III

Utilisation des informations contextuelles pour assurer la sécurité d'un processus collaboratif distribué: un exemple dans l'e-Santé

I.1- Introduction

La pyramide des âges des pays industriels a profondément évolué lors de ces dernières décennies. Ce changement est principalement dû aux fantastiques progrès réalisés dans le domaine des diagnostics et des traitements médicaux. Ainsi depuis 1960, l'espérance de vie a augmenté de 5 ans pour les femmes et de près de 4 ans pour les hommes. Parallèlement, le taux de natalité a largement diminué dans la plupart des pays développés depuis 50 ans. Ces deux facteurs ont eu pour conséquence de faire croître drastiquement la moyenne d'âge de la population. Ainsi, à l'horizon 2050, l'Union Européenne verra sa population de personnes âgées de plus de 80 ans augmenter de plus de 180% passant de 18.8 millions aujourd'hui à près de 35 millions (7).

Ce vieillissement mais aussi l'évolution de la structure familiale modifient profondément l'équilibre de la société : ces facteurs poussent à améliorer l'accompagnement à domicile et la prise en charge sociale et médicale. Les technologies de l'information et de la communication offrent en particulier la possibilité d'une liaison virtuelle quasi-permanente entre le patient d'une part et les différents acteurs du monde médical d'autre part. Ces technologies permettent d'entrevoir l'opportunité de radicalement améliorer le diagnostic, le traitement et la surveillance des malades - ce qui dans ce domaine participera au bien-être de l'humanité – mais aussi de rationaliser l'utilisation des moyens et *in fine* de réduire les coûts tout en offrant des services personnalisés et performants. Ce dernier point est d'une importance primordiale afin de résoudre les problèmes auquel se trouvent confrontés les services médicaux de nombreux pays. L' « e-Santé » est souvent présentée comme la solution magique à ces défis : qu'en est il plus précisément ?

L'*e-Santé* est une discipline émergente englobant l'usage de l'informatique médicale et des réseaux de télécommunication afin de fournir des infrastructures numériques de santé publique permettant une collaboration renforcée des acteurs professionnels et associatifs. Au sens large, l'e-Santé caractérise non seulement l'apport de la technologie mais aussi un état d'esprit, une attitude pour la résolution des problèmes de santé -via une approche globale et en réseau- grâce à l'usage de

véritables plateformes intégrées de collaboration au niveau local, régional et mondial.

Examinons plus précisément avec Eysenbach (12) quels peuvent être les bénéfices de l'e-Santé:

- Efficacité des soins – une des promesses de l'e-Santé est d'augmenter l'efficacité des soins tout en réduisant les coûts. Une des pistes pour cette réduction peut être d'éviter la duplication des actes, notamment lors du diagnostic, en partageant de manière plus efficace l'information entre les différents intervenants et en plaçant le patient au centre du processus.
- Meilleure qualité des soins – au-delà de la réduction des coûts, une meilleure collaboration et l'utilisation de technologies plus performantes pourront améliorer la qualité même des soins.
- Meilleure information des patients – la mise à disposition du patient et de son entourage des bases de données médicales, des résultats de ses examens et des analyses de ses médecins permet une meilleure compréhension et un choix avisé parmi les moyens de thérapies (méthodes, médecins, centres d'excellence …).
- Nouvelle relation patient-services de santé – où la décision est prise dans une relation de partenariat,
- Échange sécurisé des informations médicales – la standardisation des formats de données et l'utilisation de réseaux sécurisés et de techniques cryptographiques pour l'échange d'informations médicales entre les différents acteurs garantira le respect de la vie privé des citoyens.
- Élargissement de la sphère des services de santé – les limitations conventionnelles tant du point de vue géographique que du point de vue conceptuel et de l'expertise deviendront obsolètes en permettant le développement de techniques d'information, de surveillance, voire même d'interaction à distance (par exemple opérations).

Au delà de ces caractéristiques essentielles, l'e-Santé se doit aussi d'être facile à utiliser, confortable, parfois amusante, voire même excitante afin de permettre une pénétration importante et une acceptation rapide de la part des patients et des acteurs du monde médical.

Dans la suite de ce chapitre, nous montrerons comment les résultats du projet MOSQUITO ont été utilisés dans le cadre de l'e-Santé. Les prototypes développés illustrent comment notre architecture permet l'intégration des informations médicales et facilite la collaboration des différents acteurs dans deux situations particulières de surveillance médicale à distance. Au-delà des aspects fonctionnels largement évoqués dans cette introduction, le projet MOSQUITO vise à la prise en compte des contraintes de sécurité, de confiance et de respect de la vie privée trop souvent ignorées.

I.2- Vers une e-Santé Ubiquitaire

Une part importante des efforts consacrés à l'e-Santé consiste en la surveillance médicale à distance des populations à risque. La surveillance médicale à distance, notamment dans le cas des personnes âgées, permet de connaître à tout moment de la journée les paramètres physiologiques du patient tout en lui permettant de continuer son activité. Cette approche permet de retarder voire d'annuler le recours à une hospitalisation coûteuse et psychologiquement traumatique.

Notre premier scénario illustre l'utilisation d'informations contextuelles afin de garantir l'accès aux données médicales du patient en situation d'urgence. Le second scénario détaille un processus de collaboration sécurisé entre personnels médicaux dans le cadre d'une surveillance médicale à distance.

I.2.1- Scénario 1 – Urgence

Jacques profite d'un splendide week-end à la campagne. A bord de son canot, il pêche au milieu d'un superbe lac. Soudain, il est pris d'un malaise et perd rapidement connaissance. Son système personnel de surveillance médical, consistant d'un ensemble de capteurs (température, pouls, tension, etc.) et d'un terminal communicant, détectant une anomalie, envoie un signal d'alerte vers le centre de secours. Le centre tente alors de contacter Jacques sur son téléphone portable. En l'absence d'une quelconque réponse, la position de Jacques est déterminée par le système et une équipe médicale est envoyée à son secours. Comme il s'avère être au milieu d'un lac, le centre décide aussi de dépêcher un hélicoptère. Après avoir rapidement retrouvé Jacques encore inconscient, le docteur s'identifie auprès de son terminal et accède rapidement à la médication actuelle de Jacques, ses allergies ainsi qu'à l'historique de ses antécédents médicaux. Tous les symptômes et le dossier de Jacques indiquent qu'il s'agit d'une crise cardiaque, le docteur commence les premiers soins en toute connaissance de cause. Tout laisse à penser que la rapidité des secours à permis de lui sauver la vie.

I.2.2- Scénario 2 –Télésurveillance médicale à domicile

Bob, un patient à mobilité réduite qui demeure très souvent à son domicile s'est abonné à un service de télésurveillance médicale. Son état de santé est évalué en permanence par un ensemble de capteurs. Bob sait que ces données sont enregistrées en continu aussi le contrat entre le fournisseur de télésurveillance et Bob lui garantit le caractère confidentiel de toutes les données médicales collectées.

Le logiciel de télésurveillance analyse les données et s'assure qu'une réponse adaptée est déclenchée en toute circonstance. Très souvent la première étape consiste à dépêcher un docteur à son domicile. Via un service de découverte sécurisé (39), un docteur est choisi parmi une liste de médecins collaborant avec le fournisseur. Comme la rapidité de la prise en charge peut être primordiale, le médecin choisi doit confirmer la visite avant de se rendre au domicile. Afin d'assurer un suivi efficace même en l'absence du médecin traitant habituel, toutes les informations nécessaires sont stockées dans le Portail d'Information Médical (PIM). L'accès à ces données nécessite l'acceptation de la visite par le docteur mais

aussi la proximité du patient et du docteur (dans le prototype cette proximité est évaluée via GPS). Une fois le diagnostic établi, l'ordonnance électronique délivrée par le docteur est immédiatement envoyée à la pharmacie la plus proche. La disponibilité des médicaments est ensuite automatiquement notifiée à Bob.

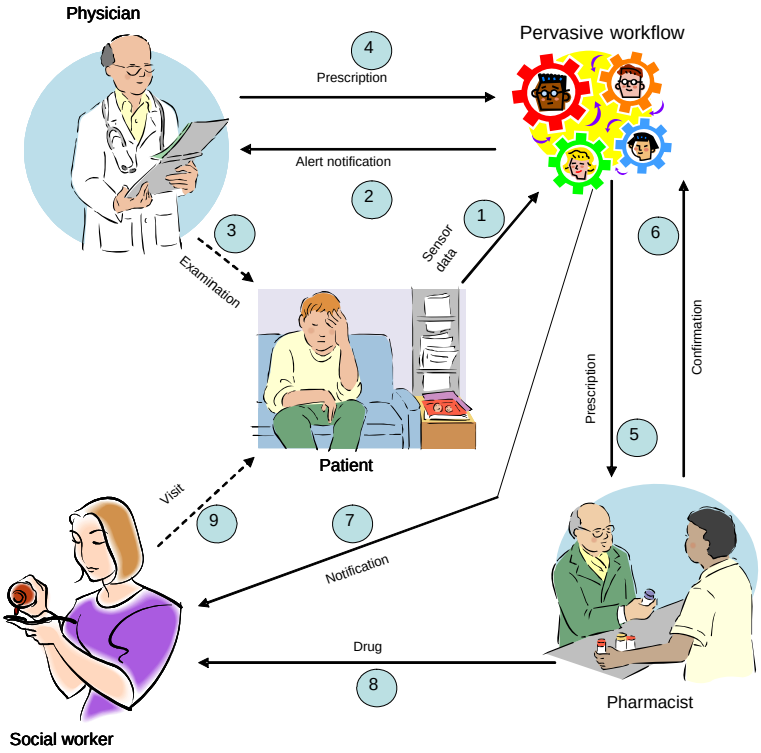

Figure 1: Exemple de télésurveillance médicale à domicile

Ce dernier peut ensuite contacter une personne de confiance, par exemple un membre de sa famille ou une assistante sociale, afin de récupérer les médicaments et de les lui apporter à domicile. La personne choisie reçoit alors une notification sur son téléphone. Après avoir accepté cette tâche, elle peut se rendre à la pharmacie qui a été prévenue de son identité et peut ainsi la vérifier en utilisant les moyens standards (carte d'identité, etc.).

I.2.3- Plateforme Logicielle et Sécurité

Un système d'e-Santé ne peut se contenter de supposer naïvement l'honnêteté des acteurs et de son environnement. Fraudes aux systèmes et mutuelles de santé[1],

[1] **PARIS, 19 juin 2006 (AFP):** *Au moins 50 M EUR de fraude par des établissements de soins en 2005 (CNAM). Plus de 300 hôpitaux et cliniques français auraient pratiqué en 2005 des surfacturations, percevant indûment au moins 50 millions d'euros, selon des estimations de la Caisse nationale d'assurance maladie (CNAM) communiquées lundi. Ces estimations, révélées par le Journal du dimanche et confirmées lundi à l'AFP par la CNAM, résultent d'une campagne de contrôle menée par un ciblage*

comportements discriminatoires envers des malades et non-respect de la vie privée sont autant de menaces : les données médicales sont extrêmement sensibles et doivent être particulièrement protégées. Cela devient encore plus important dans le contexte où ces données sont distribuées et lorsque plusieurs acteurs communiquent par des infrastructures ad hoc. Dans ces conditions, contrôler que tout accès et toute utilisation sont licites devient un véritable casse-tête, ceci d'autant plus lorsque dans certains cas – notamment lorsque la vie humaine est en danger – il est nécessaire de garantir un accès inconditionnel. L'utilisation du contexte – rôle mais aussi paramètres physiologiques ou phase d'exécution des soins – permet de prendre en compte de façon précise les circonstances d'accès. Cette utilisation pose cependant le problème de la confiance à accorder à leur fournisseur et à l'information proprement dite.

La solution préconisée par MOSQUITO est d'utiliser les informations collectées par des capteurs matériels mais aussi logiciels afin de les intégrer dans le processus logiciel. L'intégration de ces informations contextuelles permet, de façon conventionnelle, d'adapter les aspects fonctionnels de l'application mais aussi d'adapter la sécurité de l'application. Plus précisément, les informations contextuelles sont utilisées dans les politiques de sécurité de l'application qui contrôlent l'accès aux données médicales d'un patient. Comme nous le détaillons dans la section suivante, l'usage d'informations contextuelles nécessite un traitement particulier afin de garantir leur fiabilité et la possibilité d'un passage à l'échelle. Une seconde originalité de MOSQUITO est de proposer le déploiement des applications sous une forme diffuse grâce à une Architecture Orientée Services (*SOA*). Cette approche rend possible l'exécution d'applications entre clients nomades, dans laquelle les dispositifs mobiles ne se limitent pas aux rôles de simples clients de services ambiants mais partagent en plus leurs propres ressources logicielles dans le cadre de l'exécution de processus collaboratifs. La preuve d'exécution des processus précédents constitue une partie du contexte appelée contexte d'exécution : cette donnée est utilisée, en plus des autres informations contextuelles, dans l'évaluation des politiques de sécurité afin de garantir un accès licite aux ressources.

L'infrastructure nécessaire à la réalisation d'applications d'informatique diffuse médicale se doit de supporter un large ensemble de technologies englobant les réseaux sans fil, les terminaux et environnements d'exécution nomades et des capteurs hétérogènes. L'utilisation de plateforme logicielle standard, tel que les Architectures Orientées Services, offre potentiellement la solution la moins onéreuse et la plus flexible pour le développement de solutions distribuées. Le choix d'employer ces plateformes et notamment les Web Services a deux conséquences principales : d'une part l'intégration est rapide et aisée avec les écosystèmes logiciels ou systèmes d'entreprise existants tout en utilisant des environnements de développement éprouvés ; d'autre part, cette technologie impose des contraintes importantes en termes de communication et de calcul pour des matériels à faible coût. Ainsi, il est actuellement impossible d'exécuter le framework MOSQUITO et plus généralement l'ensemble complet de protocoles Web Services Sécurisés sur des

---

informatique d'anomalies sur les factures adressées à la Sécu dans le cadre de la mise en place de la tarification à l'activité (T2A).

dispositifs légers tel que Smartphone ou PDA. Cette limitation est encore plus criante lorsque l'on utilise des capteurs. Elle est pour part due à l'absence de librairies standards sur des architectures exotiques et pour une autre part à l'incapacité de certains dispositifs à mettre en œuvre des processus parfois lourds (XML, chiffrement, …).

I.3- Informatique Contextuelle

L'informatique ubiquitaire ou diffuse fait référence au phénomène d'informatisation de la vie quotidienne. Les objets les plus anodins se voient dotés de capacités de calculs et interagissent afin que l'accès à l'information soit possible dans n'importe quel lieu et à n'importe quel moment (9). L'utilisation d'informations contextuelles représente un avantage non négligeable pour des applications évoluant dans les environnements ou l'informatique est omniprésente. Le contexte ou information contextuelle définit tous types de données caractérisant une entité (un utilisateur, un composant électronique ou un logiciel) (2). La position d'un dispositif, la bande passante du réseau ou encore les protocoles de sécurité disponibles sur une plateforme sont autant d'exemples d'informations contextuelles caractérisant un service ou un réseau. Cependant, l'intégration du contexte au niveau applicatif ne va pas sans poser des problèmes de traitement dus à la grande quantité d'informations de nature très différente qui peuvent être acquises depuis des sources variées.

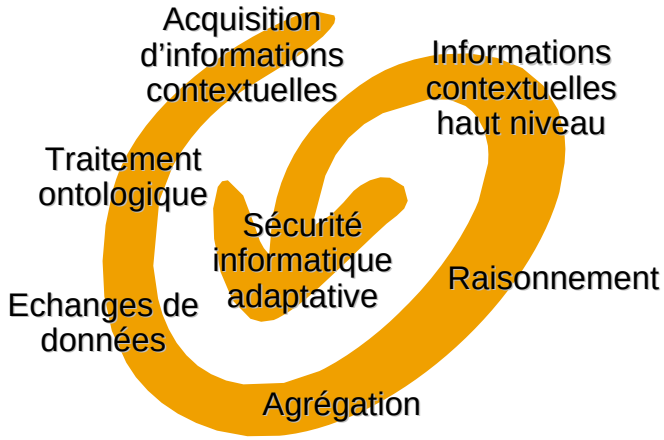

Figure 2: Cycle des vies des informations contextuelles

Dans la suite de cette section, nous détaillerons les différentes étapes qui permettent l'adaptation des politiques de sécurité par la construction d'une information contextuelle fiable et digne de confiance. Ces différentes étapes sont schématisées dans la figure ci-dessus (Figure 2: Cycle des vies des informations contextuelles).

I.3.1- Représentation du Contexte

L'utilisation et le partage d'informations contextuelles acquises directement d'un capteur restent délicats : leur format est trop souvent limité à l'essentiel (une suite de bits) et plus significatif des composants utilisés pour sa mesure que d'une sémantique qui y serait associée. L'utilisation d'une ontologie commune permet de pallier en grande partie ce problème. Une ontologie facilite la gestion de la connaissance en définissant une taxonomie des objets d'un domaine donné. Ainsi, dans une ontologie, les objets sont classifiés et caractérisés par le biais de propriétés. Ces concepts sont en outre organisés dans un graphe définissant les relations sémantiques entre concepts (spécialisation, agrégation, similarité, synonymie, etc.) (16)(38)(45). L'utilisation d'une ontologie pour les informations de contexte permet ainsi de définir clairement la syntaxe et la sémantique associées à chaque information de contexte, mais également de faciliter le raisonnement sur le contexte (8).

De nombreuses ontologies ont été définies pour décrire les informations contextuelle, parmi lesquelles CONtext Ontology (CONON (44)), SeMETHer (45) Context OntoLogy et CoOL (16). CONON et SeMETHer réutilisent des ontologies existantes. Par exemple, SeMETHer utilise SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) (14) pour les concepts de localisation et de temps, et W3C DISelect (31) ou FIPA Device (11) pour la représentation des informations relatives aux composant logiciels ou électroniques.

Afin de répondre à nos besoins, nous avons décidé d'utiliser CoOL comme ontologie pour le contexte. CoOL propose quatre catégories d'informations contextuelles : utilisateur (activité, rôle, préférence), environnement (température ambiante, position), plateforme (électronique ou logicielle) et service (description de service, protocoles supportés) (16). La Figure 3 fournit une vue d'ensemble de l'ontologie CoOL reprenant les catégories de contexte précédemment citées.

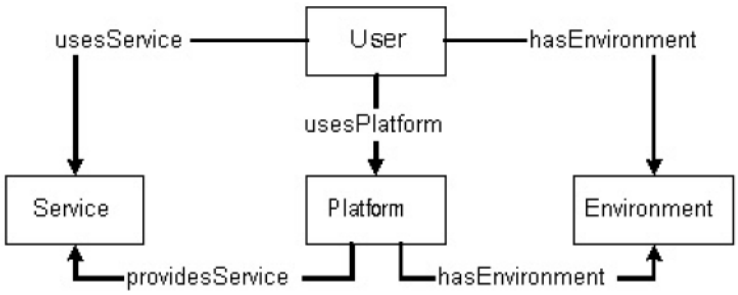

Figure 3: Vue d'ensemble de l'ontologie CoOL

CoOL s'appuie sur le langage OWL (Web Ontology Language voir (41)). OWL est un dialecte XML pour la représentation et l'instanciation de la connaissance formalisée dans une ontologie. OWL définit trois sous-langages à la complexité croissante: OWL Lite, OWL DL, OWL Full. OWL lite fournit une hiérarchie basique entre les objets et se limite à une unique relation entre deux objets. OWL DL est basé sur la logique descriptive (8). Cette dernière décrit un

domaine donné en termes de concepts (classe), rôles (relation) et individus (objet). OWL DL permet ainsi de décrire des règles d'inférence simples entre concepts tel que « Si Bob a un pouls inférieur à 10, alors il est inconscient.». OWL DL permet des raisonnements complets et décidables. De plus, il supporte tous les opérateurs définis dans OWL et n'a aucune restriction concernant la cardinalité des relations entre objets. Enfin, OWL Full offre une plus grande expressivité (par exemple une classe peut être une instance d'une autre classe) mais n'offre pas de garantie en termes de calcul : il n'existe aucun algorithme d'inférence décidable pour OWL Full. Dans un souci de compromis entre efficacité et pouvoir d'expression, nous utilisons OWL DL pour représenter les concepts définis dans CoOL. La Figure 4 illustre la représentation d'un concept de position GPS en OWL.

```
<cool:GPSLocationElement
xmlns:cool="http://discovery.service.com/cool#">
  <cool:Altitude>0.0</cool:Altitude>
  <cool:Latitude_Degree>44</cool:Latitude_Degree>
  <cool:Latitude_Minute>4</cool:Latitude_Minute>
  <cool:Latitude_Second>28.206604</cool:Latitude_Second>
  <cool:Longitude_Degree>123</cool:Longitude_Degree>
  <cool:Longitude_Minute>7</cool:Longitude_Minute>

<cool:Longitude_Second>44.2411</cool:Longitude_Second>
</cool:GPSLocationElement>
```

Figure 4: Représentation en CoOL d'une position GPS

I.3.2- Raisonnement sur le contexte

Comme nous venons de le décrire, l'interprétation des informations contextuelles dans une ontologie permet de s'abstraire des contingences physiques et, par là même, de permettre le raisonnement sur le contexte : à partir d'informations telles que la position d'un utilisateur, des informations de contexte plus complexes, comme la proximité entre deux utilisateurs, peuvent être inférées par raisonnement quel que soient le type et la marque des capteurs utilisés. Cette approche permet d'améliorer grandement la flexibilité des systèmes contextuels, notamment leurs capacités collaboratives.

Afin de raisonner sur le contexte, nous avons fait appel à deux techniques complémentaires : le raisonnement basé sur les ontologies et les règles d'inférence. Comme décrit précédemment, les ontologies supportent la définition de la relation entre informations de contexte. Ainsi, à la Figure 5, nous exprimons en OWL-DL le raisonnement suivant : « si le patient a un pouls et une température corporelle inférieur à 10, alors il est inconscient ». Nous établissons une relation entre le pouls et la température du patient pour inférer l'état de santé d'un patient.

```
<xsd:simpleType name=below10>
<xsd:restriction base=xsd:positiveInteger>
<xsd:maxInclusive value=10>
</xsd:restriction>
</xsd:simpleType>
<owl:Class rdf:ID=isUnconscious>
  <owl:intersectionOf rdf:parseType=Collection>
    <owl:Class rdf:about=#User/>
    <owl:Restriction>
      <owl:onProperty rdf:resource=#Pulse>
      <owl:someValuesFrom rdf:resource=#below10>
    </owl:Restriction>
  </owl:intersectionOf>
  <owl:intersectionOf rdf:parseType=Collection>
    <owl:Class rdf:about=#User/>
    <owl:Restriction>
      <owl:onProperty rdf:resource=#BodyTemperature>
      <owl:someValuesFrom rdf:resource=#below10>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Figure 5: Exemple de raisonnement avec OWL

De plus, comme une ontologie permet d'exprimer des relations de similarité entre des informations de contexte, nous pouvons ainsi exprimer une relation entre le pouls et le rythme cardiaque : si le pouls n'est pas disponible, le rythme cardiaque peut alors être utilisé pour dériver l'état de santé du patient.

Néanmoins, le pouvoir d'expression d'OWL DL est rapidement limité dès lors que nous devons exprimer des raisonnements complexes plus à même de traduire des raisonnements médicaux réalistes. A cause des restrictions imposées par OWL DL, un raisonnement basé uniquement sur une ontologie est limité à des relations entre les informations de contexte. Pour cette raison, nous ne pouvons pas quantifier une relation définie en OWL DL. Pour ces raisons, en complément du raisonnement basé sur l'ontologie, nous proposons de définir des règles d'inférence et d'utiliser un moteur d'inférence comme Jess (17). Une fois les relations entre les informations contextuelles identifiées, les règles d'inférence permettent de surmonter les restrictions du langage de représentation en évaluant et quantifiant les relations entre les informations. Dans notre cas, la proximité entre un docteur et son patient est un exemple simple des capacités offertes par cette alliance entre ontologie et règles. A la Figure 6, nous définissons cette règle en Jess (l'acquisition de la position a été intentionnellement omise).

```
(defrule isCloseTo
   ;; LOOK FOR A PHYSICIAN's LOCATION ;;
   …
   ;; LOOK FOR A PATIENT's LOCATION ;;
   …
   ;; GET LATITUDE, LONGITUDE AND ALTITUDE OF
PHYSICIAN's LOCATION ;;
   …
   ;; GET LATITUDE, LONGITUDE AND ALTITUDE OF
PATIENT's LOCATION ;;
   …
   ;; TEST IF THEIR DISTANCE IS HIGHER THAN 20 ;;
   (test (< (distanceInM ?PhysicianLocation ?PatientLocation)
20))

   =>

   ;; ASSERT THE RELATIONSHIP BETWEEN THE
PHYSICIAN AND PATIENT ;;

   (assert
   (triple
   (p"http://www.owl-
ontologies.com/unnamed.owl#isCloseTo")
   (s ?PatientLocation)
   (o ?PhysicianLocation)
   )
   )
```

Figure 6: Règle d'inférence en Jess [17]

Dans cet exemple, la position du docteur et du patient ainsi que leurs rôles sont représentés sous format RDF avec un triplet {prédicat, sujet, objet}. Nous utilisons le modèle de données RDF afin de communiquer entre l'ontologie et le moteur d'inférence. Ainsi, l'instance OWL générée peut être aisément traduite en RDF pour le raisonnement basé sur des règles d'inférence. Cette règle d'inférence définit la proximité entre le patient et un docteur. Deux éléments, séparés par le symbole =>, composent la définition de la règle. Dans sa partie gauche, nous acquérons en premier les positions des utilisateurs dont les rôles sont respectivement docteur et patient. Puis nous vérifions si la distance entre ces deux positions est inférieure à 20 mètres. Si tel est le cas, la partie droite de la règle enrichit l'ontologie d'une nouvelle instance de la relation *isCloseTo* entre le docteur et le patient.

I.3.3- Acquisition Sécurisée de Contexte

L'utilisation du contexte dans le cadre de la sécurisation d'applications soulève des problèmes de validité et de confiance concernant les informations obtenues

(19)(35). Les sections suivantes présentent différentes approches afin d'assurer la sécurité et la confiance vis-à-vis des informations de contexte. L'utilisation d'information contextuelle dans les applications de l'e-Santé, et plus particulièrement l'agrégation et le raisonnement, introduit plusieurs challenges quand à l'évaluation de la confiance et la fiabilité de l'information. Ainsi, l'information contextuelle peut provenir de différentes sources, qui sont toutes caractérisées par une fiabilité et un degré de confiance propres. Par exemple, certaines informations peuvent être émises par des capteurs certifiés et résistants fabriqués par une marque reconnue, alors qu'une autre information peut provenir d'un capteur bon marché manufacturé par un obscur sous-traitant et n'offrant aucune garantie de fiabilité. Notre approche vise à réduire cette hétérogénéité afin de traiter de façon identique et pondérée cette information.

I.3.3.1- Sécurité

Concernant la sécurisation des informations de contexte, nous nous concentrons sur la confidentialité et l'intégrité des informations de contexte.

Confidentialité des informations de contexte

Pour des raisons de respect de vie privée et de confidentialité, il est important de protéger les informations médicales des patients, telles que résultats d'analyse, médications, etc. Les préférences de l'utilisateur permettent de définir quels types d'informations sont à protéger. Jason (15) propose une architecture appelée Confab qui garantit la confidentialité des informations personnelles dans un environnement ubiquitaire. Cette architecture se concentre plus particulièrement sur la protection de la localisation. Confab est basé sur l'analyse des besoins de l'utilisateur en matière de confidentialité dans le but d'acquérir, manipuler ou stocker les informations contextuelles. Les utilisateurs peuvent ainsi contrôler la divulgation de leur localisation. Shankar (24) introduit quant à lui la notion de Context View. Une View est un sous ensemble du contexte général de l'entité utile à l'application. Enfin, Bussard et al. (20) proposent des mécanismes cryptographiques pour le contrôle sélectif de la divulgation d'informations de contexte.

Intégrité des informations de contexte.

L'intégrité doit garantir que les informations de contexte acquises ne sont pas corrompues par une tierce partie. Les mécanismes de hachage et de signature à base de clé publique peuvent par exemple remplir cet objectif. Si une fonction de hachage à clé secrète est utilisée pour l'authentification de messages, alors il s'agit de permettre le partage sécurisé de ladite clé entre le fournisseur d'information de contexte et le système contextuel. Le déploiement des clés sur des systèmes très fortement décentralisés constitue un problème à part entière, dans la mesure où les infrastructures à base de clés publiques imposent de trop nombreuses contraintes, en particulier sur des capteurs dont les capacités de calcul et de communication sont réduites.

I.3.3.2-Information de Contexte de Confiance

Dans le cadre des systèmes contextuels, la confiance fait référence à la fiabilité et à la précision des informations de contexte. Évaluer la confiance du contexte consiste à déterminer la marge d'erreur entre l'information de contexte obtenue et le contexte réel. Dans ce but, nous proposons plusieurs approches décrites ci-dessous.

Analyse statistique des informations de contexte

L'analyse statistique des valeurs d'informations de contexte permet de détecter des valeurs aberrantes. Dans le cas de la température ambiante, il est par exemple possible de faire la moyenne des plusieurs mesures de températures dans une même pièce provenant de plusieurs thermomètres. Notre objectif est alors de détecter les capteurs les plus fiables. Un telle approche, évidemment très naïve, soulève de nombreux problèmes : si les valeurs de température collectées sont {10 ; 10 ; 11 ; 50}, nous obtenons une moyenne de 20 degrés. Il est cependant plus probable que la dernière valeur soit erronée, et que la température de la pièce est plus proche de 10 degrés. Afin de résoudre ce problème, nous pouvons calculer la valeur médiane plutôt que la moyenne afin d'éliminer les valeurs qui s'éloignent trop de cette valeur médiane. En pratique, l'implémentation de telles approches requiert des analyses statistiques assez sophistiquées dépendant des grandeurs physiques mesurées.

Réseau de Réputation Distribuée

La confiance peut-être simplement décrite comme l'attente d'une personne ou d'un système sur les actions d'autrui. Elle est utilisée souvent par celle-ci afin de choisir un ensemble de partenaires afin d'aboutir à un comportement coopératif (32).

Son évaluation est habituellement basée sur des expériences passées avec la même entité. Malheureusement, dans le cadre de coopérations dans un environnement ubiquitaire, il est difficile de garantir une connaissance préalable. Afin d'établir une relation de confiance avec une autre entité, une alternative est de se baser sur la réputation de cette dernière. La réputation synthétise la collection de bonnes ou mauvaises expériences accumulées par différentes entités. La confiance accordée à une information de contexte est alors dépendante de la confiance en sa source. Il est donc nécessaire de pouvoir authentifier la source de l'information afin de pouvoir en dériver la confiance dans l'information de contexte. Les fondations mathématiques de cette approche se trouvent dans les statistiques et les probabilités (13). Robinson (29) introduit le concept de *Trust Context View* afin de déterminer la confiance dans une classe de contexte donnée. Cette approche ne considère pas l'authentification de la source de l'information mais permet de distribuer à grande échelle une notion de confiance dans un milieu ubiquitaire. Dans le cadre de l'architecture Gaia, Al-Muthadi et al (30) définissent une notion de valeur de confiance dans le processus d'authentification. Une valeur de confiance peut être établie en se basant sur les techniques d'authentification utilisées.

Valeur de Confiance.

Il est à noter que le raisonnement sur les informations de contexte qui les *agrège* soulève de nouvelles questions concernant la confiance de la nouvelle

information. Par exemple, si l'on peut dériver de la valeur du pouls et de la température corporelle l'état de santé d'un patient, quelle confiance attribuer à cette nouvelle information, notamment si la confiance en la valeur du pouls et en la température corporelle est différente ? L'utilisation de mécanismes de réputation utilisant les résultats de la logique subjective (1) peut constituer une solution possible.

I.3.3.3- Autorisation et contrôle d'accès contextuel

Afin de décider si un utilisateur est autorisé à accéder à une ressource d'un système, un service de contrôle d'accès applique une politique de contrôle d'accès. Dans le cadre d'un contrôle d'accès contextuel, la décision de permettre ou de refuser l'accès ne se base pas uniquement sur des propriétés statiques telles que le rôle ou les autorisations associées à un utilisateur mais également sur des informations contextuelles telles que la position ou l'état de santé. Dans le cadre de MOSQUITO, le contrôle d'accès est effectué au niveau de l'invocation de message. Ainsi la politique d'accès définit des règles en fonction des méthodes offertes par un web service. L'utilisateur qui demande accès à une méthode donnée devra fournir les pièces nécessaires – les informations de contexte – requises par la politique.

Il existe dans la littérature de nombreuses approches pour le contrôle d'accès contextuel :

Rôle Environnementaux : Covington et al (6) proposent une architecture pour le contrôle d'accès a base de rôles environnementaux étendant le modèle de contrôle d'accès à base de rôles (RBAC). Dans ce modèle les permissions sont associées à des rôles et non à des individus et un ou plusieurs rôles sont associé à un individu. De la même manière, un rôle environnemental est un rôle qui capture le contexte de l'utilisateur. Les rôles environnementaux sont basés sur les General Role Based Access Control.

Gaia : Roman et al (3) définit une architecture générique contextuelle pour les espaces physiques *(physical spaces)*, aussi appelée Gaia. Gaia est une région géographique qui est clairement limitée, contenant des objets physiques, des appareils dans un réseau hétérogène et des utilisateurs réalisant différents activités. *Active Space* permet la gestion de différents *Gaia.* Le système *Cerberus* complémente les *Active Spaces* en implémentant un framework d'authentification et un contrôle d'accès basés sur le contexte.

Notre approche s'inspire des rôles environnementaux de Covington mais afin d'exprimer nos besoins en termes de contrôle d'accès et de protection de la vie privée, nous avons décidé d'utiliser un langage existant développé à ces fins par Sun : XACML (eXtensible Access Control Markup Language) (25). Sa relative simplicité et ses capacités d'extension en font le meilleur candidat pour remplir nos besoins. XACML est un standard OASIS utilisé pour définir des politiques de contrôle d'accès. Il comprend un langage de description pour une politique de contrôle d'accès couplé à la description des requêtes de l'utilisateur. Une requête XACML est un triplet {Sujet, Action, Ressource}. Un Sujet demande l'accès à une Ressource (fichier, web service) afin d'effectuer une Action (lire/écrire, invoquer une méthode). Le Sujet peut être caractérisé par un ensemble d'attributs (rôle,

position). Basé sur ce triplet, une réponse XACML est alors retournée a l'utilisateur (permis, refuse, non applicable) en application des règles de la politique de contrôle d'accès. Des assertions SAML (26) nous permettent de construire des certificats génériques pouvant contenir différents attributs, dont des informations de contexte. Le choix de XACML pour le service d'autorisation est motivé par les raisons suivantes : son intégration avec les Web Services (27), son support des assertions SAML (36) et sa capacité à exploiter des informations contextuelles.

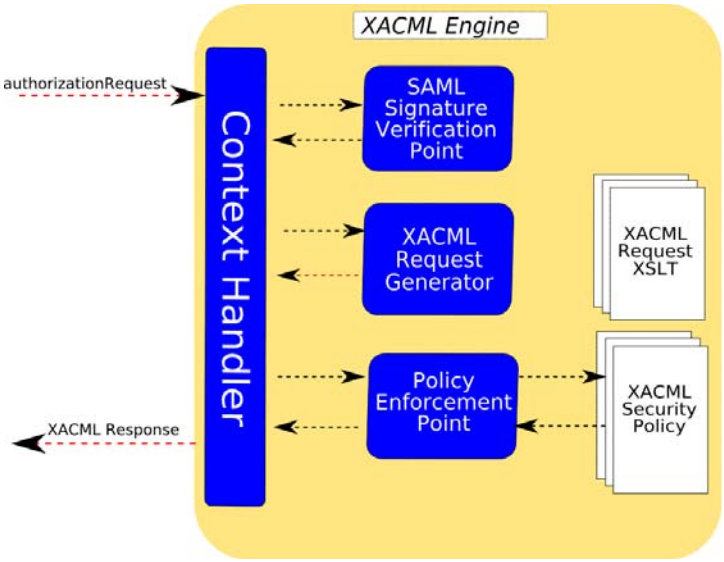

Figure 7: Point de Décision contextuel

Enfin, l'implémentation de XACML étant ouverte, l'exploitation et l'évaluation d'informations contextuelles représentées en XML est facilitée. Dans la Figure 8, nous montrons un exemple de condition définie en XACML utilisant une extension de type directement dérivé de notre ontologie. Cette condition « is-close-to» évalue la proximité entre la propriété d'une ressource et l'utilisateur demandant accès à la ressource, en fonction de leurs positions respectives comme déterminées grâce à des GPS. La position GPS de l'utilisateur est fournie dans une assertion SAML dans l'entête de la requête SOAP.

```
<Policy PolicyId="Policy" RuleCombiningAlgId="permit-
overrides">
  <Target>…</Target>
  <!-- Rule of getPatient Action -->
  <Rule RuleId="rgetPatient" Effect="Permit">
  <Target>
    <Subjects>…</Subjects>
    <Resources>…</Resources>
    <Actions>
      <Action>
      <ActionMatch MatchId="function:anyURI-equal">
      <AttributeValue
DataType="anyURI">getPatientMedical</AttributeValue>
      <ActionAttributeDesignator DataType=anyURI
AttributeId="action-id" />
      </ActionMatch>
      </Action>
    </Actions>
</Target>
  <!-- Check if the subject is physician -->
  <Condition FunctionId="function:string-equal">
    <Apply FunctionId="function:string-one-and-only">
      <SubjectAttributeDesignator DataType="string"
AttributeId="SubjectRole" />
    </Apply>
    <AttributeValue
DataType="string">Physician</AttributeValue>
    </Condition>
  </Rule>
</Policy>
```

Figure 8: Politique en XACML

La Figure 9 montre comment notre point d'accès compare la distance connue du docteur (émetteur de la requête et présenté comme assertion SAML) avec celle du patient qui est requise lors de l'évaluation de la règle. Si celle-ci est inférieure à une valeur donné, dans ce cas 2000 mètres, ils sont alors considérés comme proches.

```
<Apply FunctionId="CloseTo">
  <Apply FunctionId="findLocation">
    <SubjectAttributeDesignator
      DataType=GPSLocation AttributeId="SubjectLocation"/>
  </Apply>
  <AttributeValue
    DataType="integer">2000 meters</AttributeValue>
  <Apply FunctionId="string-one-and-only">
    <SubjectAttributeDesignator
      DataType=string AttributeId="PatientID" />
  </Apply>
</Apply>
```

Figure 9: Prise en compte de la distance

I.4- Processus collaboratifs ubiquitaires

I.4.1- Principes Généraux

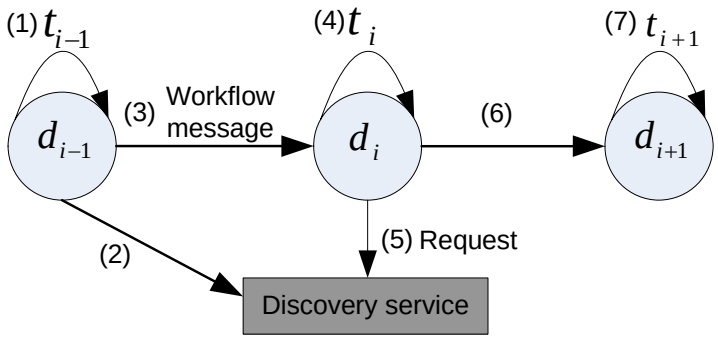

Figure 10: Principe de fonctionnement d'un processus collaboratif ubiquitaire

Les applications collaboratives basées sur l'informatique ubiquitaire permettent à des utilisateurs nomades d'accéder à des fonctionnalités offertes par des fournisseurs de services ambiants. La plupart de ces applications n'offrent cependant pas à l'heure actuelle la possibilité à leurs clients de collaborer ensemble et offrent seulement l'accès à des applications fournies par une infrastructure dédiée. Afin de rendre possible l'exécution d'applications collaboratives entre clients nomades, nous avons défini une architecture distribuée assurant la détermination et l'exécution d'un processus collaboratif ubiquitaire. Dans un tel environnement d'exécution, les utilisateurs nomades collaborent grâce à de complexes interactions qui peuvent être vues comme une extension du concept de processus collaboratif centralisé. L'architecture de processus collaboratif ubiquitaire introduit un support d'exécution permettant à des utilisateurs nomades travaillant à proximité les uns des autres de

partager leurs ressources et d'accéder à celles des autres selon un plan d'exécution prédéfini (*workflow*). Ce support d'exécution est caractérisé par les points suivant :

- *Une architecture distribuée* : la gestion des échanges ayant lieu au cours de l'exécution du processus collaboratif entre clients nomades est assurée par les clients eux-mêmes de telle sorte qu'aucune infrastructure dédiée ne soit nécessaire.
- *Un processus dynamique d'assignation des clients nomades aux différentes étapes d'exécution* : les ressources mises à disposition par les clients peuvent être découvertes au moyen d'un service spécialisé suivant les besoins fonctionnels nécessaires à la poursuite de l'exécution d'un processus collaboratif donné.

L'exécution typique d'un processus collaboratif ubiquitaire est décrite à la Figure 10. Ayant défini une représentation abstraite d'un processus collaboratif, c'est à dire l'ensemble des étapes fonctionnelles nécessaires à l'exécution d'un processus collaboratif, un utilisateur démarre le processus. Il exécute un premier ensemble de tâches [1] avant de rechercher dans son environnement immédiat, via le service de découverte, un collaborateur susceptible d'exécuter un sous ensemble des opérations suivant dans le plan d'exécution de son processus collaboratif [2]. Dès cette phase de découverte achevée, l'ensemble des données nécessaires au processus collaboratif est transmis par l'utilisateur à ce nouveau collaborateur [3] permettant la poursuite de l'exécution du processus [4]. L'enchaînement précédent : découverte d'un utilisateur nomade, transfert de données entre utilisateurs et exécution d'un ensemble de taches se répète itérativement jusqu'à ce que l'ensemble des opérations prévu par le plan du processus collaboratif soit achevé. Afin d'optimiser les ressources des utilisateurs nomades, le mode d'exécution est sans mémoire de telle sorte que les acteurs d'une instance d'un processus collaboratif ne gardent aucune information résiduelle nécessaire à la suite de son exécution et puisse se déconnecter dès qu'ils en ont terminé avec l'exécution des opérations qui leur ont été assignées. Toutes les données nécessaires à l'exécution d'une telle collaboration sont donc transférées entre chaque étape d'exécution d'un utilisateur vers le suivant. Les échanges de données entre acteurs nomades incluent le plan d'exécution du processus collaboratif pour permettre une exécution cohérente de celui-ci.

I.4.2- Implémentation basée sur le paradigme SOA

Les utilisateurs nomades impliqués dans une instance d'un processus collaboratif ubiquitaire sont en général membres d'une organisation (entreprise, hôpital, etc.) et doivent s'assurer qu'aucune information sensible n'est divulguée par leur participation à un processus collaboratif. En étant acteur de l'exécution d'un processus collaboratif, ils partagent d'une part leurs ressources applicatives et doivent d'autre part assurer la gestion locale de leur participation au processus. Chaque acteur implémente de ce fait deux modules logiciels : le premier regroupe l'ensemble des applications dont il dispose et le second est un moteur de processus collaboratif. Les applications logicielles implémentées par un utilisateur donné doivent rester la propriété de l'organisation à laquelle il appartient et donc demeurer privées. Elles ne peuvent être appelées que par cet utilisateur et le moteur de processus collaboratif local doit donc jouer le rôle d'une interface entre ces

applications et les autres utilisateurs nomades impliqués dans l'exécution d'un processus collaboratif.

Notre implémentation se base sur le langage Business Process Execution Language (BPEL) qui permet de spécifier des processus collaboratifs. Deux processus BPEL, un public et un autre privé, sont déployés sur le moteur de processus collaboratif afin de remplir ce rôle d'interface. Le processus public est contacté par les autres acteurs nomades tandis que le processus privé ne reçoit de requêtes que du processus public. Le processus public met donc à disposition les ressources d'un utilisateur et assure l'interface avec les acteurs potentiels alors que le processus privé implémente les logiciels fournissant ces ressources.

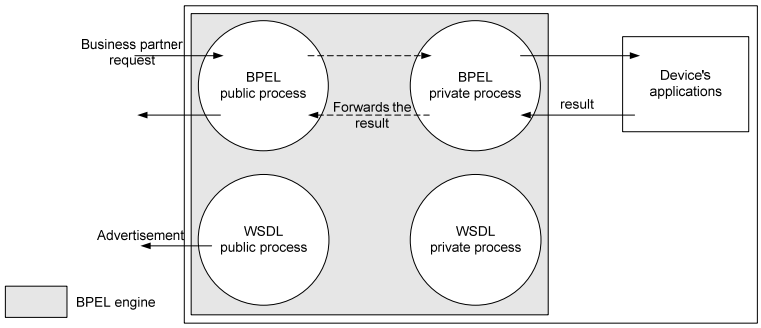

Figure 11: Implémentation basée sur le paradigme SOA

I.4.3- Sécurisation des processus collaboratifs

Par rapport aux processus collaboratifs centralisés classiques, l'exécution distribuée de processus collaboratifs introduit des contraintes en termes de sécurité informatique dues à l'absence d'un point de coordination prenant en charge la gestion des échanges entre les acteurs impliqués. Par conséquent, de simples primitives d'exécution vérifiées dans le cas centralisé comme l'adhérence de l'exécution d'un processus avec un plan prédéfini ne sont plus assurées. Nous avons classifié les prérequis que nous avons identifiés vis-à-vis de la sécurité informatique en trois catégories : autorisation, preuves d'exécution et protection des données du processus collaboratif.

I.4.3.1- Autorisation

La principale contrainte pour un processus collaboratif est d'assurer que seuls des acteurs autorisés sont assignés aux différentes étapes d'exécution. Dans le cas décentralisé et ubiquitaire, ces acteurs peuvent être sélectionnés au moment même de l'exécution par d'autres acteurs au moyen d'un service de découverte. La procédure de sélection de ces acteurs potentiels doit faire ici correspondre les contraintes de sécurité spécifiées par le plan d'exécution du processus collaboratif avec les droits que possèdent les acteurs disponibles.

I.4.3.2- Preuves d'exécution

Comme mentionné plus haut, l'adhérence de l'exécution d'un processus avec un plan prédéfini n'est pas assurée dans le cas de processus collaboratifs

décentralisés. Sans aucun coordinateur de confiance auquel se référer, les acteurs impliqués doivent avoir la possibilité de vérifier au cours de l'exécution d'un processus que cette dernière satisfait le plan qui a été au préalable défini afin qu'aucun acteur malveillant ne puisse fabriquer un faux processus collaboratif dans le but de nuire à des tiers.

I.4.3.3- Protection des données

Dans le cas décentralisé, toutes les données du processus collaboratif sont transférées entre les acteurs impliqués. Ceci nécessite des mécanismes spécifiques pour en assurer l'intégrité, la confidentialité et en contrôler l'accès dans la mesure où l'intégrité de l'exécution doit être assurée :

• Le contrôle d'accès sur les données du processus collaboratif est basé sur le plan d'exécution qui spécifie pour chaque étape le sous-ensemble de ces données qui sont accessibles en lecture

• Seul un sous-ensemble de ces données doit être modifié par l'exécution d'une étape donnée.

Assurer la protection des données du processus collaboratif semble être au coeur des mécanismes de sécurité nécessaires pour remplir les contraintes que nous avons identifiées. Nous associons à chaque étape de l'exécution d'un processus distribué deux paires de clefs : la première appelée « paire de politique de sécurité » est utilisée dans le processus de sélection des acteurs potentiels au cours d'un processus collaboratif, la seconde appelée « paire d'étape » protège l'accès aux données du processus collaboratif. Nous proposons un procédé de distribution de clefs au cours duquel les clefs privés d'étape (SKi)[1,n] sont mises à disposition des acteurs sélectionnés au moment même de l'exécution de l'étape qui leur a été assignée. Il est à noter que l'identité de ces acteurs n'est a priori connue qu'au moment même de l'exécution du processus : une distribution de clefs au préalable n'est donc pas une solution envisageable.

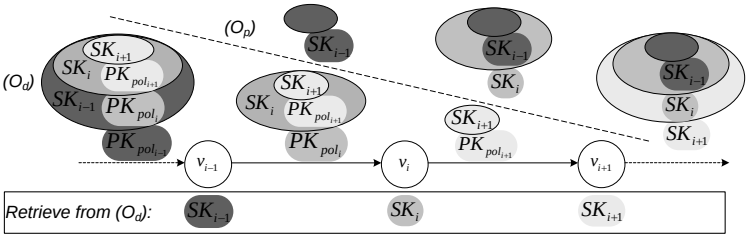

Figure 12: Protection des données avec chiffrement concentrique

La conception de ce mécanisme se base sur les techniques de chiffrement concentriques ou chiffrement en oignon utilisant les clefs publiques de politique PKpoli. En outre, des preuves d'exécution doivent être produites au cours de l'exécution d'un processus collaboratif pour assurer l'adhérence de ce dernier avec le plan préalablement défini. Dans ce but, nous utilisons aussi les techniques de chiffrement concentrique pour construire une structure regroupant les signatures des différents acteurs calculées avec les clefs privées d'étape. Ces deux mécanismes sont présentés à la Figure 12.

I.4.4- Service de découverte sécurisé

Dans l'environnement que nous avons décrit, les services sont dits « spontanés » puisque les possibilités de déploiement, d'accès, d'utilisation, de publication et de découverte sont flexibles et dynamiques. Afin de prendre en compte ces nouveaux paramètres, les techniques d'orchestration liées aux architectures orientées services s'avèrent indispensables pour garantir la flexibilité du système et faciliter le déploiement d'applications ubiquitaires. Les techniques de découverte de services sont primordiales, car elles concernent la première étape du processus d'interconnexion entre les utilisateurs et les services. Elle joue un rôle particulièrement important dans le cadre de l'exécution de nos processus collaboratifs ubiquitaires. Ainsi, n'importe quel utilisateur peut déployer facilement et rapidement un service donné qui sera accessible par tous les utilisateurs proches. Plusieurs de ces techniques existent déjà, on pourra ainsi citer entre autre le protocole Universal Plug and Play (UPnP) (40) utilisé pour interconnecter les petit appareils domestiques dans un petit réseau privé ou le Service Location Protocol (SLP) (37) un peu plus élaboré que le précédent mais se limitant aux réseaux locaux.

Avec l'émergence et la diffusion des technologies Web Services, la découverte de service se doit de franchir un nouveau cap, à savoir l'utilisation de descriptions sémantiques assez complexes, elle a ainsi recourt à l'utilisation des ontologies pour la description des services. Le volume de méta-données échangé, ainsi que le nombre de ressources mises en jeu deviennent de plus en plus importants : ces ressources sont plus délicates à gérer et exposent forcément les utilisateurs à de nouveaux dangers, parmi lesquelles l'exploitation illégale des données privées des utilisateurs (contenues dans leurs profils), ou bien les attaques en déni de service visant les serveurs. Certains standards de découverte, tel WS-Discovery(43), soulèvent ce problème dans leur spécification sans pour autant apporter une véritable solution si ce n'est la signature des messages qui semble assez inadaptée aux problèmes soulevés. Quelques solutions existent dans la littérature comme (10) ou (34), mais elles nécessitent généralement le déploiement d'une infrastructure assez coûteuse et complexe à gérer. De plus, la protection des données privées y est complètment négligée et l'aspect sécurité se limite généralement à la protection du serveur.

Dans le cadre du projet MOSQUITO, nous avons donc étendu le protocole de découverte WS-Discovery (43) pour répondre à ces différents besoins. La solution développée consiste à offrir aux serveurs et aux clients la possibilité d'associer une politique de sécurité propre à la découverte. Ainsi, chaque utilisateur pourra spécifier clairement quels sont les serveurs auxquels il souhaiterait se connecter (sélection par domaine, propriétaire, certification, signature …), et quelles sont les informations personnelles qu'il sera nécessaire de protéger. De même les serveurs peuvent eux aussi restreindre la possibilité d'être découvert à une classe de clients considérés comme de confiance (appartenant par exemple à un domaine donné, ou recommandés par une entité de confiance). Dans notre solution, nous avons choisi de conserver une architecture classique composée de trois principaux acteurs ; les serveurs, les clients et un système d'annuaire (*registry*). Cet annuaire ne sert pas seulement à l'enregistrement des coordonnées des services et de leur délivrance aux

clients en faisant la demande comme dans le protocole UDDI, mais il tient aussi lieu de tierce partie de confiance responsable des l'authentification et de l'application des politiques de découverte des utilisateurs (clients et serveurs). Les communications entre le serveur et l'annuaire et entre le client et l'annuaire sont protégées quant à elles par des mécanismes PKI traditionnel.

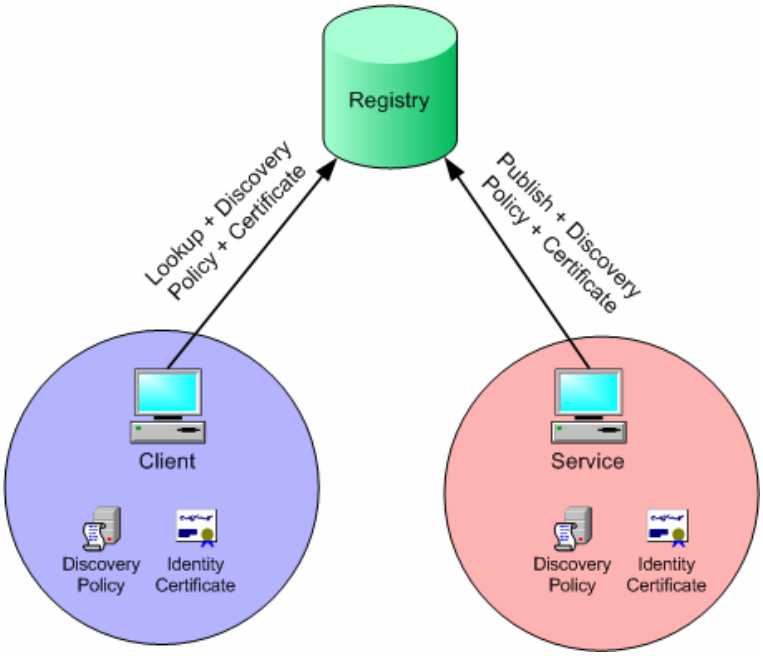

Figure 13: Architecture du service de découverte

Comme on peut le constater sur la Figure 13, lors de son enregistrement, le serveur envoie d'une part les informations de description des services à publier et d'autre part ses politiques de découverte (exprimées en XACML) ainsi qu'un certificat lui permettant de s'authentifier auprès de l'annuaire en cas de besoin. De son côté, le client à la recherche d'un service donné va envoyer dans sa requête sa propre politique de découverte, ainsi qu'un certificat pour s'authentifier auprès de l'annuaire. Celui-ci s'occupera de choisir le bon service en accord avec les politiques de découverte respectives.

Un exemple concret d'utilisation du service de découverte sécurisé peut être extrapolé de nos scénarii : le médecin recherche des services environnants appartenant à l'établissement hospitalier tout en limitant sa recherche aux services certifiés par le ministère de la santé. Parallèlement, les serveurs concernés peuvent spécifier que seuls les clients nantis du rôle de docteur hospitalier sont capables de découvrir les services proposés.

Ce service de découverte propose ainsi une nouvelle solution simple et efficace pour palier les problèmes de protection des donnés privées et de contrôle d'accès liés à la découverte dans un environnement ubiquitaire. De plus l'utilisation de langages

politiques de sécurité, aussi flexibles que XACML, permet aussi le raisonnement sur d'éventuels éléments contextuels liés à l'environnement des utilisateurs.

Conclusion et Travaux Futurs

Comme le lecteur a pu le découvrir tout au long de ce chapitre, construire une application ubiquitaire contextuelle n'est pas une tâche triviale : de nombreux problèmes demeurent et leur résolution nécessite une approche pluridisciplinaire. Les nouvelles applications médicales, de par leurs contraintes en termes de protection de la vie privée, de normes, d'interopérabilité et d'implications sont particulièrement exigeantes. Notre approche apporte des éléments novateurs dans plusieurs domaines :

1. l'utilisation et le traitement de l'information contextuelle afin de pouvoir garantir une évaluation de confiance de bout en bout et ainsi permettre son intégration sûre dans l'application, notamment pour la prise de décision et l'adaptation de la sécurité,
2. la mise en place de processus collaboratifs ubiquitaires offrant des garanties transactionnelles et de protection des données basées sur la cryptographie,
3. la prise en compte des desiderata de toutes les entités dès la phase amont, c'est à dire la découverte, afin de diminuer les possibilités de violation de vie privée et de limiter l'accès aux seuls services nécessaires.

Nous tenons à souligner que le travail présenté ici lie une approche théorique avec un prototypage des principaux composants logiciels décrits. Il ne constitue pas pour autant la solution ultime à tous les problèmes évoqués et laisse entrevoir de possibles extensions.

Dans le domaine des informations contextuelles, le développement et la mise en place d'une métrique appropriée afin d'appréhender la qualité de l'information (fiabilité et confiance) provenant d'un ensemble hétérogène de capteurs est une tâche complexe qui demeure à ce jour inaccomplie. Cette difficulté augmente encore dans le cas où des informations sont agrégées et fusionnées par des mécanismes de raisonnement complexes.

De même, les travaux sur le Traitement Complexe d'Événements[2] (TCE) proposent une approche alternative au contrôle de processus standard (*workflow*), complémentaire des processus ubiquitaires. Le TCE définit un ensemble d'outils et de techniques pour l'analyse et le contrôle des évènements dans le cadre des nouveaux systèmes de planification d'entreprise (5) Cette approche peut potentiellement s'adapter à différents aspects des applications médicales ubiquitaires, tel que l'automatisation, l'ordonnancement, la surveillance et le contrôle de la collaboration, mais aussi à la prédiction des performances et la détection d'attaques. Le TCE peut ainsi augmenter la flexibilité des solutions logicielles en permettant à chaque utilisateur (humains ou logiciel) de spécifier dynamiquement son intérêt pour des évènements à différents niveaux d'abstraction

[2] Complex Event Processing

(de la lecture d'un capteur à des conclusions nécessitant des compétences médicales).

Ces différentes voies de recherche seront explorées dans le cadre d'une expérimentation *in-vivo* dans une maison intelligente avec des intervenants du monde de la santé grâce à la mise en place récente d'une collaboration avec le laboratoire DOMUS de l'Université de Sherbrooke.

Référence

(1) A. Jøsang, C. Keser, and T. Dimitrakos. *Can We Manage Trust?*. In the Proceedings of the Third International Conference on Trust Management (iTrust) 2005

(2) A. K. Dey, "Understanding and using context," Personal and Ubiquitous Computing Journal, vol. 5(1), pp. 4–7, 2001.

(3) Campbell, Roman and K. Nahrstedt, "Gaia: A middleware infrastructure to enable active-spaces," 2002.

(4) Covington and al, "A context-aware security architecture for emerging applications," in Proceedings of the Annual Computer Security Applications Conference (ACSAC), Las Vegas Nevada USA, 2002.

(5) D. Luckham, The power of events : An introduction to complexe vent processing in distributed enterprise systems, Addison-Wesley, 2002.

(6) E. I. Covington M. Ahamd M. and V. H., "Parameterized authentication," 2004.
e20, June 2001. [Online:] http://www.jmir.org/2001/2/e20/

(7) European Commission, "Green Paper on Demographic changes." [Online:]http://ec.europa.eu/employment_social/news/2005/mar/demog_gp_en.html

(8) F. Baader, I. Horrocks, and U. Sattler. Description logics. In S. Staab and R. Studer, editors, Handbook on Ontologies, International Handbooks on Information Systems, pages 3–28.Springer, 2004.

(9) F. Stajano, "Security for Ubiquitous Computing", John Wiley and Sons, Feb. 2002.

(10) F. Zhu, M. Mutka, L. Ni "Prudent exposure: A private and user centric service discovery protocol" Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom'04) Orlando, USA, 2004

(11) FIPA. Fipa device ontology specification. Technical report, Foundation for Intelligent Physical Agents, 2001.

(12) G. Eysenbach, "What is e-health?" J Med Internet Res, vol. 3, no. 2, p.

(13) G. Shafer, "A mathematical theory of evidence," 1976.

(14) H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard ontology for ubiquitous and pervasive applications. In Proceedings of the International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004.

(15) J. I. Hong and J. A. Landay, "An architecture for privacy-sensitive ubiquitous computing," in MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services. ACM Press, 2004, pp. 177–189.

(16) J. Van den Bergh and K. Coninx. Towards integrated design of context-sensitive interactive systems, Mar. 2005.
(17) Jess. Rule engine for java. http://herzberg.ca.sandia.gov/jess/.
(18) K. Wrona and L. Gomez „ Context-aware security and secure context-awareness in ubiquitous computing environments, 13th Wireless World Research Forum, 2005
(19) Kay Römer, Oliver Kasten, Friedemann Mattern, "Middleware Challenges for Wireless Sensor Networks", ACM Mobile Computing and Communication Review, Vol. 6, No. 4, pp. 59-61, October 2002
(20) L. Bussard L., Roudier Y., "Untraceable secret credentials: Trust establishment with privacy," in PERCOMMW'04. Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.
(21) L. Gomez and K, Wrona, Context-aware security and secure context-awareness in ubiquitous computing environments, Annales Universitatis Mariae Curie-Sklodowska, Sectio AI Informatica, (in press).
(22) M. L. Wullems Chris and A. Clark, "Toward context-aware security: an authorization architecture for intranet environments," 2004.
(23) MOSQUITO Consortium, "IST 004636 MOSQUITO Project." [Online:]http://www.mosquito-online.org
(24) N. Shankar and D. Bafanz, "Enabling secure ad-hoc communication using context-aware security services," in UNBICOMP 02: Workshop on Security in Ubiquitous Computing, 2002.
(25) OASIS (IBM, Sun, BEA, Entrust et al). eXtensible Access Control Markup Language (XACML). Version 1.1. Committee Specification, 7 August 2003.
(26) OASIS (Netegrity, Sun, RSA, BEA, IBM, VeriSign). Security Assertion Markup Language (SAML). Version 1.1. OASIS Standard. 2 September 2003.
(27) OASIS, XACML profile for web-services, Working Draft 04, 29 Sept 03, http://www.oasis-open.org/committees/download.php/3661/draft-xacml-wspl-04.pdf.
(28) Organization for the Advancement of Structured Information Standards (OASIS), http://www.oasis-open.org/.
(29) P. Robinson, "Trust context spaces an infrastructure for pervasive security in context-aware environments," 2003.
(30) R. C. Jalal Al-Muhtadi Anand Ranganathan and M. D. Mickunas, "Cerberus: A context-aware security scheme for smart spaces." 2003.
(31) R. Lewis and R. Merrick. Content selection for device independence (diselect) 1.0. Working draft of w3c recommendation,WirelessWorld ResearchConsortium, Jun. 11 2004.
(32) S. Ganeriwal and M. B. Srivastava, "Reputationbased framework for high integrity sensor networks," in SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks. ACM Press, 2004, pp. 66–77.
(33) S. Trabelsi, J.C. Pazzaglia and Y. Roudier "Enabling Secure Discovery in a Pervasive Environment" 3rd International Conference on Security in Pervasive Computing (SPC 2006) – York – UK – April 2006
(34) S.E. Czerwinski et al, "An Architecture for a Secure Service Discovery Service" , In Proceedings of MobiCom '99, Seattle, WA, August 1999

(35) Salem Hadim and Nader Mohamed, "Middleware Challenges and Approaches for Wireless Sensor Networks," IEEE Distributed Systems Online, vol. 7, no. 3, 2006, art. no. 0603-o3001.
(36) SAML 2.0 profile of XACML v2.0, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf.
(37) Service Location Protocol, Version 2 RFC 2608
(38) T. Broens and al, “Context-aware, ontology-based, service discovery”, 2nd Symposium on Ambient Intelligence, 2004X. H.Wang, D. Q. Zhang, T. Gu, and H. K. Pung.Ontology based context modeling and reasoning using OWL. In PerCom Workshops, pages 18–22, 2004.
(39) Trabelsi, Slim;Pazzaglia, Jean-Christophe; Roudier, Yves; Secure Web service discovery: overcoming challenges of ubiquitous computing at ECOWS 2006, Zurich, December 2006
(40) Universal Plug and Play http://www.upnp.org/download/UPnPDA10_20000613.htm
(41) W3C. OWL - Web Ontology Language. http://www.w3.org/2004/OWL/.
(42) W3C. Ressource description framework. http://www.w3.org/TR/rdf-syntax-grammar/.
(43) WS-Discovery Specifications http://msdn.microsoft.com/ws/2005/04/ws-discovery/
(44) X. H.Wang, D. Q. Zhang, T. Gu, and H. K. Pung. Ontology based context modeling and reasoning using OWL. In PerCom Workshops, pages 18–22, 2004.
(45) Y. Lee, S. A. Chun, and J. Geller. Web-based semantic pervasive computing services. IEEE Computational Intelligence Bulletin, 4(2):4–15, 2004.