

D1.1 - *Description of Work*

Intégration d'objets connectés hétérogènes et démonstrateurs

Participant(s) :

- Casanova Adrien, acasanov@polytech.unice.fr, SI5 (IHM)
- Llorens Alexia, la002769@etu.unice.fr, Master 2 (IHM)
- Sallé Victor, salle@polytech.unice.fr, SI5 (IAM)
- Turchini Marie-Catherine, turchini@polytech.unice.fr, SI5 (IAM)

Encadrant(s)

- Rey Gaëtan, gaetan.rey@unice.fr, Laboratoire I3S (Equipe Rainbow)
- Lavirotte Stéphane, stephane@lavirotte.com, Laboratoire I3S (Equipe Rainbow)

Coût du livrable : [48h/étudiant] heures

Résumé Exécutif

Depuis le milieu des années 2000, on assiste à une expansion et une démocratisation phénoménale des objets connectés qui envahissent notre quotidien. Beaucoup ont ainsi fait l'acquisition d'un bracelet sportif, d'une station météorologique connectée ou encore d'une montre intelligente. En conséquence, l'utilisateur se retrouve plongé dans un milieu d'objets hétérogènes et doit interagir avec eux par le biais d'une multitude d'applications. Derrière cette problématique s'en cache une autre, plus technique : comment permettre aux développeurs d'homogénéiser l'accès aux objets et à leurs données ? Le but de ce projet est de répondre à cette problématique en fournissant un cadre de travail permettant de décloisonner leur usage afin de remonter l'ensemble des informations à un même niveau.

Ce document décrit notre approche du sujet, des défis et objectifs identifiés ainsi que la méthodologie adoptée.

Abstract

The past decade has witnessed the expansion and democratization of smart objects. These devices, of all kind - phones, watches, bracelets, weather stations, ... - have invaded our lives. Consequently, users are caught in a web of heterogeneous objects and interact with them through dedicated apps. The key challenge today is to homogenize the access to these objects in order to exploit and connect their data. The purpose of our project is to provide developers a framework to fill technological gap between objects by creating a unique layer to access them.

In this paper, we present the main challenges and goals of the project aside with the methodology to achieve them.

Table des matières

1. Description du Projet	3
Contexte de travail.....	3
Motivations.....	3
Défis	4
Objectifs	5
Scénarii.....	5
Critères de succès	6
2. Etat de l'art	7
Description Générale.....	7
Différents moyens de communication	8
Différentes interfaces de programmation	10
Mise en commun de plusieurs objets.....	11
Une perspective d'avenir : le "Physical Web" de Google	11
Solution proposée	11
3. Méthodologie et Planification	14
Stratégie Générale.....	14
Découpage en lots.....	14
Planification.....	15
Livrables associés au projet.....	16
Jalons.....	16
Pilotage et suivi.....	16
4. Description de la mise en œuvre du projet	17
Interdépendances des lots et tâches	17
Description des lots.....	18
Résumé de l'effort.....	24
Gestion du risque	25
5. Participants.....	26
Etudiant - Adrien Casanova (SI5 – IHM).....	26
Etudiant - Alexia Llorens (Master 2– IHM).....	26
Etudiant - Victor Sallé (SI5 – IAM)	26
Etudiant - Marie-Catherine Turchini (SI5 – IAM)	26
Encadrant - Gaëtan Rey (Laboratoire I3S - Equipe Rainbow)	27
Encadrant - Stéphane Lavirotte (Laboratoire I3S - Equipe Rainbow)	27
6. Références	28

1. Description du Projet

Contexte de travail

L'informatique ubiquitaire est un des enjeux majeurs d'aujourd'hui et de demain. Troisième grande ère de l'informatique, elle se caractérise par l'omniprésence de la technologie et de l'accès à l'information. Durant ces deux dernières années, de nombreux objets connectés sont arrivés sur le marché et s'intègrent de plus en plus à notre quotidien. On les retrouve dans de nombreux domaines - téléphonie, santé, bien-être, domotique, ... - au travers d'appareils très divers tels que des téléphones, montres, tensiomètres, pèse-personnes, matelas, etc.

Cependant il existe un très grand nombre d'objets et la toile qu'ils constituent est très hétérogène. Ainsi, le défi d'aujourd'hui est d'intégrer ces objets dans l'environnement de l'utilisateur afin d'acquérir son contexte et s'y adapter automatiquement. Il survient alors un réel besoin pour les développeurs d'avoir un cadre commun permettant de gérer l'hétérogénéité des dispositifs et donc de disposer d'un accès unifié à leur paramétrage et à leur utilisation. Ce cadre doit également permettre de découpler l'utilisation de chaque dispositif de manière à pouvoir les associer pour créer de nouveaux usages et de nouvelles applications mettant en œuvre non plus un seul dispositif mais une constellation de dispositifs.

Motivations

La multiplication des objets connectés conduit à un souhait de ne plus les utiliser séparément (chacun fournissant une information isolée) mais de réunir ceux-ci dans des familles ou des ensembles d'objets afin d'en faire des usages plus complets. En effet, associer les données de différents objets connectés dans le cadre d'informations sur le corps peut être utile dans le domaine de la santé. Néanmoins, actuellement, chaque objet dispose d'une application sur smartphone dédiée pour dialoguer avec lui. Ceci complique la mise en œuvre, en particulier pour les développeurs pour qui la communication avec un ensemble d'objets hétérogènes devient rapidement très complexe. Il ne s'agit pas seulement d'une question d'hétérogénéité au niveau des fonctionnalités, par exemple du couplage entre une montre pour mesurer l'activité et d'une balance pour mesurer l'évolution du poids. Recouper les informations de deux stations météorologiques venant de constructeurs différents est une opération difficile à cause des différents choix technologiques faits concernant les moyens de communiquer de ces objets.

Notre travail est justifié par le besoin qu'ont les développeurs de pouvoir faire communiquer différents objets connectés. Il est complexe de développer une application capable de manipuler des objets hétérogènes. Pourtant nous sommes face à une situation où le désir de ne plus faire d'application dédiée à un objet unique est capital : il en va de la pérennité même de l'utilisation des objets connectés. Nous devons donc proposer des solutions plus adaptées aux développeurs en ajoutant une couche d'abstraction et de généralisation pour passer outre les différences technologiques des objets. Le but est de pouvoir associer des objets pour créer de nouveaux usages.

Pour un utilisateur quelconque, notre projet représente la possibilité de pouvoir, sans connaissance particulière, interagir naturellement avec les objets, mais surtout pouvoir accéder à toutes les informations qui l'intéressent de façon globale et non plus en consultant une multitude d'applications diverses et variées. Le but est donc de proposer un moyen de lier ces objets, via une interface unique, et ainsi de tirer parti de chacun d'entre eux.

Défis

Ce projet a pour but de répondre à la problématique grandissante du très grand nombre d'objets connectés qui envahissent notre quotidien, et dont le concept « un objet – une application » porte grandement préjudice au phénomène.

Pour cela, nous allons nous atteler à développer une solution permettant de gérer cette extrême hétérogénéité technologique, et ainsi proposer une simplification du mode de communication entre les objets et la couche applicative. Cela permettra donc d'abstraire toute contrainte technologique et d'homogénéiser les différents protocoles utilisés. Une des conséquences directes sera un décroisement de l'utilisation de chacun de ces objets, afin de développer une solution davantage homogène et intelligente.

- Défi 1 : Uniformiser le mode de communication entre les différents objets et applications en donnant un accès simplifié au paramétrage des appareils
- Défi 2 : Uniformiser le format d'accès et de récupération de données et ainsi décroiser l'utilisation de chaque dispositif
- Défi 3 : Uniformiser le format de déclenchement d'action sur les dispositifs le permettant

Objectifs

L'objectif global est de proposer une API¹ unifiée, abstrayant tout aspect technologique et physique et rassemblant ainsi l'ensemble des API spécifiques, soit toutes les données pouvant être récoltées, à un même niveau d'abstraction.

- Objectif 1 : Etudier l'ensemble des technologies utilisées par les objets connectés possédés afin d'en extraire un modèle d'unification
- Objectif 2 : Proposer une API simple et claire permettant d'unifier les protocoles d'échanges
- Objectif 3 : Développer une solution permettant une intégration et une utilisation simplifiée d'objets connectés aux technologies hétérogènes
- Objectif 4 : Développer un démonstrateur de cette solution

Scénarii

Scenario 1 : Utilisation générale

Roger est un développeur. Il souhaite développer une application dédiée à la santé, qui lui permettrait de relier les informations de son bracelet Withings, de son pèse-personne Withings, et de sa fourchette HAPIfork.

Malheureusement, Roger se doute que le public visé est restreint du fait de l'hétérogénéité des objets. Son ami Pierre possède par exemple un équipement similaire, à la différence que son bracelet est un Fitbit plutôt qu'un Withings.

Roger se décide donc à créer une application générique, capable d'utiliser n'importe quel bracelet, n'importe quel pèse-personne, et n'importe quelle fourchette. Pour ce faire, il va utiliser la solution que nous lui proposons : celle-ci va lui permettre d'utiliser une API unifiée, offrant une syntaxe des requêtes/réponses et un mode de communication unifiés.

Lorsque Jean, utilisateur lambda, va télécharger l'application, il devra spécifier le matériel dont il dispose. Notre solution connectera alors automatiquement le dispositif à la brique logicielle correspondante, afin d'en adapter l'API à l'API unifiée.

¹ *Application Programming Interface*, interface par laquelle une application offre des services

Scénario 2 : Mise à disposition d'un nouveau proxy

Paul est un développeur impliqué dans la communauté open-source. Afin d'augmenter le nombre d'objets compatibles avec le système, il souhaite apporter sa pierre à l'édifice et créer un proxy pour l'un des objets connectés qu'il possède et pour lequel rien n'a encore été développé.

Il se renseigne alors sur Internet, afin d'en apprendre davantage sur l'API exposée par l'objet. Puis, via l'interface simple que nous lui mettons à disposition, il fait correspondre l'API de l'objet avec l'API unifiée, en agissant à la fois sur le mode de communication et sur les interfaces exposées.

Lorsque le proxy est terminé et testé, il le met à la disposition de la communauté qui pourra approuver ou non sa publication.

Critères de succès

Afin de certifier du succès de notre projet, nous aurons à vérifier plusieurs critères, découlant logiquement de nos objectifs. Tout d'abord, il faudra assurer la simplicité d'ajout d'un nouvel objet connecté à un ensemble déjà existant. Cette action doit pouvoir se faire quelle que soit la technologie utilisée ; pour cela, il faudra donc également que notre solution soit capable de gérer les principales technologies utilisées par les objets connectés. Enfin, l'un des buts de notre solution est de faciliter la manipulation de l'ensemble des données récoltées par l'ensemble d'objets. Afin d'attester de cette réussite, il faudra donc vérifier que notre système soit aisé à prendre en main.

- Critère 1 : Simplicité d'intégration d'un nouvel objet
- Critère 2 : La démonstration met bien en avant la facilité d'intégration et d'utilisation avec des objets dont l'hétérogénéité est réelle (plusieurs marques, plusieurs technologies, plusieurs fonctionnalités)
- Critère 3 : La solution gère les principaux regroupements d'objets connectés qui ont été faits

2. Etat de l'art

Description Générale

Le nombre d'objets connectés disponibles sur le marché augmente considérablement depuis plusieurs années. Chacun d'eux est aujourd'hui fourni avec une application propriétaire, permettant de le paramétrer et d'accéder aux données qu'il produit.

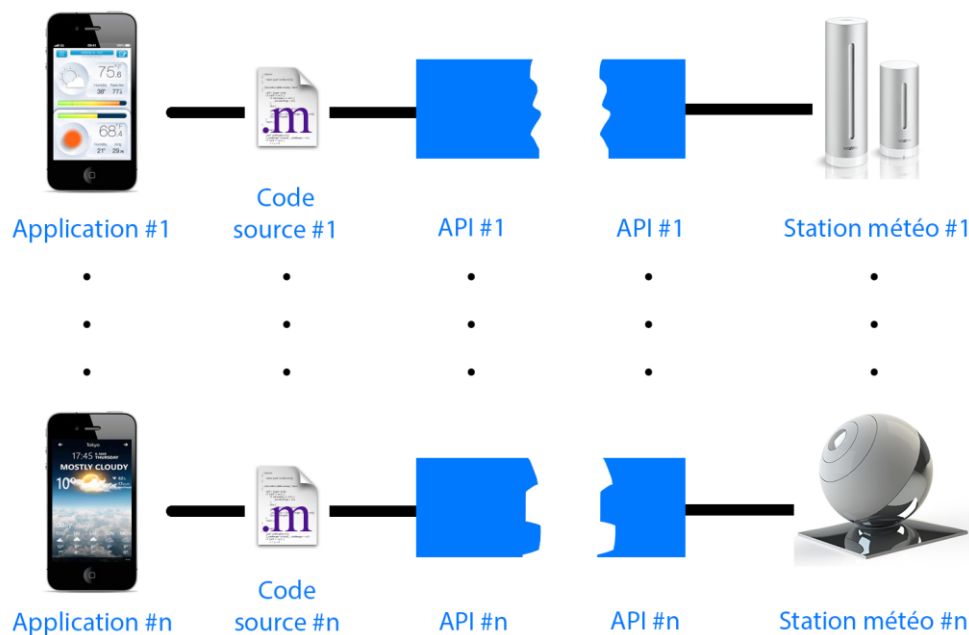


Figure 1 : Modèle "une application par objet"

De nombreux constructeurs fournissent une API publique, pour permettre aux développeurs de réaliser une application utilisant leurs objets. Cependant, aucune norme n'a été établie autour des protocoles de communication et des interfaces de programmation. Il est par conséquent difficile de proposer une nouvelle application faisant interagir plusieurs objets, puisque celle-ci sera restreinte aux utilisateurs qui disposent précisément des modèles pour lesquels l'application a été conçue. Ainsi, les développeurs se retrouvent souvent à développer un nombre exponentiel d'applications, en fonction du nombre d'objets et de leurs types. Illustration parfaite dans le cas suivant où l'on voit bien qu'il est nécessaire de concevoir deux mêmes applications alors que les objets sont similaires, uniquement afin d'adapter l'application aux deux modèles différents de station météo.

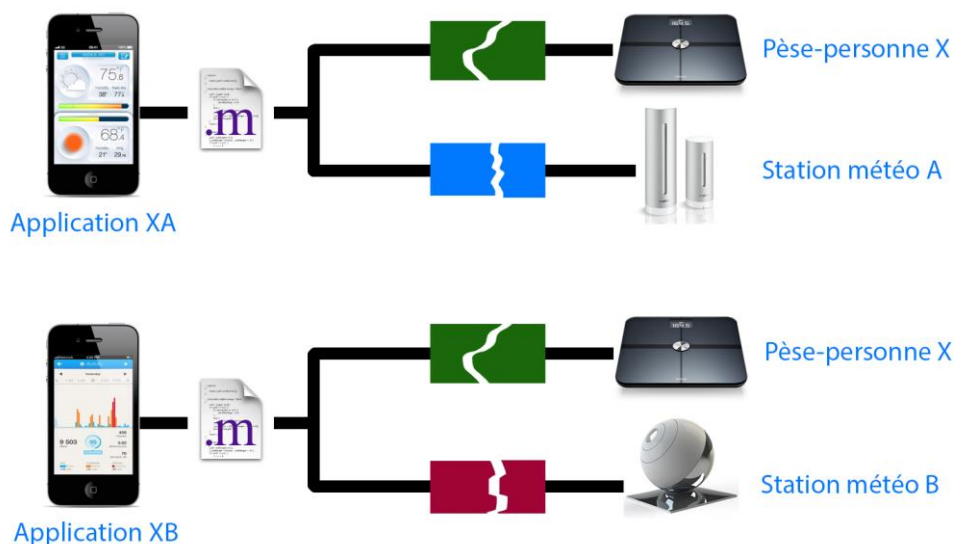


Figure 2 : Développement répété de la même application

Différents moyens de communication

A la différence du web où un protocole standard s'est imposé et est utilisé par tous les navigateurs web, à savoir le protocole HTTP, il n'existe actuellement pas de convention pour la communication avec les objets connectés. Différentes stratégies et différents protocoles de communication sont utilisés actuellement dans l'*Internet of Thing*².

Stratégies

De la même manière que le modèle client-serveur, la communication entre un objet connecté et son application cliente peut suivre deux stratégies différentes.

La première est appelée stratégie "pull". Le client va débiter le dialogue vers l'objet connecté, et va chercher à récupérer ("tirer") les informations connues par celui-ci. Cette méthode est la plus rapide à mettre en place et est de ce fait généralement la plus utilisée. Elle a cependant un inconvénient majeur puisque le client ne peut pas savoir à quel moment récupérer l'information. Ce dernier doit le faire de manière récurrente, et est donc amené à instancier de nombreux dialogues inutiles lorsqu'il n'y a rien à récupérer. Cette stratégie est donc généralement à proscrire dans le cas des objets connectés, principalement pour limiter l'usage des ressources et pour économiser la batterie de l'objet.

Pour pallier à ce problème, la stratégie dite "push" peut être mise en place. Ici, c'est l'objet qui va lancer la communication vers le client ("pousser") pour l'informer que de nouvelles données sont disponibles. Pour ce faire, chaque application qui souhaite être mise au courant du changement s'abonne aux notifications émises par l'objet. Lorsque de nouvelles données apparaissent, ce dernier n'aura alors qu'à diffuser l'information auprès de ses abonnés.

² Dénomination anglaise de l'informatique ubiquitaire

Protocoles de communication

Directement avec l'objet

Afin de communiquer avec les objets, différents protocoles existent et sont actuellement utilisés.

Des protocoles bas niveau

Bluetooth : le Bluetooth est un standard de communication né il y a vingt ans. Il est très répandu, et permet des communications entre des appareils distants d'une dizaine de mètres (voire jusqu'à cent mètres avec un émetteur puissant). Il a longtemps eu la réputation d'être énergivore, mais l'apparition récente du Bluetooth Low Energy a permis de diviser par dix sa consommation d'énergie.

ZigBee : mis au point il y a une dizaine d'années, le protocole Zigbee a pour but d'améliorer le Bluetooth, en offrant une portée supérieure, tout en diminuant les besoins en énergie et le prix du matériel. En contrepartie, il est quatre fois moins rapide que le Bluetooth.

Wi-Fi : même s'il est bien plus énergivore que les autres protocoles, le Wi-Fi offre deux avantages majeurs qui peuvent être indispensables, à savoir la portée (jusqu'à 300m) et le débit (jusqu'à mille fois plus rapide que le Bluetooth). Le Wi-Fi est donc le protocole le plus utilisé pour relier les ordinateurs à l'Internet.

Des protocoles haut niveau

UPnP : le protocole *UPnP* (« Universal Plug aNd Play ») a, comme son nom l'indique, été mis en place afin de faciliter la mise en réseau des différents périphériques, puisqu'aucune configuration n'est nécessaire. Cette simplicité d'utilisation a cependant un inconvénient, puisque ce protocole n'est de ce fait pas sécurisé.

DPWS : ce protocole (« Devices Profiles for Web Services ») correspond à la seconde révision des spécifications *UPnP*. Il apporte la compatibilité avec les standards des web services, et par là même une sécurisation des communications.

Avec un serveur

De plus en plus de constructeurs ne fournissent plus d'accès direct aux données depuis les objets, et proposent en contrepartie d'y accéder depuis leurs serveurs. Dans ce cas, l'objet envoie lui-même ses données au serveur et il est par conséquent nécessaire qu'il soit connecté à Internet.



Figure 3 : Le pèse-personne Withings envoie directement les données qu'il vient de récolter aux serveurs, puis le client communique directement avec ces mêmes serveurs

Différentes interfaces de programmation

Généralités

Lorsqu'un constructeur décide de mettre à disposition des développeurs une interface de programmation, il décide lui-même des services qu'il expose et de la manière dont il les expose, puisqu'aucune convention n'a été établie. Par conséquent, ces interfaces sont très hétérogènes, et l'accès à une même donnée diffère selon l'objet qui la propose, tant sur la structure de la requête que sur celle de la réponse, même si les formats d'échange sont généralement le JSON, voire le XML.

Sur certains de ces objets, il est également possible de déclencher des actions. Un lapin Nabaztag pourra par exemple recevoir l'ordre de lire un média ou d'allumer ses LEDs, tandis qu'un thermostat connecté pourra recevoir l'action de se régler à une température donnée.

De plus, chaque objet ne dispose pas forcément de sa propre interface. Le constructeur Withings offre par exemple une interface commune à tous ses objets, divisée en trois catégories (activité physique, sommeil). De cette manière, il est possible de récupérer une mesure (le rythme cardiaque par exemple) mais sans pouvoir savoir depuis quel objet elle a été mesurée (bracelet, pèse-personne ou tensiomètre).

Exemple concret

Afin d'illustrer cette diversité, étudions l'exemple de deux constructeurs offrant chacun un objet capable de mesurer une même valeur - la fréquence cardiaque - et voyons comment la récupérer à travers leur interface de programmation.

	Withings	Fitbit
Date	Intervalle de temps	Date précise
Format de date	Timestamp	YYYY-mm-dd
Utilisateur	Besoin de préciser	Utilise l'utilisateur actuellement authentifié

Tableau 1 : Requête

	Withings	Fitbit
Format	JSON	JSON ou XML, au choix
Date	Timestamp des heures de mesure	Heure de mesure non connue
Accès à la valeur (lecture du format de la réponse)	body.measuregrps[<i>].measures[<j>].value	heart[<i>].heartRate

Tableau 2 : Réponse

Mise en commun de plusieurs objets

L'application IFTTT, soit « If This Then That », est un outil d'automatisation de tâches, c'est-à-dire "si j'entre dans telle zone géographique alors mon téléphone active le mode vibreur", a récemment levé trente millions de dollars afin de relier les objets connectés.

Concrètement, un utilisateur disposant d'une station météo Netatmo et d'une lampe Philips pourrait définir une recette selon laquelle la lampe Philips s'allumerait en rouge si le niveau de dioxyde de carbone mesuré par la station météo dépasse un certain seuil.

Cette solution innove en proposant de relier les objets connectés d'une manière simple pour l'utilisateur. Cependant, chaque recette doit être définie autant de fois qu'il y a d'objets de même type et n'est donc pas réutilisable.

Une perspective d'avenir : le "Physical Web" de Google

Google travaille actuellement à la mise en place d'un projet nommé "Physical Web". Il s'agit d'un projet qui devrait permettre à terme de se passer de toute application dédiée aux objets. Tout serait directement intégré au système d'exploitation du smartphone, qui se chargerait de présenter à l'utilisateur les objets connectés qui l'entourent, chacun disposant de sa propre URL.

Google a décidé de faire de son projet un projet open-source, afin d'établir un véritable standard que tout le monde pourrait utiliser. Malheureusement, le déploiement d'un tel système est lent, et il est peu probable que celui-ci voit le jour avant plusieurs années.

Solution proposée

Il faudra pour cela, définir un framework³ qui permettra d'intégrer n'importe quel nouvel objet connecté à un ensemble déjà existant, et ce quelle que soit la technologie utilisée pour renvoyer les informations. Ainsi, le framework aura pour rôle

³ Cadre de travail

de créer une couche d'abstraction matérielle entre l'ensemble des objets et la couche applicative, qui sera alors libre d'accéder aux données récoltées par tous les objets connectés mais surtout de façon très simplifiée et unifiée, voire de déclencher une action sur ceux qui le permettent.

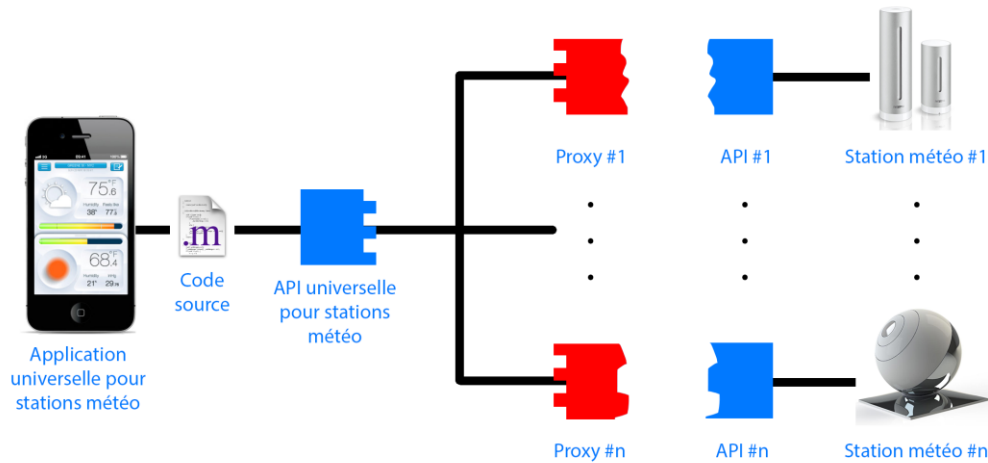


Figure 4 : Connexion de différents objets aux proxys correspondants

Pour cela, il faut voir notre projet comme un ensemble de proxys⁴ : chacun d'eux représentera un point d'entrée dans le framework ; chaque regroupement d'objets qui aura été fait possèdera son propre point d'entrée. Ainsi, un bracelet sportif renvoyant ses données par Bluetooth ne se connectera pas au même proxy qu'un pèse-personne Wi-Fi.

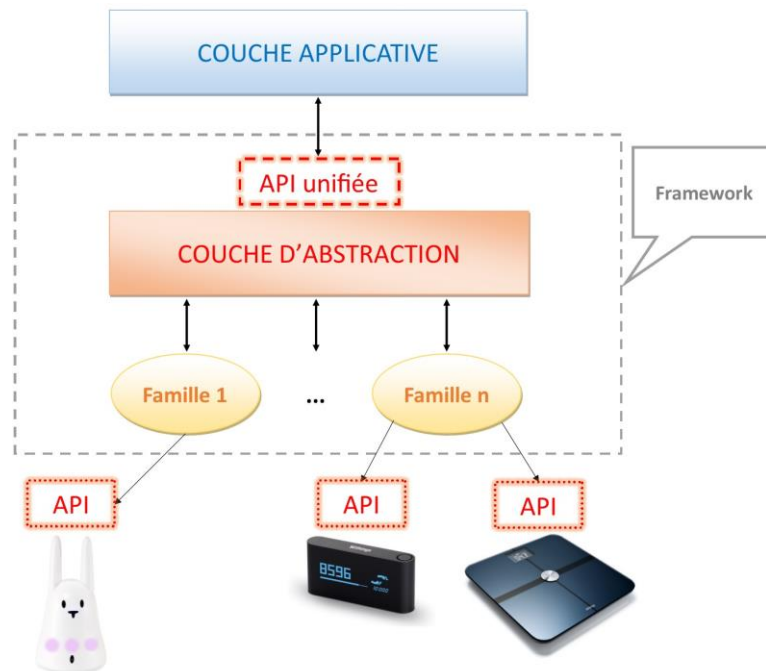


Figure 5 : Schéma de l'architecture générale du projet

⁴ Composants logiciels qui jouent le rôle d'intermédiaires

Cette architecture nous permet ainsi de masquer la technologie propriétaire de l'objet en question, et donc de briser ce lien entre objet connecté et API dédiée. Ces morceaux logiciels ont donc pour but de créer une sorte de *middleware*⁵, c'est-à-dire de réunir les objets par "famille" afin de les ramener à un même niveau d'abstraction de données, afin de faciliter le travail d'exploitation d'informations fait par la couche supérieure.

⁵ Logiciel qui crée un réseau d'échanges d'informations entre différentes applications informatiques

3. Méthodologie et Planification

Stratégie Générale

Afin de pouvoir présenter régulièrement un démonstrateur stable mettant en avant l'évolution du projet, nous avons choisi de suivre un cycle agile itératif dont chaque itération sera composée des phases suivantes :

- Phase 1 : étude des API des objets afin d'en retirer les technologies à gérer et comprendre comment les abstraire.
 - Première itération : étude des API de deux objets sélectionnés au préalable, de préférence de technologies différentes.
- Phase 2 : développement du framework et ajout d'objets de catégories différentes, c'est-à-dire utilisant des technologies différentes, par exemple : Wi-Fi et Bluetooth.
 - Première itération : ajout des deux objets présélectionnés et établissement de la méthode à appliquer par la suite.
- Phase 3 : mettre à jour le démonstrateur.

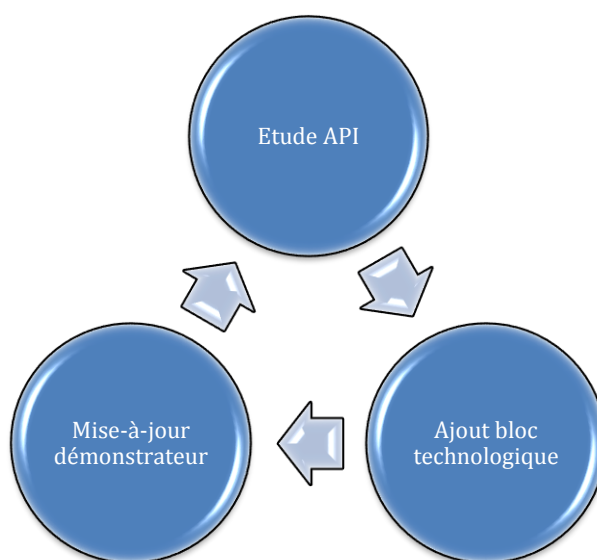


Figure 6 : Méthode itérative de développement

Découpage en lots

#	Titre du lot	Type	Leader	Budget	Début	Fin
L1	Management du projet	MGMT	Turchini	484	S1	S21
L2	Etude des technologies utilisées par les OC et de leurs API	RECH	Llorens	244	S5	S7
L3	Développement du framework	RECH/IMPL	Sallé	328	S8	S19
L4	Développement du démonstrateur	IMPL/DEMO	Casanova	208	S9	S19
Total :				[316h par étudiant]		

Tableau 3 : Liste des lots

Planification

#	Lot / Tâche	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21
	Sem. réelle	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10
L1	Management du projet																					
1.1	Planification																					
1.2	Suivi du projet																					
1.3	Rapport																					
1.4	Préparation soutenance																					
	Livrables et jalons																					
L2	Etude des technologies des OC et de leurs API																					
2.1	Etude des technologies utilisées par les OC																					
2.2	Regroupement en « familles » technologiques																					
L3	Développement du framework																					
3.1	Conception générale																					
3.2	Conception d'ajout nouveau(x) bloc(s) technologique(s)																					
3.3	Implémentation nouveau(x) bloc(s)																					
3.4	Tests fonctionnels																					
L4	Développement du démonstrateur																					
4.1	Conception du démonstrateur																					
4.2	Implémentation du démonstrateur																					
4.3	Tests fonctionnels																					
4.4	Retours utilisateurs																					

Figure 7 - Diagramme de Gantt

07	08
48	49

Périodes de travail
à temps plein

11	12
52	1

Vacances
scolaires

Livrables associés au projet

#	Titre du livrable	Lot	Nature
D1.1	Description of Work	1	DOC
D2.1	Étude des objets connectés et de leurs API	2	DOC
D2.2	Explications des regroupements par technologies	2	DOC
D3.1	Framework	3	LOG
D4.1	Démonstrateur	4	LOG
D5.1	Synthèse finale (rapport, diaporama et démonstration)	5	DOC

Tableau 4 : Liste des livrables

Jalons

#	Titre du jalon	Livrables	Lot(s)	Date
J0	Fin de la planification	D1.1	1	S04
J1	Fin de l'étude des API	D2.1-2	2	S07
J2	Démo #1	D4.1a	4	S10
J3	Démo #2	D4.1b	4	S15
J4	Démo #3	D4.1c	4	S19
J5	Fin du projet	D5.1	5	S21

Tableau 5 : Liste des jalons

Pilotage et suivi

Nous avons décidé d'employer une méthode agile itérative. Deux itérations sur les lots 3 et 4 ont été identifiées, elles correspondent aux phases de développement du framework et du démonstrateur associé. La première itération est plus longue car elle comprend le développement de deux blocs technologiques afin d'obtenir une véritable démonstration du fonctionnement du framework et notamment de l'ajout de nouvelles technologies.

Dans le but de rester cohérents, se maintenir à jour sur l'avancement de chacun des membres en cas de tâches séparées, de redéfinir et répartir les tâches ainsi que de réorienter le développement du projet si besoin, des réunions hebdomadaires sont prévues. A celles-ci s'ajoutent des discussions de groupe régulières sur Skype ainsi qu'un dossier commun sur Google Drive contenant tous les documents du projet.

Des rencontres régulières avec nos encadrants sont également programmées afin de leur faire part de notre avancement, profiter de leur recul, de leur expertise et ainsi avoir un retour sur notre travail. Une fréquence judicieuse pour ces réunions serait d'une toutes les deux semaines plus une avant chaque jalon.

Enfin, nous utiliserons Git comme gestionnaire de versions.

4. Description de la mise en œuvre du projet

Interdépendances des lots et tâches

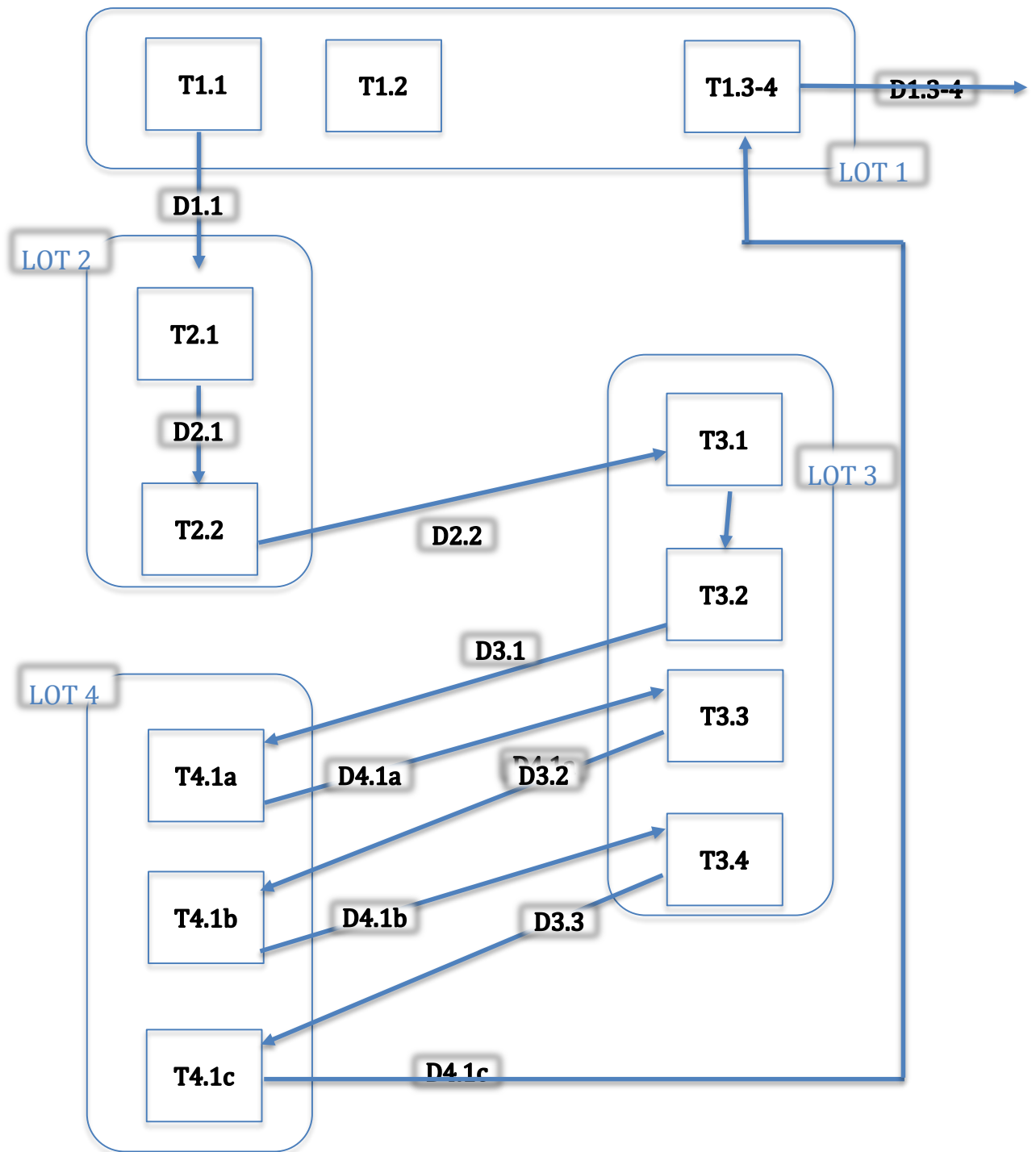


Figure 8 : Interdépendances entre lots et tâches

Description des lots

Identifiant	L1			Date de démarrage	S01	
Titre	Management du projet					
Type	MGMT					
Participant	Casanova	Llorens	Sallé	Turchini	Lavirotte	Rey
Effort	121h	121h	121h	121h	41h	41h

Objectifs du lot

Ce lot contient l'ensemble des tâches liées à la gestion de projet.

- Rédaction du document « Description of Work » (DoW)
- Suivi du projet
- Rédaction du rapport
- Préparation de la soutenance et du diaporama
- Présentation des résultats du projet lors de la soutenance finale

Description du lot

Tâche 1.1 : Rédaction du Description of Work (Turchini, S01-S04, 192h)

Cette tâche contient la rédaction de l'analyse effectuée en amont et répertoriant la description du projet, les défis et objectifs, l'état de l'art, la méthodologie, la planification et la description de la mise en œuvre du projet.

Tâche 1.2 : Suivi du projet (Sallé, S05-S21)

Cette tâche concerne le suivi de l'avancement du projet, c'est-à-dire l'ensemble des réunions, aussi bien entre les membres de l'équipe qu'avec les encadrants, mais également le respect des jalons et des livrables.

Tâche 1.3 : Rédaction du rapport (Llorens, S21-S21, 252h)

Il faudra au cours de cette tâche rédiger le rapport résumant l'ensemble de notre projet, le travail accompli, les difficultés rencontrées et décrivant les défis effectivement relevés.

Tâche 1.4 : Préparation de la soutenance (Casanova, S21-S21, 40h)

Cette tâche contient la préparation à la soutenance, les répétitions pour cette dernière, ainsi que la création du support (diaporama).

Livrable

Livrable D1.1 : DoW (Turchini, DOC, J1)

Ce document représente le premier cahier des charges du projet, résumant l'axe de travail adopté, les défis à relever, les objectifs à atteindre, ainsi qu'un ensemble d'outils de gestion de projet (jalons, planification, répartition de l'effort, etc.)

Livrable D1.2 : Rapport final (Llorens, DOC, J5)

Le rapport final du projet contiendra la description des objectifs qui auront été atteints, ainsi qu'une explication du déroulement du projet, c'est-à-dire la présentation d'un diagramme de Gantt effectif et non plus prévisionnel.

Livrable D1.3 : Diaporama de la soutenance (Casanova, DOC, J5)

Le diaporama sera notre support visuel présenté au cours de la soutenance, soit la présentation et démonstration finales de notre projet.

Identifiant	L2			Date de démarrage	S05	
Titre	Etude des technologies utilisées par les OC et de leurs API					
Type	RECH					
Participant	Casanova	Llorens	Sallé	Turchini	Lavirotte	Rey
Effort	61h	61h	61h	61h		

Objectifs du lot

Etudier les différentes technologies utilisées par les objets connectés, et plus spécifiquement par ceux mis à notre disposition, afin de les regrouper en catégories et déterminer des propriétés et fonctionnalités communes.

- Etudier les technologies utilisées par les objets connectés
- En déduire des regroupements par catégories technologiques
- Guider le développement du framework

Description du lot

Tâche 2.1 : Etude des technologies utilisées par les objets connectés (Llorens, S05-S06, 132h)

Il s'agit de l'étude approfondie et la plus exhaustive possible des technologies et protocoles utilisés aujourd'hui par les objets connectés afin de transmettre ou recevoir des informations. Elle s'intéressera plus spécifiquement aux objets mis à notre disposition par nos encadrants et qui nous permettront de tester le framework par la suite.

Tâche 2.2 : Regrouper les objets en catégories technologiques (Casanova, S06-S07, 112h)

L'étude précédente permettra de déterminer des catégories et de regrouper les objets en fonction de leur mode de communication. L'objectif est également de faire ressortir des propriétés ou fonctionnalités communes à différents objets de même catégorie ou non.

Cette étude nous permettra enfin de poser les bases du développement du framework.

Livrable

Livrable D2 : Étude des API et explication des regroupements par technologies (Sallé, S48, DOC)

Ce document décrira en détails l'étude faite au sujet des API utilisées pour les différents objets connectés, ainsi que les regroupements par technologies et fonctionnalités qui en ont été déduits. Enfin, nous y expliciterons les axes de développement du framework qui en découlent.

Identifiant	L3			Date de démarrage	S08	
Titre	Développement du framework					
Type	RECH/IMPL					
Participant	Casanova	Llorens	Sallé	Turchini	Lavirotte	Rey
Effort	66h	66h	98h	98h		

Objectifs du lot

Etudier les différentes technologies utilisées par les objets connectés, et plus spécifiquement par ceux mis à notre disposition, afin de les regrouper en catégories et déterminer des propriétés et fonctionnalités communes.

- Etudier les technologies utilisées par les objets connectés
- En déduire des regroupements par catégories technologiques
- Guider le développement du framework

Description du lot

Les tâches 3.2 et 3.3 font partie du cycle itératif et seront répétées pour chaque nouveau bloc technologique ajouté.

Tâche 3.1 : Conception générale du framework (Casanova, S08-S09, 100h)

Il s'agit de la définition de l'architecture générale du framework ainsi que des technologies utilisées pour les différentes couches, de la sémantique et des conventions liées à l'API, etc.

Tâche 3.2 : Conception de l'ajout d'un bloc technologique au framework (Casanova, S08-S09, 108h)

Nous allons ici concevoir l'architecture contenant l'abstraction de la couche technologique, la liaison avec l'API unifiée, ainsi que les choix de technologies.

Tâche 3.3 : Implémentation du framework (Sallé, S09-S17, 88h)

Nous implémenterons au cours de cette tâche le framework au travers des différents blocs technologiques.

Tâche 3.4 : Tests fonctionnels (Turchini, S11-S17, 32h)

Cette phase de tests doit permettre la vérification du bon fonctionnement du framework.

Livrable

Livrable D3 : Code source du framework (Llorens, LOG, S17)

Identifiant	L4			Date de démarrage	S09	
Titre	Développement du démonstrateur					
Type	IMPL/DEMO					
Participant	Casanova	Llorens	Sallé	Turchini	Lavirotte	Rey
Effort	68h	68h	36h	36h		

Objectifs du lot

Développer le démonstrateur permettant de montrer l'efficacité du framework

- Conception du démonstrateur
- Implémentation du démonstrateur
- Tests fonctionnels
- Retour utilisateurs

Description du lot

Tâche 4.1 : Conception du démonstrateur (Llorens, S09-S10, 56h)

La conception du démonstrateur contient le choix des technologies, le maquetage, les entretiens avec les utilisateurs (membres du groupe ne participant pas directement au développement du démonstrateur) ainsi que l'analyse de ces derniers.

Tâche 4.2 : Implémentation du démonstrateur (Casanova, S13-S18, 80h)

Il s'agit de l'implémentation des maquettes à l'aide de la (ou des) technologie(s) choisie(s).

Tâche 4.3 : Tests fonctionnels (Turchini, S13-S18, 24h)

Il s'agit de la vérification du bon fonctionnement du démonstrateur par le biais de tests d'intégration.

Tâche 4.4 : Retours utilisateurs (Sallé, S13-S18, 48h)

A la suite d'entretiens avec les utilisateurs, cette tâche aura pour but d'analyser les retours qui en sont faits, ainsi que les éventuelles améliorations évoquées par les testeurs.

Livrable

Livrable D4.1 : Rapport de développement (Casanova, S18)

Ce rapport aura pour rôle de décrire la conception qui a été faite du démonstrateur ainsi que son déroulement.

Livrable D4.2 : Code source du démonstrateur (Llorens, S51, S5, S8, S10)

Nous présenterons, aux différents jalons imposés, des versions améliorées du démonstrateur, incluant parfois, lors d'une mise-à-jour, l'intégration d'un nouveau regroupement technologique.

Résumé de l'effort

	Casanova	Llorens	Sallé	Turchini	Lavirotte	Rey
L1 – Management						
T1.1 – Planification	48	48	48	48	5	5
T1.2 – Suivi de projet					36	36
T1.3 - Rapport	63	63	63	63		
T1.4 – Préparation soutenance	10	10	10	10		
Sous-total (h) :	121	121	121	121	41	41
L2 – Etude des technologies des OC et API						
T2.1 – Etude des technologies des OC	33	33	33	33		
T2.2 – Regroupement en familles	28	28	28	28		
Sous-total (h) :	61	61	61	61		
L3 – Développement du framework						
T3.1 – Conception générale	20	20	30	30		
T3.2 – Conception ajout nouveau(x) bloc(s)	24	24	30	30		
T3.3 – Implémentation nouveau(x) bloc(s)	14	14	30	30		
T3.4 – Tests fonctionnels	8	8	8	8		
Sous-total (h) :	66	66	98	98		
L4 – Développement du démonstrateur						
T4.1 – Conception du démonstrateur	18	18	10	10		
T4.2 – Implémentation du démonstrateur	26	26	14	14		
T4.3 – Tests fonctionnels	8	8	4	4		
T4.4 – Retours utilisateurs	16	16	8	8		
Sous-total (h) :	68	68	36	36		
Total (h) :	316	316	316	316	41	41

Tableau 6 : Résumé de l'effort

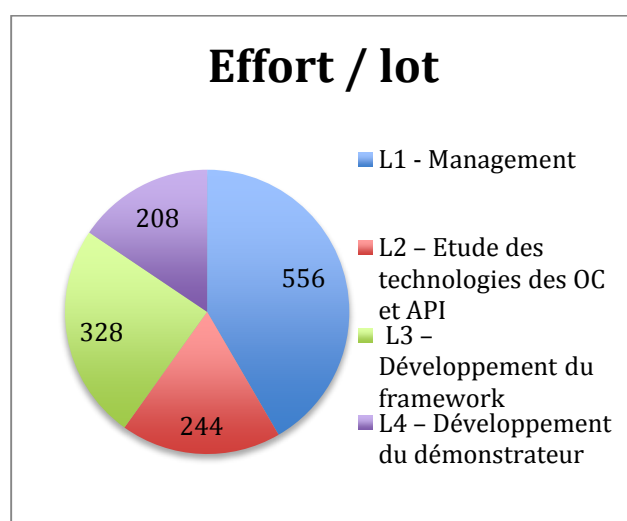


Figure 9 - Répartition de l'effort par lot

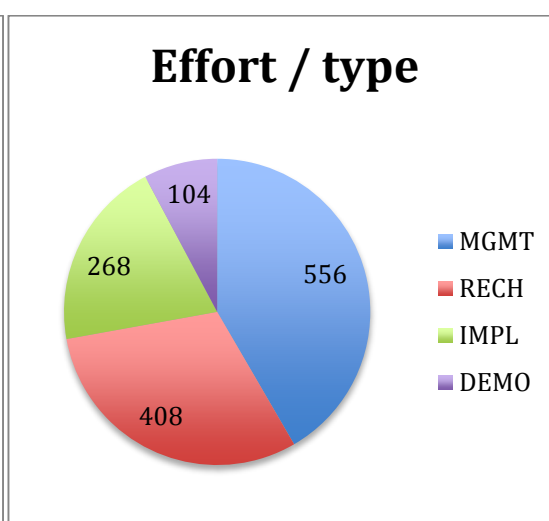


Figure 10 - Répartition de l'effort par type

Gestion du risque

Description	Probabilité	Conséquences	Impact	Cause	Évitement	Résolution
Ne pas parvenir à réunir deux familles technologiques	Faible	Le framework ne prend pas en charge plus d'une technologie	Hétérogénéité gérée au sein d'une seule et unique catégorie technologique	Accès aux objets trop restreint voire impossible (API privée, infos cryptées, etc.)	Prévoir un maximum d'objets utilisant des technologies différentes Etudier de façon approfondie les technologies, les API et les moyens d'accès	Si un objet n'est pas accessible, se rabattre sur un autre objet utilisant la même technologie
Ne pas recevoir les autres objets connectés	Moyenne	Notre étude n'a lieu que sur les trois objets initiaux	Notre démonstrateur met en avant un nombre limité de technologies	Retard des livraisons des commandes	Prévoir au plus tôt les nouvelles commandes à effectuer Suivre les commandes en cours et relancer les fournisseurs	Retirer l'objet de la planification en cas de retard trop important
Impossibilité de développer l'API du framework	Faible	Pas d'interface unifiée à proposer à la couche applicative	L'objectif d'unification du format de données n'est pas atteint	Impossibilité d'abstraire les API spécifiques	Effectuer une étude approfondie des API en amont	Redéfinir les objectifs, en particulier celui concernant le démonstrateur
Ne pas avoir de démonstrateur livrable	Faible	Aucun moyen de prouver le bon fonctionnement du framework	L'objectif de démonstration n'est pas atteint	Retard pris au cours du développement Impossibilité de développer l'API du framework	Cycle agile itératif	Utilisation d'un autre moyen de test, exemple : SoapUI

Tableau 7 : Gestion des risques

5. Participants

Etudiant - Adrien Casanova (SI5 – IHM)

Après avoir effectué le Cycle initial Préparatoire (CiP) à Polytech'Nice-Sophia, j'ai intégré le cycle ingénieur en Sciences Informatiques. Passionné de développement Web et la réflexion autour du besoin utilisateur c'est naturellement que j'ai choisis de suivre le parcours Interaction Homme-Machine (IHM).

Etudiant - Alexia Llorens (Master 2– IHM)

J'ai commencé mes études à l'université de Nice en électronique. A la fin de ma deuxième année j'ai décidé de me réorienter en 3ème année d'informatique. Après l'obtention de ma licence j'ai intégré le master IFI de Sophia Antipolis. Ayant un vif intérêt pour tout le domaine des interfaces mais aussi du besoin utilisateur j'ai opté pour une spécialisation en IHM.

Etudiant - Victor Sallé (SI5 – IAM)

Après l'obtention de mon baccalauréat scientifique, j'ai rejoint l'école Polytech Lille afin d'y effectuer le cycle de préparation aux écoles d'ingénieurs Polytech (PeiP). Ces deux années ont été l'occasion pour moi de découvrir l'informatique, un domaine qui m'a particulièrement captivé et qui m'a par la suite orienté vers le département Sciences Informatiques de l'école Polytech Nice Sophia. Intéressé par le développement mobile et par les perspectives qu'offre la croissance des objets connectés, c'est naturellement que je me suis spécialisé en Informatique Ambiante et Mobile lors de ma cinquième année.

Etudiant - Marie-Catherine Turchini (SI5 – IAM)

J'ai tout d'abord effectué une classe préparatoire intégrée à l'université Pasquale Paoli de Corse en parcours Mathématiques-Informatique, puis j'ai rejoint l'école Polytech Nice-Sophia au sein de la filière Sciences Informatiques. Au cours de ma quatrième année je choisis l'option "Systèmes et Applications réparties". Toujours attirée par les technologies nouvelles et leur expansion dans notre quotidien, sous toutes les formes possibles, c'est logiquement que j'ai opté pour la spécialité "Informatique Ambiante et Mobile" en dernière année.

Encadrant - Gaëtan Rey (Laboratoire I3S - Equipe Rainbow)

Après avoir mené à bien son doctorat en informatique dans le laboratoire CLIPS de l'université de Grenoble sur un modèle de capture du contexte d'interaction dans l'équipe IIHM, Gaëtan Rey a rejoint le System Research Group de l'University College of Dublin pour leur faire bénéficier de son expérience dans le domaine de l'informatique ambiante acquise lors de sa participation aux projets européens Gloss , Fame et Cameleon. Après un passage au LIMSI pour un post-doc sur les interfaces tactiles et son recrutement en tant que maître de conférences à l'Université Nice Sophia Antipolis en 2007, il intègre l'équipe RAINBOW afin d'enrichir l'axe « Context-awareness » dans l'ingénierie logicielle pour l'informatique ambiante et faire le lien entre les parties « systèmes auto-adaptatif » et « adaptation d'IHM » de l'équipe. Dans le cadre de ses activités de recherche, il a notamment participé au projet Continuum sur la continuité de service pour les travailleurs mobiles.

Vivement intéressé par les objets connectés, on les retrouve régulièrement dans les travaux et le site web de Gaëtan Rey, logiquement très intéressé par ce projet, puisque soulevant une problématique actuelle dans ce domaine.

Encadrant - Stéphane Lavirotte (Laboratoire I3S - Equipe Rainbow)

Stéphane Lavirotte est Maître de Conférences à l'Université de Nice Sophia Antipolis. Il est responsable du master IAM (Informatique Ambiante et Mobile). Il est membre de l'équipe de recherche Rainbow du laboratoire I3S. Il mène des recherches dans le domaine de l'informatique ambiante et ubiquitaire. Il est titulaire de la PES (Prime d'Excellence Scientifique) depuis octobre 2009.

Etant responsable d'un parcours particulièrement axé sur les objets connectés et l'informatique embarqué, Stéphane Lavirotte a naturellement été intéressé par un sujet comme celui-ci, élargissant le cadre des cours qu'ils proposent à ses élèves.

Quelques mots désormais sur le laboratoire I3S et en particulier l'équipe Rainbow, au sein de laquelle travaillent nos deux encadrants.

Tout d'abord le Laboratoire I3S (Informatique Signaux et Systèmes de Sophia Antipolis, <http://www.i3s.unice.fr/>) est une Unité Mixte de Recherche (UMR 7271) de près de trois-cents personnes commune à l'Université Nice Sophia Antipolis (UNS) et au Centre National de la Recherche Scientifique (CNRS). Au centre du Campus Sophi@tech, le laboratoire effectue des recherches dans le domaine des sciences de l'information et de la communication.

Quant à l'équipe-projet RAINBOW (de l'équipe GLC), dirigée par Michel Riveill, elle est composée de huit membres permanents enseignants-chercheurs, huit doctorants, et deux ingénieurs de recherche. Elle mène des recherches dans le domaine du génie logiciel pour les applications ubiquitaires. Elle aborde en particulier les domaines des modèles de composition et d'adaptation, la cohérence de l'adaptation à la conception et à l'exécution, les intergiciels pour l'adaptation aux variations du contexte, les architectures des systèmes interactifs et l'intelligence ambiante.

6. Références

IFTTT

[Levée de 30 millions pour relier les objets connectés entre eux](#) (02/11/2014)

[Automatisation des tâches](#) (04/11/2014)

[Exemples concrets](#) (04/11/2014)

Google : le « Physical Web »

[Description](#) (02/11/2014)

[Projet Github](#) (dossier documentation, notamment introduction.md) (02/11/2014)

Withings

[API](#) (22/10/2014)

[Pèse-personne Withings](#) (22/10/2014)

[Tracker d'activité Withings](#) (22/10/2014)

Fitbit

[API](#) (02/10/2014)

Nabaztag

[Nabaztag](#) (22/10/2014)

Protocoles de communication

[Protocoles bas niveau](#) (22/10/2014)

[Protocoles haut niveau](#) (22/10/2014)