert à un tiers ou à un ensemble de tiers identifiés.
La loi 78-17 crée également la Commission Nationale de l'Informatique et des Libertés (CNIL), garante des droits exprimés dans le texte (notamment le droit d'accès). D'une manière générale, les traitements automatisés portant sur des informations nominatives doivent faire l'objet d'une déclaration à cette autorité, qui accorde une autorisation préalable à la mise en œuvre de la collecte et du traitement.

1.1.2.2 La directive européenne 95/46/CE

La directive européenne 95/46/CE « relative à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données » [The95] est le premier et le principal texte réglementaire de l'Union européenne en matière de protection de la vie privée. Dans ce texte, le vocabulaire change par rapport à la loi 78-17, qui a alors presque vingt ans : on ne parle plus de données nominatives mais de données personnelles. Cette nomenclature prévaudra par la suite dans les textes réglementaires français.
Un des premiers points précisés par cette directive est que la collecte des données doit être loyale. Ainsi, il est généralement interdit de collecter des données à l'insu de l'utilisateur ou contre son gré. De plus, il est spécifié que les données collectées doivent être « adéquates, pertinentes et non excessives au regard » de « finalités déterminées, explicites et légitimes ».
Ainsi, on introduit un lien fort entre les données collectées et le but du traitement, sa justification : il est interdit de collecter des données superflues, non nécessaires, non directement liées au but poursuivi. Cette notion fondamentale est un axe de travail fort pour la protection de la vie privée, elle introduit le concept de collecte minimale de données, qui sera par la suite interprété du côté des usagers comme un concept de transmission minimale de données, préservant le caractère privé de ses informations. Ces aspects de la protection de la vie privée sont également désignés dans la littérature sous les termes de principe de proportionnalité et de principe de finalité. Le texte introduit des dispositions particulières pour interdire (dans la plupart des cas) les traitements portant sur des données très sensibles, comme les convictions religieuses, l'orientation sexuelle ou les informations relatives à la santé.
Au-delà de ces nouvelles notions, la directive réaffirme les principes de base déjà présents dans la loi française. Notamment, elle renforce la protection vis-à-vis de la diffusion des données, en imposant une nouvelle phase d'information de l'intéressé lorsque ses informations sont utilisées par un tiers. Les transferts de données à destination de pays tiers sont également réglementés et limités aux pays pouvant justifier d'un « niveau de protection adéquat ».
Concernant le contrôle de l'application des diverses dispositions, la directive impose à chaque pays membre la création d'une autorité de contrôle nationale (la CNIL en est le représentant français), et leur regroupement en un « groupe de protection des personnes à l'égard du traitement des données à caractères personnel ». Cette organisation européenne est communément appelée « groupe de l'article 29 » ou plus couramment encore G29. La

directive prévoit que les traitements doivent faire l'objet d'une notification préalable à l'autorité de contrôle.

1.1.2.3 La directive européenne 02/58/CE

La directive européenne 02/58/CE « concernant le traitement des données à caractère personnel et la protection de la vie privée dans le secteur des communications électroniques » [The02] est un exemple d'application sectorielle des principes de la directive de 1995. Elle définit des dispositions techniques de protection de la vie privée spécifiques au secteur des télécommunications. Elle reprend chacun des principes de base, pour les transcrire dans ce cas d'application particulier.
Un des intérêts de cette directive est qu'elle traite des transmissions électroniques non sollicitées (spam en anglais), en établissant clairement une priorité de l'opt-in dans l'union européenne : si l'intéressé n'a pas explicitement donné son consentement pour être contacté par une société particulière, alors cette société ne peut utiliser ses informations personnelles pour lui faire une offre commerciale. Néanmoins, si l'intéressé a déjà eu recours aux services d'une société, la directive considère qu'il est légitime que celle-ci le recontacte pour lui proposer des services semblables. Les communications non sollicitées sont un cas particulier de l'application des principes de la directive 95/46 (ils dérivent des principes de justification, de consentement et de transmission aux tiers des informations), mais constituent un problème social et technique suffisamment présent pour justifier une législation dédiée et contextualisée.
Ce texte réaffirme également le principe de la collecte minimale d'informations introduit par la directive de 1995 (et dérivant du principe de justification). Cette directive interdit la pratique du spoofing, qui consiste à se faire passer pour une autre entité en forgeant une adresse ou un numéro téléphonique, par exemple, dans un but de prospection.
L'usurpation d'identité, que ce soit celle d'une personne physique ou morale, est un acte illicite qui constitue une atteinte particulière aux droits moraux de la personne concernée.
Un autre point intéressant est soulevé dans l'article 14-3, qui stipule que « des mesures peuvent être adoptées afin de garantir que les équipements terminaux seront construits de manière compatible avec le droit des utilisateurs de protéger et de contrôler l'utilisation de leurs données à caractère personnel ». La législation européenne garantit donc théoriquement les utilisateurs contre la construction de matériels électroniques ou informatiques trop intrusifs, ou de manière générale présentant un risque pour leur vie privée.

1.1.2.4 La loi française 2004-801

La loi française numéro 2004-801 « relative à la protection des personnes physiques à l'égard des traitements de données à caractère personnel » [Ré04] a pour principal objectif de modifier la loi 78-17 déjà existante, pour la mettre en conformité avec les directives européennes de 1995 et 2002. La France est alors le dernier pays européen à transposer la directive de 1995.
Cette loi entérine le terme *d'informations personnelles* en remplacement des *informations nominatives*. La notion de justification de la collecte et du traitement (principe de collecte minimale) est transposée dans le texte français en utilisant les termes de la directive de 1995. Le texte de loi consolidé modifie également les pouvoirs de la CNIL. En particulier,

l'étendue de la déclaration préalable est largement diminuée, et la commission dispose d'un rôle de contrôle plus important, assorti d'un pouvoir de sanction.
Ce texte déclencha en 2004 une polémique entre parlementaires et défenseurs des libertés individuelles. Ces derniers considéraient que la nouvelle loi affaiblissait le texte original, en particulier par le statut particulier qu'elle accorde aux traitements mis en œuvre par l'administration publique. La principale contribution de cette loi aux principes généraux de la protection des données à caractère personnel dans le cadre des traitements automatisés consiste en l'introduction dans la loi française de la notion de justification. Pour le reste, elle reformule la loi existante pour l'harmoniser avec la directive européenne et redéfinit les instances de contrôle.

1.1.2.5 Synthèse

Lois et directives déterminent donc ce qui peut ou ne peut pas être fait des données informatiques susceptibles d'être rattachées à un individu particulier. Les textes réglementaires que nous avons abordés réduisent la sphère privée d'un individu à sa partie observable par un système électronique ou informatique, aux informations qui peuvent être extraites, représentées et reliées entre elles. Ces textes constituent donc une projection du principe général exprimé par le Code Civil sur le domaine de l'informatique et des télécommunications.
Il peut être intéressant ici de définir un nouveau terme aux acceptions plus restreintes. Ce que nous nommons « protection des données personnelles » sera donc, dans le cadre de nos travaux, une limitation de la protection de la vie privée à certaines données numériques. Cette acception pourra inclure des informations fournies par l'utilisateur, par son usage de l'outil informatique, ou par ses transactions avec un tiers. Elle exclura toutes les informations implicites, non représentées, qui constituent la sphère privée de l'utilisateur.

Définition 1.4 (Protection des données personnelles). La protection des données personnelles consiste en l'ensemble des mesures techniques visant à assurer le respect du droit à la vie privée, limitées aux données de la sphère privée d'un utilisateur explicitement représentées sous forme numérique et mises en jeu dans le cadre d'une application informatique.
Concernant cette protection des données personnelles, l'étude de ces textes fait ressortir six thèmes clés, suivant lesquels s'alignent les diverses réglementations. Ces thèmes sont : *l'information de l'utilisateur*, son *consentement*, son *droit d'accès et de modification*, le principe de *justification de la collecte et du traitement*, la *conservation des données et leur transmission à des tiers*. Cette partition des réglementations que nous introduisons ici appelle à une étude plus approfondie, visant à spécifier chaque axe de manière claire et distincte. Nous reviendrons dessus plus loin dans ce chapitre (section 1.3.1).
Avant cela, il faut bien noter qu'en se restreignant ainsi au monde numérique, on reconnaît que l'on ignore délibérément tous les éléments de la sphère privée qui ne sont pas directement représentables dans un fichier informatique. Cette frontière reste cependant floue et a tendance à se déplacer au fur et à mesure que la technologie permet d'inférer davantage de faits et de règles du comportement d'un individu ou d'un groupe d'individus. Il apparaît donc maintenant comme clair que les données personnelles ne sont qu'une partie de la sphère privée. La protection de ces données ne constitue qu'une partie de la protection du droit à la vie privée, même restreinte au domaine numérique. Cette conclusion appelle

donc à une classification précise des différents domaines qui pourraient constituer la protection de la vie privée et la protection des données personnelles.

1.1.3 Protection de la sphère privée au quotidien

La protection de la vie privée, on l'a vu, n'intervient donc pas exclusivement dans le cadre d'applications informatiques. La notion de sphère privée apparaît dans toutes les activités humaines à partir du moment où elles ont une dimension sociale. Le public y est d'ailleurs de plus en plus sensibilisé, même s'il ne semble pas toujours donner sa juste valeur au caractère privé de ses informations personnelles. L'émergence d'Internet est en grande partie responsable de l'évolution des inquiétudes du grand public quant à la protection de sa vie privée en général, et de ses données sensibles en particulier. Les individus mettent potentiellement en danger leur vie privée lorsqu'ils interagissent avec une société pour en obtenir un produit ou un service, lorsqu'ils utilisent des équipements électroniques communicants ou lorsque des informations particulièrement sensibles, comme les données médicales d'un patient, sont confiées à des traitements automatisés. L'évolution de la structure des applications et des usages individuels, allant vers davantage de collaboration, de délocalisation, de fluidité et de transparence des données, contribue également de manière significative à cette évolution de cette prise de conscience du grand public, cristallisée dans quelques domaines en particulier. Divers domaines de la vie courante peuvent donc servir à l'élaboration de scénarios et de cas d'utilisation dans lesquels se posent des problèmes de protection de la vie privée, ou des données personnelles plus précisément.

1.1.3.1 Domaine de la santé

Les données de nature médicale appartenant aux utilisateurs (aux patients) sont perçues par ces derniers comme particulièrement sensibles, car touchant directement à leur corps et à leur intégrité physique, au cœur de tout ce qui peut être considéré comme privé. Malgré cela, les utilisateurs sont de plus en plus amenés à confier ces données à l'outil informatique et à en perdre partiellement le contrôle. La mise en œuvre du dossier médical informatisé (et distribué) est le premier et le plus évident de ces contextes d'informatisation des données médicales. Suivant les pays, cette mise en œuvre prend des formes et des extensions diverses, mais les questions de fond qui se posent (et qui conditionnent le choix des procédés techniques) sont les mêmes :

- En dépit du fait que le patient est propriétaire de ses données, est-ce que le corps médical n'a pas son mot à dire sur la manière dont elles doivent être traitées ?
- Qui doit avoir accès à ces données ?
- Où, comment et combien de temps seront-elles stockées ?

Les outils techniques pour le maintien à domicile des personnes âgées, handicapées, ou partiellement autonomes d'une manière générale, soulèvent également des questions spécifiques. Ce type d'application mêle en effet les problématiques de protection des données personnelles liées aux informations médicales (qui sont par exemple susceptibles d'être transmises automatiquement en cas d'urgence) à des problèmes de protection de la vie privée que l'on va retrouver dans le domaine de l'intelligence ambiante.

1.1.3.2 Domaine de l'intelligence ambiante

L'intelligence ambiante[3] met en œuvre de multiples équipements électroniques et terminaux informatiques, typiquement mobiles et de petite taille, dans un environnement au sein duquel ils communiquent, collaborent et coopèrent de manière quasiment transparente pour l'utilisateur humain. Les équipements de l'environnement disposent de capacités d'adaptation, de raisonnement, de mémorisation limitées uniquement par leurs faibles ressources physiques. Les données échangées entre les diverses entités sont donc constamment soumises à des traitements essentiellement distribués. Les objectifs de ces nombreuses interactions peuvent être multiples et sont souvent organisés autour de la personnalisation et de l'augmentation de l'expérience de l'utilisateur, dont l'utilisation du système (bâtiment ou environnement de travail par exemple) doit être facilitée. Les objets de l'environnement interagissent constamment entre eux, échangent des informations et produisent des traces qui peuvent être agrégées en profils comportementaux.
Ces données sont susceptibles d'être partagées de manière intensive, dans le but d'améliorer la fluidité de l'expérience de l'utilisateur au sein de l'environnement intelligent.

L'utilisation des composants radiofréquence RFID, en particulier, a été particulièrement étudiée pour sa capacité à diffuser et à créer des informations potentiellement personnelles et sensibles [JL02, SB05]. Ici, le compromis entre aisance d'utilisation et protection de la vie privée joue à plein, et les informations de l'utilisateur se retrouvent dans une situation particulièrement risquée. Typiquement, les scénarios d'intelligence ambiante mettent en œuvre des traitements relevant de la protection de la vie privée, mais pas forcément de la protection des données personnelles. En effet, les informations échangées sont souvent implicites par nature pour l'utilisateur. N'ayant pas accès à leur représentation, il lui est plus délicat de s'assurer de leur protection efficace.

1.1.3.3 Domaine du commerce électronique

Les scénarios de commerce électronique peuvent être très variés, allant d'une simple transaction à une collaboration étroite et complexe entre agents commerciaux dépendant d'entités différentes, ou à des techniques de raisonnement élaboré sur les habitudes du client. Dans le cadre d'une transaction commerciale électronique (le plus souvent sur Internet via un site marchand), la communication d'informations personnelles est essentielle. Les informations apparaissant comme sensibles de la manière la plus immédiate et évidente sont les données bancaires de l'utilisateur (qui ici devient un client), nécessaires au paiement de la transaction. Les utilisateurs sont globalement sensibilisés aux risques afférents à la communication d'un numéro de carte bancaire. En France particulièrement, la culture de la carte à puce a amené le public à être encore plus méfiant que dans les pays anglo-saxons, où la communication d'un numéro complet par téléphone est une pratique

3 Carole Adam et al. caractérisent ainsi l'intelligence ambiante [AEG06] :
« L'Intelligence Ambiante désigne un ensemble d'interfaces intelligentes supportées par des technologies de réseau et de traitement des données enfouies dans les objets du quotidien [...]. Son but est d'être attentive aux caractéristiques spécifiques de chacun, de s'adapter aux besoins des utilisateurs, d'être capable de répondre intelligemment à des indications parlées ou gestuelles, et même d'engager un dialogue. Elle doit être non intrusive et le plus souvent invisible pour l'utilisateur au quotidien ».

courante. Pour autant, les informations de nature bancaire ne sont pas les seules à être divulguées lors d'une transaction commerciale. La nature des objets ou des services achetés, agrégés en historique, constitue en elle-même une information révélatrice du mode de vie de l'utilisateur et d'une valeur commerciale indiscutable pour une entreprise. Les informations nécessaires à la livraison devraient également être considérées avec attention, ainsi que les identifiants de la connexion et du terminal de l'utilisateur. Les transferts d'informations personnelles sont une nécessité absolue pour les applications commerciales, et l'utilisation qui en est faite est rarement maîtrisée. Les possibilités de négociation avec un service informatique étant actuellement quasiment nulles, l'utilisateur peut être contraint par la nécessité de confier plus d'informations qu'il ne le devrait ou le voudrait. Il est donc alors de fait dessaisi d'une partie de son contrôle sur ses données personnelles. Nous avons choisi de définir nos contributions dans ce domaine applicatif.

1.2 Les dimensions techniques de protection de la vie privée sur Internet

Yves Deswarte et Carlos Aguilar Melchor proposent en 2006 une classification en cinq domaines des techniques de protection de la vie privée sur Internet [DA06]. Nous reprenons ici cette classification, orientée par des considérations technologiques. Les exemples sont principalement ceux fournis par Deswarte et Aguilar Melchor.

1.2.1 Gestion des identités

La gestion des identités recouvre les technologies qui permettent à l'utilisateur de masquer sa véritable identité lorsque la connaissance de celle-ci n'est pas nécessaire au traitement[4]. L'objectif ici recherché est d'empêcher la corrélation des différentes activités de l'utilisateur qui ne souhaite pas forcément que plusieurs utilisations successives d'une même application puissent être comparées, ou encore que différents prestataires de services s'entendent pour partager des informations sur son profil d'utilisation des ressources. La principale solution technique à ce problème est le concept d'identité virtuelle. L'utilisateur en possède plusieurs, correspondant à différentes applications, différents profils comportementaux ou différents niveaux de sécurité. Ces profils doivent être suffisamment anonymes pour qu'il soit difficile d'établir des corrélations entre eux. Typiquement, chacun d'eux est identifié par une chaîne de caractères arbitraire (le pseudonyme) qu'il n'est possible de relier à l'utilisateur que sous certaines conditions. La plupart du temps, seul l'utilisateur lui-même (ainsi éventuellement que des tiers de confiance) en est capable.
Le concept d'identité virtuelle est notamment repris dans le protocole IDsec, développé par l'IETF [Int], ou dans les travaux du projet européen PRIME [Eur08]. C'est également un concept clé dans les architectures fondées sur les Trusted Computing Platforms (TCP) [Tru03].

4 Cette technique se distingue fondamentalement du spoofing en ceci que l'utilisateur n'usurpe pas l'identité d'un tiers, il se contente de ne pas révéler la sienne.

1.2.2 Communications IP anonymes

Plusieurs technologies ont été développées pour permettre la non-traçabilité des communications sur un réseau IP, que ce soit pour des applications pair-à-pair fortement distribuées et à accès essentiellement anonymes, ou bien pour des applications client-serveur avec une notion de session très forte.
Les routeurs MIX (proposés par David Chaum), par exemple, permettent de cacher le lien existant entre les messages entrants et sortants, par l'utilisation de bourrage, de chiffrement aléatoire et de brouillage statistique (émission de messages fictifs). Ces techniques permettent d'éviter qu'un observateur extérieur puisse relier un message entrant à un message sortant en analysant son contenu ou en étudiant les séquences temporelles d'entrées et sorties. La mise en réseau de routeurs MIX (de manière à limiter l'impact de la prise de contrôle de l'un des routeurs par un attaquant et à rendre l'ensemble robuste aux collusions de routeurs) est le principe de base de certains réseaux « anonymes » destinés à des applications pair à pair (comme Tarzan [FM02]) ou de courrier électronique (comme Mixminion [DDM03]), par exemple. Les Réseaux DC (pour Dining Cryptographers) sont pour leur part des réseaux dans lesquels tous les messages sont émis de manière anonyme (la protection de l'émetteur étant garantir par une émission de données, significatives ou non, en continu) et diffusés à tous les membres (ce qui garantit l'anonymat du récipiendaire, qui est cependant le seul à pouvoir déchiffrer le message si son contenu est protégé). Ce type de technique présente toutefois l'inconvénient de gaspiller les ressources réseau, ce qui constitue un frein à son passage à l'échelle.
Les foules et les hordes sont d'autres exemples de technologies visant à assurer l'anonymat des utilisateurs navigant sur le web, par le biais d'un re-routage aléatoire des requêtes entre les membres de la communauté d'utilisateurs. L'anonymat des requêtes et la possibilité que l'utilisateur reçoive la réponse associée sont garantis par le partage d'une clé de chiffrement secrète par chaque paire d'utilisateurs.

1.2.3 Accès anonymes aux services

Ce domaine concerne l'anonymisation des messages de l'utilisateur au niveau du protocole applicatif. En effet, dans le cas général la communication entre utilisateur et service comporte beaucoup d'informations identifiantes, et il est parfois possible de limiter la portée de ces informations sans détériorer la qualité du service. Une anonymisation efficace semble donc nécessairement passer par des relais mandataires (proxies) applicatifs chargés d'obfusquer les informations sensibles, qui sont souvent spécifiques au service ou à l'application.

1.2.4 Autorisation préservant la vie privée

Il existe des techniques permettant de séparer la phase d'authentification de la phase d'autorisation d'accès, afin que l'utilisateur puisse accéder de manière anonyme au service. Lors de la phase d'authentification, l'utilisateur prouve son identité auprès d'un tiers de confiance (via l'utilisation d'un login et d'un mot de passe, d'un certificat cryptographique ou d'un challenge quelconque). Le tiers de confiance peut également, en fonction du type

de requête et des exigences des fournisseurs de service, vérifier que l'utilisateur respecte certaines caractéristiques. Le tiers émet ensuite une accréditation (credential) affirmant que l'identité de l'utilisateur, ainsi éventuellement que ses autres caractéristiques, ont été vérifiées. L'utilisateur se servira ensuite de cette accréditation (qui ne mentionne pas son identité, mais dont l'authenticité peut être vérifiée auprès du tiers de confiance) pour accéder à des services ou à des ressources.
Ce type de protocole, de la même inspiration que le système de jetons d'accès de Kerberos [Mas], permet ainsi d'une part à l'utilisateur d'accéder à un service sans dévoiler son identité, et d'autre part au gestionnaire du service de se convaincre que l'intermédiaire de confiance a bien vérifié les caractéristiques requises concernant tous les utilisateurs accrédités.
Le principe de l'autorisation préservant la vie privée (utilisant des accréditations anonymes) a été mis en œuvre de diverses manières. Deswarte et Aguilar Melchor mentionnent notamment les applications de e-Cash (fondées sur les travaux de David Chaum), permettant d'effectuer un paiement électronique sans dévoiler son identité, l'architecture à base de clés asymétriques SPKI/SDSI (développée par l'IETF) et l'application IDEMIX (Identity Mixer) développée par IBM.
On peut également mentionner OpenID [Ope], un protocole de plus en plus courant sur le web, issu de l'initiative Identity 2.0 [Ide]. Le système OpenID (intégré à l'interface d'authentification de nombreux sites internet et mis en place côté client par le biais de plugiciels comme Sxipper [Sxi]) permet aux utilisateurs d'enregistrer leur identité sur un serveur OpenID de leur choix. Pour accéder à un site web nécessitant une autorisation, l'utilisateur fournit alors une URI, possédée par le serveur OpenID, qui permet au gestionnaire du site web d'obtenir une accréditation d'identité de la part du serveur OpenID, sans avoir à prendre connaissance ni à enregistrer un login ou un mot de passe. L'utilisateur peut en outre se servir d'un ou de plusieurs serveurs OpenID pour gérer plusieurs identités virtuelles, appelées persona.

1.2.5 Gestion des données personnelles

La gestion des données personnelles couvre les dispositifs de protection des données personnelles. Elle concerne notamment la protection des informations de profil, de personnalisation, les préférences utilisateur transmises à une application, les identifiants et les requêtes faites au nom de l'utilisateur. Deswarte et Aguilar Melchor mettent en valeur un point de droit qui apparaît dans les législations française et européenne : les informations personnelles se rapportant à un individu sont la propriété de cet individu, quel que soit le système sur lequel elles sont stockées. Le propriétaire du dit système n'hérite pas de la propriété des données, mais uniquement de la responsabilité de leur protection.

Les auteurs pointent deux sous-axes : la minimisation de la collecte, qui correspond au principe introduit en 1995 dans la législation européenne (section 1.1.2.2), et l'auto-détermination des données, visant à assurer la protection des données personnelles indépendamment des systèmes et des tiers auxquels elles sont confiées. Les auteurs estiment que l'utilisation d'architectures à base de composants logiciels et matériels « dignes de confiance » (les Trusted Computing Platforms [Tru03a], fondées sur des composants particuliers appelés Trusted Computing Modules ou TPM [Tru03b]) peut être un moyen efficace pour assurer cette dernière propriété. Néanmoins, ils notent également que ces mêmes technologies peuvent aussi être utilisées pour poursuivre des objectifs allant à l'encontre des intérêts de l'utilisateur. Parmi les cinq domaines technologiques identifiés

par Deswarte et Aguilar Melchor, la gestion des données personnelles est manifestement le domaine qui correspond le mieux à la définition (définition 1.4) que nous avons donnée de la protection des données personnelles. Nous considérons que c'est principalement sur le premier axe de protection (minimisation de la collecte) de cette partie de la protection de la vie privée que portent nos travaux.

1.3 La protection des données personnelles

Maintenant que nous disposons du vocabulaire adéquat ainsi que d'une vue d'ensemble sur les familles de dispositifs techniques pour la protection de la vie privée, nous proposons de nous concentrer sur un domaine en particulier, à savoir la protection des données personnelles [PDC06]. Nous essaierons de caractériser comment cet aspect particulier est traité dans les documents réglementaires, nous analyserons quelques-unes des nombreuses propositions techniques s'y rapportant et le mettrons en lien avec les domaines applicatifs déjà identifiés pour la protection de la sphère privée en général.

1.3.1 Les six axes de la protection des données personnelles

L'analyse des textes légaux et réglementaires (ainsi que des recommandations que l'on peut couramment observer dans les chartes d'utilisation de systèmes informatiques) nous a permis de classer en six axes les éléments constitutifs des réglementations en matière de protection des données personnelles. Cette classification vient compléter, de manière transversale, les « critères communs » fixés par la norme ISO/IEC 15408-2, publiée en 1999 [ISO99], décomposant la protection des données personnelles en quatre critères techniques évaluables que doit respecter un système :

- La possibilité pour l'utilisateur d'agir de manière anonyme, de manière qu'aucun autre utilisateur ne puisse l'identifier ;
- La possibilité d'agir sous un pseudonyme, interdisant l'identification directe par les autres utilisateurs mais permettant tout de même de relier l'utilisateur à ses actions ;
- L'impossibilité pour les autres utilisateurs d'établir des corrélations entre les différentes activités de l'utilisateur ;
- La non-observabilité, interdisant aux autres utilisateurs de pouvoir décider si une action est en cours.

Ces « critères » expriment les conditions que doit respecter un système pour garantir la protection de la vie privée de ses utilisateurs, sans toutefois caractériser cette protection elle-même.

Nous proposons une classification qui identifie six axes, suivant lesquels une autorité (comme un système législatif) peut émettre des exigences sur la protection des données personnelles. À la différence de l'ISO/IEC, qui a par essence un rôle normatif, nous ne préjugeons pas ici de l'étendue de ces exigences, mais uniquement de leur nature, notre but étant de pouvoir décrire des réglementations.

Néanmoins, à titre d'illustration, il nous paraît commode de nous référer aux exigences légales françaises et européennes, que nous avons déjà mentionnées.

1.3.1.1 Information

Le premier axe réglementaire concerne l'information de l'utilisateur. Dans tous les textes étudiés [Ré78, art. 27] [The95, sect. IV], [The02, préambule, alinéa 23] [Ré04, art. 32], on impose au responsable d'un traitement informatique portant sur des données personnelles d'informer les propriétaires de ces données d'un certain nombre de caractéristiques de ce traitement. Typiquement, le type d'information fourni est défini par les cinq autres axes réglementaires. Une réglementation peut par exemple imposer que lorsqu'un traitement mettant en jeu des données personnelles a lieu, les propriétaires de ces données soient informés de la nature du traitement.

1.3.1.2 Consentement

Le deuxième axe réglementaire concerne l'accord, exprimé par le propriétaire des données, à la collecte et au traitement de ses données personnelles. Dans les textes étudiés, ce consentement est qualifié suivant les cas d'explicite ou indubitable. Dans la législation européenne, le consentement opt-in est de rigueur, c'est-à-dire que par défaut l'on considère que l'utilisateur n'autorise pas le traitement. Bien évidemment, les textes légaux prévoient (pour le consentement comme pour certains autres axes) des exceptions dans les cas mettant en jeu la santé publique, la sécurité nationale, l'instruction des affaires judiciaires...

1.3.1.3 Modification

Le troisième axe intitulé «modification », regroupe plusieurs concepts liés. C'est à cet axe qui est rattaché le droit d'accès et de modification initialement introduit par la loi 78-17 [Ré78, chap. 5] ainsi que toute réglementation visant à donner à l'utilisateur les moyens de demander une mise à jour ou une suppression des informations personnelles collectées. Les réglementations en la matière sont généralement de deux ordres : la nécessité pour l'utilisateur de disposer d'un moyen d'effectuer de telles requêtes auprès du responsable du traitement, et l'obligation (généralement conditionnelle) pour ce dernier d'y accéder.

1.3.1.4 Justification

Ce quatrième axe est de loin le plus complexe des six. Il a pour vocation de regrouper les réglementations traitant de la question de fond : « est-il justifié de collecter telle donnée et de l'utiliser dans le cadre de tel traitement ? ». Cet axe se réfère donc aux notions de finalité, de proportionnalité et de minimisation des données, principalement abordés par la directive européenne de 1995 [The95]. Les réglementations s'y rapportant auront donc notamment pour objet la restriction du type d'information collectée en fonction du traitement déclaré et l'encadrement de la réutilisation des informations collectées pour d'autres traitements. Les règles édictées en la matière sont souvent très dépendantes du domaine d'application, puisqu'elles ont pour but de signifier quel type d'information peut être utilisé pour quel type de traitement.

1.3.1.5 Conservation

Le cinquième axe traite de la conservation des données personnelles après leur collecte. Les textes posent des limites (en général supérieures, parfois inférieures) aux durées de conservation en fonction du contexte : statut des acteurs, nature des données et des traitements. Dans le cas général, la législation européenne prévoit que les données personnelles ne doivent pas être conservées « plus longtemps que nécessaire », les modalités de cette règle générique étant précisées dans des cas particuliers ou laissées aux soins d'autres autorités de réglementation (contrats, réglementations sectorielles, textes de loi plus spécifiques...).

1.3.1.6 Transmission

Le sixième axe concerne la transmission à des tiers des données déjà collectées. Les réglementations en la matière autorisent, interdisent ou limitent cette transmission (qui peut éventuellement prendre la forme d'une transaction commerciale). La règle générale dans l'Union Européenne veut que de telles transmissions soient interdites, à moins que l'utilisateur n'y ait expressément consenti. Encore une fois, des exceptions sont ménagées pour permettre notamment la transmission de données personnelles aux pouvoirs exécutif et judiciaire lorsque cela est nécessaire[5].

1.3.2 Problématiques spécifiques aux domaines d'intérêt

Pour illustrer cette classification, nous reprenons ici les trois domaines que nous avons introduits plus haut, à savoir le domaine de la santé, l'intelligence ambiante et le commerce électronique. Nous proposons ici un aperçu du type de problèmes pouvant être posés par chacun de ces domaines d'application.

1.3.2.1 Problématiques dans le domaine de la santé

L'informatique médicale repose souvent sur un consentement a priori ou par défaut du patient, en particulier lorsque celui-ci est dans l'incapacité de s'exprimer. Un accès rapide et complet aux données peut être une nécessité vitale pour le patient, mais la divulgation

5 On pourrait considérer que l'axe « transmission » est en fait une partie intégrante de l'axe « justification », en ceci que la transmission des données constitue en général une utilisation de celles-ci pour une autre finalité que celle initialement prévue, déclarée et consentie. Cependant, c'est une action très particulière, à distinguer des autres traitements, car elle met en quelque sorte le responsable du traitement dans une position de mandataire de l'utilisateur : il doit prendre la décision de divulguer ou pas l'information en question à un tiers, comme l'utilisateur a pris cette décision envers lui. La transmission de données à un tiers peut constituer une brèche significative dans la protection des données, en fonction de la confiance accordée à ce dernier. Elle peut donner lieu, une fois les données divulguées, à des actions violant des réglementations relatives à tous les autres axes. C'est pour cette raison que nous considérons qu'il faut traiter cette dimension dans un axe réglementaire séparé.

Cette position est d'ailleurs confirmée par le soin apporté par les législateurs français et européens pour distinguer les réglementations traitant de la transmission des données.

(transmission) de ces mêmes données à des personnes dont l'accès n'est pas justifié par des raisons médicales peut aller à l'encontre des intérêts du patient. C'est par exemple le cas lorsqu'un établissement bancaire ou d'assurances s'enquiert du dossier médical d'un de ses clients. D'autre part, tout membre du personnel médical n'est pas forcément autorisé à accéder à tout type d'information médicale à n'importe quel moment. Les politiques d'accès dépendent a priori de l'état du patient, des actes médicaux prévus, du statut des personnes requérant les données et de leur implication dans les actes médicaux en question. La mise à jour des données (modification) et leur conservation peuvent être vitales pour le patient, mais elles peuvent dépendre de procédures dont il n'a qu'une connaissance partielle. L'utilisateur, devenu un patient, est ici plongé dans un univers complexe dont il ne maîtrise en général pas les subtilités, notamment lorsqu'il n'est pas suffisamment informé (ou qu'il n'est pas capable de comprendre l'information). Pour cette raison et pour sa propre sécurité, les responsables du personnel médical auront donc tendance à éviter à l'utilisateur de prendre, sur la manipulation de ses données, des décisions qui pourraient s'avérer dangereuses pour lui. L'utilisateur perd ici en contrôle et s'en remet à des tiers pour gérer ses données personnelles.

1.3.2.2 Problématiques dans le domaine de l'intelligence ambiante

En informatique ambiante et plus particulièrement dans le contexte de l'intelligence ambiante, l'utilisateur est en interaction constante avec les objets de son environnement. Dans ce contexte, l'utilisateur est amené à dévoiler des informations de diverses natures : des données explicitement exprimées concernant son souhait d'utilisation du système (comme par exemple ses préférences sur l'ambiance lumineuse d'un environnement ou sur le paramétrage d'un outil) mais aussi des informations non directement formulées sur son utilisation effective du système (comme la fréquence et la nature de ses déplacements dans un bâtiment). À cause des ressources limitées des objets de l'environnement, la conservation des données et leur modification sont sans doute des préoccupations mineures en intelligence ambiante. En revanche, la fréquence des interactions fait de la transmission un axe de travail important pour la protection des données personnelles. L'information de l'utilisateur et son consentement sont le plus souvent sous-entendus, et le problème de la justification des traitements se pose à divers degrés en fonction de la nature plus ou moins sensible des informations transmises. Bien que les environnements d'informatique ambiante soient en général développés pour assister l'utilisateur, ce dernier est mis dans une situation où il ne contrôle que partiellement l'utilisation de ses données personnelles, et peut même ne pas avoir conscience de leur existence.

1.3.2.3 Problématiques dans le domaine du commerce électronique

Dans le cadre du commerce électronique, que l'utilisateur cherche à profiter d'un service numérique purement immatériel (comme l'accès à un contenu) ou à acquérir un bien de consommation, il devra fournir un certain nombre de données au fournisseur du service. En effet, ce dernier a bien souvent l'obligation légale de conserver et de tenir à jour (modification) pendant un temps déterminé les informations permettant d'identifier ses clients, pour des raisons de traçabilité et de comptabilité. L'utilisateur est donc susceptible de dévoiler des informations sur son identité, sa localisation géographique, ses données bancaires, ses préférences relatives au service ou au produit commandé, son profil d'utilisation du service commercial, et a priori toute autre information exigée par le fournisseur de service. Les utilisateurs peuvent avoir des avis différents quant à la

pertinence (justification) de ces informations en regard du service ou du produit demandé. L'utilisateur dispose en général d'une déclaration du service commercial sur les informations relatives au traitement, mais bien souvent elles ne sont pas prises en considération (par manque de temps, d'intérêt ou par défaut de lisibilité) et le consentement de l'utilisateur est conditionné uniquement par son besoin d'accéder au service. Enfin, le service pouvant requérir la collaboration de plusieurs intervenants, certaines données doivent souvent pouvoir être transmises, ne serait-ce qu'à une société de livraison ou à une banque.

1.4 Technologies de protection des données personnelles

Des propositions de plus en plus nombreuses sont présentées comme améliorant la protection de la vie privée. On trouve dans cette approche des travaux « atomiques », traitant un aspect du problème en fournissant un outil informatique ou méthodologique (à l'image du standard Platform for Privacy Preferences du W3C [Wor06]), ou encore des propositions composites reposant sur de telles briques fonctionnelles. Cette dernière catégorie recouvre des travaux plus ou moins ambitieux, allant de la spécification de profil utilisateur sécurisé (comme dans la proposition de Stéphanie Riché et Gavin Brebner [RB03]) aux infrastructures intégrées portées par les projets européens PRIME [Eur08] ou PISA [Bor00], par exemple.
Nous nous proposons d'analyser les quelques outils et principes que l'on retrouve couramment dans les propositions existantes, et qui diffèrent du simple contrôle d'accès non spécifique à la protection des données personnelles.

1.4.1 Platform for Privacy Preferences

Le standard P3P du World Wide Web Consortium [Wor06] est un outil désormais incontournable de la communication des sites web sur leur politique de protection des données personnelles.
P3P est une spécification de documents XML décrivant les politiques de traitement des données personnelles déclarées par un site web. Ces documents sont conçus pour être accessibles par un navigateur à partir de la page d'accueil du site. L'objectif de ce projet est de rationnaliser la manière dont les sites web communiquent sur leurs traitements. Les données présentes dans un document P3P couvrent les aspects suivants :

- L'identité de l'entité collectant les données ;
- La nature des données collectées ;
- La destination (ou justification) de la collecte de données ;
- L'identification des données pouvant être partagées avec des tiers ;
- L'identification de ces tiers ;
- La possibilité offerte ou non aux utilisateurs de modifier la manière dont leurs données sont traitées ;
- Les méthodes de résolution des conflits éventuels (et le ressort juridique compétent) ;
- La durée de rétention de chacune des informations collectées ;

- Un lien vers une version de la politique lisible par un humain.

Il faut bien comprendre ici, et les documents du W3C le soulignent, que P3P n'impose aucune politique minimale, il ne fait que fournir le moyen de l'exprimer. De plus, P3P ne permet pas de vérifier que la politique est effectivement appliquée par le site web en question. P3P a pour seul objectif (comme précisé dans les spécifications) de résoudre le problème de l'information de l'utilisateur, à l'exclusion des cinq autres axes de la protection des données personnelles.

On peut toutefois noter que les diverses extensions à P3P permettent également de traiter partiellement le problème du consentement de l'utilisateur. En effet, le langage APPEL [Wor02] permet de spécifier des préférences du côté de l'utilisateur. Ainsi, le navigateur est capable de détecter automatiquement (via des moteurs fournis par le W3C) si une politique P3P est conforme aux préférences APPEL, le fait étant alors considéré comme un consentement a priori de l'utilisateur. Ce système est par exemple utilisé dans le cas simple de la décision d'acceptation d'un cookie par un navigateur.

Les concepteurs de systèmes de protection des données personnelles ont tout intérêt à s'appuyer sur le standard P3P, ou en tout cas à demeurer interopérable avec lui. En effet, il permet de résoudre de manière simple le problème de l'information de l'utilisateur, en étant capable de décrire les divers aspects relatifs au traitement des données. Si les listes de choix prédéfinies pour la spécification du type de traitement, de leur justification ou du type de données restent limitées, elles sont extensibles par le biais de schémas XML. De plus, P3P est déjà largement utilisé par les sites web pour leur communication, et de nombreux outils sont capables de manipuler le formalisme d'une manière ou d'une autre. Toutes ces raisons poussent à favoriser au maximum l'interopérabilité avec P3P, préférentiellement à d'autres langages de politiques comme SPARCLE [KS02], moins génériques et moins répandus.

Il faut toutefois rester conscient des limitations de P3P. Tout d'abord, la restriction à un rôle d'information (et éventuellement de consentement). Enfin et surtout, P3P exprime la politique d'un site web indépendamment de tous les types de réglementations que nous avons pu identifier. Un utilisateur n'a alors aucun moyen de savoir si ces politiques respectent telle loi ou telle directive. Il reste donc ici un travail d'information et de raisonnement à effectuer.

1.4.2 Sticky policies

Comme nous venons de le voir, des outils comme P3P n'assurent pas réellement la protection des données. Une fois qu'une politique de traitement est déterminée, il faut donc que les processus de traitement, de transmission et de stockage des données se chargent de l'appliquer. Une approche courante (et commune à de nombreuses propositions) consiste à attacher aux informations sensibles les méta-données de description de la politique de sécurité associée, les applications s'engageant à respecter cette « politique collante ». Ces sticky policies ont été introduites par Günter Karjoth et Matthias Schunter en 2002 [KS02].

La proposition, dans son principe, attache des règles aux données personnelles, qui ne peuvent être manipulées par l'application que si ces règles sont respectées. On retrouve par exemple cette idée dans l'architecture intégrée du projet PRIME [Euro08, ACC06] ou dans l'architecture proposée parallèlement par Marco Casassa Mont et al. [CPB03]. Si le concept est intuitif, pratique et adapté à la distribution des applications et des données (permettant un premier pas vers une réelle protection étendue), il ne donne cependant (dans sa version de base) aucune garantie à l'utilisateur quant au respect de la politique par une application distante. Il nous faudra donc détailler comment les sticky policies peuvent être

utilisées de manière à réellement contraindre l'utilisation des données à distance. Les différents types d'utilisation de ce concept devront être analysés en fonction des garanties qu'ils fournissent à un observateur distant et au propriétaire des données en particulier.

Il convient également d'observer, en ce qui concerne cette famille de propositions, que la source desdites politiques n'est pas toujours spécifiée. Bien souvent, elle est issue de négociations entre les parties ou directement déduite de politiques déclaratives de type P3P. Ce mode de fonctionnement ne permet donc pas nativement de prendre en compte les contraintes réglementaires ou légales applicables aux traitements, que ce soit à la création de la politique ou bien au moment du traitement de l'information. Nous sommes donc toujours ici en besoin d'un outil adapté pour manipuler et prendre en compte les contraintes issues des réglementations.

1.4.3 Gestion déportée des données sensibles

Des propositions ont également été faites pour permettre aux utilisateurs de profiter de services en ligne tout en évitant à ces derniers de pouvoir tracer leurs activités. C'est un type d'application qui relève donc davantage de l'accès anonyme aux services et des autorisations préservant la vie privée, autres dimensions de la protection de la vie privée décrites dans les sections 1.2.3 et 1.2.4. Néanmoins, certaines de ces propositions se rapportent plus particulièrement à la gestion des données personnelles de l'utilisateur dans ces scénarios. C'est le cas notamment du protocole SAML2.0 [SAM] établi par Liberty Alliance ou du protocole IDsec [Int], établi par l'IETF. Il consiste en la déportation de la gestion des données utilisateur sur un serveur spécialisé, mettant en œuvre des mécanismes de contrôle d'accès sophistiqués, visant à s'assurer du bien-fondé des différentes demandes d'accès au profil qui lui sont faites.
En préalable au déroulement d'une transaction entre l'utilisateur et le fournisseur de service, l'utilisateur s'identifie auprès du serveur gestionnaire de son profil, qui en retour lui procure un certificat de session (qui servira de jeton d'accès temporaire). Ce certificat est ensuite transmis au service, qui le présente au gestionnaire de profil, accompagné d'une requête concernant le profil de l'utilisateur et d'un certificat de créance sur ses propres propriétés techniques. Le gestionnaire de profil valide le certificat de session et examine le certificat de créance. Si le gestionnaire est satisfait par ce certificat, c'est-à-dire si la correspondance entre la requête et les propriétés, quelles qu'elles soient, du fournisseur de service correspondent aux exigences connues de l'utilisateur, alors les informations de profil sont transmises au service.
Les approches de ce type ont apporté des idées intéressantes, notamment dans le cadre de la gestion des identités virtuelles telle proposée par le projet FC2 (Fédération des Cercles de Confiances) par exemple ; mais souffrent de limitations discriminantes. Tout d'abord, la localisation des données personnelles de l'utilisateur sur un serveur délocalisé et clairement identifié pose un problème de sécurité mis en avant par les concepteurs mêmes d'IDsec : le serveur gestionnaire, dépositaire de nombreuses données potentiellement sensibles, devient en effet une cible privilégiée pour des attaques informatiques. Cet aspect du problème milite fortement en faveur d'une gestion des données personnelles directement par leurs propriétaires, de manière distribuée. De plus, ce protocole ne s'inquiète que de l'accès initial aux données et ne fournit aucun moyen pour assurer leur protection étendue. Enfin, les possibilités offertes à l'utilisateur de spécifier des propriétés techniques à vérifier pour qu'un fournisseur de service puisse accéder à telle ou telle partie de son profil personnel sont très limitées en termes d'expressivité. En effet, pour traiter réellement de protection

des données personnelles, il faudrait pouvoir définir des politiques capables de se référer aux six axes définis dans la section 1.3.1.

1.4.4 Agents utilisateurs

Certaines propositions émanant du domaine des systèmes multi-agents impliquent des agents artificiels [Fer95] dans la protection des données personnelles des utilisateurs. Dans ce contexte, les agents désignent des entités logicielles capables d'interagir de manière autonome avec d'autres entités ainsi qu'avec l'environnement dans lequel elles sont situées. Ces agents peuvent être des briques logicielles destinées à mettre en œuvre le traitement en lui-même, comme dans l'architecture proposée par John J. Borking [Bor00]. Ils peuvent également se présenter sous la forme d'agents mobiles et se déplacer avec les données. Dans des travaux comme ceux de Stéphanie Riché, Gavin Brebner et Mickey Glitter [RBG02, RB03], ces agents sont des assistants logiciels au service d'un utilisateur placé au centre de l'application. Ces agents personnels ou agents utilisateurs sont alors chargés de surveiller et de contrôler l'utilisation qui est faite des données, de manière que cette utilisation reste conforme avec la politique établie.

Cette approche permet de répondre au problème majeur posé par la gestion déportée des données personnelles par une reprise de contrôle de l'utilisateur sur ses informations, tout en décentralisant les fonctionnalités de raisonnement dans des entités autonomes. L'agent ou les agents utilisateurs permettent d'interfacer les relations entre l'utilisateur humain et les différents services et applications, en lui confiant la responsabilité de certaines décisions (comme par exemple dans le cas du consentement a priori). Dans cette perspective, c'est l'agent utilisateur qui met en œuvre les différents mécanismes (principalement de contrôle d'accès) assurant la sécurité des données personnelles de l'utilisateur.
Le paradigme des agents utilisateurs est donc séduisant à plusieurs titres. Si le principe de l'agent n'est pas en soi un outil de protection des données personnelles, il peut être chargé de leur mise en œuvre.

1.5 Discussion

Cet aperçu des problématiques inhérentes à la protection de la vie privée et des données personnelles en particulier amène diverses réflexions sur les solutions existantes et les travaux restant à conduire en la matière. Nous présentons par la suite trois grandes problématiques faisant défaut aux solutions actuelles. On ne s'intéresse dans nos travaux de recherche qu'au deux premières.

1.5.1 Représentation et raisonnement

L'étude de la sémantique du problème de la protection des données personnelles, de son contexte réglementaire et des technologies actuellement utilisées pour l'assurer mettent en avant un besoin (et un manque) fondamental : celui de permettre, au niveau des solutions

techniques, une représentation des réglementations en vigueur, un raisonnement sur ces réglementations et une adaptation dynamique à celles-ci. Nous avons vu en effet que les solutions proposées se référaient à des politiques de sécurité qui n'étaient pas directement reliées aux contraintes légales ou réglementaires pesant sur le contexte d'exécution du traitement, alors que ces contraintes ont une sémantique riche, précise et surtout variable. Suivant la localisation géographique des traitements, l'identité des tiers et l'autorité sous laquelle les traitements sont effectués, divers textes légaux, réglementaires ou contractuels peuvent entrer en jeu ou pas.
Ainsi, il est nécessaire que les outils mis en œuvre, quels qu'ils soient, disposent d'un moyen de représentation des données qui permettent de les mettre en relation avec lesdites réglementations. Il doit être également possible de raisonner sur la nature de ces données, leur destination et les contraintes qui pèsent sur elles.

1.5.2 Manque de contrôle

Le concept de « *la minimisation de la collecte* » des données personnelles, défini dans la directive européenne 95/46/CE (section 1.1.2.2), interprété du côté des usagers comme un concept de « *transmission minimale* » des données présente un axe de travail fort pour la protection des données personnelles. L'ensemble des entités impliquées dans le cycle de vie des données personnelles (SP, utilisateur, autorités légales, …etc) y adhèrent, en revanche, il fait toujours défaut aux solutions techniques proposées.
De ce fait, il est nécessaire de proposer des solutions impliquant d'avantage l'utilisateur dans le processus de protection de ses données. Ceci selon deux dimensions, la première en lui donnant plus de contrôle sur les conditions d'usage associées aux données personnelles. La deuxième, en mettant à sa disposition des moyens techniques assurant que les données transmises au SP sont vraiment minimales.**1.5.3 Le problème de la confiance**
Un autre sujet qui nous semble important à la lecture des propositions existantes est celui de la confiance. L'objectif commun des propositions citées est de permettre à l'utilisateur de protéger au mieux ses données personnelles. Cependant, dès lors que ces données sont confiées à des agents tiers, la conviction que l'utilisateur peut avoir que la dite protection soit assurée repose sur la confiance qu'il a dans les agents en question (et éventuellement sur leurs interlocuteurs).
Ces agents étant précisément les entités à qui profiterait, le plus souvent, une violation de la politique de protection des données, la confiance par défaut de l'utilisateur dans un environnement ouvert doit rester très faible.

1.6 Conclusion

Les problématiques que nous venons de présenter concernent essentiellement les utilisateurs humains du système, et les solutions proposées sont orientées par cette considération. De ce fait, les propositions que nous avons décrites comprennent peu d'automatisations de nature à mettre en relation les exigences de protection des données avec les techniques utilisées.

Dans ce chapitre nous avons classé les exigences réglementaires en matière de protection des données en six axes. Ces axes portent sur l'information, le consentement, la modification des données, la justification de la collecte, la conservation et la transmission des données.

Dans le chapitre suivant, nous proposons un langage de protection des données, qui fournit une représentation de ces axes et leur mise en place sous un angle de contrôle d'accès. Le reste de nos contributions porte principalement sur l'axe « *justification* ».

DEUXIEME PARTIE

CHAPITRE 2

LANGAGE XPACML ET MODELE D'ARCHITECTURE POUR LA PROTECTION DES DONNEES PERSONNELLES

Sommaire

Introduction 56
2.1 Problématiques et besoins 56
2.2 Langages de politiques de protection des données 57
2.2.1 Platform for Privacy Preferences (P3P) 58
2.2.2 P3P Privacy Policy Exchange Language APPEL et XPREF 63
2.2.3 Implémentations existantes côté utilisateur 66
2.3 Limitations 68
2.4 Le langage de politiques de contrôle d'accès XACML 68
2.4.1 Modèles de contrôle d'accès sensibles à la protection de la confidentialité 68
2.4.2 Principes de XACML 71
2.4.3 Architecture de XACML 77
2.5 Complémentarité des langages de politiques de protection des données et des langages de contrôle d'accès 79
2.6 XPACML 81
2.6.1 Contraintes à considérer dans les politiques de protection des données XPACML 81
2.6.2 Principes du langage XPACML 82
2.6.3 Architecture de contrôle d'accès XPACML 91
2.7 Génération des politiques XPACML 99
2.7.1 Définition du schéma de politique 99
2.7.2 Exemples de politiques XPACML 104
2.7.3 Vérification de conformité des fichiers de politiques 106
2.7.4 Comparaison des politiques (implémentation du PPDP) 107
2.8 Prime Life Policy Language (PPL) 108
2.8.1 Principes du langage PPL 109
2.8.2 Architecture de contrôle d'accès PPL 112
2.8.3 Diagramme de flux du moteur de contrôle d'accès PrimeLife 113
2.8.4 Comparaison entre PPL et XPACML 114
2.9 Conclusion 116

2 Langage XPACML et modèle d'architecture pour la protection des données personnelles

Introduction

Dans ce chapitre, nous nous attachons tout d'abord à étudier les outils permettant de répondre à la problématique de présentation de la réglementation mentionnée dans l'introduction de ce mémoire, selon les axes exposés dans la section 1.3.1 du chapitre 1, puis à l'application des éléments de présentations recueillis dans un langage de politique gérant l'accès aux données personnelles de l'utilisateur.

Pour ce faire, nous détaillons les besoins associés à la problématique de *présentation* et de *contrôle* d'usage dans la section 2.1. Puis, nous étudions un état de l'art des langages de protection des données (section 2.2) afin d'identifier les éléments utilisés dans la littérature pour la protection des données personnelles. Cet état de l'art montre des limites en termes d'applicabilité et de contrôle que nous discutons dans la section 2.2.4. Nous étudions alors les langages de contrôle d'accès dans la section 2.4, et plus particulièrement XACML.

La complémentarité des langages de protection des données (P3P principalement) avec ceux de contrôle d'accès (XACML principalement) est étudiée dans la section 2.5, pour enfin, détailler notre proposition de langage XPACML pour la protection des données personnelles dans la section 2.6 de ce mémoire. Les aspects implémentation de ce dernier (XPACML) sont exposés dans la section 2.7. Un nouveau langage pour la protection des données basé sur le standard XACML est présenté dans la section 2.8 suivie d'une discussion comparative avec le langage XPACML. La conclusion avec un bilan de nos contributions est donnée dans la section 2.9.

2.1 Problématiques et besoins

La communication des données personnelles au SP, est une nécessité dans tout scénario de commerce électronique (section 1.1.3.3), ce qui pose un certain nombre de problèmes en termes de protection des données (section 1.3.3.3). L'utilisateur est sensibilisé aux différents traitements inhérents à ses données avec des politiques de traitement des données édictées et communiquées par le SP. Ces politiques sont définies à l'aide de langages de politiques de protection des données (appelées souvent langages de protection de la vie privée-section 2.2).

En référence aux problématiques présentées dans l'introduction de ce mémoire, nous identifions les besoins fondamentaux auxquels ces langages tentent de répondre :

- Représentation et transparence
  Les politiques de traitements de données fournies à l'utilisateur par les SPs étaient jusqu'à un passé proche difficiles à lire et à comprendre, de par leurs tailles potentiellement conséquentes, ainsi que les termes juridiques complexes qu'elles contiennent. Ceci a conduit à la perte de confiance de l'utilisateur et à nuire à son adhésion aux services offerts par le SP.
  Dans le but de donner plus de transparence sur leurs pratiques d'usage des données personnelles, et prouver leur conformité aux règlementations en vigueur (résumées dans nos études aux six axes de protection-section 1.3.1 du chapitre 1), nombre de SPs ont eu recours aux langages de politiques de protection des données - section 2.2).

- Contrôle local des données personnelles
  Le contrôle local des données personnelles est le sous-ensemble des mesures de protection des données personnelles ne nécessitant que la vérification de propriétés sur le terminal de l'utilisateur propriétaire des données. La protection locale telle que nous l'entendons contient donc l'ensemble des vérifications qui peuvent être faites directement par l'utilisateur. Cela concerne donc en particulier les deux premiers axes de protection (section 1.3.1). En effet, l'utilisateur peut aisément déterminer par lui-même, localement, de quoi il a été informé et des consentements qu'il a exprimés ou non.
  Les troisième et quatrième axes (section 1.3.1) sont également concernés. En effet, concernant l'axe modification, l'utilisateur sait s'il dispose ou non d'un moyen de contacter le responsable d'un traitement en vue d'exercer un éventuel droit d'accès, de modification ou de suppression. Concernant l'axe justification, l'utilisateur a la possibilité de vérifier localement les justifications fournies par le SP à l'égard des règlementations associées.

  La mise en œuvre pratique du contrôle étant actuellement restreinte à des formalisations et implémentations des politiques de protection, et une acceptation des termes qui y sont contenus par le biais d'un consentement requis. L'utilisateur reste souvent contrarié par le manque de moyens techniques lui permettant de définir ses préférences globales et personnalisées, en termes d'usages associés aux données à révéler.

  Des extensions ont donc été apportées aux langages de politiques de protection des données pour prendre en compte le besoin de l'utilisateur dans la définition de ses préférences de façon globale (section 2.2.2).

2.2 Langages de politiques de protection des données

Comme mentionné dans la section précédente, les langages de politiques de protection des données ont été conçus dans un premier temps pour permettre aux SPs l'expression de leurs pratiques d'usage des données personnelles collectées, dans un formalisme

techniquement interprétable. Elles ont évolué par la suite pour prendre en compte les préférences de l'utilisateur.

Nous présentons ainsi les outils et langages P3P, APPEL et XPREF qui ont été proposés dans la littérature afin de faciliter l'expression de ces politiques.

2.2.1 Platform for Privacy Preferences (P3P)

Le standard P3P du World Wide Web Consortium [Wor06] est désormais l'outil le plus utilisé par les SPs. Il a pour objectif principal, la rationalisation de la manière dont les SPs communiquent sur leurs politiques de traitement des données.

P3P est une spécification de documents XML décrivant les politiques de traitement des données personnelles déclarées par un SP. Ces documents sont conçus pour être accessibles via des URIs depuis un navigateur.

Une fois récupérés, ces documents sont traités par un agent utilisateur intégré à son navigateur. Cet agent a pour rôle d'informer l'utilisateur des pratiques du SP et d'automatiser la prise de décision en fonction de ces pratiques. Les utilisateurs ont donc pas besoin de lire les politiques de traitement de données de chaque site visité.

Les spécifications P3P dans leurs deux versions (P3P 1.0 en 2002 et P3P 1.1 en 2006) définissent [Wor06] :

- Un schéma standard des données pouvant être collectées par un site donné, nommé « *P3P base data schema* » ;
- Un format XML pour l'expression de la politique de traitement des données ;
- Un standard décrivant les utilisations possibles, les destinataires, les catégories de données, …etc
- Des mécanismes pour l'association des politiques de traitement des données à des pages Web ou à la page du site du SP,
- Un mécanisme pour le transport des politiques P3P via HTTP,

La figure 2.1 illustre le mécanisme opérationnel de P3P et les différentes interactions entre l'utilisateur et le SP. Les spécifications P3P définissent quatre manières différentes [Wor06] permettant à un agent utilisateur de récupérer le fichier de références contenant la localisation des politiques de traitement de données associées à chaque partie (ou service) du SP. Ensuite, l'agent utilisateur récupère la politique, qui est ensuite comparée aux préférences globales de l'utilisateur, ainsi l'action correspondante peut être prise. Finalement la page web requise est accessible.

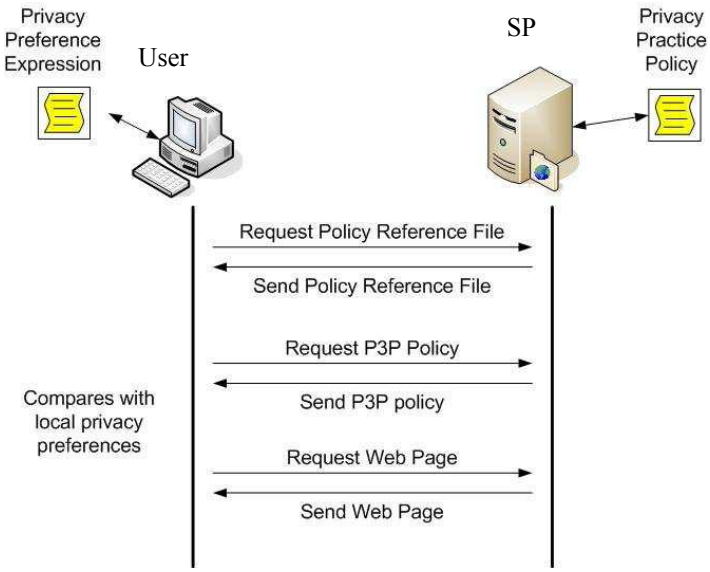


Figure 2.1 Mécanisme opérationnel de P3P

Les politiques P3P utilisent un codage XML avec des espaces de nommage appartenant à un vocabulaire spécifique à P3P afin de fournir les coordonnées de l'entité légale responsable des pratiques de protection des données d'une politique, d'énumérer les types de données collectées et d'expliciter le but de traitement (Purpose), le(s) destinataire(s) (Recipient) et le temps de rétention (Retention) des données en question.

Parmi les différents espaces de nommage, on trouve les éléments relatifs aux politiques qui sont surtout utilisés pour fournir des informations concernant la politique elle-même (Par exemple, quelle entité fournit cette politique ? sur quelles données elle porte ? comment gérer les conflits de politiques ? ...).

- ENTITY : cet élément donne une description précise de l'entité légale collectant les données (SP),

- DATA : cet élément exprime la nature des données collectées. Ces données sont regroupées dans des catégories nommées "DATA-GROUP", dont chacune regroupe les données ayant la même politique.

On trouve également les éléments relatifs aux déclarations. Les déclarations (statement) décrivent le traitement appliqué aux types de données spécifiques (Purpose/Recipient/Retention). Ce sont ces éléments qui sont particulièrement intéressants à utiliser en termes de protection de données personnelles et sur lesquels nous nous appuyons pour définir les éléments nécessaires à l'expression des axes de protection définis dans la section 1.3.1 du chapitre 01. Le détail de ces éléments est donné ci-après.

2.2.1.1 L'élément STATEMENT

Parmi les différents éléments des déclarations, on pourrait qualifier l'élément <statement> de « chef ». C'est en fait le conteneur regroupant un élément <purpose>, un élément <recipient>, un élément <retention>, un élément <data-group> et, en option, un élément <consequence>.
Il est également spécifié que toutes les données référencées par l'élément <data-group> sont manipulées conformément aux divulgations présentes dans les autres éléments contenus par la déclaration <statement>.

L'élément <consequence> est un élément optionnel permettant à un utilisateur humain d'obtenir plus d'informations concernant les pratiques d'un SP. Il pourra être utilisé par un utilisateur pour obtenir davantage d'informations et peut-être prendre des décisions plus précises quant aux préférences vis-à-vis de ses données.

Dans la spécification P3P, il est également fait mention de l'élément <non-identifiable>.
Cet élément est également optionnel. Sa présence signifie juste qu'aucune donnée n'est collectée dans le cadre de l'élément <statement> qui le contient ou que les données appelées par cet élément <statement> seront rendues anonymes lors de leur collecte. Rendre anonyme signifie qu'il sera impossible à quiconque de rattacher ces données à l'identité d'une personne en utilisant des moyens raisonnables. C'est le cas par exemple pour les identifiants de session générés aléatoirement.
Il est important de noter que si l'élément <non-identifiable> est présent dans un élément <statement>, alors les autres éléments de ce <statement> deviennent optionnels. Dès lors qu'un élément <statement> ne contient pas de sous-éléments <non-identifiable>, il doit contenir les sous-éléments <purpose>, <recipient> et <retention>.

```
POLICIES xmlns=" ht tp : //www.w3 . org /2002/01/P3Pv1">
<POLICY . . . name=" policy ">
  <ENTITY/>
    <STATEMENT>
       <CONSEQUENCE> À votre demande, nous vous enverrons des offres
commerciales soigneusement sélectionnées qui sont susceptibles de vous
intéresser.
       </CONSEQUENCE>
       <PURPOSE>
              <contact required="opt-in"/>
              <individual-decision required="opt-in"/>
              <tailoring required="opt-in"/>
       </PURPOSE>
       <RECIPIENT><ours/><same required="opt-in"/>
       </RECIPIENT>
       <RETENTION><stated-purpose/></RETENTION>
       <DATA-GROUP>
              <DATA ref="#user.name" optional="yes"/>
              <DATA ref="#user.home-info.postal" optional="yes"/>
              <DATA ref="#user.home-info.telecom.telephone"
       optional="yes"/>
              <DATA ref="#user.business-info.postal" optional="yes"/>
              <DATA ref="#user.business-info.telecom.telephone"
       optional="yes"/>
              <DATA ref="#user.home-info.online.email" optional="yes"/>
```

```
        </DATA-GROUP>
    </STATEMENT>
</POLICY>
</POLICIES>
```

Figure 2.2 Exemple basique d'une politique P3P

On remarque dans l'exemple de la figure 2.2 que le champ conséquence est bien rempli et offre une courte explication sur l'usage des données présentes dans l'élément <Data-group>. On voit donc que le serveur propose en « opt-in » [Ent] d'envoyer des offres commerciales à l'utilisateur et qu'il a besoin pour cela de son nom, de son code postal (au bureau ou à la maison), de son numéro de téléphone (professionnel ou non) et de son email. Toutes ces données sont optionnelles et peuvent ne pas être remplies.

2.2.1.2 L'élément PURPOSE

Cet élément exprime une ou plusieurs intentions concernant la collecte ou l'utilisation des données. Voici la liste détaillée des différentes valeurs que peut prendre cet élément :

- <current /> : les données collectées seront utilisées uniquement dans le but d'achever le processus pour lequel elles sont fournies, donc pour la transaction en cours. Cela peut-être par exemple un mot de passe pour confirmer un virement bancaire.
- <admin /> : les données collectées pourront servir au support technique du site Web du fournisseur de services et de son système informatique. Cela peut être – par exemple - le mot de passe utilisé par un administrateur pour confirmer la modification d'un site web.
- <develop /> : les données serviront à l'amélioration, l'évaluation ou la mise à jour d'un site, d'un service, ou d'un produit. L'exemple le plus typique est le pseudo laissé lors de l'évaluation d'un produit en ligne.
- <tailoring /> : les données seront utilisées ponctuellement pour ajuster le contenu et l'aspect du site en fonction des actions de l'utilisateur. C'est ce que fait Amazon lorsqu'il propose des articles équivalents à ceux placés dans le panier par l'utilisateur au cours d'une même session.
- <pseudo-analysis /> : les données serviront à la création d'un enregistrement pour un individu (ou un terminal) particulier sans associer de données identifiées (nom, adresse...).
  Ce profil servira à déterminer les habitudes, centres d'intérêts d'un utilisateur dans un but de recherche et d'analyse, mais pas pour identifier un individu en particulier. Cela concerne les SPs souhaitant étudier l'intérêt des visiteurs pour les différentes pages de leur site.
- <pseudo-decision /> : même principe que <pseudo-analysis /> mais les données servent cette fois à prendre une décision affectant directement un individu. Par exemple, un serveur peut vouloir ajuster ou modifier le contenu affiché par le navigateur en fonction des pages visitées antérieurement (pendant une précédente session par exemple).
- <individual-analysis /> : cette valeur est identique à <pseudo-analysis> sauf que cette fois les données collectées peuvent être des données identifiées (donc nom, adresse, téléphone...)

- <individual-decision /> : similaire à <pseudo-decision /> avec des données identifiées.
- <contact /> : les données serviront à contacter l'utilisateur par un canal de communication autre que le téléphone, afin de promouvoir un produit ou un service. Par exemple, cela peut-être la publicité envoyée par priceminister pour vous signaler qu'un de vos vendeurs favoris a ajouté des objets dans sa boutique.
- <historical /> : les données seront conservées pour être archivées / stockées pour répondre à une exigence spécifiée par une loi (ou politique) existante. Par exemple, si le serveur a un service après-vente, il aura besoin de conserver des informations concernant votre achat pour l'éventuel usage de ce service.
- <telemarketing /> : idem que <contact />, mais cette fois le canal de communication est obligatoirement un téléphone.
- <other-purpose>valeur</other-purpose> : ici le serveur y inclut d'autres intentions non présentées dans la spécification.

Chaque type d'intention (excepté <current />) peut également avoir des attributs optionnels :

- Always : signifie que l'intention doit toujours être acceptée par l'utilisateur pour que la transaction puisse avoir lieu. Si au cours d'une négociation des termes de politiques un utilisateur refuse cet usage de sa donnée, alors la transaction avortera.
- Opt-in : signifie qu'une donnée peut être utilisée conformément à l'intention déclarée si l'utilisateur confirme cet usage.
- Opt-out : la donnée collectée sera utilisée selon l'intention déclarée sauf si l'utilisateur ne le souhaite pas.

2.2.1.3 L'élément RECIPIENT

L'élément <recipient> indique le (ou les) destinataire(s) des données collectées. Les destinataires doivent être classés parmi l'un des types de destinataire suivants :

- <ours> : désigne le SP et les entités travaillant pour le SP ou celles pour qui le SP travaille. Ce sont donc les agents traitant les données pour le compte du SP afin de réaliser les intentions déclarées. Par exemple, un service publicitaire qui imprime les étiquettes d'adresses sans faire un autre usage de ces informations.
- <delivery> : les destinataires de ce type sont les services de livraison se conformant éventuellement à d'autres pratiques que celles du SP, voire des pratiques inconnues. Typiquement, un site faisant appel à la Poste et n'en connaissant pas les pratiques doit faire figurer « la Poste » comme un destinataire de type <delivery>.
- <same> : les personnes morales se conformant aux pratiques du SP. Cela concerne par exemple les partenaires du SP qui suivent une même politique que celle du SP.
- <other-recipient> : les personnes morales sous contrat et responsables auprès du SP mais qui peuvent suivre d'autres pratiques que celles du SP. Si Facebook avait dès le départ déclaré que les données seraient revendues à des entités commerciales, ils auraient certes eu moins de clients, mais il n'y aurait jamais eu de polémiques.
- <unrelated> : les entités morales dont les pratiques sont totalement inconnues et qui ne sont pas sous contrat avec le SP d'origine.
- <public> : les données auront pour destinataires des tribunes publiques (annuaires publiques, ...). Par exemple, les numéros de téléphone collectés par le site de France Télécom peuvent être référencées dans l'annuaire public.

Chacune de ces balises peut présenter en option :

- Une ou plusieurs balises <recipient-description> présentant une description du destinataire,
- Un attribut *required* (sauf pour <ours>).

2.2.1.4 L'élément RETENTION

Cet élément indique la durée maximale pendant laquelle la donnée sera conservée par les serveurs du SP. Ces durées sont à choisir parmi les suivantes :

- <no-retention /> : la donnée est conservée pour la durée de leur utilisation, lors d'une seule opération en ligne. Les données concernées doivent être totalement détruites immédiatement après la fin de cette opération et ne doivent pas être sauvegardées dans un quelconque journal.
- <stated-purpose /> : les données concernées sont conservées uniquement le temps nécessaire à la satisfaction de l'intention déclarée. S'il s'agit par exemple d'une adresse collectée pour l'envoi d'un colis, cette adresse devra être supprimée de toutes les bases de données du serveur dès que le colis sera envoyé, auquel cas le client devra renseigner de nouveau son adresse s'il fait un autre achat ultérieurement.
- <legal-requirement /> : les données sont conservées pour satisfaire l'intention déclarée mais la période de rétention peut être supérieure à celle proposée par <stated-purpose> du fait d'une obligation ou d'une responsabilité légale. Il peut s'agir, par exemple, de conserver des enregistrements pour un audit.
- <business-practices /> : les données sont conservées sous couvert des pratiques commerciales déclarées par le SP.
- <indefinitely /> : comme son nom l'indique, les données sont conservées pour une durée non déterminée. Cette option reflète l'absence de politique de rétention.

2.2.2 P3P Privacy Policy Exchange Language APPEL et XPREF

L'exploitation réelle de P3P prend du sens quand l'utilisateur est muni d'outils pour traiter de telles politiques. Ainsi, le W3C a conçu le langage APPEL [Wor02] permettant de spécifier les préférences du côté utilisateur. Ainsi, l'agent utilisateur est capable de détecter automatiquement (via des moteurs fournis par le W3C) si une politique P3P du SP est conforme aux préférences APPEL. Le fait est alors considéré comme un consentement de l'utilisateur.

Les préférences en termes de protection des données exprimées par APPEL, sont principalement un ensemble de règles (chacune est exprimée par l'élément `RULE`) contenues dans un élément `RULESET`. Chaque élément RULE contient un attribut « body », un attribut « behaviour » et optionnellement un attribut « description » qui fournit une explication simple de la règle.

- L'attribut « *body* » fournit un modèle de politique comparable à une politique P3P. En effet, Le langage APPEL réutilise quelques éléments P3P,

principalement les éléments : « POLICY, STATEMENT» et tous leurs nœuds feuilles nécessaires. Ainsi, l'utilisateur a la possibilité de spécifier comment ses données doivent être traitées.

- L'attribut « *behaviour* » est utilisé pour exprimer la volonté de l'utilisateur de révéler un groupe de données personnelles dans les conditions exprimées dans l'élément « STATEMENT ». Il peut prendre une valeur parmi les suivantes :
  - **Block** : cette valeur indique que l'échange de la donnée spécifié dans une règle est rejeté,
  - **Request** : cette valeur permet le traitement de la requête en cours,
  - **Limited** : l'accès aux données requises par le SP est limité aux données obligatoires.

Par conséquent, l'agent utilisateur compare automatiquement les préférences de l'utilisateur avec la politique P3P du SP. Pour cet objectif, les spécifications APPEL définissent des algorithmes de recherche et de comparaison. Ces derniers utilisent des connectives définies dans les spécifications APPEL pour comparer les politiques. Ces connectives peuvent prendre cinq valeurs : OR, AND (connective par défaut), NON-OR, NON-AND, AND-EXACT, OR-EXACT.
La connective « OR » signifie que la politique du SP est conforme si une ou plusieurs de ses expressions sont trouvées dans la connective. Si aucune expression de cette dernière n'est trouvée dans la politique du SP, le test de conformité échoue.
La connective « OR-EXACT » en revanche, signifie que la politique est conforme si une ou plusieurs de ses expressions (de la connective) sont trouvées dans la politique du SP. Si cette politique contient des éléments qui ne sont pas listés dans la règle, le test de conformité échoue. Seule la connective « AND-EXACT » signifie que la politique est conforme si toutes ses expressions contenues peuvent être trouvées dans la politique du SP. Comme la connective précédente, si la politique contient des éléments non-listés dans la règle, la comparaison échoue.

La figure 2.3 illustre la formulation des préférences utilisateur dans une politique APPEL :

```
<appel:RULESET xmlns :appe l=" h t t p: //www.w3 . org /2001/02/APPELv1"
    xmlns:p3p=" h t tp : //www.w3 . org /2000/12/P3Pv1">
  <appel:RULE behavior=" block " description=" Service collects personal
data for 3rd parties ">
    <p3p:POLICY>
     <p3p:STATEMENT>
      <p3p:DATA-GROUP>
         <p3p:DATA>
              <p3p:CATEGORIES appel : connective=" or ">
                <p3p:physical />
                <p3p:purchase />
               </p3p:CATEGORIES>
         </p3p:DATA>
      </p3p:DATA-GROUP>
      <p3p:PURPOSE appel : connective=" or ">
 <p3p: telemarketing />
 <p3p:other-purposes />
      </p3p:PURPOSE>
      <p3p:RECIPIENT>
 <p3p:same />
```

```
 <p3p:unrelated />
      </p3p:RECIPIENT>
      <p3p:RETENTION>
 <p3p:business-practices />
 <p3p :undefinitely />
      </p3p:RETENTION>
    </p3p:STATEMENT>
  </p3p:POLICY>
 </appel:RULE>
</appel:RULESET>
```

Figure 2.3 Exemple d'une politique APPEL des préférences utilisateur

Dans cette politique, l'utilisateur n'autorise pas le SP à collecter ses données physiques ou d'achat pour des buts de télémarketing ou des buts indéfinis. Il refuse que ses données soient retenues pour une durée indéterminée. Il n'autorise pas non plus la rétention de ses informations sous les pratiques de business fournies par le SP. Le partage de ces informations est aussi non-autorisé avec des SPs qui n'ont pas les mêmes pratiques de traitement de données que le SP en question.

Bien que défini dans un objectif limité, APPEL est de prime abord un schéma attractif et simple pour l'expression des préférences de l'utilisateur. Cependant, les travaux [Xpr], [You09], et [Hog02] montrent qu'il contient de sérieuses limitations.
En plus de son incapacité à exprimer des règles sophistiquées [Xpr], les limitations de conception d'APPEL sont les suivantes:

- L'utilisateur ne peut exprimer formellement que ce qui est inacceptable. Par conséquent, il est difficile de construire des expressions avec une logique opposante.
- Les règles sont évaluées dans l'ordre, car APPEL ne permet pas le traitement d'une combinaison de règles dans un élément RULESET.

Pour remédier à ces limitations, Agrawal et al [Xpr], suggèrent une amélioration partielle de ce langage. Ainsi ils ont proposé le langage XPREF. Ce langage réutilise deux éléments du langage d'APPEL : l'élément `RULE` et l'élément `RULESET`.

XPref exploite également le langage XPath (langage utilisé pour comparer la structure d'un document XML avec une notation de chemin utilisée pour naviguer dans une structure hiérarchique d'un document XML). XPath peut non seulement identifier les combinaisons d'éléments P3P, mais vérifie également que seules les combinaisons P3P spécifiées sont présentes. Ainsi, XPref peut spécifier les combinaisons acceptables et inacceptables des éléments P3P. A titre d'exemple, si on souhaite exprimer une préférence acceptable spécifiant que seuls les buts « *local-analysis* » et « *statistics* » sont acceptables, l'expression de la règle sera comme suit :

```
<XPref: RULESET>
  <XPref: RULE behavior = "request">
every $name in
STATEMENT/PURPOSE/* satisfies
(name ($name) = "local-analysis" or
name ($name) = "statistics")
   </XPref:RULE>
```

```
   <XPref:RULE behavior = "block" condition="true"/>
   </XPref:RULE>
</XPref:RULESET>
```

L'expression contenant l'élément « every » est une expression XPath qui cherche le but spécifié dans toute la structure de la politique P3P du SP, contrairement à APPEL qui compare seulement avec le premier « *statement* » sans parcourir la politique entière.

2.2.3 Implémentations existantes côté utilisateur

Plusieurs implémentations ont été proposées pour la création de politiques P3P. Toutefois, peu d'outils effectuent un véritable traitement de ces politiques.

2.2.3.1 Les agents natifs aux navigateurs

Les navigateurs qui intègrent nativement P3P sont Internet Explorer 6, et Netscape 7. Selon les options de ces derniers, le paramétrage peut permettre de bloquer un SP dans ses tentatives de lecture d'un cookie se trouvant sur la machine de l'utilisateur, et ce afin de récupérer une information quelconque sans son consentement, que ce soit implicite ou explicite.

Par contre, l'utilisateur peut aussi paramétrer son navigateur web de telle façon qu'un site auquel il accorde sa confiance puisse obtenir des informations confidentielles le concernant. La mise en place d'un tel système nécessite donc que l'utilisateur possède un navigateur interprétant le langage P3P.

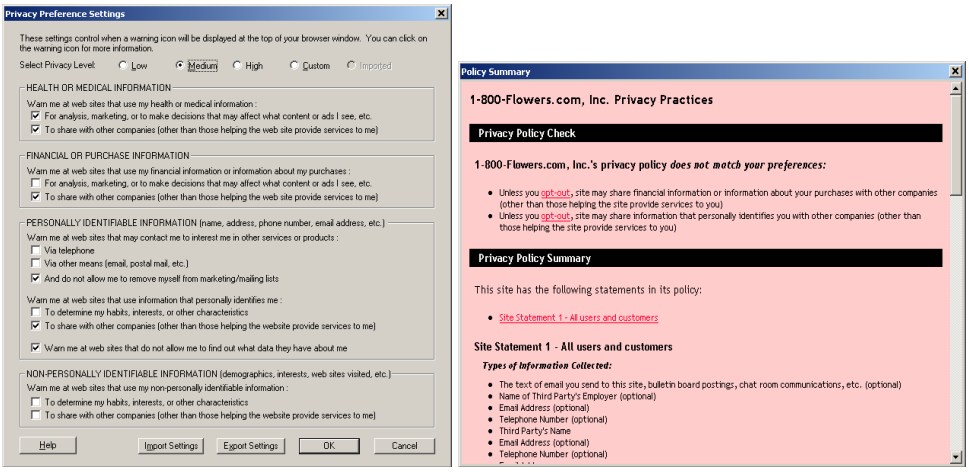
L'emploi d'agents intégrés aux navigateurs est très limité en raison du manque de précision sur le contrôle des informations que contiennent les cookies. Seuls des niveaux de confidentialité (low, medium, high) sont paramétrables avec ces navigateurs. Il est impossible à l'utilisateur d'affiner ses préférences, c'est-à-dire préciser des préférences en fonction des données qu'il voudra partager.

Tout refus de partager des informations avec le SP avec ses pratiques d'usage déclarées, entraîne immédiatement un accès refusé à sa page Internet, on perd ainsi en flexibilité et en confort d'utilisation. C'est la méthode « Take it or leave it ».

2.2.3.2 Les agents P3P

Bien qu'il existe des agents intégrés nativement dans des navigateurs, il est possible d'installer des agents qui travaillent en étroite collaboration avec les navigateurs les plus couramment utilisés sur le marché, on les appelle les plugins. Les agents P3P dispensent aux utilisateurs de lire les politiques de protection des données des SPs. Le travail essentiel de l'agent réside dans la comparaison des politiques en faisant intervenir l'utilisateur le moins possible afin qu'il puisse bénéficier d'une navigation la plus confortable possible.

C'est le cas de l'agent P3P, Privacy Bird, proposé par un grand opérateur de télécommunication aux Etats-Unis AT&T, rapidement après la sortie de la plateforme P3P. Privacy Bird est l'un des agents les plus connus sur le marché mais son utilisation est restreinte en raison de sa seule compatibilité avec Internet Explorer 6 et au-delà. Comme tout agent P3P, Privacy Bird compare les politiques de protection des données entre le client et le serveur, assiste le client pour le partage ou non de ses données, et est capable d'afficher les politiques de confidentialité du SP. Privacy Bird dispose d'une interface graphique affichant une icône d'un oiseau pouvant changer de couleur sur la barre du navigateur web. Selon la couleur de l'oiseau, l'utilisateur est averti de la nature des pratiques du SP. Par exemple, lorsque l'oiseau est vert, cela signifie que la politique des traitements des données correspond aux préférences de l'utilisateur. Il est tout à fait possible d'afficher un résumé de la politique du SP à l'aide des options du logiciel. Ce dernier est aussi en mesure d'expliquer à l'utilisateur les points qui ont manqué dans la comparaisons des politiques et lui demander s'il est prêt à vouloir partager les données manquantes à l'aide de l'option opt-out.
L'utilisateur peut manuellement décider des données qu'il veut partager avec le serveur, en cochant des cases indiquées sur le côté (voir figure 2.4) ou bien, utiliser le paramétrage assisté de gestion des données selon trois niveaux de confidentialité (low/medium/high). Un tel système de paramétrage assez ressemblant à la gestion des cookies sous Internet Explorer, où l'utilisateur peut définir ses préférences selon des niveaux sans réellement savoir à quoi correspondent ces niveaux. Aussi, ce logiciel ne fait que retourner des warnings. Ainsi, même si un site Web ne satisfait pas les préférences utilisateur, il pourra quand même s'y connecter alors que le site fera un usage non conforme à ses préférences de ses données. Le caractère « conformité non obligatoire » pour effectuer une transaction pose donc problème.



(a) **(b)**

Figure 2.4 Configuration des préférences utilisateur (a) présentation de la politique du SP (b) sous Privacy Bird

2.3 Limitations

En réponse aux besoins définis dans la section 2.1, nous avons illustré l'apport des langages de politiques de protection des données dans le processus de formalisation et de présentation des pratiques d'usages des SPs. cependant, ces formalisations restent insuffisantes dans leur présentation des axes réglementaires de protection des données. En effet, les éléments (Purpose/Recipient/Retention) définis par la plateforme P3P répondent partiellement aux trois derniers axes de protection seulement sur les six exposés (section 1.3.1 du chapitre 1).

Aussi, les implémentations exposées dans la section 2.2.3, manquent de précision dans la définition des préférences (en termes des données à révéler et d'usage associé) qui sont souvent limitées aux choix à faire sur les niveaux de protection qui ne sont pas personnalisables.
Ainsi, le besoin de spécification des règles de protection fines par l'utilisateur reste entier.

En réponse à ce besoin et afin de fournir une représentation plus complète des axes règlementaires de protection (section 1.3.1 du chapitre1), il est nécessaire qu'une solution technique propre, basée sur les axes de protection, inclut un contrôle formalisé de la révélation des données personnelles. En effet, d'un côté les requêtes pour les données personnelles et les déclarations pour leur utilisation doivent être formellement décrites avec des politiques de traitement des données. De l'autre, les données personnelles révélées côté utilisateur doivent être accompagnées par des règles de permissions et des restrictions explicites.

2.4 Le langage de politiques de contrôle d'accès XACML

2.4.1 Modèles de contrôle d'accès sensibles à la protection de la confidentialité

Les modèles de contrôle d'accès ne considèrent pas la protection des données données personnelles comme un premier objectif. Les modèles MAC (Mandatory Access Control), DAC (Discretionary Access Control), et RBAC (Role-Based Access Control) ne remplissent que quelques besoins de protection des données personnelles [Ni07]. Actuellement, de plus en plus de modèles s'intéressent à inclure des éléments de protection, et principalement l'élément « purpose » comme élément principal de contrôle d'accès. Seulement, la spécification cet élément n'est pas suffisante à elle-même pour définir des politiques de protection ou des préférences de l'utilisateur.

Qui Ni et al., proposent des extensions au modèle RBAC [Kal03] afin de prendre en

compte des contraintes liées à la protection des données. Pour ce, ils définissent une famille de modèles P-RBAC (Privacy-aware RBAC), où les politiques de protection sont exprimées à travers des affectations de permissions. Ces affectations ne correspondent pas aux permissions définies dans RBAC à cause de la présence des nouveaux éléments « purposes » et « conditions » dans les politiques d'accès. Un langage spécifique LC_0 a été proposé afin de permettre la définition des conditions. Une permission explicite définie alors les buts de l'action (l'élément « purpose »), sous quelles conditions, ainsi que les obligations qui doivent être accomplies après l'accès. Ce travail développe des algorithmes d'analyse de conflits, permettant de détecter les conflits entre les différentes affectations de permissions.

Donc, les trois extensions principales sont : l'élément « purpose », la définition des obligations, et un langage dédié aux conditions. Une permission sensible est ainsi modélisée selon des privilèges qui ont la forme générale suivante :

***role* × *action* × *data* × *purpose* × *conditions* × *obligation*.**

Le modèle de contrôle d'accès à base de « purpose » proposé en [Yan08], tend à appliquer des éléments de protection dans les domaines qualifié de non-confiance. Son objectif et de maintenir la consistance entre la politique de protection et les pratiques déclarées. Ils s'attachent à la formalisation des différents types de buts (purposes). Ensuite, les auteurs spécifient des invariants correspondants aux besoins de protection dans une politique de protection. L'objectif de représentation de ces invariants est de fournir une interprétation claire et non-ambigüe des politiques.

Les buts (purposes) sont divisés en deux classes : des buts désirés, et des buts d'accès. Les buts désirés sont liés à l'objet d'accès, spécifient ainsi les usages pour lesquelles une donnée est accédée. Les buts d'accès sont liés à l'accès aux données et spécifient les intentions pour lesquelles une donnée est accédée. Chaque demandeur doit déclarer le but d'accès, le système valide le but d'accès cité afin d'être sûr que l'utilisateur est permit pour le but d'accès. Le but d'accès doit être compatible avec le but désiré afin de permettre l'accès.
Cette classification de buts est dérivée de [Byu05]. Dans chaque organisation, les objets (données) sont organisés en utilisant des types d'objets. L'entité « rôle » a été étendue afin de prendre en compte le rôle conditionnel, qui est basé sur les attributs du rôle et les attributs du système [Yan08, Byu05]. En utilisant ces entités d'objets, de rôles, et de buts, les auteurs dans [Yan08] définissent les invariants suivants :

- un objet « donnée » est crée seulement si nécessaire pour un rôle conditionnel de la requête en cours,
- l'autorisation du rôle conditionnel et les contraintes d'accès aux données sont principalement basés sur la compatibilité entre le but désiré et celui d'accès.

La politique de protection dans ce travail, est principalement basée sur l'élément « purposes », mais ne prend pas en compte des besoins de protection tel le consentement ou l'accès selon le contexte. De l'autre côté, les entités utilisées tels les rôles conditionnels et les types d'objets augmentent la complexité d'administration des politiques.

Le PuRBAC (Purpose-Aware Rôle-Based Access Control) proposé dans [Mas08], étend le modèle RBAC en modélisant des besoins liés à la protection de la vie privée. Les buts

« purposes » représentent l'entité centrale. Ils représentent l'intermédiaire entre le rôle et les entités de permission. Le modèle définit des hiérarchies de rôles et de buts (pruposes). Il supporte l'expression des contraintes et des obligations, et les définie comme des conditions sur l'affectation des permissions aux différents buts. Ensuite, les buts (purposes) sont assignés aux rôles. La requête d'un utilisateur est formée par un identifiant de session, un but, et une permission requise. L'autorisation peut être requise pour des buts liés au rôle actif. Il y a une autre différence majeure avec le modèle RBAC, quand une requête est soumise au ADF (Access Decision Function), il peut soit refuser l'accès, ou définir une autorisation conditionnelle.

Les auteurs modélisent trois types de conditions : contraintes, pre-obligations, et post-obligations. Les contraintes sont utilisées pour vérifier l'information basée sur les variables des données dans le système. Par exemple, le consentement du possesseur des données est considéré comme une contrainte. Les Pre-obligations désignent que l'utilisateur doit exercer quelques actions avant d'avoir accès. Ceci peut inclure par exemple : la réauthentification de l'utilisateur avant qu'il accède à des données sensibles, ou l'ajustement des données. Les post-obligations, représentent les obligations après autorisation d'accès. Ceci peut concerner par exemple, la politique de rétention d'une donnée qui programme la suppression des données.

Les auteurs argumentent que les buts (purposes) d'utilisation sont considérés comme une entité séparée car la politique de protection habituelle dépend du but d'utilisation, deuxièmement ils supposent qu'il y a une relation entre la notion de « purpose » dans les politiques de protection et la notion de « rôle » en RBAC, troisièmement décomposer la politique à différentes entités et relations entre elles, fait de la gestion de différentes parties des politiques aussi indépendantes que possible.
Ainsi, l'hypothèse que le « purpose » est une entité séparée dans une telle modélisation de politiques va augmenter la complexité de la politique. Par exemple, la gestion des hiérarchies et des conflits doit couvrir cette nouvelle entité.

Ardagna et al. [Ard05] proposent un nouveau modèle sensible à la protection de la vie privée. Les auteurs identifient quatre types de politiques afin de répondre aux besoins de leur modèle sensible à la protection de la vie privée :

- politiques de contrôle d'accès : comme dans le contrôle d'accès traditionnel, ils gouvernent l'accès aux données et services gérés par le groupe,
- politiques de révélation : gouvernent la révélation des propriétés ou des informations personnelles du groupe et spécifient les conditions sous lesquelles elles peuvent être révélées,
- politiques de traitement des données : spécifient comment les données personnelles sont utilisées par l'autre partie après révélation,
- politiques de filtrage : filtrent la réponse retournée à l'autre partie pour prévenir la révélation des informations sensibles liées à la politique elle-même.

Les politiques de contrôle d'accès et les politiques de révélation sont étudiées dans [Ard05]. Une règle de contrôle d'accès ou de révélation possède la forme suivante :

subject WITH subject-expression CAN action FOR purpose ON object WITH object-expression IF conditions FOLLOW obligations

où : subject, object, et action sont des entités auxquelles la règle réfère. subjectexpression et object-expression réfèrent à des conditions que le sujet et l'objet doivent satisfaire respectivement. Purpose, spécifie comment la donnée va être utilisée. Conditions, est une expression booléenne que le serveur doit suivre quand il gère les données personnelles. Il n'y a pas d'indications sur le type des conditions qu'on peut exprimer.
La tâche d'administration n'a pas été évoquée dans ce modèle.

Fisher-Hubner et Ott, se sont intéressés à la définition d'un modèle de contrôle d'accès généralisé [Fis98]. Ce modèle inclue deux autres principes : la nécessité de traitement et la liaison au « purpose ». Les auteurs spécifient que l'utilisateur peut accéder aux données/services, si cet accès est nécessaire pour accomplir la tâche en cours, et seulement s'il est autorisé à faire cette tâche. L'accès utilisateur est contrôlé en appliquant une procédure de transformation pour laquelle la tâche courante de l'utilisateur est autorisée. Ensuite, le but de la tâche utilisateur en cours doit correspondre aux buts pour lesquels les données personnelles on été collectées.

2.4.2 Principes de XACML

XACML (eXtensible Access control Markup Language) [OAS05] a été spécifié pour implémenter les politiques de sécurité le plus précisément possible. Les politiques de haut niveau traduite dans la représentation XACML à travers des profiles. Afin de supporter la protection des données, les concepteurs ont défini un profil nommé « privacy profile » [OASb05], dont le nouvel élément principal rajouté fut l'élément « purpose » également, afin de justifier le but de la collecte. Néanmoins, XACML est doté d'une structure de langage extensible, et adaptable à tous ses niveaux de granularité.

Le langage XACML défini par l'OASIS (Advancing open standards for the information society) , est le langage libre standardisé le plus utilisé, satisfaisant ces propriétés.
XACML (eXtensible Access control Markup Language) [OAS05, Mos04,God05] est une spécification XML [Pol05, Con05] de l'OASIS pour la définition de politiques de contrôle d'accès qui devient un standard en 2003. XACML fournit un langage universel de description des politiques de contrôle d'accès de la forme : (qui peut faire quoi et à quel moment). De plus, ce langage fournit une architecture pour la mise en œuvre du contrôle d'accès : un protocole de type requête/réponse.
La politique de contrôle d'accès permet de définir les droits des utilisateurs (personne ou application) sur les ressources (données, services, etc.). XACML est un langage d'expression puissant qui utilise la logique pour combiner les règles et exprimer des politiques de protection.

Avec XACML, il est possible de définir des règles de contrôle d'accès structurées en politiques et ensembles de politiques. Ces règles permettent de répondre aux requêtes qui demandent d'effectuer des opérations sur des ressources. La réponse peut être soit positive (permit) soit négative (deny). XACML fournit également un environnement architectural pour concevoir et réaliser un système de contrôle d'accès.

Dans la suite, nous définissons les principes du langage, puis nous décrivons l'architecture de l'environnement XACML. Enfin, nous introduisons le diagramme de flux de XACML.

2.4.2.1 Requête

L'accès à une ressource protégée s'effectue par le biais d'une requête (Request). Il doit être possible de formuler une requête telle que définie dans le tableau 2.1, avec les éléments suivants :

- L'identité des demandeurs qui seront appelés les sujets (Subjects). Une requête dans XACML peut être initiée par plusieurs sujets ;
- La ressource à accéder (Resource) ;
- L'action à effectuer (Action).

	Element	Attributes	Values
Request	Subject	Identifier	…
		Role	…
		Institution	…
	Resource	Identifier	…
		Owner	…
	Action	Identifier	…

Tableau 2.1 Requête XACML au format tabulaire

Ces éléments peuvent posséder des propriétés. Un sujet peut être défini par un identificateur, une institution à laquelle il appartient, un rôle etc. ; une ressource peut être caractérisée par un identificateur, un contenu structuré et un type ; idem pour l'action qui peut être définie par un identifiant.

Les propriétés des sujets, ressources et actions sont appelées attributs. Chaque attribut possède une valeur. Une requête peut contenir, en plus des informations concernant le sujet, la ressource et l'action, des informations optionnelles concernant l'environnement. Ces informations sont les propriétés qui ne sont associées ni au sujet, ni à la ressource, ni même à l'action, par exemple des informations sur un horaire ou une date.
Nous modélisons ainsi la requête dans XACML par le diagramme objet de la figure 2.5.

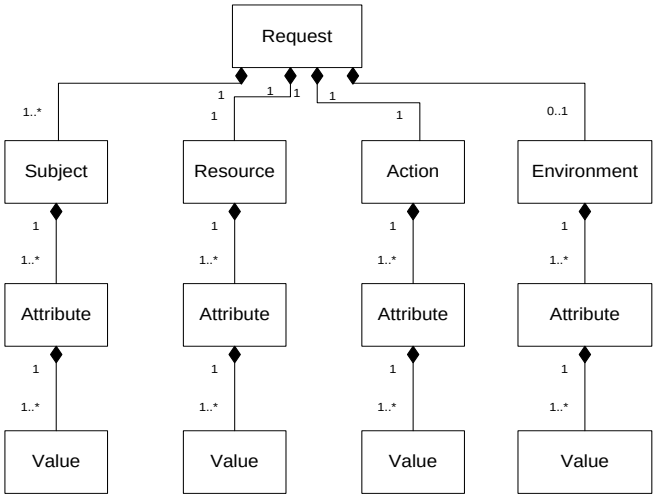


Figure 2.5 Diagramme objet d'une requête XACML

Un système de contrôle d'accès tient compte de la requête et des informations qu'elle contient pour prendre une décision.

2.4.2.2 Règles de contrôle d'accès

Pour contrôler l'accès à une ressource avec le langage XACML, il faut définir un ensemble de règles (*rules*). Chacune d'entre elles définit les éléments suivants :

- Le (les) sujet(s) concernés ;
- La (les) ressources à accéder ;
- Les actions demandées ;
- Les conditions à satisfaire ;
- La décision à renvoyer.

La décision constitue la réponse d'une règle dans le cas où les conditions imposées sont satisfaites. Elle peut être soit positive, pour permettre l'accès à la ressource (permit), soit négative, pour en refuser l'accès (deny).

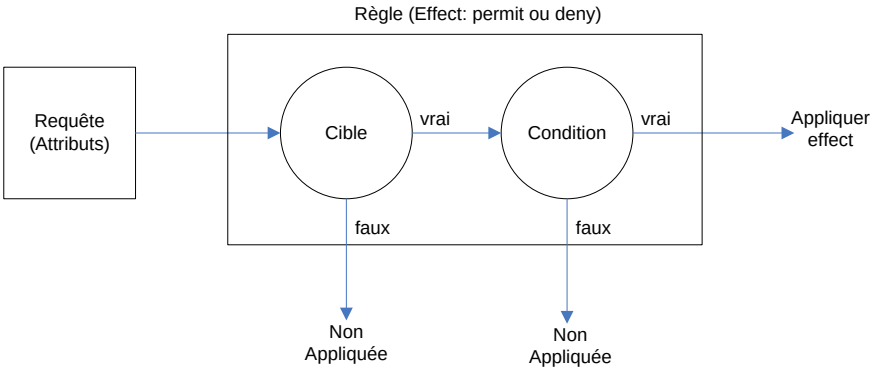


Figure 2.6 Évaluation des requêtes dans XACML

Comme mentionné dans la section précédente, une demande d'accès spécifie le sujet, la ressource à accéder, l'action à exécuter et les propriétés de l'environnement. Le système de contrôle d'accès doit évaluer ces paramètres et selon leurs propriétés générer une réponse *permit* ou *deny*.

La manière la plus directe pour aboutir à une décision est de vérifier d'abord, si une règle peut être appliquée à une requête ou non. Ceci est effectué en évaluant la cible (target) de la règle avec les éléments constituant la requête (section 2.4.1.1). Plus concrètement, la vérification si les éléments sujets, ressources, et actions de la cible correspondent à des valeurs particulières des éléments sujets, ressources et actions de la requête.
Ensuite, une règle peut spécifier un ensemble de conditions supplémentaires et plus complexes à vérifier. Ainsi, si une requête correspond à la cible d'une règle, alors les conditions de cette règle seront évaluées, et si elles sont satisfaites, la réponse sera l'effet spécifié. Sinon, si la cible d'une règle ne correspond pas à la requête, ou bien si ses conditions ne sont pas satisfaites alors cette règle ne sera pas appliquée (voir figure 2.6).

Une décision globale est générée à partir d'un ensemble de règles (*rule*). Les règles sont regroupées en politiques (*Policy*). Les politiques peuvent être regroupées en ensembles de politiques (*policySet*).

2.4.2.3 Politiques de contrôle d'accès (Policy)

Une politique (*Policy*) regroupe plusieurs règles de contrôle d'accès. En effet, l'objectif des langages de contrôle d'accès est de pouvoir fédérer plusieurs politiques. Il est clair qu'il faut grouper l'ensemble des règles d'autorisation en des ensembles, voire même des sous-ensembles, comme nous allons le voir plus loin, et ce, pour optimiser leur exploitation. A l'instar d'une règle, une politique doit avoir une cible qui restreint son

champ d'application à un ensemble limité de requêtes qui satisfont des conditions bien particulières.

Une fois une politique est appliquée à un contexte de requête, toutes les règles qui sont contenues dans la politique sont appliquées. Une sélection plus fine est alors obtenue. Les cibles des règles limitent le nombre de règles appliquées dans une politique.

Puisque plusieurs règles peuvent être contenues dans une politique, et comme à chaque règle est associée une décision, alors nous pouvons avoir plusieurs règles qui s'appliquent à un contexte de requête particulier. Par conséquent, plusieurs décisions peuvent être prises. La façon la plus simple de faire face à de telles situations est de spécifier au niveau de chaque politique la manière de combiner les différentes règles et leurs décisions. Ces comportements sont appelés en XACML, algorithmes de combinaisons des règles (Rule Combining Algorithms). Quatre comportements standards sont définis :

- Permit-overrides : si au moins une règle appliquée retourne un permit, alors la réponse de la politique sera permit. Si ce n'est pas le cas et il y a au moins une règle qui retourne deny, alors la réponse de la politique sera deny.
- Deny-overrides : si au moins une règle appliquée retourne un deny, alors la réponse de la politique sera deny. Si ce n'est pas le cas et il y a au moins une règle qui retourne permit, alors la réponse de la politique sera permit.
- First-applicable : appliquer la première règle qui s'applique
- Only-one-applicable : il faut qu'il y ait une seule règle applicable dans la politique. Si ce n'est pas le cas, la politique ne génère pas de réponse.

Le système de contrôle d'accès peut également demander des actions supplémentaires à exécuter en conjonction avec la décision générée. Ces actions sont appelées des obligations.

2.4.2.4 Ensemble de politiques (PolicySet)

Il est encore possible de regrouper les politiques de contrôle d'accès en des ensembles de politiques (*PolicySet*) et ce, pour structurer les politiques et les règles. Nous obtenons alors une structure arborescente d'ensembles de politiques puis de politiques et enfin de règles.

De la même manière, les ensembles de politiques possèdent une cible qui détermine leur applicabilité face à des demandes d'autorisation. Aussi, et du fait que dans un même ensemble de politiques, plusieurs politiques peuvent s'appliquer et générer des réponses différentes, les algorithmes de combinaisons sont encore utilisés par les ensembles de politiques, mais cette fois-ci nous parlons de combinaison des politiques (*Policy Combining Algorithms*).

Enfin, nous pouvons associer des obligations à exécuter en plus de la décision issue d'un ensemble de politiques.

La figure 2.7 présente la hiérarchie entre les ensembles de politiques, les politiques et les règles de contrôle d'accès. Un ensemble de politiques doit spécifier une cible en plus d'un algorithme de combinaison qui permet de combiner les décisions issues des politiques et des ensembles de politiques enfants. Une politique doit aussi avoir une cible ainsi qu'un algorithme de combinaison, qui permet de choisir une décision parmi

celles des règles filles. Enfin, une règle spécifie sa cible, sa condition et son Effect (permit ou deny).

2.4.2.5 Réponse

La réponse d'un système de contrôle d'accès peut être soit positive (permit) soit négative (deny), soit encore indéterminée (indetermined) quand une erreur survient ou que le système n'arrive pas à répondre. Si le système ne trouve aucune règle (politique ou ensemble de politiques) à appliquer, alors la réponse devient non-appliquée (Not Applicable). Notons que la seule réponse qui permette l'accès à une ressource est permit.

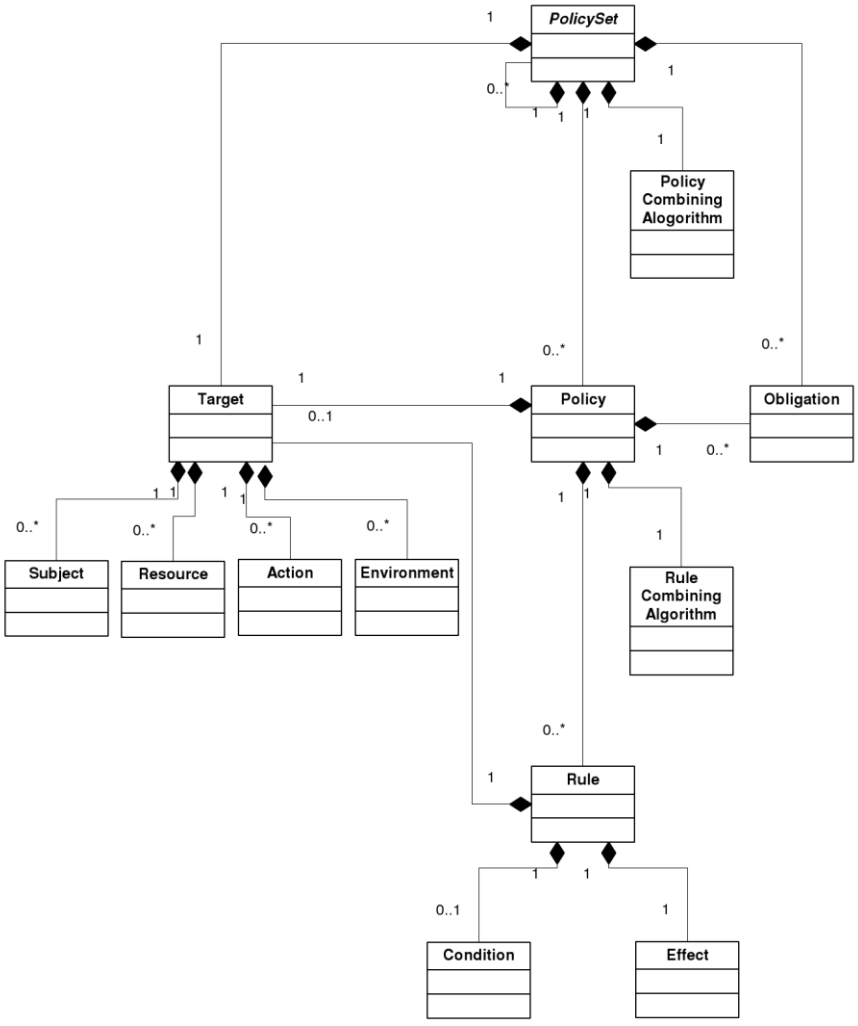


Figure 2.7 Diagramme objet du langage de politiques XACML

2.4.3 Architecture de XACML

Le langage XACML n'a pas seulement apporté une syntaxe pour l'expression des politiques de contrôle d'accès, il a aussi apporté une architecture et des concepts qui fournissent les grandes lignes pour concevoir un système de contrôle d'accès. Cette architecture vise à atteindre plusieurs objectifs :

- Assurer une protection efficace des ressources et ce, du point de vue du contrôle d'accès ;
- Permettre de concevoir un système indépendant de la plate-forme utilisée ;
- Permettre d'intégrer le système de contrôle d'accès dans des applications déjà existantes.

L'architecture globale du langage consiste en plusieurs composantes collaborant entre elles comme présenté dans la figure 2.8. Nous détaillerons dans la suite ces composantes et les interactions qui les animent.

2.4.3.1 Point d'administration des politiques (PAP)

Le point d'administration ou *PAP* pour *Policy Administration Point*, est l'entité qui crée les règles, les politiques et les ensembles de politiques de contrôle d'accès.

2.4.3.2 Point d'application des politiques (PEP)

Le point d'application des politiques ou *PEP* (*Policy Enforcement Point*) agit comme interface avec les entités extérieures (applications). Le mécanisme d'échange est sous forme de requêtes d'accès. Le *PEP* se charge d'envoyer la requête reçue au Context Handler, qui la transfert au PDP pour avoir une réponse à la requête (décision). Selon le résultat de la décision, le *PEP* accorde ou refuse l'accès.

2.4.3.3 Gestionnaire de contexte (Context Handler)

Le gestionnaire de contexte (*Context Handler*) a pour rôle de transformer les requêtes initiales dans un format spécifique appelé contexte XACML. Il contient les spécifications (attributs et valeurs) du sujet, de la ressource et de l'action. Le gestionnaire de contexte prend en charge également la transformation de la réponse en un format compréhensible par l'entité qui a généré initialement la requête.

2.4.3.4 Point de décision des politiques (PDP)

Le centre de décision des politiques ou *PDP* (*Policy Decision Point*) est l'entité en charge de sélectionner les règles, les politiques ou les ensembles de politiques qui sont applicables à une requête donnée. Il évalue les cibles et les conditions afin d'aboutir à une décision.

2.4.3.5 Source d'information de politiques (PIP)

La source d'information de politique ou *PIP* (*Policy Information Point*) a pour rôle d'extraire les informations supplémentaires qui ne sont pas présentes dans la demande d'accès. Le PIP peut lui-même chercher les informations dans des sources externes. Ces sources externes peuvent être une base de données, un annuaire d'utilisateurs etc.

2.4.3.6 Diagramme de flux XACML

La figure 2.8 montre le flux de données dans un environnement XACML. Les étapes de traitement d'une requête XACML sont :

1. Le PAP configure les politiques ou ensembles de politiques pour le PDP.
2. Le demandeur envoie une requête au PEP.
3. Le PEP envoie la requête dans son format d'origine au gestionnaire de contexte.

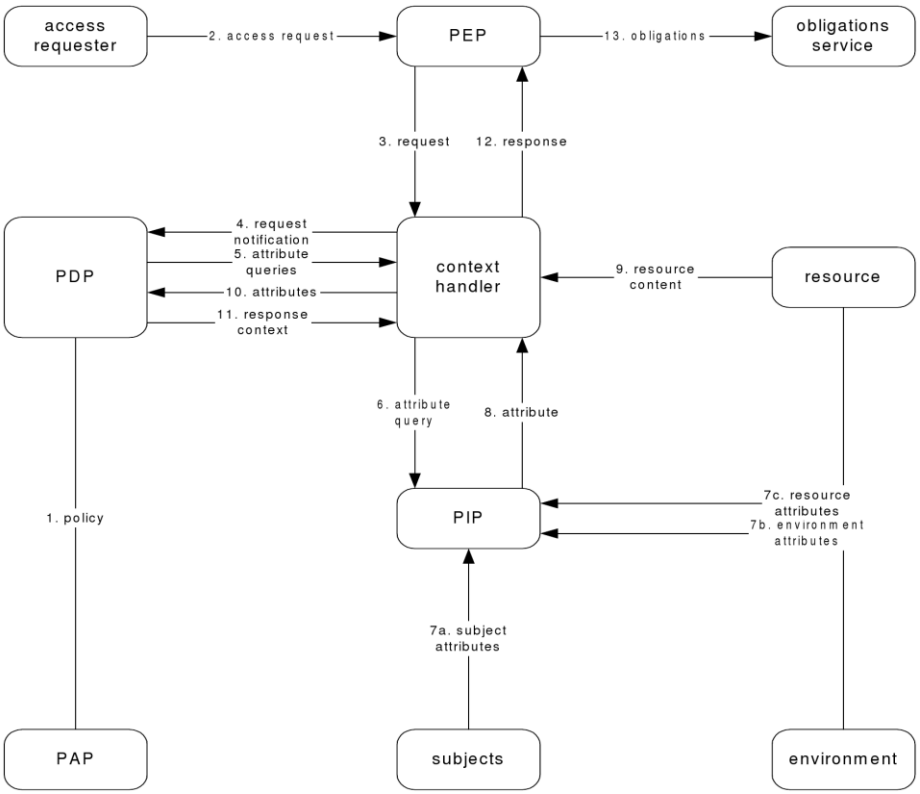


Figure 2.8 Architecture de contrôle d'accès XACML

4. Le gestionnaire de contexte extrait les attributs des sujets, ressource et action. Il génère ainsi une requête au format XACML et il l'envoie au PDP.
5. Le PDP analyse la requête et au besoin envoie une demande d'attributs supplémentaires non contenus dans la requête au gestionnaire de contexte. En effet, pour pouvoir évaluer les cibles et les conditions des politiques et des règles de contrôle d'accès, il est nécessaire de connaitre les valeurs de certains attributs. Par exemple, si une requête spécifie juste la ressource à accéder, le PDP a besoin du propriétaire de la ressource.
6. Le gestionnaire de contexte demande au PIP les attributs manquants.
7. a, b et c : Le PIP extrait les informations nécessaires sur le sujet et la ressource à partir de sources externes.
8. Le PIP renvoie les valeurs des attributs demandés.
9. Au besoin, et dans le cas où la ressource contient des données structurées (fichier XML), elle envoie des informations concernant son contenu au gestionnaire de contexte.
10. Les valeurs des attributs sont envoyées au PDP.
11. Le PDP évalue les politiques disponibles par rapport à la requête et génère une réponse au format XACML. Cette réponse est envoyée au gestionnaire de contexte.
12. Le gestionnaire de contexte transforme la réponse XACML en un format compatible avec la requête initiale émise par le demandeur.
13. Le PEP vérifie les obligations et les joint à la réponse.
14. Le PEP envoie sa réponse à l'entité qui a demandé l'accès.

Ce modèle d'architecture possède différents points extensibles que nous pouvons adapter au contexte de nos travaux comme sera montré dans la section suivante.

2.5 Complémentarité des langages de politiques de protection des données et des langages de contrôle d'accès

Deux besoins principaux (section 2.1) émergent des problématiques « *représentation* des axes réglementaires », et « *contrôle* explicite de l'usage des données personnelles » définies dans l'introduction de ce mémoire. Le premier est de définir les éléments de politiques de protection des données tirés des six axes exposés dans la section 1.3.1. Le deuxième, consiste à exprimer ces éléments dans des politiques applicables de contrôle d'accès aux données personnelles.

Il ressort des travaux cités dans les sections 2.2 et 2.4 respectivement, que P3P et XACML sont les contributions qui répondent au mieux à ces besoins.

En effet, P3P permet de résoudre de manière simple le problème de l'information et de consentement de l'utilisateur, en étant capable de décrire les divers aspects relatifs au traitement des données (justification, conservation, transmission). Si les listes de choix

prédéfinies pour la spécification du type de traitement, de leur justification ou du type de données restent limitées, elles sont extensibles par le biais de schémas XML personnalisés. Cette raison nous pousse à favoriser au maximum l'interopérabilité de notre solution avec P3P, préférentiellement à d'autres langages de politiques moins génériques et moins répandus.

Il faut toutefois rester conscient des limitations de P3P. En effet et surtout, P3P exprime la politique d'un SP indépendamment de tous les types de réglementation que nous avons pu identifier. Il reste donc ici un travail d'information et de raisonnement à effectuer.

En revanche, l'expression des six axes de protection de données avec des éléments de politiques de protection de données à elle seule n'est pas suffisante, il faut l'intégrer à une solution d'autorisation offrant un niveau de granularité très fin permettant la gestion d'accès aux données personnelles avec des règles applicables. Nos recherches nous ont amenés à étudier les infrastructures de gestion d'accès à base de politiques. Le choix de ces dernières est justifié par l'apport du complément nécessaire à la protection des données pour permettre l'expression des règles d'autorisation et de construction des politiques d'accès avec un niveau de précision assez fin.

Comme présenté dans la section 2.4, différents modèles de contrôle d'accès sont proposés, mais aucun ne permet de prendre en compte les six axes de protection. Nous allons donc étudier XACML (norme conçue par OASIS) en particulier. La conformité de ce dernier à une norme est un facteur régulateur. De plus, XACML adopte le concept *d'attributs* ce qui permet d'exprimer toutes les caractéristiques des sujets, ressources, actions et environnement sous la forme d'attributs, et ainsi apporter des extensions en se basant sur ce concept.

XACML fournit également une architecture et des concepts définissant les grandes lignes de conception d'un système de contrôle d'accès. L'architecture de contrôle d'accès est extensible et modulaire, elle définit différents points sur lesquels se base la gestion d'accès qui peuvent être adaptés à notre contexte de travail.

Le choix de ces deux outils est justifié par leur caractère libre et extensible, ce qui nous permet d'apporter les modifications nécessaires en vue de répondre à nos besoins.
Enfin, leur caractère complémentaire qui est appuyé par une analyse de Sun Microsystems [The] rapportant que : "*P3P policies and XACML policies serve complementary purposes. P3P policies express privacy policies in terms that human users can understand; they express externally published policies in a generalized, high-level form. XACML policies express the same privacy policies in terms that computer access control mechanisms can understand and enforce; they express policies in a fine-grained, internally applicable form. The two levels of policy should be consistent with each other, and together they enable an auditor to determine whether the enterprise is complying with its stated privacy policies.*"

2.6 XPACML

Nous définissons dans notre approche un langage dédié à la protection des données personnelles, et permettant l'échange de politiques dans un format structuré, et avec un vocabulaire adapté aux axes de protection des données (section 1.3.1 du chapitre 1).

Basé sur le langage XACML, EPAL (*Enterprise Privacy Authorization Language*) permet d'exprimer des conditions sur les règles. Ces conditions sont exprimées en XACML. Tout comme XACML, il peut être utilisé pour appliquer le contrôle de l'utilisation des données. Cependant, l'architecture pour réaliser cette tâche n'est pas aussi bien définie que dans XACML. On pourra utiliser des applications tierces pour appliquer cette fonction comme par exemple Tivoli Privacy Manager d'IBM. EPAL utilise une architecture semblable à celle de XACML, on y retrouve donc un PEP et un PDP.

Dans notre approche de protection des données, nous considérons les données personnelles comme des ressources à protéger. Nous avons mentionné dans les deux sections précédentes que XACML est un langage qui adopte le concept d' « *attribut* », ce qui permet d'exprimer toutes les propriétés des éléments d'une règle d'accès à une donnée avec des attributs. Nous profitons de cette caractéristique fondamentale pour apporter les extensions nécessaires au modèle de base du langage XACML, afin que ce dernier puisse répondre aux besoins de présentation et de contrôle définis dans la section 2.1.Aussi, nous définissons dans cette section :

- Les contraintes à considérer dans les règles et politiques de protection des données ;
- Le modèle du langage XPACML ;
- L'architecture qui supporte les composantes principales assurant le contrôle d'accès aux données de l'utilisateur, ainsi que les différentes contributions exposées dans l'introduction de ce mémoire.

2.6.1 Contraintes à considérer dans les politiques de protection des données XPACML

Les politiques de protection des données sont utilisées pour représenter et décrire les contraintes légales ou réglementaires en matière de protection des données personnelles, pesant sur le contexte d'exécution. Elles peuvent être également utilisées comme un cadre pour exprimer les préférences de l'utilisateur, et les futures pratiques du SP avec les données recueillies.
Les six axes de la protection des données personnelles définis dans la section 1.3.1 du chapitre 1, peuvent être représentés sous forme de contraintes sur le traitement des données personnelles, comme illustré dans la figure 2.9.

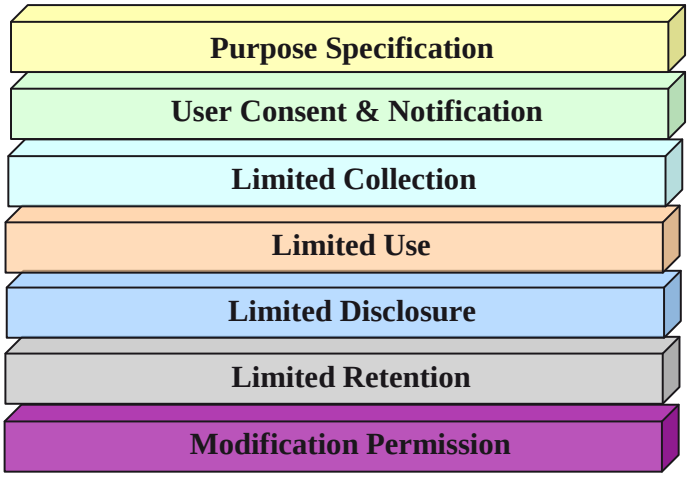


Figure 2.9 Contraintes à considérer dans les politiques de protection des données personnelles

Ces contraintes se basent principalement sur la spécification du but de la requête pour laquelle les données personnelles sont collectées, le consentement donné par l'utilisateur, les limitations sur la collecte, l'usage, la révélation et la rétention des données.

De ce fait, nos travaux prennent en compte les tags (<purpose>, <recipient>, et <retention>) définis dans la section 2.2.1, que nous complétons pour couvrir les axes de protection (cf. section 1.3.1).

Ainsi, nous nous focalisons dans la section suivante sur :

- La définition d'un ensemble d'éléments complémentaires aux tags de protection <purpose>, <recipient>, et <retention> de la plateforme P3P, et permettant la représentation technique de ces contraintes ;
- L'extension des principes du langage XACML afin d'exprimer ces éléments sous forme de politiques de protection des données, spécifiant les restrictions explicites sur l'accès aux données et les futurs usages de ces dernières pour un SP donné.

2.6.2 Principes du langage XPACML

La figure 2.10 illustre le modèle du langage XPACML. Les extensions que nous avons apportées au modèle de base du langage XACML sont représentées par le biais de classes et de méthodes en bleu. Nous détaillons les modifications apportées aux requêtes, aux règles, aux politiques, et aux ensembles de politiques XPACML. Le langage XPACML est utilisé à la fois pour l'expression des politiques de traitement des données par SP,

comme pour l'expression des préférences de l'utilisateur en termes de protection des données.

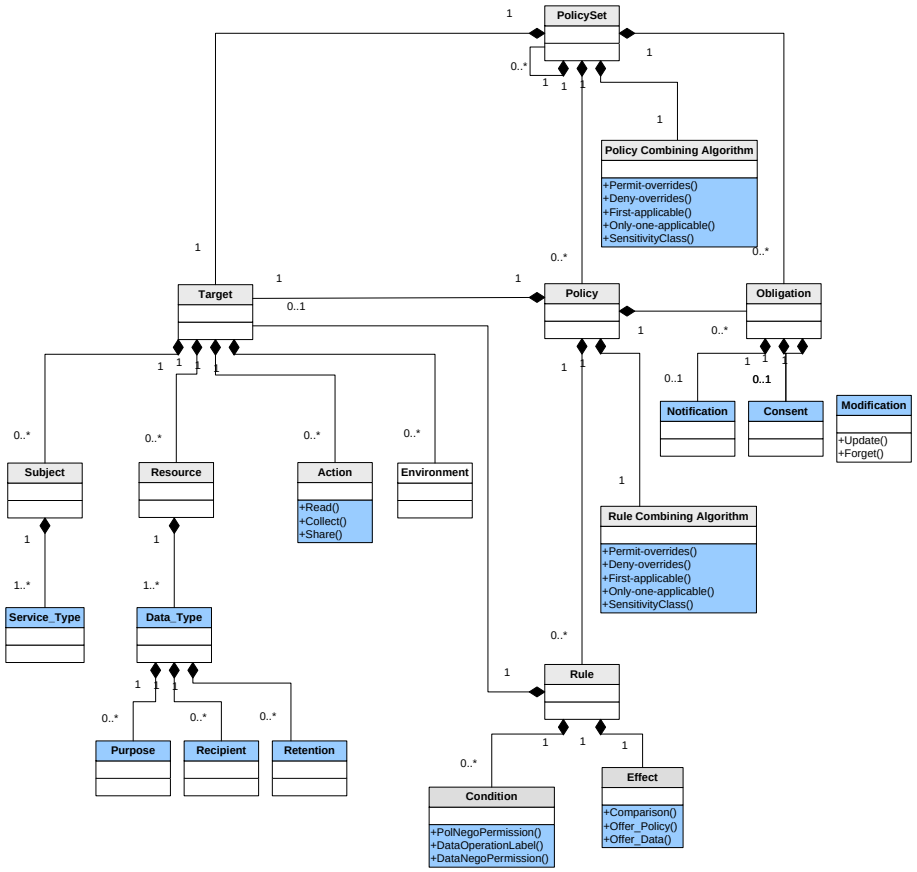


Figure 2.10 Structure du langage XPACML (modèle de langage de politique)

2.6.2.1 Requête

En référence au tableau 2.1 présentant une requête XACML dans un format tabulaire, une demande d'accès à une donnée personnelle dans notre système est une requête XPACML qui étend les éléments de base d'une requête XACML.
Ainsi, une requête XPACML se présente avec les éléments (`SERVICE_TYPE`, `DATA_TYPE et POLICY`) indiqués dans le tableau 2.2.

	Elements	Attributes	Values
Request	Subject	Identifier	…
		Role	…
		Institution	…
		SERVICE_TYPE	…
	Resource	Identifier	…
		DATA_TYPE	…
		POLICY	…
	Action	Identifier	…

Tableau 2.2 Requête XPACML au format tabulaire

- Comme une requête XACML, l'identité du SP demandeur des données est formalisée par l'élément *Subject*. Nous étendons cette identification avec la spécification du type de service en question. L'élément d'extension utilisé est appelé SERVICE_TYPE. Il peut prendre la valeur ALL, couvrant tous les types de services. En effet, l'identification du type de service en cours permet de :
  - identifier le type de données autorisées,
  - identifier le type de règle à appliquer,
  - limiter le champ d'action de la règle de protection à appliquer,
  - juger la pertinence de la justification fournie par le SP.

La formalisation des différents types de services que supporte le langage XPACML est établie via un modèle sémantique dédié à la spécification des différents aspects du langage. Le modèle sémantique est abordé en détail dans la section 3.5 du chapitre 3.

Nous définissons également l'élément SERVICE_TYPE_DESCENDANTS, qui sert à spécifier si la règle définie est applicable par héritage aux descendants du service de type SERVICE_TYPE.

- Comme une requête XACML, la donnée à accéder est formalisée par l'élément *Resource*. Nous étendons cette identification avec la spécification du type de la donnée en question. L'élément d'extension utilisé est appelé DATA_TYPE. Il peut prendre la valeur ALL, couvrant tous les types de données. En effet, l'identification du type de la donné à accéder permet de :
  - identifier le type de la règle à appliquer,
  - juger la pertinence de la justification fournie par le SP,
  - identifier l'élément POLICY associé. Cet élément englobe en effet les pratiques déclarées par le SP pour la donnée requise, en termes de tags P3P <purpose>, <recipient>, et <retention>.

La formalisation des différents types de données que supporte le langage XPACML est établie via un modèle sémantique dédié à la spécification des différents aspects du langage. Le modèle sémantique est abordé en détail dans la section 3.5 du chapitre 3.

Nous définissons également l'élément `DATA_TYPE_DESCENDANTS`, qui sert à spécifier si la règle définie est applicable par héritage aux descendants de l'élément de type `DATA_TYPE`.

A l'élément `DATA_TYPE` défini, nous associons les déclarations d'usage annoncées par le SP (tags <purpose>, <recipient>, et <retention> de l'élément `POLICY`).

- Comme dans une requête XACML, l'action à effectuer est formalisée par l'élément *Action*. Nous avons enrichi cet élément avec des actions spécifiques au contexte de protection des données. Cet élément peut ainsi prendre une valeur parmi les suivantes :
  - `READ` : utilisée quand le SP requiert la donnée seulement pour la transaction en cours,
  - `COLLECT` : utilisée quand le SP souhaite stocker la donnée collectée,
  - `SHARE` : utilisée quand le SP envisage de partager la donnée avec les tierces parties (listées dans l'élément `RECIPIENTS`).

2.6.2.2 Règle de contrôle d'accès

A l'instar d'une règle XACML, une règle de contrôle d'accès (*rule*) XPACML (figure 2.10) se base sur les éléments définis dans la requête XPACML représentant sa cible d'évaluation (*target*) pour en générer une décision (`EFFECT`) :

- Le (les) sujet (s) concerné(s) (`SERVICE_TYPE`),
- La (les) ressource (s) à accéder (`DATA_TYPE`) avec les déclarations d'usage autorisées (tags <purpose>, <recipient>, et <retention> de l'élément `POLICY`),
- Les actions autorisées pour exécution sur la ressource (`ACTION`),
- La décision à renvoyer (`EFFECT`), après évaluation des trois premiers paramètres.

En effet, la cible (*target*) avec les trois premiers éléments, représente une première étape pour savoir si une règle XPACML peut être appliquée à une requête donnée ou pas. Ainsi une décision d'une règle XPACML peut traditionnellement être soit positive (`Permit`), pour permettre l'accès à la donnée requise, soit négative (`Deny`), pour en refuser l'accès, en se basant sur les tags de l'élément `POLICY` fourni par le SP.
Ensuite, nous avons enrichi les éléments (`CONDITION`) et (`EFFECT`) pour une règle XPACML afin d'offrir plus de possibilités à l'utilisateur. Plus particulièrement, nous lui donnons la possibilité de spécifier :

- une condition (`PolNegoPermission`) sur les tags de l'élément `POLICY`, la réponse sera alors une proposition (`EFFECT`) (`Offer_Policy`) dans le cas où une négociation de politiques est autorisée,
- une condition (`DataNegoPermission`) sur les combinaisons des données requises. Dans le cas où une négociation des données est autorisée, la réponse sera l'`EFFECT` (`Offer_Data`) détaillé par le processus de négociation des données (section 5.5 du chapitre 5).
- des conditions sur les types de services faisant la requête (`SERVICE_TYPE`) et la (les) justification(s) fournie(s) (<purpose>), par le biais de la condition (`DataOperationLabel`). Cet `EFFECT` se base sur le concept « *type de sensibilité* » (`SENSITIVITY_TYPE`), que nous avons défini. Ce concept permet aux données typées d'être sujettes à différentes opérations de filtrage telles que la transformation ou la perturbation avant leur révélation au SP.

Nous présentons dans le tableau 2.3 les différents types de sensibilité que nous avons pu identifier pour une donnée élémentaire.

Sensitivity Type	**Description**
UNMODIFIED	Value returned as-is.
ABSTRACTION_LEVEL	Value returned according to the level of data in the hierarchy
QUANTIZATION	Value returned after introduction of some random error within a specified range.
PERTURBATION	Value returned after value has been re-mapped.
AGGREGATE	Value is not returned but aggregate and another derived value may be returned.
EXISTENTIAL	Value is not returned but may be used in conditional expression.
BLOCKING	No value is returned.

Tableau 2.3 Description des types de sensibilité

Les types `UNMODIFIED` et `BLOCKING` représentent les deux extrémités du spectre de la révélation des données, permettant ainsi à la donnée en question d'être révélée dans sa forme originale ou pas du tout. Ces deux types sont équivalents aux décisions `Permit` et `Deny` de l'élément `EFFECT`.

Le type `ABSTRACTION_LEVEL` exprime le niveau de précision de la donnée à révéler en se basant sur la position de la donnée dans la hiérarchisation des données fournie dans la section 3.5.

Nous avons également défini des types permettant la réduction des résultats (données) retournés au SP. En effet, le type `QUANTIZATION` permet de modifier la donnée requise avec une marge d'erreur. La `PERTURBATION` peut être appliquée pour préserver quelques propriétés statistiques de la donnée requise sans révéler la valeur actuelle de la donnée. La génération de pseudonymes peut protéger l'anonymat des utilisateurs, les noms, et les identifiants.

Un utilisateur peut ne pas vouloir révéler certaines valeurs de données, mais le souci de vie privée peut ne pas être aussi important au point de bloquer la donnée complètement. Ainsi, il est acceptable de révéler quelques résultats dérivés en se basant sur ces valeurs. Le type `EXISTENTIAL` permet à une valeur de données d'être utilisée seulement dans la portion d'évaluation conditionnelle d'une requête. Le type peut être configuré pour supporter quelques opérations de comparaison et pas d'autres.
Le type `AGGREGATE` est basé sur l'idée qu'une valeur agrégat comme `SUM` et `COUNT` peut représenter des informations moins sensibles que les valeurs individuelles sur lesquelles l'agrégat est basé.

Il y a d'autres types de sensibilité que nous avons définis pour les combinaisons de données. Leur spécification est reportée à la section 5.6 du chapitre 05 dédié à la négociation des données possible avec des combinaisons de données.

Considérons l'exemple de la règle simple de la figure 2.11 pour illustrer son fonctionnement.

- Cible :
  - Le sujet (le type du service) est *E-Commerce*
  - La donnée requise est *Address*
  - L'action est **SHARE**
- Conditions :
  - L'identificateur du destinataire principal de la donnée *Address* est égal à l'identificateur du sujet,
- Effect :
  - Permettre l'accès (`Permit`)

Figure 2.11 Règle XPACML

Si un service de type *E-Commerce* demande à partager la donnée *Address*, cette règle est appliquée et retournera un `Permit`.

Si un service de type *E-Commerce* demande à collecter la donnée *Address*, cette règle ne sera pas appliquée car elle cible seulement l'action de partage.

Si un service de type *E-Commerce* demande à partager une donnée mais ne fait pas partir des destinataires, cette règle ne sera pas appliquée car sa condition n'est pas satisfaite.

Ainsi, une règle n'est appliquée que dans les situations spécifiées par sa cible et ses conditions. Pour faire face à plusieurs situations, il est nécessaire de définir plusieurs règles de contrôle d'accès.

2.6.2.3 Politique de contrôle d'accès

A l'instar d'une politique XACML et comme illustré dans la figure 2.10, une politique XPACML regroupe plusieurs règles de contrôle d'accès concernant la même ressource (donnée). Ainsi, si nous avons des règles de contrôle de l'accès la donnée *AddressCity* du type *Address*, et d'autres qui concernent la donnée *Localisation*, il sera plus judicieux de les séparer en deux ensembles distincts. Ainsi, quand une demande de lecture de *AddressCity* parviendra à notre système de contrôle d'accès, seules les règles concernées seront utilisées.

Comme une règle XPACML, une politique XPACML possède une cible (*target*) qui restreint son champ d'application à un ensemble limité de requêtes qui satisfont des conditions bien particulières.

Considérons une politique que nous appelons *Opérations* sur *Address*. Elle est appliquée dans le cas où une requête demande un accès à *Address* et contient les règles suivantes :

1. Un service de type *E-Commerce* peut lire et collecter toutes les données *Address* dont il est destinataire principal, mais il ne peut ni lire, ni collecter les instances *Address* destinées aux autres types de services
2. Le service *E-Government* peut lire toutes les données

Dans cet exemple nous constatons d'abord que le champ d'application de cette politique est limité aux actions de lecture et de collecte sur la ressource *Address*, et aux sujets E-Commerce, *E-Governmennt*. Ainsi nous pouvons spécifier la cible de la politique comme suit :

- Le type du sujet est *E-commerce*, ou *E-Government*,
- Le nom de la ressource est *Address*,
- Le nom de l'action est lecture ou collecte.

Une fois qu'une politique XPACML est appliquée à une requête donnée, toutes les règles contenues dans la politique sont appliquées. Une sélection plus fine est alors obtenue.
Puisque plusieurs règles peuvent être contenues dans une politique, et comme à chaque règle est associée une décision, nous pouvons avoir plusieurs règles qui s'appliquent à un contexte de requête particulier. Par conséquent plusieurs décisions peuvent être prises. La façon la plus simple de faire face à de telles situations est de spécifier au niveau de chaque politique la manière de combiner les différentes règles et leurs décisions. Mises à part les extensions apportées aux éléments des règles XPACML (cible, conditions et décisions), nous apportons des extensions aux quatre comportements standards définis par XACML pour les algorithmes de combinaison, avec le comportement suivant concernant les opérations de filtrage :

- Parmi les règles définissant des opérations sur la même ressource (donnée), à un même niveau de granularité, la règle ayant l'opération la plus stricte est prioritaire.

Ainsi, nous définissons dans le tableau 2.4 l'ordonnancement des opérations de filtrage suivant des niveaux de rigueur de chaque type de sensibilité numérotés de 1 à 4:

Sensitivity Class	**Strictness**	**Sensitivity Type(s)**
Public	1	*UNMODIFIED*
Perturbation	2	*QUANTIZATION, PERTURBATION, ABSTRACTION_LEVEL*
Reduction	3	*AGGREGATE, EXISTENTIAL*
Confidential	4	*BLOCKING*

Tableau 2.4 Regroupement des types de sensibilité en des classes de sensibilité selon le niveau de rigueur

Aussi, afin de prendre en considération les trois axes : consentement, notification, et modification, nous définissons pour une politique donnée, des obligations à faire appliquer par le SP. Ces obligations forment une partie des actions qui doivent être effectuées par le SP à l'égard de l'utilisateur :

- `NOTIFICATION`: spécifie si le SP a l'obligation de notifier l'utilisateur destraitements à appliquer sur la donnée (autres que ceux déjà conclus dans la politique exprimée par l'élément (`POLICY`)).
- `CONSENT`: spécifie si le SP a l'obligation de demander le consentement de l'utilisateur pour toute action accomplie ultérieurement sur la donnée en question.
- `MODIFICATION`: spécifie si le SP a l'obligation d'autoriser l'utilisateur à faire des modifications ultérieures sur la donnée (ressource) en question. Ces modifications peuvent prendre les valeurs suivantes : `Update` (pour la mise à jour de la donnée), `Forget` (pour la suppression de la donnée).

Le tableau 2.5 récapitule le lien entre une partie des éléments utilisés pour l'expression des politiques de protection de données dans XPACML et les axes de protection de données définis dans la section 1.3.1 du chapitre 01, traitant ainsi la problématique de transparence définie dans la section 2.1.

Eléments		Axes Réglementaires: Information	Consentement	Modification	Justification	Conservation	Transmission
Subject		*					
SERVICE-TYPE		*			*		
SERVICE-TYPE_DESCENDANTS		*			*		
Resource		*					
DATA-TYPE		*			*		
DATA-TYPE_DESCENDANTS		*			*		
POLICY	PURPOSE	*			*		
	RECIPIENTS	*					*
	RETENTION	*				*	
Action		*			*		
Obligation	NOTIFICATION	*					
	CONSENT		*				
	MODIFICATION			*		*	

Tableau 2.5 Description des liens entre les éléments de politique XPACML et les axes de protection des données

2.6.2.4 Ensemble de politiques

Pour structurer les politiques et règles XPACML nous avons repris l'élément *PolicySet* du langage XACML défini dans la section 2.4.1.1.4.
Aussi, et du fait que dans un même ensemble de politiques, plusieurs politiques peuvent s'appliquer et générer des réponses différentes, les algorithmes de combinaisons des règles XPACML sont encore utilisés par les ensembles de politiques, mais cette fois-ci nous parlons de combinaison de politiques (*Policy Combining Algorithms*).

Enfin, nous pouvons associer les mêmes obligations définies pour les politiques XPACML, afin d'être exécutées en plus de la décision d'un *PolicySet*.

Concernant les réponses du système, en plus des réponses définies dans XACML (cf section 2.4), la réponse de notre système peut également prendre les valeurs définies pour l'élément `Effect` (`Offer_Policy`) et (`Offer_Data`) définis dans la section 2.6.2.

Remarque
Il faut noter que les règles, politiques, et ensemble de politiques que nous définissons au sein du langage XPACML portent sur des données atomiques. De ce fait, nous traitons

l'accès aux données une à une en nous focalisant sur l'aspect usage (élément `POLICY`) et les conditions qui peuvent y être associées, traitant ainsi la problématique du contrôle définie dans la section 2.1.

En effet, les politiques des SPs demandent souvent des combinaisons de données. Les opérations de filtrage que subissent ces dernières seront spécifiées dans la section 5.6 du chapitre 03. Elles seront reprises au chapitre 05 dédié à la négociation des données possible à ce niveau là.

2.6.3 Architecture de contrôle d'accès XPACML

L'architecture que nous proposons permet d'interfacer les échanges entre les utilisateurs, les SPs et d'autres entités intermédiaires comme les autorités législatives. Elle suit l'idée d'un « courtier de vie privée ». Une hypothèse basique de son approche architecturale est que chaque SP doit s'interfacer avec notre architecture qui constitue un proxy entre lui et l'utilisateur. La gestion des données personnelles au sein de l'architecture est gouvernée par un ensemble de règles (de politiques), qui reflètent les besoins législatifs. Ces règles de courtage sont issues techniquement d'un modèle d'information sémantique (cf. section 3.5 du chapitre03), permettant l'expression des données, les éléments reflétant les six axes de protection, et les types de services ayant des interactions avec notre système. Les règles de courtage issues du modèle sémantique sont définies par des politiques XPACML qui sont exécutées au sein de notre architecture.
Nous utilisons le même modèle de langage de politique XPACML pour l'expression des préférences utilisateur en termes de protection des données, comme pour l'expression des politiques du SP.

L'idée principale derrière notre architecture, est que chaque donnée personnelle injectée dans le système est sujette à des politiques qui spécifient quelle partie de la donnée doit être exposée, à quels types de services et sous quelles contraintes d'usage.

Notre architecture forme un domaine à un haut niveau architectural avec les autres composants :

- Les composants de notre architecture, qui fournissent des interfaces adéquates à l'utilisateur, construisent des enveloppes de vie privée, …etc.
- Les applications du SP, qui sont les consommateurs des données personnelles,

Notre architecture peut s'imbriquer à une architecture de plus bas niveau de gestion d'identité et de pseudonymisation. Elle peut effectuer l'orchestration des mécanismes de protection de telles couches comme la pseudonymisation, l'anonymisation, et les solutions de confidentialité de trafic afin d'atteindre un niveau protection de vie privée plus élevé.

La section suivante fournie une vue des fonctions de l'architecture que nous proposons.

2.6.3.1 Fonctions de l'architecture

2.6.3.1.1 Fonction de définition de politique

Cette fonction est assurée par le modèle sémantique spécifiant pour chaque type de service, les données autorisées et les règles d'accès appliquées. Elle aide également l'utilisateur à définir/modifier ses préférences de vie privée à travers des composants (drop/down listes).

2.6.3.1.2 Fonction de décision

Cette fonction génère une autorisation (ou un refus) d'accès suite à une demande d'accès à une donnée personnelle, en vérifiant la compatibilité entre les politiques de traitement déclarées par le SP, les politiques XPACML exprimant les besoins législatifs et les préférences de l'utilisateur définies pour la donnée en question.

2.6.3.1.3 Fonction négociation

Cette fonction génère des contre propositions en cas de conflits entre les préférences de l'utilisateur et la politique du SP. Les contrepropositions sont établies en appliquant des opération de filtrage sur la donnée requise , ou en proposant de nouveaux usages plus strictes en utilisant le protocole de négociation de politique qui sera abordé dans le chapitre 04. On suppose que le SP est équipé également d'une fonction de négociation lui permettant d'établir des contre propositions à l'utilisateur.

2.6.3.1.4 Fonction de notification

Cette fonction est assurée par le composant PNCM qui sera présenté dans la section 2.6.3.2. Elle a pour rôle de notifier l'utilisateur des termes de la politique de traitement appliquée sur la donnée. Elle permet également d'intercepter toute notification de traitement ultérieur appliqué par le SP sur la donnée.

2.6.3.1.5 Fonction de consentement

Cette fonction requiert de l'utilisateur un consentement explicite à travers les fenêtres suivantes :

- Une fenêtre de consentement simplifié
- Une fenêtre de consentement avancé, qui contient les termes de politiques avec trois niveaux de détail (« overview », « medium », « condensed »). Cette décomposition de la politique appliquée en trois niveaux permet aux utilisateurs avertis d'avoir une version simplifiée et une autre plus détaillée de la politique de protection appliquée par le système. Elle permet également d'intercepter toute demande de consentement envoyée par le SP concernant des traitements non conclus dans la politique initiale.

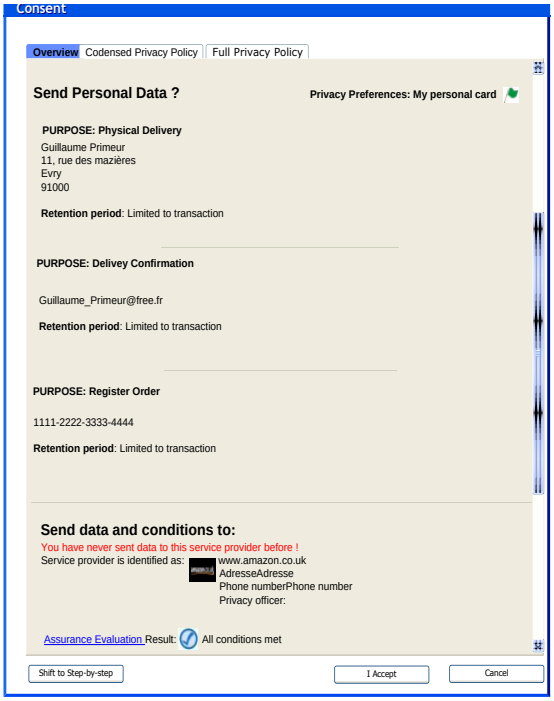


Figure 2.12 Interface de consentement utilisateur

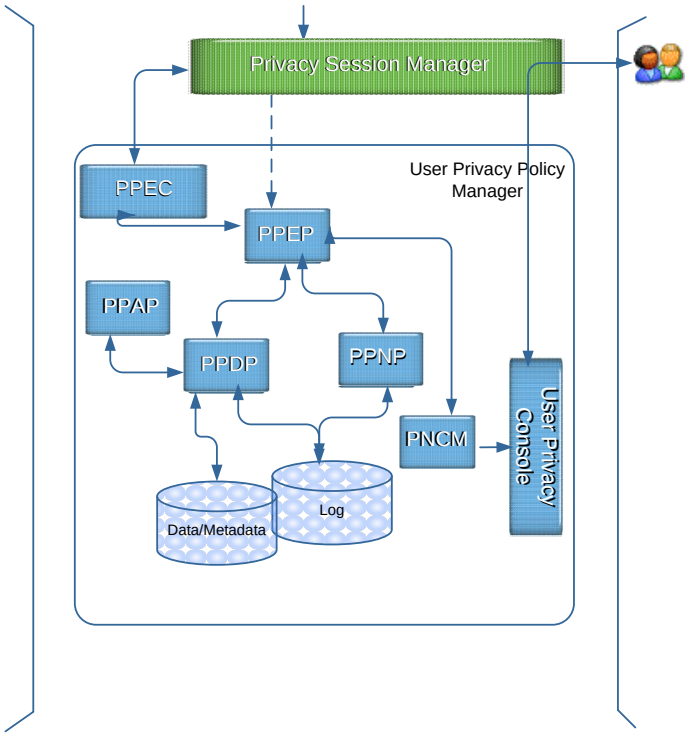
2.6.3.1.6 Fonction de log

Cette fonction journalise l'historique des transactions passées avec chaque type de service. Chaque transaction est identifiée avec un identifiant « Id » lié à l'ensemble des termes de politiques conclus et appliqués.

2.6.3.2 Composants du modèle d'architecture

Les fonctions présentées dans la section précédente sont assurées par un ensemble de composants issus en partie de l'architecture de contrôle d'accès XACML exposée dans la section 2.4.2. Nous avons repris principalement le PEP, le PDP, le PAP, et le PIP, vu l'intérêt que représentent ces composants pour le processus de contrôle de la révélation des données.
Notre architecture est guidée par la modélisation orientée objet. Nous justifions ce choix par la modularité offerte par ce type de modélisation, permettant ainsi des extensions possibles de l'architecture. On dispose nativement d'une couche de données, dans laquelle sont stockées toutes les données personnelles liées à l'utilisateur et ses préférences ainsi qu'aux normes législatives, une couche de traitements qui constituent les procédures d'actions spécifiques du système utilisateur, et d'une couche interface responsable

d'intercepter les actions et préférences de l'utilisateur. Un schéma de la structure interne de l'architecture avec une vue composants, est fourni dans la figure 2.13.



Notre architecture est batie autour des elements suivants :

- Un système de politiques : il comprend les composants PPAP/PPDP/PPNP/PPEP dont le rôle est expliqué ci-dessous.
- PPEC (Pricay Policy Enveloppe Constructor): il lie les données personnelles de l'utilisateur avec les méta-données de vie privée qui leur sont liées, dans une enveloppe chiffrée et signée qui sera transmise au SP.
- Un gestionnaire de notification et de consentement (PNCM) : il gère toutes les tâches liées à la notification et au consentement de l'utilisateur.

2.6.3.2.1 Le gestionnaire de session de vie privée PSM (Privacy Session Manager)

Ce composant représente l'interface vie privée du système utilisateur avec les agents externes. Il est conçu pour interagir avec l'agent du SP pour récupérer sa politique d'usage des données en XPACML. Dès que cette dernière est obtenue, elle est transférée au composant PPEP puis PPDP pour comparaison avec les règles législatives et/ou les préférences de l'utilisateur.

2.6.3.2.2 Le gestionnaire de politiques de protection des données UPPM (User Privacy Policy Manager)

UPPM est le composant qui gère les interactions de l'utilisateur avec les composants internes de l'architecture. Ceci inclut des interactions comme l'ajustement du niveau de protection des données souhaité (comme montré dans la figure 2.14), l'édition des préférences de l'utilisateur pour chaque type de données, et l'information de l'utilisateur des politiques de protection activés.
Le composant UPPM possède une interface utilisateur via laquelle le niveau de protection et les préférences de l'utilisateur sont éditées.

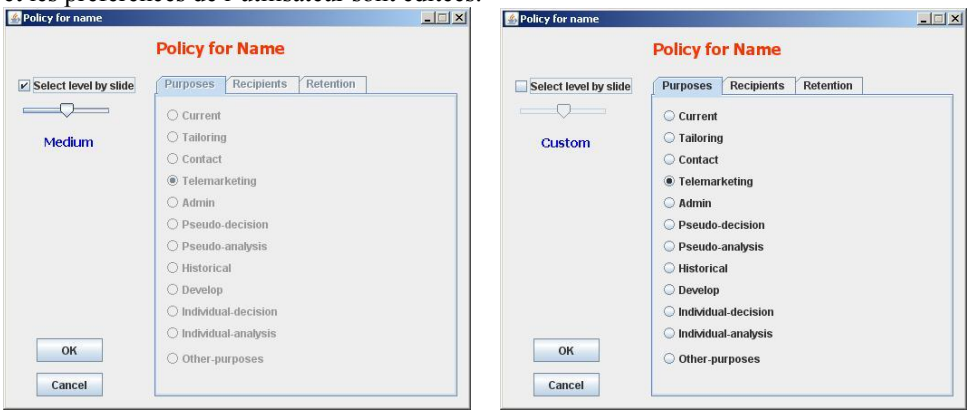


Figure 2.14 Choix d'un niveau de protection prédéfini (a) ou à personnaliser (b)

2.6.3.2.3 Le gestionnaire de notification et de consentement PNCM (Privacy Notification and Consent Manager)

Ce composant prend en charge toutes les tâches liées à la notification ou le consentement de l'utilisateur. Il prend en charge la fonction de consentement décrite dans la section 2.5.1 en utilisant les fenêtres de consentement. Les alertes de violation des termes légaux comme les préférences utilisateur peuvent être également fournies via ce composant.
Pour le processus de négociation présenté dans la section 2.5.1, ce composant est utilisé afin de demander l'avis d'acceptation ou pas de l'utilisateur suite à un conflit non résolu par le processus de négociation établi.

2.6.3.2.4 Le point d'administration des politiques PPAP (Privacy Policy Administration Point)

Ce composant présenté par le modèle sémantique définit les règles de contrôle d'accès et les politiques XPACML.

2.6.3.2.5 Le point d'application des politiques PPEP (Privacy Policy Enforcement Point)

Ce composant reçoit toutes les requêtes d'accès aux données personnelles et les décompose en plusieurs requêtes XPACML, chacune est dédiée à une donnée particulière. Chaque requête élémentaire est envoyée par la suite au composant PPDP pour obtenir une autorisation de délivrance de la donnée requise. En effet, le PPEP établit une requête d'autorisation, spécifiant l'identifiant du SP, le type de service faisant la demande, le type de la donnée requise et d'autres informations requises par le PPDP (ex : conditions) afin d'établir une décision. Une fois les décisions élémentaires reçues, le PPEP les regroupe pour en faire une globale.

2.6.3.2.6 Le point de décision des politiques PPDP (Privacy Policy Decision Point)

Le PPDP est le composant central de notre architecture, chargé de sélectionner les règles/politiques et ensembles de politiques applicables à une requête donnée.

Le PPDP prend en compte plusieurs paramètres afin d'établir une décision :

- Les règles issues du modèle sémantique sont translatées dans un format XPACML. Ces règles expriment les besoin législatifs conventionnellement exprimée/transformée en un ensemble de règles internes. L'évaluation des cibles fournies dans la requête peut être effectuée par rapport à ces règles ou alors par rapport aux préférences de vie privée définies par l'utilisateur (ces préférences sont exprimées également dans un format XPACML).

- Les conditions exprimées sur les données, combinaisons de données ou politiques de traitement (élément POLICY).

2.6.3.2.7 Le point de négociation des politiques PPNP (Privacy Policy Negotiation Point)

Ce composant effectue la négociation des termes de politique du SP avec les termes des préférences de l'utilisateur. Un terme d'une règle ou d'une politique de contrôle d'accès est composé de la paire <DATA-TYPE, un tag de l'élément POLICY>.
En effet, pour toute requête refusée dont le conflit porte sur un ou plusieurs terme(s) de la politique de traitement de données, le PPDP envoie le terme en question au PPNP à travers le PPEP afin de trouver un arrangement avec les préférences éditées par l'utilisateur. Le composant PPNP entame à travers le PPEP un processus de négociation de politiques avec le SP jusqu'à ce que la négociation aboutisse ou un signal de terminaison arrive d'une des deux parties.

La réponse finale de la négociation est ainsi transférée au composant PPEP. Ce dernier applique la décision globale en autorisant ou pas l'accès aux données personnelles.

Le processus de négociation de politiques entre l'utilisateur et le SP est détaillé dans le chapitre 04 de ce mémoire.

2.6.3.2.8 Le composant enveloppe de vie privée PPEC (Privacy Policy Enveloppe Constructor)

Ce composant assemble les données à révéler avec les politiques de protection associées, dans une enveloppe chiffrée et signée qui sera transmise dans sa totalité au SP.

2.6.3.2.9 Les bases de données

Base de données utilisateur
Le répertoire des données personnelles est l'endroit de stockage pour les données de l'utilisateur qui sont à fournir pour un but spécifique.

Base de préférences utilisateur (méta-données)
Cette base contient les préférences édictées par l'utilisateur à l'égard des différents types de données personnelles et des types de services. Elle peut être physiquement stockée avec la base des données personnelles.

Base Log
C'est une base historique permettant de faire une sauvegarde des interactions entre l'utilisateur et le SP. Pour chaque transaction avec un type de service particulier, deux fichiers sont stockés, le premier incluant les données avec autorisation d'accès et le deuxième avec les éléments de données avec un refus d'accès. Ces deux fichiers sont utilisés par la suite dans le processus de négociation.

2.6.3.3 Diagramme de flux XPACML

La figure 2.15 montre les flux de données dans un environnement XPACML. Les étapes de traitements d'une requête XPACML sont :

1. Le PPAP génère des politiques ou des ensembles de politiques XPACML et les rend disponibles au PPDP pour évaluation.
2. Une demande d'accès XPACML parvient au PSM.
3. Le PSM envoie la requête dans son format d'origine au PPEP.

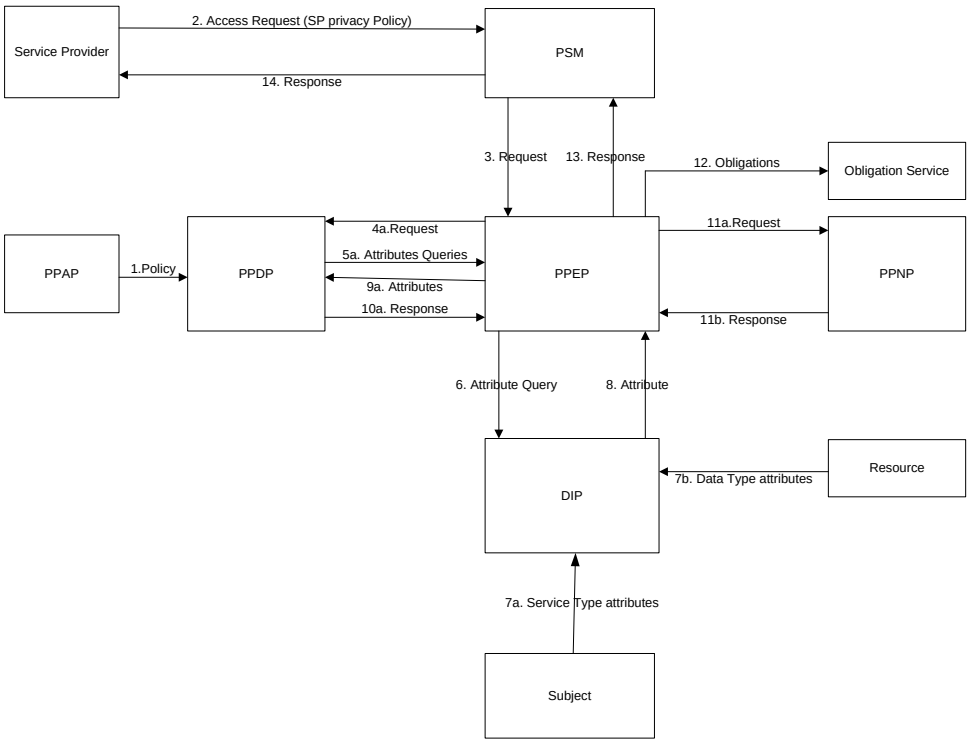


Figure 2.15 Diagramme de flux XPACML

4. a : Le PPEP envoie la requête au PPDP.
5. a : Le PPDP analyse la requête et optionnellement envoie une demande d'attributs supplémentaires non contenus dans la requête. En effet, pour pouvoir évaluer les cibles et les conditions des politiques et des règles de contrôle d'accès, il faut les valeurs de certains attributs. Par exemple, si une requête spécifie juste la ressource à accéder, le PPDP a besoin du type de la ressource.
6. Le PPEP demande au DIP les attributs manquants.
7. a, et b : Le DIP extrait les informations nécessaires sur le sujet et la ressource.
8. Le DIP renvoie les valeurs des attributs demandés.
9. a : Les valeurs des attributs sont envoyées au PPDP.
10. a : Le PPDP évalue les politiques disponibles par rapport à la requête et génère une réponse au format XPACML. Cette réponse est envoyée au PPEP. Si la réponse est négative et si les conditions spécifiées par l'utilisateur autorisent une négociation de politiques, le PPEP envoie la requête au PPNP (11.a), sinon le PPEP passe à l'étape (12).
11. b : Le PPNP effectue une négociation avec le SP à travers le PPEP (les flux associés n'apparaissent pas sur la figure 2.15). Le détail des flux induits sont fournis dans les chapitres 4 et 5. Dans le cas d'une négociation réussie, les types

d'attributs sont précisés par le PPNP au PPEP. Ce dernier charge les valeurs des attributs correspondant (6, 7.a, 7.b, 8) et passe à l'étape (12).

12. Le PPEP vérifie les obligations et les joint à la réponse (12).
13. Le PPEP envoie sa réponse (13) au PSM.
14. Le PSM transfère la réponse à l'entité qui a demandé l'accès (14).

2.7 Génération des politiques XPACML

Nous décrivons dans cette section les aspects implémentation de certains aspects du langage XPACML. On se focalise principalement sur la génération de politiques XPACML en définissant un schéma de politiques XPACML, et la validation des fichiers de politiques générés.
Ensuite, nous présentons l'algorithme et les méthodes de comparaison de politiques associés au PPDP. La phase de négociation de politiques de protection est reportée au chapitre 04.

2.7.1 Définition du schéma de politique

Le langage XPACML est basé sur le langage XML (eXtensible Markup Language). Ce langage assez puissant (voir la description dans section 3.4 du chapitre 03), nous permet de définir nos propres tags pour stocker des informations suivant un schéma ordonné. Il peut être utilisé par différents langages de programmation (java dans notre cas) afin de traiter le contenu des fichiers générés. Nous nous focalisons dans cette partie du mémoire sur la spécification des espaces de nommages (*namespaces*) définis par XML et nécessaires pour notre schéma de politique. On montrera par la suite, comment on peut limiter le vocabulaire qu'un tag donné peut prendre. Nous montrons enfin, comment nous établissons la liaison entre un fichier XML et son schéma XML de base dans la partie conception de politique XPACML.

2.7.1.1 Déclaration des espaces de nommage (*namespaces*)

En référence à [Nam], un espace de nommage XML est identifié par une référence URI, et peut contenir plusieurs éléments [Uni]. L'utilisation de la déclaration de l'espace de nommage supprime tout conflit possible entre ces éléments. Nous utilisons dans notre schéma de politiques trois *namespaces* principaux :

- `xmlns:xpacml="urn:xpacml:policy":`

Comme notre modèle de langage de politique est une adaptation du langage XACML, nous définissons notre propre *namespace* (`xmlns:xpacml`) associé à XPACML. Ce namespace contient des éléments liés principalement à la politique à appliquer par défaut, et à la cible associée.

- `xmlns:xsd="http://www.w3.org/2001/XMLSchema":`

Dans le schéma de définition de politiques XPACML, il y a éventuellement des éléments ayant un préfixe `xsd`. Ce dernier est associé à l'espace du nommage du schéma XML via la déclaration suivante :
(`xmlns:xsd="http://www.w3.org/2001/XMLSchema"`). Le préfixe (`xsd`) est utilisé pour dénoter les espaces de nommage du schéma XML. Le même préfixe et la même association apparaissent également sur les noms des types simples, e.g : `xsd :string`. A titre d'exemple, on utilise ce préfixe pour déclarer l'élément PolicySet. (`xsd :element name="PolicySet".`

Le but de l'association est d'identifier les éléments et les types simples comme appartenant au vocabulaire du schéma de langage XML.

- `xmlns:p3p="http://www.w3.org/2002/01/P3Pv1":`

Cet espace de nommage est utilisé pour introduire les tags de protection <purpose>, <recipient>, et <retention> de la plateforme P3P dans notre schéma de politique à travers le schéma P3P existant dans l'url suivante :
`http://www.w3.org/2002/01/P3Pv1`.

Par exemple, pour déclarer l'utilisation du tag `RECIPIENT` en accord avec les spécifications P3P, on utilise l'expression suivante:

```
<xsd:elementref="p3p:RECIPIENT"minOccurs="0"maxOccurs="unbou
nded"/>
```

2.7.1.2 Spécification des restrictions sur le vocabulaire

Il y a des tags XML ayant des restrictions sur les valeurs. Le vocabulaire de restriction associé doit être présent dans le schéma de politique.
Par conséquent, on utilise l'élément « `restriction` » qui définit des restrictions sur la définition d'un type simple (`simpleType`), le contenu qu'il soit simple ou complexe (`simpleContent`) ou (`complexContent`) du schéma de définition ; et éventuellement sur des éléments ayant un préfixe (`xsd`) . En résumé, on peut spécifier des contraintes sur tous les tags ayant des contraintes sur leurs valeurs.

Les types de l'élément `EFFECT` par exemple doivent être déclarés pour prendre les valeurs suivantes : `Permit, Deny, Offer_Policy, et Offer_Data`.

La figure 2.16 illustre comment nous spécifions ce vocabulaire avec le schéma de politique.

```
<xsd: element name="Rule " type=" xpacml :RuleType "/>
<xsd:complexType name="RuleType">
<xsd: sequence>
<xsd: element ref=" xpacml :Description " minOccurs="0"/>
<xsd: element ref=" xpacml :Target " minOccurs="0"/>
<xsd: element ref =" xpacml :Condition " minOccurs="0"/>
</ xsd: sequence>
```

```
 <xsd : attribute name="RuleId " type=" xsd : string "use=" required "/>
<xsd : attribute name=" Effect " type=" xpacml :EffectType " use=" required
"/>
</ xsd:complexType>
<!-- -->
<xsd: simpleType name="EffectType ">
<xsd : restriction base=" xsd : string ">
<xsd: enumeration value="Permit "/>
<xsd: enumeration value="Deny"/>
<xsd: enumeration value="Offer_Policy "/>
<xsd: enumeration value="Offer_Data "/>

</ xsd: restriction>
</ xsd: simpleType>
```

Figure 2.16 Définition des restrictions sur les valeurs de l'élément Effect

L'élément `Action` doit contenir également des restrictions dans sa définition dans le schéma de politique

```
<xsd: element name="Actions " type=" xpacml :ActionsType "/>
 <xsd:complexType name="ActionsType ">
 <xsd: sequence>
 <xsd: choice maxOccurs="3">
 <xsd: element minOccurs="0" name=" read "/>
 <xsd: element minOccurs="0" name=" collect "/>
 <xsd: element minOccurs="0" name=" share "/>
 </ xsd: choice>
 </ xsd: sequence>
 </ xsd:complexType>
```

Figure 2.17 Définition des restrictions sur les valeurs de l'élément Action

Nous utilisons également des restrictions pour définir le vocabulaire de l'élément `POLICY` qui est basé sur les tags P3P. La figure 2.18 montre l'introduction de ces éléments dans notre schéma de politique

```
<xsd: element name="Resource "
 type=" xpacml :ResourceType "/>
 <xsd:complexType name="ResourceType ">
 <xsd: sequence>
 <xsd: element ref=" xpacml :ResourceMatch " maxOccurs="unbounded"/>
<!-- This is the basic p3p poloicy related to a resource -->
 <xsd: element ref="p3p:PURPOSE" minOccurs="0" maxOccurs="unbounded"/>
 <xsd: element ref="p3p:RECIPIENT" minOccurs="0" maxOccurs="unbounded"/>
 <xsd: element ref="p3p:RETENTION" minOccurs="0" maxOccurs="1"/>
 </ xsd: sequence>
 <!-- This is the basic p3p poloicy related to a resource -->
 <xsd : attribute name="ResourceId " type=" xsd : string " use=" required
"/>
 </ xsd:complexType>
 <!-- -->
<xsd: element name='PURPOSE'>
```

```
 <xsd:complexType>
 <xsd: sequence>
 <xsd : choice maxOccurs="unbounded">
 <xsd: element name=' current ' type=" p3p:purpose-value "/>
 <xsd: element name=' admin ' type=" p3p:purpose-value "/>
 <xsd: element name=' develop ' type=" p3p:purpose-value "/>
 <xsd: element name=' tailoring ' type=" p3p:purpose-value "/>
 <xsd: element name=' pseudo-analysis ' type=" p3p:purpose-value "/>
 <xsd: element name=' pseudo-decision ' type=" p3p:purpose-value "/>
 <xsd: element name=' individual-analysis ' type=" p3p:purpose-value "/>
 <xsd: element name=' individual-decision ' type=" p3p:purpose-value "/>
 <xsd: element name=' contac t ' type=" p3p:purpose-value "/>
 <xsd: element name=' historical ' type=" p3p:purpose-value "/>
 <xsd: element name=' telemarketing ' type=" p3p:purpose-value "/>
 <xsd: element name=' other-purposes ' type=" p3p:purpose-value "/>
 <xsd: element name='EXTENSION' type=" p3p:purpose-value "/>
 </ xsd : choice>
 </ xsd: sequence>
 </ xsd:complexType>
 </ xsd: element>
 <!-- -->
 <xsd: element name='RECIPIENT '>
 <xsd:complexType>
 <xsd: sequence>
 <xsd : choice maxOccurs="unbounded">
 <xsd: element name=' ours ' type=" p3p : recipient-value "/>
 <xsd: element name=' same ' type=" p3p : recipient-value "/>
 <xsd: element name=' other-recipient ' type=" p3p : recipient-value "/>
 <xsd: element name=' delivery ' type=" p3p : recipient-value "/>
 <xsd: element name=' public ' type=" p3p : recipient-value "/>
 <xsd: element name=' unrelated ' type=" p3p : recipient-value "/>
 <xsd: element name='EXTENSION' type=" p3p : recipient-value "/>
 </ xsd : choice>
 </ xsd: sequence>
 </ xsd:complexType>
 </ xsd: element>
 <!-- -->
 <xsd: element name='RETENTION'>
 <xsd:complexType>
 <xsd: sequence>
 <xsd: element name=' no-retention ' type=" p3p: retention-value "/>
 <xsd: element name=' stated-purpose ' type=" p3p: retention-value"/>
 <xsd: element name=' legal-requirement ' type=" p3p: retention-value"/>
<xsd: element name=' indefenitely ' type=" p3p: retention-value "/>
 <xsd: element name=' business-practices ' type=" p3p: retention-value "/>
 <xsd: el mentname='EXTENSION' type=" p3p: rretention-value "/>
 </ xsd: sequence>
 </ xsd:complexType>
 </ xsd: element>
```

Figure 2.18 Introduction des principaux éléments P3P dans le schéma de politique XPACML

2.7.1.3 Conception d'une politique XPACML

Comme nous l'avons vu, l'expression des politiques XPACML se base sur le langage XML avec l'utilisation des balises spécifiques au contrôle d'accès aux données. Nous avons ainsi utilisé l'API Jaxb pour créer des documents XML (disponibles en plugin pour éclipse, par exemple). Le schéma opérationnel de cette API est présenté dans la figure 2.19

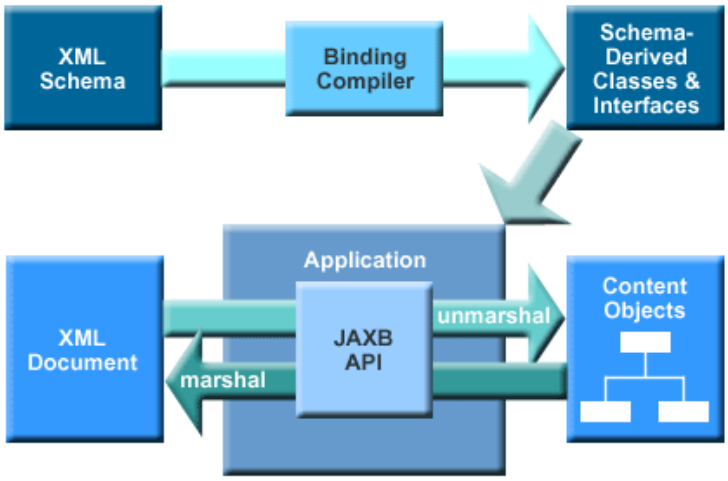


Figure 2.19 Flux de traitements pour la conception de documents XML [Pol05, Con05]

Pour générer une politique XPACML, nous créons premièrement un schéma de politique XML, dans lequel on crée des éléments et des types, ex : l'élément Resource servira comme balise pour insérer la valeur de la donnée…etc.
Ensuite, nous utilisons un compilateur d'attachement (XJC (*XML Java Compiler*) dans notre cas) pour créer des classes dérivées des éléments du schéma XML. On récupère ces classes Java pour chacun des types et pour les éléments qui n'ont pas de type défini (comme *PolicySet*). C'est dans ces classes que l'on va faire les traitements des balises (remplissage des valeurs et affichage du document XML).

On doit également gérer la création des différents éléments du schéma XML. Le compilateur crée donc une fonction «*Object Factory*» s'occupant de cette tâche. Ensuite, c'est aux programmeurs de remplir convenablement (pour un traitement des données adéquat à leur modèle) ces différentes classes Java.

L'API JAXB récupère ces classes dérivées et leur fait subir un traitement différent suivant que l'on veut créer un document XML ou récupérer des informations à partir d'un tel document. L'API JAXB propose donc deux fonctions : « *marshalling* » permet à partir du

schéma XML et des classes dérivées, de créer un document XML et la fonction « *unmarshalling* » sert à récupérer un document XML et à extraire les informations présentes entre les différentes balises.

L'application que nous avons développée permet ainsi de créer un schéma de politique XPACML, reprenant les éléments XACML qui nous intéressent, en intégrant les éléments de politiques de protection que nous avons définis dans la section 2.6. Grâce à cette application, on obtient une interface graphique simple d'utilisation permettant de créer une politique XPACML.

2.7.2 Exemples de politiques XPACML

Comme mentionné dans les sections précédentes, notre schéma proposé permet l'expression de différents types de politiques :

- Préférences de l'utilisateur,
- Politiques de traitement des données du SP.

La figure 2.20 illustre l'exemple d'une politique XPACML correspondant à une règle législative concernant un service de type e-commerce demandant le numéro de carte bancaire de l'utilisateur.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xpacml:PolicySet xmlns:xpacml="urn:xpacml:policy"
xpacml:p3p="http://www.w3.org/2002/01/P3Pv1"
xpacml:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance">
  <xpacml:Description>XPACML example SP's privacy
policy</xpacml:Description>
      <xpacml:Policy PolicyId="eCommerce">
       <xpacml:Target>
      <xpacml:Subject>
      <xpacml:Service_Type ApplyToDescendent="No">
   <xpacml:eCommerce />
       </xpacml:Service_Type>
   </xpacml:Subject>
       </xpacml:Target>
<xpacml:Rule Effect="Permit" Category_Id="CCNumber_Value"
ApplyToDescendant="Yes">
  <xpacml:Description>One rule describes a policy for a specific
category</xpacml:Description>
      <xpacml:Target>
      <xpacml:Resources>
   <xpacml:Resource ResourceId="CCNumber_Value" DataCompostion="Composed">
       <p3p:PURPOSE xmlns:p3p="http://www.w3.org/2002/01/P3Pv1">
    <p3p:current />
    <p3p:admin />
    <p3p:develop />
         <p3p:historical />
  </p3p:PURPOSE>
```

```
     <p3p:RECIPIENT xmlns:p3p="http://www.w3.org/2002/01/P3Pv1">
   <p3p:ours />
  </p3p:RECIPIENT>
      <p3p:RETENTION xmlns:p3p="http://www.w3.org/2002/01/P3Pv1">
   <p3p:no-retention />
  </p3p:RETENTION>
  </xpacml:Resource>
  </xpacml:Resources>
      <xpacml:Actions>
   <xpacml:read />
   <xpacml:collect />
   <xpacml:share />
       </xpacml:Actions>
        </xpacml:Target>
        </xpacml:Rule>
        </xpacml:Policy>
        </xpacml:PolicySet>
```

Figure 2.20 Exemple d'une politique législative XPACML

La figure 2.21 illustre une politique exprimant les préférences de l'utilisateur, pour la donnée « Address » qui est requise par un service financier.

```
<?xml version=" 1 . 0 " encoding="UTF□8"?>
 <xpacml :PolicySet
 xmlns:p3p=" http : //www.w3c.org/2002/01/P3Pv1"
 xmlns:xml=" http : // www.w3c.org/XML/1998/namespace"
 xmlns:xpacml=" urn: xpacml :policy "
 xmlns : xsi=" ht tp : //www.w3c.org/2001/XMLSchema□instance "
 xsi : schemaLocation=" urn: xpa cml :policy XPACML.xsd">
 <xpa cml :Description>XPACML example SP's privacy policy </xpacml
:Description>
 <xpacml :Policy PolicyId="FinSerpol">
 <xpacml :Description>this is the privacy preferences for the Financial
Services </
xpacml :Description>
 <xpacml :Target>
 <xpacml :Subject>
 <xpacml :Service Type ApplyToDescendant="No">
 <xpa cml :FinancialServices/>
 </xpacml :Service Type>
 </xpacml :Subject>
 </xpacml :Target>
 <xpacml :Rule Effect="Permit " CategoryId="Address Category "
ApplyToDescendant="Yes">
 <xpacml :Description>One rule describes a policy for a specific category
</
xpacml :Description>
 <xpacml :Target>
 <xpacml :Resources>
 <xpacml :Resource ResourceId="Address Category "
DataComposition="Composed">
 <p3p:PURPOSE>
 <p3p: current/>
 <p3p:admin/>
 <p3p:develop/>
```

```
 </p3p:PURPOSE>
 <p3p:RECIPIENT>
 <p3p:ours/>
 <p3p:same/>
 <p3p : delivery/>
 </p3p:RECIPIENT>
 <p3p:RETENTION>
 <p3p: stated purpose/>
 </p3p:RETENTION>
</xpacml :Resource>
 </xpacml :Resources>
 <xpacml :Actions>
 <xpacml : read/>
 < xpacml : collect />
 <xpacml : shar e/>
 </xpacml :Actions>
 </xpacml :Target>
 </xpacml :Rule>
 <xpacml :Rule Ef f e c t="Deny" CategoryId="Address Category "
ApplyToDescendant="Yes">
 <xpacml :Description>One rule decribes a policy for a specific category </
xpacml :Description>
 <xpacml :Target>
 <xpacml :Resources>
 <xpacml :Resource ResourceId="Addres s Category "
PolNegoPermission="Allowed">
 <p3p:PURPOSE>
 <p3p: contact/>
 <p3p: telemarketing/>
 <p3p:other purpose/>
 </p3p:PURPOSE>
 <p3p:RECIPIENT>
 <p3p:public/>
 </p3p:RECIPIENT>
 <p3p:RETENTION>
 <p 3 p : indefinitely />
 </p3p:RETENTION>
 </xpacml :Resource>
 </xpacml :Resources>
 <xpacml :Actions/>
 </xpacml :Target>
 </xpacml :Rule>
 </xpacml :Policy>
 </xpacml :PolicySet>
```

Figure 2.21 Exemple de préférences utilisateur

2.7.3 Vérification de conformité des fichiers de politiques

Il est important de valider chaque fichier de politique XPACML avec le schéma de politique que nous avons défini. Ainsi, nous avons conçu une classe PolicyValidator (voir annexe A) permettant de vérifier la conformité des fichiers de politiques générés avec le schéma de politique défini. Cette fonction retourne un résultat booléen exprimant la conformité de la politique avec le schéma de base.

Ainsi, chaque politique XPACML ayant comme source le modèle sémantique, l'utilisateur ou le SP doit respecter les contraintes de vocabulaire, la multiplicité des sous éléments, les différents types définis, etc.

2.7.4 Comparaison des politiques (implémentation du PPDP)

2.7.4.1 Méthodes

Nous implémentons le composant PPDP sous forme d'une classae Java, dont nous avons défini un constructeur ayant quatres entrées (figure 2.22): la politique du SP, les préférences de l'utilisatur, un fichier des termes communs des deux politiques, et un autre pour les différences.

```
public PPDP(Document xpacml policy User , Document xpacml policyServer ,
 Document xpacml policydiff, Document xpacml policyequal)
 {
 try
  {
   //Get the root element o f each Document
  racineUser = xpacml policyUser.getRootElement( );
  racineServer = xpacml policyServer.getRootElement( );
  racinediff = xpacml policydiff.getRootElement( );
  racineEqual = xpacml policyequal.getRootElement ( ) ;

  //
  policyUser = racineUser.getchlid ("Policy", namespace);
  policyServer = racibeServer.getChild ("Policy", namespace);

  policyDiff = racineDiff.getChild ("Policy", namespace);
  policyEqual = racineEqual.getChild ("Policy", namespace);
 }
  catch ( Exception e ) {}
}
```

Figure 2.22 Constructeur du PPDP

2.7.4.2 Algorithme de comparaison

Considérant la comparison pour une simple donnée, nous utilisons à titre illustrative les quatres methodes présenté dans la figure 2.23.

```
PPDP ppdp1 = new PPDP( xpacml policyUser, xpacml policyServer),
 xpacml policydiff, xpacml policyequal) ;
 ppdp . Compare Action ( ) ;
 ppdp . Compare Purpose ( ) ;
 ppdp . Compare Recipient ( ) ;
 ppdp . Compare Retention ( ) ;
```

Figure 2.23 Comparaison simple

La figures 2.24 illustre une partie des résultats de comparaison concernant le fichier des différences.

```
 <xpacml :Target>
  <xpacml :Resources>
   <xpacml :Resource ResourceId=”Res1”>
    <xpacml :ResourceMatch . . . >
      <xpacml :AttributeValue . . . />
        <xpacml :ResourceAttributeDesignator . . . />
    </ xpacml :ResourceMatch>
        <p3p:PURPOSE>
          <p3p:pseudo-analysis>Server</ p3p:pseudo-analysis>
          <p3p:telemarketing>Server</ p3p: telemarketing>
        </p3p:PURPOSE>
        <p3p:RECIPIENT>
         <p3p:delivery>User</p3p:delivery >
        </p3p:RECIPIENT>
        <p3p:RETENTION />
    </ xpacml :Resource>
  </ xpacml :Resources>
        <xpacml :Actions>
         <xpacml :ActionMatch . . . >
         <xpacml :ActionAttributeDesignator . . . />
          <xpacml :AttributeActivationValue>User : share</ xpacml :
AttributeActivationValue >
         </ xpacml :ActionMatch>
        </ xpacml :Act ion>
 </ xpacml :Target>
```

Figure 2.24 Exemple des résultats de différences

2.8 Prime Life Policy Language (PPL)

Une nouvelle famille de langages de politiques basée sur le standard XACML [OAS05] a fait récemment son apparition. Cette famille de langages est basée sur une approche nommée « Credential Based Access Control », dont l'unité de contrôle d'accès de base est le « credential ». Un credential est défini dans cette approche comme une agrégation de données certifiées. Un concept assez proche du concept d'identité proposé dans les systèmes de gestion d'identité [Euro08], à la différence qu'on peut choisir la collection des données à révéler, à l'instar de ce qui a été proposé par idemix (IDEntity MIXer) [CL01, CV02], et UProve [U-P07].

PPL (Prime life Policy Language) [PPL11] est le plus mature des travaux existants dans cette famille de langages. Il fait usage du standard XACML pour réguler l'accès des

utilisateurs aux différents services fournis par le SP. Cette régulation d'accès se fait à base de données certifiées (credentials) fournies par l'utilisateur, en retour de garanties fournies par le SP. PPL a recours au standard SAML 2.0 afin de spécifier l'autorité de certification (l'issuer) des données, tâche qui est également proposée par XACML, mais juste pour un attribut donné, et non une agrégation de données.

2.8.1 Principes du langage PPL

Dans cette section, nous nous attachons à expliciter les éléments de base du langage PPL. La figure 2.25 illustre les extensions apportées au modèle de base du langage XACML.

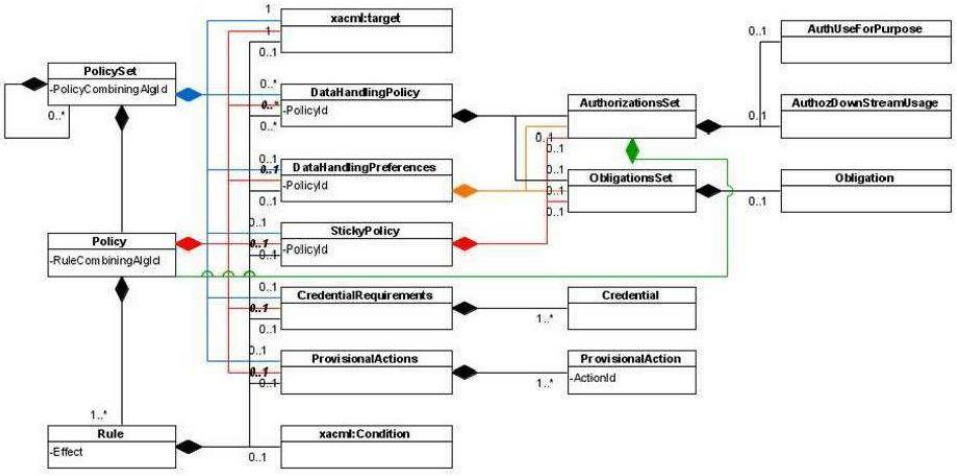


Figure 2.25 Modèle du langage PPL

2.8.1.1 Règles, politiques, et ensembles de politique

PPL maintient la structure globale du langage XACML. Il introduit plusieurs nouveaux éléments, et modifie le schéma de plusieurs éléments déjà existants. Règles, politiques, et ensembles de politiques sont directement issus du langage XACML. Chaque règle possède un élément « effect » avec les deux valeurs « Permit » et « Deny » indiquant les conséquences si les conditions d'une règle sont satisfaites.

Les éléments principaux d'une règle PPL sont :

- « target » : décrit la ressource à protéger au niveau SP, le sujet, et les variables d'environnement pour lesquelles cette règle est applicable,
- « credential requirements » : décrit les credentials à présenter afin de

donner l'accès aux ressources,

- « provisional actions » : décrit les actions (ex : révélation des attributs, ou signature de déclarations) qui doivent être accomplies par l'utilisateur afin d'avoir accès à la ressource requise au niveau du SP,
- « condition » spécifie d'autres restrictions sur l'applicabilité de la règle au-delà de celles spécifiées dans les éléments « target » et « credential »,
- « data handling policies (DHP)» : décrit comment les données requises satisfaisant la règle seront traitées par le SP,
- « data handling preferences DHPref » : décrit les préférences de l'utilisateur pour ses données requises.

Les éléments <xacml : Target> et <xacml : Condition> sont réutilisés dans PPL tels qu'ils sont définis dans XACML. L'élément <CredentialAttributeDesignator> est comme les conditions XACML du type <xacml : Expression>. Il est utilisé pour désigner un attribut depuis dans un credential. Le reste des éléments sont nouveaux. Ils sont décrits dans les sous-section suivantes.

2.8.1.2 Les éléments d'autorisation

Les éléments d'autorisations spécifient les actions que le SP a la possibilité de pratiquer sur les données. Les concepteurs du langage PPL expriment qu'il est impossible d'établir une liste exhaustive des autorisations couvrant les besoins du monde réel. Ainsi, ils ont mis en place une structure assez générique d'autorisations, qui permet l'ajout de nouvelles autorisations spécifiques. La spécification actuelle [PPL11] fournit un vocabulaire d'autorisation très basique couvrant l'usage des données pour certains buts (purposes), et le transfert des données à une tierce partie (downstream).

2.8.1.3 Les besoins de Credential

Chaque règle contient un élément <CredentialRequirements> qui spécifie les données certifiéesà présenterpour satisfaire la règle d'accès.Chaque élément <CredentialRequirements> contient deux éléments : un élément <Credential> pour chaque attribut certifié à présenter, qui est identifié par un élément CredentialId de type URI, et, un élément <Condition> décrivant les conditions que cet attribut doit remplir.

L'élément <Credential> peut contenir des restrictions qui s'appliquent sur l'attribut certifié. Ces restrictions sont exprimées par le biais d'un élément nommé <AttributeMatchAnyOf> permettant la comparaison des attributs contenus dans un credential avec une liste des valeurs candidates au niveau du SP.

2.8.1.4 Les politiques de traitement des données

Les politiques de traitement des données sont exprimées en PPL par le bais de l'élément <DataHandlingPolicy>. Chaque politique contient deux ensembles. Un ensemble d'autorisations que le SP veut avoir sur les données collectées, exprimé par le biais de l'élément <AutorisationSet>. Et un ensemble d'obligations qu'il doit remplir vis-à-vis de l'utilisateur. Ces obligations sont exprimées par le biais de l'élément <ObligationsSet>.

Les concepteurs du langage PPL mettent en place deux types d'autorisations comme mentionné dans la section 2.8.1.2. Les autorisations pour un type de but précis sont exprimées par l'élément <AuthzUseForPurpose>. Pour ceci, les concepteurs mettent en place une première liste des buts de collecte tels que définis par la plateforme P3P [Wor06], et des autorisations liées au transfert des données à des tierces parties. Cette deuxièmecatégorie d'autorisations exprimée par le biais de l'élément <AuthzDownstreamUsage>, contient un attribut booléen indiquant si l'autorisation de l'utilisateur pour un transfert des données est accordée ou pas.

Les obligations que le SP doit respecter sont exprimées par le biais de l'élément <Obligation>. Cet élément contient à son tour trois éléments :

- <TriggersSet> : décrivant les évènements déclenchant la mise en application des obligations,
- <Action> : décrivant les actions à effectuer,
- <Validity> : décrivant la durée de validité de l'obligation en question.

Les concepteurs du langage PPL ont défini un vocabulaire basique pour l'ensemble des éléments laissant la porte ouverte pour de futures extensions.

2.8.1.5 Les préférences de l'utilisateur

Les préférences de l'utilisateur contiennent également un ensemble d'autorisations et un autre dédié aux obligations. Elles sont exprimées par le biais de l'élément <DataHadlingPreferences>.

Les préférences utilisateur peuvent contenir optionnellement une politique spécifique pour le transfert des données à des tierces parties par le biais de l'élément <AuthzDownstreamUsage>.

2.8.1.6 Les politiques collantes

Les accords conclus après une procédure de comparaison automatique entre les politiques du SP et les préférences de l'utilisateur sont exprimés par le biais de l'élément <StickyPolicy>.

La différence principale entre les éléments <StickyPolicy> et <DataHadlingPreferences> se situe au niveau des autorisations et des obligations. La politique collante contient les autorisations et les obligations contenues dans les politiques que le SP doit respecter, alors que les préférences utilisateur peuvent contenir en plus, des autorisations et/ou des obligations qui doivent être appliquées par la tierce partie en cas de transfert.

2.8.1.7 Les actions provisionnelles

Ce sont des actions provisionnelles supplémentaires à celles définies par XACML, que l'utilisateur doit remplir afin de lui garantir l'accès au service requis.

Dans la version actuelle des spécifications PPL, la liste des actions provisionnelles contient la révélation des données de l'utilisateur, la signature d'un consentement, la restriction du nombre de fois qu'un credential peut être utilisé. Le vocabulaire de ces termes a été défini comme un ensemble unique d'identifiants URI.

2.8.2 Architecture de contrôle d'accès PPL

Comme présenté dans la figure 2.26, l'architecture PPL emploie deux modules indépendants, un dédié au contrôle d'accès (XACML Engine), et un autre dédié à la gestion et à l'évaluation des politiques de protection des données (Data Handling Decision Function ou DHDF). Il est important de noter que le DHDF n'implémente pas un système de contrôle d'accès sensible à la protection de la vie privée.

Quand une décision d'accès est requise, l'enveloppe de décision (Decision Wrapper) transfère la requête d'accès au gestionnaire de contrôle d'accès (AC Manager), avec les politiques de traitement (DHP) attachées à la ressource requise. Le gestionnaire de contrôle d'accès transfère la requête d'accès aux moteurs XACML et DHDF, et établit la décision finale en combinant leurs réponses. Le moteur DHDF reçoit la requête avec les politiques de protection associées, alors que le moteur PrimeLife XACML accède au gestionnaire de politiques pour trouver les politiques de contrôle d'accès applicables. Enfin, DHDF et PrimeLife XACML accèdent au composant (Request Context) à travers le composant (Data Reader) afin de trouver les informations (probablement certifiées par le système credential) requises pour faire une évaluation.

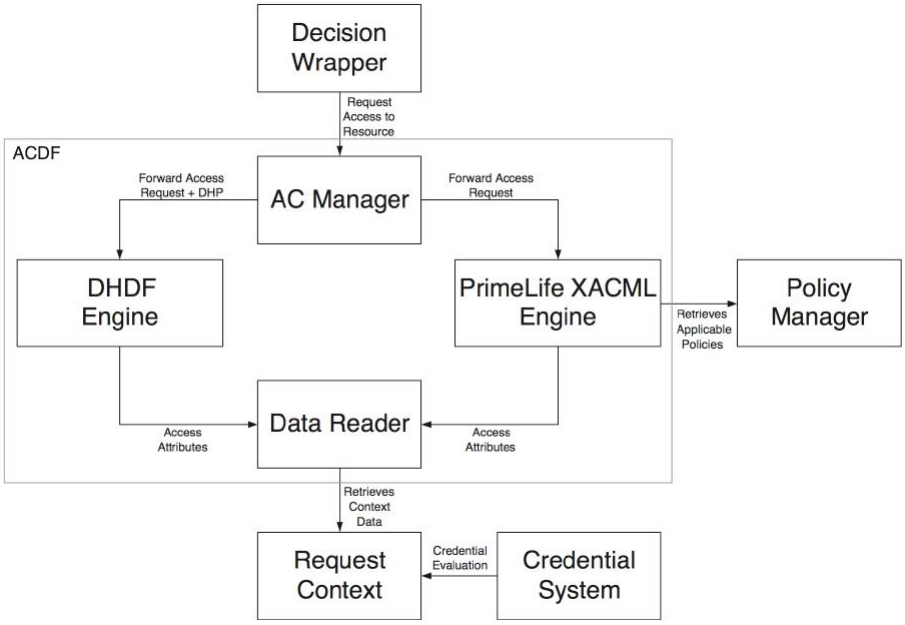


Figure 2.26 Architecture de contrôle d'accès PrimeLife

2.8.3 Diagramme de flux du moteur de contrôle d'accès PrimeLife

La figure 2.27 illustre une représentation détaillée des flux de données du moteur PrimeLife XACML déjà présenté dans la figure 2.26. La structure présente une grande correspondance avec l'architecture du standard XACML (voir figure 2.8) et ses composants traditionnels. Le flux de données est comme suit.
Premièrement, la requête d'accès est transférée par le composant AC Manager au PEP (étape 1). Le PEP crée une requête XACML et la transfère au gestionnaire de contexte (Context Handler) (étape 2). Comme dans XACML, ce dernier invoque le PDP (étape 3), qui est responsable de l'évaluation des politiques XACML. Par le biais d'une extension lui permettant d'accéder au gestionnaire des politiques, le PrimeLife PDP accède aux politiques XACML appliquées à la requête (étape 4). Pour permettre ce transfert, le gestionnaire de politiques situé au niveau du SP fournit une interface externe compatible avec ce composant standard nommé PAP (Policy Administration point). Après avoir récupéré les politiques applicables, le PrimeLife PDP accède au gestionnaire de contexte pour retrouver les données requises pour l'évaluation de la requête (étape 5). Tout comme dans XACML, si des données requises à la prise de décision sont manquantes, le gestionnaire de contexte peut accéder à différentes extensions du PIP. Ces extensions comportent un PrimeLife PIP qui prend en charge la communication avec le lecteur de

données (DataReader) afin d'accéder au contexte de la requête (étapes 6 à 9). Après la réception des données, PrimeLife PDP (étape 10) évalue finalement les politiques applicables et prend une décision. La décision finale est envoyée au gestionnaire de contexte (étape 11), et est ainsi transférée au PEP (étape 12). Enfin, la décision finale est retournée à l'AC manager (étape 13), qui la combine avec la réponse récupérée de l'évaluation des politiques effectuée par le DHDF.

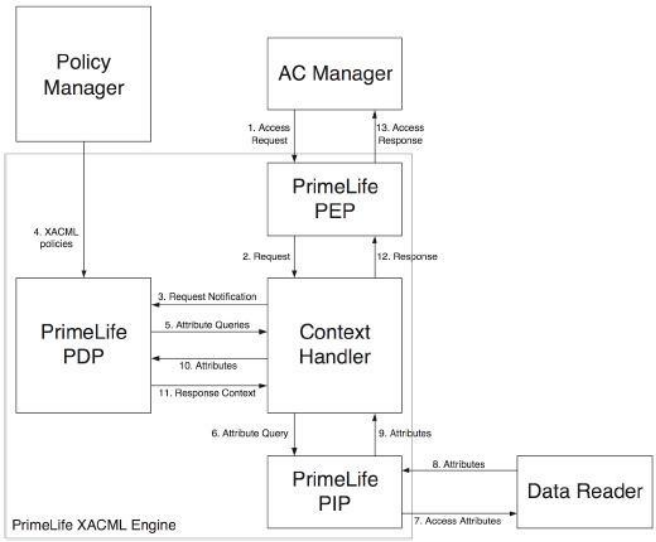


Figure 2.27 Flux de données du moteur PrimeLife XACML

2.8.4 Comparaison entre PPL et XPACML

Comme présenté dans les sections 2.6 et 2.8, PPL comme XPACML font usage du standard XACML pour permettre la protection des ressources.

Il est important de souligner dans un premier temps la différence entre les concepts de base de chacun des deux langages. Le langage PPL suit un schéma traditionnel de préservation des ressources au niveau du SP. La protection de ses ressources est effectuée par le biais de politiques de contrôle d'accès, tout en mettant en place davantage d'éléments liés à la protection des données collectées en vue d'une autorisation d'accès. En revanche, la ressource principale à protéger dans notre langage XPACML est la donnée elle-même avec une approche centrée utilisateur. Ceci est rendu possible par le biais de la mise en place d'un système de contrôle d'accès sensible au respect de la vie privée, permettant de gérer les politiques de traitement des données, et celle d'accès aux données en premier lieu.

Toute politique PPL [PPL 11] contient deux fragments principaux. Un fragment lié aux autorisations à appliquer et un autre fragment lié aux obligations exigées du SP.

Pour la partie autorisations, une analyse approfondie des éléments de politiques PPL soulève les premières différences avec les éléments de politiques XPACML. Comme mentionné dans la section 2.8.1.4, la partie autorisations des politiques de traitement des données (DHP) dans le langage PPL se base sur le but de la collecte comme élément principal de la gestion d'autorisations. Les auteurs mentionnent dans leur spécification [PPL11], que le reste des éléments à intégrer sont à tirer des besoins généraux d'un modèle d'interaction à base de credential, et le modèle business de l'entreprise et ses besoins de contrôle d'accès. Alors que, de part notre premier objectif de mise en application des règles législatives (par le biais des axes de protection que nous avons définis dans la section 1.3.1), nous avons proposé dans la section 2.6.2 les éléments remplissant toutes les contraintes législatives liées au traitement des données personnelles. Aussi, dans la section 3.5.1, nous proposons un modèle sémantique détaillé des données permettant d'éditer plus finement les politiques associées, alors que le démonstrateur actuel du langage PPL ne propose pas de modèle précis des données.

De ces éléments, il découle une différence importante au niveau du détail et de l'expressivité des politiques de protection, qui sont plus importants dans XPACML que dans PPL.

D'autre part, PPL ne réutilise les classes de base du modèle XAMCL qu pour le contrôle d'accès (section 2.8.1 et 2.8.2). Comme nous pouvons le remarquer de nouvelles classes (section 2.8.1) sont définies les politiques de protection. Ce qui augmente considérablement la tâche d'administration des politiques, contrairement à notre modèle où toutes les extensions effectuées se font sur les classes déjà existantes.

Pour la partie obligations, les auteurs du langage PPL mettent en place un mécanisme de spécification des obligations, ce qui permet une application assez simple et intuitive. Ce formalisme constitue un support important facilitant au SP l'application des engagements pris vis-à-vis de l'utilisateur. En XPACML, nous n'avons pas poussé aussi loin nos recherches, et nous nous somme contentés de mettre en place trois classes destinées à la gestion des obligations (figure 2.15)

En termes d'architecture, et comme illustré dans les sections 2.6 et 2.8 respectivement, les extensions apportées au modèle de contrôle d'accès XACML sont nettement différentes dans les deux langages. Les concepteurs du langage PPL proposent de garder l'architecture XACML de base et de rajouter des composantes séparées pour la gestion des politiques de traitement des données (sections 2.8.2 et 2.8.3). Ainsi, la gestion des deux types de

politiques (politiques de contrôle d'accès et politiques de protection des données) se fait séparément.Dans notre modèle d'architecture (figures 2.15 et 3.12) le mécanisme de contrôle d'accès est lui-même basé sur des politiques de protection des données issues du modèle sémantique que nous définissons (section 3.5). Notre modèle d'architecture est simple, et cohérent. Par les différentes extensions que nous avons apportées (section 2.6.2), nous avons fais un pas vers l'adaptation du modèle de contrôle d'accès classique pour le rendre sensible à la protection de la sphère privée. Ces extensions que nous apportons, comme celles proposées par les auteurs du langage PPL réutilisent au maximum les classes XACML déjà existantes.

PPL ne fournit pas à l'utilisateur un éditeur de politiques lui permettant de spécifier ses préférences. Tel que précisé par les concepteurs du langage, les politiques sont actuellement écrites manuellement.Dans XPACML, notre prototype est doté de plusieurs interfaces (section 2.6.3) implémentées afin d'accompagner l'utilisateur dans la définition de ses préférences simples pour une gestion d'accès, comme celles plus évoluées (figures
4.6 et 4.7) permettant la négociation des politiques avec le SP. Ces interfaces servent à
cacher la complexité du modèle sémantique (section 3.5.1) en arrière plan permettant la génération automatique des politiques de protection.

2.9 Conclusion

Les langages de politiques comme P3P proposent des solutions aux problèmes de formulation des pratiques d'usage faites des données personnelles, en offrant une boite à outils pour l'expression des éléments de politiques de protection des données. Néanmoins, ces langages présentent des limitations en termes de prise en compte des réglementations portant sur les six axes de protection des données personnelles.

couvrant la représentation des six axes de protection des données, nous nous sommes intéressés à l'intégration des concepts de protection des données à ceux des langages de contrôle d'accès [Bek10]. Nous avons établi un état de l'art des modèles de contrôle d'accès proposant des éléments de protection des données et nous avons présenté leurs limitations. On s'est intéressé plus particulièrement à XACML. Ce dernier adoptant le concept d'attributs pour la spécification des règles d'accès, permet de définir des politiques d'accès fines et mieux adaptées aux cas de figures réels. De ce fait, nous avons

pu définir des extensions à ce langage pour couvrir les aspects de protection des données personnelles.

Dans ce chapitre, nous avons présenté notre langage XPACML pour la protection des données personnelles, et les aspects implémentation qui lui sont associés. Nous avons

présenté également PPL, un nouveau langage de protection des ressources d'entreprise basé sur le standard XACML. PPL fournit à l'utilisateur des éléments liés à la protection des données collectées en vue d'une autorisation d'accès. Puis, nous avons fourni une comparaison de PPL avec XPACML dont la ressource de base à protéger est la donnée elle-même.

Reposant sur des contraintes de traitement issues des six axes de protection des données (section 1.3.1 du chapitre1), XPACML répond à la problématique de présentation des axes règlementaires, et celle de contrôle d'usage associé aux données (section 2.1). L'implémentation de quelques composants qui y sont contenus sera détaillée dans le chapitre 4 de ce mémoire.

La présentation des aspects réglementaires à elle seule n'est pas suffisante. Il faut comme nous l'avons mentionné dans la première problématique de l'introduction, les mettre en relation avec les différents contextes d'usage des données. Nous présentons dans le chapitre suivant un modèle d'information sémantique dédié à cet effet.

CHAPITRE 3

PRISE EN COMPTE DU CONTEXTE SITUATIONNEL DANS LA PROTECTION DES DONNEES PERSONNELLES

Sommaire

Introduction 120
3.1 Problématique 120
3.2 Etat de l'art 121
3.3 Discussion et idée principale 122
3.4 Prérequis en modélisation sémantique et gestion des contextes 124
3.4.1 La modélisation sémantique 124
3.4.2 Gestion de contextes 127
3.5 Approche proposée 129
3.5.1 Modélisation des contextes 130
3.5.2 Politiques de protection des données à base de contextes 137
3.6 Scénario 140
3.6.1 Modélisation des contextes sémantiques 142
3.6.2 Politiques de protection des données à base de contextes 145
3.7 Prototype d'implémentation 146
3.8 Conclusion 148

3 Prise en compte du contexte situationnel dans la protection des données personnelles

Introduction

Les politiques de protection XPACML que nous avons proposées dans le chapitre 2, reposent sur un vocabulaire statique pour l'expression de ses éléments de gestion d'accès aux données. Dans une perspective de rendre les politiques XPACML dynamiques et sensibles aux différents contextes situationnels (liés aux différentes transactions), nous démontrons dans ce chapitre comment les contextes situationnels peuvent être modélisés sémantiquement et intégrés dans les politiques de protection XPACML en nous reposons sur l'aspect « contexte ».

Pour ce faire, nous détaillons les besoins associés aux politiques de protections sensibles au contexte dans la section 3.1. Puis, nous étudions un état de l'art des solutions proposées dans les domaines de la protection des données, du contrôle d'accès, et des langages sémantiques (section 3.2) afin d'identifier les éléments ciblés dans la littérature pour la prise en compte des contextes dans les politiques d'accès. Cet état de l'art montre des limites en termes d'approches et d'outils utilisés que nous discutons dans la section 3.3. Nous abordons dans la section 3.4 les prérequis nécessaires à la représentation sémantique et à la gestion des contextes. Ensuite, nous détaillons dans notre approche de prise en compte des contextes dans les politiques de protection (section 3.5). Enfin, un déroulement détaillé de notre approche est donné sous forme de scénario (section 3.6). Les aspects liés à l'implémentation de notre système sont donnés dans la section 3.7, avant de conclure avec un bilan de nos contributions (section 3.8).

3.1 Problématique

Les contraintes légales de traitement (et les règles/politiques XPACML qui en découlent) à appliquer sur les données personnelles de l'utilisateur, diffèrent selon leur nature, leur destinataire, et l'objectif d'usage. Ainsi, il est nécessaire de disposer d'un moyen de représentation des SPs, des données personnelles, et des cas d'usage qui leur sont associés (première problématique citée dans l'introduction). L'identité du SP avec le but de la requête en cours constituent actuellement les éléments principaux caractérisant ces cas d'usages. Si la présentation des SPs reste relativement simple, la formalisation des cas d'usage peut aller au-delà d'un simple but à un contexte transactionnel plus complexe.

Selon Sun et Sauvola [Daw06], un contexte est défini comme une information implicite sur une situation, ou un environnement donné de deux ou plusieurs entités en communication. Dans notre travail, nous définissons un contexte situationnel comme toute information implicite liée à une situation transactionnelle, plus particulièrement, toute

information de haut niveau liée à un SP, une ressource, et/ou un environnement. Pour des raisons de simplification, nous l'appelons « contexte » dans le reste de ce mémoire.

De ce fait, les politiques XPACML à appliquer sur les données ne sont plus basiques. Elles sont désormais basées sur des informations sémantiques relatives à chaque situation, plutôt que sur la *syntaxe* des ressources, sujets, actions et conditions contenues dans les requêtes. En effet, nous nous concentrons particulièrement sur la problématique du de protection des données basée sur le contexte, et comment les contextes peuvent être intégrés dans les politiques de protection des données.

3.2 Etat de l'art

Chen et al. [Che03] se sont concentrés sur la présentation des contextes de manière formalisée. Les contextes considérés étaient principalement des concepts basiques, comme l'individu, les évènements, …etc. Ce travail a servi comme une première approche utilisant les technologies du web sémantique pour la présentation d'un contexte. Seulement, dans plusieurs environnements comme celui de la protection des données personnelles, ou du contrôle d'accès, ces concepts sont insuffisants, voire inadéquats.

Un travail intéressant dans le domaine du contrôle d'accès adoptant le concept du « contexte » comme élément principal pour la spécification des politiques de sécurité et leur application, est nommé UbiCOSM (Ubiquitous Contexte-based Security Middleware)[Cor04]. Contrairement aux modèles d'accès traditionnels basés sur l'aspect du rôle (sujet), les règles/politiques de contrôle d'accès dans ces modèles sont directement associées aux contextes.

UbiCOSM adopte un format basé sur RDF [Wor04] pour la présentation des contextes afin de couvrir l'hétérogénéité des données prises en compte. Seulement, il n'étend pas ce format pour inférer des relations entre les entités. En d'autres termes, il ne supporte pas la déduction des contextes de haut niveau depuis ceux de bas niveau.

Dans le domaine de la protection des données personnelles, les auteurs dans [Dhi07] poussent l'idée d'utiliser les technologies du Web sémantique comme base pour traiter la problématique du contrôle d'accès aux données basé sur le *contexte*. Les auteurs prennent l'acte fédéral allemand comme base pour la présentation d'un contexte législatif et la déduction des politiques de contrôle d'accès qui en découlent. Les données à protéger (ressources) ou les SPs (sujets) demandant l'accès à ces données n'ont pas été pris en compte dans la conception du contexte.
Le travail fournit dans [Dhi07] a mis le point sur l'aspect architectural de la plateforme de contrôle d'accès basé sur XACML, et les outils nécessaires pour l'établissement du modèle sémantique, sans pour autant présenter le modèle conceptuel des contextes en question ou leur implémentation.

Tout comme dans [Dhi07], les auteurs dans [Mic08] proposent un modèle sémantique pour les principes fondamentaux de la directive européenne 95/46/EC, en se basant sur le modèle d'authentification et d'autorisation classique. Le modèle sémantique proposé est assez générique, que ce soit dans les concepts pris en compte dans la modélisation

(DataSubject, Entity, Ressource), ou dans les politiques de protection à appliquer sur les données. Ces dernières sont limitées à l'ajout d'un élément *Resource* aux tags P3P (Purpose, Recipient, Retention) présentées dans la section 2.2.1 du chapitre 02.

Le travail le plus proche du notre est celui de D.N.J [Daw06]. Les auteurs proposent un modèle sémantique regroupant les croyances qu'un utilisateur peut avoir sur une situation transactionnelle donnée. Le modèle proposé est très générique, il est inspiré du modèle de contrôle d'accès RBAC [Hu06], car il associe des rôles aux utilisateurs et aux SPs. La spécification du modèle de données de l'utilisateur ainsi que ses préférences en termes de vie privée reposent entièrement sur le schéma des données et des politiques P3P [Wor06].
La prise en compte de la législation a été très grossière dans le modèle, où une classe « *Law* » a été spécifiée. Le détail de cette classe à été mis de côté à cause de la divergence des textes législatifs d'un pays à l'autre. Ainsi, la spécification des politiques de protection des données n'a pas été abordée dans ce modèle.

En termes de langages de politiques sémantiques, Kagal [Kag04] a proposé le langage de politique *Rei* qui permet aux politiques d'être écrites en utilisant des sémantiques. *Rei* a été implémenté en utilisant le langage *N3* et le framework de politiques *Rein* [Kag05]. Les travaux présentés dans [Kag03] et [Pat04] sont des applications de *Rei*.

KAoS [Usz04] est un autre langage sémantique proposé. Il est dédié à la spécification, la gestion, la résolution de conflits et l'application des politiques basées sur des sémantiques. Les politiques *KAoS* sont basées sur OWL. Le langage *KAoS* utilise des mécanismes de Description Logic (DL) pour raisonner sur ces politiques afin de vérifier leur applicabilité.

3.3 Discussion et idée principale

Comme mentionné à la section 3.1, nous nous penchons dans ce chapitre sur la problématique de protection des données basée sur les contextes, et comment ces derniers peuvent être intégrés dans les politiques exprimées avec le langage XPACML (section 2.6 du chapitre 2).

Les travaux liés au domaine de la protection des données présentés dans la section 3.2 sont basés soit sur un modèle d'authentification et d'autorisation classique, ou sur un modèle RBAC [Hu06], ou dans les meilleurs des cas sur un modèle ABAC [Wan04] comme le travail [Dhi07] basé sur XACML.

En effet, l'approche ABAC (*Attribute Based Access Control*) [Wan04], substitue le concept de *rôle* utilisé par le modèle *Role Based Access Control* (RBAC) [San96] [Fer01] par celui *d'attribut*. Un *attribut* est toute caractéristique, pertinente associée à un sujet, à une ressource, à une action, ou encore à l'environnement.
Ce concept a apporté une grande flexibilité et extensibilité à l'expression des politiques de contrôle d'accès. Néanmoins, leur charge d'administration en termes de complexité de spécification et de maintenance ne cesse d'augmenter.

Afin de spécifier les règles d'accès, un administrateur doit définir un schéma d'attributs à appliquer sur les données personnelles de l'utilisateur (ressources). Ce schéma est souvent différent du schéma d'attributs spécifié par le (les) SP(s) faisant les requêtes d'accès aux données personnelles.

Ainsi, le besoin d'une solution de contrôle d'accès basée sur un modèle sémantique unifiant le vocabulaire entre l'utilisateur et le SP reste présent. Ce modèle sémantique permet alors de raisonner et d'interpréter les requêtes contenant des termes de haut niveau, et qui n'ont pas été spécifiés dans les règles de contrôle d'accès.

La solution UbiCOSM [Cor04] vient répondre à ce point qui fait défaut aux solutions proposées dans le domaine de la protection des données personnelles, seulement, elle ne permet pas d'inférer des liens entre les classes (section 3.4) d'un même contexte. En d'autres termes, elle ne supporte pas la déduction des contextes de haut niveau, depuis ceux plus basiques.

Les travaux [Kag04, Usz03] fournissent un moyen pour l'écriture des politiques de contrôle d'accès par le biais de langages sémantiques, seulement aucune des solutions proposées donne une *séparation* claire entre la modélisation des contextes et le contrôle d'accès.

De ce fait, il est difficile de différencier entre les règles/politiques de contrôle d'accès et les règles de dérivation d'un contexte donné. Comme le contexte doit être évalué en temps réel, la viabilité de telles solutions se trouve alors vite diminuée.

Pour répondre à ces problématiques, nous proposons d'étendre l'architecture XPACML (section 2.6.3 du chapitre 2), avec des composants gérant les informations sémantiques, afin de la rendre sensible aux contextes. Plus concrètement, nous proposons une représentation formelle des informations qui peuvent être associées aux sujets, aux ressources, et/ou à l'environnement par des contextes sémantiques.

Afin de favoriser la prise en compte et le traitement de ces informations sémantiques, nous nous basons sur le concept *d'attribut* du modèle ABAC précédemment expliqué. Ceci permet de partager un vocabulaire commun entre l'utilisateur et le SP, et aux règles/politiques de protection des données personnelles d'être établies en se basant sur les *sémantiques* liées aux SPs et aux données personnelles plutôt que sur la *syntaxe* des requêtes, des règles/politiques de contrôle d'accès correspondantes. Les politiques de protection des données seront alors crées en utilisant des termes sémantiques de haut niveau et déployées sur l'architecture XPACML (section 2.6.3 du chapitre 2) afin de contrôler la révelation des données personnelles.

L'utilisation de la sémantique permet également l'analyse à différents niveaux d'abstraction des politiques de protection des données. Nous permettons ainsi de faire un raisonnement sur les données, leur destination et les contraintes légales qui s'appliquent sur elles.

Dans notre travail de modélisation sémantique et de gestion des politiques de protection, nous proposons une approche permettant d'établir une séparation claire entre les deux.

Ainsi, au moment de l'écriture des politiques XPACML, un administrateur (ou l'utilisateur) n'a pas à se soucier du sens du (des) contexte(s) impliqué(s) dans la règle/ politique XPACML. De la même manière, tout changement (ajout/modification) apporté aux contextes, est sans influence sur les politiques de protection des données.
Ainsi, notre solution qui se veut évolutive en termes d'attributs (du SP, et des données personnelles), permet une nette réduction de la charge d'administration des politiques.

Nous apportons les prérequis nécessaires pour notre solution en termes de modélisation sémantique des connaissances et gestion des contextes via des outils du Web Sémantique dans la section suivante.

3.4 Prérequis en modélisation sémantique et gestion des contextes

3.4.1 La modélisation sémantique

Nous avons précisé dans l'introduction de ce mémoire que les règles/politiques de protection des données seront basées sur des contextes situationnels de vie privée. Ces contextes peuvent être : la localisation, une activité, une identité, … etc. Ils sont issus d'un domaine de connaissances (E-Commerce par exemple) qui se veut commun entre les entités en communication (Utilisateur et SP). Ce domaine a pour objectif principal de permettre le partage et l'échange des informations entre l'utilisateur et les différents SPs potentiels.

Différentes initiatives du Web Sémantique sponsorisées par le consortium W3C ont été proposées pour répondre à cet objectif.

A titre d'exemple, le langage XML [Xml] fournit des mécanismes nommés XML DTD, et des schémas XML pour déclarer et échanger des structures de données entre deux parties ayant établi un accord préalable sur des définitions précises. Cependant XML manque de moyens de présentation et de raisonnement sur les sémantiques associées à ces données. Ainsi, un même terme peut être utilisé pour des éléments ayant le même sens. En conclusion, XML est un langage puissant permettant d'établir des documents structurés, néanmoins, il n'impose pas de contraintes sémantiques sur la signification des éléments contenus dans ces derniers.

Basé sur XML qui est utilisé pour la syntaxe, RDF [Wor04] contient un ensemble clair de règles pour fournir une information descriptive simple au niveau sémantique. Le schéma RDF (RDFS) fournit une manière pour combiner plusieurs descriptions RDF avec le même vocabulaire. RDF est basé sur un modèle formel concret, qui utilise les graphes dirigés pour représenter les sémantiques des méta-données.

Avec le schéma RDF, il est possible de définir des classes (concepts) (qui peuvent avoir des super-classes ou des sous-classes), des propriétés (qui peuvent avoir des sous-propriétés), des domaines, et des rangs. En ce sens, RDF est un langage adapté pour représenter des modèles sémantiques simples. Cependant il est nécessaire d'utiliser des langages de présentation des sémantiques riches non assurées par RDF (particulièrement au moment d'interopérabilité de schémas RDF indépendants).

OWL (Ontology Web Language) [Web10] a été conçu pour répondre à ce besoin. Il est fondé sur la capacité du langage XML à définir des schémas de marquage personnalisés, et l'approche flexible du RDF liée à la représentation sémantiques des données. Cette représentation des données et leurs relations est appelée « *une Ontologie* ».

Plus concrètement, une ontologie est une description formelle et explicite des concepts dans un domaine de classes. Les propriétés de chaque classe décrivent ses caractéristiques, ses attributs, et les restrictions appliquées sur ses relations [Web10]. La syntaxe normative d'échange OWL est le RDF/XML.

Fondé sur RDF et RDFS, OWL ajoute un vocabulaire formel pour décrire les propriétés et les classes. Par exemple, si une classe est équivalente à une autre classe, si une propriété donnée est transitive, symétrique, fonctionnelle ou inverse à une autre.

Quoique les standards du Web sémantique soient à l'origine conçus pour les applications web, nous pensons que ces outils sont bien adaptés pour notre contexte de travail, pour les raisons suivantes :

- Une ontologie exprimée avec un langage du Web sémantique fournit des moyens pour être développée indépendamment du système auquel elle appartient. Le partage des informations contextuelles est possible, permettant ainsi une minimisation du coût de la redondance.

- RDF et OWL sont des langages de représentation des connaissances, avec une forte expressivité, permettant la modélisation de différents types d'informations contextuelles (exemple : les informations liées à un utilisateur, des évènements, …etc).

- L'ontologie formalisant les contextes de protection fournit une représentation explicite de leurs sémantiques, qui peuvent être raisonnées par les moteurs d'inférence logiques existants. En appliquant un moteur d'inférence du Web sémantique, les applications sensibles au contexte peuvent utiliser des règles logiques spécifiques pour déduire un contexte implicite d'un contexte explicite. En adoptant cette approche, il ainsi est possible de personnaliser les inférences du contexte.

- Les langages du Web sémantique peuvent être utilisés comme des méta-langages pour définir d'autres langages plus spécifiques comme les langages pour la communication et le partage de connaissances, des langages de politiques de sécurité et de protection de la vie privée. L'interopérabilité est un avantage clé des langages du Web sémantique.

Comme illustré dans la figure 3.1, en termes de modélisation, une ontologie est considérée comme un ensemble de « things » individuels.

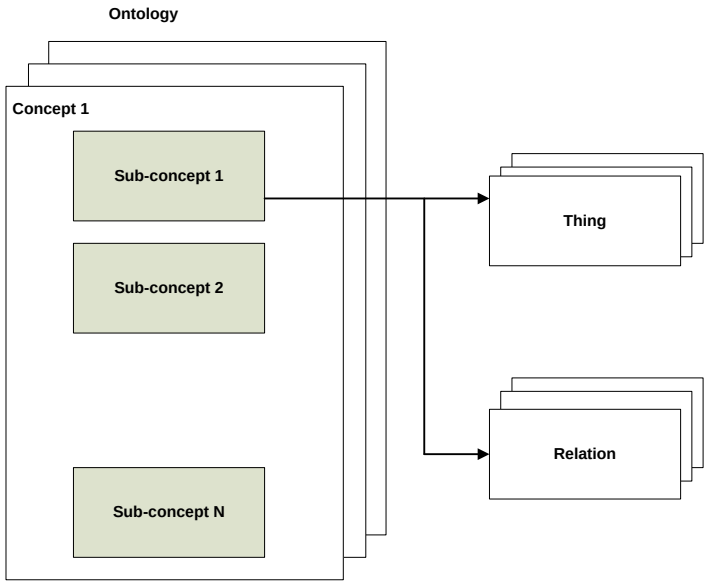


Figure 3.1. Modèle conceptuel d'une ontologie

Ils peuvent être résumés par les quatre composants suivants :

- **Ontologie** : Une ontologie est un répertoire de connaissances responsable de la création, l'accès, et la suppression de « Things ». Elle fournit également un « namespace » uniforme pour tous les « Things » appartenant à l'ontologie.

- **Thing** : un membre de l'ontologie ayant une signification logique. Il encapsule la connaissance à propos d'un individu spécifique pour un domaine d'intérêt donné. Un « Thing » peut être considéré comme un ensemble de déclarations qui spécifient les relations entre un « Thing » donné et d'autres « Thing », ou traite les valeurs de données. Chaque « Thing » possède un identifiant unique avec une portée dans l'ontologie. En combinaison avec un namespace de l'ontologie, un identifiant forme un URI pour un « Thing » donné pour l'adresser depuis l'extérieur de l'ontologie.

- **Concept et sous-concept** : Il est défini comme une classe dans la terminologie OWL. C'est un type spécifique de « Things », fournissant une classification hiérarchique (taxonomie) d'autres « Things ». Un concept ou un Sous-Concept spécifique, définit un groupe de « Things » qui partage quelques relations communes. Un « Thing » appartenant à un Concept spécifique est nommé « Instance » de ce Concept. Cette séparation entre Concept et Sous-Concept nous permet de réserver les Concepts aux *subjects* généraux de l'ontologie, et Sous-Concepts aux *subjects* spécifiques du Domain.

- **Relation** : elle est définie comme une propriété dans la terminologie OWL. Une relation est un type spécial de « Things » fournissant une spécification d'une *Relation* entre « Things », ou depuis « Things » vers les valeurs des données actuelles.

3.4.2 Gestion de contextes

3.4.2.1 Acquisition du vocabulaire des contextes

L'ontologie des contextes de protection fournit une structure globale des concepts et des termes spécifiques utilisés pour décrire un modèle du domaine « protection des données ». Seulement, un contexte ne peut être utilisé sans assertions. Ces dernières sont un ensemble d'hypothèses sur le sens qui doit être attaché aux éléments du contexte, facilitant ainsi le processus d'acquisition et de catégorisation des contextes réels de protection dans une base de connaissances. Une base de connaissances peut être ainsi considérée comme un couple constitué d'une ontologie et d'assertions (faits) décrivant les individus et les relations dans lesquelles ils sont engagés. De ce fait, le processus d'acquisition s'effectue en renseignant les vocabulaires des classes et des relations liés à chaque contexte. Nous avons effectué cette étape en utilisant une plateforme nommée Protégé V3.4 [Pro03].

3.4.2.2 Raisonnement sur le vocabulaire des contextes

OWL possède une capacité de raisonnement sur les vocabulaires définis dans une ontologie.

- Les relations entre classes : si X est une instance de la classe C, et C est une sous classe de D, on peut inférer que X est une instance de D.
- L'équivalence des classes : si une classe A est équivalente à une classe B, et la classe B est équivalente à la classe C, alors la classe A est équivalente à la classe C.
- Cohérence : Supposons qu'une classe X est déclarée être une instance de la classe A, et la classe A est une sous classe de B $\cap$ C. A est une sous classe de D, et B et D sont des classes disjointes. Ainsi, il y a une incohérence car la classe A doit être vide, mais elle possède tout de même une instance X. Ceci est une indication d'erreur dans l'ontologie.
- Classification : si on déclare qu'une paire propriétés/valeurs sont des conditions suffisantes pour appartenir à une classe A. Alors si un individu X satisfait de telles conditions, on peut conclure que X doit être une instance de A.

3.4.2.3 Raisonnement sur les relations

Les relations sont utilisées pour arranger les structures hiérarchiques ainsi que les concepts de l'ontologie. Chaque relation peut être une sous-relation plus générale d'une autre. Une nouvelle sous-relation hérite de l'ensemble des concepts du domaine et les rangs de ces prédécesseurs, et peut l'affiner pour des besoins plus spécifiques. Un exemple de raisonnement sur la relation de généralisation est donné dans la section 3.5.1.3.

3.4.2.4 Gestion des contextes

La figure 3.2 illustre le processus de gestion du contexte. Un « context manager » est un mécanisme de médiation entre les informations issues du monde réel et une base de connaissances sémantiques. Il est composé de deux processus : le processus d'acquisition de contexte et le gestionnaire des règles.

L'acquisition des contextes via des assertions (directes dans notre cadre de travail), déclenche le gestionnaire des règles, afin d'inférer sur les instances renseignées. Les résultats d'inférence de ces instances sont renseignés dans une base sémantique (assertion indirecte)

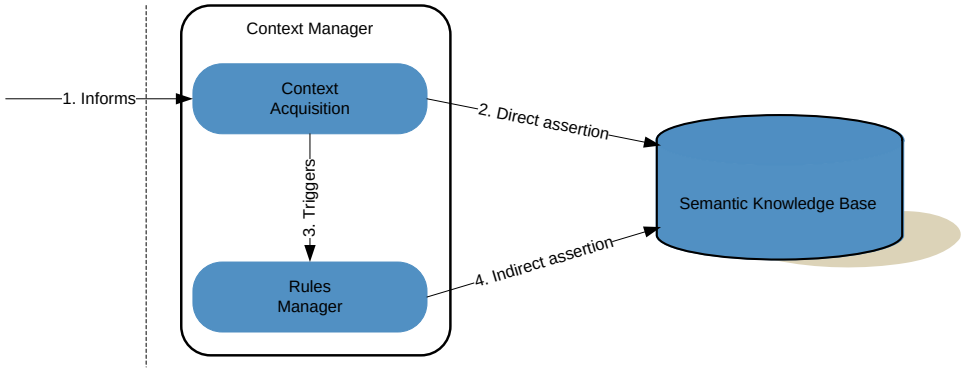


Figure 3.2. Gestion du contexte

3.5 Approche proposée

La figure 3.3 illustre notre processus de création de politiques à base de contextes.

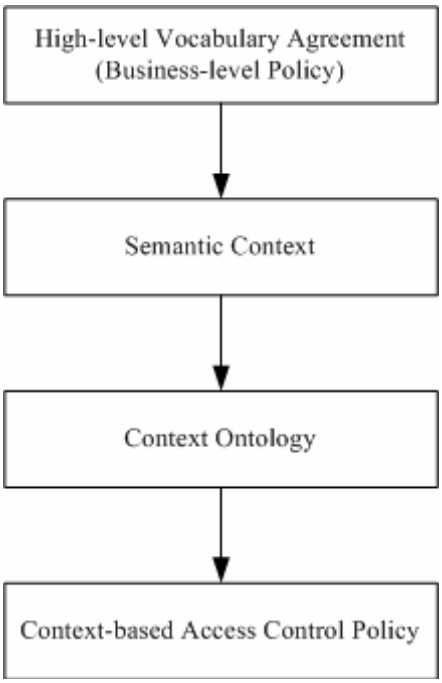


Figure 3.3. Processus de création de politiques XPACML basées sur le contexte

Tout d'abord, il est nécessaire d'avoir un accord entre l'autorité de protection et le SP sur le vocabulaire de haut niveau (termes liés aux contextes de protection). Ensuite, il est nécessaire d'avoir une couche sémantique, formalisant les sémantiques liés aux concepts de ces contextes.

Une fois les sémantiques liées aux concepts des contextes clarifiées, il est alors possible de les modéliser avec une ontologie. Cette dernière offre la possibilité d'une représentation formelle des concepts de protection des données dans un domaine de classes. En effet, l'ontologie inclut des définitions interprétables par machine des concepts, ainsi que les relations qui les animent. Ce sont ces définitions que nous utilisons pour construire des règles /politiques de protection.

Les sous sections suivantes illustrent la formalisation de notre modèle sémantique de protection des données, comment ce dernier est utilisé pour définir des contextes de

protection, et comment ces contextes sont intégrés dans les règles/politiques de protection XPACML (section 2.6.2 du chapitre 2).

3.5.1 Modélisation des contextes

3.5.1.1 Représentation du vocabulaire

Nous nous basons dans cette partie sur les avantages fournis par les langages du Web Sémantique, et principalement OWL afin de répondre aux objectifs suivants dans la conception du vocabulaire nécessaire à la définition des contextes de protection des données.

- La représentation du vocabulaire d'une manière formelle, traitable par un système de politiques.
- Le partage d'un même vocabulaire entre l'utilisateur et le SP.
- L'ajout fait de nouveaux vocabulaires.

Comme expliqué dans la section 3.4, les concepts sont les premiers éléments à définir dans une ontologie.
La figure 3.4 illustre les concepts principaux de notre ontologie

Figure 3.28 Concepts de l'ontologie de protection des données

Le concept DataType : correspond au profil utilisateur et présente l'ensemble de ses données personnelles (nom, prénom, adresse, …etc),
Le concept ServiceType : une taxonomie des types de services selon le secteur (ex : e-commerce, e-learning, …etc),
Le concept Rule est une représentation des règles de protection des données personnelles, conformément à la législation présentée dans la section 2.2,

L'ontologie que nous proposons met en relation les modèles DataType et ServiceType par le biais des règles de protection, représentées par la classe ***Rule***. Ainsi, notre ontologie fournit un vocabulaire détaillé des types de données et des types de services, structurés d'une manière hiérarchique avec des règles d'héritage bien définies. L'ontologie obtenue grâce à la plateforme Protégé est illustrée dans la figure 3.4. Elle considère les éléments de l'ontologie comme des objets Java. Nous avons utilisé la plateforme Protégé 3.4.4 [Pro03] basée sur JAVA pour éditer l'ontologie présentée dans la figure 3.4.

Nous démontrons avec l'exemple suivant la création du concept DataType .
Ontology ***Privacy environment*** =

*Connector.createOntology(‘‘’’) ; try {Concept **DataType** = Privacy environment.createConcept(‘‘DataType’’);*
*Concept **DataType** =Privacy environment.createConcept(‘‘DataType’’);*

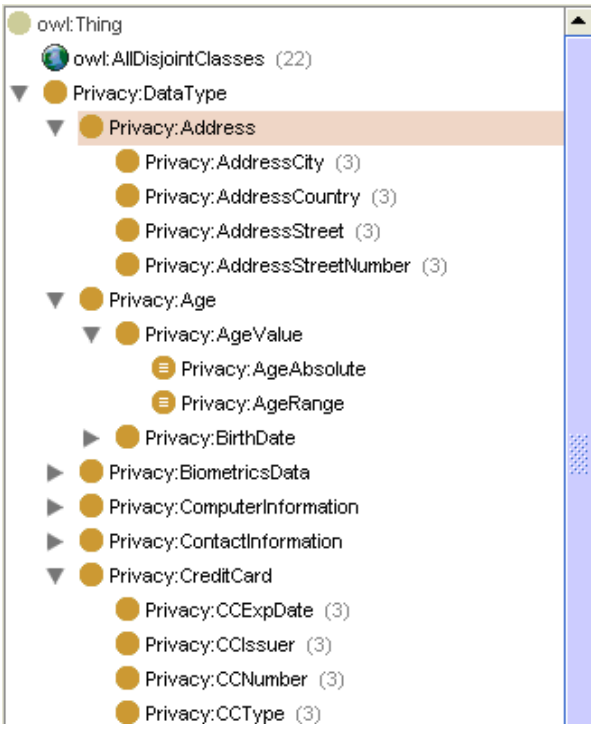


Figure 3.4. Sous-graphe DataType de l’ontologie de protection des données

Après la création du concept DataType, nous adoptons la même stratégie pour construire le sous-graphe ServiceType présenté dans la figure 3.6. Les différents types des services sont définis comme sous-classes de la classe « ServiceType », sous forme d’une hiérarchie de types de services permettant l’héritage des caractéristiques. Les objets finaux ServiceType, sont en effet les feuilles des différentes classes du sous-graphe services. Elles représentent des instances OWL des classes « ServiceType ».

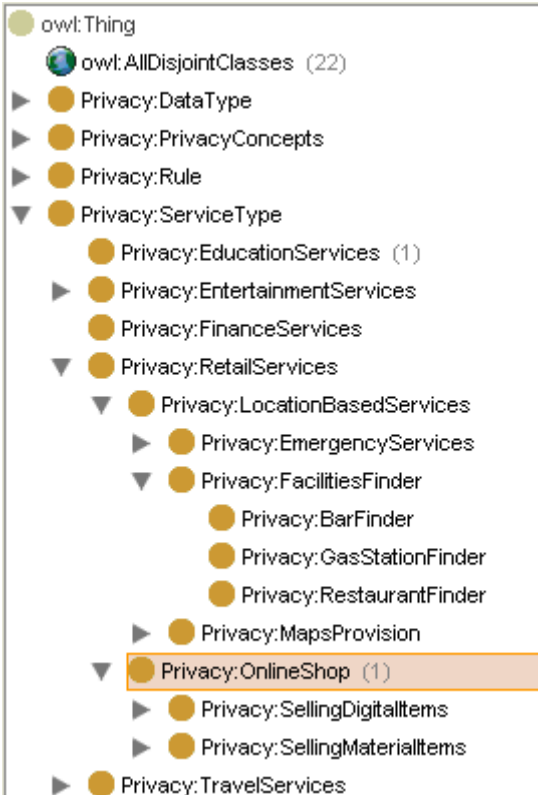


Figure 3.5. Sous-graphe ServiceType de l'ontologie Privacy

Les éléments des règles de protection requises pour réguler la révélation et la collection des données personnelles sont une partie d'un troisième sous graphe ayant le concept « Rule » comme classe mère. Sous-Concepts et instances du Concept « Rule » expriment les axes législatifs de protection (section 1.3.1 du chapitre 1). La figure 3.7 démontre les sous classes de la classe « Rule » et ses instances.

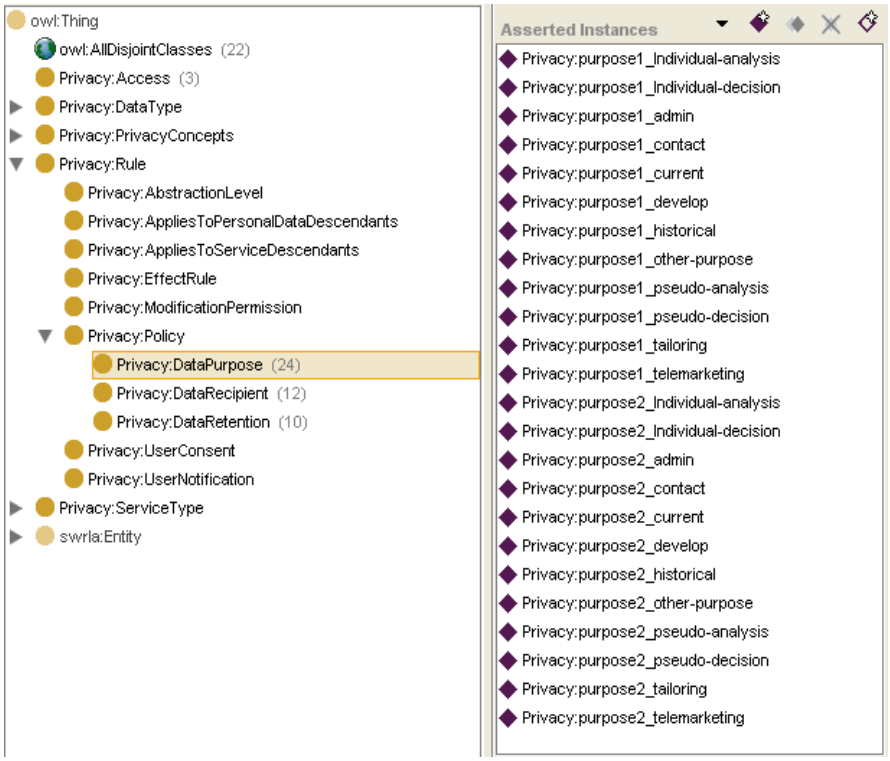


Figure 3.6. Sous-graphe des éléments des règles de protection des données

Ce sous-graphe a été conçu comme support à la négociation des politiques entre l'utilisateur et le SP (chapitre 4). C'est important de noter que deux instances sont assignées à chaque élément « `POLICY` » du Concept « Rule ». Une instance définit la politique de protection minimaliste, et l'autre définit la politique de protection la plus large allouée pour un service de type ServiceType demandant un élément de donnée de type DataType spécifique. Nous avons pu faire ceci grâce à notre classification ordonnée des instances de chaque sous élément de « `POLICY` » en accord avec le niveau de risque qu'il représente sur la vie privée de l'utilisateur (section 4.3 du chapitre4). Comme mentionné dans la même section, ce classement est dynamique en fonction du type de donnée en question. Ainsi, nous avons lié les trois sous graphes avec une classe nommée « Access ». Les objets instanciés de cette classe (« AccessObjects ») lient les instances de services et celles des données personnelles avec les instances des règles législatives appropriées, en utilisant les propriétés OWL "refersToService", "refersToData", et "refersToRule" (figure 3.8).

Il est à préciser que nous somme contenté de cette modélisation qui répondait bien à nos objectifs de négociation de politiques (chapitre4). La définition de relations fines entre les différents types de données et les sous éléments de « `POLICY` » n'ont pas été traités.

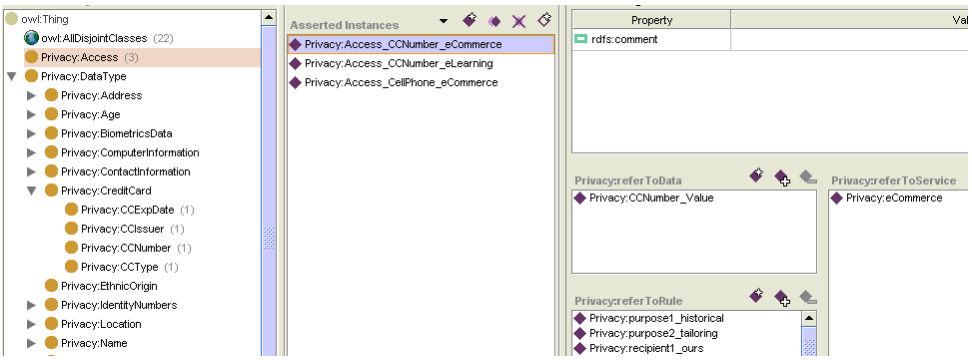


Figure 3.7. La classe Access

Les nouveaux concepts et sous-concepts sont intégrés facilement vu que notre ontologie est basée sur un modèle hiérarchique extensible.

L'ontologie de protection des données doit être déployée sur le côté utilisateur, comme sur le côté SP. Ainsi les deux entités partage un vocabulaire commun.

3.5.1.2 Relations

Une fois les concepts et les sous-concepts définis, nous avons à rajouter du sens en complémentant l'ontologie avec des relations entre les concepts et sous-concepts. Tout d'abord, nous présentons ci-dessous une liste de relations sous forme d'objets JAVA. Chaque relation est un objet indépendant. Une relation a les trois caractéristiques suivantes : transitivité, symétrie et généralisation. Nous utilisons ces caractéristiques afin de permettre la déduction des contextes implicites depuis ceux plus explicites.

inheritsFromData :
Cette propriété OWL exprime la relation d'héritage entre un type de donnée général et un autre plus spécifique. Elle est implémentée par le biais de la propriété OWL « inheritsFromData Object».

hasMoreDetailed/hasLessDetailed :
Cette propriété OWL pour supporter différents niveaux de révélation des données. Ainsi, s'il y a un conflit entre l'utilisateur et le SP sur une donnée, il est possible de la substituer par une autre donnée de niveau d'abstraction plus élevée.

containsType/isContainedToType :
Cette propriété OWL exprime la complexité d'un élément de donnée (par ex : FullName data type contient FirstName, LastName et MiddleName). Cette relation est implémentée par le biais de la propriété OWL « containsType/isContainedtoType », qui en effet, définit un arbre de composition hiérarchique. La figure 3.9 illustre une partie des relations du sous-graphe (DataType).

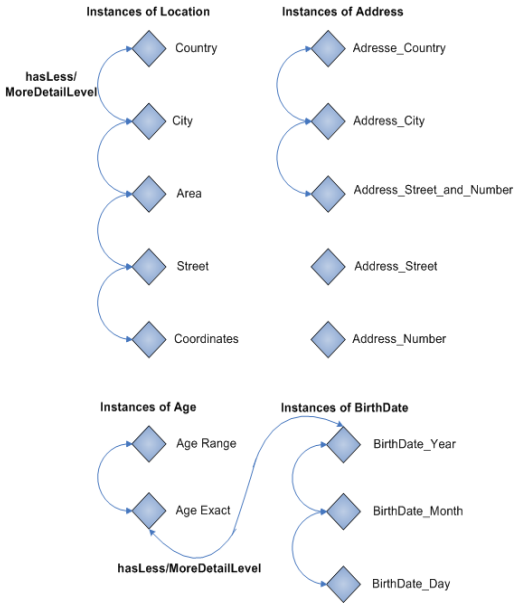


Figure 3.8. Illustration des relations

En accord avec les propriétés définies pour le sous graphe DataType, les propriétés « inheritsFromData » et « containsType/isContainedToType » sont implémentées pour les classes du sous graphe ServiceType.

3.5.1.3 Raisonnement sur les relations

Dans cette section, nous prenons à titre d'exemple, le principe de généralisation de relations qui joue un rôle important dans le raisonnement de l'ontologie. Une déclaration avec une sous-relation implique un ensemble de déclarations implicites avec tous ses prédécesseurs.

Dans le modèle ServiceType, le service « facilitiesFinder » se base sur la localisation de l'utilisateur. Ainsi, nous avons créé le concept «Location » dans le modèle DataType. Nous avons également créé à côté de ce concept, les deux concepts : Region et State. Nous utilisons trois régions pour les exemples suivants : la ville d' « Evry » est localisée dans la région de l ' « Essonne », qui appartient à l'île de France (IDF). Supposons que nous voulons rajouter une nouvelle relation au concept « Region » qui peut déclarer qu'une instance « Region » donnée est une partie d'une autre. Il est évident que cette relation est un cas spécial de « LocatedIn », vu que toutes les parties de la zone sont localisées dedans. Ainsi, pour maintenir une consistance logique de notre ontologie, on doit maintenir et changer ces deux relations. Seulement, on peut éviter un tel travail, si on déclare notre nouvelle relation comme une sous-relation de « LocatedIn ».

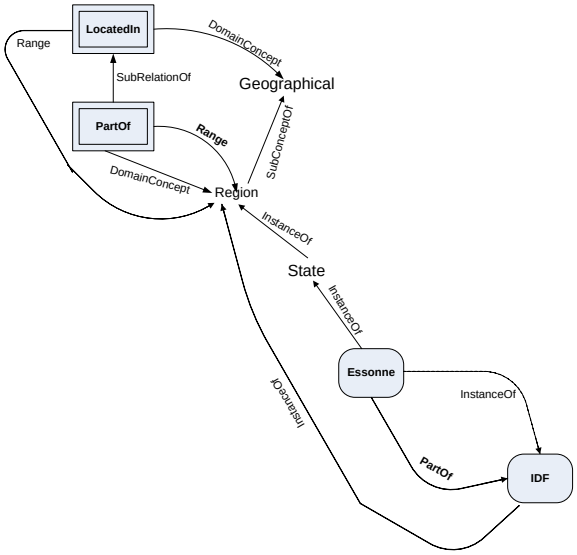


Figure 3.9. Relation « Generalization » « Sub relation »

La relation « locatedIn » apparait automatiquement car la relation « partOf » est une sous relation.

Nous nous contentons de cet exemple d'illustration de relations de généralisation et sous-relations. D'autres types de relations sont possibles : relations transitives, relations symétriques, relations inverses, …etc

3.5.2 Politiques de protection des données à base de contextes

Nous entendons par politiques de protection des données à base de contextes, une politique qui a la capacité d'utiliser le(s) contexte(s) associés à ses termes afin d'établir une décision d'accès. De ce fait, chaque politique doit donc être associéeà un domaine de connaissances. Ainsi, il est nécessaire d'effectuer un raisonnement sur le domaine de connaissances afin d'obtenir les contextes en question. La figure 3.11 illustre les relations entre une politique XPACML, les règles de politique, et l'ontologie des contextes.

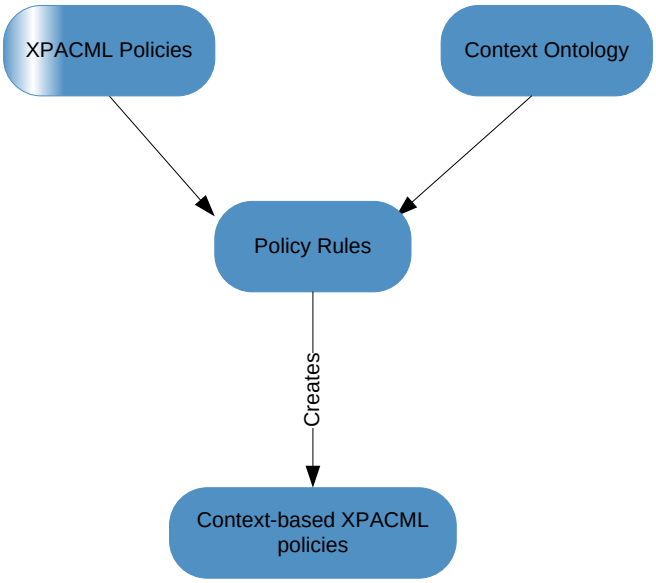


Figure 3.10. Construction des politiques de contrôle d'accès à base de contexte

Une politique peut avoir besoin (ou pas) des contextes définis. Dans le cas où une politique nécessite des contextes, elle doit être alors écrite en utilisant le vocabulaire du contexte directement dans la politique.

A titre d'exemple pour les SPs étrangers engagés dans des activités de E-Commerce, une politique peut être « des SPs qui sont en collaboration avec les SPs locaux alors ils sont autorisés àaccéder aux données des utilisateurs de chacun des SPs ».

Ici, le vocabulaire « enCollaboration », doit être sémantiquement défini dans une ontologie. La politique doit être alors écrite conformément à ce contexte. Pour ce faire, le langage XPACML doit comprendre et interpréter la représentation sémantique du vocabulaire en question.

Une approche pour intégrer les contextes dans la politique XPACML est d'insérer les règles qui infèrent le contexte directement dans la politique. Cependant, cette démarche est lourde dans son application et induit des erreurs. A chaque fois qu'un contexte est requis dans une politique, l'insertion des règles d'inférence dans la politique XPACML doit être faite.

Pour remédier à cet inconvénient, nous préconisons l'usage d'une base de connaissance sémantique qui maintient les informations à propos des contextes définis (vocabulaires …), l'objectif est de donner aux politiques XPACML un accès direct aux termes définis et renseignés dans la base de connaissance sémantique, sans pour autant écrire les règles qui interprètent les contextes une deuxième fois.

Ainsi, une politique de protection XPACML sensible au contexte, est formée en utilisant des contextes définis et probablement leurs valeurs (instances) accessibles dans la base sémantique. Dans l'intégration des contextes dans une politique XPACML, nous faisons usage du concept « attribut » de la spécification XACML.

De ce fait, les politiques XPACML sensibles au contexte, peuvent être établies en se basant sur les attributs des sujets ou des ressources (données personnelles). Ainsi, les contextes sont intégrés dans une politique XPACML comme des valeurs du tag <AttributeID>, et leurs instances comme des valeurs du tag <AttributeValue>.

Du moment que les contextes peuvent être intégrés comme des valeurs d'attribut, il est aussi important d'inclure les contextes associés à un sujet /ressource dans une requête d'accès auprès du PPEP.

La figure 3.12 montre les flux de données dans un système XPACML sensible au contexte.

Par rapport à l'architecture de politiques XPACML illustrée dans la figure 2.13 (chapitre2), nous avons rajouté principalement deux éléments pour permettre d'établir des décisions sensibles au contexte.

Tout d'abord, nous avons créé l'élément OAP (Ontology Administration Point) qui permet de créer l'ontologie de protection et de la charger dans le gestionnaire du contexte (Context Manager). Comme expliqué précédemment, ce dernier, analyse et gère les informations contextuelles en se basant sur l'ontologie, et stocke les connaissances inférées associées aux sujets et aux ressources dans une base de connaissances.

Le PPAP, qui avait pour rôle la génération des politiques XPACML avant de les mettre à disposition du PPDP, doit d'abord chercher les éléments de ces politiques dans les termes contextuels modélisés par l'ontologie.

Une fois ces politiques mises à disposition du PPDP, ce dernier pourra faire appel au DIP pour chercher un complément d'information sur les sujets et/ou les ressources (SP/données personnelles). Recherchées auprès des composants sémantiques « Semantic Subjects/Ressources), les informations retournées sont principalement des contextes situationnels comme « inCollaboration ».

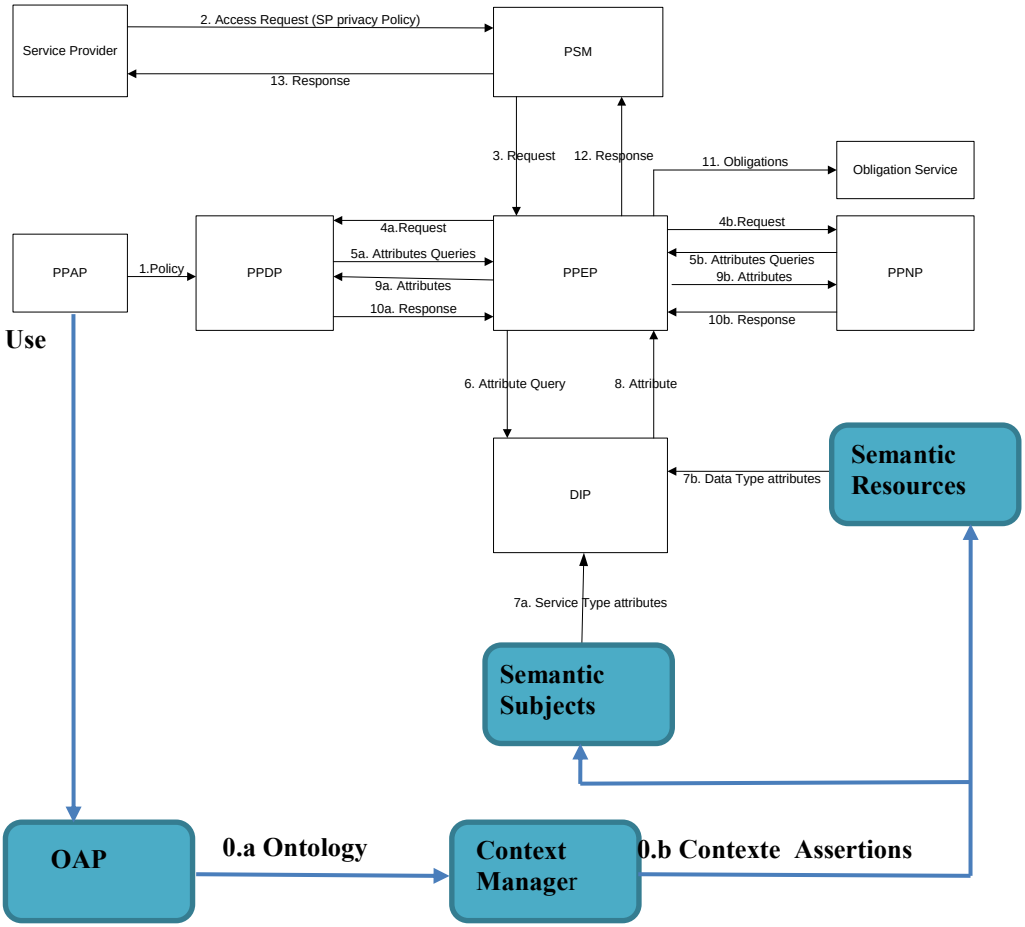


Figure 3.11. Diagramme de flux de données XPACML avec prise en compte du contexte

3.6 Scénario

Nous consacrons cette section à la démonstration de la modélisation des contextes situationnels en nous basant sur notre approche.

Supposons un scénario de commerce électronique, où un utilisateur « Alice » souhaite se procurer un article d'un SP1 local.
Supposons que durant l'accomplissement du service, le SP1 fait appel à un SP2 étranger.

Contrôler qui peut voir les données de l'utilisateur et sous quelles conditions est un challenge significatif dans ce type de scénarios. Il est donc important de limiter l'accès distant aux données dans le temps (par exemple, juste pendant la durée de la transaction). En d'autres termes, l'accès aux données de l'utilisateur doit être contrôlé en se basant sur le contexte situationnel.

Les données de l'utilisateur sont accessibles en se basant sur la connaissance humaine implicite comme : en collaboration, en communication téléphonique. Cette connaissance humaine a besoin d'être capturée et représentée d'une manière formelle comme contextes afin de permettre le partage des données personnelles.

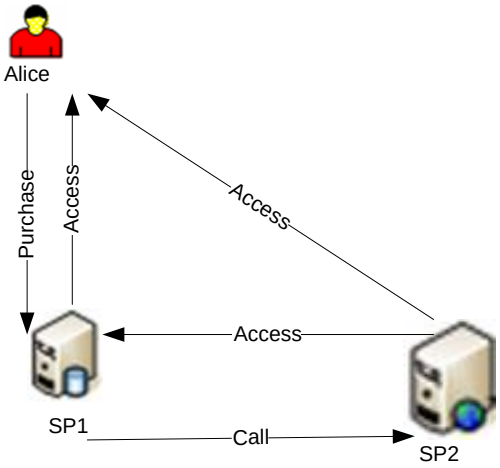


Figure 3.12. Scénario contextuel

Notre but étant de permettre le partage des données personnelles de Alice entre les participants dans une communication, du moment que ce partage se conforme à une politique acceptable.

Un exemple de cette politique pourrait être :
« Tout fournisseur de service appartenant à l'union européenne et participant dans une collaboration avec un fournisseur de service local est autorisé à accéder aux données d'une communication établie par ce dernier ». Cette politique est de haut niveau comparée aux politiques XPACML (section 2.6 du chapitre 2).

La question immédiate qui nous parvient quand on lit une telle politique, est le sens des termes « en collaboration » et « d'une communication ». Ces termes représentent dans notre cas les contextes à définir via une représentation sémantique, pour être utilisés directement dans une politique de protection XPACML.

3.6.1 Modélisation des contextes sémantiques

Dans un souci de lisibilité, nous réécrivons cette politique (« Tout fournisseur de service appartenant à l'union européenne et participant dans une collaboration avec un fournisseur de service local est autorisé à accéder aux données d'une communication établie par ce dernier»), dans un pseudo code XPACML non exécutable, dans un premier temps comme suit :

```
If {
Subject == AnySP
Resource == AnyResource
Action == AnyAction
Condition {
AttributeValue(SubjectAttribute(inCollaboration))==
AttributeValue(ResourceAttribute(inCollaboration)
}
}
Then Permit
```

Dans cette politique, le terme « inCollaboration » a besoin d'être défini. Pour ce faire, nous avons à définir d'autres termes contextuels intermédiaires comme : « inCall », et« insameCall ». « inCall » est un contexte secondaire dérivé des contextes primaires (règles 1 et 2). « inCollaboration », est également un contexte secondaire dérivé du contexte primaire et des contextes secondaires dérivés (règle 3). Ces contextes se réfèrent aux vocabulaires de situation du monde réel qui doivent être intégrés dans notre ontologie de protection. Ils seront ensuite utilisés dans une politique de XPACML pour établir des décisions. Comme mentionné dans la section 3.5, nous pouvons représenter un contexte dans une ontologie en utilisant des classes, et des relations. La figure 3.14 montre les entités définies comme des classes de notre exemple et les relations entre elles.

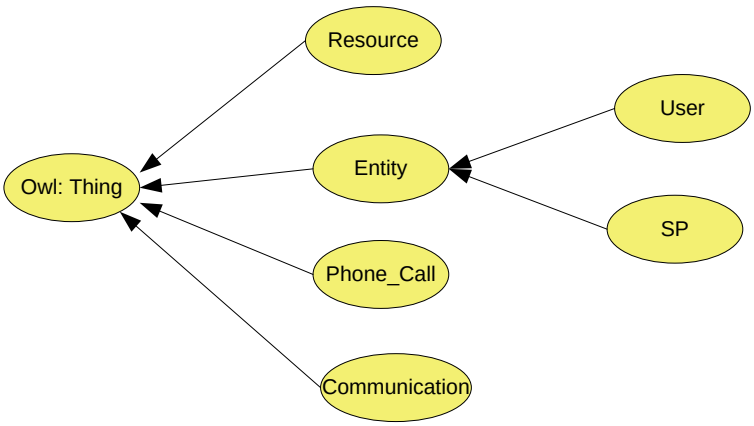


Figure 3.13. Classes et sous classes du contexte « Communication »

Une communication, une entité, une ressource, et un appel téléphonique sont définis comme des classes. La classe « Entity » a deux sous classes : user et SP. Dans le scénario, un appel téléphonique entre les SPs peut être basé sur l'établissement d'une session d'appel en utilisant le protocole SIP (Session Initiation Protocol).

Les relations suivantes sont définies : « inCall », « inSameCall », « inCollaboration », comme montré dans la figure 3.15 avec des lignes discontinues. Ces relations sont les contextes de situation qui ont besoin d'être capturés et utilisés dans une politique XPACML :

- La relation « inCall » veut dire une entité en appel.
- La relation « inSameCall » veut dire que deux ou plusieurs entités sont dans le même appel.
- La relation « inCollaboration », veut dire qu'une entité est dans une collaboration. Une entité peut être une personne, ou un SP.
- Les relations « is a » veulent dire une sous classe de ….

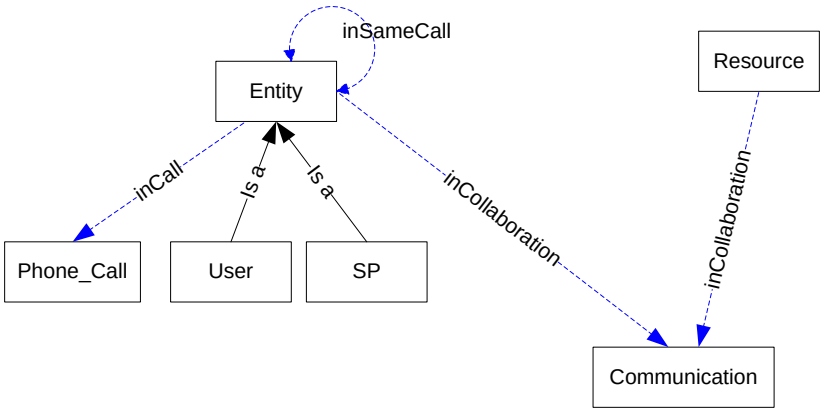


Figure 3.14. Classes et relations liées au contexte « Construction »

Il est maintenant nécessaire de saisir les instances en se basant sur les assertions, et relations définies dans la figure 3.15. Pour l'appel téléphonique, l'identifiant de la session peut être utilisé pour saisir l'instance (de type string) dans la classe « Phone_Call ».

Pour le reste des contextes, nous utilisons un système de règles. Ce type d'assertion, permet d'instancier les relations comme le contexte « inCollaboration ». Nous utilisons dans notre travail, le langage SWRL [Hor04], pour capturer le sens des relations entre classes. Règle 1 définit une entité en appel. Règle 2 définit une entité dans le même appel en utilisant l'identifiant Call-ID de l'appel entre eux.

La règle 3 définit deux entités dans une même communication. Le terme « in a Collaboration » veut dire « y être connecté via un appel », ces deux paramètres sont capturés avec la règle 3.

Règle 1: une entité en appel

$Entity(?entity\_x) \wedge Phone\_Call(?call) \wedge hasCallID(?entity\_x, ?ID) \wedge hasCallID(?call, ?ID) \rightarrow inCall(?entity\_x, ?call)$

Règle 2: deux entités dans un même appel

$Entity(?entity\_x) \wedge Entity(?entity\_y) \wedge Phone\_Call(?call) \wedge hasCallID(?call, ?ID) \wedge hasCallID(?entity\_x, ?ID) \wedge hasCallID(?entity\_y, ?ID) \rightarrow inSameCall(?entity\_x, ?entity\_y)$

Règle 3: deux entités impliqué dans une collaboration via un appel

$Entity(?entity\_x) \wedge Entity(?entity\_y) \wedge Collaboration(?con) \wedge inSameCall(?entity\_x, ?entity\_y) \rightarrow inCollaboration(?entity\_y, ?con)$

En se basant sur les règles précédentes, il est à présent possible de faire les assertions des instances avec leurs classes comme illustré dans la figure 3.16.

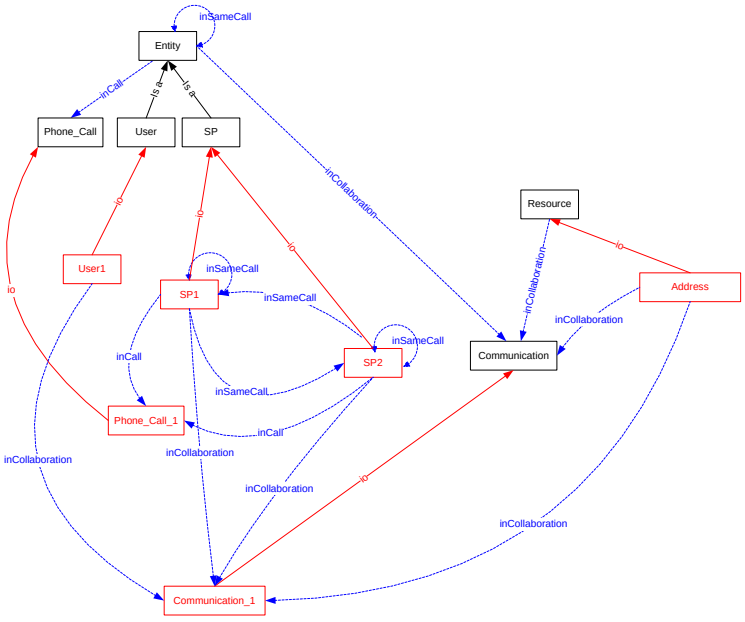


Figure 3.15. Les assertions des différentes instances avec leurs classes

En se basant sur les règles précédentes, la figure 3.16 montre les assertions des différentes instances avec leurs classes, dans une forme graphique.

Dans la figure 3.16, la relation "io" veut dire une instance d'une classe. La relation "isa" signifie "une sous classe de", et chaque ligne discontinue est une propriété définie. Ainsi, on peut remarquer une communication nommée "consultation_1". Après inférence des règles précédentes, on peut noter que:

- La règle 1 exprime que SP2 est dans un appel "Phone_Call_1", et l'utilisateur est dans un appel "Phone_Call_1",
- La règle 2 exprime que SP2 et l'utilisateur sont dans le même appel,
- La règle 3 exprime que SP1, l'utilisateur, et le SP2 collaborent tous dans "communication_1".

Les connaissances renseignées peuvent être stockées dans une base de données relationnelle sous forme de triplet « *Subjet-Property-Object* ».

3.6.2 Politiques de protection des données à base de contextes

En accord avec la modélisation du contexte montrée dans la section précédente, il est actuellement possible d'écrire une politique XPACML en se basant directement sur les contextes définis. La figure 3.17 montre un exemple d'une politique XPACML partielle basée sur le contexte « inCollaboration ». Pour des raisons de lisibilité URI et URN de la ressource ont été supprimés de la politique (figure 3.17) et de la requête (figure 3.18).

```
<Rule RuleId= "1" Effect= "Permit">
<Target>
<Subjects> <AnySubject/> </Subjects>
<Resources> <AnyResources/> </Resources>
<Action> <AnyAction/> </Action>
</Target>

<Condition FunctionId="string-equal">
<Apply FunctionId="string-one-and-only">
<SubjectAttributeDesignator AttributeID="inCollaboration"
DataType="string"/>
</Apply>

<Apply FunctionId="string-one-and-only">
<RessourceAttributeDesignator AttributeId="inCollaboration"
DataType="string"/>
</Apply>
</Condition>
</Rule>
```

Figure 3.16. Exemple partiel d'une politique XPACML basée sur le contexte

Cette politique interprète notre politique de haut niveau, en utilisant le mot clé « inCollaboration ». Et comme les règles précédentes ont déjà capturé et renseigné les

participants de la collaboration, cette politique utilise simplement les contextes et les termes définis dans l'ontologie, et se focalise sur la gestion d'accès. La condition dans la politique est de déterminer si le sujet et la ressource sont de la même communication ou pas en utilisant l'opérateur « égal » (FunctionId = « string_equal »).

Cette approche aide l'administrateur dans le sens où il est possible d'utiliser les contextes dans les politiques XPACML sans se soucier comment les représenter sémantiquement.

```
<Subject>
        <Attribute AttributeID="subject-id" DataType="string">
        <AttributeValue>SP2</AttributeValue>
        </Attribute>
        <Attribute AttributeId="inCollaboration" DataType="string">
        <AttributeValue>collaboration_1</AttributeValue>
        </Attribute>
</Subject>
<Resource>
        <Attribute AttributeId="resource-id" DataType="string">
        <AttributeValue>Address</AttributeValue>
        </Attribute>
        <Attribute AttributeId="inCollaboration" DataType="string">
        <AttributeValue>collaboration_1</AttributeValue>
        </Attibute>
</Ressource>
```

Figure 3.17. Exemple partiel d'une requête d'accès XPACML

3.7 Prototype d'implémentation

Nous avons validé notre approche avec l'implémentation d'un prototype comme preuve de concept. Ce prototype se présente en deux parties : un système de gestion de contexte, et un système de gestion d'accès basé sur l'architecture XPACML (section 2.6.3 du chapitre 2).

Les deux systèmes sont implémentés séparément et sont liés par une base de connaissances via laquelle le système de gestion de contexte fournit les contextes sémantiques collectés dans la base de connaissance, au moment où le système basé sur l'architecture XPACML utilise directement ces contextes pour une évaluation des requêtes d'accès extérieures.

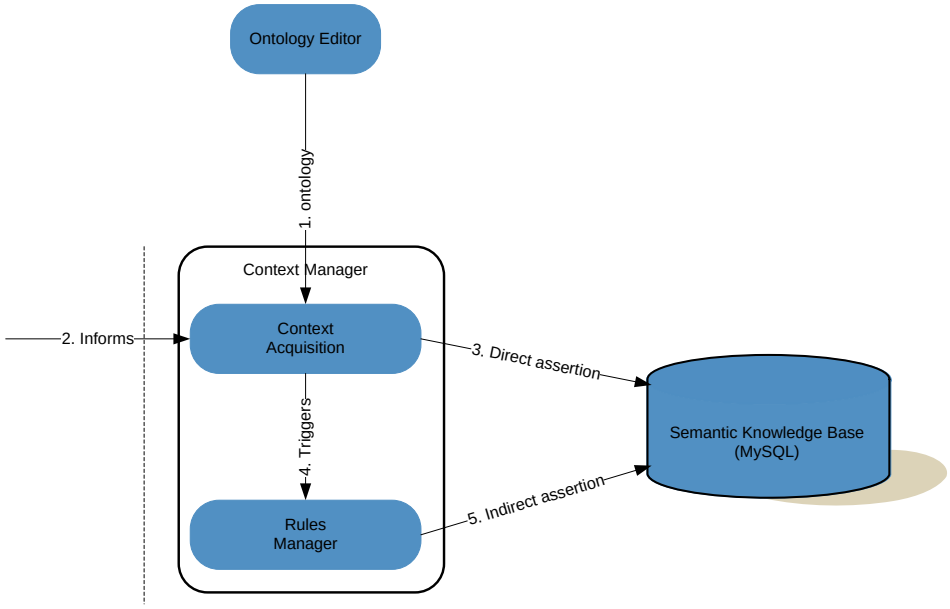


Figure 3.18. Architecture de gestion de contextes

La figure 3.19 montre les composants et le flux des données d'un système de gestion de contexte. Le flux de données est comme suit :

1. L'ontologie de contexte est chargée depuis l'éditeur d'ontologie, qui agit comme un OAP, vers le composant « Context Acquisition ».
2. Les nouvelles instances sont à rentrer à travers le composant « Context Acquisition ».
3. Une assertion directe est effectuée, ou,
4. L'information est passée au moteur de règles « Jess Rule Engine » [Jes08] pour plus de traitements. Ce moteur infère les informations en utilisant les règles SWRL.
5. Les résultats sont saisis (sous la forme Subject-Relation-Object) dans la base sémantique qui est implémentée sous forme d'une base de données MySQL [Mys07].

L'implémentation du système de gestion de contexte fournit ainsi des contextes sémantiques dans la base des connaissances. Notre système de protection XPACML peut désormais se focaliser sur l'utilisation des informations stockées dans la base des données.

3.8 Conclusion

Dans ce chapitre nous avons proposé une extension de notre solution XPACML en nous basant sur l'aspect « contextes ». Cette contribution enrichit XPACML avec des connaissances sémantiques sur les sujets et les ressources. Ces sémantiques permettent de décider de la révélation des données en se basant sur les sémantiques des sujets, et des ressources plutôt que sur leur syntaxe. Ainsi, notre système a l'avantage de comprendre et d'interpréter les requêtes contenant des termes de haut niveau.

Ceci est possible grâce au modèle sémantique unifiant les schémas d'attributs de l'utilisateur et du SP, donnant au final un vocabulaire commun pour les deux. Dès lors, les politiques de protection des données peuvent être analysées à différents niveaux d'abstraction.

Ceci implique une gestion des sémantiques, qui dans notre approche proposée est séparée de la gestion des politiques XPACML sensibles aux contextes. Ceci offre plusieurs avantages :

- La modularité de notre solution, ainsi, toute modification dans la signification des termes du contexte n'a aucun effet sur les politiques XPACML et inversement,
- La réduction du temps d'écriture des politiques, car l'administrateur n'a pas à écrire toutes les instances possibles des contextes comme dans un ancien système XPACML,
- La réduction de la charge d'administration des politiques. En effet, l'administrateur se focalise sur l'écriture de politiques sans se soucier de l'interprétation possible des connaissances inclues.

Il est connu que les contextes changent de façon permanente et non prédictible dans le temps. Tout changement de contexte implique le déclenchement du processus d'acquisition et du système des règles pour acquérir le nouveau contexte situationnel. Les nouveaux contextes doivent remplacer les anciens dans la base de connaissances. Cependant, l'hypothèse de monotonie du RDF (une fois un triplé saisi, il est difficile de le supprimer ou de le changer) utilisé dans nos règles OWL/RDF, notre solution se trouve limitée par la non prise en charge des nouveaux contextes. Les pistes permettant de résoudre ce problème n'ont pas été abordées dans nos travaux de recherche.

TROISIEME PARTIE

La directive européenne 95/46/CE [The95] introduit le concept de « *collecte minimale* », interprété du côté des usagers comme un concept de « *transmission minimale* » des données personnelles (section 1.1.2.2 du chapitre 1). En nous basant sur ce concept, et dans une optique d'implication de l'utilisateur dans le processus de protection de ses données personnelles, nous essayons dans cette partie de donner à l'utilisateur davantage de *contrôle* sur l'usage et la transmission. Plus concrètement, nous proposons des protocoles de négociation portant sur les pratiques d'usage et sur les données à révéler.

Le chapitre 4 traite la dimension associée au *contrôle d'usage*. Nous y proposons un protocole de *négociation des politiques* de protection de données entre l'utilisateur et le SP, en nous basant sur les risques associés aux différents types d'usage proposés par le SP.

Le chapitre 5, traite la dimension liée à la minimisation de la transmission des données, en explorant deux voies :

- l'axe « *justification* » (section 1.3.1 du chapitre 1) pour définir des *opérations de filtrage* sur des combinaisons de données à révéler.

- un protocole de *négociation des combinaisons de données* personnelles à fournir au SP. Nous examinons dans cette partie l'apport de la théorie des jeux dans la conception de ce protocole en nous basant sur la notion de gain et d'utilité de chacun des acteurs impliqués dans la négociation.

CHAPITRE 4

PRISE EN COMPTE DU CONTEXTE SITUATIONNEL DANS LA PROTECTION DES DONNEES PERSONNELLES

Sommaire

Introduction 152
4.1 Problématique 153
4.2 Etat de l'art 153
4.3 Limitations de l'état de l'art 156
4.4 Classification des valeurs de tags P3P 156
4.4.1 Classification des valeurs de l'élément Purpose 156
4.4.2 Classification des valeurs de l'élément Recipient 158
4.4.3 Classification des valeurs de l'élément Retention 159
4.5 Protocole de négociation de politiques à base de classification de valeur des tags P3P 161
4.5.1 Schéma de négociation 161
4.5.2 Création des préférences de l'utilisateur 162
4.5.3 Exemple illustratif 163
4.5.4 Limitations de l'approche 166
4.6 Protocole de négociation de politiques à base de fonction de risque 166
4.6.1 Principe de négociation 166
4.6.2 Vocabulaire de négociation et d'expression des règles 168
4.6.3 Modèles des préférences et des politiques du SP 169
4.6.4 Côté utilisateur 170
4.6.5 Côté SP 172
4.6.6 Protocole de négociation à deux tours 173
4.7 Conclusion 175

4 Négociation des politiques de protection des données

Introduction

Les solutions permettant la comparaison des politiques de protection des données actuelles (section 2.2.3.2 du chapitre 2) s'inscrivent toutes sous le principe de "*take-it-or-leave-it*". Nous avons également montré que notre langage XPACML (section 2.6) permettait la mise en œuvre de ce principe. En effet, comme le montre la section 2.7.4, le PPDP est en mesure de comparer de façon stricte une politique de protection du SP avec les préférences de l'utilisateur.
Le modèle "*take-it-or-leave-it*" qui peut être approprié pour la navigation sur le Web, ne peut l'être pour le cadre commercial dans lequel nous nous plaçons. En effet, un cadre commercial est basé sur deux éléments, les bénéfices de la consommation du service (fondée sur la confiance instaurée entre l'utilisateur et le SP) et la satisfaction de l'utilisateur.

Ainsi, pour apporter plus de flexibilité aux interactions e-commerce, nous nous focalisons dans ce chapitre sur la négociation des politiques de protection des données. Cette négociation permet de générer des contrats de protection plus fins. Plus concrètement, nous proposons deux approches de négociation qui portent sur des données élémentaires. Chaque élément de données à négocier possède une politique associée. L'utilisateur doit définir pour cela ses préférences en termes de protection, ce qui permettra de conduire la négociation avec le SP. Dans nos approches de négociation, nous nous intéressons aux éléments `POLICY` associés à chaque donnée, de type <purpose>, <recipient>, et <retention> (section 2.6.2.1 du chapitre 2).

Pour pouvoir négocier sur les valeurs de ces éléments, il faut passer par une étape de comparaison assurée par le PPDP (section 2.6.3 du chapitre2). Seulement, une comparaison stricte de la sorte, est défavorable vis-à-vis de l'utilisateur dont les transactions échoueront souvent car le SP devra, pour chaque type de données collecté, avoir une politique identique aux préférences de l'utilisateur.

Nous avons donc décidé de classer les différentes valeurs des éléments <purpose>, <recipient>, et <retention> en nous basant sur leurs niveaux de risque. Cette classification offre deux avantages :

1. le SP peut définir une fourchette de valeurs qui pourront être acceptées par l'utilisateur.

2. Deuxièmement, le fait que ce soit l'utilisateur qui définisse la limite de cette fourchette implique que la négociation se déroulera forcément en respectant ses préférences. La section suivante traite le classement de ces valeurs.

Les protocoles de négociation proposés sont sensés augmenter les chances pour trouver une solution aux situations conflictuelles avec le SP. Le premier protocole est basé exclusivement sur la classification précitée des valeurs des sous éléments `POLICY`, alors que le deuxième se base en plus sur une fonction de risque associée à chaque donnée requise par le SP.

Nous présentons d'abord la problématique générale de négociation des politiques de protection des données dans la section 4.1. Ensuite nous présentons les deux grands axes de protocoles de négociation de politiques existants dans la section 4.2 et leurs limitations (4.3). Nous définissons dans la section 4.4 la classification des valeurs de tags P3P nécessaire aux deux approches de négociations présentées dans les sections 4.5 et 4.6 respectivement, avant de conclure avec un bilan de nos contributions (section 4.7).

4.1 Problématique

De part son caractère statique, l'approche "*take it or leave it*" ne peut être applicable au contexte E_Commerce. Pour l'améliorer, il faut prendre en considérations les informations liées à chaque contexte situationnel (section 3.1 du chapitre3) lié à chaque transaction, et la volonté de l'utilisateur et du SP à aboutir à la consommation/fourniture du service en question.

Une volonté qui se traduit par des politiques de protection des données de part et autre. Il y a alors un réel intérêt à définir une négociation de politiques entre l'utilisateur et le SP, afin de produire des contrats de politiques plus fins auxquels les deux parties adhèrent.

A notre connaissance, il n'existe actuellement que deux protocoles de négociation de politiques de protection ([Maa05] et [Thi00]), seulement ils souffrent de deux limitations majeures. La première est le nombre infini de tours de négociation (rounds), rendant ainsi les protocoles inapplicables pour les transactions réelles. La deuxième est le manque de granularité de la négociation des pratiques d'usage pour un ensemble de données. Or, n'ayant pas la même sensibilité, les données personnelles doivent avoir un traitement différent.

Les deux parties ayant des préférences différentes, le protocole doit être équitable en ne permettant à aucune des parties d'inférer des informations sur le modèle de politiques de l'autre.

4.2 Etat de l'art

L'idée de négocier des politiques de protection des données dans une session donnée a été considérée pour la première fois dans les spécifications P3P, pour y être inclues. Cette idée a été rapidement rejetée par les concepteurs de la plateforme P3P, par manque de scénarios où cette possibilité de négociation est utile [Cra02]. Principalement conçue pour simplifier

les interactions entre utilisateurs et SPs, les outils basés sur les spécifications P3P (section 2.2.3.2) sont en faveur du SP au détriment de l'utilisateur en appliquant le concept "*take it or leave it*". Depuis, deux propositions ont été publiées pour les protocoles de négociation de politiques de protection.

La première a été faite par Bennicke et Langendorfer [Ben03]. Dans le protocole proposé (figure 4.1), une négociation est un processus qui aboutit à un contrat initial de politiques. Ce contrat est ensuite modifié de façon incrémentale pour répondre aux besoins des deux parties de la négociation. Ces dernières peuvent avoir des demandes obligatoires (mandatory) ou optionnelles (optional). L'objectif de chacune est que ces demandes obligatoires soient approuvées, et qu'un maximum de ces demandes optionnelles soient également approuvées dans le contrat final. Chaque partie assume alternativement les rôles de « demandeur » et « acceptant/refusant ». Le demandeur propose un contrat qui remplie au moins ses demandes obligatoires, et l'autre partie peut répondre de différentes manières pour affiner ce contrat de politiques jusqu'à ce que les deux parties se mettent d'accord sur une formulation finale, et l'acceptent.

Comme illustré par la figure 4.1, la négociation commence par la partie A qui fait une première proposition de contrat (contract0), avec l'ensemble de ses demandes obligatoires et optionnelles. La partie B compare la demande avec sa propre politique. S'il n'y a aucun conflit, la partie B confirme la proposition et approuve le contrat. Si la partie A a d'autres demandes, elle les envoie à B pour confirmation (contrat 1). Ce mécanisme présente à A deux options : présenter toutes ses demandes à B en une seule fois, ou les présenter une à une. Chacune des deux parties peut signaler sa satisfaction en envoyant un message « finish ». Le processus de négociation se termine quand les parties s'envoient mutuellement un message « finish ».

Il est important de noter, que les auteurs ont introduit des modifications aux spécifications P3P et APPEL [Wor02] pour pouvoir définir ce protocole de négociation.

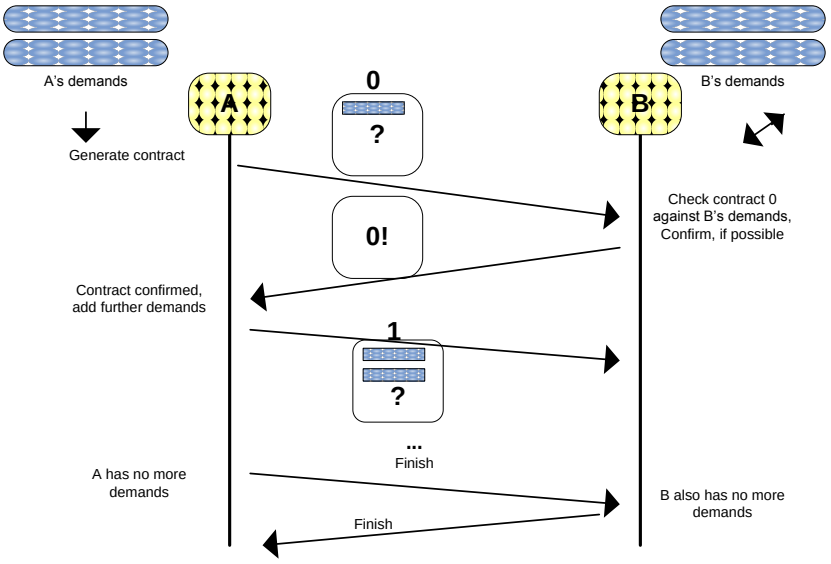


Figure 4.1 Processus de négociation basique

Privacy Server Protocol (PSP) [Thi00] est la deuxième proposition en matière de protocoles de négociation. Il a été conçu pour permettre aux utilisateurs et aux SPs de produire des contrats *mutuels* de politiques de protection. Ces contrats sont mutuels dans le sens où ils engagent l'utilisateur comme le SP dans l'établissement des contrats. Avec cette nature bilatérale, PSP est très similaire à la proposition de Bennicke-Langendorfer. Le protocole PSP exige de l'utilisateur et du SP de s'échanger des propositions et des contrepropositions, jusqu'à ce que l'une des deux parties accepte la dernière proposition reçue. PSP utilise également un modèle de préférence fondé sur des règles en APPEL. La seule différence, est que les auteurs avancent l'hypothèse que les deux parties peuvent révéler des informations sensibles. Ainsi, chaque partie s'engage à respecter la vie privée de l'autre.

En supposant qu'il n'est pas flexible de négocier la politique de protection des données dans sa totalité, dans [Pre06], les auteurs définissent le concept de « dimensions de vie privée ». Ce concept été introduit pour simplifier le processus de négociation. Chaque dimension est une des quatre dimensions génériques introduites par P3P (<purpose>, <recipient>, <retention>, et <data>). Différents niveaux de volonté de l'utilisateur à révéler l'ensemble de ses données sont associés à chaque dimension. Ces niveaux sont déterminés en utilisant une fonction d'utilité [Alt07].

Dans [She08], les auteurs présentent un protocole de négociation entre plusieurs utilisateurs et plusieurs SPs. Pour cela, ils adoptent une fonction de risque proposée par [Tao06]. Cette fonction permet de calculer le risque lié aux données personnelles en se basant sur une catégorisation du schéma de données P3P qui leur est associée. Le risque dérivé est ensuite utilisé avec une variable de remise (discount) dans une fonction d'utilité formulant l'utilité de l'utilisateur à révéler ses données. Une fonction d'utilité est également calculée pour le

SP en se basant sur la vente de produits par rapport aux remises faites. Le processus de négociation entre utilisateurs et SPs se base sur la théorie des jeux pour le calcul des stratégies optimales pour les utilisateurs comme pour les SP.

4.3 Limitations de l'état de l'art

Les protocoles de négociation cités possèdent quelques limitations. Le nombre de tours de négociation est potentiellement infini, ce qui aboutit à des délais de négociation potentiellement importants. De plus, les auteurs n'expliquent pas les modèles des politiques et des préférences proposées par chacun des agents de négociation. A cela, se rajoute un problème d'équité entre les deux parties vu que l'une peut inférer des informations sur le modèle de politique de l'autre, résultant ainsi des concessions qui sont contreproductives pour la négociation.

Pour les travaux basés sur la théorie des jeux, nous notons que la fonction de risque représentée ne prend en considération que le type de donnée mise en jeu. De même, pour les fonctions d'utilité qui se basent sur des variables économiques qui ne sont pas strictement liées à la vie privée de l'utilisateur.

4.4 Classification des valeurs de tags P3P

Nous proposons dans cette section un classement pour chaque élément de l'élément POLICY. Cette classification nous permet d'ordonner les valeurs de chaque sous élément POLICY, afin de pouvoir créer des préférences en termes de protection des données. . Le classement peut différer selon le type du service et le type de données collectées. Nous avons pu introduire dans l'ontologie le classement en fonction du type de données/services. Dans un premier temps, nous avons voulu tester la faisabilité du protocole, ainsi, nous avons fixé le classement précisé ici.

4.4.1 Classification des valeurs de l'élément Purpose

Les valeurs de l'élément <purpose> sont les plus difficiles à classer car le niveau d'importance d'un usage conforme des données dépend fortement du type de données qui est concerné. Il serait ainsi étrange de voir qu'un SP collecte un numéro de téléphone avec <contact /> comme valeur de <purpose>. (Rappelons que la valeur <contact />signifie que la donnée sera utilisée par le SP pour contacter le client en utilisant n'importe quel autre canal, **sauf** le téléphone.)
Un problème se pose alors : comment faire pour changer cette classification suivant le type de données collectées ? La seule réponse à cette question est l'usage des sous-graphes

`DataType` et `Rule` (section 3.5 du chapitre 3). Dans un premier temps, il fallait donc pouvoir classer les valeurs indépendamment du type de données récoltées (figure 4.2).
Afin de pouvoir classer les valeurs de <purpose> du plus strict pour l'utilisateur (c.-à-d. protégeant le mieux ses données) au moins strict, trois paramètres ont été pris en compte : la durée de rétention par défaut que l'intention implique, l'importance de l'intention pour le SP et le(s) risque(s) que peut poser l'intention pour la vie privée de l'utilisateur. Ces trois paramètres ont donné lieu à la classification ci-dessous :

- <current /> **:** se retrouve en haut de cette classification, car la donnée aura une durée de rétention courte. L'intention est critique pour le SP - (c'est pour satisfaire la transaction en cours, comme un envoi de colis par exemple) et elle ne pose pas de problème quant à la vie privée de l'utilisateur (il est sensé savoir quelle transaction il est en train d'effectuer avec le SP).

- <admin /> **/** <dévelop /> **:** ces deux intentions sont critiques du côté SP. Si un SP récupère une donnée avec l'intention <admin /> ou <dévelop />, on peut vraisemblablement supposer que l'utilisateur est un administrateur ou un modérateur du SP.

- <historical /> **:** implique une durée de rétention assez élevée, mais est également très critique pour le SP étant donné que c'est pour répondre à un cadre législatif. L'utilisateur quant à lui, doit de toute façon se conformer à la loi indépendamment du SP.

- <pseudo decision /> **/** <pseudo analysis /> **:** ce ne sont pas forcément des données critiques pour le SP et cela ne pose aucun problème pour la vie privée de l'utilisateur, étant donné que le SP ne collecte aucune donnée identifiée. Nous avons considéré qu'une analyse était un peu plus intrusive dans la navigation d'un utilisateur qu'une simple décision.

- <tailoring /> **:** cette intention implique une durée de rétention courte car les données sont collectées pour un ajustement « ponctuel ». Il n'est pas fait mention dans la spécification P3P d'une collecte des données identifiées ou non, donc, <tailoring /> peut collecter des données identifiées.

- <individual decision /> / < individual analysis />**:** ces intentions récoltent forcément des données identifiées. Elles peuvent donc poser problème pour la vie privée de l'utilisateur. Ensuit, c'est la même réflexion que pour <pseudo decision /> / < pseudo analysis />leurs homologues quant au pseudo : les données collectées ne seont pas critiques pour le SP, mais cela l'aidera à améliorer son site.

- <contact /> **:** dans la spécification P3P, il est clairement écrit que l'intention <contact /> signifie que la donnée sera collectée pour envoyer aux visiteurs de la publicité. Les données collectées ne sont donc pas critiques pour le SP. La durée de rétention peut être indéterminée (puisque c'est souvent à l'utilisateur de choisir de ne plus recevoir de publicités), et cela peut poser problème pour la vie privée

de l'utilisateur puisque le SP aura un moyen de le contacter, voire de savoir où il habite.

- <telemarketing /> **:** c'est la même réflexion que pour la valeur <contact />, sauf que seul le numéro de téléphone doit être collecté. Nous avons considéré qu'il était souvent plus ennuyeux, pour un utilisateur, d'être contacté pour de la publicité par téléphone.

- <other purpose /> **:** cette intention peut vraiment être tout et n'importe quoi. Elle se situe donc en fin de classement.

1	2	3	4
Current	Admin	Develop	Historical

5	6	7	8
Pseudo-decision	Pseudo-analysis	Tailoring	Individual-analysis

9	10	11	12
Individual-analysis	Contact	telemarketing	Other purpose

Figure 4.2 Classification des valeurs du tag Purpose

4.4.2 Classification des valeurs de l'élément Recipient

Les valeurs de l'élément <recipient> dépendent très peu du type de donnée collectée, elles dépendent surtout de l'intention déclarée. Par exemple, il serait étrange de voir une donnée récoltée pour envoyer un colis à l'utilisateur et la valeur <public /> comme destinataire. Là encore, l'ontologie présentée dans la section 3.5 du chapitre 3 pourrait servir à changer la classification effectuée pour l'adapter à l'intention déclarée. Dans nos travaux nous nous sommes contentés de poser la structure sémantique générale du sous graphe `Rule` et le lier aux deux sous graphes `DATA_TYPE` et `SERVICE_TYPE`. Focalisés sur la négociation, nous n'avons pas exploré toutes les liaisons sémantiques entre les différents types de données et les intentions, ainsi que les valeurs des sous éléments de `POLICY`.

Toutefois, il n'était pas compliqué de faire abstraction de cette relation pour créer une classification facile à utiliser en Java. Deux paramètres ont été utilisés pour créer cette classification (figure 4.3) : le nombre d'entités morales que le type de destinataire implique, et le fait que ces entités morales respectent une politique de protection similaire à celle du SP ou non. Ces deux paramètres ont donné lieu à la classification suivante :

- <ours /> **:** il concerne le SP et les agents pour lesquels il travaille et/ou les agents qui travaillent pour lui. Ces destinataires ont forcément la même politique de protection que celle du SP et ne concerne qu'un nombre relativement faible d'entités morales.

- <same/> **:** ces destinataires ont forcément une politique de protection équivalente à celle du SP. En revanche, cela peut concerner un plus grand nombre d'entités que la valeur <ours />.

- <other-recipient/> **:** cela peut concerner un grand nombre d'entités dont les pratiques ne sont pas forcément celles du SP mais sont tout de même connues du SP. De plus, le SP a un droit de regard sur l'usage des données récupérées par ces destinataires. En effet, un mauvais usage de celles-ci pourrait nuire à ses intérêts ainsi qu'à ceux de l'utilisateur.

- <delivery /> **:** concerne un faible nombre d'entités (personnes morales opérant un service de livraison) mais ce sont des entités pouvant utiliser les données pour d'autres intentions que celles déclarées, voire dont les pratiques sont totalement inconnues et qui ne sont pas sous contrat avec le SP. Un tel destinataire pourrait poser problème quant à la confidentialité des données fournies par un utilisateur.

- <unrelated/> **:** ce sont des destinataires dont les pratiques sont inconnues du SP. Cela peut concerner un très grand nombre d'entités morales, d'où sa place dans la classification.

- <public /> **:** une donnée récoltée ayant pour valeur de recipient <public /> signifie que n'importe qui pourra y avoir accès.

1	2	3	4	5	6
Ours	Same	Other-recipient	Delivery	Unrelated	Public

Figure 4.3 Classification des valeurs du tag Recipient

4.4.3 Classification des valeurs de l'élément Retention

La durée de rétention dépend de l'intention déclarée mais également des destinataires. En effet, la durée de rétention pour une donnée collectée dont l'intention est définie à <current /> ne sera pas la même que celle pour une donnée dont l'intention est définie à <contact />.
De même, la durée de rétention pourra varier selon les destinataires : elle ne sera pas la même si le destinataire est de type <delivery /> ou s'il est de type <public />.

La classification pour la durée de rétention fut la plus simple à mettre en place, en effet, nous n'avons fait que prendre en compte la durée de rétention maximale que la valeur impliquait.

- <no retention /> **:** c'est la durée de rétention la plus courte : la donnée est totalement supprimée dès que la transaction en cours est terminée.

- <stated purpose /> **:** dépend fortement de l'intention déclarée, mais les SPs (selon la spécification P3P) doivent avoir une politique de rétention établissant un calendrier de destruction des données collectées avec une telle valeur de rétention, qui peut être connue. Cette durée est utilisée pour satisfaire l'intention déclarée. Elle est donc fortement utile du côté SP.

- <legal requirement /> **:** de même que pour <stated purpose />, les SPs doivent avoir un calendrier de destruction des données. Comme écrit dans la spécification P3P : « *les renseignements sont conservés pour satisfaire à une intention déclarée, mais la période de rétention est plus importante du fait d'une obligation légale ou d'une responsabilité légale.* »

- <business practices /> **:** suivant les pratiques commerciales du SP. Cela peut donc être n'importe quelle durée de rétention, définie dans un calendrier de destruction établi dans une politique de rétention spécifique. Comme la durée maximale de rétention dépend fortement du SP , cette valeur se retrouve en avant dernière place de la classification.

- <indefinitely /> **:** la durée de rétention la plus haute, à savoir « indéterminée ». C'est celle qui peut être appliquée lorsque le destinataire est une tribune publique, par exemple.

1	2	3	4	5
No retention	Stated purpose	Legal requirements	Business practices	Indefinitely

Figure 4.4 Classification des valeurs du tag Retention

4.5 Protocole de négociation de politiques à base de classification de valeur des tags P3P

4.5.1 Schéma de négociation

La négociation que nous avons proposée est un processus itératif. Une itération consiste en deux étapes. Dans la première étape, la proposition de politique faite pas le SP est comparée aux préférences de l'utilisateur, les termes acceptés et ceux rejetés sont alors déterminés. Cette étape est supportée par la classification détaillée dans la section 4.4.

Dans la seconde étape, une réponse appropriée est déterminée par l'agent utilisateur selon une stratégie de négociation que nous avons définie.

Cette stratégie de négociation est basée sur différentes préférences et politiques organisées dans un ordre de préférence de l'utilisateur et du SP. Du côté de l'utilisateur, ses préférences sont organisées des plus strictes aux plus larges. Du côté du SP, les politiques sont organisées inversement (des plus larges aux plus strictes).

La négociation est effectuée exclusivement du côté de l'utilisateur. Quand ce dernier initie une transaction, le SP lui fournit une liste de politiques auxquelles il adhère dans un ordre de préférence (de la politique la à la plus stricte). Ensuite, l'agent de négociation côté utilisateur commence la négociation des politiques, comme illustré dans la figure 4.5. La négociation des termes des politiques a lieu dans un premier temps entre la politique la plus large du SP et les préférences les plus strictes de l'utilisateur. Si les deux politiques sont équivalentes la négociation aboutit, sinon la politique suivante préférée de l'utilisateur est utilisée. Ce processus se poursuit jusqu'à épuisement des préférences de l'utilisateur (ou qu'une équivalence des politiques ait lieu). Dans le cas où il n'y a aucune équivalence, si le SP fournit à l'utilisateur une autre politique de protection, cette dernière est utilisée dans un nouveau tour de négociation comme expliqué précédemment. Ainsi, il y aura autant de tours, qu'il y a de politiques proposées par le SP.

Dans le pire cas, le dernier tour de négociation se joue entre la politique du SP la plus stricte, et les préférences les plus larges de l'utilisateur. S'il n'y a pas de conflits, la transaction peut avoir lieu, sinon, l'utilisateur est sollicité pour donner son consentement pour les termes incompatibles entre les deux politiques. Suite au consentement donné (ou refusé) de l'utilisateur, la transaction peut avoir lieu ou pas.

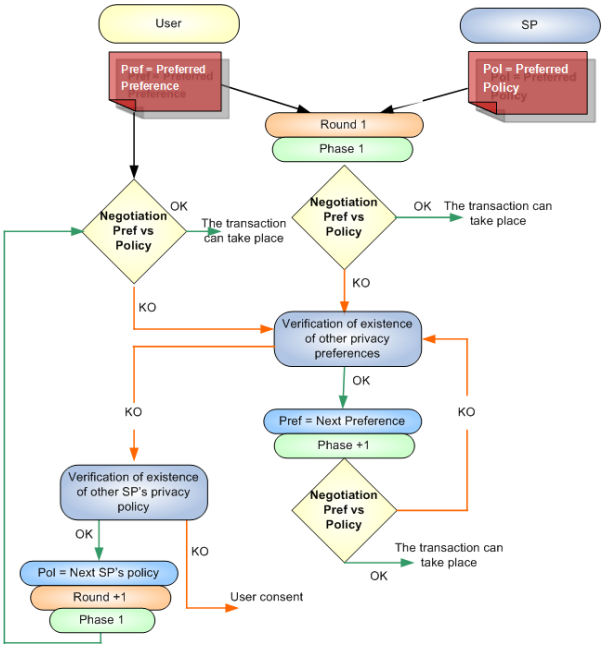


Figure 4.5 Schéma de négociation de politiques - classification de tags P3P

4.5.2 Création des préférences de l'utilisateur

Nous avons développé une application utilisateur d'aide à la définition de ses préférences. Elle se présente sous forme d'une interface qui demande en premier lieu à l'utilisateur de sélectionner un type de service (figure 4.6). Puis, elle propose à l'utilisateur de définir ses préférences pour ce type de services là et pour chacune de ses données personnelles.
Ainsi, et comme illustré dans la figure 4.7, l'utilisateur doit positionner un curseur pour chacun des sous éléments <purpose>, <recipient>, et <retention> en accord avec la classification présentée dans la section 4.4 par chacun de ses éléments de données.

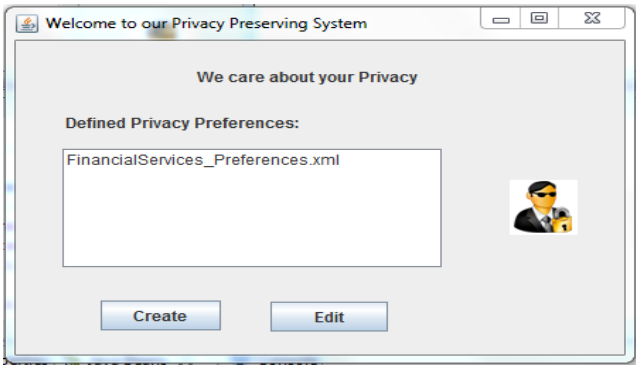


Figure 4.6 Première interface utilisateur pour créer et mettre à jour ses préférences pour un type de service spécifique

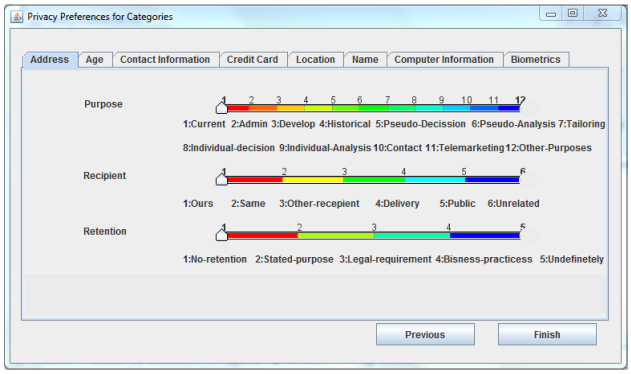


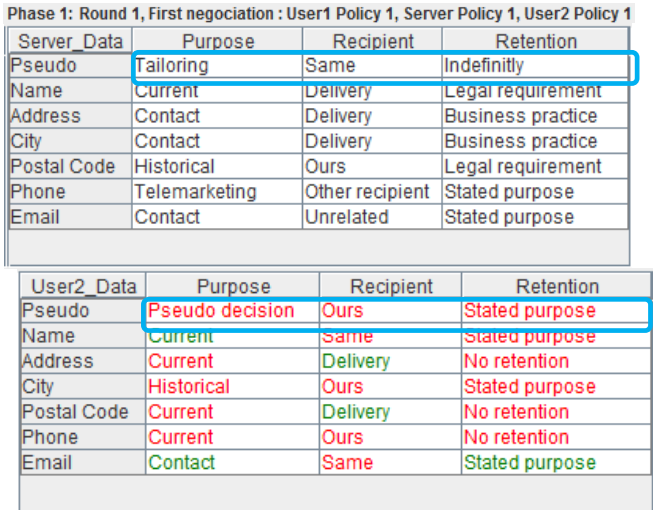
Figure 4.7 Définition des préférences utilisateur pour la donnée « *address* »

L'application stocke les préférences résultantes de l'utilisateur sous forme d'une politique XPACML.

4.5.3 Exemple illustratif

Les figures de 4.8 à 4.11 illustrent le schéma de négociation complet quand deux politiques de protection sont proposées par le SP, et deux politiques sont configurées par l'utilisateur. Le premier tour compare la première préférence de l'utilisateur (figure 4.8) avec la première politique du SP. Une incompatibilité claire est détecter pour la valeur pseudo comme le montre la figure 4.8. En effet, la valeur <Tailoring/> (requise pour l'attribut pseudo par le SP) est inférieure à la valeur <Pseudo decision/>" (définie dans les

préférences de l'utilisateur pour l'élément pseudo) selon la classification de l'élément <purpose>.
La valeur <Same/>" est inférieure à <Ours/> selon la classification des valeurs <recipient>. Enfin, la valeur < Indefinitly/> est aussi inférieure à la valeur <Stated Purpose/> selon la classification de <retention>.



Phase 1: Round 1, First negociation : User1 Policy 1, Server Policy 1, User2 Policy 1

Server_Data	Purpose	Recipient	Retention
Pseudo	Tailoring	Same	Indefinitly
Name	Current	Delivery	Legal requirement
Address	Contact	Delivery	Business practice
City	Contact	Delivery	Business practice
Postal Code	Historical	Ours	Legal requirement
Phone	Telemarketing	Other recipient	Stated purpose
Email	Contact	Unrelated	Stated purpose

User2_Data	Purpose	Recipient	Retention
Pseudo	Pseudo decision	Ours	Stated purpose
Name	Current	Same	Stated purpose
Address	Current	Delivery	No retention
City	Historical	Ours	Stated purpose
Postal Code	Current	Delivery	No retention
Phone	Current	Ours	No retention
Email	Contact	Same	Stated purpose

Figure 4.8 Premières préférences (tableau du bas) vs première politique SP (tableau du haut)

Négociation –Tour 1-Phase 1

Notre agent de négociation cherche ensuite la préférence suivante de l'utilisateur pour compléter la négociation avec la première politique préférée du SP.
Ainsi, une deuxième négociation démarre pour le même tour de négociation. Malgré l'équivalence de plusieurs valeurs, comme illustré dans la figure 4.9, la deuxième et la dernière préférence de l'utilisateur ne solutionne pas les conflits de politiques avec le SP. Comme l'utilisateur n'a plus de préférences définies, un deuxième tour de négociation démarre avec la politique suivante du SP. La comparaison de cette dernière avec les premières préférences de l'utilisateur donne toujours lieu à des incompatibilités, ainsi, la préférence suivante de l'utilisateur est testée. Trois incompatibilités restent (figure 4.11), et comme les politiques du SP sont épuisées, tout comme les préférences de l'utilisateur, le consentement de l'utilisateur est alors requit pour pouvoir continuer la négociation ou y mettre fin.

Phase 2: Round 1, Second negociation : Server Policy 1, User2 Policy 2

Server_Data	Purpose	Recipient	Retention
Pseudo	Tailoring	Same	Indefinitly
Name	Current	Delivery	Legal requirement
Address	Contact	Delivery	Business practice
City	Contact	Delivery	Business practice
Postal Code	Historical	Ours	Legal requirement
Phone	Telemarketing	Other recipient	Stated purpose
Email	Contact	Unrelated	Stated purpose

User2_Data	Purpose	Recipient	Retention
Pseudo	Tailoring	Public	Indefinitly
Name	Current	Same	Stated purpose
Address	Current	Delivery	No retention
City	Historical	Ours	Stated purpose
Postal Code	Current	Delivery	No retention
Phone	Telemarketing	Same	Stated purpose
Email	Other purposes	Public	Indefinitly

Figure 4.9 Deuxièmes préférences vs première politique SP

Négociation –Tour1-Phase 2

Phase 3: Round 2, Third negociation : Server Policy 2, User2 Policy 1

Server_Data	Purpose	Recipient	Retention
Pseudo	Tailoring	Same	Indefinitly
Name	Current	Delivery	Legal requirement
Address	Current	Delivery	No retention
City	Historical	Ours	Legal requirement
Postal Code	Current	Delivery	No retention
Phone	Telemarketing	Same	Stated purpose
Email	Contact	Unrelated	Stated purpose

User2_Data	Purpose	Recipient	Retention
Pseudo	Pseudo decision	Ours	Stated purpose
Name	Current	Same	Stated purpose
Address	Current	Delivery	No retention
City	Historical	Ours	Stated purpose
Postal Code	Current	Delivery	No retention
Phone	Current	Ours	No retention
Email	Contact	Same	Stated purpose

Figure 4.10 Premières préférences vs deuxième politique SP

Négociation –Tour2-Phase 1

User2_Data	Purpose	Recipient	Retention
Pseudo	Tailoring	Public	Indefinitly
Name	Current	**Same**	**Stated purpo...**
Address	Current	Delivery	No retention
City	Historical	Ours	**Stated purpo...**
Postal Code	Current	Delivery	No retention
Phone	Telemarketing	Same	Stated purpose
Email	Other purposes	Public	Indefinitly

Figure 4.11 Secondes préférences vs deuxième politique SP

Avec trois incompatibilités restantes

4.5.4 Limitations de l'approche

Le protocole de négociation que nous proposons possède un nombre de tours fini dépendant du nombre de préférences de l'utilisateur et du SP. Cependant l'utilisateur a accès aux informations de politiques du SP, et ainsi, il a la possibilité d'ajuster ses préférences pour être le plus gagnant possible dans le processus de négociation. L'équité n'est donc pas vérifiée.

Aussi, malgré le caractère intuitif de l'interface utilisateur et la simplicité relative du schéma de négociation, il est assez compliqué pour l'utilisateur, comme pour le SP, de définir un ensemble de préférences pour chaque type de donnée et chaque catégorie de SPs/utilisateur. Le modèle de préférence permet à l'utilisateur la définition de deux sous ensemble de préférences « acceptables » et « inacceptables ». Ainsi, les possibilités de négociation sont restreintes. Aussi, malgré sa simplicité, le schéma de négociation présente un nombre de tours assez conséquent, pour chaque donnée requise. En effet, dans un scénario concret, le SP demande une multitude de données simultanément, chacune donnant lieu à une négociation locale dans l'agent de négociation de l'utilisateur. D'une part, le temps de négociation peut être conséquent. D'autre part, l'envoi de plusieurs politiques du SP pour chaque donnée est consommateur de ressources réseau.

4.6 Protocole de négociation de politiques à base de fonction de risque

4.6.1 Principe de négociation

Avant que toute transaction n'ait lieu, le SP est sollicité pour définir une politique de protection où il définit ses besoins en termes de protection de sa vie privée. Cette politique de protection telle exprimée par le SP, lui permet de distinguer les éléments obligatoires de ceux optionnels. Nous utilisons le langage XPACML pour pouvoir stocker et échanger les politiques.

Par le biais de l'interface illustrée dans la figure 4.16, l'utilisateur spécifie ses préférences en termes de protection des données. En effet, pour chaque type de service, et pour chacune de ses données, l'utilisateur définit (grâce à la classification présentée dans la section 4.4) les valeurs acceptables, inacceptables ou indéfinies des éléments <purpose>, <recipient>, et <retention>. Ainsi, trois sous ensembles de politiques (section 4.6.4.1) sont définis pour chaque donnée, permettant d'introduire une flexibilité dans le processus de négociation.

Comme illustré à la figure 4.12, la négociation de politiques se fait en deux tours (Round) maximum pour chaque session. Dans chaque tour, une partie fait une proposition, l'autre partie peut l'accepter, la rejeter, ou faire une contre proposition. Dans le premier tour, la proposition est émise par le SP sous la forme de sa politique initiale (Message2). Il est à noter que cette politique exige à la fois les éléments optionnels comme les éléments obligatoires. Ainsi, l'utilisateur n'a pas la possibilité de rejeter les éléments optionnels de façon systématique. Si les préférences de l'utilisateur sont satisfaites, cette politique est acceptée, autrement, l'agent utilisateur établit une politique qui comprend l'ensemble idéal de ses préférences (Message3).

Dans le deuxième tour, le SP peut accepter directement la contre proposition de l'utilisateur, ou former une seconde proposition pour l'utilisateur (Message4). Cette deuxième proposition comprend les éléments obligatoires et/ou optionnels de la première politique du SP qui étaient déjà acceptés par l'utilisateur dans le premier tour de négociation, et les éléments obligatoires qui ont été rejetés. Pour chaque élément de politique obligatoire qui a été rejeté, l'utilisateur calcule une valeur de risque. Si la valeur calculée du risque est plus élevé qu'un certain seuil fixé par l'utilisateur, l'agent utilisateur rejette la dernière proposition de politique du SP(Message5). Ainsi, la négociation échoue. Sinon, la deuxième proposition du SP est acceptée (Message5). Si une des deux parties accepte la proposition à n'importe quel point de la négociation, la négociation réussit et se termine.

Les détails du processus de négociation sont donnés dans la section 4.6.6.

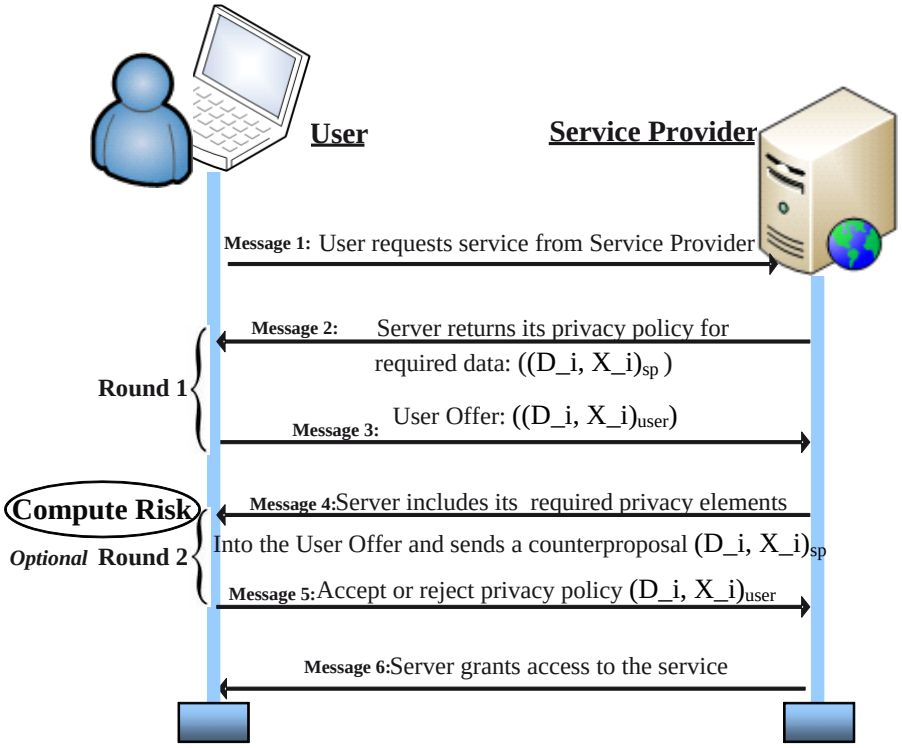


Figure 4.12 Négociation de politiques à base de fonction de risque

Il est à noter que la négociation se passe sur des fragments de politiques que nous appelons règles (section 2.6.2.2 du chapitre2). Ainsi, une négociation d'une politique est un ensemble de négociations de règles simultanées. Plus précisément, chaque règle associée à un élément de donnée est composée de plusieurs paires (élément de donnée, tag de l'élément `POLICY` <purpose>, <recipient>, et <retention>). De ce fait, pour chaque donnée, trois négociations élémentaires doivent avoir lieu (une pour l'élément <purpose>, l'élément <recipient>, et l'élément <retention>). Les modèles de politiques du SP et le modèle des préférences de l'utilisateur sont expliqués dans les sections 4.6.4.1 et 4.6.5.1. Si toutes les négociations unitaires réussissent, les résultats sont alors combinés pour produire un contrat de politique final. Si une des négociations élémentaires échoue, la négociation globale échoue également.

4.6.2 Vocabulaire de négociation et d'expression des règles

La figure 4.13 montre un exemple de règle exprimant les préférences de l'utilisateur définies pour un type de données. Dans cet exemple, les éléments principaux présents sont ("Rule", "Effect", "Resource", " Target "), avec les tags de pratiques d'usage associés <purpose>, <recipient>, et <retention>.

La règle présentée spécifie les préférences de l'utilisateur pour la donnée "*address*". L'utilisateur accepte de révéler son adresse seulement pour le traitement de la transaction en cours <current />, pour accomplir le support système <admin />, et évaluer/mettre à jour le service/site/produit <develop />. Seuls le SP (<ours />), et les entités ayant les mêmes pratiques de protection <same />, et les services de livraison <delivery />, sont autorisés à accéder à la donnée, et seulement pour la période déclarée suffisante pour compléter les intentions déclarées <stated-purpose />.

Il est important de noter que la négociation porte seulement sur les types de données, et non sur les valeurs réelles de l'utilisateur (exemple : donnée *Address,* et pas l'adresse elle-même).

Rappelons que le langage XPACML a été défini entre autres pour supporter la négociation de politiques entre l'utilisateur et le SP. Ainsi, pour exprimer les propositions et les contre propositions du premier et deuxième tours, l'élément "Effect" de la structure de politique XPACML a été défini selon les valeurs `Permit`, `Deny`, et `Offer_Policy` (section 2.6.2.2) pour une règle spécifique associée à une donnée.

```
<xpacml:Rule Effect="Permit" CategoryId="Address_Category" ApplyToDescendant="Yes">

<xpacml:Target>
  <xpacml:Resources>
    <xpacml:Resource ResourceId="Address_Category" DataComposition="Composed">
        <p3p:PURPOSE>
         <p3p:current/>
          <p3p:admin/>
          <p3p:develop/>
        </p3p:PURPOSE>
        <p3p:RECIPIENT>
          <p3p:ours/>
          <p3p:same/>
          <p3p:delivery/>
        </p3p:RECIPIENT>
        <p3p:RETENTION>
          <p3p:stated-purpose/>
        </p3p:RETENTION>
```

Figure 4.13 Une règle dans les préférences utilisateur sous format XPACML

4.6.3 Modèles des préférences et des politiques du SP

Trois ensembles disjoints de tags sont définis et exprimé avec les tags P3P : RecipientTags, RetentionTags, et PurposeTags.

Formellement, chaque règle R_i est un triplet de la forme R_i = (D_i,Rec_i,Ret_i,Pur_i,Eff), où D_i ∈ DataType, Rec_i ⊆ RecipientTags, Ret_i ⊆RetentionTags, Pur_i ⊆ PurposeTags, et Eff représente l'élément Effect associé au vocabulaire de la négociation.

La règle R_i peut encore être décomposée en termes T_i = (D_i,X_i), où X_i est un tag parmi Rec_i, Ret_i, ou Pur_i. Ceci dit, chaque règle consiste en trois termes, chacun pour chaque tag d'usage, et une valeur « Effect ».

Chaque négociation de politique peut être vue comme un ensemble de négociations synchronisées, une pour chaque terme T_i de la politique ; ce niveau de négociation atomique est nécessaire, car chacun des tags d'usage est appliqué sur chaque donnée avec leur propre sémantique et leur propre classement. De ce fait, si une négociation atomique n'aboutit, toute la négociation échoue. Il est à noter que les décisions, les propositions et les contre propositions faites durant une négociation élémentaire ne peuvent affecter les autres. Ainsi dans le reste des sections, nous nous focalisons sur une négociation élémentaire avec des termes individuels T_i.

4.6.4 Côté utilisateur

Pour supporter le schéma de négociation, l'utilisateur doit être capable de comparer la politique du SP avec ses préférences dans le premier tour de négociation, et prendre une décision rapide des termes acceptés et ceux rejetés. Nous nous basons sur la classification présentée dans la section 4.4 pour déduire le modèle de préférences de l'utilisateur. Nous définissons également une fonction de risque qui estime la valeur de risque pour chaque terme T_i de la politique SP proposée.

4.6.4.1 Modèle de préférences de l'utilisateur

En se basant sur la classification présentée dans la section 4.4, l'utilisateur doit définir deux limites pour chaque terme T_i= (D_i, X_i), et pour chaque type de service, subdivisant les pratiques X_i associées en trois sous ensembles comme illustré dans la figure 4.14. Les sous ensembles suivants expriment la tolérance de l'utilisateur au risque quant à sa vie privée :

Ideal_i: ce sous ensemble du terme (D_i, X_i) délimité par la limite "Id_i", contient les premières valeurs des pratiques selon la classification, que l'utilisateur considère comme acceptables pour lui par rapport au respect de sa vie privée.

Unacc_i (pour Unacceptable): ce sous ensemble du terme (D_i, X_i) délimité par la limite "In_i" contient les valeurs qui sont inacceptables par l'utilisateur.

Nego_i (pour Negotiable): ce sous ensemble du terme (D_i, X_i) délimité par les limites "Id_i" et "In_i" contient les valeurs des pratiques d'usage que l'utilisateur ne souhaite pas mettre dans ses préférences, mais qui ne sont pas rejetés automatiquement. Ainsi, ces valeurs sont utilisées durant le processus de négociation, et la décision d'accepter ou de refuser ces valeurs est déterminée selon une fonction de risque et une valeur seuil du risque (section 4.6.4.2). Ce sous ensemble contient ainsi des valeurs qui sont relativement tolérables si elles sont inévitables.

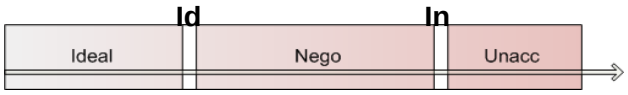


Figure 4.1429 Sous ensembles des risques de tolérance de l'utilisateur

La figure 4.15 illustre l'interface utilisateur pour définir ses sous ensembles Ideal_i, Nego_i et Unacc_i pour son adresse, et pour un type de service spécifique.
Cette interface est implémentée avec un composant bislider [Bis] qui permet la définition de deux sous ensembles (Ideal_i et Unacc_i), et ainsi déterminer le sous ensemble Nego_i.

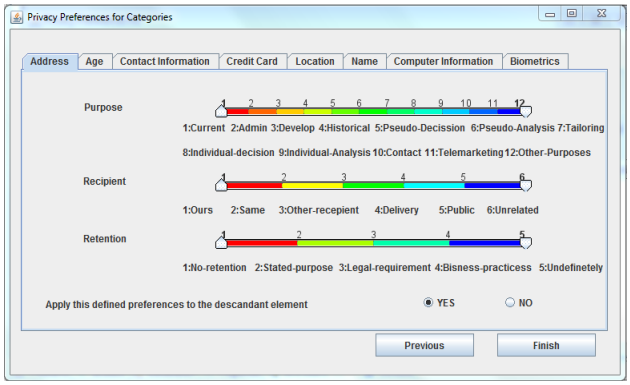


Figure 4.15 Interface utilisateur pour la spécification des limites acceptables et inacceptables de politiques associées aux données personnelles

4.6.4.2 Fonction de risque

La fonction de risque permet de calculer le risque de révélation d'un terme spécifique T_i= (D_i, X_i) où X_i appartient au sous ensemble Nego_i. Il prend en considération le niveau de sensibilité de la donnée D_i, mais aussi le niveau de risque du tag X_i associé. Pour une valeur donnée de X_i, le niveau de risque est estimé en se basant sur la distance entre les tags de pratiques requises par le SP et les limites Ideal_i / Unacc_i. Cette distance est évaluée grâce à la classification présentée dans la section 4.4 : plus le tag est proche de la limite Unacc_i, plus le risque est élevé ; plus le tag est proche de la limite Ideal_i, moins le risque est élevé.

Par conséquent, la fonction de risque a l'expression donnée dans la formule 4.1 avec les paramètres suivants :

- S_i: le niveau de sensibilité de l'élément de donnée D_i variant sur une échelle du moins sensible au plus sensible de 0 à 1.

- X_i: la valeur du tag de protection (figure 4.16) requis par le SP pour D_i selon notre classification des tags.
- Id_i: le numéro associé à la limite "Id_i" défini par l'utilisateur pour D_i et le tag de type X_i.
- In_i: le numéro associé à la limite "In_i" définie par l'utilisateur pour D_i et le tag de type X_i.

$$R(D_i, X_i) = \left[S_i + \frac{X_i - Id_i}{In_i - Id_i - 1}\right] \times \frac{1}{2}$$

Formule.4.1 Fonction d'estimation du risque pour le terme T_i = (D_i,X_i)

Il est à noter que la fonction de risque suppose que l'utilisateur a déjà défini un seuil reflétant son seuil de protection acceptable.

4.6.4.3 Stratégie de négociation de l'utilisateur

Durant les deux tours de négociation, pour chaque terme $T_i = (D_i, X_i)_{sp}$ requis par le SP, l'utilisateur se réfère aux trois sous ensembles Ideal_i, Nego_i et Unacc_i et les limites In_i, Id_i, et leur applique les exigences suivantes:

In Round 1

Req 1: si $(D_i, X_i)_{sp} \subseteq$ Ideal_i, l'agent utilisateur accepte la proposition $(D_i, X_i)_{sp}$.

Req 2: si $(D_i, X_i)_{sp} \subseteq$ Unacc_i ¿ Nego_i, l'agent utilisateur rejette la proposition $(D_i, X_i)_{sp}$.

In Round 2

Req 3: si $(D_i, X_i)_{sp} \subseteq$ Unacc_i, l'agent utilisateur rejette la proposition $(D_i, X_i)_{sp}$.

Req 4: si $(D_i, X_i)_{sp} \subseteq$ Nego_i, l'agent utilisateur calcule la valeur de risque R $(D_i, X_i)_{sp}$. si R $(D_i, X_i)_{sp} \leq$ Threshold-$Risk_{user}$ (valeur constante entre 0 et 1, fixée par l'utilisateur, et représentant le seuil de risque accepté par ce dernier) l'agent utilisateur accepte la nouvelle offre $(D_i, X_i)_{sp}$. Sinon, l'agent utilisateur rejette l'offre $(D_i, X_i)_{sp}$.

Req 5: si $(D_i, X_i)_{sp} \subseteq$ Ideal_i, l'agent utilisateur accepte la nouvelle offre $(D_i, X_i)_{sp}$.

4.6.5 Côté SP

4.6.5.1 Modèle de politique du SP

Le modèle de préférences du SP est plus simple que celui de l'utilisateur pour plusieurs raisons. Premièrement l'utilisateur utilise le Web pour un large panel d'activités : recherche, web-mail, achats, … Les SPs quant à eux exécutent un nombre plus réduit de fonctions, qui sont gérées par un modèle économique donné. Ainsi, les politiques du SP

sont plus statiques que les préférences de l'utilisateur. Dans notre cas, nous supposons qu'elles sont définies par un administrateur de sécurité au niveau du SP.
Le modèle du SP définit deux sous ensembles de termes :

Required_i: à cause des contraintes fonctionnelles du modèle business, quelques éléments (D_i) doivent être délivrés par l'utilisateur dans les conditions X_i, sinon la négociation échoue. Ainsi, *Required_i* = {(D_i, X_i) / (D_i, X_i) est requis par le SP}. Requis est à interpréter comme "Mandatory" en anglais.

Optional_i: le SP a intérêt à obtenir les termes (D_i, X_i) de la part de l'utilisateur. Cependant la négociation reste valable si l'utilisateur ne les délivre pas. Ainsi, *Optional_i* = {(D_i, X_i) / (D_i, X_i) est optionnellement requise par le SP}

4.6.5.2 Stratégie de négociation du SP

Durant les deux tours de négociation, pour chaque terme (D_i, X_i), le SP se réfère à ses sous ensembles Required_i et Optional_i, et applique les règles suivantes :

In Round 1

Req 1: le SP envoie sa propre politique $(D\_i, X\_i)_{sp}$, où $(D\_i, X\_i)_{sp} \subseteq$ Required_i ¿ Optional_i et demande aussi les deux sous ensembles Req_i et Opt_i.

In Round 2

Req 2: la réussite finale de la négociation dépend du succès de toutes les négociations atomiques (D_i, X_i). Ainsi, les règles suivantes s'appliquent :

- si $\forall$ i, Required_i $\subseteq (D\_i,X\_i)_{user}$, le SP accepte l'offre utilisateur.
- Sinon, le SP établit une contreproposition comme suit:

Counterproposal = $\{(D\_i, X\_i)_{user} / (D\_i, X\_i)_{user} \subseteq$ Required_i$\}$ ¿ $\{(D\_i, X\_i)_{user} / (D\_i, X\_i)_{user} \subseteq$ Optional_i $\}$

4.6.6 Protocole de négociation à deux tours

Cette section donne le détail du protocole de négociation complet qui se passe en deux tours maximum. Le protocole a été implémenté en utilisant des sockets Java et notre langage XPACML.

4.6.6.1 Tour 1

Au début de chaque transaction, l'utilisateur initie le premier tour en se connectant au SP pour obtenir la politique du SP (Message 1 de la Figure 4.12). Le SP répond dans Message 2 avec sa propre politique de protection où l'ensemble des termes optionnels et obligatoires sont demandés. Pour un terme (D_i, X_i)sp, si l'utilisateur accepte le tag de pratique d'usage par défaut X_i, il envoie son accord en utilisant la valeur "Permit" de l'élément (Eff) de la règle, sinon, il envoie ses préférences idéales $(D\_i, X\_i)_{user} \subseteq$

Ideal_i (voir le détail dans la section 4.6.4.1) avec la valeur "*Offer_Policy*" dans l'élément (Eff) de la règle. Toutes les règles sont concaténées pour former un message « Message 3 ». La figure 4.16 illustre la génération d'une offre utilisateur incluant l'ensemble Ideal de ses préférences pour la donnée "*address*". Ensuite le deuxième tour commence.

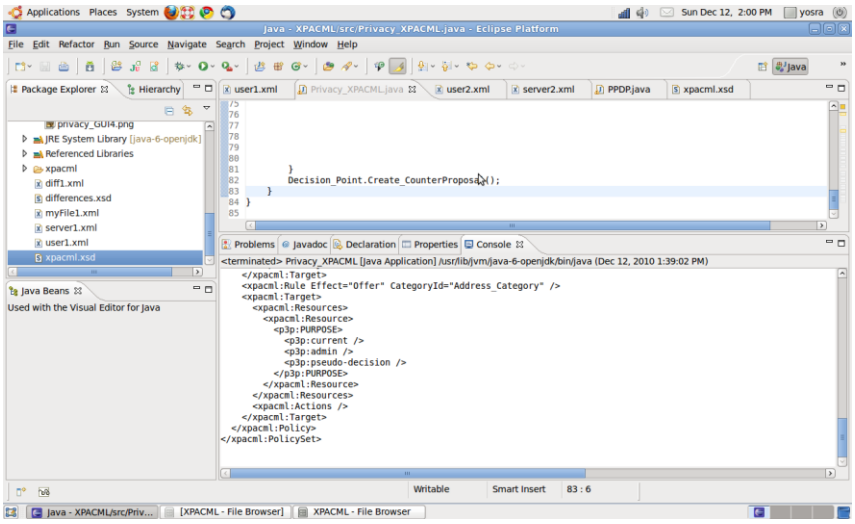


Figure 4.16 Génération de l'offre utilisateur –Tour 1

4.6.6.2 Tour 2

Le SP construit le Message 4 (voir figure 4.12) comme suit : si Required_i ⊆ $(D\_i, X\_i)_{user}$, alors l'offre de l'utilisateur est acceptée et le SP envoie son accord avec "*Permit*" dans l'élément (Eff) de la règle R_i. Sinon, le SP monte une contre proposition où toutes les valeurs optionnelles acceptées par l'utilisateur sont gardés (avec Eff="*Permit*") et les valeurs obligatoires du SP sont introduites encore une fois (voir section 4.6.5.2) avec la valeur Eff mise à "*Offer_Policy*". Toutes les contre propositions acceptées font partie du Message 4.

Du moment que le message 4 inclut au minimum une contre proposition, l'agent utilisateur peut refuser, accepter directement, ou évaluer le risque des nouvelles valeurs proposées (section 4.6.4.2). Si le risque est inférieur au seuil définie par l'utilisateur, la négociation réussie, et l'utilisateur envoie son accord dans Message 5 (avec Eff="*Permit*") dans la règle acceptée. Sinon, la négociation échoue, et l'utilisateur envoie un Message 5 avec (avec Eff="*Deny*") dans la règle rejetée.

4.7 Conclusion

Dans ce chapitre, nous avons conçu et implémenté deux protocoles de négociation de politiques de protection des données. Le premier [Bek11] se base exclusivement sur la classification des valeurs des éléments `POLICY` pour définir les préférences utilisateur. Ce protocole est simple d'utilisation, seulement il n'est pas équitable. En effet, disposant chacun d'un nombre de politiques ordonnées, l'utilisateur comme le SP peut inférer des informations sur la politique du l'autre partie. Aussi, étant en fonction du nombre des préférences spécifiées par l'utilisateur et les politiques définies par le SP, le nombre des tours de négociation augmente rapidement si le nombre de politiques à disposition est important.

Ainsi, nous avons défini le deuxième protocole de négociation de politiques [Bek12] qui s'effectue en maximum deux tours de négociation, ce qui le rend plus réaliste et approprié pour les transactions réelles. La définition des préférences dans ce protocole repose sur la définition de trois ensembles au lieu de deux, ce qui permet ainsi une plus grande flexibilité pour le processus de négociation. Le protocole de négociation est aussi équitable, ainsi, ni l'utilisateur, ni le SP ne peut inférer d'informations sur la politique de protection de l'autre partie.

L'originalité de notre travail concerne les négociations élémentaires des données (termes de politiques), ainsi, chaque donnée est traitée séparément avec son tag de protection.

Il est à noter que la classification que nous avons présentée à titre démonstratif dans ce chapitre est statique. Pour que la phase de définition des préférences soit plus automatisée et fidèle au type de la donnée et du service en question, le classement des politiques doit être personnalisable en fonction du type de donnée, du type de service concerné, par le biais de définitions des relations dans l'ontologie de protection des données présentée dans la section 3.5 du chapitre 3.

CHAPITRE 5

Négociation des données personnelles

Sommaire

5.1 Problématique 178
5.2 Etat de l'art 179
5.3 Cadre de travail choisi 180
5.4 Les concepts privés et risques de révélation 180
5.4.1 Définitions 180
5.4.2 Risques de révélation des données personnelles 181
5.5 Eléments de la théorie des jeux 185
5.5.1 Taxonomie partielle des jeux 186
5.5.2 Concept de solution 187
5.5.3 Notre cas : Jeu non coopératif à somme non nulle 188
5.5.4 Fonctions d'utilité 188
5.6 Théorie des jeux appliquée à la négociation des données 189
5.6.1 Modélisation de la révélation des données comme un jeu statique 190
5.6.2 Formalisation 192
5.6.3 Calcul des stratégies optimales de révélation 201
5.7 Communication des données de concepts privés selon une révélation optimale 204
5.8 Contraintes complémentaires au protocole de négociation à base de théorie des jeux 205
5.9 Conclusion 206

5 Négociation des données personnelles

En accédant à un service donné, l'utilisateur est souvent contraint de dévoiler plus de données personnelles qu'il ne le faut. Trouver la juste quantité des données à révéler, est le problème auquel nous nous intéressons dans ce chapitre. Nous présentons une approche de négociation des données basée sur la théorie des jeux en considérant l'accès au service et la révélation des données comme un jeu où l'utilisateur et le SP sont les entités principales [Ros11].

Après la présentation de la problématique (section 5.1) et l'état de l'art (section 5.2), nous posons notre cadre de travail dans la section 5.3. Ensuite, nous définissons dans la section 5.4 les concepts privés sur lesquels nous établissons la négociation de données et nous calculons le risque inhérent à leur révélation. Nous présentons les éléments principaux issus du domaine de la théorie des jeux, et nécessaires à notre cas d'étude dans la section 5.5. Nous consacrons la section 5.6 à la modélisation du jeu de négociation en nous basant sur la théorie des jeux, le calcul des fonctions d'utilité de l'utilisateur et du SP permettant de trouver des stratégies optimales de révélation et effectuer ainsi des choix rationnels.

Une alternative basée sur les opérations de filtrage décrites à la section 2.6.2.2 du chapitre 2 est fournie dans la section 5.8, avant de conclure avec un bilan de nos contributions (section 5.9).

5.1 Problématique

Dans leurs échanges divers avec les SPs, les utilisateurs donnent souvent plus d'informations qu'il ne le faut. Par exemple, Alice qui souhaite prouver que c'est une personne adulte peut fournir son permis de conduire comme justificatif. Ainsi, elle révèle inutilement le numéro du permis de conduire, la date de naissance, l'adresse, le sexe, … etc. De ce fait, au lieu de donner la juste information qui prouve qu'elle est un adulte parmi tant d'autres, elle fournit une combinaison de données l'identifiant en tant que personne unique.

Si aujourd'hui les pratiques d'usage des données sont contrôlées par la directive européenne 95/46/CE (section 1.1.2.2 du chapitre 1), l'estimation des données à révéler est laissée quant à elle, à l'appréciation individuelle de chaque utilisateur.

A la lumière de ce constat, nous nous somme focalisés sur la recherche de méthodes aidant l'utilisateur à faire des choix rationnels quant à la révélation de ses données personnelles. Ainsi, nous nous somme placés dans un cadre de négociation des données à révéler lors d'une transaction électronique. De ce cadre découle une problématique

principale liée à l'interaction qui peut avoir lieu entre l'utilisateur et le SP. Quelle forme peut avoir cette interaction entre l'utilisateur et le SP ? Comment est-elle représentée ?

5.2 Etat de l'art

Nombreux sont les travaux qui s'intéressent à la protection de l'identité (approchée dans nos recherches par le concept générique « Concept privé » défini dans la section 5.4), mais réellement trop peu sont les outils qui permettent de protéger ces identités et de prendre des décisions quant à leur révélation.

k-Anonymity [Mae09] utilisé côté SP se trouve à la tête des méthodes permettant de protéger les informations personnelles identifiables, en noyant une identité x dans une masse d'identités ayant toutes des valeurs similaires. Traitant de grandes tables de données, chaque tuple possède au moins k-1 tuples dont les valeurs sont indistinctes sur un ensemble quasi-identifiant.

A titre d'exemple, si nous avons à disposition une table médicale avec des champs comme : Nom, Adresse, Race, Date de naissance, Sexe, code ZIP, problème de santé, …etc. En combinant une partie de ces champs, la révélation de l'identité de la personne est possible. Ces champs sont appelés des champs quasi- identifiants (QI). Ainsi, un champ quasi-identifiant pourrait être dans notre exemple : date de naissance, code zip et sexe. La table 5.1 montre un exemple de k-anonimity où k=2 et le champ quasi-identifiant QI = (âge, race, sexe, et code zip).

Age	Race	Gender	ZIP	Problem
65	White	Male	02456	Short breath
54	Black	Female	01776	Obesity
48	***White***	***Female***	***03187***	Diabete
48	***White***	***Female***	***03187***	Chest pain
70	Black	Male	03187	Cancer

Tableau 5.1 k-anonymity avec k=2 et QI={age, race, gender, zip}

Cette méthode appliquée ne permet pas de prendre de décisions quant à la quantité de données à révéler par un utilisateur. Elle ne permet pas également de visualiser l'aspect interactif entre l'utilisateur et le SP.

5.3 Cadre de travail choisi

La théorie des jeux est probablement le modèle formel le plus abouti pour l'étude des interactions stratégiques entre entités. Plusieurs raisons nous ont amenés à choisir ce cadre de travail. Tout d'abord les jeux permettent de décrire des situations sociales très différentes : les marchés en économie peuvent être vus comme des jeux dans lesquels les participants sont des producteurs ou des consommateurs ; et, plus généralement, une partie d'échecs, où la formation de négociations sont autant des jeux différents obéissant à des règles spécifiques. Ensuite, ce cadre de travail est, et a été, l'objet de nombreuses recherches, et est très riche en résultats. C'est également un cadre de travail très intuitif à manipuler : il est en effet facile de visualiser les interactions entre entités.

Informellement, un jeu consiste en un ensemble de joueurs, et pour chaque joueur, la donnée d'un ensemble de stratégies possibles et une fonction d'utilité reflétant ses préférences. Nous étudierons ici les interactions entre l'utilisateur et le SP, et la représentation de leurs préférences en nous plaçant dans un cadre simple : les jeux statiques à information incomplète. Un jeu est statique si les agents choisissent leur stratégie en parallèle et en une seule étape. Il est à information incomplète si chaque joueur connaît exactement l'état du monde, mais ne connait pas les préférences et les actions disponibles pour chacun des joueurs.

Plusieurs modes de représentation sont utilisés en théorie des jeux, notamment les formes extensives et les formes normales qui coïncident dans le cas des jeux statiques. La spécification d'un jeu statique nécessite la description des préférences des entités impliquées (l'utilisateur et le SP). Ces préférences sont décrites explicitement dans ces représentations par le biais de fonctions d'utilité.

Comme nous l'avons vu, notre objectif ici est de spécifier de façon concise et efficace les interactions entre l'utilisateur et le SP. Pour cela, nous avons choisi de nous appuyer sur la théorie des jeux, qui est un modèle formel abouti pour l'étude de ces interactions.

5.4 Les concepts privés et risques de révélation

5.4.1 Définitions

Définition 5.1 : nous définissons un concept privé comme un ensemble de données contribuant à une même information globale. Ainsi, le concept privé « carte d'identité » (figure 5.1) contient les éléments : Numéro de carte, Nom, Sexe, Date de naissance, Lieu de naissance, Taille. Chaque type de données peut lui aussi être divisé en sous ensembles. Par exemple, Nom se décompose en 'Nom de famille' et 'Prénom'.

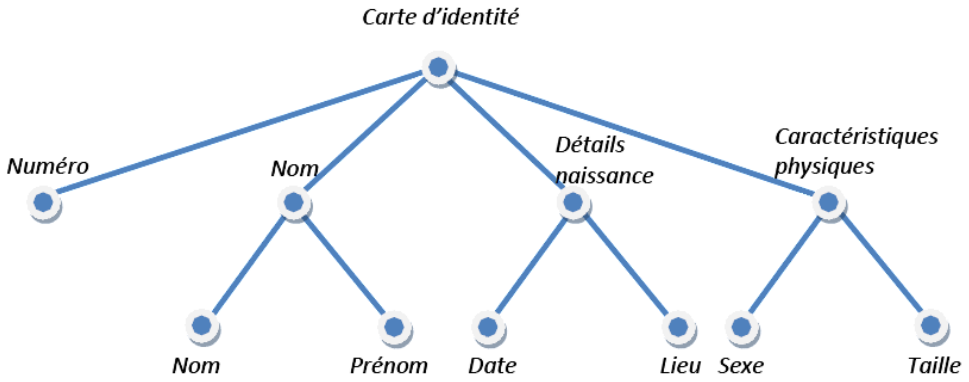


Figure 5.2 Concept privé - Carte d'identité

En nous basant sur le sous graphe `DATA_TYPE` de l'ontologie de protection des données (figure 3.3 du chapitre 3), nous exprimons les concepts privés (figure 5.2) et leurs éléments dans une structure hiérarchique comme illustré dans la figure 5.1. Du bas en haut dans cette structure, les éléments d'un concept deviennent de plus en plus spécifiques et peuvent représenter ainsi une identité.

Plus les éléments sont proches de la racine, plus l'information est précise, plus les éléments sont loin de la racines, plus ils tendent vers des éléments généraux communs entre plusieurs individus.

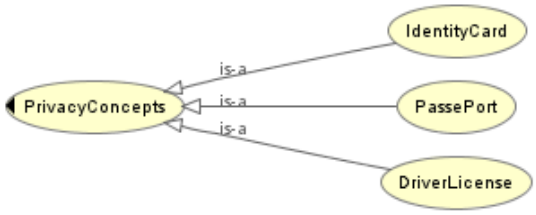


Figure 5.3 Classe -Concepts privé

5.4.2 Risques de révélation des données personnelles

Dans cette section, nous déterminons le risque lié à la révélation des concepts privés, connaissant les données à dévoiler. Cette **mesure du risque** repose sur un modèle que nous avons défini. Ce modèle est inspiré d'une méthode probabiliste expliquée dans [Tao06]. Notre modèle repose sur les deux notions suivantes : le concept privé CP et le

type de service TS. En effet, le concept privé est défini à la fois par sa taille, c'est-à-dire le nombre de données qu'il contient, et par le poids du risque qu'il engendre selon son appartenance à un type de service.
Notre modèle d'évaluation du risque utilise les notations présentées dans le tableau 5.2 et comporte les quatre étapes suivantes :

- Définir le tableau des tailles des concepts privés (section 5.4.2.1)
- Assigner un poids au risque de révélation de concepts privés après un type de service caractérisant le contexte de la transaction (section 5.4.2.2)
- Calcul du risque pour chaque concept privé selon le type de service (section 5.4.2.3)
- Calcul du risque global (section 5.4.2.4)

Notation	**Comments**
PC_i	Privacy Concept i
Z_i	Size of Privacy Concept i
NZ_i	Normalized data size of Privacy Concept i
ST_i	Service type i
RW_{ij}	Weighted Risk of privacy concept i under service type j
R_{ij}	Privacy risk of privacy concept i under service type j
NR_{ij}	Normalized privacy risk of privacy concept i under service type j
DR_i	Current disclosure risk of privacy concept i

Tableau 5.1 Notations utilisées

Ces étapes sont détaillées comme suit.

5.4.2.1 Conception du tableau des tailles des concepts privés

La **taille des données** de chaque concept privé correspond au nombre de données caractérisant ce concept privé. La **taille normée** représente le ratio de chaque concept privé par rapport à l'ensemble des concepts privés.

Privacy Concept	PC_1	…	PC_K
Data Size	$Z_1 = \|PC_1\|$	…	$Z_K=\|PC_K\|$
Normalized Data Size	$NZ_1 = \frac{Z_1}{\sum_{i=1}^{K} Z_i}$	…	$NZ_K = \frac{Z_K}{\sum_{i=1}^{K} Z_i}$

Tableau 5.2 Tailles des concepts privés

5.4.2.2 Assigner un poids au risque selon le contexte de la transaction

L'utilisateur assigne un **poids au risque de révélation de concepts privés dans un contexte situationnel** lié au contexte associé à chaque concept privé. Pour simplifier, nous avons réduit le contexte dans notre cas d'étude au type de service (élément `SERVICE_TYPE`) . On obtient donc le tableau 5.3 avec chaque poids dans l'intervalle [0..1]. Plus le poids de risque est élevé, plus le concept privé est sensible pour le type de service concerné. Par exemple, nous pouvons considérer que concernant le concept privé Carte_Identité, le risque possède un poids plus élevé pour le type de service eLearning que pour le type de service eCommerce. En effet, il est difficile de voir l'intérêt d'un site de eLearning à posséder de telles informations sur l'utilisateur alors qu'un site de e-Commerce peut avoir besoin des informations comme le nom et l'adresse pour livrer l'achat à l'utilisateur du service.

Weight	ST_1	…	ST_M
PC_1	RW_{11}	…	RW_{1M}
…	…	…	…
PC_K	RW_{K1}	…	RW_{KM}

Tableau 5.3 Poids associés au risque de révélation de concepts privés selon le type de service

5.4.2.3 Calcul du risque pour chaque concept privé selon le type de service

La **valeur du risque** dépend du type de service et s'obtient en multipliant le poids de risque du concept privé par sa taille normalisée, comme le présente le tableau 5.5.

Risk	ST_1	…	ST_M
PC_1	$R_{11}=NZ_1*RW_{11}$	…	$R_{1M}=NZ_1*RW_{1M}$
…	…	…	…
PC_K	$R_{K1}=NZ_K*RW_{K1}$	…	$R_{KM}=NZ_K*RW_{KM}$

Tableau 5.4 Poids du risque des concepts privés selon le type de service

Pour s'assurer d'obtenir un résultat de risque dans l'intervalle [0..1], il est nécessaire de normaliser les valeurs, comme indiqué dans le tableau 5.6.

Risk	ST_1	…	ST_M
PC_1	$NR_{11}=\frac{R_{11}}{\sum_{i=1}^{K} R}$	…	$NR_{1M}=\frac{R_{1M}}{\sum_{i=1}^{K} R_{iM}}$
…	…	…	…
PC_K	$NR_{K1}=\frac{R_{K1}}{\sum_{i=1}^{K} R}$	…	$NR_{KM}=\frac{R_{KM}}{\sum_{i=1}^{K} R_{iM}}$

Tableau 5.5 Poids normalisés du risque des concepts privés selon le type de service

5.4.2.4 Calcul du risque final

Soit $\boldsymbol{\alpha_i}$ le nombre de données d'un concept privé à qui seront révéler : il est possible de pondérer le risque normalisé avec le coefficient $\frac{\alpha_i}{Z_i}$. Le risque de révélation RR_{ij} pour un concept privé i selon le type de service j s'écrit :

$$RR_{ij}=\frac{\alpha_i}{Z_i}* NR_{ij}$$

Le **risque total** (dépendant toujours du type de service) correspond donc à la somme des risques de révélation soit :

$$R_j = \sum_{i=1}^{K} RR_{ij} \quad \text{Si} \quad R_j \leq 1$$

$$R_j = 1 \quad \text{Sinon}$$

Dans le cas où le risque de révélation où le risque total dépasserait la valeur 1, le résultat serait automatiquement ramené à 1 afin d'obtenir des résultats significatifs.

5.5 Eléments de la théorie des jeux

La théorie des jeux est un outil mathématique permettant d'étudier les comportements - prévus, réels, ou justifiés a posteriori - d'individus face à des situations d'antagonisme. Elle étudie des situations dans lesquelles le sort de chaque participant dépend non seulement des décisions qu'il prend, mais également des décisions prises par d'autres participants. Le choix optimal pour un joueur dépend donc généralement des choix des autres joueurs. Comme chaque joueur n'est pas totalement maître de son sort, on dit que les joueurs sont en situation **d'*interaction stratégique***. On suppose dans un jeu en interaction stratégique que les joueurs se connaissent : ils savent combien il y a de joueurs, et qui ils sont.

Du fait que le gain de chacun dépend en partie des actions des autres, un joueur ne peut pas se contenter de choisir ses propres plans d'actions, en négligeant ce que font les autres. Il doit au contraire se faire une idée aussi précise que possible des stratégies choisies, ou susceptibles d'être choisies, par les autres joueurs. Pour cela, on admet que **les entités sont rationnelles**, c'est-à-dire que chaque joueur s'efforce de prendre les meilleures décisions pour lui-même, et sait que les autres joueurs font de même.

Un **jeu stratégique** est donc caractérisé par :

- les joueurs, ils obéissent au principe de rationalité, c'est-à-dire que chacun recherche à prendre les meilleurs décisions pour lui-même dans le but d'optimiser ses gains,
- les espaces de stratégies (actions ou décisions), une stratégie étant définie comme un plan d'action complet spécifiant ce que le joueur fera à chaque étape de décision et face à chacune des situations pouvant survenir au cours du jeu,
- la séquence des décisions,
- les gains ou l'utilité des joueurs (fonction des décisions des joueurs),
- l'information à la disposition des joueurs.

Notre objectif ici n'est pas de donner un état de l'art exhaustif sur la théorie des jeux, nous voulons juste introduire quelques concepts qui nous seront utiles dans la suite de ce

chapitre. Nous allons donc tout d'abord présenter une taxonomie partielle des jeux en Section 5.5.1, puis nous présenterons en Section 5.5.2 quelques concepts de solution.

5.5.1 Taxonomie partielle des jeux

Nous allons présenter ici quatre types de jeux, les jeux statiques, dynamiques, coopératifs et non coopératifs. Un jeu peut réunir plusieurs de ces caractéristiques : il peut être statique et coopératif, statique et non coopératif, dynamique et coopératif ou encore dynamique et non coopératif.

En termes de représentation du jeu. Un jeu stratégique peut être représenté de deux façons différentes mais équivalentes : sous forme normale (dite aussi stratégique) et sous forme extensive (dite aussi développée).

5.5.1.1 Jeux statiques et dynamiques

La première distinction que nous allons faire est celle entre jeu statique et dynamique. Un jeu est statique lorsque tous les joueurs jouent simultanément en une seule étape, alors qu'il est dynamique lorsque le jeu se déroule en plusieurs étapes (un ou plusieurs joueurs peuvent jouer à chaque étape).

5.5.1.1.1 Jeux statiques

Un jeu est dit **statique** lorsque les joueurs choisissent **simultanément** leurs actions, et reçoivent ensuite leurs gains respectifs.

5.5.1.1.2 Jeux dynamiques

Un jeu dynamique est un jeu qui se déroule en **plusieurs étapes**, où un joueur prend une décision à chaque étape. Selon les informations que possède un joueur sur les actions des autres, un jeu dynamique peut être en information complète ou incomplète. Un jeu est **en information complète** si chaque joueur connaît l'ensemble des actions choisies par *tous* les joueurs qui sont intervenus avant qu'il ne sélectionne sa stratégie, et qu'il connaît toutes leurs stratégies possibles. Il est le seul joueur à prendre une décision à cette étape. Si plusieurs joueurs choisissent leurs actions simultanément à une étape donnée, ou si les joueurs ne connaissent pas toutes les stratégies des autres joueurs, le jeu est dit **en information incomplète**. Ces actions ne sont pas connues et chacun des joueurs intervenant à cette étape se comporte un peu comme dans un jeu statique, à la différence que dans ce cas, l'histoire du jeu influence le choix de chacun.

5.5.1.2 Jeux coopératifs et non coopératifs

5.5.1.2.1 Jeux coopératifs

Un jeu coopératif (appelé aussi jeu coalitionnel) est un jeu dans lequel les joueurs peuvent former des coalitions et agir de concert.

5.5.1.2.2 Jeux non coopératifs

Les jeux non coopératifs se divisent en deux grandes familles : les jeux à somme nulle, et ceux à somme non nulle. Les **jeux à somme nulle** sont tous les jeux où la somme "algébrique" des gains des joueurs est constante : ce que gagne l'un est nécessairement perdu par un autre. Stricto sensu, il est possible que les jeux ne soient pas à somme nulle, mais à somme constante, et cela n'a aucune importance en pratique : l'enjeu est de répartir entre tous les joueurs un total de gains préalablement fixé. Les échecs, le poker, …etc, sont des jeux à somme nulle, les gains d'un joueur étant très exactement les pertes d'un autre joueur. Tandis que les jeux non coopératif à **somme non nulle** (tels le dilemme du prisonnier), les joueurs ne peuvent pas communiquer et passer un accord pour former une coalition. Ainsi, ce que gagne l'un n'est pas forcément perdu pour l'autre.

5.5.2 Concept de solution

Plusieurs concepts de solution existent : l'équilibre de Nash, et les stratégies dominantes, s'appliquent à des jeux statiques non coopératifs. Aussi il y a le concept de noyau (core) qui s'applique à des jeux statiques coopératifs, et enfin les équilibres parfaits de Selten pour les jeux dynamiques.

Dans ce chapitre, nous calculons l'équilibre de Nash existant entre les stratégies des deux joueurs. Introduit par John Nash en 1950 [Nas50], l'**équilibre de Nash** est un concept fondamental en théorie des jeux. Il décrit une issue du jeu dans laquelle aucun joueur ne souhaite modifier sa stratégie étant donnée la stratégie de chacun de ses rivaux.

5.5.2.1 L'équilibre de Nash

L'analyse d'un jeu permet de prédire l'**équilibre** qui émergera si les joueurs sont rationnels. Par équilibre, nous entendons un état ou une situation dans lequel aucun joueur ne souhaite modifier son comportement compte tenu du comportement des autres joueurs. De façon plus précise, un équilibre est une combinaison de stratégies telle qu'aucun des joueurs n'a d'intérêt à changer sa stratégie compte tenu des stratégies des autres joueurs. Une fois que l'équilibre a été atteint dans un jeu (et peu importe la manière dont il a été obtenu), il n'y a aucune raison de le quitter. Les jeux pour lesquels il est possible de calculer un équilibre de Nash sont représentés dans le tableau 5.7.

	Non Coopératif	
	Somme nulle	**Somme non nulle**
Statique	Nash	Nash
Dynamique		

Tableau 5.6 Taxonomie des jeux : Domaine d'application des équilibres de Nash

5.5.2.2 Stratégies dominantes

Une **stratégie dominante** pour un joueur est une stratégie qui lui donne toujours un gain supérieur ou égal au gain qu'il peut attendre de toutes ses autres stratégies (quelles que soient les stratégies des autres joueurs) : une stratégie dominante domine alors toutes les autres stratégies.

Si chacun des joueurs possède une stratégie dominante, alors il existe au moins un équilibre de Nash consistant pour les joueurs à choisir leur stratégie dominante : un équilibre en stratégies dominantes est un équilibre de Nash (réciproque non vérifiée).

Il est à noter que le concept de stratégies dominées pouvait conduire à des résultats alors que celui d'équilibre de Nash en stratégies pures aboutissait à une impasse. Mais l'inverse peut être vrai aussi : le processus d'élimination des stratégies dominées ne conduit pas forcément à un résultat, alors que l'on peut obtenir un équilibre de Nash,

5.5.3 Notre cas : Jeu non coopératif à somme non nulle

Nous avons précisé dans la section 5.3 que nous nous plaçons dans le cadre d'un jeu statique à information incomplète. En effet, l'utilisateur et le SP dans notre cas d'études ne communiquent pas concernant leurs stratégies. Aussi, chacun d'eux est rationnel et cherche à prendre les meilleures décisions pour lui-même. Le jeu est donc **non-coopératif**, et le but est de trouver l'**équilibre de Nash**.
L'issue dans notre jeu est profitable pour tous : finalement, l'utilisateur obtient le service, et le SP fournit son service (ou alors elle n'est profitable pour aucun des deux joueurs s'ils n'arrivent pas à se mettre d'accord). Le jeu est donc à **somme non nulle**. Enfin, notre cas d'étude est un **jeu statique, non-coopératif, à somme non nulle**, dont on se contente de faire une représentation normale.

5.5.4 Fonctions d'utilité

Dans la spécification d'un jeu statique, il est impératif de décrire les préférences des joueurs impliqués. L'utilité [Alt07] est une fonction reflétant les préférences d'un joueur.

Elle est croissante par rapport à ses préférences : l'utilité d'un joueur est plus élevée pour un choix de décisions par rapport à un autre s'il préfère le premier choix par rapport à l'autre. Dans notre cas d'étude, il va s'agir de définir de façon explicite **deux fonctions d'utilité**, une pour chaque joueur (utilisateur et SP).

5.6 Théorie des jeux appliquée à la négociation des données

Dans le cadre de la négociation des données dans laquelle nous nous sommes placés, l'utilisateur comme le SP sont rationnels et essayent de satisfaire au mieux leurs préférences. Notre objectif est de spécifier de façon concise et efficace les interactions qui les animent dans un processus de négociation. Pour cela, nous avons choisi de nous appuyer sur la théorie des jeux, qui est un modèle formel abouti pour l'étude de ces interactions stratégiques. Ainsi, nous nous focalisons dans cette section sur la modélisation du problème de négociation, en un problème de jeu statique, non coopératif, à somme non nulle. Nous nous attachons particulièrement à faire émerger les formules correspondant aux fonctions d'utilité des joueurs.

Les fonctions d'utilité (section 5.6.2.2) vont permettre de modéliser les préférences de chacun des joueurs, afin d'ensuite dérouler la méthodologie propre à la négociation, qui nous permettra d'obtenir l'existence ou non d'un équilibre de Nash.
Notre méthodologie consiste à s'appuyer sur les fonctions d'utilités, afin de trouver les stratégies (les **expositions optimums** ($e_u{}^*$ et $e_s{}^*$)) des joueurs correspondant à l'équilibre de Nash.
Cela consiste en 4 étapes : tout commence par l'utilisation de la contrainte qui rentre en compte du côté utilisateur. En effet, comme on est dans le cas d'un jeu stratégique, la fonction d'utilité *Pu* est dite sous contrainte (section 5.6.3), c'est-à-dire que e_u est bornée par une fonction qui peut dépendre des stratégies des autres joueurs. Dans notre cas la fonction en question est dépendante des expositions e_u et de e_s (section 5.6.1), et sera explicitée plus loin dans ce rapport (section 5.6.2).

La méthodologie peut être ainsi résumée par les points suivants :

- Etablissement d'une fonction d'utilité au niveau de l'utilisateur *Pu* avec une contrainte. En appliquant le Lagrangien sur cette fonction, il est possible de trouver l'exposition optimum de l'utilisateur $e_u{}^*$ en fonction de e_s.

- Réinjection du résultat dans la fonction d'utilité du SP *Ps*. Puis, calcul de la dérivée du *Ps* en fonction de e_s pour trouver $e_s{}^*$ (qui ne dépend pas de e_u).

- Réinjection du résultat dans l'expression de $e_u{}^*$ pour que cette dernière ne dépende plus de e_s.

- Réinjection du résultat obtenu dans *Pu* et dans *Ps*.

Pour les différents calculs et traçages de courbes, nous avons utilisé le logiciel **Maple** V 9.5. En effet, ce dernier est particulièrement adapté aux applications graphiques et aux mathématiques.

5.6.1 Modélisation de la révélation des données comme un jeu statique

5.6.1.1 Ensemble de joueurs

$J = \{u, s\}$

Dans notre étude de cas, l'échange se fait uniquement entre deux joueurs, l'utilisateur u et le SP s. Ainsi, l'interaction entre un SP et plusieurs utilisateurs, ou plusieurs utilisateurs et plusieurs SPs n'est pas prise en compte.

5.6.1.2 Ensemble de stratégies

$S_u = \{e_u\}$ et $S_s = \{e_s\}$

Chaque joueur possède sa propre stratégie concernant la révélation des données.
e_u représente la sensibilité de l'utilisateur à dévoiler ses données personnelles tandis que e_s représente la volonté du SP à faire dévoiler les données personnelles de l'utilisateur.
Nous posons : $e_u \in [0,1]$ et $e_s \in [0,1]$. En effet, ainsi e_u et e_s représentent des taux d'exposition, ce qui est intéressant pour notre étude.
Nous verrons dans la section 5.7 le lien concret entre exposition et données personnelles.

5.6.1.3 Fonctions d'utilités

$F = \{P_u(e_u, e_s), P_s(e_u, e_s)\}$

Chaque joueur possède une fonction d'utilité propre qui représente son bénéfice en fonction des stratégies de l'ensemble des joueurs.
Par exemple, l'utilité du SP augmente lorsque les révélations augmentent (ceci est bien sûr une simplification), au contraire l'utilité de l'utilisateur diminue avec ces augmentations.
Dans le cadre de notre étude, on considère quelques limitations : les prix des services ne sont pas comptabilisés car ils sont considérés comme indépendants de la révélation d'attributs.

5.6.1.4 Cheminement

Nous détaillons dans cette section le cheminement de notre raisonnement, qui nous a amené à établir les fonctions finales que nous proposons dans la section 5.6.2.
Nos recherches sur la théorie des jeux, nous ont amenés à établir deux fonctions d'utilité : une pour l'utilisateur (P_u) et l'autre pour le SP (P_s).
Cependant, malgré nos multiples recherches dans la littérature concernant la révélation des données, aucune situation ne correspondait à celle que nous étudions. Il a donc fallu essayer d'imaginer l'allure que pouvaient avoir les courbes représentant ces fonctions d'utilité, avant de trouver des formules adaptées pour les représenter.

5.6.1.5 Formules générales

Dans un premier temps, nous avons établi les formules générales suivantes :

$$P_u = Access\ Service\ \times Weight_u - exposure_u\ = AS\ \times W_u\ - e_u$$
$$P_s = Provide\ Service\ \times Weight_s - exposure_u\ = PS\ \times W_s\ +\ e_u$$

Pour l'utilisateur comme pour le SP, le premier membre de la fonction d'utilité est l'accès au service / la fourniture du service, multiplié par un poids propre au joueur concerné (utilisateur ou SP). *AS* et *PS* représentent donc respectivement le bénéfice de l'utilisateur et du SP à accéder ou à fournir le service. L'intérêt peut être différent pour les deux agents.
Le second membre est l'exposition.
Comme vu dans la section 5.6.1, e_u désigne l'exposition de l'utilisateur, c'est-à-dire la quantité de données personnelles qu'il révèle pour obtenir le service. Nous avons choisi e_u ∈ [0,1], où 0 correspond à « aucune révélation » tandis que 1 correspond à « une révélation complète de toutes les données personnelles comprises dans un concept privé.».
La préférence de l'utilisateur est évidemment de minimiser son exposition e_u, tandis que celle du SP est de la maximiser. La fonction d'utilité étant une fonction croissante en fonction des préférences, le but est de la maximiser pour chaque joueur. Cela explique pourquoi on soustrait e_u dans le cas de l'utilisateur, et pourquoi on ajoute e_u dans le cas du SP.

Nous avons ensuite détaillé d'avantage ces formules générales. Ainsi, nous avons établi :

$$P_u(e_u) = \frac{1}{K_1 + K_2} \times [K_1 \times \mathrm{QoS_u}(e_u) + K_2 \times préférence_u(e_u)]$$
$$P_s(e_u) = \frac{1}{K_3 + K_4} \times [K_3 \times \mathrm{QoS_s}(e_u) + K_4 \times préférence_s(e_u)]$$

Nous avons introduit la notion de qualité de service pour le premier membre, puisque ce dernier semblait correspondre à cette notion. En effet, QoSu représente l'intérêt de l'utilisateur à accéder au service, et QoSs l'intérêt du SP à fournir le service, le tout en

fonction des données dévoilées (e_u). En effet, plus l'utilisateur communique ses données privées, plus le service auquel il a accès est complet.
La $préférence_u$ représente le bénéfice de l'utilisateur à dévoiler ses données privées, et $préférence_s$ celui du SP à récolter les données privées de l'utilisateur. La préférence de l'utilisateur est toujours de minimiser son exposition e_u, tandis que celle du SP est toujours de la maximiser.
On voit apparaître K1, K2, K3 et K4, qui sont des constantes positives : elles permettent de pondérer la priorité entre la QoS et la préférence.
Ainsi, la négociation d'attribut correspond à une étude monocritère basée sur le paramètre e_u.

5.6.2 Formalisation

5.6.2.1 Formules

5.6.2.1.1 Exposition corrélée

Dans un premier temps, il est nécessaire de définir l'**exposition corrélée** E dépendant de l'exposition propre de chacun des deux joueurs. En effet, l'exposition corrélée dépend des variables e_u et e_s car son objectif est de lier les deux expositions entre elles afin de les rendre **interdépendantes**. Cette fonction interviendra dans les fonctions d'utilités des deux joueurs. Comme expliqué dans la section 5.4, dans une modélisation de jeu, il est indispensable d'avoir des **fonctions d'utilité interdépendantes** car l'utilité d'un joueur dépend de l'action de l'autre.

Soient e_u et e_s tels que définis dans la section 5.6.1.
Elle est définie par :

$$E(e_u, e_s) = \frac{e_u \times e_s}{e_u + e_s}$$

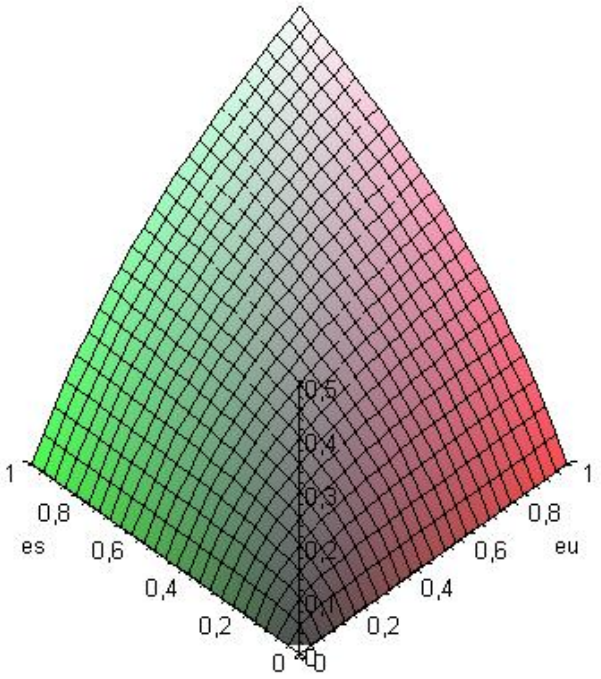


Figure 5.4 Exposition corrélée dépendant de *eu* et *es*

La fonction $E(e_u, e_s)$ est trivialement définie et continue sur $\mathbb{R}_+^2 \setminus (0,0)$ et prolongeable par continuité en (0,0) par $E(e_u, e_s)$ = (0,0).
On remarque que si e_u=0 ou e_s=0 alors $E(e_u, e_s)$ =0 et que si e_u augmente ou e_s augmente alors $E(e_u, e_s)$ augmente.

5.6.2.1.2 Qualité de service

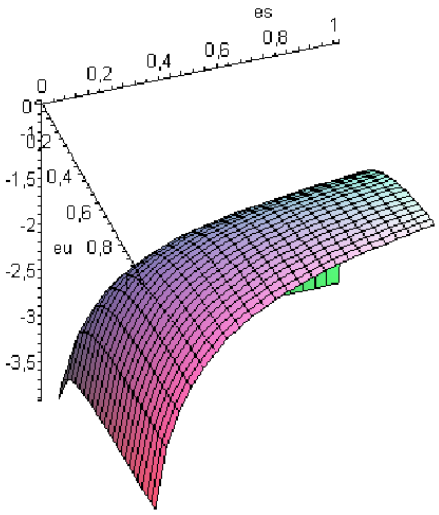
La variable dépendant de la révélation des données SP est la **qualité de service**. Cette variable représente l'apport de l'accès au service pour l'utilisateur, et le coût de la fourniture du service pour le SP. La variable QoS est exprimée en fonction de l'exposition corrélée (donc en fonction de e_u et de e_s). En effet la fourniture du service aussi bien que son accès, en ce qui concerne la QoS, dépend largement de la perception des deux entités du jeu, ainsi, il faut faire intervenir les deux stratégies. Nous estimons aussi qu'une fonction symétrique par rapport à ces stratégies s'adapterait mieux à notre cas vu qu'elles sont interchangeables.

Pour l'utilisateur:
Avoir accès au service est l'un de ses objectifs principaux, cela augmente donc son utilité globale.

Aussi, plus l'utilisateur fournit un grand nombre de données personnelles plus la QoS est élevée, la fonction est donc croissante par rapport à l'exposition corrélée. De plus, il faut que le comportement de cette dernière respecte les deux contraintes suivantes :
- pour des petites valeurs d'exposition (< 1/2), une petite augmentation de l'exposition se traduit par une grande augmentation de la QoS,
- pour des grandes valeurs d'exposition (> 1/2), une petite augmentation de l'exposition se traduit par une petite augmentation de la QoS.

Nous pourrions traduire mathématiquement ces contraintes par des plans tangents en (e_0, e_0) qui seraient de moins en moins inclinés par rapport au plan formé par les axes e_u et e_s lorsque e_0 augmente.
Nous avons choisi la fonction :

$$QoS(e_u, e_s) = \ln\left(\frac{e_u \times e_s}{e_u + e_s}\right)$$



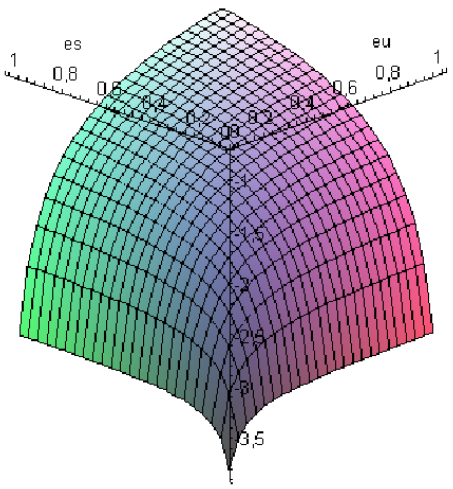


Figure 5.5 Qualité de service de l'utilisateur Vue1

Figure 5.30 Qualité de service de l'utilisateur - Vue 2

Nous obtenons bien le comportement attendu, une fonction symétrique et les variations désirées, plus les valeurs d'exposition sont élevées, plus la variation de la QoS (provoquée par une variation des expositions) est petite. En effet, nous pouvons témoigner de ce comportement par l'utilisation de plans tangents en différents points (e_u, e_s) tels que $e_u = e_s$: plus e est grand moins les plans tangents sont inclinés.

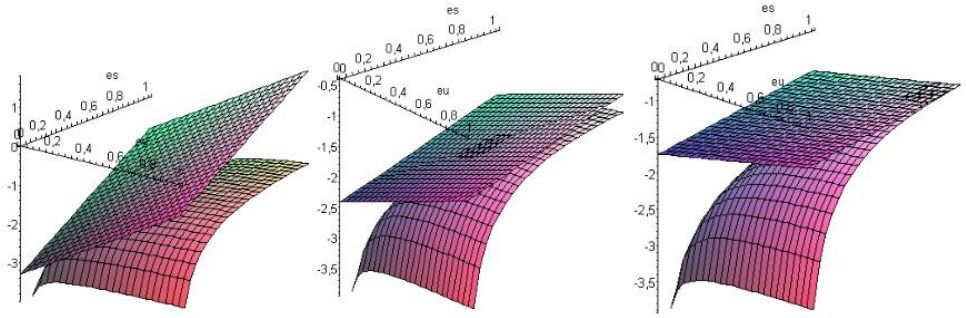


Figure 5.6 Qualité de service de l'utilisateur avec plan tangent aux points : (0,;,2), (0,5;0,5) et (1;1)

De plus, le fait que $\lim_{(e_u,e_s)\to(0,0)} QoS(e_u,e_s) = -\infty$ montre qu'aucun service n'est possible sans échange d'attributs privés.

Pour le SP

Fournir le service représente un coût. La fonction devient :

$$QoS(e_u,e_s) = -\ln\left(\frac{e_u \times e_s}{e_u + e_s}\right)$$

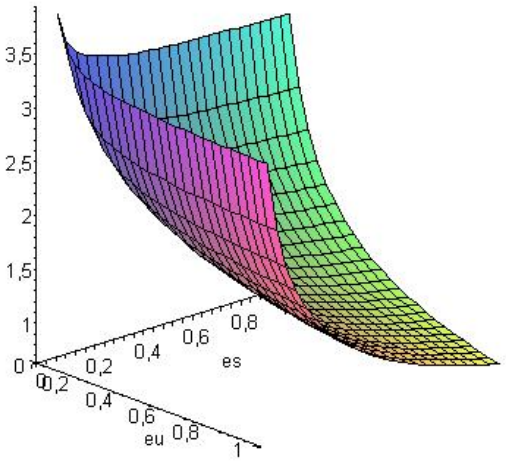


Figure 5.7 Qualité de service du SP du service

Nous retrouvons le même comportement vis-à-vis des variations avec une fonction cette fois-ci décroissante en fonction de (e_u, e_s):
- pour des valeurs d'expositions faibles, une petite augmentation de ces dernières entraine une grande diminution de la QoS (autrement dit, une petite augmentation de l'exposition entraine une grande diminution du coût),

- pour des valeurs d'expositions élevées, une petite augmentation de ces dernières entraine une petite diminution de la QoS (autrement dit, une petite augmentation de l'exposition entraine une petite diminution du coût).
D'autre part, $\lim_{(e_u,e_s)\to(0,0)} QoS(e_u, e_s) = +\infty$ montre qu'aucun service ne peut être fourni sans l'échange de données personnelles.

5.6.2.1.3 Préférence

La **préférence** représente la partie de la fonction d'utilité purement liée à l'exposition, c'est-à-dire ce qu'apporte ou ce que coûte la révélation des données aux différents acteurs. Il n'y a plus d'interdépendance, par conséquent nous obtenons des fonctions dissymétriques dépendant seulement de e_u pour la préférence utilisateur, et de e_s pour celle du SP.

Pour l'utilisateur:
La préférence doit être décroissante en fonction de e_u vu que fournir des données personnelles représente un coût et diminue donc son utilité globale.
La fonction de préférence s'écrit :

$$pref(e_u, e_s) = -e_u$$

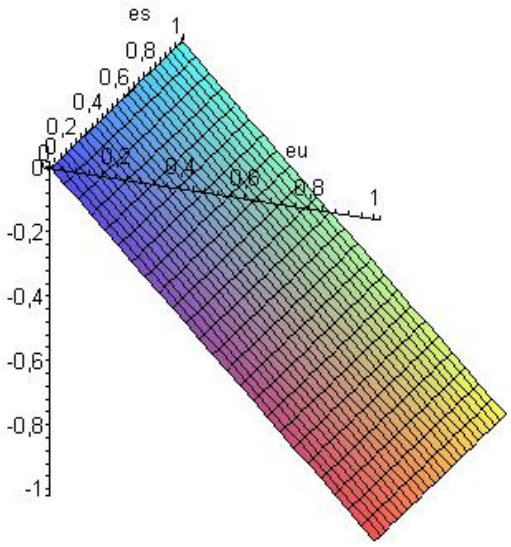


Figure 5.8 Préférence de l'utilisateur

Pour le SP:

La préférence doit être croissante en fonction de e_s vu qu'obtenir les données personnelles de l'utilisateur est l'un de ses objectifs et augmente donc son utilité globale.
La fonction devient :

$$pref(e_u, e_s) = + e_s$$

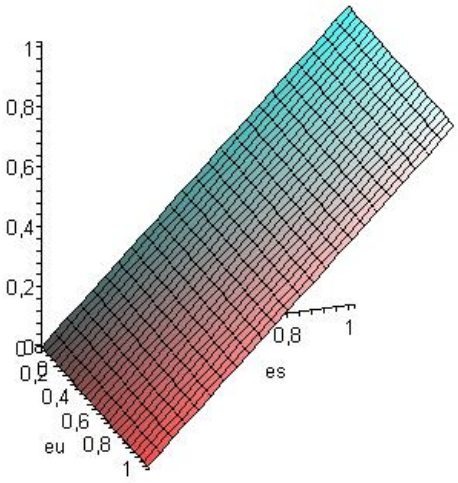


Figure 5.9 Préférence du SP du service

5.6.2.2 Fonctions d'utilité

5.6.2.2.1 Fonction d'utilité de l'utilisateur

Soient $K1$ et $K2$ des coefficients strictement positifs, la fonction d'utilité globale de l'utilisateur s'écrit (on a rassemblé les parties QoS et préférence) :

$$P_u(e_u, e_s) = \frac{1}{K_1 + K_2} \times [\, K_1 \ln\left(\frac{e_u \times e_s}{e_u + e_s}\right) - K_2\, e_u \,]$$

Traçons cette courbe avec par exemple les coefficients suivants : $K1$=1 et $K2$=1.

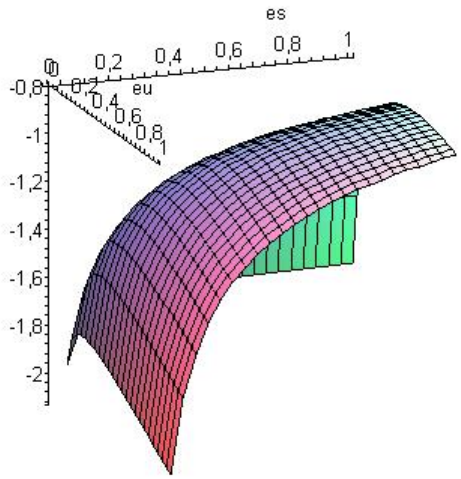


Figure 5.10 Fonction d'utilité de l'utilisateur - vue 1

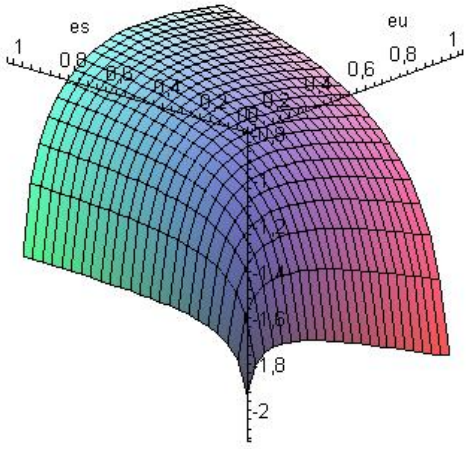


Figure 5.11 Fonction d'utilité de l'utilisateur - vue 2

5.6.2.2.2 Fonction d'utilité du SP

Soient $K3$ et $K4$ des coefficients strictement positifs, la fonction d'utilité globale du SP s'écrit (on a rassemblé les parties QoS et préférence) :

$$P_s(e_u, e_s) = \frac{1}{K_3 + K_4} \times [\, -K_3 \ln\left(\frac{e_u \times e_s}{e_u + e_s}\right) + K_4\, e_s\,]$$

Traçons cette courbe avec par exemple les coefficients suivants : $K3$=1 et $K4$=1.

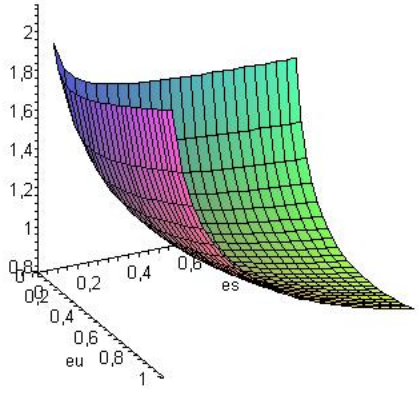


Figure 5.12 Fonction d'utilité du SP du service - vue 1

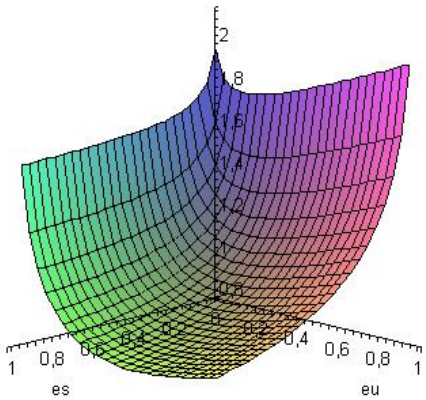


Figure 5.13 Fonction d'utilité du SP du service - vue 2

5.6.3 Calcul des stratégies optimales de révélation

Dans cette section, nous déroulons la méthodologie décrite au début de la section 5.6. Pour faciliter ce calcul, nous avons développé une application, toujours sous Maple, afin d'obtenir des résultats de manière rapide. Cette application est visible dans l'annexe B.

5.6.3.1 Stratégie optimale de l'utilisateur

En effet, la fonction d'utilité de l'utilisateur P_u est dite **sous contrainte**, c'est-à-dire que la stratégie de l'utilisateur e_u est bornée par une fonction.
Dans notre cas la fonction en question dépend de e_s et d'une constante *R* représentant un risque maximal à ne pas dépasser pour l'utilisateur. Notons que $R \in [0..1]$: afin d'avoir un résultat cohérent e_u, e_s et *R* se doivent d'être homogènes.

5.6.3.1.1 Contrainte

Le but de cette contrainte est d'imposer aux deux variables, e_u et e_s, une obligation de proximité. En effet, pour pouvoir dérouler notre méthodologie et obtenir l'optimum, il était obligatoire d'avoir une contrainte sur e_u, dépendant de e_s. D'autre part, cela permet de maintenir une certaine cohérence entre les deux stratégies. Ainsi, elles ne peuvent partir dans des directions qui seraient totalement opposées, ce qui ne serait pas crédible dans une situation réelle.

Nous posons donc la contrainte : $e_s - e_u \leq \frac{R}{2}$

5.6.3.1.2 Méthode du Lagrangien

Soit λ le multiplicateur de Lagrange ($\lambda \geq 0$), nous avons :

$$Max\ \mathcal{L}_u(e_u, \lambda) = P_u(e_u, e_s) - \lambda \left[e_s - e_u - \frac{R}{2} \right]$$

$$Max\ \mathcal{L}_u(e_u, \lambda) = \frac{1}{K_1 + K_2} \times [\, K_1 ln\left(\frac{e_u\, e_s}{e_u + e_s}\right) - K_2 e_u] - \lambda \left[e_s - e_u - \frac{R}{2} \right]$$

Nous remarquerons que, ici, e_s est vu comme une constante.
Résoudre cette équation revient à chercher les points solutions du système :

$$\begin{cases} \dfrac{\partial \mathcal{L}_u(e_u, \lambda)}{\partial e_u} = 0 \\ \dfrac{\partial \mathcal{L}_u(e_u, \lambda)}{\partial \lambda} = 0 \end{cases}$$

$$\frac{\partial \mathcal{L}_u(e_u,\lambda)}{\partial e_u} = 0 \quad \Leftrightarrow \frac{1}{K_1+K_2} \times [\, K_1 \frac{e_s}{e_u\,(e_u+\,e_s)} - K_2] + \,\lambda = 0$$

$$\Leftrightarrow \lambda = -\frac{1}{K_1+K_2} \times [\, K_1 \frac{e_s}{e_u\,(e_u+\,e_s)} - K_2]$$

$$\frac{\partial \mathcal{L}_u(e_u,\lambda)}{\partial \lambda} = 0 \quad \Leftrightarrow e_s - \,e_u - \frac{R}{2} = 0$$

$$\Leftrightarrow e_u = \,e_s - \frac{R}{2}$$

Les solutions sont :

$$\begin{cases} e_u{}^* = e_s - \dfrac{R}{2} \\ \lambda^* = -\dfrac{1}{K_1+K_2} \times [\, K_1 \dfrac{e_s}{(e_s-\frac{R}{2})(2\,e_s-\frac{R}{2})} - K_2] \end{cases}$$

Nous avons donc obtenu une fonction $e_u{}^*(e_s)$.

5.6.3.2 Stratégie optimale du SP

La fonction d'utilité de P_S s'écrit :

$$P_s(e_u,e_s) = \frac{1}{K_3+K_4} \times [\, -K_3 \ln\left(\frac{e_u \times e_s}{e_u+e_s}\right) + K_4\, e_s\,]$$

5.6.3.2.1 Utilisation de l'exposition utilisateur optimale

En remplaçant e_u par $e_u{}^*(e_s)$, nous obtenons une fonction uniquement dépendante de e_s :

$$P_s(e_u{}^*,e_s) = \frac{1}{K_3+K_4} \times [\, -K_3 \ln\left(\frac{(e_s-\frac{R}{2}) \times e_s}{e_s-\frac{R}{2}+e_s}\right) + K_4\, e_s\,]$$

$$P_s(e_u{}^*,e_s) = \frac{1}{K_3+K_4} \times [\, -K_3 \ln\left(\frac{e_s{}^2-\frac{R}{2}e_s}{2\,e_s-\frac{R}{2}}\right) + K_4\, e_s\,]$$

5.6.3.2.2 Méthode de la dérivée

Pour obtenir l'exposition du SP optimale, il faut résoudre l'équation suivante :

$$\frac{\partial P_s(e_u*, e_s)}{\partial e_s} = 0$$

$$\frac{\partial P_s(e_u*, e_s)}{\partial e_s} = 0 \Leftrightarrow \frac{1}{K_3 + K_4} \times [\, -K_3 \frac{\left(2\, e_s - \frac{R}{2}\right)^2 - (2\, e_s{}^2 - R\, e_s)}{\left(2\, e_s - \frac{R}{2}\right)\left(e_s{}^2 - \frac{R}{2}\, e_s\right)} + K_4 \,] = 0$$

$$\Leftrightarrow \; -K_3 \frac{2\, e_s{}^2 - R\, e_s + \frac{R^2}{4}}{\left(2\, e_s - \frac{R}{2}\right)\left(e_s{}^2 - \frac{R}{2}\, e_s\right)} + K_4 = 0$$

$$\Leftrightarrow \; \frac{-K_3 \left(2 e_s{}^2 - R\, e_s + \frac{R^2}{4}\right) + K_4 \left(2\, e_s{}^3 - \frac{3R}{2}\, e_s{}^2 + \frac{R^2}{4}\, e_s\right)}{\left(2\, e_s - \frac{R}{2}\right)\left(e_s{}^2 - \frac{R}{2}\, e_s\right)} = 0$$

$$\Leftrightarrow \; 2\, K_4\, e_s{}^3 + (2\, K_3 - \frac{3R}{2}\, K_4)\, e_s{}^2 + (\frac{R^2}{4}\, K_4 - R\, K_3)\, e_s - \frac{R^2}{4} K_3 = 0$$

Nous obtenons une équation de degré 3 à résoudre. Pour obtenir un résultat, nous utilisons un logiciel de calcul. Celui-ci ayant une forme algébrique assez compliquée, nous avons estimé qu'il n'était pas utile de le reporter ici, cependant ce résultat est présenté dans l'annexe C. Nous traiterons un exemple dans la section 5.6.3.3.

Nous avons donc obtenu $e_s{}^*$. Supposons la bonne rationalité du SP, nous supposons que K3≤K4.
Ensuite, il suffit d'injecter cette solution dans la fonction $e_u*(e_s)$ pour obtenir e_u* (indépendant de e_s).
Dans le cas où R=1, il est nécessaire d'imposer des contraintes sur les constantes K3 et K4 pour obtenir des solutions. En effet, seuls quelques couples de coefficients amènent à des résultats. Pour des valeurs de ces derniers inférieures ou égales à 10, la liste des couples (K3, K4) est :
- (3,5)
- (3,10)
- (6,10)

5.6.3.3 Exemple

5.6.3.3.1 Expositions optimales

Pour obtenir un résultat numérique de $e_s{}^*$, nous allons fixer les constantes K3, K4 et R. Soient, par exemple, R=1, K3=3 et K4=5, nous obtenons le résultat suivant :

$$e_s* \in \left\{1, \frac{7}{40} + i\, \frac{\sqrt{71}}{40}, \frac{7}{40} - i\, \frac{\sqrt{71}}{40}\right\}$$

Nous éliminons les résultats négatifs et ceux ayant une partie imaginaire. Par conséquent, il ne reste plus qu'une solution unique : $e_s* = 1$

Or $e_u* = e_s* - \frac{R}{2}$ donc : $e_u* = \frac{1}{2}$

Les stratégies optimales sont : $(e_u*, e_s*) = (\frac{1}{2}, 1)$.

5.6.3.3.2 Stratégies optimales

Pour calculer les utilités maximales, il nous faut fixer les deux coefficients restants : K1 et K2. Nous choisissons par exemple : K1=1 et K2=1 sachant que nous avons déjà fixé K3=3 et K4=5.

$$P_u(e_u*, e_s*) = \frac{1}{K_1 + K_2} \times [\, K_1 \ln\left(\frac{e_u* \times e_s*}{e_u* + e_s*}\right) - K_2\, e_u* \,]$$

$$P_u\left(\frac{1}{2}, 1\right) = -\frac{1}{2} \times [\ln(3) + \frac{1}{2}]$$

$$P_u\left(\frac{1}{2}, 1\right) = -\frac{1}{2} \times [\ln(3) + \frac{1}{2}]$$

$$P_u\left(\frac{1}{2}, 1\right) \approx -0.799$$

$$P_s(e_u*, e_s*) = \frac{1}{K_3 + K_4} \times [\, -K_3 \ln\left(\frac{e_u* \times e_s*}{e_u* + e_s*}\right) + K_4\, e_s* \,]$$

$$P_s\left(\frac{1}{2}, 1\right) = \frac{1}{8} \times [3\ \ln(3) + 5\,]$$

$$P_s\left(\frac{1}{2}, 1\right) \approx 1.037$$

Ainsi, pour notre exemple la stratégie (Pu, Ps) = (-0.799, 1.037), permet d'atteindre un état d'équilibre entre l'utilisateur et le SP.

5.7 Communication des données de concepts privés selon une révélation optimale

Maintenant que nous avons étudié l'exposition optimum, nous repartons des concepts privés pour essayer de faire le lien entre les deux, et ainsi établir ce que cela représente concrètement pour l'utilisateur et ses données personnelles.

Nous sommes arrivés dans la section 5.4.1 au résultat suivant :

$$RR_{ij} = \frac{\alpha_i}{Z_i} * NR_{ij}$$

Cela signifie que le risque de révélation des données d'un concept privé, sous un type de service donné, est proportionnel au risque normé dudit concept sous ce type de service, pondéré par la protection de données divulguées divisée par la quantité maximale (la taille du concept privé).

D'une part, les variables telles que la taille du concept privé et le poids du risque sont connues du côté utilisateur. Par conséquent, le risque normalisé NR_{ij} peut être calculé aisément.

D'autre part, la théorie des jeux porte à notre connaissance la révélation optimale $e*$ (voir la section 5.6.3). On peut donc assimiler cette révélation au risque de révélation des données personnelles (RR_{ij} s'il n'y a qu'un concept privé, et R_j s'il y en a plusieurs). En effet, quand l'utilisateur révèle des données personnelles, il prend un risque. Ce risque est proportionnel par rapport à la révélation des données. De plus, la révélation e est comprise entre 0 et 1, de même que RR_{ij} ou R_j . Tout concorde alors pour assimiler ces deux notions.

Nous avons donc à disposition NR_{ij} et Z_i par le biais des concepts privés, puis RR_{ij} (ou R_j) grâce à la théorie des jeux.

Dans le cas d'un seul concept privé RR_{ij} , nous pouvons donc en **déduire la variable α_i** $\alpha_i = \frac{RR_{ij} * Z_i}{NR_{ij}}$ représentant la quantité de données à révéler pour atteindre un état d'équilibre quant aux stratégies des deux joueurs, utilisateur et SP. Cet équilibre, nommé équilibre de Nash par la théorie des jeux, permet d'atteindre un état où les stratégies des deux joueurs sont optimales, c'est-à-dire qu'aucun des participants ne peut de manière unilatérale améliorer ses gains. Concrètement, cela signifie que lors de l'obtention de cet équilibre permis grâce à la révélation d'un nombre adéquat des données personnelles, l'utilisateur et le SP du service sont d'accord pour procéder à la transaction, c'est-à-dire la fourniture du service par le SP en échange des données personnelles de l'utilisateur.

5.8 Contraintes complémentaires au protocole de négociation à base de théorie des jeux

Nous avons défini dans le section 2.6.2.2 du chapitre 2, le concept « *type de sensibilité* » (`SENSITIVITY_TYPE`), qui permet aux données typées d'être sujettes à différentes opérations de filtrage telles que la transformation ou la perturbation avant leur révélation au SP.

Le tableau 5.8 vient apporter des éléments supplémentaires sur les types de sensibilités liées aux combinaisons de données (concepts privés), en complément du tableau 2.3 (section 2.6.2.2 du chapitre2). Spécifiés par l'utilisateur, ces types peuvent être utilisés comme alternative au processus de négociation fondé sur la théorie des jeux.

Sensitivity Type(s)	Description
CONJUNCTIVE	Value returned only if at most (N-1) of N values of the private concept (conjunctive clause) are requested
DISJUNCTIVE	Value returned only if at most 1 of N values of the private concept (disjunctive clause) are requested
OVERLAP	Value returned only if at most M of total data were released to the same SP in K queries recently

Tableau 5.7 Description des types de sensibilité associés à un donnée et prenant en compte les concepts privés

Les types `DISJUNCTIVE` et `CONJUNCTIVE` expriment des conditions dans lesquelles l'utilisateur ne souhaite révéler qu'une partie d'une combinaison des données. A titre d'exemple, la combinaison (date de naissance, sexe, et code ZIP) peut être utilisée pour identifier un utilisateur avec une grande probabilité. Dans ce cas, l'utilisateur peut spécifier une politique `CONJUNCTIVE` sur ces données, ainsi seulement deux données seront révélées.

Le type `OVERLAP` restreint l'accès aux données en se basant sur la fraction et du rang des données requise par le SP. Le type `OVERLAP` peut prévenir du zoom qui peut être fait par le SP sur un ensemble de données plus large qu'un concept privé. Ceci est possible en limitant le nombre de chevauchements des résultats entre requêtes.

5.9 Conclusion

Dans ce chapitre, nous avons donné une approche analysant les interactions stratégiques entre l'utilisateur et le SP lors d'un processus de négociation entre eux afin de trouver une révélation optimale des concepts privés. Un concept privé est un ensemble de données contribuant à une même information globale (ex : permis de conduire).

Le but étant de spécifier de façon concise et efficace les interactions entre l'utilisateur et le SP lors de cette négociation, nous nous somme basé sur un modèle formel à base de théorie des jeux pour l'étude des interactions entre les deux joueurs. Nous avons ainsi modélisé le processus de négociation des données comme un jeu en interactions stratégiques. Plus particulièrement, un jeu statique, non coopératif à somme non nulle.

La spécification du jeu, a nécessité la définition des préférences de l'utilisateur et du SP. Ces préférences donnent lieu à des fonctions d'utilité exprimant les gains de chacun et les conduisant à être rationnels dans le choix de stratégies de négociations optimales.

Il est également possible de modéliser la révélation des données d'un concept privé sous forme d'un jeu dynamique. En se référant à l'arbre de présentation de concepts privés (figure 5.1), il est possible de faire une négociation par nœud. Les fonctions d'utilité sont alors calculées pour chacun des les nœuds. Ainsi, la branche du nœud ayant l'utilité maximale reflètera la meilleure stratégie d'actions prises entre l'utilisateur et le SP.

A la fin du chapitre nous avons défini des types de sensibilité qui peuvent compléter ce processus de négociation pour les concepts privés particulièrement, et pour les combinaisons de données en général.

Conclusion et perspectives

6 Conclusion et perspectives

Dans cette thèse nous avons tout d'abord présenté et discuté le concept de sphère privée tel qu'il apparaît au quotidien, et notamment dans les réglementations opposables aux utilisateurs humains. Nous avons limité notre champ d'application à la protection des données personnelles en informatique (E-commerce particulièrement) et extrait six axes législatifs concernant cette protection en nous basant sur la directive Européenne 95/46/CE. Nous avons enfin discuté de la pertinence de quelques familles de méthodes et d'outils techniques existants.

Face aux manques des solutions de représentation de la législation et de leur mise en application selon les contextes d'usage des données, notre première contribution portait sur la définition d'un langage de protection des données nommé eXtensible Privacy Access Control Langage (XPACML) [Bek10]. Ce langage a un double objectif. Tout d'abord, il permet d'exprimer des politiques de protection des données personnelles et les préférences utilisateur suivant les six axes législatifs: l'information de l'utilisateur, son consentement, sa capacité à accéder aux données et à les modifier, la justification de la collecte et du traitement, la conservation des données et enfin leur transmission à des tiers. Ensuite, l'application de ces politiques de protection est rendue possible par la définition d'une architecture de contrôle d'accès aux données personnelles, qui est une extension de l'architecture de contrôle d'accès classique. Cette architecture inclut des modules destinés à la gestion de la protection des données personnelles en termes de contrôle d'usage et de négociation.

Pour que les politiques appliquées soient adaptées au contexte de chaque transaction électronique, nous avons défini (en nous basons sur le concept *d'attribut* du modèle ABAC [Wan04]) un modèle sémantique [Bek11a], [Bek12a] qui prend en compte la nature des données personnelles en question, leurs destinataires potentiels, et leur usage en cours. Ce modèle sémantique formalisé par une ontologie favorise la prise en compte et le traitement des informations sémantiques liées aux entités de chaque contexte transactionnel (en particulier, les données et les SPs). Cette conception sémantique a premièrement favorisé le partage d'un vocabulaire commun entre l'utilisateur et le SP. Ensuite, l'adaptation dynamique au contexte situationnel de chaque transaction a permis la génération de politiques de protection sensibles au contexte. Enfin, elle a favorisé l'analyse à différents niveaux d'abstraction des politiques de protection des données. Nous permettons ainsi de faire un raisonnement sur les données personnelles, leur destination et les contraintes d'usage qui s'appliquent sur elles.
Dans notre travail de modélisation sémantique et de gestion des politiques de protection, nous avons proposé une approche permettant d'établir une séparation claire entre les deux. Ainsi, au moment de l'écriture des politiques de contrôle d'accès (XPACML) aux données, on ne s'interroge guère du sens du (des) contexte(s) impliqué(s) dans la règle/ politique d'accès. De la même manière tout changement (ajout/modification) apporté aux contextes, est sans influence sur les politiques de protection des données. Par conséquent, notre solution qui se veut évolutive en termes d'attributs (du SP, et des données personnelles) pris en compte, permet une nette réduction de la charge d'administration des politiques.

Toujours inscrit dans une démarche globale, qui porte sur la maximisation du contrôle de l'utilisateur sur ses données personnelles, en nous appuyant sur le concept de « *collecte minimale* » de la directive européenne 95/46/CE, nous avons défini un protocole de négociation des politiques de traitement des données [Bek11], [Bek12] entre l'utilisateur et le SP. Ce protocole a pour objectif d'apporter plus de flexibilité aux échanges E-Commerce actuels basés sur le concept « *take it or leave it* ». Notre protocole permet d'établir des négociations de politiques par session. Toute négociation de politique porte sur les termes individuels qui y sont contenus, prenant en compte la sensibilité de chaque donnée mise en jeu, les préférences utilisateur, les pratiques d'usage déclarées, et le contexte situationnel de la transaction. Notre démarche a non seulement favorisé la flexibilité des échanges entre l'utilisateur et le SP, mais aussi, l'établissement de contrats de politiques d'une plus grande granularité.

Si aujourd'hui les pratiques d'usage des données sont contrôlées par la directive européenne 95/46/CE, l'estimation des données à révéler au SP est laissée quant à elle, à l'appréciation individuelle de chaque utilisateur. Or, voulant accéder à un service donné, l'utilisateur est souvent contraint de dévoiler plus de données personnelles qu'il ne le faut. Pour ce faire, nous avons proposé un protocole de négociation des données à base de théorie des jeux (section 5.5). Nous avons modélisé l'accès au service et la révélation des données comme un jeu, où les joueurs (l'utilisateur et le SP) cherchent à maximiser leurs gains. Notre protocole permet de trouver un point d'équilibre entre les deux joueurs correspondant à la quantité optimale de données personnelles à révéler, sans s'exposer à un risque élevé de violation de la vie privée. Notre protocole correspond à un processus décisionnel qui aide l'utilisateur à prendre des décisions quant à la révélation de ses données personnelles.

Nous avons développé plusieurs prototypes pour l'ensemble de nos contributions, accompagnés de scénarios applicatifs illustrant leurs rôles dans des interactions E-Commerce.

Bilan

L'objectif de cette thèse était de fournir des solutions techniques offrant à l'utilisateur la capacité de protéger, de manière pratique et efficace, ses informations à caractère personnel dans la mesure des limitations que nous nous sommes fixé (à savoir notamment la restriction à la « protection des données personnelles » telle que nous l'avons définie dans la section 1.1.1, et le positionnement dans un cadre de transactions électroniques).

Nous nous proposons dans cette section de fournir des éléments d'évaluation des fonctionnalités développées au cours de cette thèse vis-à-vis des contributions défendues. Ce travail d'évaluation a pour but d'estimer dans quelles mesures, les solutions techniques que nous avons mises en place sont adaptées aux problèmes que nous traitons.

Langage XPACML

Il est tout d'abord nécessaire d'évaluer la pertinence du modèle de génération de politiques XPACML proposé dans la section 2.7.

La déclaration des espaces de nommage présentés dans la même section offre la possibilité de :

- déclarer de nouveaux éléments de protection dans le schéma de politique initial,

- de spécifier à partir de ce nouveau schéma des restrictions en termes de valeurs possibles (voir fig.2.16).

Elle nous a permis de fait, une prise en compte des axes réglementaires que nous avons définis dans la section 1.3.1.

Pour la manipulation de ces déclarations, nous nous somme appuyés sur l'API Jaxb (section 2.7.1.3). Cette API possède deux caractéristiques principales, la possibilité de transformer des objets Java en une structure XML et vice versa. Ainsi, il nous a été possible de manipuler le schéma de politique XPACML que nous avons mis en place par le biais d'une hiérarchie de classes Java, et d'en faire les extensions/modifications souhaitées. Ces aspects offrent une modularité remarquable au schéma de politique XPACML, permettant de faire au besoin des extensions futures.

Il est à noter que les politique XPACML générées sont validées par un PolicyValidator que nous avons conçu (section 2.7.3) assurant ainsi la consistance des politiques générées par le biais de notre modèle.

Ensuite, le fait de passer par le modèle XACML au niveau architectural, renforce notre hypothèse de mise en application des politiques générées, confirmant ainsi la capacité de notre modèle à assurer une protection locale des données.

L'utilisateur quant à lui est muni d'interfaces faisant abstraction du procédé de génération de politiques. Ces interfaces lui permettent de spécifier de façon simplifiée ses préférences en termes de protection des données, pour une comparaison simple (figures 2.14, 4.6, 4.7), comme pour une négociation des politiques (figure 4.15). Ces interfaces font de notre démonstrateur un agent client.

Modèle sémantique

La mise en œuvre d'un modèle sémantique à travers une ontologie est en particulier digne d'attention à cause des possibilités qu'offre cet outil de modélisation. D'un côté, l'ontologie développée nous a donné la possibilité d'automatiser le procédé de génération des politiques XPACML [NTMS2012], notamment par le biais des sous graphes représentant les éléments du vocabulaire de politique (section 3.5.1.1). De l'autre, elle montre la capacité du modèle sémantique que nous avons implémenté à s'adapter à des

contextes situationnels différents (section 3.5.2). Nous pouvons affirmer ainsi, que nous avons répondu à notre objectif initial qui est de générer de façon dynamique et automatique des politiques basées sur les informations sémantiques liées à chaque situation.
Il est à noter, que l'ontologie de notre démonstrateur offre deux caractéristiques majeures :

- Celle de fournir un support de communication commun comme nous l'avons souligné dans l'introduction de ce mémoire de part le langage de développement OWL, qui est un outil de partage de vocabulaire très répandu,

- La capacité d'analyser les politiques à différents niveaux, selon le degré d'abstraction du vocabulaire utilisé dans la politique (section 3.6).

Protocole de négociation des politiques

Les algorithmes et méthodes de comparaison développés pour le PPDP (section 2.7.4) présentent une première étape vers l'établissement de notre protocole de négociation des politiques. En plus de l'automatisation de comparaison des politiques du SP avec les préférences de l'utilisateur, elle permet de doter l'utilisateur d'un historique des comparaisons des politiques conclues, comme celles qui n'ont pas abouti. Cet historique permet de prévenir l'utilisateur des éventuels regroupements des données sollicitées par le SP.

Par le développement du premier protocole de négociation (section 4.5), il nous a élé possible de montrer le caractère intuitif de l'interface utilisateur (figure 4.7), la simplicité relative du schéma de négociation (section 4.5.1), et de prouver l'utilité du classement des valeurs des éléments de politiques dans les comparaisons successives comme nous l'avons supposé dans l'introduction du chapitre 4. Le prototype nous a permis également de soulever quelques limitations liées à l'équité et la flexibilité dans la définition des ensembles de préférences de négociation (section 4.5.4).

Ces limitations étant identifiées, nous avons y remédier dans la conception et le développement de notre deuxième prototype (section 4.6). Ainsi, nous avons pu :

- Prouver l'hypothèse de flexibilité attendue en élargissant le périmètre de contrôle de l'utilisateur sur ses données personnelles,

- Marquer une équité entre lui et le SP en gardant à chacun la possibilité de ne pas communiquer sur son modèle de préférences.

Protocole de négociation des données

Ayant comme objectif la spécification de façon concise et efficace les interactions qui animent le processus de négociation des données. La modélisation formelle que nous

proposons dans la section 5.6.2 permet de visualiser les interactions stratégiques entre l'utilisateur et le SP, selon leurs modèles de préférences. Les calculs des stratégies optimales (section 5.6.3) par le biais de Maple, ont affirmé la convergence des stratégies de l'utilisateur et celles du SP vers un point d'équilibre, modélisant la révélation optimale des données nécessaire pour avoir accès au service offert avec une certaine qualité.

Les fondements mathématiques sur lesquels nous nous basons dans la modélisation font de notre solution, une réponse fiable permettant de préserver l'identité de l'utilisateur (par le biais des concepts privés définis dans la section 5.4), dans le cadre du scénario que nous avons mis en place.

Il est important de noter qu'on se limite à un ensemble prédéterminé et fini de quelques modèles d'interactions dans les scénarios que nous proposons. Notre utilisation à ces scénarios a mis en avant un besoin d'amélioration assez pressant, qui est l'intégration des divers modèles de fonctionnement (par exemple : le nombre d'interlocuteurs au sein du protocole de négociation).

La mise en œuvre des différentes contributions que nous proposons dans cette thèse n'a qu'une valeur de démonstration. Nos différents prototypes sont conçus pour mettre en évidence les possibilités apportées, en termes de fonctionnalités pratiques, par nos contributions. L'intégration dans une application destinée à un usage réel et immédiat différent nécessiterait plus ou moins d'ajustements suivant les modules considérés.

Perspectives

Sur la protection des données personnelles

Il convient de considérer que dans le cadre des travaux présentés ici, nous nous sommes limités à une certaine vision du problème de la protection de la sphère privée. Les résultats et questionnements qui découlent de cette recherche ouvrent des axes de travail plus précis, concernant des points délicats que nous avons soulevés, ou bien au contraire visant à étendre et élargir ces travaux à des problématiques connexes. Ces nouvelles perspectives de recherche portent sur la protection des données personnelles.

Une orientation de recherche qui nous semble intéressante au vu du bilan des travaux accomplis se rapporte de manière plus générale à la notion de données personnelles. Dans le cadre de la conception du modèle sémantique, nous avons nous-mêmes identifié et typé les données personnelles au sens du référencement dans une ontologie commune entre l'utilisateur et le SP. Cependant, la question du typage et de l'identification des données personnelles n'est pas du tout triviale. Il nous semblerait intéressant d'explorer les critères qui permettraient de considérer une information comme personnelle ou d'en déterminer la nature.

Aussi, un axe de travail intéressant consiste à s'affranchir d'une des limitations de nos travaux jusqu'à présent, qui consistait à considérer individuellement les données personnelles manipulées. La corrélation d'informations étant une menace majeure pour la vie privée, il nous semble nécessaire de la modéliser de manière appropriée. Le modèle sémantique permet en partie cela (de manière quelque peu implicite), de par les liens exprimés entre les traitements et les données (et entre les données elles-mêmes).

Sur la gestion de la sphère privée

La problématique des données non explicites est liée, au niveau du principe de raisonnement, à celle de l'inférence des données personnelles déjà mentionnée. Comme nous l'avons précisé dans notre définition de la protection des données personnelles (section 1.1.2.5 du chapitre 1), nous nous sommes limités à travailler sur des informations « explicitement représentées sous forme numérique et mises en jeu dans le cadre d'une application informatique ». Les présents travaux ne permettent donc pas de travailler sur des données propres à l'utilisateur, issues par exemple des traces laissées par l'utilisateur et dont il n'a pas explicitement connaissance tel son profil consommateur. Une protection plus efficace de la sphère privée de l'utilisateur exige donc un raisonnement sur les informations pouvant être produites ou déduites sur la base du comportement de l'utilisateur.

Enfin, lorsque nous nous sommes attelés à la définition de solutions côté utilisateur, nous avons travaillé sur l'hypothèse simplificatrice qui veut que le SP soit de confiance, et possède des outils « privacy by design » lui permettant de respecter la vie privée de l'utilisateur. Or, la plupart des architectures côté SP ne sont pas conçues dans un tel esprit. Il serait particulièrement intéressant de concevoir une méthodologie appliquée par un auditeur et permettant d'évaluer le bon respect de la vie privée d'un SP. Une telle méthodologie pourrait ainsi servir de base commune pour l'évaluation d'applications sur le plan de la protection des données personnelles.
D'une manière plus spécifique, il serait intéressant de développer un calcul formel pour raisonner sur les primitives spécifiques aux Trusted Computing Platforms. Si l'on pouvait décrire de manière formelle une architecture d'application, en précisant la localisation des éléments matériels et des processus cryptographiques, il serait possible d'associer à cette description une caractérisation des garanties et des risques (relatifs à la vie privée) associés à cette architecture. Une telle logique serait alors un outil puissant pour la conception et la certification d'architectures d'applications orientées vers la protection de la vie privée.

Ouverture

Nos travaux sur la protection de la sphère privée centrée utilisateur, laissent entrevoir des évolutions possibles dans les interactions entre les différents interlocuteurs potentiels.

D'un côté, il est possible de nommer une instance au niveau européen s'appuyant sur une structure juridique (exemple : G29) pour définir les politiques de protection des données

personnelles minimalistes/maximalistes en fonction des types de services, téléchargeables par les utilisateurs et SP pour intégration immédiate. Ainsi, la réactivité des SP aux différentes normes réglementaires s'en trouverait améliorée.

D'un autre côté, il est important d'approfondir les travaux sur les normes réglementaires sources des politiques de protection des données. En effet, le travail de représentation que nous avons fourni peut être complété avec des solutions à base d'intelligence artificielle pour faire un raisonnement sur les normes recueillies auprès de différentes autorités selon le contexte de la transaction en cours.

Enfin, les implémentations des protocoles de négociation que nous avons proposées, peuvent être étendues en prenant d'autres cas de figures possibles : une négociation entre plusieurs utilisateurs et un seul SP, une négociation entre un SP et un autre SP, …etc.

ANNEXE A

IMPLEMENTATION DU POLICY VALIDATOR

```
1  import java.io.File;
2  import java.io.IOException;
3  import javax.xml.parsers.DocumentBuilder;
4  import javax.xml.parsers.DocumentBuilderFactory;
5  import javax.xml.parsers.ParserConfigurationException;
6  import javax.xml.parsers.SAXParser;
7  import javax.xml.parsers.SAXParserFactory;
8  import org.w3c.dom.Document;
9  import org.xml.sax.ErrorHandler;
10 import org.xml.sax.SAXException;
11 import org.xml.sax.SAXParseException;
12 import org.xml.sax.helpers.DefaultHandler;
13
14 public class PolicyValidator {
15
16 static final String JAXP_SCHEMA_LANGUAGE = "http://java.sun.com/xml/jaxp/properties/
       schemaLanguage";
17 static final String W3C_XML_SCHEMA = "http://www.w3.org/2001/XMLSchema";
18 static final String JAXP_SCHEMA_SOURCE = "http://java.sun.com/xml/jaxp/properties/
       schemaSource";
19
20 static boolean isValid;
21 static {isValid = false;}
22
23 public static boolean isXMLValid(String xmlFile, String xsdFile)
24 {
25 isValid = true;
26 DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
27 dbf.setIgnoringComments(true);
28 dbf.setNamespaceAware(true);
29 dbf.setValidating(true);
30 dbf.setAttribute(JAXP_SCHEMA_LANGUAGE, W3C_XML_SCHEMA);
31 dbf.setAttribute(JAXP_SCHEMA_SOURCE, new File(xsdFile));
32 DocumentBuilder db = dbf.newDocumentBuilder();
33 db.setErrorHandler(new ErrorHandler() {
34 public void fatalError(SAXParseException e) {
35 System.out.println("Erreur de validation XSD - Erreur fatal");
36 isValid = false;
37 }
38 public void error(SAXParseException e) {
39 System.out.println("Erreur de validation XSD - Erreur");
40 isValid = false;
```

```
41 }
42 public void warning(SAXParseException e) {
43 System.out.println("Erreur de validation XSD - Warning");
44 isValid = false;
45 }
46 });
47 Document doc = db.parse(xmlFile);
48 } catch (ParserConfigurationException pcee) {
49 System.out.println(pcee);
50 return false;
51 } catch (IOException ioe) {
52 System.out.println(ioe);
53 return false;
54 } catch (SAXException saxe) {
55 System.out.println(saxe);
56 return false;
57  }
58 return isValid;
59 }
60 }
```

ANNEXE B

PROGRAMME MAPLE CALCULANT LES STRATEGIES OPTIMALES

```
> equilibreNash := proc(K1,K2,K3,K4,R)

local Pu, Ps, contrainte, `eu*`, `es*_tmp`, `es*`, i, j, es,
result, P1, P2;
# Définition des fonctions d'utilité
Pu:= (eu,es) -> 1/(K1+K2) * (K1*ln((eu*es)/(eu+es))-K2*eu):
Ps:= (eu,es) -> 1/(K3+K4) * (-K3*ln((eu*es)/(eu+es))+K4*es):
#contrainte (positive)
contrainte:= es-eu-R/2:
# Traitement de Pu
`eu*`:= solve( contrainte=0, eu):
#Traitement de Ps
`es*_tmp`:= solve(diff(Ps(`eu*`,es),es)=0,es):
j:= 0:
for i from 1 to nops([`es*_tmp`]) do
  if (Im(`es*_tmp`[i]) <> 0) then
    print(`Le résultat `||i||` est imaginaire`):
  else
    if (`es*_tmp`[i] < 0) then
      print(`Le résultat `||i||` est négatif`);
    else
      print(`Ca marche pour le résultat `||i);
      j := j+1:
      `es*` := `es*_tmp`[i]:
      print ((subs(es=`es*`,`eu*`),`es*`));
    end if:
  end if:
end do:
end;

> equilibreNash(1,1,3,5,1);
```

Ca marche pour le résultat 1

$$\frac{1}{2}, 1$$

Le résultat 2 est imaginaire

Le résultat 3 est imaginaire

ANNEXE C

SOLUTION DE L'EQUATION ALGEBRIQUE

La solution de l'équation comporte trois valeurs : les deux premières sont imaginaires donc à éliminer, la dernière est réelle positive, c'est la solution que nous garderons.

$$
\begin{aligned}
&S_1 \\
&= -\frac{\left(72\,K_3\,K_4{}^2R^2 + 64\,K_3{}^3 + 3\,R\sqrt{-3K_4{}^4\,R^4 + 528\,K_4{}^2\,K_3{}^2\,R^2 + 768\,K_3{}^4}\right)^{1/3}}{24\,K_4} \\
&- \frac{3\,K_4{}^2R^2 + 16\,K_3{}^2}{24\,K_4\left(72\,K_3\,K_4{}^2\,R^2 + 64\,K_3{}^3 + 3\,K_4\,R\sqrt{-3\,K_4{}^4\,R^4 + 528\,K_4{}^2\,K_3{}^2\,R^2 + 728\,K_3{}^4}\right)^{1/3}} \\
&- \frac{-4\;K_3 + 3\,K_4\,R}{12\,K_4} \\
&+ \mathbf{i}\,\frac{\sqrt{3}}{2}\left[\frac{\left(72\,K_3\,K_4{}^2R^2 + 64\,K_3{}^3 + 3\,R\sqrt{-3K_4{}^4\,R^4 + 528\,K_4{}^2\,K_3{}^2\,R^2 + 768\,K_3{}^4}\right)^{1/3}}{12\,K_4}\right. \\
&\left. - \frac{3\,K_4{}^2R^2 + 16\,K_3{}^2}{12\,K_4\left(72\,K_3\,K_4{}^2\,R^2 + 64\,K_3{}^3 + 3\,K_4\,R\sqrt{-3\,K_4{}^4\,R^4 + 528\,K_4{}^2\,K_3{}^2\,R^2 + 728\,K_3{}^4}\right)^{1/3}}\right]
\end{aligned}
$$

$$
\begin{aligned}
&S_2 \\
&= -\frac{\left(72\,K_3\,K_4{}^2R^2 + 64\,K_3{}^3 + 3\,R\sqrt{-3K_4{}^4\,R^4 + 528\,K_4{}^2\,K_3{}^2\,R^2 + 768\,K_3{}^4}\right)^{1/3}}{24\,K_4} \\
&- \frac{3\,K_4{}^2R^2 + 16\,K_3{}^2}{24\,K_4\left(72\,K_3\,K_4{}^2\,R^2 + 64\,K_3{}^3 + 3\,K_4\,R\sqrt{-3\,K_4{}^4\,R^4 + 528\,K_4{}^2\,K_3{}^2\,R^2 + 728\,K_3{}^4}\right)^{1/3}} \\
&- \frac{-4\;K_3 + 3\,K_4\,R}{12\,K_4} \\
&- \mathbf{i}\,\frac{\sqrt{3}}{2}\left[\frac{\left(72\,K_3\,K_4{}^2R^2 + 64\,K_3{}^3 + 3\,R\sqrt{-3K_4{}^4\,R^4 + 528\,K_4{}^2\,K_3{}^2\,R^2 + 768\,K_3{}^4}\right)^{1/3}}{12\,K_4}\right.
\end{aligned}
$$

$$-\frac{3\,K_4{}^2R^2+16\,K_3{}^2}{12\,K_4\left(72\,K_3\,K_4{}^2\,R^2+64\,K_3{}^3+3\,K_4\,R\sqrt{-3\,K_4{}^4\,R^4+528\,K_4{}^2\,K_3{}^2\,R^2+728\,K_3{}^4}\right)^{1/3}}\Bigg]$$

$$S_3 = \frac{\left(72\,K_3\,K_4{}^2R^2+64\,K_3{}^3+3\,R\sqrt{-3K_4{}^4\,R^4+528\,K_4{}^2\,K_3{}^2\,R^2+768\,K_3{}^4}\right)^{1/3}}{12\,K_4}$$

$$+\frac{3\,K_4{}^2R^2+16\,K_3{}^2}{12\,K_4\left(72\,K_3\,K_4{}^2\,R^2+64\,K_3{}^3+3\,K_4\,R\sqrt{-3\,K_4{}^4\,R^4+528\,K_4{}^2\,K_3{}^2\,R^2+728\,K_3{}^4}\right)^{1/3}}$$

$$-\frac{-4\;K_3+3\,K_4\,R}{12\,K_4}$$

BIBLIOGRAPHIE

[ACC06] C. Ardagna, J. Camenisch, M.Mont, S. Clauss, S. Crane, Y.Deswarte, G. Hogben, Siani Pearson, L. Pimenidis, T. Roessler, and D. Sommer. "An architecture for privacy-enhancing identity management". LAAS report 05206, LAAS-CNRS. Toulouse. France. 2006.

[Fer95] J. Ferber. "Les systèmes multi-agents, vers une intelligence collective". Inter-Editions. Paris. France. 1995.

[AEG06] C. Adam, F. Evrard, B. Gaudou, A. Herzig, and D. Longin. "Modélisation logique d'agents rationnels pour l'intelligence ambiante". Actes des 14èmes Journées Francophones des Systèmes Multiagents (JFSMA'06). Annecy, France. 2006.

[Alt07] Altman, Eitan, and R. El-Azouzi. "La théorie des jeux non-coopératifs appliquée aux réseaux de télécommunication". 2007.

[Baa03] S.Baase. A Gift of Fire. "Social, Legal, and Ethical Issues in Computing". Prenctice-Hall. 2003.

[Bek10] K. Bekara, Y. Ben Mustapha, and M. Laurent. "XPACML eXtensible Privacy Access Control Markup *Language* ". Second International Conference on Communications and Networking (ComNet'2010). ISDN 978-1-4244-8840-7. Tozeur. Tunisia. 2010.

[Bek11] K. Bekara, M. Laurent, and R.Millet. "Privacy Policy Negotiation at User's Side Based on P3P Tag Value Classification". EEE'11 - The 10th International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, Las Vegas, Nevada, USA, July 18-21, 2011

[Bek11a] K. Bekara, M. Laurent, "A Semantic Information Model based on the Privacy Legislation". 6th Conference on Network Architectures and Information Systems
Security. La Rochelle, France. 2011.

[Bek12] K. Bekara, M. Laurent, "Risk-Based Privacy Policy Negotiation Scheme ". The 8th International Conference on Security and Cryptography for Networks, September 5-7, 2012, Amalfi, Italy.

[Bek12a] K. Bekara, M. Laurent, T. H. Nguyen, "Technical Enforcement of European

Privacy Legislation: An Access Control Approach". Fifth International Conference on New Technologies, Mobility and Security NTMS 2012. Istanbul, Turkey. 2012.

[Ben03] M. Bennicke, P. Langendorfe. "Towards automatic negotiation of privacy contracts for internet services". The 11th IEEE International Conference on, page 319-324. 2003

[Bis] Bislider java component, https://bislider.dev.java.net/.

[Bor00] J.J. Borking. "Privacy incorporated software agent (pisa) : proposal for building a privacy guardian for the electronic age". In International Workshop on Design Issues in Anonymity and Unobservability. volume 2009/2001 of LNCS. pages 130–140, Berkeley. California. USA. 2000.

[Che03] H. Cheng, T Finin, and A. Joshi. "An Ontology for Context-Aware Pervasive Computing Environments". In IJCAI Workshop on Ontologies and Distributed Systems. IJCAI. 2003

[CPB03] M.C Mont, S.Pearson, and P. Bramhall. "Towards accountable management of identity and privacy : Sticky policies and enforceable tracing services". HPL-2003-49. HP Laboratories Bristol. 2003.

[Con05] OASIS Context Schema. http://www.oasis-open.org/committees/download.php/919/cs-xacml-schemacontext. 2005.

[Cor04] A. Corradi, R. Montanari, and D. Tibaldi. "Context-based Access Control for Ubiquitous Service Provisioning". pp. 444-451. In 28th Annual International Computer Software and Applications Conference (COMPSAC'04). 2004.

[CVJ08] L.Crépin, L. Vercouter, F.Jacquenet, Y.Demazeau, and O. Boissier. "Hippocratic multi-agent systems". The International Conference on Enterprise Information Systems (ICEIS'08). Pages 301–307. 2008.

[DA06] Y.Deswarte, and C.A. Melchor. "Sécurité des Systèmes d'Information, chapter Technologies de Protection de la Vie Privée sur Internet". pages 49–71. Hermès, Paris. France. 2006.

[Daw06] J. Dawn, P. Bodorik.P, and Y. Zhang. "PeCAN: An Architecture for Privacy-aware Electronic Commerce User Contexts". In Elsevier's Information Systems Journal, Vol. 31, Issue 4-5, pp. 295-320.2006

[Dhi07] D. Diehn, S. Berlik, and U. Kelter. "Enforcing privacy by means on an ontology driven XACML framework". Third International Symposium on Information Assurance and Security. 2007.

[DDM03] G. Danezis, R. Dingledine, and N. M. Mixminion. "Design of a type iii anonymous remailer protocol". In IEEE Symposium on Security and Privacy, pages 2–15. Oakland. CA. USA, 2003.

[Ent] Entêts P3P. http://publib.boulder.ibm.com/tividd/td/ITAME/SC32-1359-00/fr_FR/HTML/am51_webseal_guide30.htm

[Eur08] European Union's Sixth Framework Programme. PRIME - Privacy and Identity Management for Europe. IST-507591, 2004–2008. https://www.prime-project.eu/.

[Fer01] D.F. Ferraiolo, R.Sandhu, S.Gavrila, D.R. Kuhn, and R.Chandramouli. "Proposed NIST Standard for Role-Based Access Control". ACM TISSEC. 4(3), p. 222_274. 2001

[FM02] M. J. Freedman and R. Morris. "A peer-to-peer anonymizing network layer". In Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02). pages 193–206. Washington DC. USA. 2002.

[God05] S.Godik , and T. Moses. "extensible access control markup language (XACML) version 1.1". http ://www.oasis-open.org/committees/xacml/repository/cs-xacml-speci_cation-1.1.pdf, accédé en février 2005.

[Hat10] I. Hattak. "Analyse formelle des politiques de sécurité". http://w4.uqo.ca/dii/etudMaitrise/uploads/35.pdf

[Hog02] G. Hogben. "A technical analysis of problems with P3P 1.0 and possible solutions". W3C Workshop on the Future of P3P. 2002.

[Hor04] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML". W3C Member Submission.2004.

[Hu06] V. C. Hu, D. F. Ferraiolo, and D. R. Kuhn. "Assessment of Access Control Systems". National Institute of Standards and Technology (NIST). Technology Administration.U.S. Department of Commerce. Interagency Report 7316, 2006.

[Ide] Identity 2.0. http://www.identity20.com/.

[Int] Internet Engineering Task Force. IDsec : Virtual identity on the internet. http://idsec.sourceforge.net/.

[ISO99] ISO/IEC. "Information technology - security techniques - evaluation criteria for it security, part 2 : Security functional requirements". Technical Report 15408-2, International Organization for Standardization, 1999.

[Jes08] Jess: the Rule Engine for the Java Platform, http://www.jessrules.com/, Nov 2008.

[JL02] Xiaodong Jiang and James A. Landay. Modeling privacy control in context-aware systems. IEEE Pervasive Computing, 1, issue 3 :59–63, 2002.

[Kag03] L. Kagal, T. Finin, and A. Joshi. "A Policy Language for a Pervasive Computing Environment". In IEEE 4th International Workshop on Policies for distributed Systems and Networks, 2003.

[Kag04] L. Kagal. "A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments". Dissertation. 2004.

[Kag05] L. Kagal, and T. Berners-Lee. "Rein: Where policies meet rules in the semantic web". Technical report. MIT. 2005.

[KS02] G. Karjoth and M. Schunter. "A privacy policy model for enterprises". IEEE Computer Security Foundations Workshop. IEEE, IEEE Computer Society Press, 2002.

[Maa05] M. Maaser, P. Langendoerfer. "Automated negotiation of privacy contracts". In 29th Annual International Computer Software and Applications Conference. 2005.

[Mae09] M. Ashouri, S. Talouki, M. NematBakhsh, and A. Baraani, "k-Anonymity Privacy Protection Using Ontology". In 14th International CSI Conference . 2009.

[Mas] Massachusetts Institute of Technology. Kerberos : the network authentication protocol. http://web.mit.edu/Kerberos/.

[Mic08] M.Hecker, T.S. Dillon, and E.Chang. "Privacy ontology support for E-Commerce". IEEE Internet computing Journal.2008.

[Mü06] G. Müller. "Introduction of privacy and security in highly dynamic systems". Communications of the ACM, 49(9) :1013–1022. 2006.

[Mos04] T. Moses. "extensible access control markup language (XACML) version 2.0". Tech. rep. OASIS Committee Draft. 2004.

[Mys07] "MySQL database". http://www.mysql.com.2007

[Nam] "Namespaces in xml 1.0". Third edition. http://www.w3.org/TR/REC-xml-names/.

[Nas50] J. NASH. " Equilibrium points in n-person games". Procedings of the National Academy of Sciences,36:48–49. 1950.

[OAS05] "OASIS. OASIS eXtensible Access Control Markup Language (XACML) TC". http ://www.oasis-open.org/committees/tc_home.php ?wg_abbrev=xacml. 2005

[Ope] "Openid : an actually distributed identity system". http://openid.net/.

[Pat04] A. Patwardhan, V. Korolev, L. Kagal, and A. Joshi. "Enforcing Policies in Pervasive Environments". International Conference on Mobile and Ubiquitous Systems: Networking and Services. 2004.

[Pol05] "OASIS Policy Schema". http ://www.oasis-open.org/committees/download.php/915/cs-xacml-schemapolicy-01.xsd. 2005.

[Pro03] "Protégé Project". The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/. 2003

[Pre06] S.Preibusch. "Implementing Privacy Negotiations in E-Commerce". The 8th Asia-Pacific web conference. Harbin. China. 2006

[Ré78] République française. Loi numéro 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés. In Journal Officiel de la République Française, January 1978.

[Ré03] République française. Article 9. In Code civil, 1803.

[Ré04] République française. Loi numéro 2004-801 du 6 août 2004 relative à la protection des personnes physiques à l'égard des traitements de données à caractère personnel. In Journal Officiel de la République Française, August 2004.

[RB03] S. Riché and G. Brebner. "Storing and accessing user context". In 4th International Conference on Mobile Data Management, pages 1–12, Melbourne, Australia, january 2003.

[RBG02] S. Riché, G. Brebner, and M. Gittler. "Client-side profile storage". NETWORKING Workshops on Web Engineering and Peer-to-Peer Computing. pages 127–133. Pisa. Italy. 2002.

[Ros 11] E. Rosseto, and D.Caille. "Préserver sa vie privée lors des transactions électroniques". Projet de fin d'études-Télécom Sud-Paris. 2011

[SAM] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[San96] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. "Role-Based Access Control Models", Computer, 29(2), p.38-47. 1996.

[She08] S.Shervin, and Y. Abdulsalam. "Privacy and the Market for Private Data: A Negotiation Model to Capitalize on Private Data". In Computer Systems and Applications Conference, Doha, Qatar. 2008

[SB05] B. Subirana and M. Bain. "Legal Programming : Designing Legally Compliant RFID And Software Agent Architectures For Retail Processes and Beyond". Springer. USA. 2005.

[Sxi] "The sxipper plugin for firefox". http://www.sxipper.com/.

[Tao06] T. Yu, Y. Zhang, L. Kwei-Jay; "Modeling and Measuring Privacy Risks in OoS Web Services". The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on Digital Object Identifier. 2006.

[The] "The relationship between xacml and p3p privacy policies". http://labs.oracle.com/projects/xacml/XACML_P3P_Relationship.html.

[The95] The European Parliament and the Council. Directive 1995/46/EC of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. In European Union, editor, Official Journal of the European Communities, October 1995.

[The02] The European Parliament and the Council. Directive 2002/58/EC of the european parliament and of the council of 12 july 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector. In European Union, editor, Official Journal of the European Communities, July 2002.

[Thi00] R.Thibadeau. "Privacy server protocol: Short summary". http://yuan.ecom.cmu.edu/psp/SummaryInterop.pdf. 2000.

[Cra02] L.F. Cranor. "Web Privacy with P3P". O'Reilly & Associates, Inc., Sebastopol, CA, first edition. 2002

[Tho75] J. J. Thomson. "The right of privacy". Philosophy and Public Affairs, 4 :295–314, 1975.

[Tru03a] Trusted Computing Group. TCG main specification, version 1.1b. available via http://www.trustedcomputinggroup.org/, 2003.

[Tru03b] Trusted Computing Group. TCG TPM specification, version 1.2. Available via http://www.trustedcomputinggroup.org/, 2003.

[Uni] "Uniform resource identifier (uri): Generic syntax". http://www.rfc-editor.org/rfc/rfc3986.txt.

[Usz03] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and L. Lott. "KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement". In 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'03), page 93. 2003.

[Usz04] A. Uszok, J. Bradshaw, R. Jeffers, M. Johnson, A. Tate, J. Dalton, S. Aitken. "KAoS Policy Management for Semantic Web Services". In IEEE Intelligent Systems Journal, Vol. 19.2004

[Wan04] L.Wang, D.Wijesekera, and S. Jajodia. "A Logic-based Framework for Attribute based Access Control". 2nd ACM Workshop on FMSE. 2004.

[Web10] Web Ontology Language OWL, http://www.w3.org/2004/OWL. May 2010.

[WB90] Samuel D. Warren and Louis D. Brandeis. The right of privacy. Harvard Law Review, 4 :193–195, 1890.

[Wor04] World Wide Web Consortium. "Resource Description Framework(RDF) Model and Syntax Specification". http://www.w3.org/TR/rdf-syntax-grammar/. 2004.

[Wes67] A.F. Westin. "Privacy and freedom". New York. USA. 1967.

[Wor02] World Wide Web Consortium. "A P3P Preference Exchange Language 1.0 (APPEL 1.0)", http://www.w3.org/TR/P3P-preferences/.2002

[Wor06] World Wide Web Consortium. "Platform for Privacy Preferences specification 1.1., 2006. http://www.w3.org/P3P/ ».

[Wor99] World Wide Web Consortium. "Using P3P for e-commerce".. http://www.w3. org/TR/P3P-for-ecommerce". 1999

[Xml] XML, http://www.xml.org/

[You 09] B.Youakim, S.Layth, and F. Biennier. "A Security policy framework for context-aware and user preferences in e-services". Journal of System Architecture 55, 2009.

[Xpr] "Xpref: a preference language for p3p". http://www.sciencedirect.com/.