

UNIVERSITÉ DE MONTRÉAL

AN OVERLAY GATEWAY FOR THE INTEGRATION OF IP MULTIMEDIA SUBSYSTEM  
AND MOBILE SINK BASED - WIRELESS SENSOR NETWORKS

MARCELA VÉLEZ-PULGARÍN

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE INFORMATIQUE)

DÉCEMBRE 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

AN OVERLAY GATEWAY FOR THE INTEGRATION OF IP MULTIMEDIA SUBSYSTEM  
AND MOBILE SINK BASED - WIRELESS SENSOR NETWORKS

Présenté par : VÉLEZ-PULGARÍN, Marcela

en vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées

a été dûment accepté par le jury d'examen constitué de :

Mme. BOUCHENEB Hanifa, Doctorat, présidente

M. QUINTERO Alejandro, Doct., membre et directeur de recherche

M. GLITHO Roch, Ph.D., membre et codirecteur de recherche

M. PIERRE Samuel, Ph.D. membre

## **DEDICATION**

*To my husband and colleague, who gives me the love and courage I need every once in a while.*

*To my parents, my brother and my sister for their infinite love and support.*

*And to Sofia, my little angel.*

## **ACKNOWLEDGEMENTS**

I would like to thank my director of research, Mr. Alejandro Quintero and my co-director Mr. Roch Glitho who provided me the means, the supervision and the advices that made this work possible.

## RÉSUMÉ

D'une part les Réseaux de Capteurs (WSN par ses sigles en *anglais*) constituent un domaine de recherche qui a reçu beaucoup d'attention de la part de la communauté scientifique grâce à ses avantages en différents domaines. Chaque réseau est généralement conçu à partir de périphériques de petite taille appelés capteurs qui peuvent capter, effectuer des calculs et communiquer entre eux. De plus, inclure des stations de base mobiles dans les réseaux de capteurs sans fils s'est avéré utile dans une large gamme de scénarios puisque ces derniers permettent d'améliorer la durée de vie globale du réseau et d'augmenter la capacité de transmission de données. D'autre part, le sous-système IP Multimédia (IMS) vise à fournir un accès aux réseaux cellulaires à l'ensemble des services Internet. Il s'agit d'une surcouche de contrôle implantée sur une couche IP dont les objectifs sont de fournir et maintenir une qualité de service (QoS) donnée, un schéma de chargement équitable et des services intégrés; le tout en utilisant des interfaces standards.

Combiner les fonctionnalités d'IMS et l'ensemble d'informations contextuelles capturées par les stations de base des réseaux de capteurs sans fils ouvre la porte à une nouvelle gamme de services multimédias. Cette dissertation propose une architecture de surcouche pour l'intégration des IMS avec les réseaux de capteurs sans fils. Le service "présence" sert de point d'entrée au domaine des IMS. La passerelle, qui constitue le cœur de notre architecture, est une couche de recouvrement construite sur les mêmes téléphones mobiles qui agissent également en tant que canaux mobiles, publicateurs / écouteurs de présence et périphériques utilisateurs.

## **ABSTRACT**

On one hand, Wireless Sensor Networks (WSNs) are a research area that has been gaining attention from the research community. They are made up of small scale devices called sensors that can sense, compute and communicate. Moreover, including mobile sinks in WSNs has shown to be useful in a wide range of scenarios since they can improve the overall network lifetime and increase data capacity. On the other hand, the IP Multimedia Subsystem (IMS) aims to provide cellular access to all Internet services. It is an overlay control layer on top of an IP layer whose goals is based in provisioning of Quality of Service (QoS), a fair charging scheme and integrated services through standard interfaces. Combining the capabilities of IMS with the rich set of contextual information captured by mobile sink WSNs opens the door to a wide range of novel multimedia services. This dissertation proposes an overlay architecture for the integration of IMS with mobile sink-based WSN. The Presence service is used as entry point to the IMS world. The gateway which is the heart of our architecture is an overlay built on top of the very same mobile phones that act as mobile sinks, presence publishers / watchers, and end-user devices.

## CONDENSÉ EN FRANÇAIS

### 1. Introduction

Depuis quelque temps, la tendance est davantage aux systèmes et technologies intégrées qu'aux solutions isolées. Les clients désirent avoir accès à plusieurs services sans en connaître la source ni le système d'origine. Par conséquent, l'intégration de systèmes et la standardisation ont acquis une place privilégiée au sien de l'informatique omniprésente.

Ce projet porte sur l'intégration du sous-système multimédia IP (ou IMS, selon l'abréviation *anglaise*) et les réseaux de capteurs (ou WSN, selon l'abréviation *anglaise*). Les deux sont des technologies qui évoluent rapidement. Ces dernières ont attiré l'attention de la communauté de recherche en raison de leurs avantages et des différents services qu'ils peuvent offrir.

Les réseaux de capteurs sont des réseaux sans-fils capable de rassembler et de transmettre des données. Chaque réseau se compose, en général, d'un grand nombre de nœuds appelés *capteurs*. Ces derniers rassemblent, traitent et transmettent des données à une station de base (ou à un *puits*) qui est souvent centralisé. Les WSNs se divisent en deux catégories: surveillance et dépistage. La première catégorie inclut notamment les systèmes de santé, les systèmes propres à l'environnement et le control de trafic. La deuxième catégorie sert principalement à surveiller des personnes, des animaux ou des véhicules.

Les réseaux de capteurs mobiles (WSN mobile) constituent un sous-domaine des WSN. Dans le cadre de ce projet, nous considérons les WSNs dont le *puits* est mobile, c'est-à-dire que le *puits* peut se déplacer via un chemin déterminé ou bien de façon autonome.

Au début, la mobilité était perçue comme un problème, mais des travaux de recherches ont démontré qu'il peut avoir des avantages dans ses types des réseaux [7-9]. WSN mobile peut augmenter la vie du réseau à long terme, l'espace couvert ainsi que la capacité des données.

Le sous-système multimédia IP (IMS) est une architecture clé dans le réseau 3G. Ce système a été proposé comme une architecture standard qui fournit un accès permanent aux services Internet à travers des réseaux cellulaires. Ses avantages sont, tel que mentionné précédemment, l'intégration des services, la qualité de service et un schéma de facturation différentielle. L'intégration permet aux utilisateurs d'accéder aux différents services en utilisant

les mêmes points d'entrée. En plus, celle-ci donne l'option aux fournisseurs de créer et d'offrir de nouveaux services en intégrant ceux déjà existants. IMS définit également le déroulement de la session en considérant la qualité de service. Enfin, IMS propose un système de facturation différentiel qui peut dépendre du service utilisé et pas seulement de la quantité d'octets transmis.

Le but de ce mémoire est de combiner l'information ramassée par les réseaux de capteurs avec les capacités de dissémination et d'intégration offertes par IMS. Les informations sont préalablement récoltées par le sous-système IMS. Par la suite, des applications et des utilisateurs peuvent les analyser dans le but d'offrir une rétroaction effective et efficace. À cette fin, on propose une architecture décentralisée basée sur les systèmes pair-à-pair.

### **Objectif**

Le but de ce projet est de proposer et construire une passerelle sur des téléphones mobiles qui intègrent IMS et les réseaux de capteurs à *puits* mobiles. Les objectifs spécifiques sont de :

- Proposer une architecture pour l'intégration.
- Concevoir l'architecture du logiciel pour la passerelle qui permettra l'intégration.
- Développer un prototype de la passerelle.
- Valider l'architecture.

### **Motivation et exigences**

On base notre étude sur un cas particulier dans le domaine de l'agriculture. Dans ce domaine, le WSN a été étudié et proposé parce qu'il permet aux utilisateurs d'avoir accès à l'information du cultivateur en temps-réel.

Dans le domaine de l'agriculture, des capteurs peuvent être installés à différents endroits de la ferme afin de mesurer l'humidité du sol, la température et l'exposition aux ravageurs. L'information récoltée par les capteurs est ramassée par plusieurs *puits* mobiles que les travailleurs de la ferme ont en leur possession. La collecte d'informations a lieu pendant que les travailleurs se déplacent, au moyen de dispositifs mobiles tels qu'un PDA. Si aucun *puits* n'est situé à proximité et le nœud détermine que l'information doit être disséminée, il envoie les données à des stations fixes de recollection qui peuvent communiquer entre elles à des distances plus grandes [5].



Lorsque le *puits* a récolté l'information, il la renvoie vers le sous-système IMS où les données sont analysées par un serveur d'application. Par exemple, si l'humidité du sol est inférieure à une valeur dans une partie de la culture, des alertes pourraient être envoyées au(x) travailleur(s) au moyen de services multimédia d'IMS (par exemple SMS, MMS, IM). Dans le futur, d'autres services pourraient également être considérés car les données sont déjà disponibles sur IMS et le but de l'architecture est d'unifier et intégrer les différents services. Voici quelques exemples d'actions proactives à tenir en compte : activation automatique du système d'irrigation, activation de pesticides et chauffage (au besoin).

En se basant sur cette application, on a défini quelques exigences pour notre système:

- Indépendance : La solution doit minimiser les changements requis à IMS et WSN.
- Les types d'informations WSN suivants doivent être disponibles sous IMS : les réseaux de capteurs peuvent récolter des données spatiales (exemple : localisation), physiologiques (exemple : fréquence cardiaque) et environnementales (exemple : température) qui ne sont pas considérés à ce point sous IMS.
- Traduction : la passerelle devra traduire l'information de WSN au format utilisé par IMS.
- Publication: les données doivent être publiées sous IMS.
- Stockage et traitement : ces deux processus doivent être présents afin de permettre un accès synchrone et asynchrone à l'information.
- IMS « traditionnel » et pair-à-pair : la solution proposée doit être capable de se connecter aux deux types de sous-systèmes IMS.
- Extensibilité et tolérance aux pannes.
- Communication allégée car la passerelle doit se connecter aux dispositifs mobiles de capacité limitée.

## 2. Revue de la littérature

Notre revue de la littérature a été réalisée selon deux perspectives : les intégrations proposées entre les réseaux de capteurs et Internet et les intégrations proposées entre les réseaux de capteurs et IMS.

### **Intégration entre WSN et Internet**

Une solution souvent utilisée est l'intégration en utilisant des services web. Un service web est un logiciel conçu pour supporter la communication et l'échange de données entre applications et systèmes hétérogènes.

Wang et al. [20] ont proposé l'utilisation de services web afin d'intégrer les réseaux de capteurs et Internet. Le système proposé possède une passerelle qui assigne une adresse IP unique à chaque nœud du WSN, donc donnant un accès direct. Leur architecture est basée sur ZigBee, un standard du WSN. Le coordonateur est connecté à la passerelle qui envoie, traite et récolte les données et les transmet vers le client à travers le réseau IP. Une autre proposition a été faite en [21], où six services web sont mis en place au sein d'une passerelle qui les expose aux clients Internet. L'architecture est basée sur le paradigme I-centric; c.-à-d. que les besoins, les préférences et les paramètres environnementaux du client sont utilisés au sein du système de communication.

En regardant ces solutions à la lumière de nos exigences, on voit qu'elles ne sont pas complètement satisfaites. La solution proposée par [21] offre aussi des services de traduction et de publication. Même si les deux solutions décrites fournissent un stockage et un traitement des données, IMS n'est pas considéré. Wang et *al.* font une intégration simple, où les nœuds sont des entités et non un seul système. La performance est aussi un problème car le traitement du protocole SOAP pour les services web est assez lourd. De plus, les passerelles ne sont ni extensibles ni tolérantes aux pannes.

### **Intégration entre WSN et l'architecture IMS**

Gluhak and Schott en [5] présentent l'architecture e-SENSE pour distribuer l'information des capteurs aux utilisateurs via IMS. La passerelle e-SENSE s'enregistre auprès du sous-système IMS. Elle place les données en mémoire tampon, les ordonne, établit les priorités d'envoi et permet la publication des services. Cependant, la passerelle n'offre pas de service de

stockage ni de traitement de données. Les auteurs ne considèrent pas non plus l'extensibilité, la tolérance aux pannes et les mécanismes d'allègement de la communication.

Arbanowski et al ont proposé en [4] une passerelle qui intègre les WSNs et les réseaux personnels (PAN sigles en anglais) en IMS. La solution est indépendante, partiellement extensible et prend en considération l'allègement de la communication et la publication de données. Malheureusement, elle n'offre ni stockage, ni traitement de données ni traduction. Finalement, elle ne supporte pas toutes les entités et types d'informations des WSN.

El Barachi et al expose en [3] une autre architecture pour l'intégration entre IMS et les réseaux de capteurs. Une passerelle générique est définie. Cette dernière sert à fournir des fonctions de traitement de l'information pour l'acquérir, la traiter, la stocker et la disséminer. Ces fonctions s'apparentent à des fonctions de contrôle afin de permettre l'interconnexion. Comme point d'entrée, El Barachi et al proposent le service de présence. Cette proposition satisfait presque toutes les exigences. Cependant, ni l'extensibilité ni la tolérance aux pannes ne sont considérées. De plus, il ne serait pas facile d'y intégrer IMS pair-à-pair puisqu'il n'y aurait qu'un seule point d'entrée.

Nous les utilisons dans notre projet comme point d'entrée à IMS puisqu'ils manipulent déjà des données contextuelles. En plus, ils proposent une extension au modèle standard PIDF afin d'intégrer les types des données inclus dans les réseaux de capteurs et afin d'être capable de différencier l'information des utilisateurs des objets.

### **3. Architecture**

Le but principal de cette architecture est d'accomplir et de satisfaire aux exigences présentées. À cette fin nous proposons une passerelle construite sur un réseau de recouvrement qui est lui-même bâti sur des dispositifs cellulaires qui participent au sein du réseau de capteurs et dans IMS. D'abord, nous présenterons le design et les définitions de base. Ensuite, nous verrons les règles du réseau, en incluant le modèle d'information, le protocole utilisé et les processus.

#### **Principe de l'architecture**

- **Suppositions**

Nous supposons que les réseaux sont soient gérées par le même fournisseur, ou alors il existe un accord entre fournisseurs qui permet la communication entre les appareils.

De plus, WSN, IMS et la passerelle sont configurés pour être capable d'établir une communication entre eux.

On suppose aussi que les appareils cellulaires sont capables de communiquer entre eux afin de créer un réseau de recouvrement en surcouche.

Finalement, on suppose que le WSN envoie un format de message spécifique qui devra être traduit au PIDF-étendu.

- **Principes**

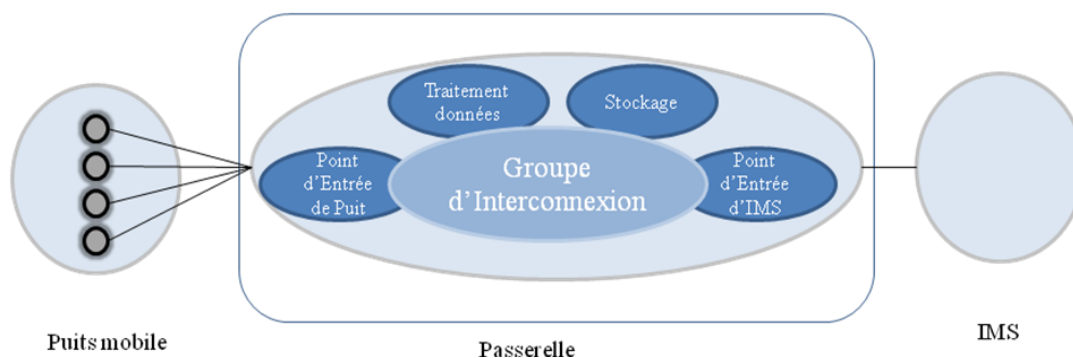


Figure 1. Architecture générale

Les principes de notre architecture sont basés sur une passerelle qui est le point central de l'intégration et qui servira de réseau de recouvrement. La Figure 1 montre l'architecture générale, divisée en cinq groupes. Chaque groupe s'occupe de fournir un ou plusieurs services à un nœud *leader*.

Le premier groupe (i.e. Point d'Entrée d'IMS) se connecte à IMS. Le deuxième groupe (i.e. Point d'Entrée de Puits) se connecte aux *puits* mobiles qui appartiennent au réseau de capteurs. Le troisième et le quatrième fournissent des services de stockage et de traitement des données, respectivement. Finalement, le dernier groupe sert à interconnecter les groupes précédents. Ce dernier est composé des nœuds leaders de chaque autre groupe et il est situé au milieu de la Figure 1.

- **Topologie, rôle des nœuds et interactions**

La Figure 2 représente le réseau de recouvrement qui sert de passerelle entre WSN et IMS. La topologie est à deux-étages : le premier est dédié au stockage et le

deuxième est dédié aux autres groupes définis (Point d'entrée vers IMS et WSN, traitement des données et le groupe d'interconnexion).

Chaque groupe a un ensemble de nœuds qui jouent certain rôles; ceci détermine les services qu'il peut fournir ainsi que la logique qu'il doit exécuter.

Le premier groupe (i.e. Point d'Entrée d'IMS) a deux rôles: *Point d'Entrée du Service de la Présence (PSE)* et *Leader de Point d'Entrée du Service de la Présence (SPSE)*. *PSE* offre le service de point d'entrée, en fournissant une connectivité aux serveurs CSCFs d'IMS. Ce dernier ne communique qu'avec le *SPSE* à l'intérieur de la passerelle. Le *SPSE* est le leader du groupe, mais il peut aussi jouer le rôle de *PSE*. Il gère les nœuds du groupe, traduit l'information au format PIDF-étendu, envoie et reçoit les requêtes entre IMS et les autres groupes de la passerelle.

Le deuxième groupe (i.e Point d'Entrée de Puits) a aussi deux rôles: *Point d'Entrée de Puits (SEP)* et le *Leader de Point d'Entrée de Puits (SSEP)*. *SEP* permet aux *puits* du WSN de se connecter à la passerelle. Le *SSEP* est le leader de ce groupe. Il gère les nœuds du groupe, interprète et analyse les données reçues du WSN et est capable de déterminer si l'information doit être stockée, traitée ou publiée à IMS. Il offre aussi le service de traduction au modèle d'information du réseau de capteurs.

Le troisième et le quatrième groupe sont deux groupes différents. Toutefois, le leader est partagé. Le groupe Stockage est composé des rôles *Stockage* et *Leader de Gestion des Données(SDM)*. Les nœuds qui jouent le rôle *Stockage* sont au premier étage et ils ont la responsabilité de stocker et récupérer l'information sauvegardée. Le leader (*SDM*) est au deuxième étage et est en charge des nœuds du groupe. L'autre groupe (*Traitement de données*) regroupe le rôle *Traitement*, qui traite l'information via le leader qui est, tel que mentionné précédemment, le *SDM*.

Finalement, les trois leaders (*SPSE*, *SSEP* and *SDM*) forment le dernier groupe. Ils collaborent afin d'offrir les services au réseau de recouvrement et permettre l'accès aux groupes de la passerelle.

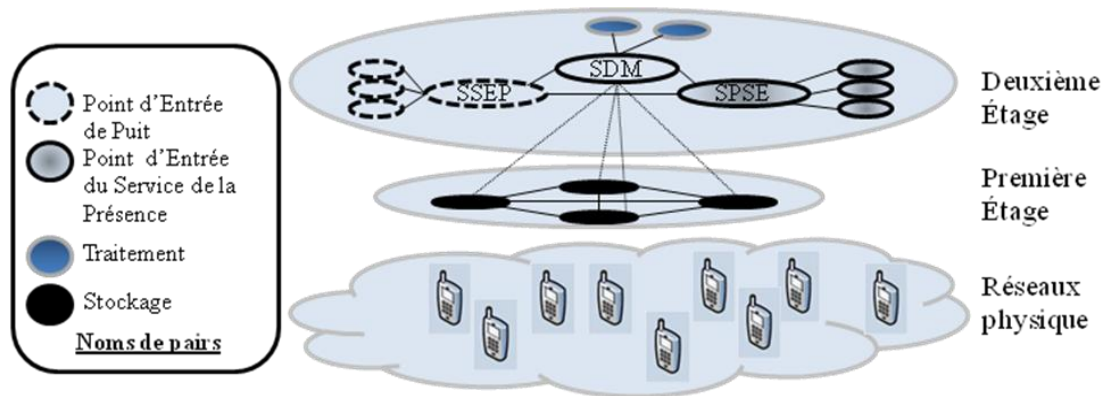


Figure 2. Passerelle

Maintenant que les rôles ont été définis, les interactions à l'intérieur des groupes seront explicitées. Dans le premier groupe (i.e. Point d'Entrée d'IMS), si un message est reçu par le *PSE* d'IMS, il est renvoyé vers le *SPSE*. Également, si un message est reçu d'un autre nœud du groupe de leaders, le *SPSE* le renverra vers l'un de *PSEs* pour qu'il le renvoie à IMS.

Au deuxième groupe (i.e. Point d'Entrée de Puits) le *SSEP* échange des données entre WSN et la passerelle via des *SEPs*. Chaque *SEP* se connecte à un ou plusieurs puits. Quand les données arrivent aux *SEPs*, elles sont renvoyées vers le *SSEP* qui décide si elles doivent être stockées, traitées ou publiées. Si le *SSEP* reçoit une requête d'un autre leader pour récupérer de l'information du WSN, il renvoie cette requête à tous les *SEPs*.

Le *SDM* reçoit toutes les requêtes concernant le stockage et le traitement des données des autres nœuds leaders et les renvoie vers le groupe qui offre le service. Si l'information doit être stockée, la requête est envoyée vers tous les nœuds du groupe *Stockage*. Sinon, si elle doit être traitée, le *SDM* envoie la requête vers un des nœuds du groupe *Traitement*.

Présentons les interactions entre le groupe des leaders (i.e. Groupe d'Interconnexion). Si le *SSEP* détermine que les données doivent être stockées ou traitées, il envoie la requête vers le *SDM*. Dans le cas contraire, si elles doivent être publiées, la requête est envoyée vers le *SPSE*.

Le *SPSE* publie l'information vers IMS. Il reçoit la requête de publier et l'envoi au groupe (i.e. groupe de Point d'Entrée d'IMS). Si une requête arrive d'IMS, le *SPSE* la renvoie vers le *SDM*.

Le *SDM* répond aux requêtes de stocker, récupérer et traiter. Les opérations de stockage et de traitement des données sont gérées par les groupes en charge, mais les données récupérées sont d'abord analysées par le *SDM*. Si l'information demandée appartient à l'historique ou si la date de dernières valeurs stockées est inférieure d'une valeur, les données sont récupérées du groupe *Stockage*. Si l'information demandée n'appartient pas à l'historique, la requête est renvoyée vers le *SSEP* qui doit demander l'information au WSN.

## **Règles de gouvernance du réseau de recouvrement**

- **Protocole**

Le protocole du réseau de recouvrement permet aux nœuds de se communiquer en utilisant le même langage. Il définit les règles de la syntaxe et la sémantique du réseau. Ce protocole doit être capable d'accomplir les exigences de la passerelle ainsi que les services et les caractéristiques héritées des réseaux pair-à-pair.

Comme protocole du réseau de recouvrement on a sélectionné le Protocole d'Initiation de Session (SIP). SIP est un protocole standard, largement déployé et simple à étendre. En outre, il a déjà été étendu pour travailler avec des dispositifs à capacité limitée.

Deux extensions facilitent l'utilisation dans notre réseau. La première comprend les fonctions SIP SUBSCRIBE/NOTIFY, décrit dans le RFC 3265. La deuxième est le message SIP INFO présenté en RFC 2976.

- **Processus**

Le protocole qui gère la communication entre les nœuds a été défini dans la section précédente. Voyons maintenant comment cette communication est réalisée au moyen de l'échange des messages qui permettent le fonctionnement de la passerelle.

### **Auto-organisation**

La passerelle devra être disponible dès qu'un *puits* est disponible sur le WSN. Par conséquent, le premier nœud qui doit joindre la passerelle est le *SSEP*, qui offre le point d'entrée au puits.

Une adresse multidiffusion qui est connue par tous les nœuds du réseau est assignée aux nœuds du groupe d'interconnexion. Le mécanisme que chaque nœud utilise pour découvrir autres nœuds dans le réseau est l'adressage multidiffusion de SIP (*SIP multicast addressing*).

Quand un nœud se connecte au réseau, on suppose qu'il connaît les rôles et les groupes qu'il peut jouer. D'abord il envoie un message SIP INFO à l'adresse multidiffusion en demandant l'adresse IP du nœud leader pour chaque groupe qu'il doit rejoindre. S'il n'y a pas encore de leader pour ce groupe, le nœud devient le leader. Cela se fait en changeant le rôle et en habilitant la réception des messages SIP SUBSCRIBE. Sinon, si le leader existe déjà, le nœud envoie un SIP SUBSCRIBE au nœud leader afin de rejoindre le groupe.

On veut que tous les nœuds du groupe connaissent les membres du groupe. À cette fin, chaque fois que le groupe change, le leader envoie un message SIP INFO avec l'action « *groupInfo* » et la liste de tous les membres.

Dans le cas des rôles de point d'entrée (i.e. *SEP* et *PSE*), les données arrivent aux nœuds des réseaux externes et doivent être renvoyées vers les leaders. Dans ce cas, dès que le nœud rejoint le groupe, le leader envoie aussi un message SIP SUBSCRIBE qui lui permet de recevoir l'information récoltée par les nœuds du groupe.

Dès que les groupes formés sont conformes, le *SPSE* s'enregistre à IMS en envoyant un message SIP REGISTER.

### **Auto-dépannage**

L'auto-dépannage est requis en deux occasions : soit un nœud veut sortir du réseau ou soit une panne arrive. À cette phase du projet, nous ne considérons que le premier cas.



Les nœuds sont informés du départ d'un nœud puisque ce dernier envoie un message de type SIP INFO avec l'action « *leaving* ».

Si le nœud est simple (c.-à-d. il n'est pas le leader du groupe), c'est son leader qui analyse la situation. Dans le cas des rôles de points d'entrée (i.e. *SEP* ou *PSE*), le leader choisit un autre nœud pour le remplacer en se connectant au *puits* mobile ou via IMS. Dans le cas du nœud de stockage, l'information est récupérée et redistribuée.

Si le nœud qui quitte est un leader, il informe le groupe des leaders et les nœuds des autres groupes. Les nœuds simples choisissent un nouveau leader et l'annoncent au groupe des leaders duquel il fera partie. Pour le *SDM*, les nœuds de stockage ont la responsabilité de choisir le nouveau leader.

### **Gestion de l'information**

Les processus de gestion de l'information nécessaires au fonctionnement de la passerelle sont : Traduction, Stockage, Extraction, Traitement et Publication de l'information.

La traduction est faite en deux parties. La première partie est réalisée entre le *puits* et la passerelle et elle est effectuée par le *SSEP*. La deuxième partie est réalisée à destination du module IMS, par le *SPSE*, qui traduit du modèle d'information de la passerelle au modèle d'information utilisé par le service d'IMS et vice versa.

Le stockage commence au niveau du *SSEP* lorsqu'il reçoit des données qui doivent être stockées. Il envoie un message SIP INFO avec l'action « *store* » vers le *SDM* qui le renverra comme SIP NOTIFY aux nœuds du groupe.

L'extraction de données commence en *SPSE* grâce à une demande d'IMS. Il envoie un SIP SUBSCRIBE au *SDM* qui le renvoie vers le groupe de stockage et attend la réponse. Une fois la réponse arrivée, le *SDM* l'analyse. Si ce dernier détermine que les données ne sont pas appropriées, il envoie un message SIP SUBSCRIBE vers le *SSEP* afin de les récupérer du réseau de capteurs. La réponse sera donnée au *SPSE* par le *SDM*.

Le traitement de l'information est géré par le *SDM* qui reçoit un message SIP SUBSCRIBE en demandant le traitement de données. Il renvoie cette pétition vers un nœud au groupe de *Processing* et envoie la réponse dans un message SIP NOTIFY.

Finalement, la publication est gérée par le groupe « IMS Entry Point ». Le *SPSE* reçoit un message SIP PUBLISH avec les données à être publiées et le renvoie vers le *PSEs*. Ces derniers renvoient un message SIP PUBLISH vers les serveurs IMS.

- **Modèle d'information**

Le corps du message SIP INFO doit être défini par chaque application. On a décidé d'utiliser un modèle d'information indépendant à l'intérieur de la passerelle. Ce modèle est présenté à la figure 3. Il est basé sur XML et sert à décider quelle action sera exécutée, les paramètres requis et toute information additionnelle.

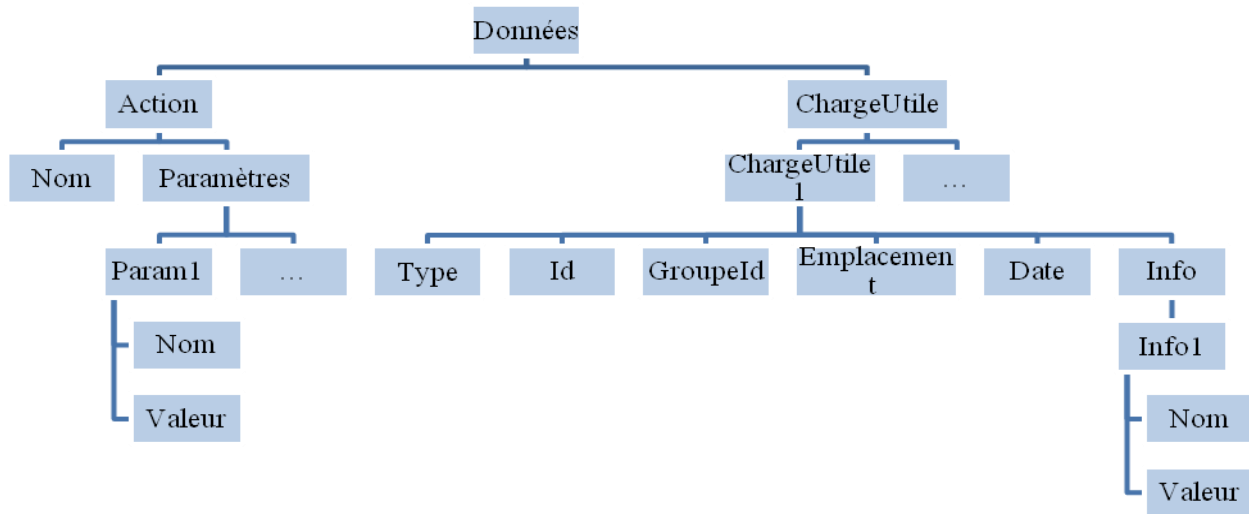


Figure 3. Modèle d'information

#### 4. Validation

Cette partie montre le processus de développement et de validation du réseau de recouvrement qui est la passerelle. Premièrement, l'architecture du logiciel sera présentée avec un prototype. Par la suite, nous validerons l'architecture à partir des exigences initialement

proposées et en effectuant des comparaisons avec les autres modèles d'intégration présentés à la revue de la littérature.

## Implémentation

### • Architecture du logiciel

L'architecture du logiciel devra être distribuée à chaque nœud qui veut faire partie du réseau de recouvrement. L'architecture est au-dessous d'un intergiciel pair-à-pair. Cette couche intermédiaire permet l'indépendance de l'application et la gestion de fonctions communes aux systèmes pair-a-pair.

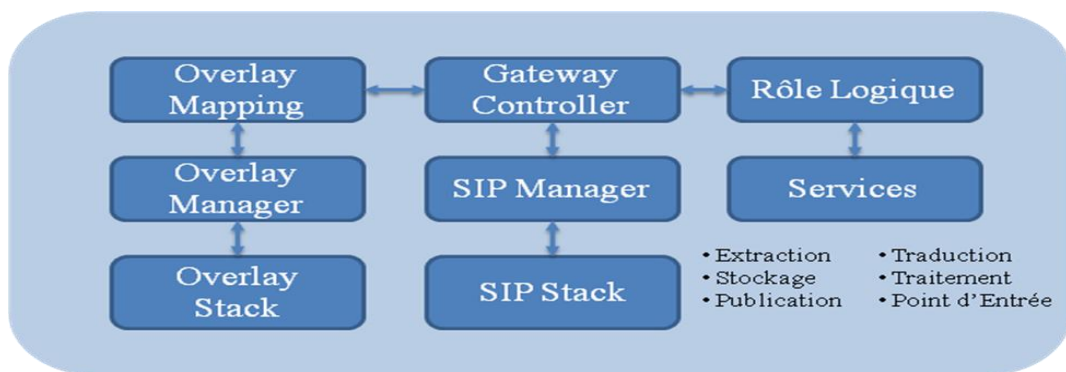


Figure 4. Architecture d'un nœud

Cette architecture est divisée en certains modules communs : « Overlay Manager », « Overlay Mapping », « Gateway Controller » et « SIP Manager ». Les modules sont catégorisés selon le rôle que joue le nœud : « Rôle Logique » et « Services ». Le *puits* mobile aura aussi ce logiciel installé, mais il s'agit d'une version simplifiée qui lui permet de se connecter à la passerelle, d'obtenir l'information spécifique et de se déconnecter.

« Overlay Manager » permet aux nœuds d'avoir un identificateur unique (PeerID) ainsi que de découvrir le numéro d'identification d'autres nœuds. Grâce au module « Overlay Mapping », l'architecture peut être indépendante de l'intergiciel. La traduction entre les primitives de l'intergiciel et le modèle d'information de l'application est effectuée.

Nous avons déjà mentionné que SIP est le protocole du réseau et que pour communiquer, le nœud utilise les modules « SIP Manager » et la pile de SIP

(représentée par l'entité *SIP Stack* à la figure 4). Il permet le fonctionnement de tous les processus décrits à la section précédente.

« Gateway Controller » est le contrôleur du logiciel. Il contient la logique qui permet la communication entre tous les modules. Il connaît aussi les rôles qu'un nœud joue et les groupes auxquels il appartient.

Finalement, les deux modules dépendants du rôle sont « Rôle Logique » et « Services ». Le premier contient l'information du rôle, son nom, les groupes qu'il doit rejoindre, les services qu'il doit fournir et la façon de lire et écrire les messages aux membres des groupes. « Services » possède l'implémentation de chaque service fourni par chaque rôle.

- **Prototype**

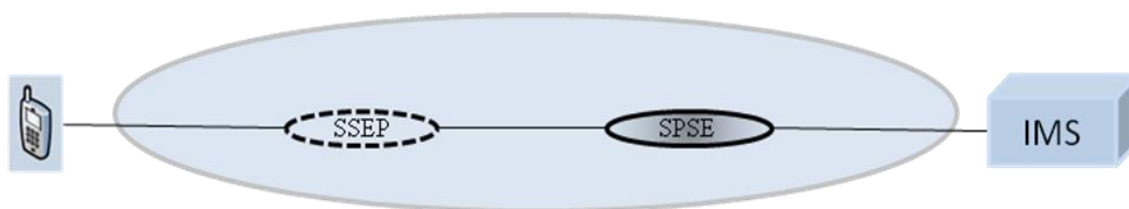


Figure 5. Scénario prototype

Le scénario proposé pour le prototype est présenté à la figure 5. Les actions sont les suivantes :

- Le *SSEP* se connecte au réseau
- Un *PSE* rejoint le réseau et devient le leader *SPSE* car il n'y a pas encore de leader pour le groupe de Point d'Entrée d'IMS. Il envoie un message SIP REGISTER vers les serveurs IMS.
- Le *puits* envoie un message vers la passerelle.
- Le *SSEP* reçoit l'adresse et se connecte au *puits*.
- Le *puits* envoie des données au moyen d'un message SIP INFO.
- Le *SSEP* analyse l'information : puisque la température dépasse la valeur limite, il envoie un message SIP PUBLISH vers le *SPSE*.
- Le *SPSE* envoie un message SIP PUBLISH vers IMS.

Les deux nœuds de la passerelle sont représentés par deux ordinateurs portables. IMS a été simulé dans un autre ordinateur avec le SDS (Ericsson's Service Development Studio). Le *puits* était un processus exécuté dans un autre ordinateur qui envoyait de l'information à chaque 3 secondes.

Le développement du logiciel a été fait avec J2ME (Java Micro Edition) et l'éditeur NetBeans IDE 6.8. Le middleware sélectionné est JXME, un sous-ensemble du JXTA. En ce qui concerne l'implémentation de SIP, nous avons utilisé le JSR180 (SIP API for J2ME) qui possède la fonctionnalité d'un utilisateur SIP.

Le scénario a été exécuté avec succès. La figure 6 montre les écrans des deux nœuds pendant le processus de connexion.



Figure 6. Prototype, nœuds *PSE* et *SSEP*

## Passerelle vs. Exigences

Le tableau 1 présente comment chaque exigence architecturale est satisfaite par une ou plusieurs caractéristiques.

Table 1. Passerelle vs. Exigences

	Passerelle comme intégrateur	Réseau P2P	Définition des services	PIDF Étendu	Modèle d'information	Dispositifs mobiles
<b>Indépendance</b>	X				X	
<b>Types d'information</b>				X		
<b>Traduction</b>			X		X	
<b>Publication</b>			X			
<b>Stockage</b>		X	X			
<b>Traitement des données</b>			X		X	
<b>IMS traditionnel et pair-à-pair</b>	X	X				
<b>Extensible</b>		X				
<b>Tolérant aux pannes</b>		X				
<b>Communication légère</b>						X

L'indépendance est réalisée grâce à l'utilisation d'un passerelle comme intégrateur et à l'utilisation d'un modèle d'information indépendant de la passerelle. Les types d'information en WSN sont disponibles en IMS. Cela est accompli par le PIDF-étendu. La même traduction est satisfaite par la définition d'un service en charge de cette action et par le modèle d'information. La publication est fournie comme service par le groupe IMS Entry Point, ainsi que le traitement des données. De plus, le stockage est présent comme service. Les exigences de stockage sont satisfaites également en utilisant un réseau de recouvrement comme passerelle, ce qui permet d'avoir un stockage distribué. Également, IMS traditionnel et pair-à-pair sont réalisés au niveau du design et de l'implémentation au moyen d'un réseau de recouvrement comme passerelle, ce qui permet que les intégrations avec IMS pair-à-pair soient plus transparentes dans le futur. Le fait d'avoir le réseau de recouvrement permet de considérer cette solution comme extensible et tolérante aux pannes. Finalement, le fait de construire notre réseau par-dessus des dispositifs mobiles permet de s'assurer que la communication soit légère.

## Passerelle vs. Revue de la littérature

Nous allons maintenant comparer le fonctionnement de notre architecture par rapport aux architectures proposées par des pairs et qui ont été présentées à la revue de la littérature. Les propositions choisies sont : Passerelle WSN/IMS par El Barachi et *al.* [3], Architecture e-SENSE par Gluhak et Schott en [5] et la passerelle PN par Arbanowski et al en [4].

Table 2. Comparison  
S: Supported, N: Not supported, P: Partially supported and N.C.: Not considered

Exigence	Passerelle P2P	WSN/IMS passerelle	Architecture e-SENSE	Passerelle PN
<b>Indépendance</b>	<i>S</i>	<i>S</i>	<i>N</i>	<i>S</i>
<b>Types d'information</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>Traduction</b>	<i>S</i>	<i>S</i>	<i>N</i>	<i>N</i>
<b>Publication</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>Stockage</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>Traitement des données</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>IMS traditionnel et pair-à-pair</b>	<i>S</i>	<i>N.C.</i>	<i>N.C.</i>	<i>N.C.</i>
<b>Extensible</b>	<i>S</i>	<i>N</i>	<i>N</i>	<i>P</i>
<b>Tolérant aux pannes</b>	<i>P</i>	<i>N</i>	<i>N</i>	<i>N</i>
<b>Communication légère</b>	<i>S</i>	<i>N</i>	<i>N</i>	<i>S</i>

On peut noter que notre architecture est la seule qui satisfait toutes les exigences du système. La seule exigence qui est partiellement accomplie est la tolérance aux pannes, car nous n'avons pas défini une solution au processus de départ involontaire des nœuds.

## 5. Conclusion

Dans le cadre de ce projet, nous avons proposé une architecture basée sur un réseau de recouvrement qui sert de passerelle pour connecter les réseaux de capteurs au *puits* mobile et à l'architecture IMS. Cette architecture permet de fournir des nouveaux services aux utilisateurs en IMS ainsi qu'une utilisation plus efficace et optimale de l'information récoltée dans le réseau de capteurs.

Elle ne requiert aucune infrastructure additionnelle car elle est construite au-dessus des mêmes dispositifs qui s'utilisent comme *puits* en WSN et clients en IMS.

Pour la validation de concept de notre proposition, nous avons développé et exécuté un prototype en J2ME, en utilisant le JSR180 comme implémentation de SIP et JXME comme middleware.

### **Limitations**

Les limitations de cette proposition correspondent aux suppositions qui ont été définies pendant le design de l'architecture. Premièrement, nous avons parlé d'un accord entre fournisseurs qui permet la communication entre les périphériques des trois réseaux, *i.e.*, WSN, IMS et la passerelle. Cette hypothèse pourrait s'avérer invalide dans certains contextes.

De plus, nous avons supposé que les appareils cellulaires sont capables de communiquer entre eux pour construire un réseau de recouvrement. Cette supposition limite l'architecture à un certain nombre de dispositifs possibles.

Finalement, une de nos hypothèses portait sur un format de message supposé qui est envoyé par le WSN et traduit au format PIDF-étendu. Cette supposition limite l'architecture à fonctionner avec le format spécifié, ce qui n'est pas le cas pour le WSN.

### **Publications**

Un article de conférence basé sur ce travail a été présenté et approuvé pour le « 2010 IEEE 72nd Vehicular Technology Conference: VTC2010-Fall that will be held 6–9 September 2010 in Ottawa, Canada ».

### **Travaux futurs**

La définition du départ involontaire serait la prochaine étape à réaliser. Le processus de récupération et les actions dépendamment de chaque rôle devront être considérés.

L'extension du prototype pourra être aussi un autre travail futur, parce que celui qui a été fait n'est qu'un sous-ensemble très simple. Il faudrait ajouter d'autres services, rôles et algorithmes pour prendre en considération les départs involontaires.

Un prototype qui se connecte à IMS pair-a-pair pourrait être analysé. Finalement, nous pourrions tester un véritable déploiement de cette architecture dans une ferme.



## INDEX

DEDICATION .....	III
ACKNOWLEDGEMENTS .....	IV
RÉSUMÉ.....	V
ABSTRACT .....	VI
CONDENSÉ EN FRANÇAIS .....	VII
INDEX .....	XXV
INDEX OF TABLES .....	XXVII
INDEX OF FIGURES.....	XXVIII
LIST OF ACRONYMS AND ABBREVIATIONS .....	XXIX
Chapter 1 INTRODUCTION.....	1
1.1 Definitions and basic concepts .....	2
1.1.1 Wireless Sensor Networks (WSNs) .....	2
1.1.2 IP Multimedia Subsystem (IMS).....	4
1.2 Aspects of the problem.....	6
1.3 Research objectives .....	8
1.4 Outline.....	8
Chapter 2 STATE OF THE ART.....	9
2.1 WSN .....	9
2.2 IMS & the Presence Framework .....	11
2.3 P2P systems.....	15
2.4 Scenario and requirements for the integration .....	16
2.5 Integrating WSNs with Internet and other specific solutions.....	18
2.6 Integrating WSNs and IMS Architecture .....	20

Integrating WSN and IMS.....	20
2.7 Architectures for system integration .....	23
Chapter 3 PROPOSED ARCHITECTURE .....	25
3.1 Architecture design.....	25
3.1.1 Assumptions .....	26
3.1.2 Architecture principles .....	26
3.1.3 Peer roles, interactions and topology .....	27
3.2 Overlay rules .....	32
3.2.1 Protocol .....	32
3.2.2 Procedures .....	33
3.2.3 Information Model .....	41
3.3 Scenario.....	44
Chapter 4 IMPLEMENTATION AND VALIDATION.....	46
4.1 Implementation.....	46
4.1.1 Software Architecture.....	46
4.1.2 Prototype .....	49
4.2 Validation .....	55
4.2.1 Requirements vs. Architecture .....	55
4.2.2 Architectures comparison.....	57
Chapter 5 CONCLUSION .....	60
5.1 Summary of the work .....	60
5.2 Publications .....	61
5.3 Future Works.....	62
REFERENCES.....	63

**INDEX OF TABLES**

Table 4.2-1 Requirements vs. Architecture.....	56
Table 4.2-2. Comparison between proposed gateways for integrating WSN and IMS .....	58

## INDEX OF FIGURES

Figure 1.1.1. IMS Architecture .....	5
Figure 2.2.1. IMS simplified architecture overview .....	12
Figure 3.1.1. General architecture .....	26
Figure 3.1.2 Overlay Gateway .....	27
Figure 3.1.3. Global topology .....	31
Figure 3.2.1. Flow diagram for a node joining the overlay.....	35
Figure 3.2.2. Flow diagram for node departure.....	37
Figure 3.2.3. Information storage process.....	39
Figure 3.2.4. Information retrieval process .....	40
Figure 3.2.5. Information publication .....	41
Figure 3.2.6. Gateway Information Model.....	42
Figure 3.2.7. Information Model, Retrieval process .....	43
Figure 3.3.1. Sequence diagram for agricultural scenario, moisture below threshold .....	44
Figure 4.1.1. General architecture .....	46
Figure 4.1.2. Peer architecture.....	47
Figure 4.1.3. Role Logic structure.....	49
Figure 4.1.4. Proof of concept.....	50
Figure 4.1.5. Prototype .....	52
Figure 4.1.6. Sequence diagram joining node – discovery request.....	53
Figure 4.1.7. Sequence diagram SPSE - receiving discovery request.....	54

## LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
CSCF	Call/Session Control Function
DNS	Domain Name System
HSS	Home Subscriber Server
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
I-CSCF	Interrogating-CSCF
LR-WPAN	Low Rate – Wireless Personal Area Network
P2P	Peer-To-Peer Systems
PA	Presence Agent
PEA	Presence External Agent
PN	Personal Network
PUA	Presence User Agent
P-CSCF	Proxy-CSCF
QoS	Quality of Service
SIP	Session Initiation Protocol
S-CSCF	Serving-CSCF
WSAN	Wireless Sensor Actuator Network
WSN	Wireless Sensor Network

# Chapter 1 INTRODUCTION

Recently, applications and technologies are rarely seen isolated. Users expect to have access to a set of services without needing to know which system handles which service. Thus, systems integration and standardization has become a key-point toward ubiquitous access.

This project is focus in integrating two technologies, the IP Multimedia Subsystem (IMS) and Wireless sensor networks (WSNs). Both are evolving technologies and have gain interest in the research community in recent years because of the advantages and services they provide. As will be proven in this work, the integration of both provides even further services and applications in a standardized and decoupled environment.

WSNs have been widely researched and studied. This type of network can collect specific information from the environment, thus enabling applications such as healthcare, home automation, object tracking and environmental monitoring at a low cost. They are made up of small scale devices that can sense, compute and communicate [1] called sensor nodes that send data to a generally centralized unit called *Sink*.

One of the technical challenges found in sensor networks is the lifetime of the network which significantly depends on energy consumption by each component. Basically, the design of a sensor network is attached and dependent to this factor. One solution that has been proposed to improve the overall network lifetime is mobile sink-based WSNs. Mobile sink nodes move along the deployed field and sensors transmit collected data when a sink is closer thus saving energy. This approach allows the implementation of efficient data collection, sensor reprogramming and data querying exploiting the mobility nature.

IMS, on the other hand, is one of the key elements of 3G networks [2]. Its goal is to provide ubiquitous access to Internet services. It is based on three main characteristics:

1. Provisioning of Quality of Service (QoS) to real-time sessions. That allows an operator to control the service a user gets; considering that it could vary from one user to another.
2. Allowing a fair charging scheme to multimedia sessions. This is an important factor since sessions can be created among users or services located in different networks.

3. Integrating services through standard interfaces. IMS enables services such as instant messaging, conferencing and third-party call to cellular user through standard interfaces.

The contextual information collected by WSNs and the multimedia capabilities of IMS can be combined to provide a wide range of novel applications. When information from WSN reaches IMS, applications and users could analyze it to provide effective and real-time feedback.

Integration between both technologies has already been proposed in [3-5]. However, no decentralized solution has been proposed that provides at the same time independence, scalability, fault-tolerance and essential data management processes (e.g. translation, processing, storage).

Bringing P2P advantages to the integration could provide scalability, fault-tolerance and resilience to the solution. Peer to peer (P2P) is a computing paradigm where each node acts as both client and server [6] and have been proven to increase scalability, availability and resource utilization over traditional client-server systems.

This project proposes a decentralized architecture that enables the integration between IMS (traditional and P2P) and mobile sink based – WSNs.

## **1.1 Definitions and basic concepts**

This section presents the background information pertinent to this project. Its goal is to give the reader a better understanding as to what are the networks and the technologies used as base for the integration.

Firstly a brief survey of WSNs, specifically focusing on mobile sink WSNs, is presented. Later on, IMS is introduced as technology and is explained in further detail.

### **1.1.1 Wireless Sensor Networks (WSNs)**

A sensor network is generally made up by a large number of sensor nodes deployed with or without a pre-established structured in a specific area in order to monitor and to collect information from the environment [1]. Since they do not necessarily rely on any pre-established infrastructure, sensor networks can be deployed randomly in inaccessible terrains.

Two basic entities are found in a sensor network: a sink and sensor nodes. The sink, generally centralized, gathers the information sent by the sensors; it also sends control information towards the nodes and acts as an interface to the user. Each sensor node is composed by, at least, four main components: A transmission or transceiver unit that provides network capabilities; A sensing unit that collects the information from the environment; A processing unit, which manages the procedures to collaborate with other nodes and has a small storage unit; And a power source that gives energy to the node and that is usually in the form of a battery.

A wide range of applications has been enabled by sensor networks but two main categories can be established: monitoring and tracking. Monitoring applications include healthcare, environmental monitoring, power monitoring and traffic control among others. While tracking serves to object surveillance purposes such as: animal, human or vehicle tracking.

WSN can be categorized using many other aspects; one of them being the mobility of the network. A network could be composed by mobile/fixed sensor nodes and mobile/fixed sinks.

### **Mobile WSN**

Mobile sensor network refers to a WSN with a dynamic nature, i.e. mobile sinks and/or nodes are an essential part of it. Here we focus on mobile sink sensor networks whose infrastructure can rely solely on mobile sinks or on a combination of both mobile sinks and fixed collectors.

In mobile sink-based WSN, sensor nodes communicate and send information when a sink gets close. The sink moves along the deployed field autonomously or in a predetermined path. Predetermined paths can be determined as to provide the optimum path when collecting information from sensors. In an autonomous way, sinks can change their location depending on nearby sensors energy.

Although mobility was considered an issue when designing WSN, recent work has been done in exploiting this characteristic and benefiting from it [7-9]. Generally, mobile WSNs have been proposed as a solution to improve the overall network lifetime, the covered area and the data capacity. Their feasibility highly depends on the application, since the solutions proposed considered benefits in a long time interval and not necessarily at a specific time which could not be allow in some scenarios.



The overall network lifetime can be improved in sensor networks using mobile sinks, mobile sensors or mobile relays [8]. Efficient energy consumption is one of the critical factors behind this enhancement. In a static sensor network, where nodes have a pre-established infrastructure, sensors closer to the sink consume more energy than those deployed farther. However in mobile sensor networks, the energy consumption can be minimized when exploiting the mobility factor of the sink and the sensors.

Additionally the network coverage can be extended when considering sensors mobility [9]. As exposed by the authors, the area covered by sensors changes in time, meaning that a location alternates between being covered and uncovered. Although not all the field can be covered at a given time, during a time interval more area will be covered; resulting in an improvement of coverage. Furthermore, environment factors that could not be sensed with stationary sensors can be now detected.

WSN design highly depends on the intended application since cost, time, reliability and environmental constraints can considerably change from one application to another. Mobile WSN inherits most characteristics from WSN (energy limitation, routing, and redundancy) but they also have some specific issues that are crucial due to the dynamic nature of these networks such as: Self-organization with mobility, nodes localization, maintaining network connectivity, data distribution, mobility management and coverage.

Despite the advantages they provide, these challenges need to be considered and continue to be a subject of research.

### **1.1.2 IP Multimedia Subsystem (IMS)**

IMS has been proposed as a standard architecture that provides ubiquitous access to Internet services everywhere using cellular networks. As explained by Camarillo in [2], “Third generation (3G) networks aim to merge two of the most successful paradigms in communications: cellular networks and Internet”.

The advantage of using IMS is the provisioning of service integration through a standard architecture, the consideration of QoS and a differential charging scheme. Firstly, integration of services allows operators and users to access different services in the same way by using standard interfaces. Furthermore, it allows operators to build and provide new services combining or

integrating existing ones. Additionally, IMS deals with the synchronization in session establishment considering QoS provision, which improves considerably real-time communications, important for example in multimedia sessions. Finally, in IMS charging is determined by the operator and depends in the services used by the user and not by the bytes he/she transfers as it is done with today's architecture.

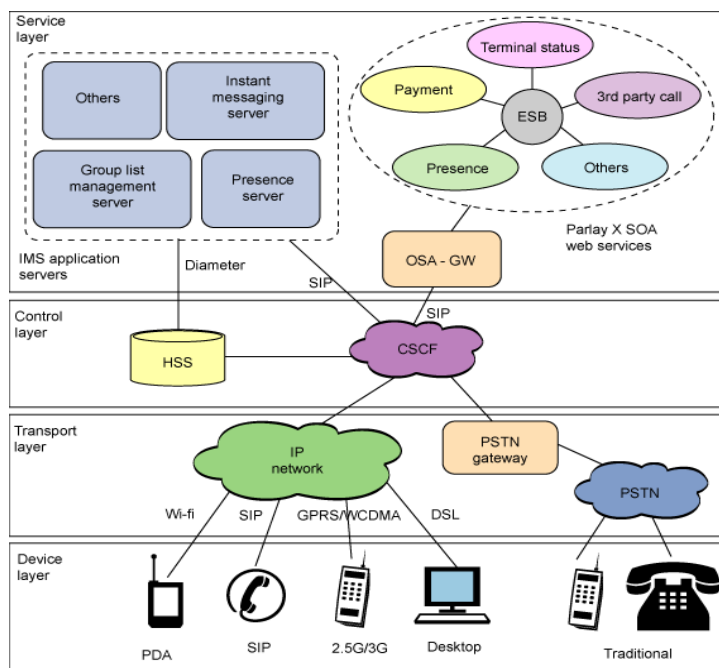


Figure 1.1.1 IMS Architecture

IMS is an overlay control layer on top of an IP layer. As shown in Figure 1.1.1, four logical layers are defined as part of the IMS architecture. The device layer represents the different networks that could access IMS when connecting to an IP network. The transport layer is responsible for initiating and terminating SIP sessions and providing conversion of data. In addition to the IP network, the transport layer allows IMS devices to make and receive calls to and from the PSTN.

The control layer is composed by the CSCF (Call/Session Control Function) and the HSS (Home Subscriber Server), which are essential entities in the IMS architecture. CSCF refers to a set of SIP servers (i.e. Proxy-CSCF, I-CSCF and S-CSCF) that process SIP signaling. And the HSS is a central repository that contains user-related information.

Finally, on top of the IMS network architecture is the service layer. At this layer, application servers are found. An application server is a SIP entity that hosts and executes

services. One of the services already being provided at this layer is the Presence Service that will be further explained in chapter 2.

## **P2P IMS**

Researchers have begun to study a new area called P2P IMS. Two approaches can be identified so far, P2P-over-IMS or IMS-over-P2P. The first approach refers to the provisioning of P2P services through the traditional IMS architecture as proposed in [10] and [11]. The second one is focused mainly in SIP P2P to IMS; being SIP the signaling protocol defined on IMS architecture. Further details are given in [12-14].

All of these studies proposed at first or at a future stage, the implantation of P2P in IMS. This approach is known as P2P-SIP. Although two User Agents can communicate directly, most deployments require SIP servers that provide connectivity (rendezvous, session initiation and multimedia session management). The idea with P2P-SIP is to decentralize this location service. This project is currently being standardized by Internet Engineering Task Force (IETF) [15]. From which we can see the proximity of a realistic scenario and thus we should consider it when designing our prototype.

## **1.2 Aspects of the problem**

Integrating WSN and IMS opens the door to a wide range of applications. Basically it allows the provisioning of new services and an optimized and efficient use of data.

On one hand, optimization and efficiency could be achieved once the information collected by the WSN reaches IMS. To further explain this case we could analyze a scenario for agricultural purposes where sensors are deployed at various locations on a vineyard to measure aspects such as: soil moisture, temperature and pest risks. Information collected by the sensors is gathered by multiple mobile sinks that a farm manager and workers have. They collect the information when passing by the crop rows using their PDAs or mobile devices. When there is no sink close by, fixed collectors that can reach the mobile devices over a long range can receive the information to resend it to a mobile sink. Once the mobile sink has the information it will forward it to IMS. There, it will be sent towards an application server that can analyze the data and act upon it. For example, when soil moisture is below a threshold in a specific area,

multimedia services in IMS could be sent to a specific worker(s) noting that irrigation systems should be activated there.

In agricultural applications, the data gathered become important and useful only if users can have access to it in real time and anywhere they are. These two requirements can be fulfilled by IMS, where multimedia services are already available; text messages (SMS), multimedia messages (MMS) and even instant messages (IM).

On the other hand, provisioning of new services or integrations could be developed in the future since the data will already be available in IMS. Proactive actions exposed in [16] include: irrigation system activation, pesticide sprays and frost detection in winter that could be automatically activated by other networks such as Sensor Actuator Networks when integrated with IMS. This is possible thanks to the standardization, unification and integration of services proposed by IMS.

To successfully enable the above benefits, some minimum requirements need to be accomplished from the integration's point of view. Independence is one of them; the integration should not only require minimum changes in the two technologies involved but should also be independent enough to work with both traditional and P2P IMS. Additionally efficient translation needs to be done to merge the contextual information from WSN into IMS. Furthermore, to proactively react and optimize the use of information, the integration should consider storage and processing capabilities. And finally, since mobility is considered in WSN sinks and in IMS users, the integration needs to allow nodes to join and leave the networks, thus scalability and fault-tolerance are required.

Several solutions have been proposed to bring information from sensor networks to other networks or technologies like IMS. A common approach to the integration often used in the literature adds a centralized entity or gateway that connects both technologies [3-5]. These models present some drawbacks mainly because scalability, fault tolerance and lightweight communication mechanisms are not supported or not specified.

To the best of our knowledge, no decentralized architecture has been provided considering mobile sink-based sensor networks and IMS. A decentralized architecture based on P2P systems will bring the benefits from P2P into the integration core point; scalability and fault-tolerance could be improved.

### 1.3 Research objectives

Our main goal is to design and build a mobile phone-based gateway that integrates IMS and mobile sink-based WSNs, enabling data dissemination between the sensor network and interested users in traditional and P2P IMS architecture. More precisely, this research project has the following goals:

- Define a set of application-scenarios for the integration of mobile sink WSN into traditional and P2P IMS.
- Propose an architecture for the integration of both technologies.
- Design the software architecture for the mobile phone-based gateway that enables the integration.
- Develop a prototype of the gateway as a proof of concept.
- Validate the proposed architecture.

### 1.4 Outline

The rest of this dissertation is organized as follows. Chapter II presents the state of the art regarding the integration of wireless sensor network with Internet services and, more specifically, with IMS. In chapter III the proposed architecture is exposed. Chapter IV is devoted to the implementation of the prototype and the analysis of the architecture. Finally, we give the conclusions and future work.

## Chapter 2 STATE OF THE ART

This chapter presents the state of art from the different technologies and integrations conceived in this project. Firstly, we will further explain sensor networks and show some of the standards that have been proposed so far. After that, IMS architecture will be described and explained, as well as the presence framework that plays an important role in our integration. The third section shows the scenario that is the base to analyze and define the requirements for the system. Then a brief survey on the integrations that have been done between WSN and other technologies apart from IMS, specifically connecting sensor networks to Internet will be depicted. Later on, the benefits from integrating WSN and IMS are presented along with proposals that have been done in the literature. Finally, benefits and weaknesses of current overlay solutions for systems integration are presented.

### 2.1 WSN

A wireless sensor network is generally composed by a large number of sensors that are deployed over an area to detect or sense a phenomenon. These sensor nodes are seen as sources of information, providing a rich set of contextual information such as: Spatial data (e.g. location), physiological data (e.g. heart rate, blood oxygen or blood pressure) or environmental data (e.g. temperature or soil properties). Additionally, they have the ability to compute simple tasks and to communicate with other nodes to transfer information or to perform other networking activities (e.g. routing).

The *Sink* or base station is the gateway or coordinator of the network. It collects information from sensor nodes and queries them if necessary. Besides the basic functionality, this functional entity is the entry point to the sensor network. Applications can create reports or analyze the data collected throughout the *Sink*. It has been generally conceived as a single and centralized entity in the WSN, however depending on the application several sinks and/or a decentralized architecture can be deployed.

Sensor networks have several features that made them unique. Among them, sensor nodes are densely deployed and prone to failures, the topology of the network varies in time due to

mobility or failure factors, the nodes are limited in power, computational capacity and memory, and they may not have a unique global ID because they are densely deployed.

All these characteristics are being studied by researchers and several proposals for protocols, standards and algorithms that could fulfill them in a satisfying way have been made. Surveys in WSN [1, 17] have gathered this information and outline new challenges. Akyildiz et al. present in [17] a survey based on the communication architecture of wireless sensor networks and the proposals and challenges in the different layers (e.g. physical, data link, etc.). While Yick et al. [1] described the state of the art by dividing it in three lines of study: internal platform and operating system, communication protocol stack and network services. The following section briefly describes some of the current standards for sensor networks, to further detail refer to [1].

## **Standards**

Wireless sensor networks requirements have been developed and design with low power consumption as a key characteristic. These requirements can change from different types of areas in sensor networks. Thus, several standards have been proposed. They include not only the functions but also the protocols necessary to interact with other nodes in the network. Some of the standards proposed for WSN are: IEEE 802.15.4, ZigBee, WirelessHART and IEEE 802.15.3.

IEEE 802.15.4 is a standard for low rate wireless personal area networks (LR-WPAN). LR-WPAN characteristics are ease of installation, reliable data transfer, short-range operation, extremely low cost battery life and a simple and flexible protocol. IEEE 802.15.4 allows the formation of either peer-to-peer or star network composed by the two types of devices defined by the standard; a full-function device (FFD) and a reduced-function device (RFD). The standard is designed for applications that require short range communication to maximize node lifetime, including residential, industrial and environmental monitoring, control and automation.

ZigBee is based in the IEEE 802.15.4 and defines the upper layers. It is designed for applications that require low data rate, long battery life, and secure networking. The standard defines three types of devices: ZigBee coordinator, ZigBee router and ZigBee end device. The ZigBee coordinator starts the network, can store information about it and can act as a bridge to other networks. ZigBee router links groups of devices and provides multi-hop communication. Finally, ZigBee end device limits its function to transmit information collected to the parent

nodes; includes sensors, actuators or controllers. ZigBee has been proposed as standard for home automation, healthcare monitoring and industrial control.

WirelessHART as ZigBee is based in IEEE 802.15.4. It provides a communication protocol for process measurement, control and asset management applications. One of the main advantages is that it is compatible with existing devices, tools and systems. Their architecture is based on three main features: reliability, security and effective power management.

IEEE 802.15.3 is a physical and medium access control layers in high rate WPAN. It has been enhanced to address the specific needs of digital imaging and multimedia applications; audio and video.

The design of these platforms is tightly coupled with a range of applications, since energy efficiency, cost and other design challenges can dramatically change with the application requirements. Although these standards provide support for a wide range of applications, new applications could be easily be defined, thus, a more practical platform could be useful.

## **2.2 IMS & the Presence Framework**

As described before, IMS aims to integrate services and to provide QoS and a differential charging scheme. IMS architecture proposes several functional entities linked by standardized interfaces; meaning that two functional entities could be implemented in a single physical node.



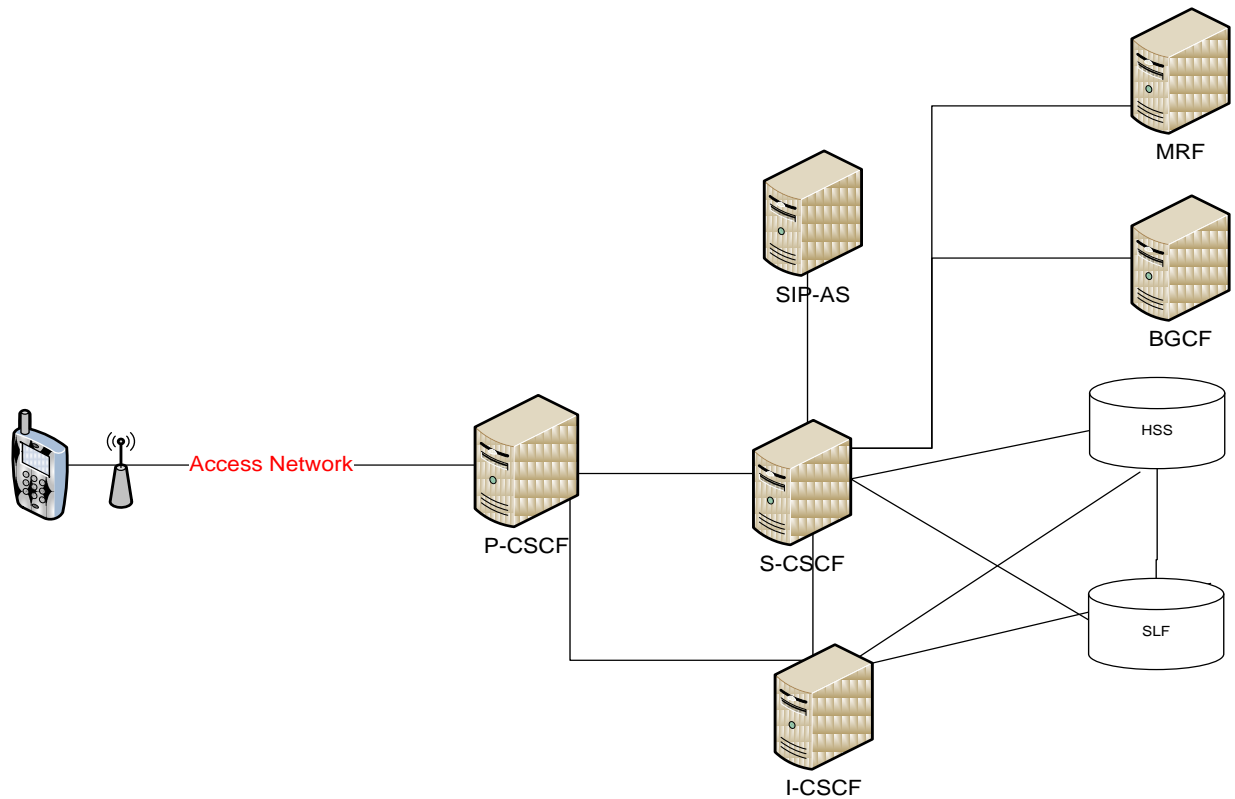


Figure 2.2.1. IMS simplified architecture overview

Figure 2.2.1 depicts a simplified version of the IMS architecture. On the left side of the picture there is the IMS mobile terminal, commonly named as User Equipment (UE). Its contact point in the signaling plane is the Proxy-CSCF (P-CSCF).

Two databases are defined, the Home Subscriber Server (HSS) and the Subscription Locator Function (SLF). The former is the central repository for user-related information. It contains information regarding location, security, user profile and the corresponding Serving-CSCF (S-CSCF) to the user.

The P-CSCF forwards SIP requests and responses towards the UE and the IMS network. It also handles several security functions, verifies, compress and decompress SIP messages and generates charging information.

The Interrogating-CSCF (I-CSCF) is a SIP proxy server located at the edge of an administrative domain. Its address is listed in the DNS (Domain Name System) records of the domain, thus providing access to its domain in case a SIP server requests to find the next SIP hop

for a specific message. Besides, the I-CSCF interacts with the HSS and the SLF where it retrieves information and routes it to the appropriate destination.

The S-CSCF is the central node in the signaling plane. It acts as a SIP server and performs session control as well. It maintains a binding between the user location (e.g. IP address) and its SIP address (the Public User Identity). It also interacts with both the HSS and the SLF. All SIP messages sent and received by the UE go through the S-CSCF; which analyzes it and determines, if any, the AS where the message should be forwarded before reaching its destination. It also manages some security features.

The Media Resource Function (MRF) provides a source of media to the network. It provides the ability of playing and mixing media streams. And it is divided in two main functions; a control or signaling part called MRFC (MRF Controller) and a media part called MRFP (MRF Processor).

The BGCF is a SIP server that allows interworking with users in a circuit-switched network such as the PSTN. It selects the appropriate network and gateway.

The AS is an entity located in the service layer that hosts and executes services. It connects with the control layer (i.e. S-CSCF and I-CSCF) by using SIP and the HSS by using Diameter. The AS can act as a SIP Proxy, a SIP UA or a SIP B2BUA (Back-to-Back User Agent). Three types are defined: SIP AS that executes services based on SIP; OSA-SCS (Open Service Access-Service Capability Server) which is an interface to the OSA framework, acting as an AS on IMS and an interface between the OSA AS and the OSA API; IM-SSF (IP Multimedia Service Switching Function) that allows the reuse of CAMEL (Customized Applications for Mobile network Enhanced Logic) services developed for GSM in IMS.

Most vendors follow the architecture closely by implementing each function into a single node, however as explained before it is possible to combine them in the same physical entity. At May 2008, there were over “30 commercial IMS networks running live traffic, adding up to over 10 million IMS users around the world” [2]. The most common applications offered by these commercial IMS architectures are IP telephony, messaging, video sharing and presence.

## Presence Framework

The presence service allows a user to be informed about the reachability, availability, and willingness of communication of another user. It can provide an extensive amount of information about a person to a set of interested users. Even more, services are able to read and analyze this information to provide further services.

The presence framework defines four roles:

- *Presence Entity* – “*Presentity*”. It refers to the entity providing presence information (e.g. status, capabilities and location). It has several *Presence User Agents* (PUAs) which provide this information to the Presence Service. Each PUA can collect different pieces of information.
- *Presence Agent (PA)*. It gathers information sent by the PUAs and obtains an idea of the user’s presence.
- *Presence Server*. It is a functional entity that acts as either a PA, as proxy server for SUBSCRIBE requests or as both. In IMS, this entity is represented as an Application Server that acts as a PA.
- *Watcher*. It refers the user that requests presence information from a *presentity*.

It is built on top of the SIP event notification framework; which is based in SUBSCRIBE/NOTIFY requests. A watcher subscribes to receive information from a *presentity* for a period of time or for requesting some specific information. The *presentity*’s PA will send the information to the watchers using a SIP NOTIFY request. Presence information is sent in the body of the messages and it is a XML document called PIDF. The PIDF carries the semantics of presence information between presence entities or roles. It is protocol independent and highly flexible; in fact some extensions have already been proposed to overcome some limitations.

Presence service is divided in three processes. The first one is the publication process where *presentity*’s PUAs send PIDF documents in a SIP PUBLISH message. IMS CSCFs forward the request to the Application Server that represents the *Presence Server*; which finally replies with an OK. The second process consists in the subscription of watchers. Through SIP SUBSCRIBE transaction, watchers request to receive information from a *presentity*, watchers can

be users or even other services. Once new information reaches the *Presence Server* a SIP NOTIFY is sent to subscribed users and services that can exploit the information.

## 2.3 P2P systems

The concept of P2P systems have been discussed and analyzed for several years. P2P networks consist of several peers that collaborate to achieve a collective purpose. Each node can play a different role in the network, thus providing different services that will allow the correct functioning. Some of its main characteristics are:

- *An overlay network.* Provides communication, integration and assistance in locating and communicating between peers.
- *Sharing of resources or services.* Each node can provide both client and server functionality. Examples of resources or services: information, files, bandwidth, storage and processor cycles.
- *Decentralization.* Communication is directly; theoretically speaking there is no central authority.
- *Autonomy.* Peers can decide what and when to share a resource in the network.
- *Interoperability.* Ability of any entity to speak to, exchange data with, and be understood by any other. Efforts to establish a common infrastructure for P2P applications with standardized interfaces have been done, some candidates are: JXTA, Magi, Web services, Jabber, and Groove

There are three types of P2P networks: purely decentralized architectures; partially centralized architectures and hybrid architectures. In the first type of architecture all peers in the overlay network perform exactly the same tasks, acting as clients and servers; thus there is no central authority. In the partially centralized architecture, some peers assume a more important and demanding role; these super peers act as local central indexes to find the resources or services. Finally, the third architecture includes a central server in the initiating phase that locates the resources and services and could also include super nodes as in the partially centralized architecture.

In general, P2P systems are more scalable, resilient and resource-efficient than client-server solutions. They have the ability to exploit unused resources through the network and have in general low cost of ownership.

## 2.4 Scenario and requirements for the integration

Commercial applications appear as a promising area for WSN thanks to their intrinsic features such as scalability and reliability joined to low cost deployments [18]. Specifically for agricultural purposes, WSN have been proposed to sense critical information to users allowing real-time access to information from the field. Soil conditions such as temperature, humidity, mineral content, lighting levels and pest exposure can be monitor to improve and optimize precision and quality of the final product. Cotton, wheat and wine production are some of the areas in which sensors have been considered.

Vineyard computing, specially, has been extensively studied because the quality of the wine relies on the quality of grapes. Important processes like irrigation, fertilization and pesticides are often left to the vineyard manager's discretion and could lead to unnecessary used of chemicals and water [16].

WSN is a technology suitable to fit different requirements when monitoring conditions in a vineyard. Sensors are deployed directly in the field between plants and in the soil with a network infrastructure that could run for years without man attendance. Akyildiz and Stuntebeck in [19], for instance, propose a hybrid topology composed by underground sensors and a mobile terrestrial sink, where the sink moves around the surface and collects data from sensors or relay nodes.

This proposal could be adapted to our case where underground and aboveground sensors are deployed in the vineyard and mobile sinks collect the information when moving around the field. Mobile sinks could be PDA or other mobile devices carried by the farm workers who during the production process are constantly moving along the crops.

Once the information is collected by mobile sinks, this could be forwarded to the integration point with IMS. Depending on the user role (e.g. vineyard manager, winemakers, and vineyard workers) the integration point could publish different types of information in IMS. For

example, the manager might want to know when and where the field was irrigated by receiving information every 6 hours but for a worker it will be more important to request the moisture level of a specific part through a SMS message in real-time.

Furthermore, [16] has proposed several proactive actions and scenarios that could be implemented in the future in IMS. Wireless Sensor Actuator Networks (WSANs) for instance, could optimize some processes. WSAN are sensor networks that integrate a new entity called actors that can perform appropriate actions upon the environment called actuation tasks. In our scenario, for example, the irrigation process could be improved by specifying the amount of water and the specific parts where it is needed.

Based on the agricultural scenario some minimum requirements are defined to fulfill the benefits of the integration between WSN and IMS:

- *Independence.* The integration should be as independent as possible from both the WSN and IMS. This ensures minimal changes to the WSN (including mobile sinks) and IMS and in the integration point if different WSN solutions want to be adapted.
- *All types of information related to all types of entities sensed by a WSN should be made available to IMS.* As mentioned before WSNs sense different types of information (spatial, physiological and environmental) that up to this point are not considered in IMS. This information is also related to different types of entities (i.e. a place, a user, an object).
- *Translation* from the information model used in the sensor network to the information model used in IMS and vice versa.
- *Publication* of the information into IMS. Information could be pushed or pulled in the gateway, meaning that applications in IMS could ask for specific information when needed and information can also be pushed into IMS if it is determined by the integration point once analyzed.
- *Storage and processing capabilities* to handle the requests from IMS as a way to support both synchronous and asynchronous mode to access the information.

- *Consider traditional and P2P IMS.* As mentioned above, the architecture should be able to connect with the traditional architecture and the new P2P proposals for IMS.
- *Scalable and fault-tolerance.* Allowing nodes to join and leave the networks (mobile WSN and IMS).
- *Support lightweight communication mechanisms* to be able to connect with capacity- and capability-constrained mobile sinks (i.e. mobile phones and PDAs).

## 2.5 Integrating WSNs with Internet and other specific solutions

While wireless sensor networks per se are useful to collect information from the environment, they present even more advantages if available to users in other networks. The trend now for WSNs is towards allowing users in different domains or technologies to access the data collected.

An often used solution is the integration by using Web Services. As defined by W3C, a web service is “a software system designed to support interoperable machine-to-machine interaction over a network”. The advantages behind web services is that it has a high level of abstraction, it can allow synchronous and asynchronous data access, is language-independent and allows services to be described, published, located and invoked on the network.

Wang et al. proposed a gateway between WSN and the Internet using web services in [20]. The gateway assigns a unique IP address to each node, thus providing direct access via the web service. Their system architecture is based in ZigBee standard. The ZigBee coordinator is connected to the gateway which is in charge of forwarding, processing and providing data obtained from the network to clients on IP networks, for instance, a client connected to Internet. Each sensor node will be identified with an IP, and for this reason the gateway maintains a virtual web server for each one and generates the web application code for providing the service. Clients can also ask for specific information which can be stored in the gateway’s database or can be retrieved from the WSN.

Another web service approach is proposed in [21], where six web services are placed at a sensor network gateway to expose its services. They based their architecture in the I-centric

paradigm that puts the individual needs, preferences and environment parameters in the communication system. The six web services defined provide spatial and environmental information. Among the information that could be provided they propose: location information for an entity (e.g. laptop or PDA), sensed information for a specific phenomenon, entities within a specified range from a given entity, entities present in an area, the velocity magnitude and direction from an entity and finally, sensed information around an entity.

When comparing these solutions with the requirements defined in the previous section, and having clear that it does not integrate with IMS but with the Internet, we could see that it does not accomplish several of them. Wang et al. [20] do a basic integration, since each node is seen as an entity and in real world the information could be more useful when seen as a whole. They both provide storage and processing of information, furthermore, authors in [21] offer translation and publication services. However, they suffer from performance problems due to SOAP processing and XML encoding, not supporting light-weight communication mechanisms. For this same reason scalability and fault-tolerance are not accomplished, specifically due to the overhead caused on the network and on the centralized gateway.

For agricultural applications other specific architectural solutions have been studied [18, 22]. In [18], authors describe a case study for agricultural monitoring where data coming from the environment (e.g. air temperature and soil moisture) is gathered by a GPRS gateway board that forwards it to a remote server using TCP/IP protocol. It provides storage and processing of information. Data translation is not necessary since the end-to-end system is provided in the remote server using the same information collected from WSN. This solution also considers fault-tolerance by using a recovery strategy algorithms and scalability from the WSN point of view. However it is software- and hardware- dependent and information is not available for publication or integration with other systems. It can only be requested but not pushed to users in case something important arrives.

Authors in [22] proposed an architecture where sensors are deployed to monitor a vineyard. The information is collected by a centralized unit called a “reader node” which is a sensor node connected to a gateway PC. The information can be monitored locally and remotely. Locally, the solution provides an interface that allows retrieving information from the network or from a database. Remotely an email is sent periodically to the user with the readings. Storage,



processing and publication of information are provided. The drawback of this architecture is that it is also solution-dependent and integrations with other systems are not foreseen. Fault-tolerance and scalability considerations are not explained, thus we assume they have not been worked on. Furthermore, the solution is tightly coupled with specific WSN nodes and protocols.

## **2.6 Integrating WSNs and IMS Architecture**

IMS is a platform that has been selected as a key component of third generation networks. It is foreseen as a main component in service integration and provisioning. Moreover, it provides unified access to information, whose suppliers could include other networks like WSN.

A scalable framework on top of IMS [23] could facilitate the enhancement and development of more intelligent, invisible and autonomous applications that exploit WSN information (e.g. about people, places and objects). Strohbach et al. expose some of the benefits in building such a framework in IMS: message routing across administrative domains, access to heterogeneous networks, reusable core functionalities, extensive set of services and extended and flexible session control. Furthermore, actual services such as the Presence Service can be extended to exploit WSN information.

Previously we described Presence Service. Under the basics of this framework, exploiting WSN information by extending and using it as an entry point in IMS becomes logical. Firstly, Presence already manages some rudimentary context information such as location and status and the PIDF defined is highly extensible to support other types of information. Furthermore, it allows not only users but also services (e.g. applications or other networks) to have access to sensor's information in real-time with a pre-established QoS. Thirdly and finally, already existing services like SMS, MMS and IM can be used as they exist today and future integrations (e.g. with Actuator Networks) can be foreseen.

### **Integrating WSN and IMS**

Integrating WSN and IMS is recently been advocated since its importance is been widely acknowledge as shown in the previous subsection.

Gluhak and Schott in [5] present the e-SENSE architecture that integrates wireless sensor networks and IMS. The purpose of this architecture is to enable the delivery of context

information from the sensors to the user in different application environments offering significant advances compared to ZigBee based systems.

Since our focus is on the integration we will not explain the protocol stack of the e-SENSE system but in the integration done of this system into IMS. A new service called e-SENSE Service Enabler (SE) is added in IMS. This entity is in charge of providing sensor-based context information from several e-SENSE systems, thus processing and storing information. The entry point to integrate an e-SENSE system with the e-SENSE SE is the e-SENSE Gateway. This functional entity registers to IMS, buffers, schedules and prioritizes the data and enables the publish/subscribe services. However, when analyzed in the light of the requirements the solution is not suitable. Firstly, the proposed solution introduces a new entity in IMS that does not assure independence. Moreover, storage and processing are supported by this new entity and not by the gateway. Additionally, the adaptation of the information received by WSN into IMS is not explained. Scalability and fault tolerance are not considered and, since the gateway is a centralized unit without load-balancing or recovery mechanisms, we could assume that it is not supported. Neither are lightweight communications, the architecture is to be deployed in servers with enough capacity to support all the operations and services previously shown.

Another work was exposed in [4] to integrate Personal Networks (PNs) and WSNs into IMS. Their objective is to enable the future creation of smart environments where different types of devices provide different services. The interconnection is done with a key component called the PN Gateway. It uses TinySIP since it is meant to work in any hardware with limited resources; TinySIP enables communication between a client on a traditional network and a sensor node. The PN Gateway has access to different technologies e.g. WLAN, Bluetooth and ZigBee. The gateway translates messages between TinySIP and SIP and it is directly connected to a SmartDust Enabler which routes SIP messages between IMS user agents and the different PNs. Although this proposal offers integration between WSN and IMS, our requirements are not fully satisfied. It assures independency since minimal changes have to be done to assure that the WSN will locate the PN Gateway, it supports partial scalability, considers light weight communication mechanisms via TinySIP and somehow allows the publication of information. However, this architecture offers neither storage nor processing nor translation. It does not show how the different types of information and entities could be identified in IMS. Furthermore, this solution does not implement recovery mechanisms in case the PN gateway fails or is overloaded.

El Barachi et al. expose in [3] another architecture for the integration between the IMS and WSN. The integration is made by defining a generic gateway that offers several information management functions to acquire, process, store and disseminate information and other support functions to enable a real internetworking. The gateway integrates both architectures by connecting with the WSN sink and acting in IMS as a Presence External Agent (PEA). The latter will actually publish the information captured by WSNs and send it to an extended Presence Server (capable of managing types of data and entities from WSN). The information is published using one of three methods defined: interval-based publication on regular-time intervals (e.g. every  $x$  seconds), event-based (certain events are detected) or trigger-based (information is published upon the direct request from the PS). Once information reaches the gateway it is processed and then stored. An event monitor is constantly analyzing the stored information to determine if it needs to be publish, additionally a trigger handler module is sensing to see if there is a request coming from IMS; if any of the above happens a publisher module formats the document into an extended-PIDF and send a SIP PUBLISH with the information.

In the light of our requirements the WSN/IMS gateway meets most of them. First of all it is independent. The changes made to the PS are generic and are made to support the different types of data (i.e. spatial, physiological and environmental) and the types of entities (i.e. a user, a place and an object) that exist in WSN. Additionally, it stores, process, translates and publishes information.

However, it does not scale and it is not fault tolerant because it is centralized and no additional procedures have been defined for this purpose. It does not consider P2P IMS -- the design will depend on one single node to pass the information to the P2P network, which could easily become a bottleneck.

Although this last proposal is not completely suitable some of the work that has been used is suitable to accomplish our objectives and will be considered further in the design process. Specifically, they propose and demonstrate how the Presence Service could be used as entry point for WSNs. Moreover, they already extended the standard XML-based PIDF to support WSN types of information and to allow the distinction between user and non-user related information.

## 2.7 Architectures for system integration

Here we have shown how integrations between WSN and other networks have been done and they all considered a gateway to interconnect the networks. However this is not the only way to do it. In this section we will show the available methods to interconnect WSN with other technologies and will select the most suitable for our project.

There are mainly three methods: Delay Tolerant Networking (DTN) overlay, TCP/IP and gateway or proxy architecture for WSN that have been studied and compared in [24]. These approaches will be explained in further detail below.

DTN is intended for environments with long and variable delays and potentially high-bit error rates. It is based in an overlay architecture where messages called *bundles* are transmitted containing user data and relevant meta-data. The network is composed by a set of regions that share a layer called *bundle layer* which manages the messages according to the network conditions, it stores if no link is available, fragments if necessary and could even implement end-to-end reliability. A DTN gateway node is used to connect several regions and exchange *bundles* between them.

When considering sensor networks, DTN are suitable since this networks can be frequently partitioned when nodes go in sleep mode or due to node failure and packet loss rates can be very high. Two regions are defined for the interconnection, one is the TCP/IP network and the other one the sensor network. A DTN gateway is put in the middle to connect both regions. It acts as a relay proxy, meaning that its function is to forward messages between regions.

This architecture is also useful since it could be extensible enough to allow for instance the division of the sensor network in several DTN regions according to the user needs.

Another option is to use TCP/IP protocol stack directly in the sensor network. In this case no special intermediary nodes will be needed since the sensor will be connected in the TCP/IP network. Not only have it been proved that sensor nodes can run TCP/IP stack but also a small uIP TCP/IP implementation has been implemented [24].

This type of integration is probably the easiest way to interconnect the networks; however it is not a complete solution and needs to be integrated into a larger architecture.

Finally, the proxy or gateway is a very simple and straightforward form of interconnection. All interactions between clients in TCP/IP network and the sensor nodes are done through the proxy. It can act as a relay or as a front-end. In the first case the gateway simply forwards information to the client according to a data interest previously expressed. In the second one the proxy acts in a proactive way, been able to collect data, process it and even store it in a database.

Although this approach decouples both networks, the gateway is often a dependent solution that integrates specific set of protocols or work for some specific scenarios. In a centralized architecture, the gateway also becomes a single point of failure which could cut communication between both networks.

Now that we have described the possible methods to interconnect both technologies, we could say that the proxy is a suitable solution for our requirements but it needs to overcome the fault-tolerance and scalability issues. DTN overlay networks offers a solution for fault-tolerance in the network by introducing the concept of overlay; a logical layer on top of the protocol stack. Combining these two approaches could be a solution for our integration. In fact, an overlay architecture using Peer-To-Peer systems can be analyzed.

A gateway made up by peers seems like a possible solution that will help integrate mobile sink WSNs into traditional and P2P IMS.

## Chapter 3 PROPOSED ARCHITECTURE

This chapter presents the architecture designed to integrate mobile sink WSN with IMS. Firstly, we will explain the design and basis of the proposal, this part includes assumptions, principles, elements and topology of the architecture. Later on, the rules governing our overlay will be described; this includes the information model, the protocol and the procedures. Finally, we explore a scenario using the proposed architecture.

### 3.1 Architecture design

The goal of the design is to accomplish all the requirements exposed in chapter II. We have shown that a gateway is a simple and straightforward solution to interconnect two technologies. Furthermore, we believe that an overlay architecture using P2P systems as a principle can satisfy our requirements.

An overlay gateway inherits the main characteristics of P2P systems: *scalability*, *fault-tolerance* and resilience; which are part of the required features. Moreover, *storage* capabilities have already been proven to be feasible in P2P architectures.

It can help us satisfied the proposed requirements and is also a solution that can be implemented with the existing infrastructure; by using as peers the same mobile phones that act as sinks in WSN and as user agents in IMS. Moreover, in this proposal *light-weight communication mechanisms* become implicit to enable the functioning of the gateway.

Additionally to the overlay system, the architecture design will include *translation* and *processing* mechanisms as well as enable *all types of entities and information*. This can be executed by individual peers that offer services through a flexible information model and, also, by using the extended-PIDF document proposed by El Barachi et al. in [3]. This extended-PIDF, as presented in chapter II, was defined to overcome the limitations of PIDF to inform about non-user entities and enables the transmission of the different types of information present in WSN (i.e. spatial, physiological and environmental).

We propose an overlay gateway to fulfill the defined needs and we design the solution from this decision. Once the architecture is presented a detail analysis on how the requirements are met by the proposal will be done in chapter IV.

### 3.1.1 Assumptions

Our idea is to provide a gateway that will interconnect WSN with IMS; however there are some assumptions that need to be done at this point as a way to present achievable goals at this stage in the design process.

Concerning the entire system, we assume that either all networks are managed by the same operator or that there is a business agreement between them enabling direct communication. Furthermore, we assume that the WSN, the overlay gateway and IMS are statically configured to know each other. Additionally, we assume that all mobile devices in the system can communicate with each other to enable building an overlay on top of it.

Analyzing the mobile WSN we assume that the WSN sends a specific message format that needs to be translated to the extended-PIDF. However, during the design of the architecture we try to ensure that minimal adjustments will have to be made in case the format changes. Moreover, to read and collect this information, we assume that the mobile sinks send the information received from the sensors to a common interface or port in the mobile device.

### 3.1.2 Architecture principles

The principles to our architecture are based in the gateway which is the central point of the integration. Figure 3.1.1 depicts the general architecture, including the overlay gateway which is divided in five groups. Each one of these groups is in charge of providing one or more services and has one leader peer.

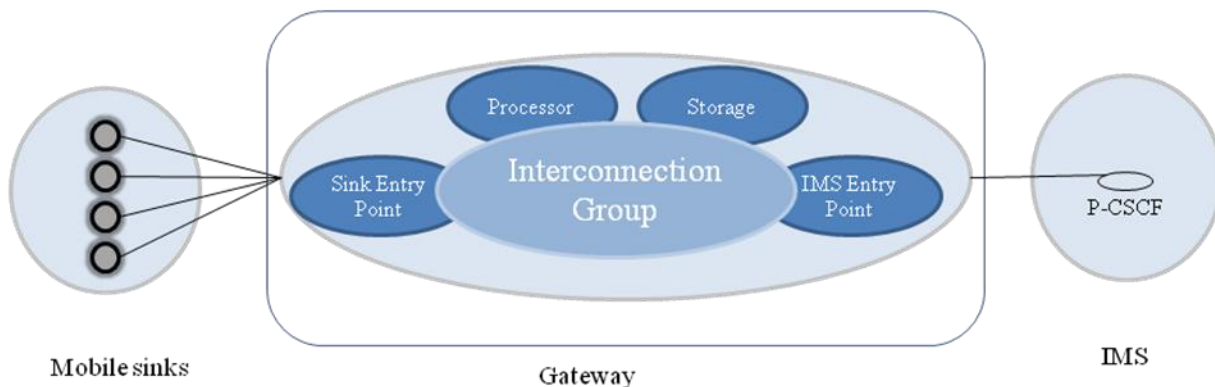


Figure 3.1.1. General architecture

The first group is in charge of the interactions with the IMS Presence Service, meaning that it manages all integration with this technology. This group is the entry point to IMS; it is in charge of connecting with SIP Servers (CSCFs) in IMS, to translate and publish the information to the Presence Server and to process request from IMS towards the mobile sink WSN.

The second group is similar to the previous one but offers connectivity to the sensor network. It interacts and connects the mobile sinks with the gateway. Furthermore, it receives the information sent by the mobile sink WSN, translates it to the information model used in the gateway and it is able to send requests to the sensor network when requests for specific information arrived.

The third and fourth groups deal handle data management process. They are in charge of storage and processing respectively. Storing information allows users to have a history and to get information without having to send the request to the sensor network. Processing of information is done through predefined parameters; it consists for example in the aggregation of the data collected from sensors in a specific area.

Finally, there is one last group whose role is to interconnect services and peers in the overlay gateway. It manages interactions between the other groups mentioned, allowing communication between them. This group is formed by the leader peers from each group.

### 3.1.3 Peer roles, interactions and topology

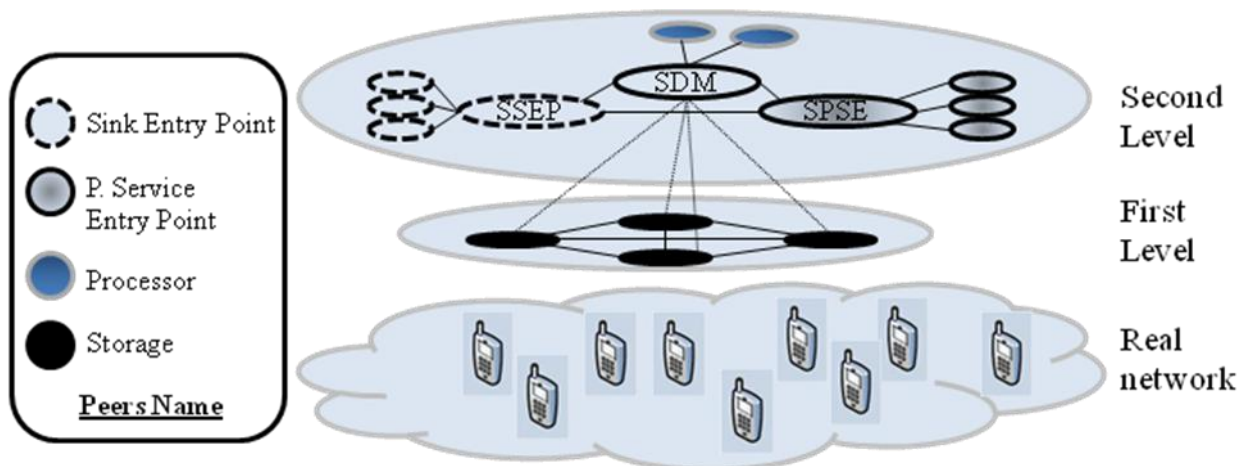


Figure 3.1.2 Overlay Gateway



Figure 3.1.2 shows the topology that has been designed based on the key principles of our architecture. It is designed as a two-tier topology where the first level is exclusively for repository nodes (i.e. storage); which is done in a distributed and collaborative way among the peers participating in the group. The second level provides all other services. Each of the five groups defined in the previous section has two or more participant roles interacting to accomplish their functions; we will explain this in more detail in the following subsections. Once the roles and the interactions are described, the topology will be further analyzed and presented.

## Roles

When describing the principles of our architecture we described the five groups that compose it. Each one has participating roles that enable the services as was previously defined. A role is an identification that determines the services and the logic provided by each peer in the overlay.

The first group is in charge of providing an integration point with IMS and on translating the extended-PIDF format used by the Presence Service. This group has two participant roles: the *Presence Service Entry Point (PSE)* and the *Super Presence Service Entry Point (SPSE)*. The *PSE* acts strictly as an entry point, it provides connection to IMS CSCF's servers, enabling communication between IMS and the gateway. Inside the overlay, this role interacts exclusively with the *SPSE*. The *SPSE* is the leader of the group, and therefore, there is only one in a given time but any *PSE* could play the role. This peer manages the other nodes in the group; but it also has the ability to provide the connection with IMS. Additionally, it translates information to the extended PIDF format, sends request and responses to and from IMS and other groups in the gateway.

The second group is the one in charge of the connection with the sensor network; made exclusively through the mobile sinks. This group includes the *Sink Entry Point (SEP)* and the *Super Sink Entry Point (SSEP)*. *SEP* allows connectivity with the mobile sinks; enabling the transference of messages between networks. The *SSEP* is the leader peer of this group. It is in charge of *SEP* nodes, interprets and analyzes information received from the WSN and is capable to determine whether this data should be stored, processed or sent directly to IMS. This node also understands the XML format used by the WSN and translates in both directions. As the *SPSE*,

this leader also has the ability to play the basic role, connecting the mobile sink with the gateway. Although any *SEP* could play the role, at a given time only one can do it.

The third and fourth groups, as defined in the architecture principles, handle storage and processing and even if they are different groups, they share the same leader. The storage group consists of the *Storage* and the *Super Data Management (SDM)* roles. *Storage* peers are in the first level and handle requests to store and retrieve information. Its leader (i.e. *SDM*) is in the second level and manages the group and the interactions between its groups and the rest of groups in the gateway.

The processing group is composed by the *Processing* role that can transform, aggregate or compress information and the *SDM*; as mentioned before, the *SDM* is also the leader.

The final group is composed by the leaders (i.e. *SPSE*, *SSEP* and *SDM*) from the four groups previously shown. They collaborate with each other to offer their services to the overlay and allow access to other groups.

## **Interactions**

In this part we will show how interactions are made inside group members. The communication and message exchange depends in the source and the destination of the messages.

The first group is IMS Entry Point that offers connection with IMS. As explained before, the *SPSE* connects all *PSEs* to the gateway. When a message is received by the *PSE* from IMS, it is forwarded by the peer to the *SPSE*. Equally, when a message is received from another peer in the leader's group, the *SPSE* sends the message to one of the *PSEs* so it can be forwarded to IMS.

The second one is the Sink Entry Point group that connects with the mobile sink WSN. The *SSEP* receives data from and sends requests to the WSN through the *SEPs*. Each *SEP* is connected to one or several mobile sinks. Once information is collected from the mobile sink WSN, the *SEP* receiving the information, forwards it to the *SSEP*. It decides where the data should go (whether to store, to process or to publish), thus it sends a request to one of the other peers in the leader group. Additionally, when a request is sent to the group to retrieve information from the WSN, the *SSEP* will forward the request to all *SEPs*.

The third and fourth groups are related to storage and processing. Firstly, the *SDM* receives requests from the *SSEP* and the *SPSE* to store or retrieve information respectively. This

request will be resend to the *Storage* peers in the first level of the architecture which will later send a response with the information to the *SDM*. Secondly, if processing of data is demanded, the *SDM* forwards the request to one of the *Processor* peers that will perform the action and return the transformed data.

Finally, interactions inside the Interconnection Group are explained. We mentioned before that the *SSEP* decides whether the information should be stored, processed or published. If storage or processing needs to be done, then the *SSEP* sends a message to the *SDM* and if it needs to be published then it is routed to the *SPSE*.

The *SPSE* handles publication towards the IMS Entry Point group, thus it receives the requests and disseminate it according to the definitions inside the group. When information is requested from IMS and reaches the *SPSE*, it forwards the request to the *SDM*.

Finally, there is the *SDM* which responds to requester nodes asking to store, to process or to retrieve any information. Storage and process are handled inside their respective groups; however retrieval of information is first analyzed by the peer. If history data is required or if the time of the stored data is below the demanded time threshold, information is retrieved from the Storage group; otherwise the *SDM* sends the request to the *SSEP* so the retrieval is made directly from the mobile sink WSN.

## **Topology**

Before connecting the gateway with the two networks (i.e. mobile sinks and IMS CSCF's), the topology of the overlay should be presented.

The topology is depicted in figure 3.1.2. The first level is exclusively *Storage* nodes which are the immediate and first option of data retrieving for our architecture. They are fully-meshed to accommodate and distribute the information keeping in mind the possibility of node failures or disconnections; additionally this topology allows a more effective response to queries. This decision makes the topology less scalable; however it is just a design option that could change as the architecture evolves. This first level is connected with the second level through the *SDM* which, as mentioned above, handles the *Storage* peers.

The second level includes the leaders group connected in a fully-meshed network. This structure allows easily exchange of information about the groups and the nodes that each one of

them handles. This second level also contains the *SEPs*, the *PSEs* and the *Processors*; each connected to its leader in a star-like topology.

## Global architecture

Now that the gateway architecture, its roles and its topology have been explained, we will show how we connect integrate the overlay with the mobile sink WSN and with IMS. At this point, the topology and the basics of the integration will be described, the process on how they connect with each other is explained below in the next subsection. Figure 3.1.3 shows the gateway connected with the mobile sinks and in this case using the traditional IMS architecture.

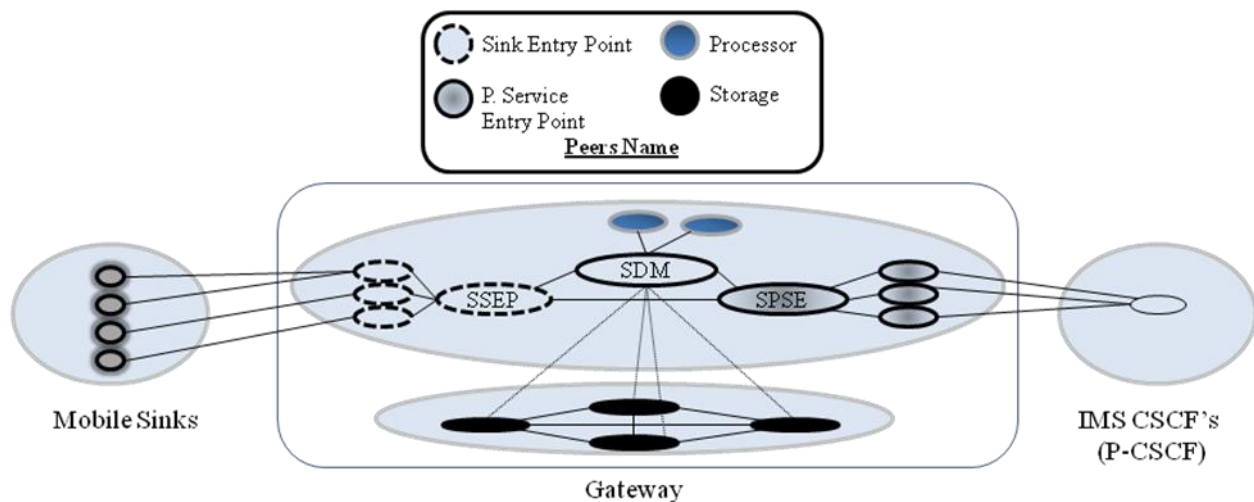


Figure 3.1.3. Global topology

To enable the connectivity between the WSN and the gateway a minimal piece of software will be installed in the mobile sinks. They connect to the gateway by using the *SEPs*. Each mobile sink use one and only one interface to connect to the gateway but each *SEP* can handle one or more mobile sinks.

We proposed that the mobile WSN be considered as a provider in IMS. It has one identifier (e.g. sip:gatewayWSN@domain.com) and depending on the type of information collected by WSN (i.e. spatial, physiological or environmental) one or several “*presentities*” are created. As will be shown in the extended-PIDF, a *presentity* is determined by the type of information that collects, thus, implying that in case the mobile WSN has more than one type of data it should create a *presentity* for each one of them.

Considering this precept each *PSE* acts as a Presence User Agent (PUA) in IMS; meaning that they are considered as the devices that collect pieces of information from the *presentities*. The *PSE* connects in IMS through the SIP Servers that will then forward the PUBLISH messages to the Presence Server (PS). Thus the PS in IMS receives information from one or several PUAs.

## 3.2 Overlay rules

This section presents the rules governing the architecture described in the previous section. This includes the information model used inside the gateway, the protocol used by nodes to communicate to each other and the description to the different procedures supported by the overlay.

### 3.2.1 Protocol

The overlay protocol allows peers in the network to communicate with each other in a standard language. A protocol defines the rules for the syntax, semantics and synchronization in the network. It not only defines the messages that can be exchange between nodes but also their structure and the rules for exchanging them (e.g. the answer expected).

To become a suitable protocol for the overlay, it should accomplish the requirements mentioned in the second chapter, as well as support the services and characteristics inherit in any P2P system. Basically the protocol needs to:

- Enable self-organization and self-recovery. This means that it should allow nodes to join as well as to create groups. Additionally it should enable a mechanism to leave the network and to select a new leader for a group.
- Allow information storage, processing, retrieval and publication.
- Be as simple as possible. The protocol should be simple and low consuming since it is intended to be installed in mobile devices with limited capacity.
- Enable scalability. It should enable more nodes to join the network without overloading or causing denial of service.
- If possible, be standard. Even more, available standard protocols should be analyzed to determine whether or not they can be reused.

As the overlay protocol for the gateway we selected the Session Initiation Protocol (SIP) since it meets our requirements. Firstly, SIP is a standard protocol, widely deployed and easily extensible. It has already been implemented to work in devices with limited capacity and its structure is simple.

Secondly, the different procedures are enabled by some of the existing extensions. Although they will be described in detail in the following section, here we present a brief introduction. The SUBSCRIBE / NOTIFY methods included in the RFC3265 provides an option for asynchronous notification of events. The event-notification framework is proposed to enable entities in the network to subscribe to receive the current state or updates from a remote node. If the state changes in the remote node, NOTIFY messages are used to inform it to subscribers. The Presence Framework, for instance, uses this extension to control the watchers registered to each “*presentity*”. This event-notification framework enables the creation of groups inside the gateway where information needs to be disseminated from the super peer to one or several participants in an asynchronous way.

Additionally, the INFO method described RFC2976 enables the exchange of application level information. Its goal is not to change the state or parameters of the session but to send information required by applications. The advantage behind this extension is that it allows the application to determine the message body.

Furthermore, self-organization is enabled by SIP multicast addressing, since in the beginning the nodes do not know who their super peer is, they send a message to a multicast address that will allow its location.

### **3.2.2 Procedures**

In the previous section we described the protocol that rules the exchange of messages among the gateway, in this part we show how those messages are used by the different entities in the overlay to enable the effective functioning of it. We described each of the main procedures; the roles involved in the process and the message exchanges between them.

## Self-Organization

Self-organization is mandatory since we are considering an ad-hoc network built from mobile phones that do not know each other at the creation of the network. This process consists then in building the network when no node is in the overlay.

The mechanism defined for discovering other peers is SIP multicast addressing. A multicast address is assigned to all leader peers and known by all nodes in the network. We assume that each peer knows the roles it can play in the overlay and thus, the groups it needs to connect to. The joining process is depicted in Figure 3.2.1 and explained here below.

The gateway should be functional when at least one mobile sink is available, so we decided that the first node to join the network is the *SSEP*. Once it starts to play its role, it creates two groups, the first one for the leader peers, meaning that it enables the reception through the multicast address. And the second one for the sink entry points, offering connectivity to the mobile sinks.

When another node joins the network, it determines the roles it can play and the groups that it should join. Then, it sends a SIP INFO message to the multicast address requesting the address of the peer in charge of its groups (the selected leader). The answer is sent by any of the three leader peers in another SIP INFO message indicating whether there is a leader peer (sending its address) or not.

If there is no leader yet, the node becomes the leader peer for the group. This implies that the multicast address is assigned to it and that the reception of SIP SUBSCRIBE messages from peers belonging to its groups is enabled.

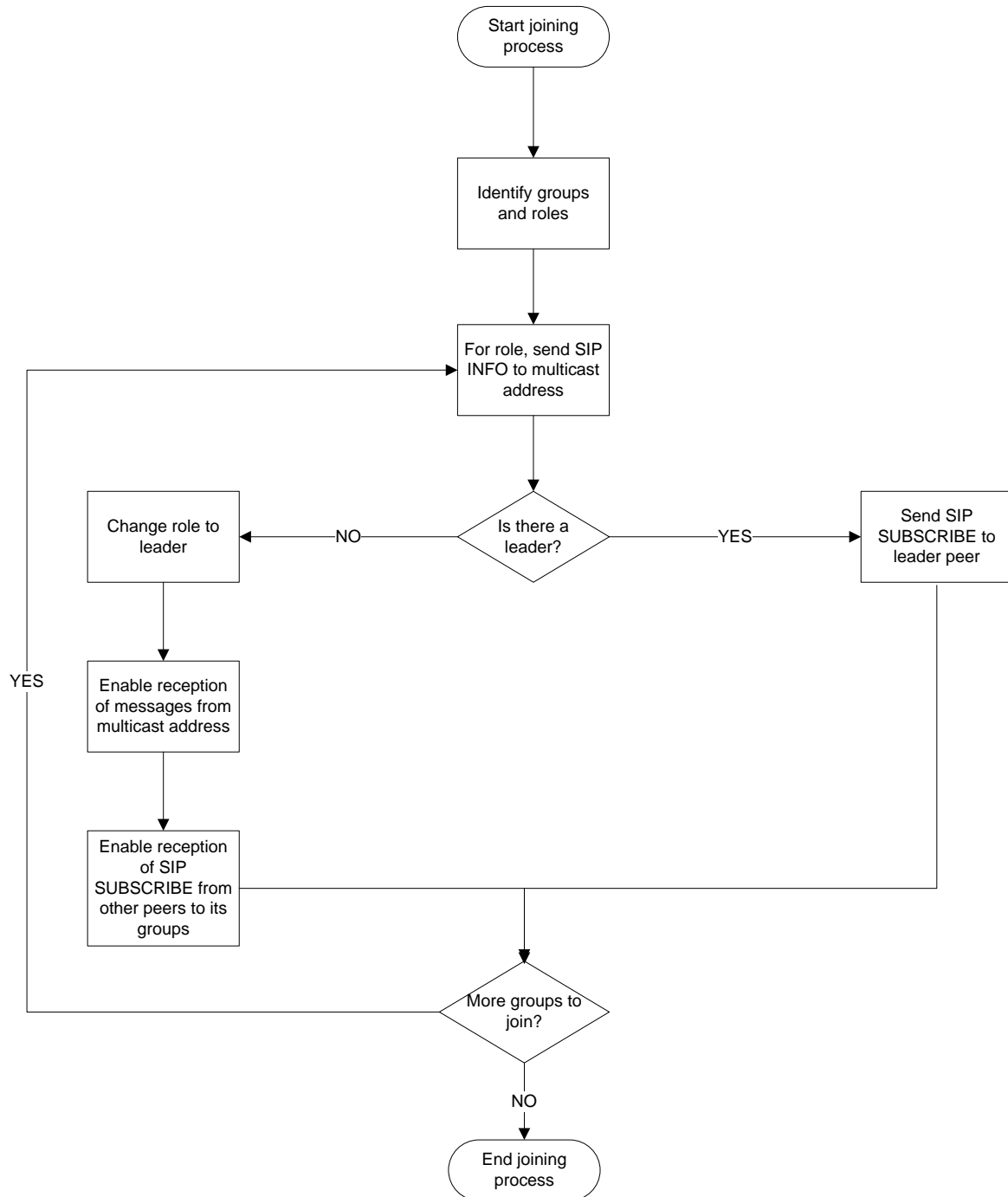


Figure 3.2.1. Flow diagram for a node joining the overlay

For example, if a node joins the network and identifies that it should play the role of *Storage*. The node takes for each role the groups to which it should be part of. First for the *Storage* role, it sends a multicast SIP INFO requesting the address of the *SDM* that handles the storage group; if there is no leader then this node changes its role and becomes the *SDM*. Once this happens, the node enables the reception of SIP INFO messages through the multicast address



and of SIP SUBSCRIBE request from not only the *Storage* nodes but also the *Processor*. Otherwise, if the leader peer has already been selected, it sends a SIP SUBSCRIBE to it.

For self-recovery purposes we decided that all peers in a group should be aware of the group members. So, each time a group changes (i.e. a peer joins or leaves the group) the leader sends a SIP INFO message with “groupInfo” action and the list of all members to all peers registered in the group.

In some roles (e.g. *SEP* and *PSE*) information is received by the simple peers and they need to communicate it to the leader. For these cases, once a node joins the group, the leader peer sends also a SIP SUBSCRIBE request to the peer which allows that each time the peer gets information it forwards it with a SIP NOTIFY message.

Once the gateway is set up (there is at least one peer playing each role), the *SPSE* registers to IMS by sending a SIP REGISTER. The *SEP* and *PSE* roles can be played by the leader peers, thus implying that the gateway could be considered as set up even with the two leader peers and no other nodes in these two groups.

### **Self-Recovery**

There are two possible scenarios where self-recovery is required for P2P systems. The first one is when a node leaves voluntarily and the other one when it is unexpected. At this point in the project we will only consider voluntarily departure from the overlay.

Each peer is responsible for informing other members of its departure. To do this, it uses a SIP INFO message with a “leaving” action in the message body. The following paragraphs present how the self-recovery process is done depending on the role of the node leaving as depicted in figure 3.2.2.

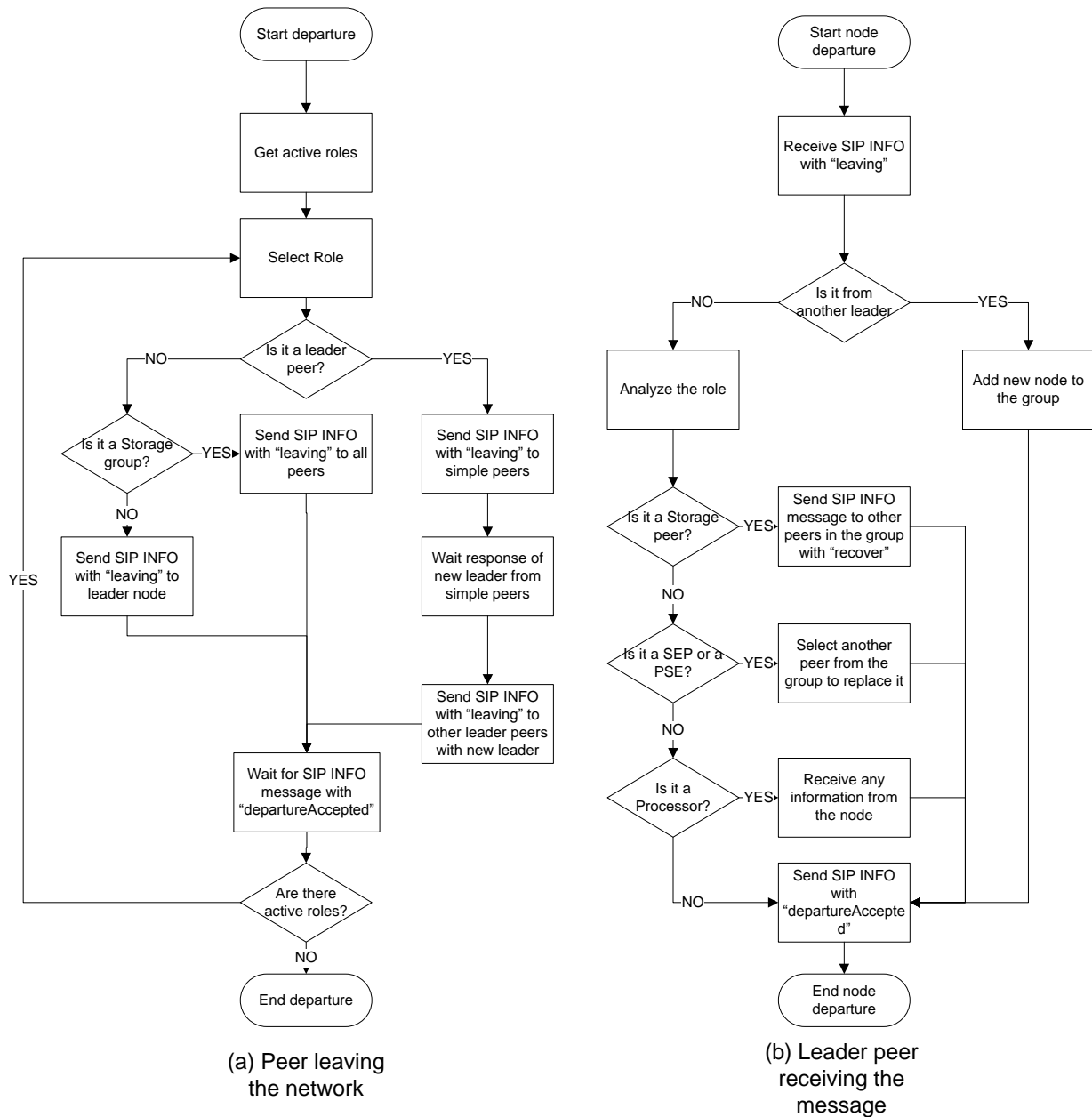


Figure 3.2.2. Flow diagram for node departure

When a peer is leaving, as shown in Figure 3.2.2 (a), the first thing it needs to identify is the roles that it is actually playing in the overlay. For each peer it should analyze if it is a simple peer or a leader peer. If it is a simple peer, it sends a SIP INFO message to its leader informing that the departure process has started. In case it is a *Storage* peer it should send the message to its leader peer as well as to other peers in the group, since they are all connected in a fully meshed way. When the peer is the leader of a group, it sends the message to all simple peers in its groups and waits for a response from one of the peers that will be elected as new leader. In the case of

the *SDM* departure, the simple peers in charge of the selection of a leader are the *Storage* group. Once it receives the message, it forwards the decision to other leader peers in the leader group. Then, the node waits to receive a SIP INFO message with “*departureAccepted*” to disable the role and does the same procedure with all roles until there is no active role. Finally it gets disconnected.

Figure 3.2.2 (b) shows the process executed in the leader peer when it receives a SIP INFO message with a “*leaving*” tag on it. If the message is from another leader peer then the new leader has already been selected and the node just updates the reference. If it comes from a simple peer, it determines the group to which the node belongs to and thus, the role it is playing. When a *Storage* peer is leaving, the leader peer sends a “*recover*” action to all peers in the group where they should execute a recovery measure to get the pieces of information stored in it. When it is one of the entry points (i.e. *SEP* or *PSE*), the leader peer selects another peer to replace the leaving node, in case of the *SEP* connect with the mobile sink. It sends a SIP INFO message with “*recover*” to the selected peer, informing the actions that should taken place and waits for the SIP INFO message “*recoverAccepted*” that means it has accepted and established the new connection. Otherwise, it is a *Processor* node in which case the leader peer only tries to get any information still in the node. Finally, once actions have been taken, the leader node sends the SIP INFO message to the peer leaving with the expected “*departureAccepted*”.

### **Mobile sink connection:**

The mobile sinks will be able to connect to the gateway through the piece of software that will be deployed in the mobile device. This API sends a SIP INFO message to the multicast address alerting the presence of a mobile sink and requesting a connection to the gateway. When the *SSEP*, as leader of the sink entry point group, receives the message it selects a *SEP* or itself (in case there are no simple peers) to start the connection process.

Three basic services are provided in the API: connect to sink, get specific information and disconnect from sink. The connection process sends a message to the mobile sink to connect with the port or interface where the information is being transmitted, this starts the communication to allow getting periodically data sent by the sensor nodes. Each time data is received, the API sends a SIP INFO message to the *SEP* with the new information. Get specific information is used when data needs to be retrieved in the WSN, thus the *SEP* connects with the API and sends the

request to the available interface. Finally when a node is disconnecting the API re-starts the connection process previously described.

### Information Management:

As explained and described in the requirements, the gateway should support different data management procedures such as Translation, Storage, Retrieval, Processing and Publication.

Translation is actually done in two parts. The first one is to and from the mobile sink WSN format and it is done by the *SSEP*. Mobile sinks send the information and receive requests in a specific XML format; as it was previously defined. The data is transmitted in the body of the SIP messages that are exchange between the *SEP* and the *SSEP*. The latter does the translation process by mapping from one information model to another.

The second translation is to and from the extended PIDF that will be used; it has been defined in [3]. The peer in charge of this part is the *SPSE*.

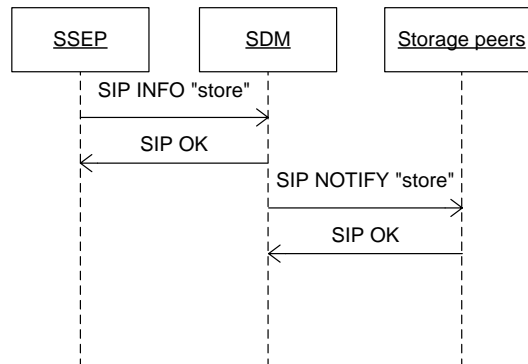


Figure 3.2.3. Information storage process

Information storage, depicted by a sequence diagram in figure 3.2.3, is basically started by the SSEP when it receives information from the mobile WSN. It sends a SIP INFO message with “store” as action to the storage group. The leader of the group (i.e. *SDM*) received the message and sends a SIP NOTIFY message with the information to the group represented by the first level in the overlay which actually stores the data.

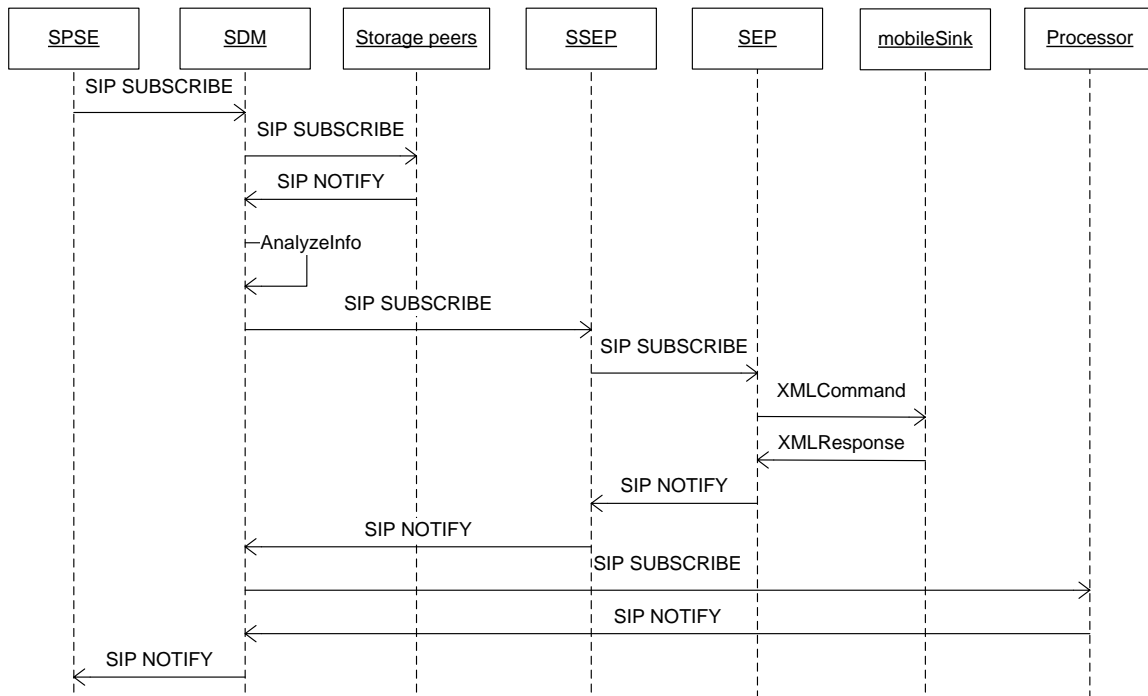


Figure 3.2.4. Information retrieval process

Information retrieval is started by the *SPSE* since information is requested from IMS as shown in figure 3.2.4. To request information it sends a SIP SUBSCRIBE with the request to the *SDM*. The *SDM* forwards the request to the storage group and receives the SIP NOTIFY with the response. It analyzes the received info and if the information is not up to date, it sends a SIP SUBSCRIBE to the *SSEP* in order to get the data from the mobile WSN directly. Once the information is returned by the mobile sinks to the *PSEs*, they send the SIP NOTIFY to the leader peer which finally sends the response to the *SDM*. The information is sent to process if needed to a *Processor* node. Finally the requester receives a SIP NOTIFY with the retrieved information.

Information processing was described as part of the retrieval information. Basically, it is the *SDM* which receives the SIP SUBSCRIBE and resends it to one of the *Processor* peers. Once the information is processed, a SIP NOTIFY message is sent with the result to the *SDM* and then to the requester.

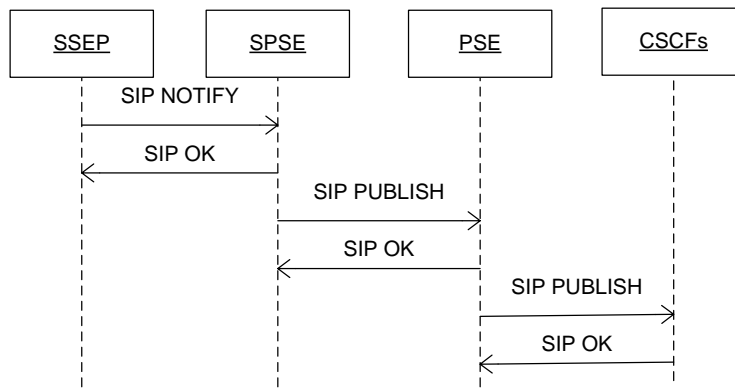


Figure 3.2.5. Information publication

Publication of information is presented in figure 3.2.5. It is mainly handled by the *SPSE* and the *PSEs*. The *SPSE* receives a SIP NOTIFY; we assume that it is already subscribed to receive notifications from the *SSEP*. Then, it forwards it to the *PSEs* which send the SIP PUBLISH request to the presence service.

### 3.2.3 Information Model

#### Gateway Information Model

Although the rules for exchanging messages have been defined, as mentioned above, the SIP message body for the INFO and the SUBSCRIBE/NOTIFY methods can be defined by each application using it. For this purpose we designed an information model that governs the semantics and syntaxes of the data transmitted through SIP messages.

Since we are connecting with two existing technologies that offer their own information models, when deciding the information model that should be used inside the gateway three options become available:

1. The information model could be the same used in the mobile sink WSN;
2. It could be the extended PIDF used by the Presence Service in IMS;
3. A new information model could be defined.

In the light of our requirements, and since we have proposed as much independence and generalization as possible, we have decided to have a unique information model inside the gateway. The information model is depicted in figure 3.2.6.

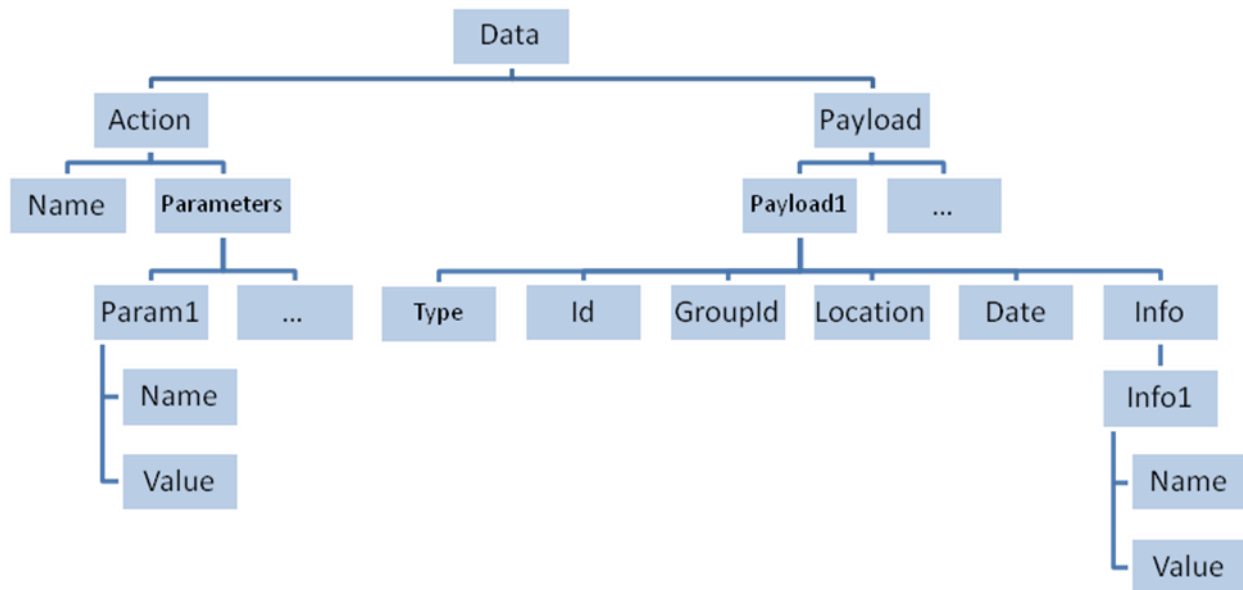


Figure 3.2.6. Gateway Information Model

It is a XML format divided into *Action* and *Payload*. *Action* defines the procedure or action that wants to be executed inside the gateway and includes any parameters required to perform the corresponding action. For example, “discover” is an action where the parameter is the name of the node that wants to be discovered.

*Payload* has the information from none, one or several entities. The entity is defined in Type; it could be a Sensor or a “PeerRole”. A “PeerRole” identifies a role that is been played by a peer, since it could play several roles at the same time and each one needs to be identified. For example when raw data is retrieve from the distributed database (Figure 3.2.7), the type of payload is sensor and for each sensor comes the Id, the group, the location, the date in which the data was received and Info refers to moisture and temperature in the case of an agricultural application but this could change to other measures depending on the target application.

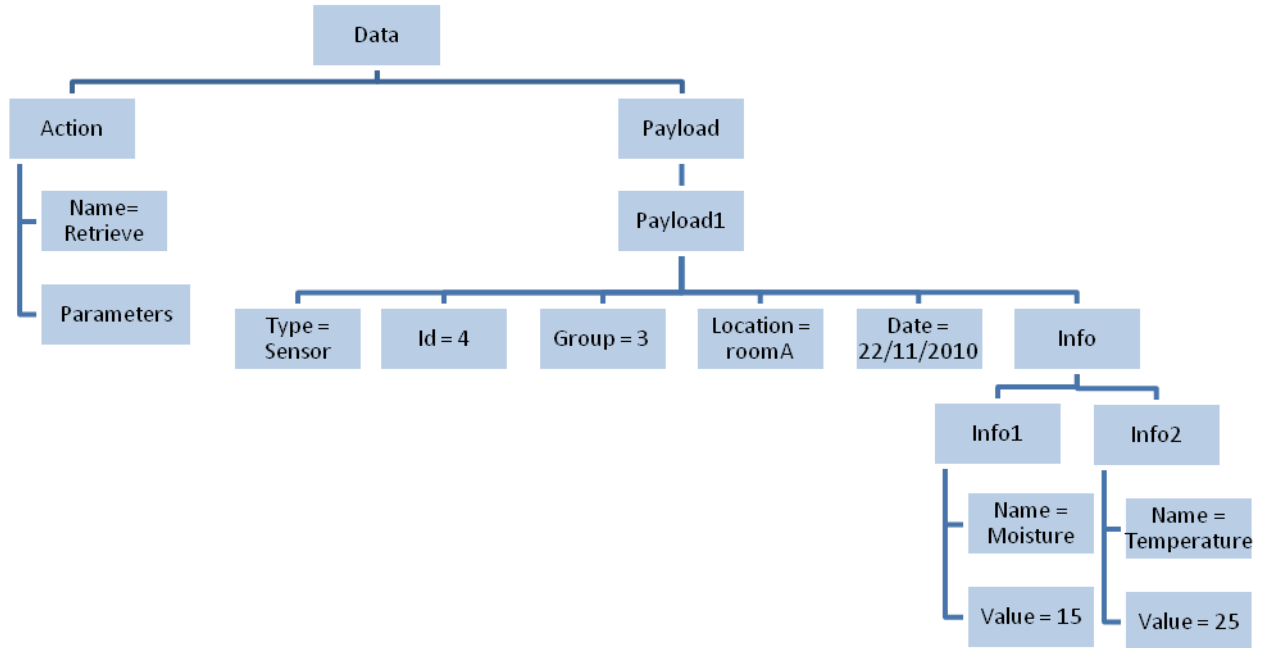


Figure 3.2.7. Information Model, Retrieval process

## Extended PIDF

The information model inside the gateway has been defined; however the extended PIDF had not yet been described and is also a key part of the integration. The basic PIDF defined in the RFC3863 includes basic status information and communication address. However the format was designed to be extensible.

We previously mentioned that we were going to use the PIDF extension proposed in [3]. Their extension is based on another proposed PIDF extension called GEOPRIV which focus is to provide location information. The geographical information describes a physical position in the world. By extending the PIDF status element, GEOPRIV includes the location information in a tuples called “geopriv”; that includes location data and usage rules.

El Barachi et al. [3] add two new attributes in the presence element to identify its type and a brief description, the attributes are “entityType” and “entityDescription” respectively.

When translating information for publication the *SPSE* should have this in consideration to map from the gateway format into the extended PIDF.



### 3.3 Scenario

So far we have shown the proposed architecture design, its procedures and the rules governing interactions and roles. This subsection will show how the architecture comes together in a specific scenario. In fact, it explores the agricultural scenario briefly described in chapter II in greater detail.

Firstly, we assume the overlay is already organized and that there is at least one peer in each role. Connections with the mobile sinks and IMS have also been established. Finally, watchers are subscribed in the PS to receive context information from the mobile WSN.

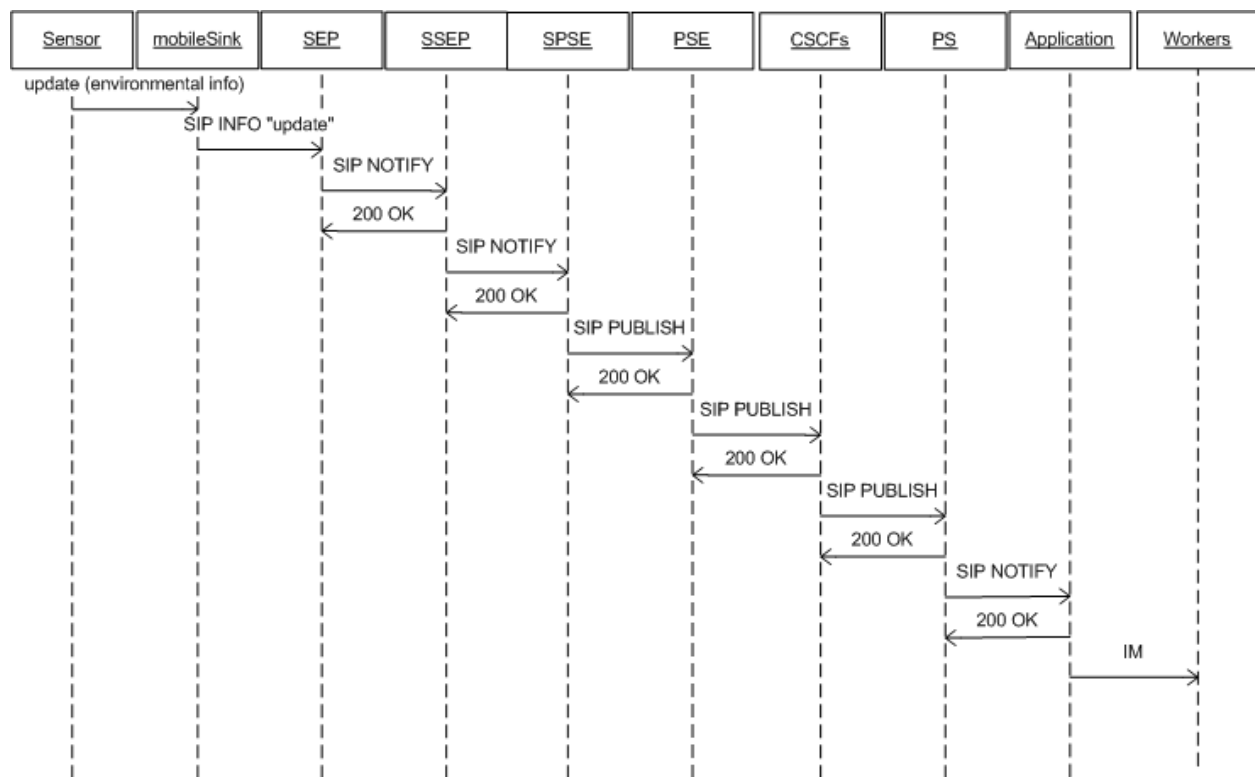


Figure 3.3.1. Sequence diagram for agricultural scenario, moisture below threshold

Figure 3.3.1 shows the exchanges between functional entities in the integration for the agricultural scenario where moisture is under a preset threshold, thus, immediate publication is required. Sensor in the mobile sink WSN are constantly sending update of information to the mobile sinks passing nearby. The mobile sink will then resend the information towards the gateway through the *SEP*. Since the leader peer has previously subscribed to the *SEP*, it receives a SIP NOTIFY message. The first operation that the *SSEP* does is to translate information from the WSN XML format to the gateway information model. And then, it analyzes the information

and in this case determines that the moisture is below the threshold so the data should be published in IMS. It sends a SIP NOTIFY to the *SPSE* which has also subscribed before.

The *SPSE* receives the information and analyze the request. When it determines that the information is to be published, the *SPSE* translates the data from the gateway information model to the corresponding extended PIDF. Once this is done, the peer sends a message to one of the *PSEs* in a SIP PUBLISH message. The *PSE* receives the message and forwards it to IMS CSCF's servers that are already configured to send the presence SIP PUBLISH message to the Presence Server (PS).

Up to this point the information has reached IMS and is available to users and other applications that are subscribed as watchers for the mobile WSN. As the example scenario shows, a SIP NOTIFY is sent from the PS to an application that sends an Instant Message (IM) to the workers.

## Chapter 4 IMPLEMENTATION AND VALIDATION

This chapter describes the process of implementation and validation of the overlay gateway depicted in the previous chapter. Firstly, as implementation we designed the software architecture for the system and developed a prototype as a proof of concept. Then, we validate the architecture from two perspectives: the first is to show how the architecture meets the requirements and compare it with the architectures presented in chapter two.

### 4.1 Implementation

At this stage of the project only a subset of the overlay was implemented. However the software architecture for the system was designed and is useful for future improvements. First we will explain and described the software architecture and then the prototype.

#### 4.1.1 Software Architecture

The software architecture was designed based on the general architecture presented in chapter three and depicted in Figure 4.1.1. To accomplish the overlay architecture here proposed as well as the independence required we proposed a general software architecture that will be deployed in each peer willing to participate.

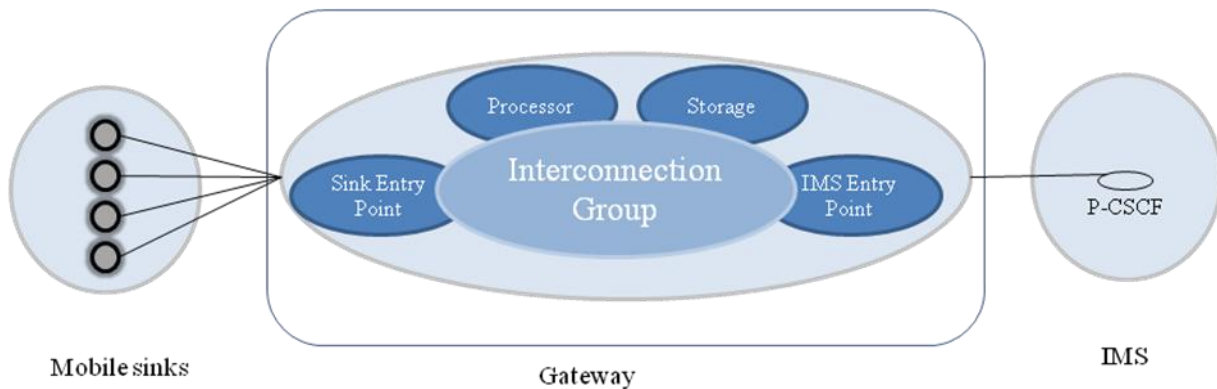


Figure 4.1.1. General architecture

We base our development in a P2P middleware. This in-between layer in P2P systems allows the application-independence of distributed resources in the overlay. They have been designed to place and locate resources in the devices and to identify them with a unique id (GUID), relieving clients of these tasks.

This software architecture is depicted in Figure 4.1.2. It has been designed, as mentioned before, for all participating peers in the overlay gateway. The basic logic is divided in several common modules such as the Overlay Manager, the Overlay Mapping, the Gateway Controller and the SIP Manager. The logic for each role is determined in the Role Logic module, which enables the different services from the Services module as defined. This decoupling allows a peer to play one or several roles in a giving time.

The mobile sink has also an API that enables communication and access with the gateway. It implements a simplified version including the basics of each module and the services are the ones that allow connecting to the sink, getting specific information and disconnecting.

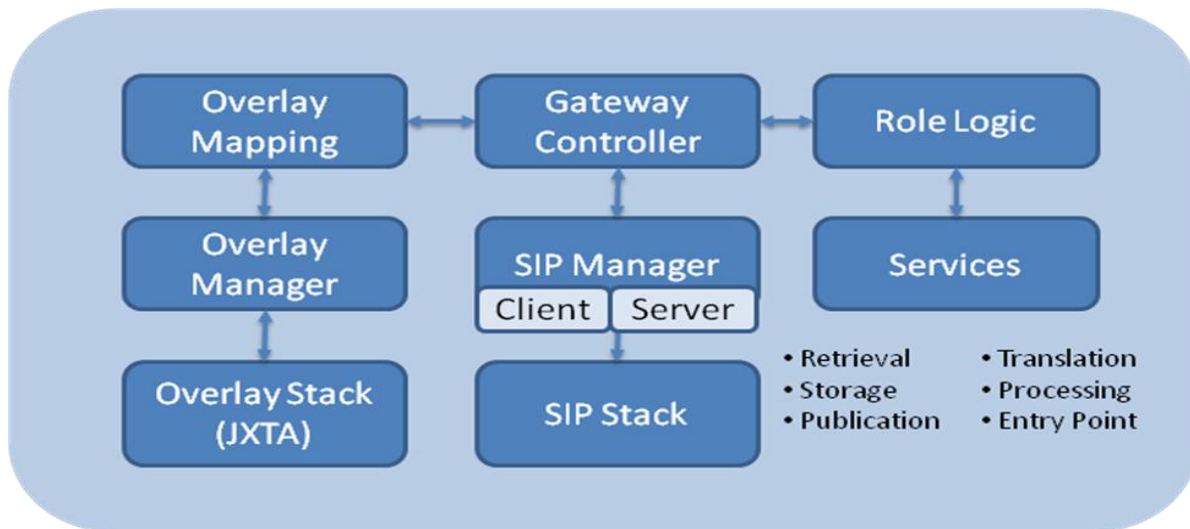


Figure 4.1.2. Peer architecture

The Overlay Manager is the module that allows nodes in the overlay to become peers and have an ID. This is done via a unique peerID given in the overlay that is independent from the IP network address. Each peerID identifies one and only one node in the overlay. Furthermore, this

module allows the discovery of other nodes in the overlay. It provides these options through the Overlay Stack module that has the actual connection to the peer overlay.

The Overlay Mapping allows the rest of the software architecture to be middleware-independent. This module talks both middleware and application primitives. Meaning that when information is sent towards the peer middleware overlay, e.g. to get an ID, the Overlay Mapping module will get a request and map it to a request that the Overlay Manager can understand. Once it receives a response the opposite mapping is done to return the answer.

In chapter three we defined that SIP was going to be the overlay protocol. This means that for P2P primitives we use the middleware and the Overlay modules, but direct communication and message exchange between peers is done by using this protocol. A peer can access SIP through the SIP Stack and the SIP Manager components.

The SIP Manager supports all procedures explained in chapter three regarding self-organization and self-recovery and the information management services (i.e Translation, Storage, Retrieval, Publication and Processing). It creates all the SIP messages required in the overlay and exposed in chapter three and sends them through the SIP Stack. The SIP Manager acts as client when sending messages and as a server when to receive them.

The Gateway Controller is the orchestrator of the application. It contains the logic to connect all main modules, providing an interface and connection point between them. It has the information regarding the roles the peer is playing, the groups it has joined and the messages it has sent. The Gateway Controller is completely independent from the P2P middleware and from the SIP implementation. Information is exchanged between the modules by using the message format from the information model defined for the gateway.

Role Logic and Services are the only two modules that change with each role played by the peer. This is due to the difference in the logic that should be implemented for each role and the services that should be enabled according to the definitions made in chapter three.

Figure 4.1.3 presents the components that need to be defined for each role in the overlay, including the mobile sink and that are included in the Role Logic module. Firstly, each role should be identified by a name. The groups that the peer should join are also specified, including a list of all members. The list of members in a group at a given time is to support the self-recovery procedure. Additionally, the list of services provided by the role is defined at this point.

Finally, each role has a Message Reader and Writer component that prioritizes actions and decisions; e.g. whether a message needs to be translated and then analyzed or vice versa. The Message Reader and Writer main input/output is the message format from the information model. The actions, analysis and results are based in the actions, parameters and payloads included in each message received.

Each role can identify the services it has to provide as where previously defined. The Role Logic module identifies these services but it is the Service module that offers the actual implementation. It has the algorithms and procedures that allow each service to be supplied.

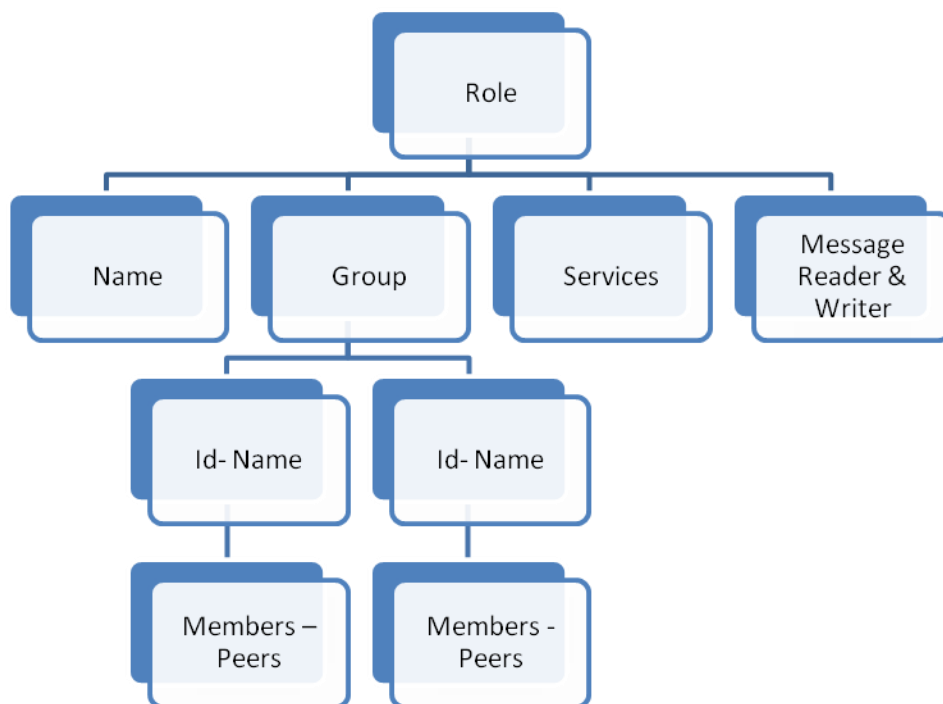


Figure 4.1.3. Role Logic structure

## 4.1.2 Prototype

As a proof of concept we tested the scenario proposed in chapter three, section 3.3. The idea of this prototype is to verify the feasibility of building an overlay gateway by using the proposed software architecture.

The development was done by following the software architecture previously depicted. The common modules (i.e. Overlay Manager, Overlay Mapping, SIP Manager and Gateway

Controller) were developed almost completely, however several features are missing since they were not mandatory for the proposed scenario.

Role Logic was developed for the roles needed in the scenario, basically *SSEP*, *PSE* and *SPSE*. Finally, Services was based on the selected roles, and included Discovery, Translation, Analysis and Publication.

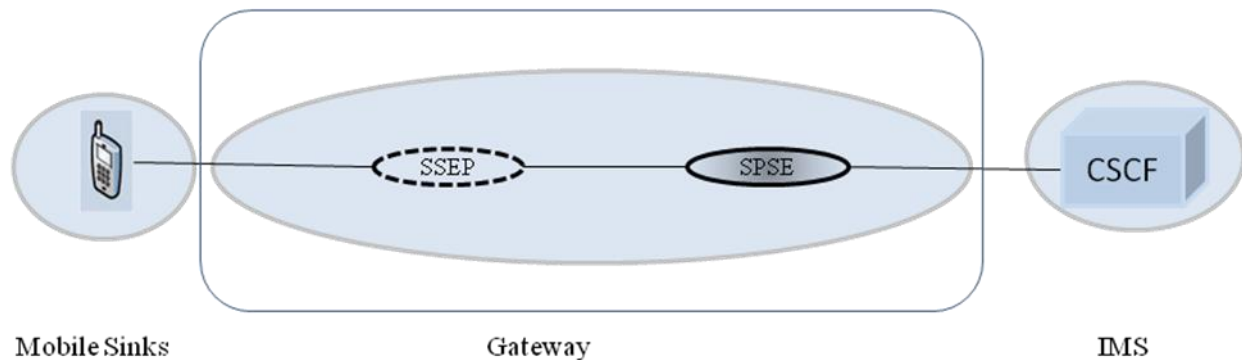


Figure 4.1.4. Proof of concept

Figure 4.1.3 depicts the entities involved in the scenario as it was described in chapter three. The following steps show how it was executed:

1. The *SSEP* joins the network.
2. A *PSE* enters the network, joins the overlay, and becomes the super peer *SPSE* since one does not yet exist. It then sends a SIP REGISTER to IMS servers.
3. The mobile sink's API discovers a sink entry point.
4. The *SSEP* receives the mobile sink address and connects to it.
5. The mobile sink's API resends the information collected to the *SSEP* through a SIP INFO message.
6. The *SSEP* analyzes the information received and when the temperature limit is above a pre-established threshold it sends a message using SIP NOTIFY to the *SPSE*, which then sends a SIP PUBLISH to IMS.

## Deployment

The prototype was built over two peers running in two different laptops. IMS was simulated in another laptop with the same characteristics and the mobile sink in an Intel Core 2 desktop computer.

On one hand the mobile sink was developed as a dummy process that sends a XML document towards a port periodically. On the other hand, IMS servers were simulated with Ericsson's Service Development Studio (SDS) [25]. It contains a standard-based IMS network simulator with communication services emulators that allow testing of both the client and server side of IMS architecture.

For the implementation of the software architecture we worked with Java Micro Edition (J2ME) which is Java Runtime Environment for mobile devices and used NetBeans IDE 6.8.

We selected JXME [26] as our middleware. It is a subset of JXTA; a set of protocols and an API that provides a generic implementation framework for P2P networks. It provides an automatic placement and a location procedure common in all P2P systems which helped us to focus in the specific designs of the application instead of the basics of P2P.

As the SIP implementation for the SIP stack we used JSR180 (the SIP API for J2ME). It provides basic SIP user agent functionality, including SIP dialog messages (e.g. SIP SUBSCRIBE/NOTIFY), required in our project.

The Role Logic component was developed for *PSE*, *SPSE*, *SSEP* and the mobile sink. For the *PSE* however none of the services were implemented since it is not going to be needed in the prototype. *SPSE*'s services include discovery, publish and translate. Discovery is a common service to all peers in the interconnection groups; its goal is to retrieve the identification of other leader peers when requested by joining nodes via the SIP INFO message. Publish is in charge of sending SIP PUBLISH when a request is received. But, before publishing any information, *SPSE* translates using the translation service that translates from the gateway information model to the defined PIDs.

For the *SSEP* the services implemented were discovery, translate and analyze. Translation was developed from the mobile sink WSN information model to the gateway information model. After translation, the information is analyzed firstly to determine if the threshold was violated and



thus if it needs to be published. Storage and processing actions were not developed as part of the prototype.

Finally for the mobile sink at this point, the only service needed to be implemented is the connection request that is sent to the *SSEP*. After this the information is sent periodically to the *SEP* (or the *SSEP* as is the case in our scenario).

The scenario was tested and executed successfully. Figure 4.1.5 presents some of the screen shots that we obtained during the execution of the prototype. Part (a) shows the connection process of the *PSE* from its side while (b) depicts the process from the *SSEP* perspective. From the *PSE* point of view Figure 4.1.5 (a), it firsts joins the overlay and requests to the Overlay Manager a peer ID to identify. Then the roles are defined, starting with the *PSE*. The SIP Client in the SIP Manager module sends a discovery request and waits for a discovery answer to arrive. The SIP INFO message indicates that there is no leader peer yet, thus the *PSE* changes its role for the *SPSE*. Once the change is done, the peer sends a SIP SUBSCRIBE to the *SSEP* as defined in the procedures in chapter III.



Figure 4.1.5. Prototype

The process in the *SSEP*, Figure 4.1.5 (b), shows the messages received from the peer connecting. Firstly, the SIP Server from the SIP Manager module indicates that a SIP INFO message requesting a discovery has arrived and thus, the *SSEP* processes the request and sends a

SIP INFO message with a discovery answer, informing that no leader for the group has been selected. After the other peer changes the role, the *SSEP* receives a SIP SUBSCRIBE message and a SIP dialog is established between both nodes.

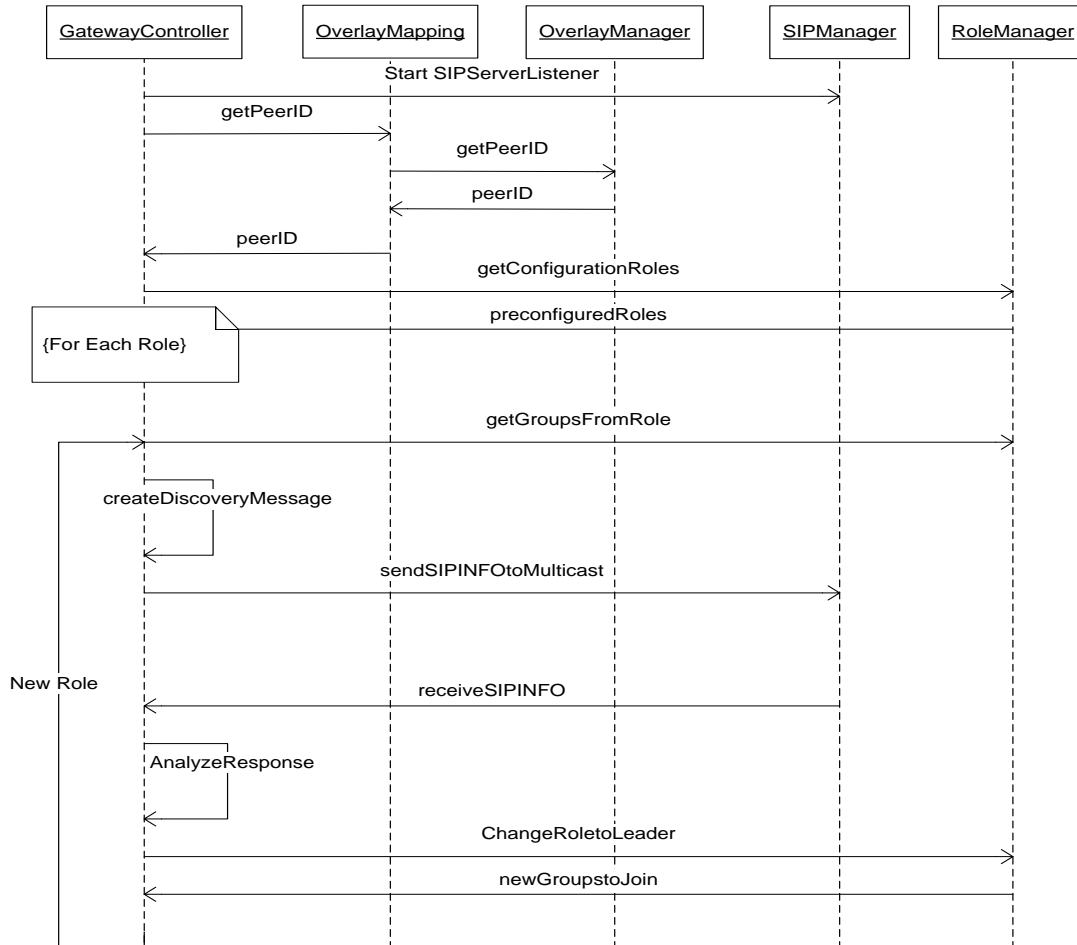


Figure 4.1.6. Sequence diagram joining node – discovery request

Figure 4.1.6 presents a sequence diagram which depicts in further detail interactions between the components inside the *PSE* when sending the discovery request to the *SSEP*. The first step for every peer joining the network is to start a SIP Server Listener that enables the reception of SIP messages. Then a request is sent to the Overlay for a unique peer identifier. After that, the preconfigured roles are obtained and the process to join all groups starts for each role. In this case the only preconfigured role is the *PSE*. The Role Manager returns the groups for the role, i.e. presence service entry point. Then, a message with action “*discovery*” is created by the Gateway Controller and then sent to the SIP manager as a request to send a SIP INFO to the

multicast address. Later on, once the *SPSE* responds, a SIP INFO message is received by the SIP Manager and transmitted to the Gateway Controller. When the Gateway Controller gets the information, it analyzes it to determine what message was received. The “*discovery answer*” specifies that there is no leader of that group registered, thus the Gateway Controller sends a request to the Role Manager to change the role to the leader.

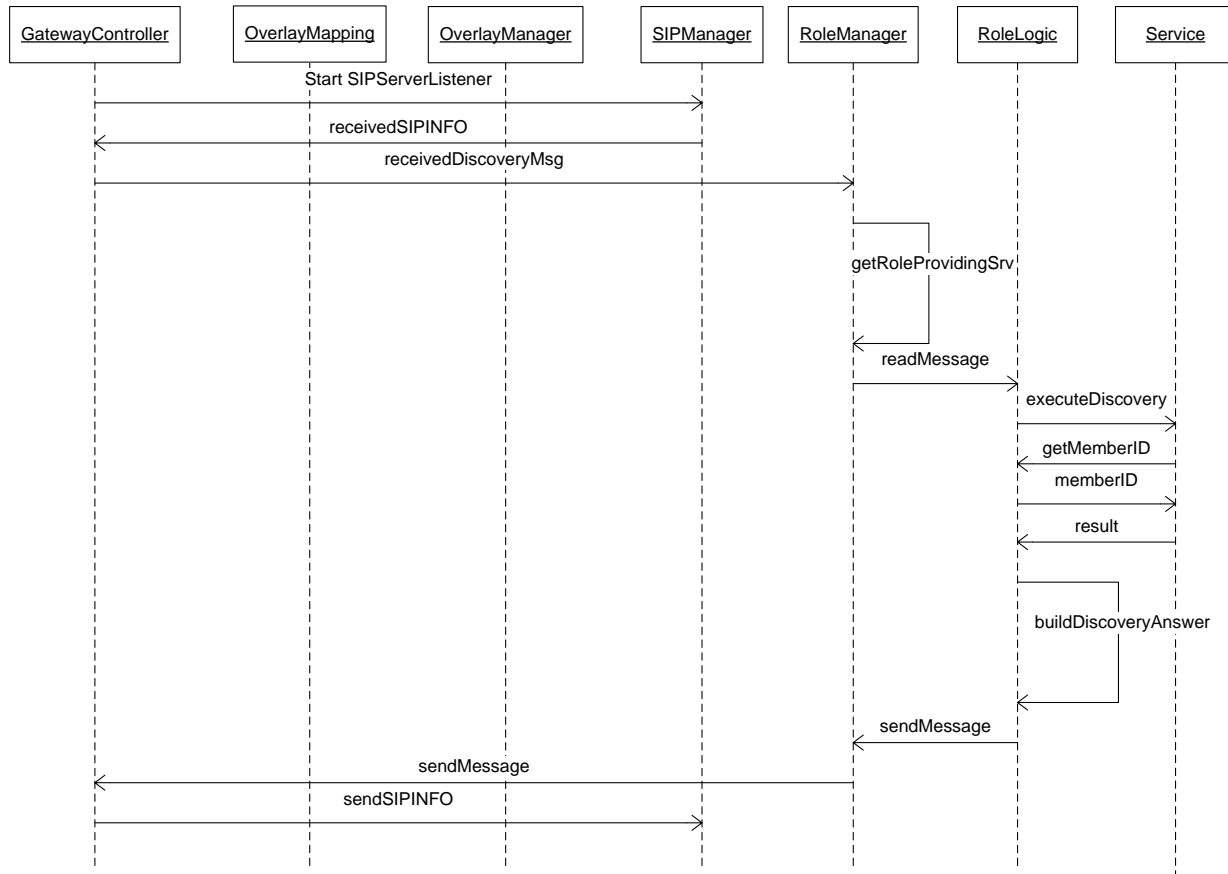


Figure 4.1.7. Sequence diagram SPSE - receiving discovery request

Figure 4.1.7 presents the sequence diagram inside the *SSEP*. Firstly, the SIP Manager receives a SIP INFO message which contains a “*discovery*” and forwards it to the Gateway Controller. The Gateway Controller sends the request to the Role Manager which identifies the role that offers the requested service. The data is sent to the Role Logic for the *SSEP* to read the message. The discovery service is activated, and thus a request to get a peer ID from the member list that is kept by each role as was explained previously. The result is sent to the Role Logic which then builds a “*discovery answer*” and sends the request to the Role Manager. Finally, the request is

resent to the Gateway Controller which forwards it to the SIP Manager that actually sends the message through the SIP Stack.

This prototype is a proof of the concept that we proposed in this project. It shows the feasibility of building a P2P overlay gateway that will interconnect mobile sink WSNs and IMS. The software architecture is generic and open to support other services not included in this scenario, serving as a guide to build them in future developments.

## 4.2 Validation

The validation process is done from three perspectives: firstly we will show how the architecture meets the requirements; secondly we will compare our proposal with the architectures that integrate WSN and IMS and were described in the state of the art; finally, it is our goal to prove through a simulation that using a P2P overlay gateway as part of the solution increases scalability and fault-tolerance when compared with a centralized gateway solution.

### 4.2.1 Requirements vs. Architecture

Before verifying how the architecture assesses the requirements, let us review what they are as defined in chapter II:

- *Independence.* The integration should be as independent as possible from both the WSN and IMS.
- *All types of information related to all types of entities sensed by a WSN should be made available to IMS.*
- *Translation* from the information model used in the sensor network to the information model used in IMS and vice versa.
- *Publication* of the information into IMS.
- *Storage and processing capabilities* to handle the requests from IMS as a way to support both synchronous and asynchronous mode to access the information.
- *Consider traditional and P2P IMS.*
- *Scalable and fault-tolerance.*

- *Support lightweight communication mechanisms.*

Although the architecture was defined in the light of these requirements, it is important to validate which features of the architecture satisfy which requirement. Table 4.2.1 presents a matrix where each requirement is achieved by one or several features presented in the proposed architecture.

Table 4.2-1 Requirements vs. Architecture

	Gateway as integration solution	P2P overlay gateway	Service definition	Extended PIDF	Information Model	Mobile devices
<b>Independence</b>	X				X	
<b>Information and entities from WSN</b>				X		
<b>Translation</b>			X		X	
<b>Publication</b>			X			
<b>Storage</b>		X	X			
<b>Processing</b>			X		X	
<b>Traditional and P2P IMS</b>	X	X				
<b>Scalability</b>		X				
<b>Fault-tolerance</b>		X				
<b>Light-weight communication</b>						X

**Independence**, for instance is achieved by several features including the use of a gateway and defining a new information model. Using a gateway as an integration point provides independence from the two technologies merged as was shown in chapter II section 2.7, it requires minimal or no changes in the architectures involved since all the translation is done by the gateway. Additionally, independence from the gateway point of view is also achieved by the information model used inside. It assures that minimal changes will need to be done in case the information model from the mobile WSN or the presence service in IMS change.

The provisioning of *all types of information and entities* from WSN in IMS is accomplished by the extended PIDF that we have chosen to use in our architecture. This extension allows the modification of presence information to add non-user related information.

**Translation** is provided as explained in the previous chapter by two services that were defined in the overlay. The first one is the service that translates between the information model

from the mobile sink WSN and the one from the gateway. The second one is between the extended-PIDF and the information model from the gateway. The information model is flexible enough to support both models. Thus, the features included in the architecture to meet the requirement are the defined services and the information model.

**Publication** is supplied by the service defined as part of the IMS entry point group. This service allows the publication of information in IMS by using SIP PUBLISH. Similarly, **Processing** is done in the service that has been defined for this purpose; which is supplied by the processing group and the information model.

Data **Storage** is met by two features, the P2P overlay and the service definition. The service definition as shown before with the other information management processes, is achieved through the service that has been defined for this purpose and that is supplied from a specific group (i.e. Storage group). However, this is also possible thanks to the P2P overlay; since the feasibility to provide distributed storage in P2P networks has been proven.

**Traditional and P2P IMS** were considered all along the design. Having a P2P overlay as gateway makes the future integration with P2P IMS more natural. Additionally, this feature also helps achieved the **Scalability** and **Fault-Tolerance**, which, as seeing before, are characteristics in P2P systems. However, it is important to clarify that in our project we do not consider unexpected departure, which is also important to improve the fault-tolerance in the solution.

Finally, the fact that we built our overlay gateway with the same mobile phones that act as mobile sinks and presence user agents, obliges us to work with **lightweight communication mechanisms**.

#### 4.2.2 Architectures comparison

Once it has been proved that the proposed architecture meets the requirements defined according to the motivations, we will compare the solution with other gateways proposed for the integration of these two technologies. The proposals that have been chosen for the comparison are WSN/IMS gateway from El Barachi et al. [3], the e-SENSE architecture proposed by Gluhak and Schott in [5] and the PN gateway to integrate Personal Area Networks and Sensor Networks with IMS described in [4] by Arbanowski et al.

Table 4.2.2 depicts the comparison between proposed architectures; it shows for each requirement whether it is Supported, Not Supported, Partially Supported or Not Considered in the design.

Table 4.2-2. Comparison between proposed gateways for integrating WSN and IMS  
S: Supported, N: Not supported, P: Partially supported and N.C.: Not considered

Requirement	Overlay Gateway	WSN/IMS gateway	e-SENSE architecture	PN Gateway
<b>Independence</b>	<i>S</i>	<i>S</i>	<i>N</i>	<i>S</i>
<b>All types of information and entities</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>Translation</b>	<i>S</i>	<i>S</i>	<i>N</i>	<i>N</i>
<b>Publication</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>Storage</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>Processing</b>	<i>S</i>	<i>S</i>	<i>S</i>	<i>N</i>
<b>Enable traditional and P2PIMS</b>	<i>S</i>	<i>N.C.</i>	<i>N.C.</i>	<i>N.C.</i>
<b>Scalability</b>	<i>S</i>	<i>N</i>	<i>N</i>	<i>P</i>
<b>Fault-tolerance</b>	<i>P</i>	<i>N</i>	<i>N</i>	<i>N</i>
<b>Lightweight communication mechanism</b>	<i>S</i>	<i>N</i>	<i>N</i>	<i>S</i>

The PN gateway proposed by Arbanowski et al. considers some of the non functional factors that other proposals do not. The design is meant to work in hardware devices with limited resources like wireless sensors or other tiny smart devices. For this purpose they use TinySIP (standard protocol for any kind of resource constrained device) to enable interactions between nodes through SIP messages. It is independent from IMS since the connection is made through an entity called SmartDust Enable or via specific devices that connect directly to CSCFs servers. However, this solution does not explain how the information will be processed, stored and publish in IMS or other entities.

The e-SENSE architecture supports most of the information management procedures we defined, except for translation which is not needed in this proposal. As mentioned in chapter two, the functional entity added in IMS is the e-SENSE Service Enabler SE that provides sensor-based information, thus including all types of information in IMS. It additionally offers storage and processing of information. This entity receives the information from one or several e-SENSE

Gateways. The gateway offers publication of information to the e-SENSE SE. Translation is not supported because there is no need of translation, the information is delivered as sensor-based information and not in other format.

However this solution does not support the non functional requirements related to scalability, fault-tolerance and lightweight communication mechanisms. The main drawback of this solution is that it is not independent from IMS. It adds a new server to IMS called the e-SENSE SE, instead of using one of the existents such as the presence service. Furthermore, storage and processing is supported by the e-SENSE SE and not by the gateway.

WSN/IMS gateway proposed by El Barachi et al. supports all information management requirements. It is a generic gateway that proposes minimal changes to the Presence Server in order to support diverse types of information. These types of information are also included in the proposal of the extended Pidf. Additionally, the gateway supports processing, storage, translation and publication of information.

Scalability, fault-tolerance and lightweight communication mechanisms are not considered as part of the solution. Additionally, P2P IMS is not considered for future scenarios and the integration may not be done straightforward since it can become a bottleneck.

As can be seen in the analysis, our architecture meets all the requirements; it was previously exposed which features in the architecture contributed to their accomplishment. The only one that is partially supported is the fault-tolerance, due to the lack of definition for an unexpected departure of a peer in the overlay. However the procedure can be later extended and is flexible enough to support the changes.



## Chapter 5 CONCLUSION

In this dissertation we have proposed an overlay architecture that serves as a gateway to connect mobile sink WSNs and IMS. This architecture allows the provisioning of new services to users and an optimized and more efficient use of the information collected in the mobile WSN.

Its main contribution is that it opens the door to a wide set of current and future applications in IMS based on the rich contextual information collected from the sensor network.

Another advantage that this proposal offers is that the gateway does not require any new infrastructure since it has been designed to be built over the same mobile phones that are used as mobile sinks and as user agents in both technologies.

This last chapter presents a summary of the work that has been exposed in this project. Later on, accepted publications are described. Finally, future applications and possible extensions to overcome the limitations of the current proposal are described.

### 5.1 Summary of the work

Mobile sink WSNs have been proposed as a solution to improve the overall network lifetime by implementing efficient data collection, sensor reprogramming, and data querying, among others.

These networks enable applications such as environmental monitoring, home surveillance and healthcare delivery. In agricultural applications, for instance, the data gathered is important and useful if it can be accessed in real-time. Thus, it could be further enhanced by using IMS enablers such as messaging and multimedia sessions. Our architecture was designed based on several requirements defined in chapter two and obtained from the analysis of the agricultural scenario, specifically for vineyard computing.

This proposal not only offers the possibility to access information from one or several mobile sinks but it also stores and processes data when required.

Moreover, during the design process, independence was constantly considered to ensure minimal changes in the involved technologies or when proposing further extensions. Independency is present in the architecture from three different perspectives; firstly no changes

are needed from the mobile sinks and only minimal are required in the presence service. Secondly, we propose the use of an independent information model to transport data inside the gateway. And finally, the software architecture was designed in components which make the proposal easily flexible to be overlay middleware- and protocol- independent.

Additionally, the overlay gateway here proposed considers factors such as scalability and fault-tolerance; which have hardly been considered in current proposals.

The integration with IMS was done through an already existing IMS service, the presence framework. This service already offers some basic contextual information and its information model (i.e. PIDF) was extended to support the non-user related data that could be recovered from a WSN.

To prove our concept, the software architecture for a peer in the overlay was designed and a prototype based on this architecture was implemented. It was developed in J2ME using the SIP API JSR180 as the overlay protocol and JXTA specification for mobile devices, JXME, as the overlay middleware. The prototype was successfully executed to test the proposed agricultural scenario.

The creation of a gateway to use the sensing capabilities and the information from mobile sink WSNs in the presence service exposed in the IMS architecture opens the door to a wide range of applications such as agricultural monitoring as described in this dissertation. It enables applications and other services in IMS, providing effective and real-time feedback to users. Furthermore, P2P IMS services can also be connected with the gateway and receive WSN information through several *PSEs*.

A mobile phone-based architecture in which one or several mobile phones cooperate to deliver the information, gives better results regarding self-recovery and scalability, compared with the current solutions built in personal computers.

## **5.2 Publications**

The work done in this project has been accepted to the 2010 IEEE 72nd Vehicular Technology Conference: VTC2010-Fall that was held 6–9 September 2010 in Ottawa, Canada.

### 5.3 Future Works

At this stage, our proposal only includes a procedure of self-recovery for voluntarily departure. A further extension would include the procedure when a node fails. The recovery process should consider the different roles present in the overlay, since the actions could differ from one another. This extension will increase the fault-tolerance of the overlay, complementing the already defined self-recovery process.

The implemented prototype that was presented in this memoire is a subset of the architecture. A future phase will be extending the prototype to support more functionality. Processing of information should be further detailed to determine when and how data should be sent to the processor group. The storage process in P2P systems should also be further analyzed, data distribution is especially important in this types of networks since it should be determined where data is stored and how to efficiently retrieve it. Flexibility when considering these new functions should be constantly present, since each service could change depending on the target application. It is important to define a clear way of indicating when information should be stored, processed or published and where should the retrieval process take place (e.g. from the stored information or directly from the mobile WSN).

Furthermore, integration with P2P IMS although considered has not yet been defined. A later extension could analyze the integration with this new IMS proposal.

In a final phase the efficiency of the proposed architectural concepts will be verified in field test.

## REFERENCES

- [1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, pp. 2292-2330, August 2008.
- [2] C. Gonzalo and G.-M. Miguel, *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*, Third ed., 2004.
- [3] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek, and R. Dssouli, "The design and implementation of a gateway for IP multimedia subsystem/wireless sensor networks interworking," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference* Barcelona, Spain, 2009.
- [4] S. Arbanowski, L. Lange, T. Magedanz, and L. Thiem, "The dynamic composition of personal network services for service delivery platforms," in *4th IEEE International Conference on Circuits and Systems for Communications, ICCSC* Shanghai, China, 2008.
- [5] A. Gluhak and W. Schott, "A WSN System Architecture to Capture Context Information for beyond 3G Communication Systems," in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007* Melbourne, Australia, 2007.
- [6] G. Coulouris, J. Dollimore, and K. Tim, "Peer-To-Peer Systems," in *Distributed Systems - Concepts and Design*, Fourth ed, pp. 397 - 432, 2005.
- [7] A. M. Saad, B. Ren, W. Jiao, B. Wang, D. Xie, and J. Ma, "Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)* Niagara Falls, Canada, 2007.
- [8] Y. Yang, M. I. Fonoage, and M. Cardei, "Improving network lifetime with mobile wireless sensor networks," *Computer Communications*, vol. 33, pp. 409–419, March 2010.
- [9] L. Benyuan, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *International Symposium on Mobile Ad Hoc Networking & Computing*, ACM, 2005.

- [10] A. Liotta and L. Lin, "The Operator's Response to P2P Service Demand," *IEEE Communications Magazine*, vol. 45, pp. 76-83, July 2007.
- [11] I. Radovanovic, J. Lukkien, S. Chen, C. Molanus, and T. Ozcelebi, "Virtual community management for enabling P2P services in the IMS network," in *IMSAA'08 - 2nd International Conference on Internet Multimedia Services Architecture and Application* Bangalore, India, 2008.
- [12] J. Hautakorpi, A. Salinas, E. Harjula, and M. Ylianttila, "Interconnecting P2PSIP and IMS," in *2nd International Conference on Next Generation Mobile Applications, Services, and Technologies, NGMAST 2008* Cardiff, Wales, United kingdom, 2008.
- [13] L. Le and G.-S. Kuo, "A novel P2P approach to S-CSCF assignment in IMS," in *2008 5th IEEE Consumer Communications and Networking Conference, CCNC 2008* Las Vegas, NV, United states, 2008.
- [14] M. Matuszewski and E. Kokkonen, "Mobile P2PSIP - Peer-to-Peer SIP communication in mobile communities," in *2008 5th IEEE Consumer Communications and Networking Conference, CCNC 2008* Las Vegas, NV, United states, 2008.
- [15] IETF, "P2PSIP WG.," 2010.
- [16] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard Computing: Sensor Networks in Agricultural Production," *IEEE Pervasive Computing*, vol. 3, pp. 38-45, January - March 2004.
- [17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102-105, August 2002.
- [18] L. Bencini, F. Chiti, G. Collodi, D. Di Palma, R. Factacci, A. Manes, and G. Manes, "Agricultural monitoring based on Wireless Sensor Network technology: real long life deployments for physiology and pathogens control," in *2009 Third International Conference on Sensor Technologies and Applications*, 2009.
- [19] I. Akyildiz and E. Stuntebeck, "Wireless underground sensor networks: Research challenges," *Elsevier*, vol. 4, pp. 669-86, November 2006.

- [20] W. Wang, Y.-X. Zou, G. Shi, and Y. Zhu, "A web service based gateway architecture for wireless sensor networks," in *11th International Conference on Advanced Communication Technology, ICACT 2009* Phoenix Park, Republic of Korea, 2009.
- [21] T. Ta, N. Y. Othman, R. H. Glitho, and F. Khendek, "Using Web Services for Bridging End-User Applications and Wireless Sensor Networks," in *11th IEEE Symposium on Computers and Communications, ISCC'06*, 2006.
- [22] C. Jardak, K. Rerkrai, A. Kovacevic, J. Riihijärvi, and P. Mähönen, "Email from the vineyard," in *2009 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities and Workshops, TridentCom 2009* Washington, DC, United states, 2009.
- [23] M. Strohbach, J. Vercher, and M. Bauer, "A case for IMS," *IEEE Vehicular Technology Magazine*, vol. 4, pp. 57-64, March 2009.
- [24] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, and J. Schiller, "Connecting Wireless Sensornets with TCP/IP Networks," in *Wired/Wireless Internet Communications*. vol. 2957/2004, pp. 583-594, 2004.
- [25] Ericsson, "Ericsson Service Development Studio," 2010. Available at: <http://www.ericsson.com/developer>
- [26] "JXTA-JXME Project.". Available at: <https://jxta-jxme.dev.java.net/>