EPSI Bordeaux

73 rue de Marseille

33 000 Bordeaux

Net-Sense

7 rue Fénélon

33 000

Bordeaux

Le mobile, assistant au quotidien :

Capacités, limites et devenir

Alexis Léauté

Directeur de Recherche

Sylvain Labasse

Promotion 2013

Remerciements

Je tiens à remercier M. Sylvain Labasse de m'avoir accompagné dans la rédaction de ce mémoire et d'avoir répondu présent lors de mes différentes sollicitations.

Je remercie également le corps enseignant de l'EPSI qui a assuré le côté formation théorique et qui m'a permis d'avoir une bonne base de connaissances pour la réalisation de ce mémoire.

Enfin, je remercie M. Franck Louis Victor de m'avoir accepté en tant qu'alternant au sein de l'entreprise net-Sense. Grâce à mon tuteur M. David Audrain et à toute l'équipe de Bordeaux, j'ai pu approfondir, au cours de cette année, mes connaissances en développements en particulier sur les plateformes mobiles.

Sommaire

Introduction

Aujourd'hui, le téléphone portable occupe une place toujours plus importante dans la vie quotidienne. En effet, d'après l'« Ericsson Mobility Report » de novembre 2012, 61% de la population mondiale, soit 4,1 milliards de personnes, possèdent un téléphone portable dont 1.3 milliard sont des Smartphones*. De plus, selon une étude du site Web « eMarketer » les utilisateurs passe en moyenne 82 minutes par jour sur leurs téléphones en 2012, hors communication, alors qu'ils n'en passaient que 22 minutes en 2009.

Ce qui est intéressant dans ce domaine c'est que l'on travaille sur des plateformes qui sont en constante évolution comme nous le montre l'illustration 1.

Nokia 3410
2002

Nokia 5800
2008

Nokia Lumia 1020
2013

Illustration 1 : Évolution de la gamme de mobiles Nokia de 2002 à 2013

D'une part, du point de vue des fonctionnalités, le mobile permet la démocratisation des dernières technologies telles que la réalité augmentée*, le NFC* ou encore la reconnaissance vocale. D'autre part, d'un point de vue technique, les mobiles possèdent des composants de plus en plus puissants ce qui permet d'exécuter des algorithmes* toujours plus complexes. Tout cela permet de repousser le champ d'applications possible du mobile. De plus, la plupart des plateformes mobiles permettent d'avoir un retour direct et rapide de ses utilisateurs. En effet, via des notes et des commentaires, on peut vérifier l'adéquation entre la demande des utilisateurs et l'orientation que va prendre l'application.

Durant ces trois dernières années, j'ai eu la chance de pouvoir mener mes études en alternance. C'est ainsi que j'ai eu l'occasion de découvrir le monde du mobile lors de ma première année d'ingénierie informatique et dans lequel j'ai continué a évolué depuis. Au cours de ces années, j'ai eu l'occasion de travailler sur différents types d'applications. Tel qu'un assistant d'aide à la conduite : iCoyote, un portefeuille de carte de fidélité numérique : FidMe, un gestionnaire de ticket restaurant : TicketResto, un assistant pour les assurées Axa : Axa Service Mobile ou encore un gestionnaire de cave à vin : Smartcave. Toutes ces applications, bien qu'évoluant dans des domaines totalement différents (la conduite, les vins, le commerce, etc.), ont un objectif commun : celui d'aider l'utilisateur dans ses démarches, ses choix, ses recherches ou son comportement. C'est en faisant ce constat que j'ai décidé d'approfondir, au travers de mon mémoire, la problématique suivante : « Le mobile, assistant au quotidien : Capacités, limites et devenir ».

Les mobiles ne se limitent plus au simple appel téléphonique, mais proposent une multitude de services et de fonctionnalités. En effet, on va pouvoir, par exemple, naviguer sur Internet, consulter sa boite mail, planifier et être avertie de ses rendez-vous ou encore publier sur les réseaux sociaux. Ces téléphones sont appelés « Smartphone », que l'on peut traduire par « téléphone intelligent », faisant référence à l'étendue de leurs capacités. Évidemment, ce sont ces derniers qui vont nous intéresser pour ce mémoire.

Tous ces éléments qui sont désormais basiques sur ce type de téléphone ne représentent qu'une partie de leur utilisation possible. Une des caractéristiques de ces Smartphones, c'est que l'on va pouvoir y installer des applications, tout comme sur son ordinateur, afin d'enrichir les fonctionnalités de son mobile. Ces dernières vont essayer d'exploiter au maximum les services offerts par le téléphone pour proposer une expérience utilisateur la plus adaptée possible. Par exemple, on va se servir de la position GPS récupérée par le téléphone pour ne proposer que les services à proximité. En plus de cela, les applications vont pouvoir accéder à différents services du téléphone tels que la connexion Internet ou encore les différents capteurs présents sur le téléphone comme l'appareil photo, ou le capteur d'orientation*.

Les applications disponibles pour les mobiles sont regroupées en différentes catégories, dont voici les principales :

- Jeux : regroupe tous les jeux vidéo.
- Enseignement : application aidant l'utilisateur à apprendre des choses.
- Livre : application de lecture numérique.
- Divertissement : application gadget, cinéma, programme TV
- Style de vie : application de commerce et de loisirs.
- Économie et entreprise : principalement des applications d'offre d'emploi.
- Utilitaire : outils bureautiques, mail, agenda, etc.

Ces catégories se répartissent de la manière suivante (Schéma 1) :

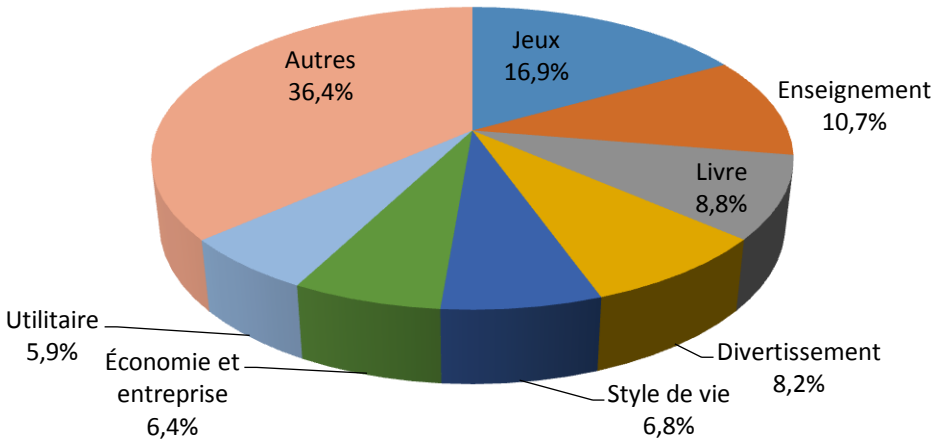


Schéma 2 : Répartition des catégories d'application mobile

Le quartier « Autres » regroupe toutes les catégories ayant un pourcentage moins important.

Les applications que l'on va qualifier d'assistant au quotidien, et qui vont nous intéresser pour ce mémoire, se classeront, pour la plupart, dans les catégories « Divertissement », « Style de vie » et « Utilitaire ». Grâce à ces applications, on va pouvoir se servir de son téléphone comme d'un assistant pour diverses tâches du quotidien.

Grâce à votre mobile, vous allez pouvoir faire vos courses simplement et efficacement. Pour créer votre liste de course, vous pouvez la générer simplement à partir de vos anciennes listes de course, suivant vos achats récurrents ou directement à partir des recettes que vous avez choisi, ou qui vous auront étaient proposées pour avoir des repas équilibrés.

Ensuite, plus besoin d'aller faire vos courses dans les magasins, avec les services « drive » accessibles depuis votre téléphone, vous pouvez commander vos courses, les payer directement depuis votre téléphone et indiquer à quelle heure vous voulez passer les récupérer. De plus, si vous chercher a réalisé des économies vous trouverez des applications vous permettant, à partir du nom de l'article ou de la photo de son code-barres, de comparer son prix dans tous les magasins les plus proches. Et enfin, afin de soulager votre portefeuille de toutes vos cartes de fidélité et être sûr de l'avoir au moment voulu, utiliser une application de virtualisation de vos cartes de crédits et de fidélités dans un portefeuille numérique.

Pour effectuer vos déplacements, vous retrouverez les classiques applications de calcul d'itinéraire et de navigation par GPS. Mais ce n'est pas tout, grâce aux applications d'aide à la conduite vous pourrez indiquer, et être avertie, des zones dangereuses, des bouchons, des accidents et autres perturbations se situant sur votre route. Lorsque vous devez faire le plein d'essence, penser à utiliser les applications de recherche de tarifs de station essence. Ces dernières vous permettront suivant votre position GPS ou une adresse de vous indiquer les tarifs de chaque carburant des stations essences à proximité. Si malheureusement vous assistez à un accident ou en avez un, certaines assurances mettent à disposition des applications afin de vous accompagner dans ce moment. Celles-ci vous permettront de connaitre l'adresse exacte de votre position, les gestes à effectuer ou encore vous faciliteront les démarches de constats et de déclaration d'accidents. Pour les personnes se déplaçant en transport en commun, vous pourrez aussi consulter les itinéraires en indiquant votre destination depuis votre mobile.

Vous avez aussi à disposition tout un ensemble d'application permettant de vous accompagner dans vos loisirs. Si vous voulez sortir dans un restaurant, un bowling ou autres, vous pouvez retrouver tous les points d'intérêt se trouvant à proximité avec des notes et commentaires d'autres utilisateurs. Une autre vous permettra de visionner tous les films projetés dans les cinémas autour de vous et de consulter ses critiques, son résumé, sa bande-annonce ou encore la liste des acteurs pour vous aider à faire votre choix. Pour les sportifs, faites vous accompagner par des applications de coaching.

Ces dernières vous proposeront les exercices les plus adaptés à vos objectifs, elles calculeront les distances parcourues, les calories brulées, la durée de l'effort et même votre rythme cardiaque en branchant un capteur à votre téléphone. Certaines ajoutent aussi un côté communautaire pour publier ses performances et les comparer aux autres. Si vous êtes fan de séries télévisées, utilisez les applications de suivi vous permettant de consulter l'agenda des sorties des épisodes et de lister ceux que vous n'avez pas encore vus.

D'autres applications vous aideront dans la gestion de vos différents contrats et de votre trésorerie. Par exemple, vous pourrez suivre votre consommation téléphonique en détail ainsi que votre hors forfait, la consommation Internet permettant d'éviter des surprises sur les factures. Ces dernières pourront d'ailleurs être téléchargées directement depuis l'application. De la même manière vous pourrez suivre votre contrat EDF, avec le détail de votre contrat et de tarifs, la consultation de votre consommation électrique et des conseils pour économiser de l'énergie. Si votre banque dispose de son application, vous pourrez gérer tout votre compte directement depuis cette dernière, avec consultation de vos comptes, des agences et distributeurs ou encore la possibilité d'effectuer un virement bancaire. Pour gérer votre trésorerie, vous pouvez trouver aussi des applications plus ou moins professionnelles pour gérer vos différents budgets avec un suivi par mois et une notification lorsque vous dépassez un budget.

Il y a aussi à votre disposition un large choix d'outils pour faciliter votre quotidien. On retrouve des applications toutes simples de conversion de devise, de longueur, de poids et de toute unité. Mais aussi des applications de traduction complète, c'est-à-dire, avec un grand choix de langue et la possibilité de traduire du texte taper sur votre clavier, des voix enregistrées avec le microphone du téléphone ou encore du texte présent sur une photo. On va aussi trouver des applications de prise de notes, avec la possibilité d'y joindre une image ou un commentaire vocal et de les partager sur les réseaux sociaux, par mail ou SMS. Vous pourrez aussi suivre la liste des tâches que vous avez à réaliser quotidiennement pour être sûr de ne pas en oublier.

On voit donc que le téléphone portable peut déjà servir d'assistant au quotidien pour de nombreuses tâches. Cependant, on peut se demander si cette technologie est aussi développée que l'on aurait pu l'imaginer.

En effet, si on regarde les œuvres de science-fiction de la fin du siècle dernier, on s'aperçoit qu'à l'époque on s'attendait déjà à ce que la technologie soit beaucoup plus présente dans notre quotidien. Si on fait le parallèle entre le futur imaginé par ces auteurs et ce que l'on a réellement aujourd'hui, on peut constater certaines différences concernant l'utilisation de la technologie au quotidien.

Tout d'abord, on remarque que le plus souvent, les auteurs associent les assistants du quotidien aux robots, comme « 6PO » dans Star Wars. En effet, pour eux, un assistant du futur devait reprendre la forme physique d'un assistant humain tout en bénéficiant de l'intelligence artificielle et de toute la connaissance que peut avoir un robot humanoïde. Seulement, même si la robotique est toujours en développement de nos jours, elle n'en est pas encore au stade d'être disponible au grand public. De plus, du point de vue financier un robot reviendrait surement très cher. C'est pourquoi les assistants numériques du quotidien d'aujourd'hui se développent principalement sur les Smartphones. Ces derniers étant déjà bien implantés sur le marché et ne nécessitant pas d'achat matériel supplémentaire, il parait logique de distribuer ces solutions d'assistants via des applications pour mobile.

Ensuite, ces technologies d'assistant, même si relativement présentes, sont beaucoup moins utilisées que ce que l'on avait imaginé. On peut prendre pour exemple les applications de reconnaissance vocale, dont on a beaucoup entendu parler ces dernières années, notamment avec la sortie de l'assistant vocal sur iPhone : Siri. Pour rappel, grâce à ce dernier, on peut, par exemple, demander la météo d'aujourd'hui, appeler un de nos contacts ou encore planifier un rendez-vous dans notre agenda électronique, et tout cela en parlant directement à notre mobile. Seulement si on regarde l'utilisation qui en est faite, on remarque que celui-ci sert plus de gadget lors des premiers jours d'utilisation que de réel outil d'assistance au quotidien. Cependant, dans les films de science-fiction, la reconnaissance vocale occupe une place beaucoup plus importante. Ils en ont une multitude d'utilisations, les plus basiques étant le contrôle de tous les appareils, ordinateur et véhicule par la voix. Cela leur permet, par exemple, de choisir leur destination lorsqu'ils montent dans leur voiture ou de contrôler entièrement leur maison ou leur ordinateur. Plus besoin de clavier, souris ou d'interrupteur.

Un autre exemple est la reconnaissance faciale qui est beaucoup utilisée dans la science-fiction. Elle est utilisée par exemple pour effectuer un paiement où la technologie associe le compte bancaire au visage du personnage, pour démarrer un véhicule ou déverrouiller des portes aux accès limités. Cependant dans la réalité, cette technologie souffre encore d'imprécisions. Elle est donc utilisée dans des situations moins critiques que le paiement ou l'automobile. On retrouve cette technologie principalement pour déverrouiller son ordinateur ou son téléphone portable.

D'autre part, on remarque que, dans ces ouvrages, la technologie est réellement présente dans la vie quotidienne des personnages. Elle est beaucoup plus développée et elle sert vraiment pour toutes les activités du quotidien. En restant sur la technologie de la reconnaissance vocale, on peut voir dans Start Trek ou dans l'Âge de diamant de Neal Stephenson qu'ils améliorent ce dernier afin de créer un traducteur universel instantané. Cela permet ainsi aux différents personnages de communiquer entre eux simplement et efficacement. Il est vrai que l'on trouve aujourd'hui des applications qui permettent de traduire nos paroles. Seulement, ces dernières ne sont pas instantanées. Elles ont besoin de temps pour analyser l'enregistrement et ne peuvent pas détecter automatiquement le langage utilisé. On retrouve aussi souvent comme assistant au quotidien une intelligence artificielle qui va suivre le personnage tout au long de l'œuvre. Elle se retrouvera dans sa maison, sa voiture et tout le temps sur lui dans un terminal portable que l'on peut comparer à un téléphone portable. Cette intelligence peut se manifester de différentes façons, dans un robot humanoïde ou non, un hologramme ou même juste une voix. Elle agit comme un assistant humain, comprend vos phrases, vos questions et peut y répondre. Cette dernière regroupe toutes les fonctionnalités des divers assistants. Les personnages peuvent lui demander la météo, l'itinéraire pour se rendre à une adresse ou les nouvelles de la journée. Bien sûr, elle rappelle à son propriétaire ces rendez-vous et taches de la journée à effectuer et peut contrôler toute votre maison et votre voiture.

Ce mémoire a pour but, d'une part, de montrer le potentiel du mobile en tant qu'assistant au quotidien et les freins à ce développement. D'autre part, il proposera des solutions possibles pour pallier à ces problématiques.

Pour cela, après cette introduction, je vais d'abord faire le point sur les aptitudes d'un assistant mobile. Pour plus de compréhension, je regrouperai ces aptitudes par thème et j'y apporterai des détails techniques sur leurs fonctionnements ainsi que leurs architectures. Ensuite, j'expliquerai les limites et freins des technologies mobiles d'assistant au quotidien. Suite à cela, je proposerai des solutions possibles pour pallier à ces limites et freins. Dans cette partie, je détaillerai les architectures et expliquerai les changements à apporter aux solutions existantes. Je finirai par une conclusion qui va d'abord résumer ce mémoire en y montrant les apports et solutions de ce dernier. Elle nuancera ensuite ce mémoire avec les risques liés à ces technologies d'assistant au quotidien.

I. Capacités d'un assistant mobile

A. Systèmes de reconnaissance

1. Reconnaissance vocale

a. Fonctionnement

Même si chaque système de reconnaissance utilise ses propres algorithmes*, ils utilisent tous le même mode de fonctionnement (Schéma 2).

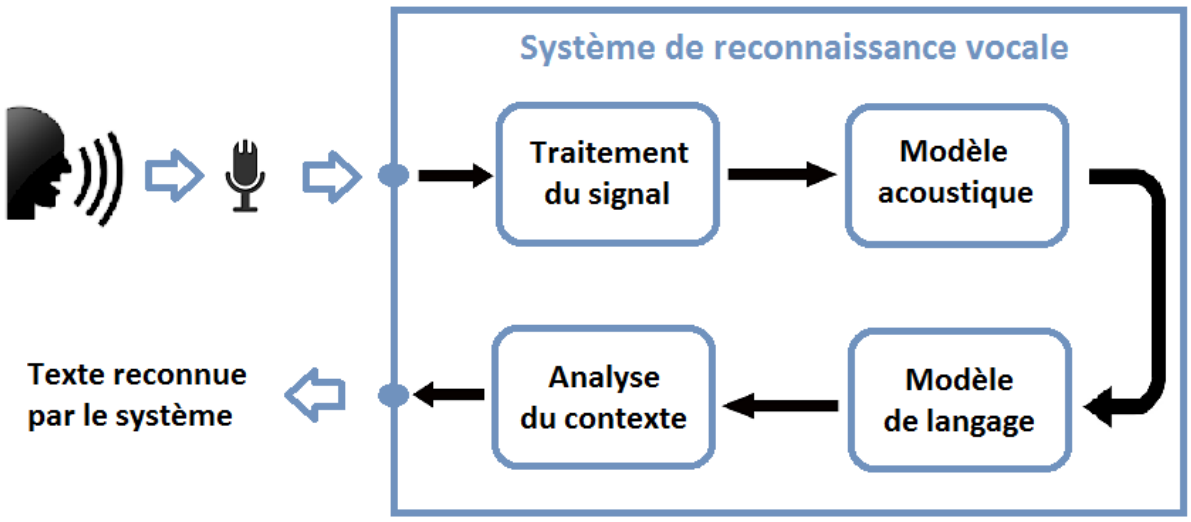


Schéma 2 : Fonctionnement d'un système de reconnaissance vocale

Le traitement du signal va permettre, à partir des données brutes du microphone, d'isoler la voix et de supprimer les bruits inutiles. En plus de cela, il va extraire les paramètres de la voix, tels que sa mélodie, son rythme et son intensité. Ce sont ces données qui vont être utilisées pour les prochaines étapes.

Le modèle acoustique analyse les paramètres de la voix pour le traduire en phonèmes. Le phonème est l'unité distinctive de prononciation dans une langue. Pour que deux sons soient des phonèmes, il faut que, si on substitue l'un par l'autre, cela entraine un changement de sens, par exemple : dans chat et dans rat, les sons « ch » et « r » sont des phonèmes. En français on retrouve 37 phonèmes.

Le composant suivant des systèmes de reconnaissance vocale est le modèle de langage. Ce dernier va analyser le contenu de votre diction. Il compare la combinaison des phonèmes, traduits par le modèle acoustique, aux mots contenus dans son dictionnaire digital. Ce dictionnaire est une grande base de données contenant les mots les plus utilisés dans le langage voulue. La plupart des dictionnaires accompagnant les systèmes de reconnaissance vocale contiennent plus de 150 000 mots pour chaque langue. En théorie, le modèle de langage va donc rapidement choisir quels mots vous avez énoncés.

Seulement, cela n'est pas aussi simple. En effet, certains mots ont la même prononciation, mais pas la même écriture ni la même signification. On peut prendre comme exemple : « ver », « vers », « verre » et « vert ». Afin de résoudre ce problème, l'application de reconnaissance vocale va analyser le contexte dans lequel est utilisé le mot. Dans la plupart des cas, ils vont pouvoir reconnaitre un mot en regardant les deux mots le précédant. Par exemple, si vous dites : « Il marche vers… », grâce aux mots « il marche » le système va pouvoir choisir le mot « vers » au lieu de « verre ».

Afin d'améliorer son efficacité, le système de reconnaissance vocale va pouvoir s'adapter à l'utilisateur lui-même, à sa façon de parler, son accent. Pour cela, l'application va faire lire un texte à l'utilisateur pour déterminer les paramètres à utiliser. De plus, elle va lui permettre d'ajouter ses propres mots au dictionnaire du modèle de langage.

b. Application

Il existe une multitude d'applications possibles pour la reconnaissance vocale. Cependant, on peut distinguer deux types d'utilisations principales.

Tout d'abord la reconnaissance de commande vocale. Ici, on va chercher à exécuter des actions suivant des mots clés prononcés par l'utilisateur. L'application aura une liste de mots ou de suites de mots qui correspondra à une action. Dans cette utilisation, on retrouve peu d'erreurs d'interprétation du logiciel, car son dictionnaire de mot ne contiendra que ceux compris dans ses commandes, au lieu des 150 000 mots d'un dictionnaire complet. Sur la plupart des Smartphones du marché, cet outil est intégré. Il permettra d'appeler un contact, de préparer un SMS ou encore de lancer une application.

Le système de reconnaissance vocale intègrera donc au fur et à mesure les contacts et applications de l'utilisateur à son dictionnaire.

Une autre application est la diction. La diction va permettre à l'utilisateur de saisir du texte directement en parlant sans utiliser de clavier. C'est l'utilisation la plus complexe, le système de reconnaissance vocale va devoir interpréter mot pour mot ce qui a était dit pour le retranscrire à l'écran. Cette technologie se retrouve aussi intégrée dans une grande partie des Smartphones. Ses utilisations les plus courantes sont pour écrire un SMS, effectuer une recherche ou encore prendre des notes.

Ces deux utilisations vont aussi pouvoir être combinées entre elles afin de compléter une commande vocale. La commande sera séparée en deux parties, la première sera la commande elle-même, et la deuxième partie sera un paramètre de la commande. L'exemple le plus simple est pour la rédaction de SMS. On va pouvoir dire : « Dit à Jean que je suis bien arrivé ». Le téléphone créera donc un SMS destiné à Jean avec pour texte « Je suis bien arrivée ».

Comme je le disais dans l'introduction, on entend beaucoup parler de la reconnaissance vocale sur mobile depuis l'arrivée de Siri sur l'iPhone. La particularité de ce système de reconnaissance vocale et qu'il est combiné à une intelligence artificielle qui va comprendre ce que vous dites. Il va se décomposer en trois parties, la reconnaissance vocale, la compréhension qui va interpréter le sens de la phrase et l'action qui pourra être de fournir une réponse ou d'exécuter une commande. Cela permet de faire de ce système un assistant personnel intelligent : il pourra répondre à vos questions, vous donner l'heure ou la météo et cela simplement en le lui demandant.

c. Architectures

Différentes architectures sont utilisées pour effectuer de la reconnaissance vocale sur mobile. On en distingue trois :

- Système embarqué dans le mobile
- Système déporté sur un serveur
- Système distribué entre le mobile et le serveur

i) Système embarqué

Dans cette architecture, comme son nom l'indique, tout le système de reconnaissance vocale — le traitement du signal, le modèle acoustique, le modèle de langage et l'analyse du contexte — est embarqué dans le téléphone portable (Schéma 3).

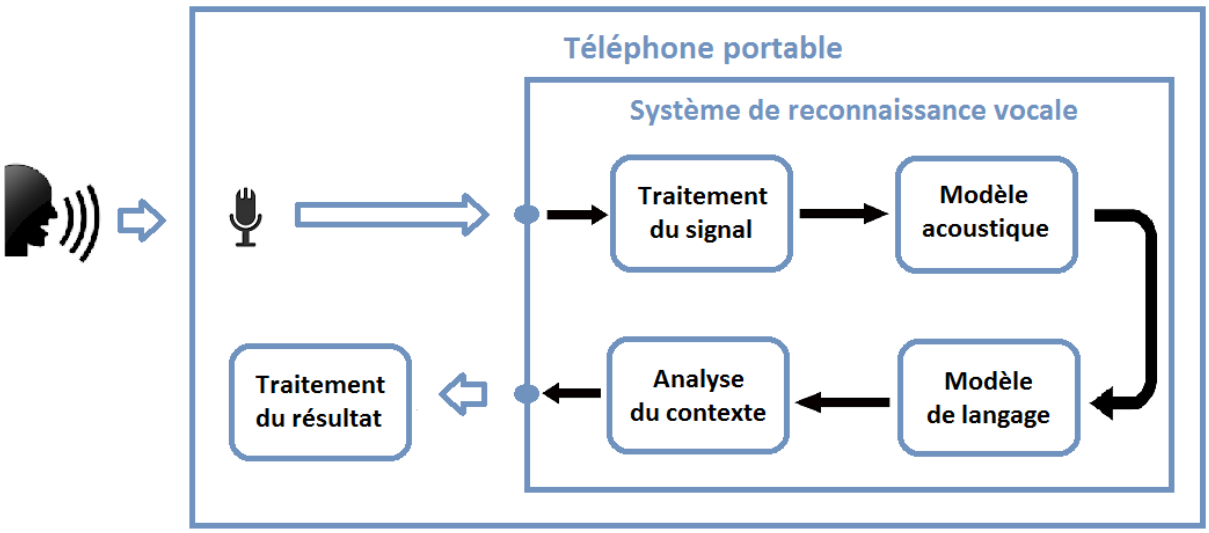


Schéma 3 : Architecture Système de reconnaissance vocale embarquée

L'avantage principal de cette architecture est qu'aucune communication avec un serveur n'est nécessaire pour son bon fonctionnement. Le système est donc accessible immédiatement et le temps de réponse ne dépend pas de la qualité du réseau.

Cependant, il existe aussi un inconvénient à cette architecture. Le système va dépendre des ressources limitées du téléphone. Les deux contraintes principales seront la mémoire vive* utilisée et le temps d'exécution du système. Mais il faudra aussi penser à la taille des dictionnaires utilisés pour chaque langue qui devront être stockés sur l'appareil. Il faudra donc optimiser ces points si on veut utiliser cette architecture et peut-être penser à enlever certaines fonctionnalités et/ou perdre en précision de reconnaissance.

Cette contrainte peut être relativisée aujourd'hui, car les mobiles sont de plus en plus puissants. En effet, les derniers Smartphones sortis sont équivalant en termes de performance à certains ordinateurs portables. Il faut évidemment continuer à travailler sur ce problème de performance, car d'une part tout le monde ne possède pas le dernier téléphone à la pointe et d'autre part une réactivité importante ne peut être que meilleur pour l'expérience utilisateur.

ii) Système déporté sur serveur

Contrairement à l'architecture précédente, ici tout le système de reconnaissance vocale est déporté sur un ou des serveurs. Le mobile va servir uniquement pour enregistrer la voix avec son microphone et le transmettre directement au serveur (Schéma 4).

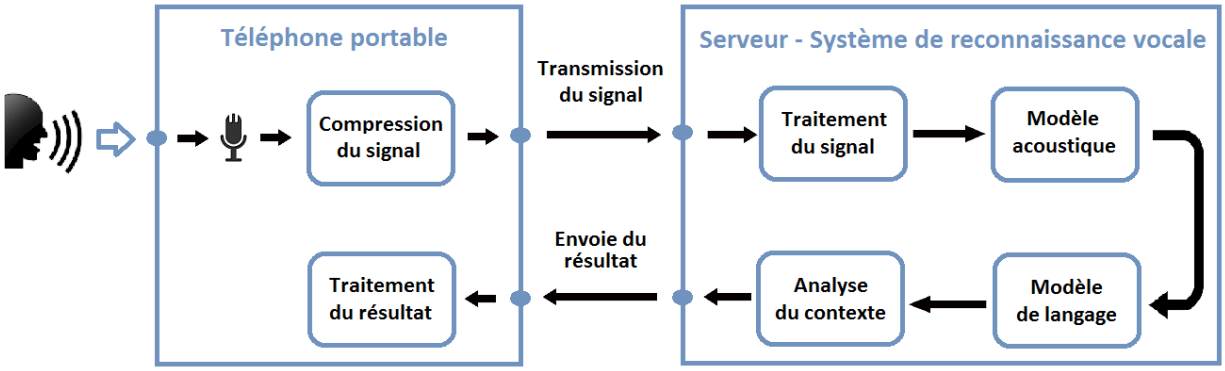


Schéma 4 : Architecture Système de reconnaissance vocale déportée sur serveur

Ce système va résoudre tous les problèmes liés aux ressources limitées des téléphones portables. Il va permettre à n'importe quel appareil disposant d'un microphone et d'une connexion Internet d'effectuer de la reconnaissance vocale. Avec la puissance offerte par les serveurs, on va pouvoir utiliser un système performant et complet avec autant de vocabulaire et de langage que nécessaire.

Néanmoins, cette architecture possède des inconvénients dus à la nécessité de devoir communiquer avec un serveur. Tout d'abord, l'appareil devra disposer d'une connexion Internet à chaque fois qu'il voudra utiliser la reconnaissance vocale. Cette dernière doit être de bonne qualité pour fournir un temps de réponse correcte. Ensuite, le signal reçu du microphone va devoir être compressé* pour pouvoir être envoyé au serveur. Cette compression* entraine forcement une perte de précision du signal qui diminuera la performance du système. De plus, l'architecture côté serveur devra être assez solide pour répondre aux demandes simultanées de tous les utilisateurs.

iii) Système distribué

Dans cette architecture, le système de reconnaissance vocale est réparti sur le mobile et sur le serveur. Le téléphone ne va plus se contenter de transmettre le signal du microphone, il va aussi effectuer la phase de traitement du signal et surtout d'extraction des paramètres de la voix (Schéma 5).

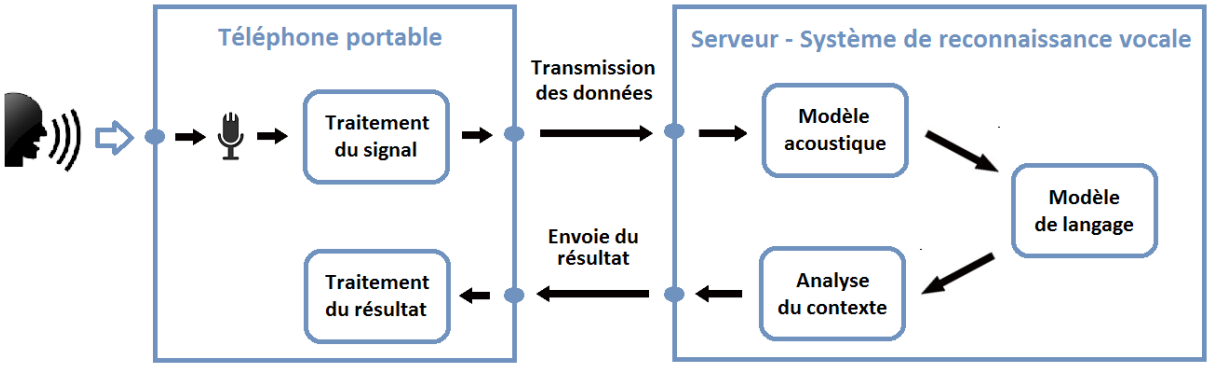


Schéma 5 : Architecture Système de reconnaissance vocale distribuée

Cette architecture reprend tous les avantages d'un système déporté sur serveur, donc principalement un système de reconnaissance vocale puissant et complet accessible au plus grand nombre. Mais il permet aussi de pallier à un de ces problèmes, la perte de précision lors de la transmission. En effet, les données extraites après la phase de traitement du signal sont beaucoup plus légères que le signal lui-même. Ces données n'auront donc pas besoin d'être compressées. De plus, la phase de traitement du signal ne nécessite pas un appareil particulièrement performant, cette tâche est donc parfaite pour être exécutée sur le mobile.

Côté inconvénient, il reprend aussi ceux d'une architecture déportée sur serveur, hormis celui énoncé précédemment.

2. Reconnaissance d'image

a. Fonctionnement

Les étapes principales d'un système de reconnaissance d'image sont : la détection des caractéristiques, l'extraction de caractéristiques, la sélection des caractéristiques et la classification (Schéma 6).

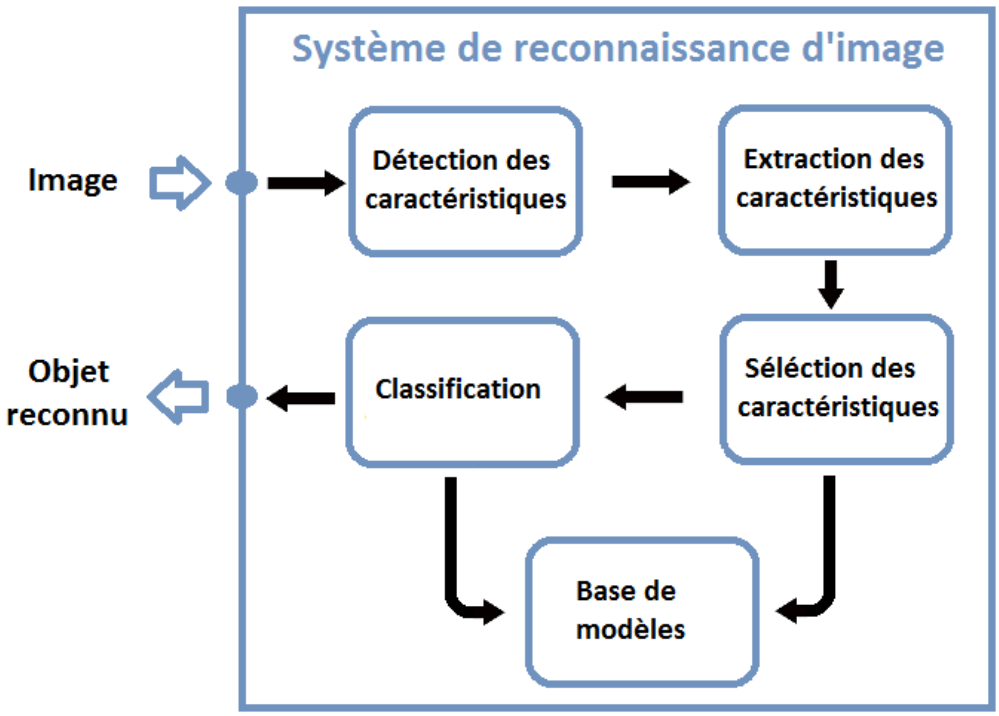


Schéma 6 : fonctionnement d'un système de reconnaissance d'image

La base de modèle est une base de données contenant tous les modèles d'objet connu par le système. Les informations contenues dans cette base varient en fonction de l'approche utilisée par le système. Cela peut aller de la simple description fonctionnelle jusqu'à des informations géométriques précises de l'objet. Généralement, les modèles d'objet sont des tableaux de caractéristique. Une caractéristique correspond aux attributs d'un objet considéré comme important pour décrire et reconnaitre l'objet par rapport à un autre. La taille, la couleur, et la forme sont des caractéristiques couramment utilisées. La base de modèle est organisée de manière à faciliter l'élimination des objets improbable de ceux qui devront être vérifiés.

La détection des caractéristiques va consister à localiser sur l'image les caractéristiques et les zones « clés » qui seront intéressantes pour la reconnaissance. Pour réaliser cette phase, des algorithmes* de traitement d'image sont utilisés comme les algorithmes* de détection de contour ou d'arêtes.

Une fois que les caractéristiques et les zones intéressantes sont localisées, l'extraction des caractéristiques va permettre de récupérer et de décrire ces dernières afin qu'elles puissent être comparées et analysées. Les caractéristiques utilisées par le système vont dépendre du type d'objet à reconnaitre et de l'organisation de la base de modèle. Cette étape est la plus complexe et la plus importante du système reconnaissance. En effet, c'est ici que les caractéristiques des objets présents dans l'image vont être extraites et c'est grâce à ces caractéristiques que le système va pouvoir reconnaitre l'objet. Voici un exemple de description d'un marteau utilisant l'angle de ses arêtes et leur position respective comme caractéristique (Illustration 1) :

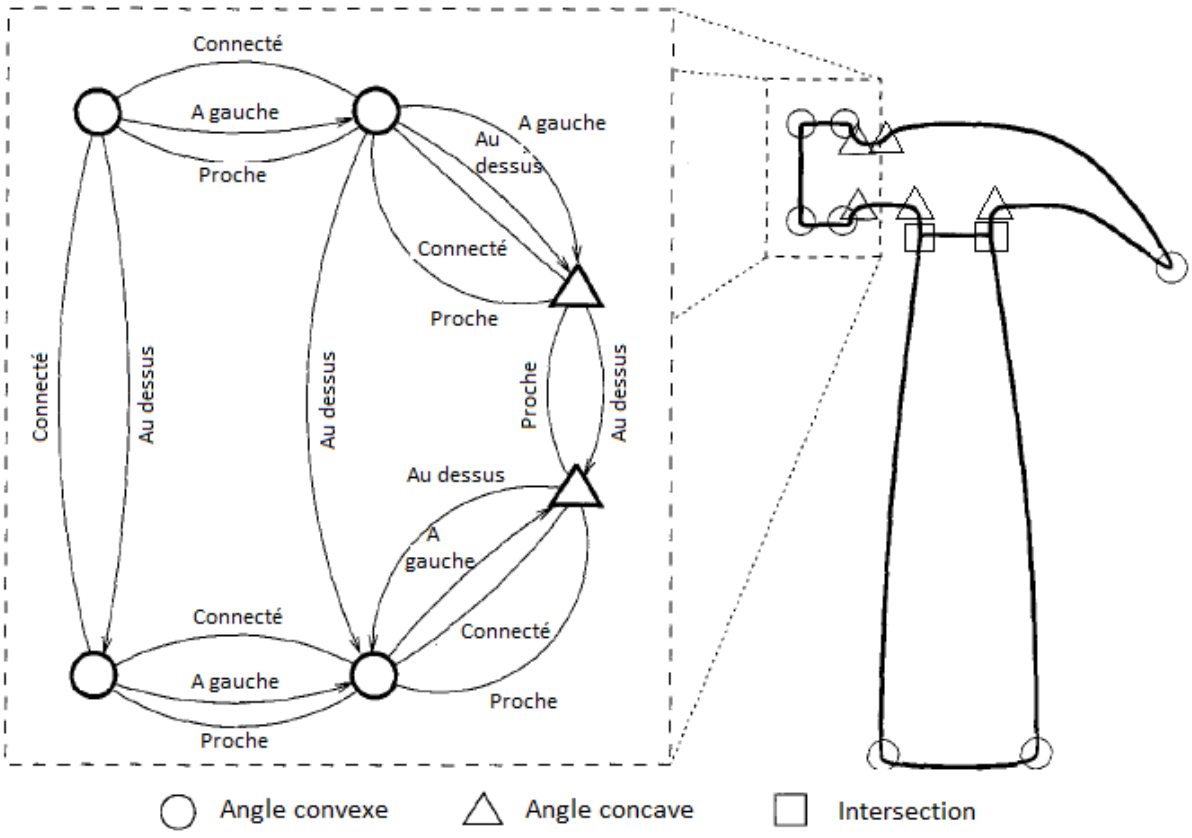


Illustration 1 : Exemple de description d'un marteau avec ces caractéristiques

La sélection des caractéristiques va ensuite éliminer de la base de modèles, de manière grossière (sans se focaliser sur les détails), les objets qui ne correspondent pas du tout aux caractéristiques extraites. Cela permet de réduire le champ de recherche de la classification en se basant sur certaines caractéristiques simples.

Enfin, la classification compare chacune des caractéristiques extraites avec les caractéristiques de chaque objet présent dans la base de modèles. Il va ensuite assigner des probabilités pour chaque objet et sélectionner l'objet avec le plus de probabilité comme étant l'objet se trouvant sur l'image.

Tous les systèmes de reconnaissance d'image utilisent une base de modèle, la détection et l'extraction des caractéristiques. Cependant, l'importance de la sélection des caractéristiques et de la classification varie suivant l'approche du système. Certains systèmes utilisent seulement la sélection des caractéristiques et sélectionnent directement l'objet avec la plus grande probabilité. Alors que d'autres systèmes, quant à eux, privilégient l'utilisation de la classification avec peu ou pas de sélection de caractéristiques.

b. Application

On peut distinguer plusieurs types d'application d'un système de reconnaissance d'image pour servir d'assistant. Ces applications vont utiliser deux types principaux de reconnaissance d'image, la reconnaissance faciale et la reconnaissance d'objet. La reconnaissance faciale va donc permettre de reconnaitre un visage et de pouvoir l'identifier. Alors que la reconnaissance d'image va permettre d'identifier, plus ou moins précisément, un objet sur une image.

Plutôt que l'utilisation classique de la reconnaissance faciale pour déverrouiller son téléphone ou autre, ce qui va nous intéresser c'est d'identifier une ou plusieurs personnes sur une série d'images. Grâce à cela, l'application va pouvoir savoir qui se trouve sur la photo et ainsi proposer différentes fonctionnalités. Elle va pouvoir classer vos photos suivant la personne ou le groupe de personnes se trouvant dessus ou encore vous permettre de rechercher des photos en saisissant le nom de la personne souhaité.

En poussant encore plus loin la reconnaissance faciale, on peut aussi détecter l'expression de la personne se trouvant sur la photo. Cela peut servir à différentes choses, comme sélectionner les meilleurs clichés en se basant sur le sourire des personnes ou encore classer et recherche les personnes dans sa galerie de photo suivant son expression.

Concernant la reconnaissance d'objet, on va pouvoir retrouver les mêmes fonctionnalités qu'avec la reconnaissance faciale. C'est-à-dire, le classement et la recherche de photo dans sa galerie suivant les objets présents dessus. Mais ce qui est plus intéressant, c'est de prendre un produit commercial en photo, comme une chaussure ou un meuble, et de pouvoir retrouver, en plus de la référence et de la description de l'objet, la possibilité d'acheter directement le produit ou de lister les magasins proposant ce produit. Grâce à cela, vous pourrez, dès l'instant où vous voyez un produit qui vous intéresse, récupérer la référence de ce produit pour l'acheter plus tard ou le commander directement.

Une autre utilisation de la reconnaissance d'objet est de pouvoir identifier les bâtiments que vous prenez photo. Évidemment ce ne sont pas n'importe quels bâtiments qui seront reconnus, mais plutôt les lieux touristiques, les monuments culturels ou encore les bâtiments historiques. Cela permettra à l'utilisateur, en prenant en photo le bâtiment désiré, de retrouver toutes les informations relatives à ce dernier, comme son nom, son histoire ou encore sa particularité.

c. Architectures

Comme pour les systèmes de reconnaissance vocale, on va distinguer les trois mêmes architectures pour la reconnaissance d'images :

- Système embarqué dans le mobile
- Système déporté sur un serveur
- Système distribué entre le mobile et le serveur

Ces architectures étant déjà détaillées dans la partie Reconnaissance vocale, je vais juste vous expliquer les spécificités dues à la reconnaissance d'image et vous présentez les schémas des architectures.

Avec un système de reconnaissance d'image embarqué dans le mobile, tous les composants se retrouvent sur le Smartphone (Schéma 7). Cela permet de ne pas dépendre du temps de communication avec un serveur, mais pose le problème des ressources limitées d'un téléphone. En effet, les algorithmes* de détection et surtout d'extraction vont nécessiter beaucoup de puissance processeur* pour avoir un temps de réponse raisonnable. De plus la base de modèle se trouvant embarqué sur le mobile, il faudra faire attention à la taille de cette dernière.

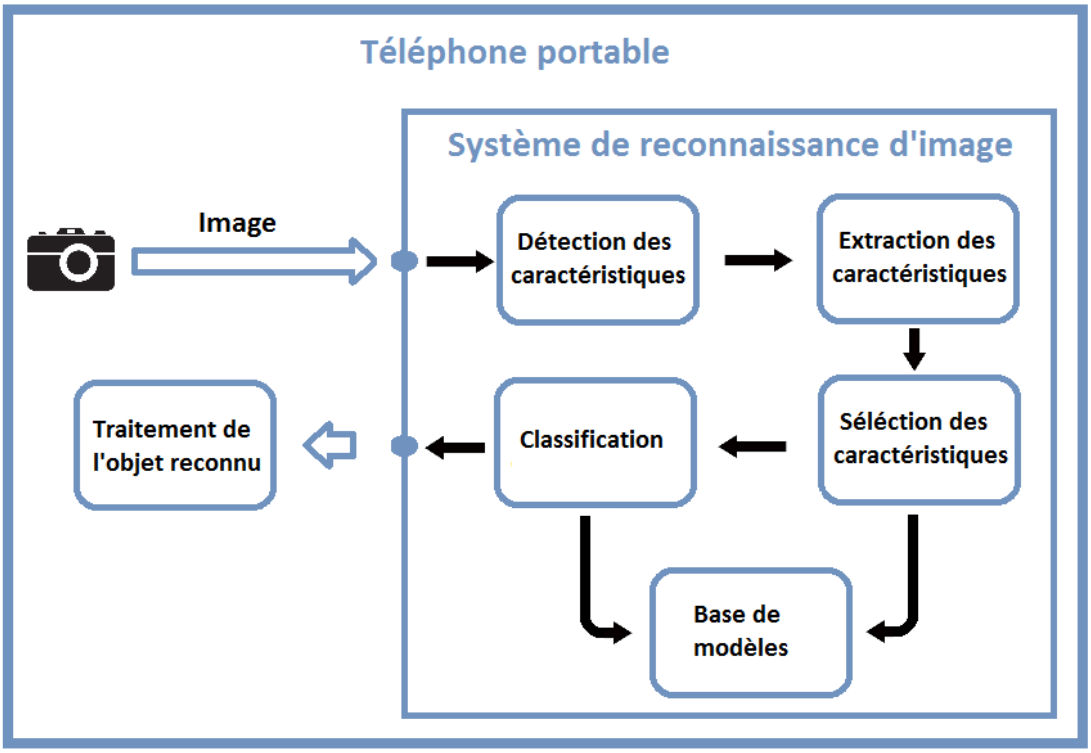


Schéma 7 : Architecture système de reconnaissance d'image embarquée

L'architecture déportée sur serveur permet donc de pallier au problème des ressources limitées du téléphone, mais engendre des problèmes du a la communication avec le serveur (Schéma 8). Principalement le temps de réponse, car, plus l'image sera détaillée, plus sa taille sera grande et plus la communication avec le serveur sera longue. Cependant, utilisant des serveurs relativement puissants permets d'avoir un système de reconnaissance performant et précis.

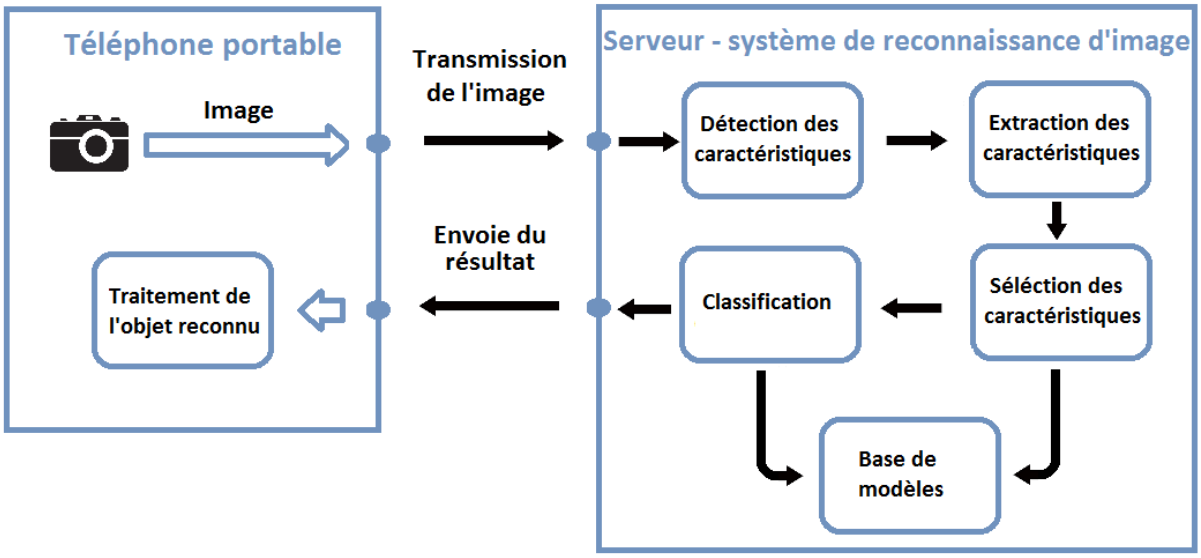


Schéma 8 : Architecture système de reconnaissance d'image déportée sur serveur

Enfin, les systèmes utilisant une architecture distribuée entre le serveur et le mobile permettent de réduire le temps de réponse due à la communication au serveur, en envoyant uniquement les caractéristiques extraites qui sont plus légères que l'image elles-mêmes (Schéma 9). La détection et l'extraction des caractéristiques se retrouvent donc sur le mobile et tout le reste du système sur le serveur. Il faut noter que le mobile ne permettra pas d'exécuter des algorithmes* de détection et d'extraction aussi poussés que sur un serveur, le système perdra donc en précision.

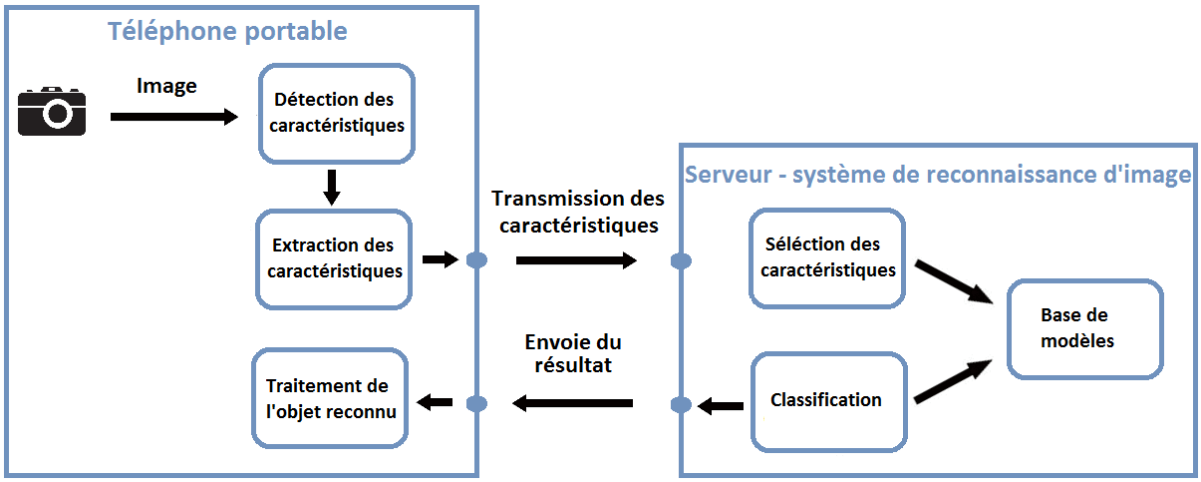


Schéma 9 : Architecture système de reconnaissance d'image distribuée

Le choix entre chacune de ces architectures dépendra des contraintes de temps de réponse, de précision du système et du nombre d'objets que le système pourra reconnaitre. Plus on voudra un système précis, plus il faudra tendre vers une architecture serveur. Et si le nombre d'objets à reconnaitre n'est pas trop élevé et donc que le système n'a pas besoin d'une grande précision, une architecture embarquée sera la meilleure solution. On peut effectuer un compromis avec une architecture déportée qui permet d'utiliser une base de modèle aussi grande que souhaité, mais en se limitant à la précision offerte par une détection et une extraction de caractéristiques effectuée sur mobile.

B. Situation dans l'espace

1. Les caractéristiques d'une position

Une position dans l'espace peut se caractériser par trois données. La géolocalisation qui correspond aux coordonnées latitude, longitude et altitude du point. La direction de ce point, vers où il se déplace, exprimé en degrés par rapport au Nord. Et son orientation par rapport à la Terre et au Nord (Illustration 2).

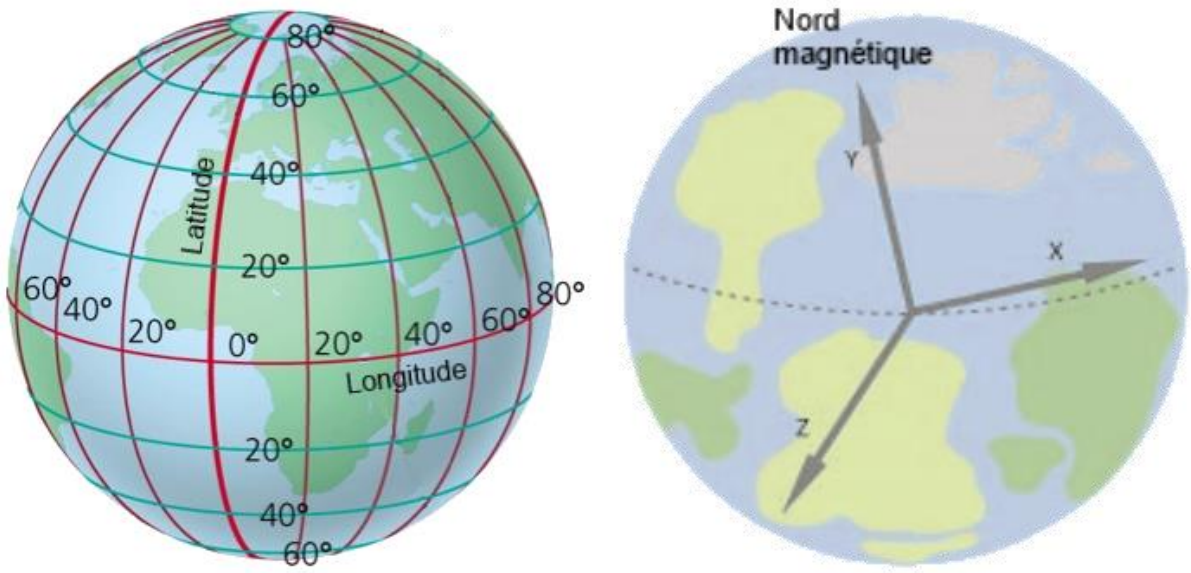


Illustration 3 : Repère de la géolocalisation (à gauche) et de l'orientation (à droite)

a. Géolocalisation et direction

Il existe différent moyen pour connaitre sa position sur Terre. Sur un téléphone portable, on retrouve trois technologies le permettant :

- Le GPS
- Le positionnement par WiFi
- Le positionnement par réseau téléphonique

Pour la direction, elle peut être obtenue très simplement par le récepteur GPS. Ce dernier a juste à calculer, grâce à la trigonométrie, l'angle entre la droite formé par les deux dernières positions calculées et le Nord.

i) GPS

Le GPS pour « Global Positionning System » que l'on peut traduire par « Système de localisation mondial » est le plus connue des trois. Il se base sur 27 satellites tournant en orbites autour de la Terre à 20 200 kilomètres, dont 3 servent de remplaçant si un des 24 autres à un problème. Ces satellites envoient à fréquence régulière un message par radio contenant l'heure exacte, grâce à une horloge atomique, à laquelle ce dernier a été envoyé.

Pour déterminer une position valide, le récepteur GPS, qui va recevoir les messages des satellites GPS se trouvant au-dessus de lui, a besoin d'au moins trois satellites pour avoir les coordonnées en 2D, x et y, et de quatre pour avoir l'altitude. La position de chaque satellite est d'ailleurs calculée afin qu'à tout moment à n'importe quel endroit de la planète un récepteur puisse capter quatre satellites. Il faudra ensuite qu'il connaisse la position des satellites lui envoyant les messages ainsi que la distance entre ces derniers et lui-même.

Pour cela, il compare l'heure à laquelle le message a était envoyé et à laquelle il a était reçu pour ainsi obtenir le temps de transmission du message. Les messages étant envoyés à vitesse constante, la vitesse de la lumière, il peut ainsi déterminer sa distance avec le satellite. Pour connaitre la position des satellites, le récepteur dispose d'une table avec les positions de chaque satellite pour une durée de quatre heures qui est mises à jour grâce aux informations contenues dans les messages envoyés par les satellites. Une fois qu'il dispose de ces données, il utilise la triangulation pour déterminer sa position.

On peut se représenter cette méthode en imaginant que chaque satellite est entouré d'une sphère dont le rayon est égal à la distance calculée par le récepteur. Une fois que l'on capte trois satellites, on remarque que ces sphères ont un seul point d'intersection sur Terre, c'est votre position.

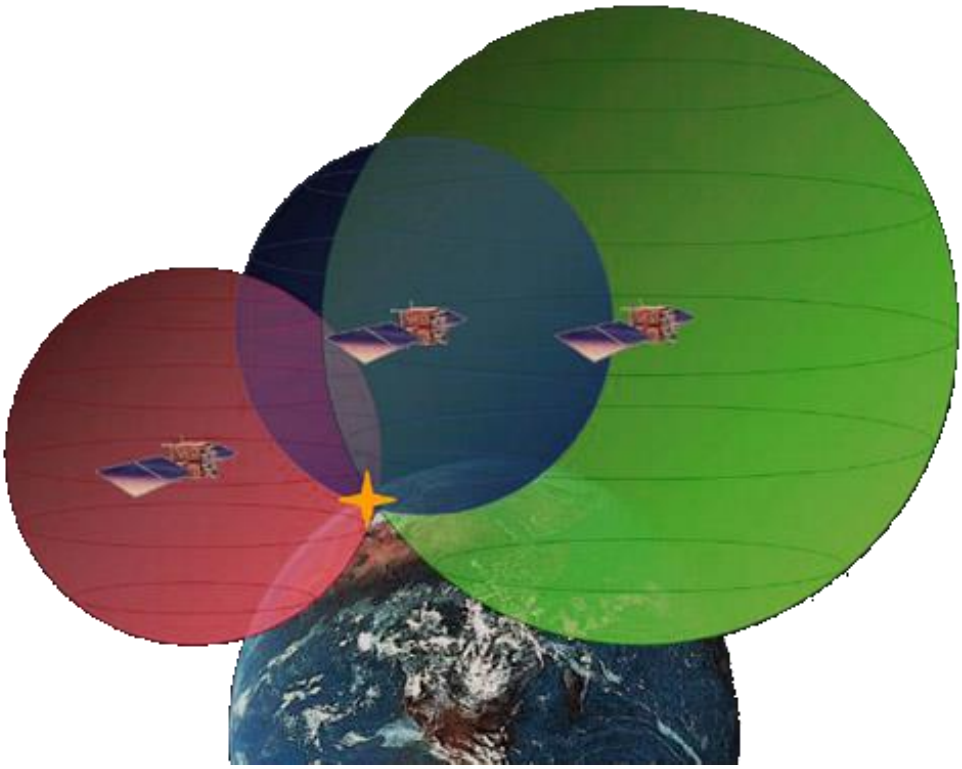


Illustration 4 : Triangulation GPS

Le temps de la première acquisition GPS est long, entre une et deux minutes. Cela est dû au téléchargement de la table de la position des satellites. Pour pallier à cela, une technique appelée A-GPS pour « Assisted-GPS » permet le téléchargement de cette table via le réseau téléphonique qui est beaucoup plus rapide et qui permet de télécharge une table valable pour sept jours au lieu des heures avec un GPS standard.

ii) Positionnement par WiFi et par réseau téléphonique

Le positionnement par WiFi et par réseau téléphonique fonctionne globalement de la même manière que le système GPS. Seulement, il se base sur la position des bornes WiFi et des antennes téléphoniques pour effectuer la triangulation et calculer votre position. De plus, pour calculer sa distance avec la borne ou l'antenne il va utiliser la puissance du signal.

Même si ces techniques sont plus rapides, elles sont aussi moins précises et nécessitent d'être dans une zone relativement urbaine pour pouvoir capter assez de borne WiFi ou d'antenne téléphonique.

b. Orientation

Pour récupérer l'orientation du téléphone, le système va se baser sur deux capteurs : le magnétomètre et l'accéléromètre. Grâce à ces derniers, on va pouvoir mesurer l'orientation en trois dimensions du téléphone. Cette orientation sera représentée par trois angles dont les noms font référence à l'aviation :

- Azimut : direction du téléphone par rapport au Nord
- Roulis : inclinaison horizontale (de gauche à droite)
- Tangage : inclinaison verticale (d'avant en arrière)

Illustration 5 : Angles d'orientation du téléphone

Le magnétomètre mesure la direction du champ magnétique terrestre en trois dimensions. Grâce à cela, on peut connaitre l'orientation du téléphone par rapport à ce champ magnétique et donc par rapport au Nord. L'angle formé entre la direction du champ magnétique et celle du téléphone est l'azimut.

L'accéléromètre, quant à lui, mesure les accélérations du téléphone en trois dimensions. Ce qui va nous intéresser avec ce capteur, c'est que même sans mouvement, il enregistrera une accélération : la gravité. En effet, la gravité terrestre engendre en permanence une accélération de 9.78m/s^2 en direction de la terre. Avec l'accéléromètre, on va donc connaitre la direction de la force de gravité et donc de la terre. On va ensuite utiliser l'angle entre l'axe horizontal du téléphone et la direction de la gravité pour déterminer le roulis. On effectue la même mesure avec l'axe vertical pour calculer le tangage.

2. Navigation GPS

a. Fonctionnement

Un système de navigation GPS peut être décomposé en plusieurs fonctions :

- Positionnement sur la carte
- Recherche d'adresse
- Calcul d'itinéraire
- Guidage audio et visuel
- Affichage de la carte

Le système a besoin de données pour fonctionner. C'est pourquoi un système de navigation GPS est toujours accompagné de deux bases de données. La première, qui contient la carte, c'est-à-dire les routes et les données géographiques telles que les lacs, les forêts ou les zones urbaines. Et la deuxième qui contient toutes les adresses que l'on pourra rechercher et choisir comme destination.

Pour commencer le système va récupérer la position de la voiture grâce au récepteur GPS. Une fois que l'on dispose de cette position, il va rechercher dans la base de données de cartographie la cellule associée à cette position pour l'afficher à l'écran. À partir de là, l'utilisateur va pouvoir choisir sa destination. Au cours de cette étape, le système propose les routes disponibles qui correspondent aux premières lettres tapées par l'utilisateur. Dès que la destination est choisie, le système va calculer l'itinéraire entre ces deux positions. L'itinéraire sera composé des routes à emprunter, mais aussi des instructions de guidage tel que tourner à gauche, prenez la troisième sortie, etc.

Pendant le trajet, l'affichage de la carte et des instructions sera mise à jour à chaque position. De plus, à chaque changement d'instruction, cette dernière pourra être dictée par le système pour prévenir le conducteur sans qu'il ait besoin de tout le temps regarder l'appareil.

La plupart de ces systèmes proposent de choisir le type d'itinéraire que l'on désire : le plus court, le plus rapide, éviter les péages, etc. De plus, certains intègrent aussi les données de trafic en temps réel pour proposer le meilleur itinéraire tout en évitant les bouchons, travaux ou accidents.

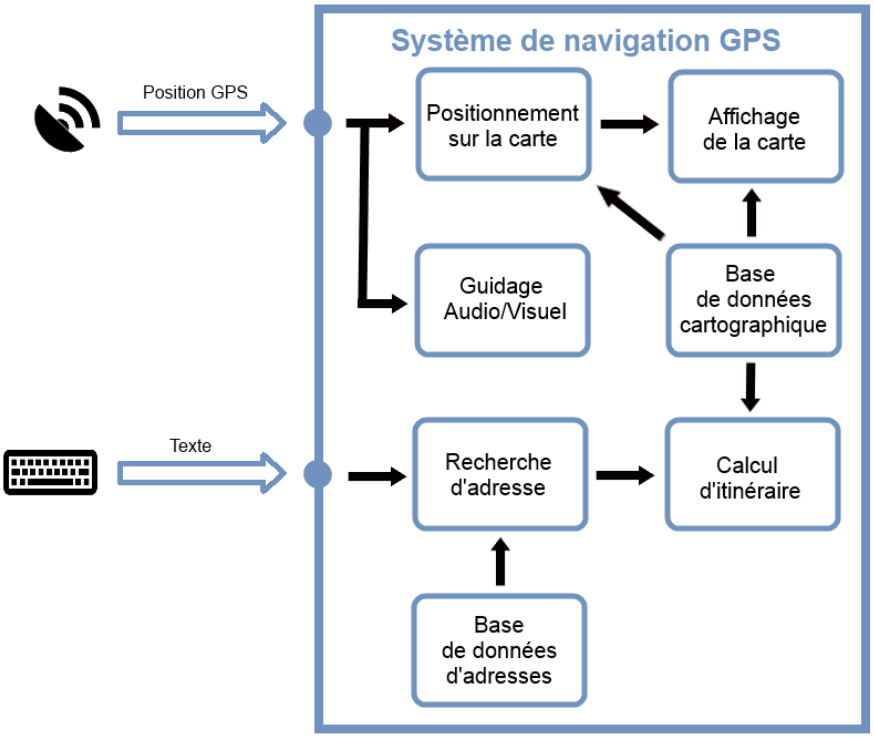


Schéma 10 : Fonctionnement d'un système de navigation GPS

La base de données de cartographie sert au calcul de l'itinéraire, au positionnement sur la carte, mais aussi à l'affichage de la carte. Deux contraintes doivent être prises en compte pour cette base de données. Tout d'abord, on doit pouvoir charger uniquement la partie de la carte qui nous intéresse afin de ne pas saturer la mémoire vive* de l'appareil. Pour cela, la carte sera découpée en cellule carrée. Ainsi pour calculer un itinéraire on pourra charger uniquement les cellules présentes sur notre itinéraire et pour l'affichage on chargera uniquement les cellules présentes à l'écran.

D’autre part, la recherche d’itinéraire ou d’adresse doit être le plus rapide possible. Pour résoudre ce problème, la base de données va être organisée afin de ne pas avoir à vérifier chaque cellule de la carte ce qui prendrait trop de temps. L’organisation retenue est appelée Quadtree*. Avec cette organisation, on distingue trois types de cellules. La cellule de type racine, qui regroupera par exemple un pays, qui n’aura pas de cellule parente, mais qui contiendra quatre cellules fille. Les cellules de type nœud qui ont un parent de type racine ou nœud et quatre cellules filles. Et enfin les cellules de type feuille qui ont un parent généralement de type nœud, mais qui n’ont pas de filles. Avec cette organisation on va découper chaque cellule en quatre sous cellule en partant de la cellule racine jusqu’aux cellules feuilles. La taille de ces dernières sera définie par l’application en se basant par exemple sur le nombre de routes par cellule. De plus, chaque cellule possède un identifiant contenant sa position dans la cellule parente, entre 0 et 3, précédé de l’identifiant de sa cellule parent (Illustration 5). Cet identifiant permet de retrouver facilement une cellule dans le Quadtree*.

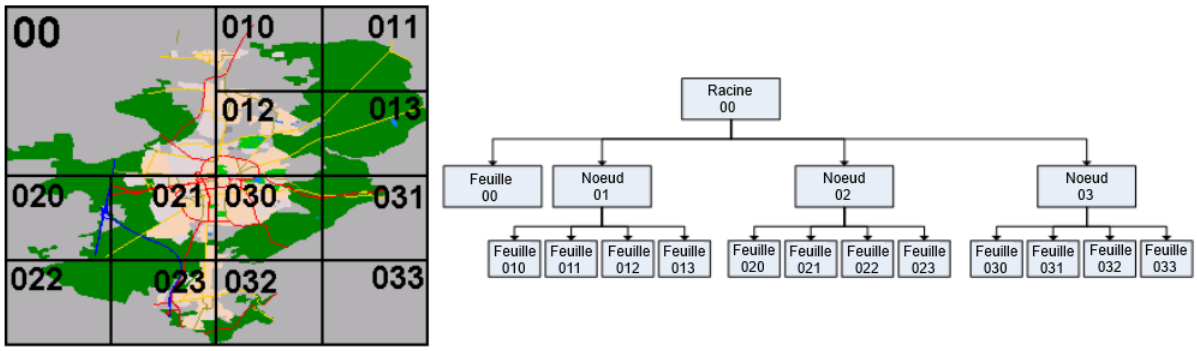


Illustration 6 : Exemple d’organisation de Quadtree pour une ville

Chaque cellule contiendra les coordonnées géographiques de la zone qu’il englobe et uniquement les cellules feuilles contiendront les données de cartographie (routes, intersection et donnée géographique). Grâce à cela on va pouvoir rechercher rapidement une cellule à partir de ses coordonnées. En effet, pour chaque cellule on aura besoin de tester uniquement les sous-cellules contenant les coordonnées recherchées. À chaque niveau, on divise donc par quatre le nombre de cellules à tester.

b. *Architectures*

On pourra distinguer deux types différents d'architecture pour un système de navigation GPS que l'on a déjà abordé avec les précédents systèmes :

- Architecture embarquée
- Architecture distribuée

Dans le cas d'un système embarqué, tout le système se retrouve donc sur le mobile. Sur un tel système, la partie la plus gourmande en temps de calcul et la phase de calcul d'itinéraire. Cependant, cette dernière n'ayant pas besoin d'un temps de réponse immédiat et s'exécutant déjà dans un délai de quelques secondes, ce n'est pas une contrainte pour l'exécuter sur un mobile. La vraie contrainte de cette architecture est la taille des bases de données de cartographie. En effet, ces dernières mesurent environ 300 mégaoctets pour un pays comme la France. C'est pourquoi ces systèmes de navigation sont rarement fournis avec des cartes préinstallées et proposent à la place de télécharger au sein de l'application les cartes des pays voulues. Il faudra donc prévoir le temps de téléchargement avant de pouvoir utiliser ces systèmes pour la première utilisation dans un pays. Il faut aussi remarquer qu'une fois les cartes téléchargées, l'application n'aura plus besoin d'Internet pour fonctionner, très utile lors d'une utilisation à l'étranger, de plus l'affichage de la carte sera instantané.

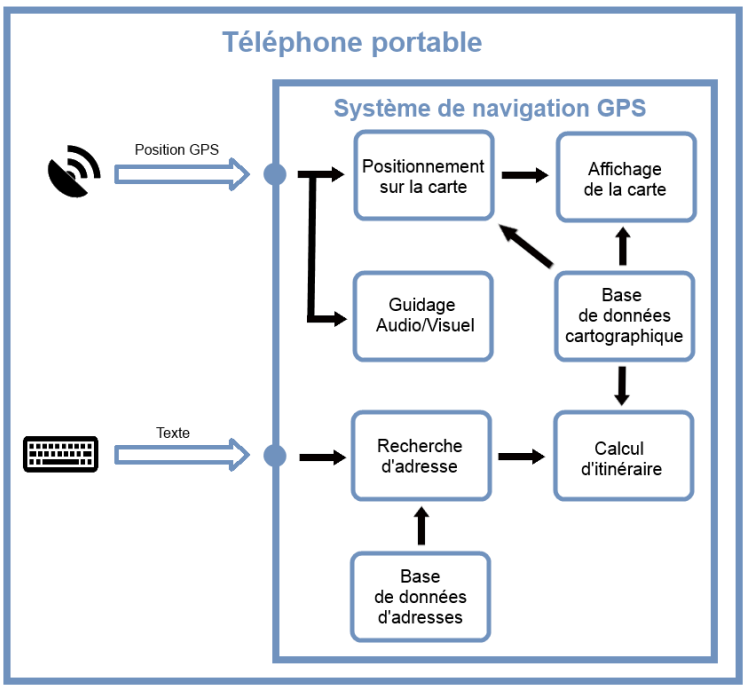


Schéma 11 : Architecture système de navigation GPS embarquée

Avec un système distribué, la plupart du système va se retrouver sur des serveurs. Le mobile envoie sa position GPS aux serveurs qui vont le positionner sur la carte et lui renvoyer uniquement la partie de l'affichage de la carte le concernant. Pour la recherche de destination, c'est le même principe, le portable envoie l'adresse recherchée et le serveur recherche cette adresse, calcule l'itinéraire et renvoie l'itinéraire au téléphone. Ensuite, en se basant sur la position GPS et l'itinéraire renvoyé par le serveur, le mobile va effectuer le guidage vocal et visuel. Cette architecture permet d'avoir une application légère, les bases de données n'étant pas stockées sur le mobile, et ne nécessite pas de préparation pour l'utiliser dans un pays, elle est fonctionnelle directement. Cependant, comme pour tous les systèmes utilisant des serveurs, il faut bien évidemment une connexion Internet. Le temps de réponse du système, surtout pour l'affichage de la carte, sera grandement augmenté.

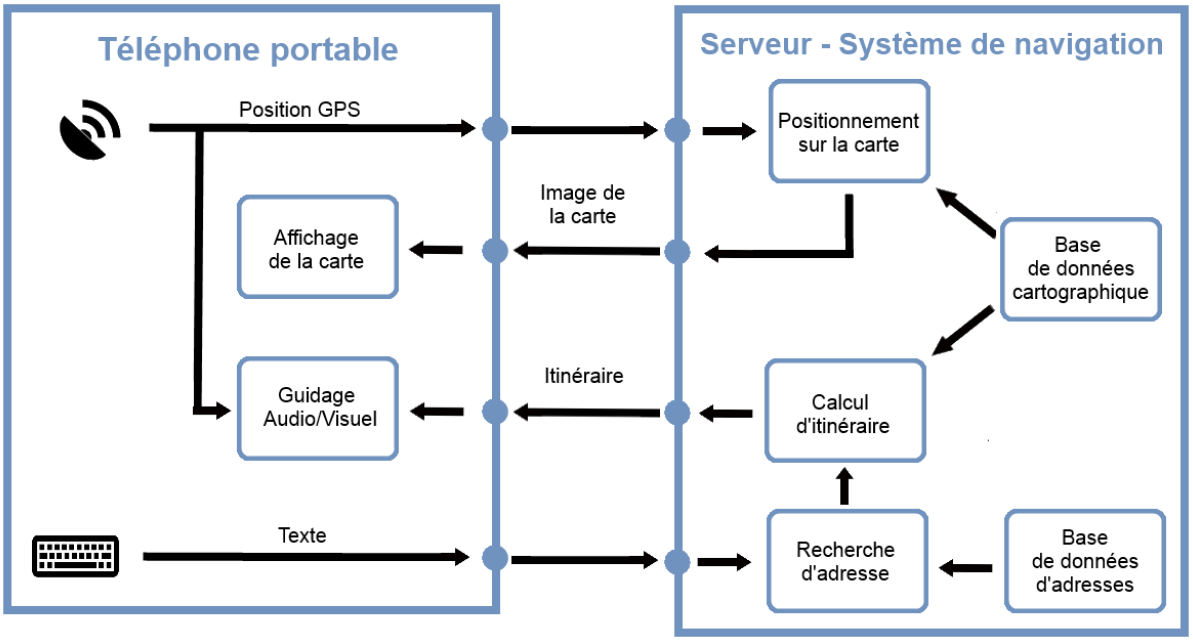


Schéma 12 : Architecture système de navigation GPS distribuée

3. Recherche de points d'intérêts à proximité avec réalité augmentée

Les applications de recherche de points d'intérêts à proximité permettent à l'utilisateur d'effectuer une recherche, comme avec un moteur de recherche standard, en rajoutant un critère de recherche qui est la position. Les résultats de cette recherche, que l'on appelle points d'intérêts, peuvent être de différentes sortes : un magasin, une banque, un restaurant, un cinéma ou encore un lieu public. En se basant sur la position de l'utilisateur, l'application va rechercher les endroits correspondant au mot clé recherché et dont la position se situe à moins de « X » kilomètres de l'utilisateur. Le rayon de recherche étant, soit défini par l'application, soit paramétré par l'utilisateur.

a. La réalité augmentée

La réalité augmentée regroupe les systèmes informatiques qui permettent de superposer à la perception que nous avons de la réalité, des objets virtuels en deux ou trois dimensions et ceci en temps réel. On peut distinguer deux types de réalité augmentée. La première, qui va nécessiter un marqueur, un genre de code-barres, pour savoir quel objet est à afficher et où. La deuxième, celle qui nous intéresse, se base sur les positions de l'utilisateur et des objets virtuels, ainsi que l'orientation du téléphone afin d'afficher ces objets.

Le système doit tout d'abord récupérer la partie « réalité » perçue par l'utilisateur. Pour cela, on va récupérer le flux vidéo de la caméra du téléphone que l'on va afficher à l'écran. C'est sur ces images que vont être superposés les objets virtuels. Il faut aussi déterminer la situation de l'utilisateur et de son téléphone dans l'espace. Le système aura besoin uniquement de la géolocalisation et de l'orientation, la direction du déplacement n'étant pas nécessaire.

Pour la partie « virtuelle », le système va créer un monde virtuel, comme dans un jeu vidéo, contenant les objets virtuels à afficher. Ensuite, il positionne le téléphone dans ce monde grâce aux données de géolocalisation et d'orientation. Pour finir, il affiche le flux vidéo récupéré de la caméra sur lequel il superpose ce monde virtuel du point de vue du téléphone.

Dans notre application de recherche de point d'intérêts, cela va permettre, en plus de visualiser les résultats sur une carte, de voir leur emplacement par rapport à notre position sur l'image de la caméra (Illustration 6).

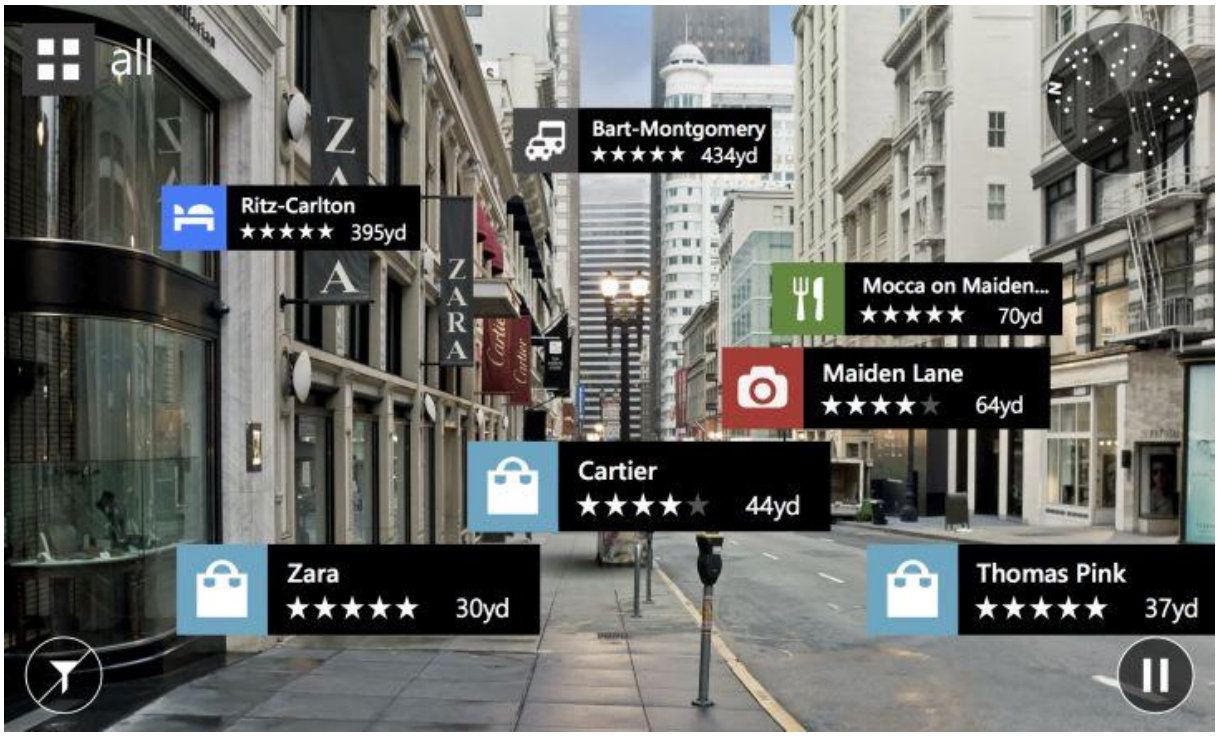


Illustration 7 : Capture d'écran de l'application de réalité augmentée Nokia : Here City Lens

b. Fonctionnement

Le fonctionnement d'un tel système est relativement simple et se décompose en deux parties :

- La recherche
- La réalité augmentée

Le système va d'abord commencer par rechercher les points d'intérêts correspondant aux mots clés entrés par l'utilisateur. Ensuite, en se basant sur la position reçue par le récepteur GPS, il va filtrer les résultats en gardant uniquement ceux se trouvant dans le rayon de recherche souhaité. Pour cela, on va calculer la distance entre les points d'intérêts et la position de l'utilisateur grâce à la trigonométrie. Seulement si on effectue ce calcul sur chacun des résultats, cela pourrait prendre un certain temps en fonction de leur nombre.

On peut optimiser facilement ce processus en supprimant les résultats ne se trouvant pas dans un carré centré sur l'utilisateur et dont les côtés font deux fois le rayon de recherche (Schéma 13).

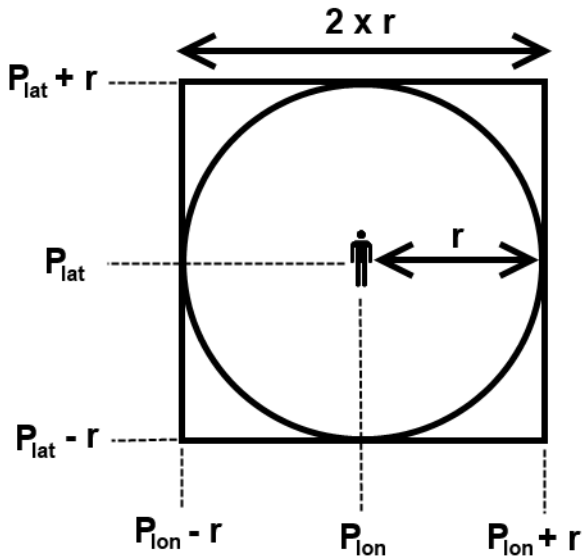


Schéma 13 : Filtrage rayon de recherche

En se basant sur un carré et non plus un cercle, on peut définir simplement des bornes supérieures et inférieures pour la latitude et la longitude. Par exemple, si on se trouve au point de latitude 44 et de longitude 0.5 et qu'on veut les résultats dans un rayon de 15 kilomètres. À cette position, 15 kilomètres représentent environ 0.14 en latitude et 0.2 en longitude. On va donc filtrer les résultats en supprimant ceux ayant une latitude supérieure à 44.14 et inférieure à 43.86 et une longitude supérieure à 0.7 et inférieure à 0.3. Filtrer des bornes est beaucoup plus rapide que de calculer les distances et supprime la grande majorité des mauvais résultats. Il ne restera plus qu'à calculer la distance des points d'intérêts restant pour filtrer les quelques résultats à supprimer.

Pour effectuer la recherche, l'application doit évidemment se baser sur une base de données contenant tous les points d'intérêts nécessaires à l'application. On pourra soit utiliser une base de données interne, par exemple pour l'application d'une marque particulière où seront stockés les positions, adresses et horaires de chacun de ses magasins, soit interroger directement un moteur de recherche, comme Google, qui retournera directement les résultats suivant une position et des mots clés.

La partie réalité augmentée va donc récupérer les résultats de la recherche et les afficher, comme décrite précédemment, en se basant sur la position reçue du récepteur GPS et l'orientation du téléphone du magnétomètre.

c. Architecture

Concernant l'architecture d'un tel système, on s'orientera vers une architecture distribuée. On pourrait utiliser un système complètement embarqué, mais ça limiterait grandement le résultat des recherches, surtout qu'effectuer une recherche par Internet est relativement rapide et permet d'avoir une base de données à jour.

La partie recherche avec la base de données se retrouvera donc déportée sur un ou plusieurs serveurs alors que la partie réalité augmentée devra rester embarquée sur le mobile (Schéma 13).

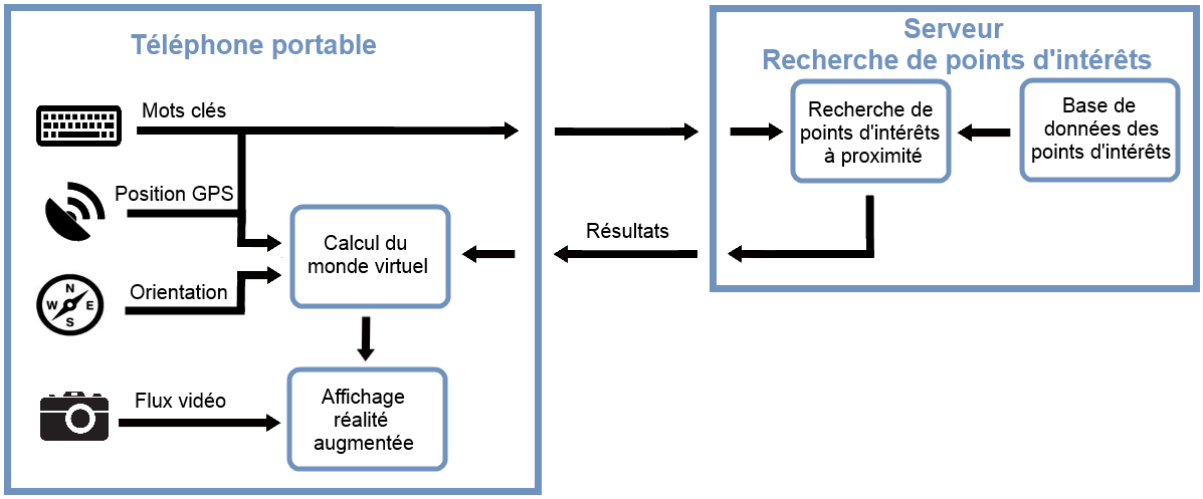


Schéma 14 : Architecture d'un système de recherche de points d'intérêts à proximité avec réalité augmentée

C. Services

1. Bancaire et commercial

Parmi les applications d'assistant permettant d'accéder à certains services depuis un mobile, on retrouve deux grandes catégories :

- Les applications de service bancaire, appelé mBanking
- Les applications de service commercial, appelé mCommerce

L'avantage d'accéder à ces services sur mobile c'est que l'on va pouvoir les utiliser à n'importe quel moment et depuis n'importe quel endroit.

a. mBanking

Les applications de mBanking permettent aux clients d'une banque d'accéder à chacun des services proposés par celle-ci directement depuis son mobile. Ces applications se décomposent en trois fonctionnalités principales :

- Accès à ses informations bancaires
- Gestion de son compte
- Transactions bancaires et boursières

Dans les informations bancaires, on retrouve différents types d'information. On trouve évidemment l'historique de son compte avec les débits et crédits sur un ou plusieurs mois, mais aussi le suivi de ses placements bancaires avec le taux courant et les intérêts engendrés et à venir. On dispose aussi de relevés plus détaillés, pour ses crédits bancaires avec, par exemple, la durée et le montant restant à rembourser, et pour ses actions boursières avec des informations comme le cours actuel d'une cotation, son évolution par rapport à la veille et son gain. De plus, on peut suivre ses demandes de crédit, pour savoir s'ils sont validés ou non, de carte de crédit et de chèques, pour savoir quand aller les chercher ou quand on va les recevoir. Et enfin, on peut localiser les emplacements des distributeurs automatiques de billets et des agences de sa banque.

Pour la gestion de son compte, de telles applications vont permettre de rentrer en communication avec sa banque, soit par mail ou téléphone, soit via un chat intégré à l'application, directement avec son conseiller ou son agence. Elles proposent aussi la possibilité de faire opposition sur son chèque ou sa carte de crédit en cas de perte ou de vol pour être le plus réactif possible. On pourra aussi paramétrer une alerte qui se déclenchera suivant l'activité du compte, par exemple en définissant un seuil du solde restant à ne pas dépasser ou lorsqu'un débit trop important est réalisé. Pour finir, on pourra aussi gérer ces polices d'assurance souscrite auprès de sa banque pour déclarer un sinistre, demander une souscription ou une résiliation ou encore changer de formule. Les banques proposent de plus en plus d'assurance, comme les assurances perte et vol de son portefeuille, les assurances habitation ou encore les assurances vie.

Du côté des transactions, on va pouvoir effectuer un virement bancaire directement depuis l'application vers un compte bancaire national ou à l'international. Il faudra bien sûr posséder le numéro de compte (RIB ou IBAN) du destinataire. On dispose aussi de différentes fonctionnalités en relation avec la bourse. On peut consulter le cours de la bourse en temps réel, gérer son portefeuille d'action, acheter et vendre, ou paramétrer des alertes et notifications quand une action atteint un certain seuil.

b. mCommerce

Le mCommerce regroupe les applications permettant de faire du commerce sur son mobile. Une application permettant de réserver un billet de train ou un ticket de cinéma sont des applications de mCommerce. On va cependant retrouver plusieurs types d'application de mCommerce :

- Billetterie, vente de ticket
- Carte de fidélité, bon de réduction
- Vente par correspondance

Les applications de billetterie permettent de réserver, acheter et annuler ses billets (de train, cinéma, avion, spectacle…) directement depuis son Smartphone. Une fois que son billet est acheté, on le reçoit au format numérique directement dans l'application ou par mail. Ce billet numérique est utilisable instantanément, la vérification du billet se fera en scannant le code-barres se trouvant dessus, chaque billet ayant un code-barres unique.

Les applications de cartes de fidélités et de bons de réductions ont pour but de remplacer leurs versions papier ou plastique en une version numérique moins encombrante et toujours à portée de main. Comme pour les billets numériques, les bons de réduction se présentent sous la forme d'un code-barres à scanner en caisse pour en bénéficier. Pour les cartes de fidélité, on distingue deux types de cartes. Les cartes classiques, contenant un code-barres à scanner en caisse, permettant d'engendrer des points ou de bénéficier de réduction immédiate. Les cartes à tampon dont on doit la faire tamponner un certain nombre de fois pour bénéficier d'une offre (réduction ou produit gratuit). Pour les cartes classiques, on reproduit simplement le code-barres dans l'application et il n'y aura plus qu'à scanner ce dernier au lieu de sa carte.

Pour les cartes à tampon, c'est le commerçant qui possédera un code-barres qu'il vous fera scanner à chaque passage en caisse. Lorsque l'application scanne un code-barres, elle cherche la carte correspondant à ce dernier et ajoute un tampon virtuel à cette carte. Lorsqu'on atteint le nombre de tampons requis, on le montre au commerçant qui vous fera bénéficier de la réduction.

Enfin, on retrouve les applications de vente par correspondance classique, permettant la consultation d'un catalogue, la commande et le paiement d'articles. Ce type d'application représente souvent le site Web de vente par correspondance adapté aux applications mobiles.

c. Architecture

Dans ce type d'application, toute la logique métier se retrouve dans le système d'information déjà existant du commerçant ou de la banque. Tous les services étant déjà présents sur leur serveur, l'application sert uniquement d'interface utilisateur. D'ailleurs, la plupart de ce type d'application vient en complément d'un site Web se basant déjà sur le SI de l'entreprise. On retrouve ainsi pour ce type d'application une architecture client-serveur, ou le client, le Smartphone, va envoyer une requête au serveur. Cette dernière servira, soit pour récupérer les données à afficher, soit pour demander le démarrage d'une procédure (exemple : opposition de son chéquier), soit de mettre à jour des données (exemple : mises à jour de son mail de contact).

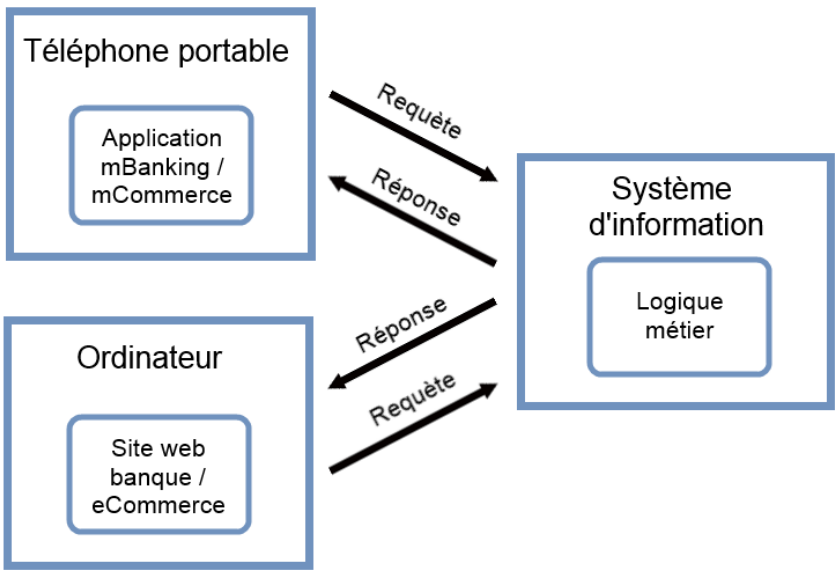


Schéma 15 : Architecture des applications de mBanking et de mCommerce

2. Paiement

Le paiement mobile va permettre à son utilisateur d'effectuer une transaction qui sera débitée sur sa carte bancaire, sa facture téléphonique ou sur un porte-monnaie électronique. On va distinguer trois types de paiement par mobile :

- Le paiement à distance
- Le paiement de proximité (via une borne)
- Le transfert d'argent de personne à personne

a. Types de paiement

Le paiement à distance est très utilisé sur mobile. Il permet de régler l'achat d'application et toutes les transactions au sein de ces applications. Il est aussi utilisé pour effectuer des paiements sur des sites de e-commerces. La transaction peut s'effectuer via sa carte bancaire ou sa facture téléphonique. Le paiement par carte bancaire peut être effectué soit en renseignant ses numéros de carte bancaire, soit en passant par un service de paiement en ligne tel que PayPal. Le règlement via sa facture téléphonique s'effectue en appelant ou en envoyant un SMS à des numéros surtaxés. Lorsque la transaction est effectuée, on reçoit un code que l'on va transmettre au commerçant prouvant que l'achat a bien été effectué. Ces moyens de paiements sont identiques à ceux utilisés pour régler ces achats sur Internet.

Le transfert d'argent de personne à personne permet, comme son nom l'indique, de transférer de l'argent directement entre deux personnes. On peut considérer cela comme un virement sur mobile, sauf qu'ici il n'y aura pas besoin de connaitre les informations bancaires du destinataire. En effet, on définit à quelle personne transférer l'argent par un identifiant (son numéro de téléphone ou son adresse email). Certains systèmes permettent aussi d'identifier une personne en scannant un code-barres unique à chaque compte qui se retrouvera affiché sur l'écran du propriétaire. Une autre possibilité est d'utiliser les technologies de communication sans contact pour transférer les identifiants en rapprochant le téléphone des deux personnes.

Le paiement de proximité consiste à payer avec son téléphone directement chez le commerçant. Pour cela, ce dernier doit disposer d'une borne de paiement, comme celle utilisée pour le paiement par carte bancaire, et l'utilisateur doit avoir enregistré une carte bancaire sur son téléphone. Grâce aux technologies de communication sans contact, lorsque l'on va approcher le téléphone de cette borne, le paiement va être effectué sur la carte bancaire enregistrée sur le téléphone. On parle de paiement sans contact.

J'ai décidé de développer ce dernier type de paiement qui reçoit un grand effort de déploiement de la part des banques, des commerçants et des fabricants de téléphones portables. De plus, ce dernier permet réellement de remplacer au quotidien les moyens de paiement habituel par son mobile.

b. Paiement sans contact

La technologie utilisée pour communiquer les informations bancaires lors d'un paiement sans contact est la technologie NFC*, pour Near Field Contact ou, en français, Communication en champ proche. Le NFC* est une technologie de communication radio à courte portée. Elle permet l'échange d'information entre deux périphériques à une distance maximum d'environ dix centimètres.

Avant de pouvoir utiliser son téléphone pour effectuer des paiements, l'utilisateur devra d'abord renseigner les informations de sa carte de crédit et, s'il le veut, définir un code PIN pour sécuriser l'accès à ses cartes de crédit.

Une fois que le téléphone est configuré, lorsque l'utilisateur voudra effectuer un paiement, il n'aura qu'à choisir la carte avec laquelle il veut effectuer le paiement, saisir le code PIN s'il en a défini un et approcher le téléphone de la borne de paiement NFC*. Le commerçant aura au préalable saisi le montant à prélever. La borne va alors récupérer les informations de la carte bancaire que l'utilisateur a choisie et va initier le processus de transaction. Ce dernier étant identique à celui d'un paiement avec une carte de crédit.

Le processus de transaction commence avec l'envoi des informations de carte bancaire et du montant à la banque du commerçant. Cette dernière va demander l'autorisation pour effectuer la transaction à la banque de l'utilisateur.

Si le compte a assez d'argent pour effectuer le paiement la banque envoie une autorisation de paiement. La banque du marchand effectue alors la transaction et envoi la validation ou le rejet du paiement à la borne du marchand.

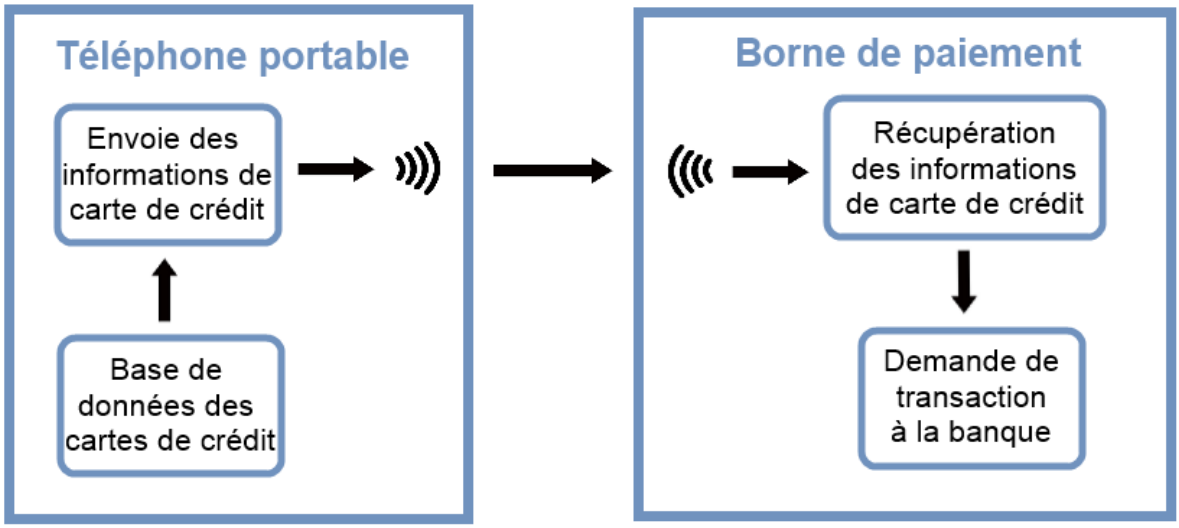


Schéma 16 : Fonctionnement du paiement sans contact via NFC

II. Limitations de l'assistant mobile

A. Contraintes matérielles spécifiques au mobile

Le mobile en lui-même possède déjà certaines limitations :

- Une puissance limitée
- La qualité et l'imprécision des capteurs
- L'autonomie de la batterie

1. Puissance limitée

La puissance limitée des mobiles est due à leur faible taille ne permettant pas d'embarquer de matériel plus performant. On va recenser trois composants qui vont être critiques pour le bon fonctionnement des assistants :

- La mémoire de stockage
- La mémoire vive*
- Le processeur

La mémoire de stockage est le composant servant à stocker les données du Smartphone. On retrouve dans cette dernière le système d'exploitation*, les applications et leurs données ainsi que les documents de l'utilisateur (photos, musiques, vidéos, etc.). Seulement, sur les Smartphones, on retrouve globalement des espaces de stockage allant de 4 giga-octets à 32 giga-octets suivant la gamme du téléphone. Même si cela peut paraitre suffisant, on arrive vite à la limite. En effet, de cet espace disponible, il faut enlever environ un giga-octet pour le système d'exploitation*, ensuite il faut compter l'espace prit par les applications et les documents qui peuvent prendre tout l'espace restant suivant votre utilisation. De plus, comme on a vu dans le chapitre précédent, certains systèmes d'assistants ont besoin de stocker des bases de données plus ou moins conséquentes pour fonctionner dans le cas d'une architecture embarquée. Par exemple, dans le cas d'un système de navigation GPS, une base de données pour un pays va peser environ 300 à 400 mégaoctets, un giga-octet représentant 1000 mégaoctets.

La mémoire vive* sert à stocker les applications et les processus du système d'exploitation* en cours d'exécution ainsi que leurs données. Cette dernière, étant beaucoup plus rapide que la mémoire de stockage, permet de réduire les temps d'accès aux données. Lorsqu'une application sera exécutée, cette dernière sera copiée de la mémoire de stockage vers la mémoire vive*. Ensuite, ce sera l'application elle-même qui demandera à copier les données nécessaires à son fonctionnement dans cette mémoire. La quantité de mémoire vive* disponible sur les Smartphones varie de 256 mégaoctets à 2 giga-octets. Comme pour la mémoire de stockage, ce sont les bases de données nécessaires à l'application qui pourront poser problème si elles sont trop importantes. Dans ce cas, on va essayer de pallier à cela en optimisant le chargement en mémoire. Un exemple est le système utilisé par les applications de navigation GPS pour charger uniquement les données de cartographie nécessaires (les cellules présentent à l'écran). Si une optimisation n'est pas possible ou qu'elle ne suffit pas, il faut donc réduire la taille de cette base de données ce qui entrainera forcément une baisse de précision ou de pertinence du système.

Enfin, le processeur est celui qui va exécuter les applications et tous les processus du système d'exploitation*. Plus précisément, ces derniers seront traduits en suites d'instructions et ce sont celles-ci qui seront exécutées par le processeur. Les caractéristiques principales indiquant la puissance de calcul d'un processeur sont sa fréquence et son nombre de cœurs. La fréquence correspond au nombre d'instructions que chaque cœur va pouvoir exécuter par seconde, exprimé en hertz. Sur le marché des Smartphones, on retrouve des processeurs allant du simple cœur cadencé à 1 gigahertz jusqu'au quadruple cœur cadencé à 2.2 gigahertz. La puissance de calcul du processeur aura un impact sur le temps de réponse du système : plus il est puissant, plus le système répond rapidement. Tous les assistants mobiles n'ont pas besoin d'une grande puissance de calcul, seulement ceux dont leurs algorithmes* nécessitent un grand nombre de calculs. Parmi ceux-là, on retrouve :

- les systèmes de reconnaissance vocale avec la comparaison entre la combinaison de phonèmes et tous les mots du dictionnaire.
- les systèmes de reconnaissance d'image avec la détection et l'extraction de caractéristiques, ainsi que la classification.
- les systèmes de navigation GPS avec l'affichage de la carte et le calcul d'itinéraire.

2. Qualité et imprécision des capteurs

Comme on l'a vu dans la partie I, les assistants mobiles se basent souvent sur des capteurs pour fonctionner. Par exemple :

- pour le système de reconnaissance vocale, le microphone récupère la voix de l'utilisateur
- pour le système de reconnaissance d'image, l'appareil photo enregistre ce que voit l'utilisateur
- pour le système de navigation GPS, le récepteur GPS et le magnétomètre permettent de connaitre la position et l'orientation du téléphone.

Tous ces capteurs permettent à ces systèmes de prendre conscience de leur environnement grâce aux données qu'ils collectent. Seulement, il faut que ces données soient assez précises et d'assez bonne qualité pour pouvoir être exploité correctement par le système.

Pour le microphone, on peut rencontrer certains problèmes si la qualité de ce dernier n'est pas optimale. Le plus fréquent, c'est la saturation du microphone. Cela se produit lorsque le son enregistré dépasse le niveau sonore maximum supporté par ce dernier. Dans ce cas-là, le signal enregistré est inutilisable, on entend seulement des « craquements ». Un autre problème peut se produire, c'est que le signal soit perturbé par un bruit parasite. Lorsque cela se produit, l'isolation de la voix de l'utilisateur est rendue plus compliquée et donc diminue la précision de l'extraction des caractéristiques et donc de la traduction en phonème. Enfin, la précision du signal lui-même peut être affectée suivant la qualité du microphone. Quand cela se produit, le microphone ne retransmet pas exactement ce que l'on entend. Dans ce cas, l'extraction de caractéristiques perd en précision tout comme la traduction en phonème. Tout cela peut entrainer une baisse de précision de la reconnaissance vocale, voir une impossibilité de reconnaitre les mots.

En ce qui concerne l'appareil photo, la qualité nécessaire de ce dernier va dépendre de son utilisation. Pour un système de reconnaissance d'image, cela va aussi dépendre du niveau de détail que l'on veut pouvoir détecter. Pour avoir le niveau de détail souhaité, il faut d'abord avoir une résolution de capteur adapté. Cette résolution correspond au nombre de pixels de la photo, plus elle est élevée plus on pourra voir de détails.

Cependant, ce n'est pas tout, il faut aussi prendre en compte la gestion du bruit par le capteur photographique. Ce bruit se produit sur les zones peu éclairées, mais aussi sur les zones uniformes en couleur et donne lieu à une variation de couleur et de luminosité (Illustration 8). Les conditions dans lesquelles la photo sera prise seront aussi déterminantes pour la qualité de celle-ci. La faible luminosité, l'exposition au soleil et la faible distance avec le sujet de la photo sont des conditions que l'on peut qualifier d'extrêmes pour un capteur photo. Cela aura pour effet une luminosité trop faible ou trop élevée ou un « flou » sur le sujet. Avec une photo de mauvaise qualité, peu importe sa résolution, on a une perte de définition et donc de détail. Cela diminue donc l'efficacité de la détection et de l'extraction des caractéristiques déterminantes pour la reconnaissance d'un objet.

Illustration 8 : Bruit sur une image prise en fable luminosité

En ce qui concerne le récepteur GPS, la qualité de son antenne sera déterminante pour son fonctionnement. Avec une antenne de mauvaise qualité, les signaux des satellites GPS seront de faible intensité, voire inexistant. En conséquence, le délai pour obtenir une première position sera plus long, de l'ordre de plusieurs minutes, voire impossible. De plus, avec un signal des satellites faibles, la précision de la position sera tout aussi faible, supérieure à cinquante ou cent mètres. Avec une précision aussi faible, le système de navigation GPS ne pourra vous annoncer les instructions à temps et ne pourra pas déterminer exactement la route sur laquelle vous vous trouvez.

Dans l'idéal, il faudrait donc que chaque Smartphone soit équipé de capteurs de bonne qualité. Seulement, de tels capteurs ont un coût évidemment bien plus élevé. Par ailleurs, le développeur devra prendre en compte que l'application sera susceptible d'être installée sur différents téléphones équipés de capteurs de différentes qualités.

3. Autonomie de la batterie

Comme pour tout appareil mobile, le fonctionnement d'un téléphone portable dépend de sa batterie et donc de sa consommation électrique. Seulement, les Smartphones ont tendance à consommer beaucoup d'énergie. En effet, maintenant la plupart des derniers Smartphones ont une autonomie rarement plus élevée qu'une journée en utilisation standard. Et cela peut s'expliquer simplement.

Tout d'abord, on a des téléphones que l'on veut légers, fins et puissants. De telles caractéristiques interdisent l'installation d'une batterie de grande capacité, ces dernières étant lourdes et imposantes. De plus, plus le mobile sera puissant, plus il aura tendance à consommer de l'électricité. En outre, les Smartphones ont pour vocation de proposer toujours plus de fonctionnalités, donc d'être utilisé de plus en plus souvent avec toujours plus de capteurs. Ce qui est encore plus vrai avec les assistants du quotidien qui ont pour but de vous assister tout au long de la journée.

La consommation de la batterie et sa capacité sont donc aussi des contraintes à prendre en compte. En effet, en utilisation mobile (ce qui est le but premier d'un téléphone portable), on ne pourra pas brancher son téléphone à la prise électrique ou à l'allume-cigare pour le recharger et donc continuer son utilisation. Car évidement, sans batterie, le téléphone est inutilisable et l'assistant inopérant.

B. Contraintes techniques

1. Les systèmes d'exploitation

Le système d'exploitation est un programme installé nativement sur le Smartphone. Il est le premier à démarrer et c'est lui qui permet l'accès à toutes les fonctionnalités du téléphone pour l'utilisateur et aux applications installées. Aujourd'hui, on recense principalement quatre systèmes d'exploitation se partageant la quasi-totalité du marché.

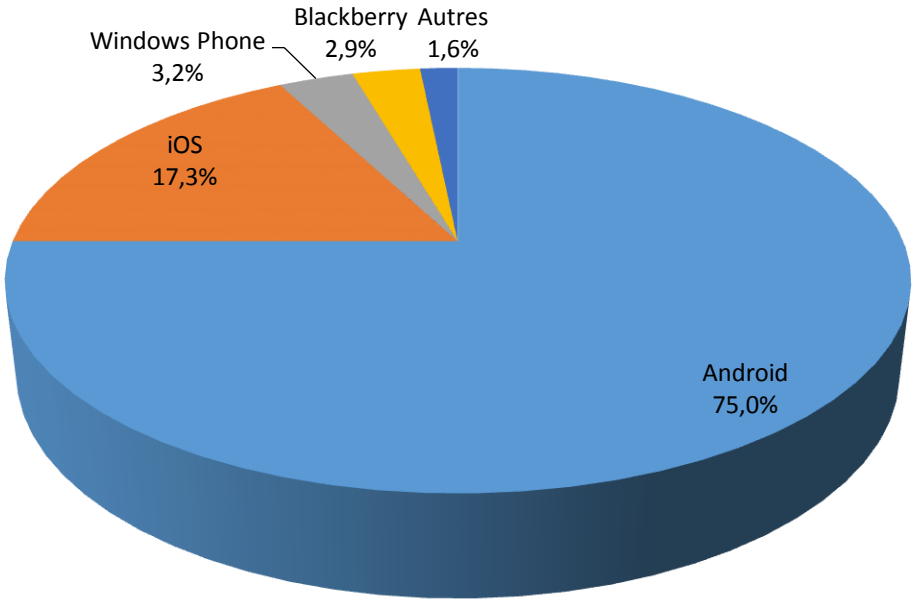


Schéma 17 : Répartition des systèmes d'exploitation mobiles

a. Les principaux systèmes d'exploitation

Android est le système d'exploitation mobile de Google lancé en 2007 dont la dernière version est la 4.2. On le retrouve sur divers appareil mobile tel que, bien évidemment, des Smartphones et des tablettes, mais aussi sur des téléviseurs, des radioréveils ou encore des autoradios. Il est « open source » (le code source est rendu public avec autorisation d'utilisation et de modification) et se base sur un noyau Linux*. Le langage de programmation utilisé pour développer une application Android est le Java*.

iOS est le système d'exploitation développé par Apple dont la première version est sortie en 2007. Actuellement, il est en version 6.1, mais une version 7 est attendue pour la fin de l'année apportant un grand changement de design. Il est destiné exclusivement aux produits mobiles d'Apple, à savoir l'iPhone, l'iPad, l'iPod et l'Apple TV. C'est un dérivé du système d'exploitation Mac OS destiné aux ordinateurs Mac. Il est propriétaire, on ne peut donc pas accéder au code source, ni le dupliquer ou le modifier. Les applications pour iOS sont développées avec le langage de programmation Objective-C*.

Windows Phone est le système d'exploitation de Microsoft déployé en 2010 sous le nom Windows Phone 7, car successeur de Windows Mobile 6. La version actuelle est la 8 avec une mise à jour en 8.1 prévue pour le début 2014. Il est disponible uniquement sur les Smartphones répondant à certaines contraintes afin de garantir à l'utilisateur une expérience utilisateur optimale et identique sur chaque appareil. Il se démarque par son interface utilisateur appelée Modern UI qui se veut simple, claire et moderne et qui met en avant les informations importantes. Il est basé sur le noyau Windows NT et reprend de nombreux composants de Windows 8 afin de faciliter le portage des applications entre ces deux plateformes. Pour le langage de programmation, on va avoir différents choix :

- XAML* pour l'interface et C#*
- CSS* pour l'interface et HTML5*
- C*/C++*

BlackBerry OS est le système d'exploitation développé par RIM lancé en 1999 et dont la dernière version est la 10.1. On le retrouve uniquement sur les téléphones et tablettes de la marque BlackBerry. Il est connu pour son support natif des boites mail d'entreprise qui permet une synchronisation complète avec les messageries d'entreprise telle que Microsoft Exchange*, Lotus Domino* ou encore Novell GroupWise*. Il est propriétaire et se base sur un noyau QNX*. Il propose différents langages de programmation :

- CSS*/HTML5*
- Java*
- C*/C++*

Illustration 9 : Interfaces des principaux systèmes d'exploitation

b. *Les limitations*

Avec ces différents systèmes d'exploitation mobiles sur le marché, l'utilisateur va pouvoir choisir le téléphone qui lui correspond le mieux. De plus, grâce à ce marché concurrentiel, les développeurs de ces systèmes sont obligés de les faire toujours évoluer afin de proposer les meilleurs services possibles. Ceci est autant bénéfique pour les utilisateurs que pour les développeurs d'applications. Seulement, cela apporte aussi quelques contraintes à ces développeurs.

En effet, chaque système d'exploitation apporte son propre écosystème de développement. Ainsi, chaque système utilise des langages de programmation différents et possède ses propres librairies* permettant d'accéder aux fonctionnalités du téléphone. Certaines d'entre elles n'étant d'ailleurs pas forcément accessibles sur chaque plateforme.

Si on veut être présent sur toutes les plateformes du marché, cela implique donc de devoir développer une application différente pour chaque système d'exploitation. Sachant que certaines fonctionnalités ne pourront peut-être pas être présentes dans toutes les versions. De plus, chaque système d'exploitation à sa propre charte graphique, il faudra donc penser à adapter le design de son application à chacun d'entre eux. Tout cela à un coût, c'est pourquoi certaines applications se retrouvent uniquement sur les deux plateformes principales, Android et iOS représentant à elles seules 92% de part de marché.

2. Les architectures orientées serveur

Pour pallier au problème de la puissance limitée des téléphones ou pour accéder à des services en ligne, les systèmes d'assistants ont besoin de se tourner vers des architectures orientées serveur. Soit, une architecture distribuée entre le mobile et un serveur ou complètement déportée sur un serveur. Ce type d'architecture a cependant aussi des limites à différents niveaux :

- communication entre le mobile et le serveur
- la partie serveur elle-même

a. Communication mobile / serveur

Tout d'abord afin que le téléphone puisse communiquer avec le serveur, il faut que ce dernier ait une connexion Internet. Cela empêche donc les utilisateurs n'ayant pas accès à Internet, à cause de leur forfait ou du réseau téléphonique, d'accéder à ces applications.

Ensuite, le temps de réponse du système, c'est-à-dire le temps que met le système pour traiter la requête de l'utilisateur et lui fournir une réponse, en sera affecté. En effet, en plus du temps d'exécution du traitement, il faut rajouter le temps de communication de la requête au serveur et de la réponse au mobile. De plus, ce temps de réponse sera grandement variable en fonction de la qualité du réseau téléphonique, lui-même aussi très irrégulier en condition de mobilité. Par exemple, si on veut transférer au serveur une image capturée avec le téléphone pesant 1 mégaoctet. Avec un réseau de troisième génération (3G) de type UMTS*, ayant un débit moyen de 48 kilo-octets par seconde, le transfert prendra environ 20 secondes. Seulement, si le Smartphone capte uniquement un réseau de deuxième génération (2G) de type GPRS*, avec un débit moyen de 6 kilo-octets par seconde, le transfert prendra à ce moment-là 166 secondes soit 2 minutes et 46 secondes.

Pour réduire ce délai de transfert, certains systèmes ont recours à la compression* des données envoyées au serveur. Les données concernées seront majoritairement des images et des sons. Pour une image, la compression* va s'effectuer soit sur la taille de l'image, sa résolution, soit sur le nombre de couleurs dans l'image.

Dans la plupart des cas, on commencera déjà par redimensionner l'image. En effet, les Smartphones prennent des photos avec toujours plus de mégapixels, en moyenne 8 mégapixels, soit des photos avec une résolution de 3200 pixels de largeur sur 2500 pixels de hauteur. Avec cette démarche, on observe un gain à peu près proportionnel au rapport de redimensionnement. Avec une photo de 8 mégapixels, soit 3200 par 2500 pixels, pesant 2 mégaoctets, on obtient une photo d'environ 100 kilo-octets, en la redimensionnant à 800 par 480 pixels, soit 0,38 mégapixel.

Illustration 10 : Impact d'une diminution de résolution sur une image

Pour les couleurs, il faut savoir que chaque pixel d'une image est représenté par sa couleur. Plus on voudra un grand nombre de couleurs, plus le pixel sera volumineux. Par exemple pour seize millions de couleurs, chaque pixel occupera 24 bits dans lesquels on peut stocker 16 277 216 valeurs. Ainsi avec 8bits, on pourra utiliser 256 couleurs et 4 bits, 16 couleurs. Côté gain de place, c'est proportionnel au nombre de bits par pixel. En passant de 24 bits de couleur à 8 bits on diminuera par trois la taille de l'image.

Illustration 11 : Impact de la réduction du nombre de couleurs sur une image

Pour le son, on va aussi pouvoir agir sur deux paramètres, la résolution et le taux d'échantillonnage. La résolution correspond au nombre de valeurs possible pour notre signal. Il est exprimé en nombre de bits, 8bits donnant 256 valeurs possibles. Plus cette valeur est faible, moins on pourra distinguer des sons différents.

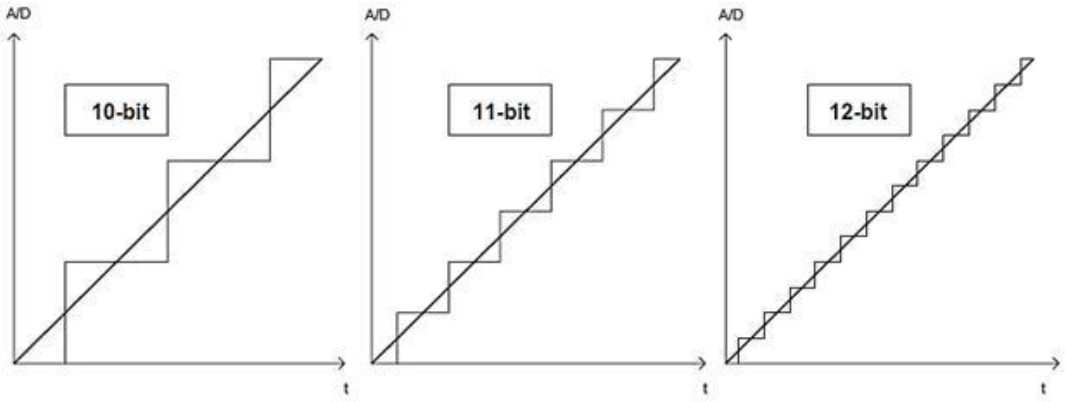


Illustration 12 : Impact de la résolution sur un signal

Le taux d'échantillonnage est le nombre de valeurs, d'échantillon, que l'on va enregistrer par seconde. Plus on le réduit, plus il « manquera » des sons intermédiaires. Le gain en taille est aussi proportionnel au rapport de réduction de ces paramètres.

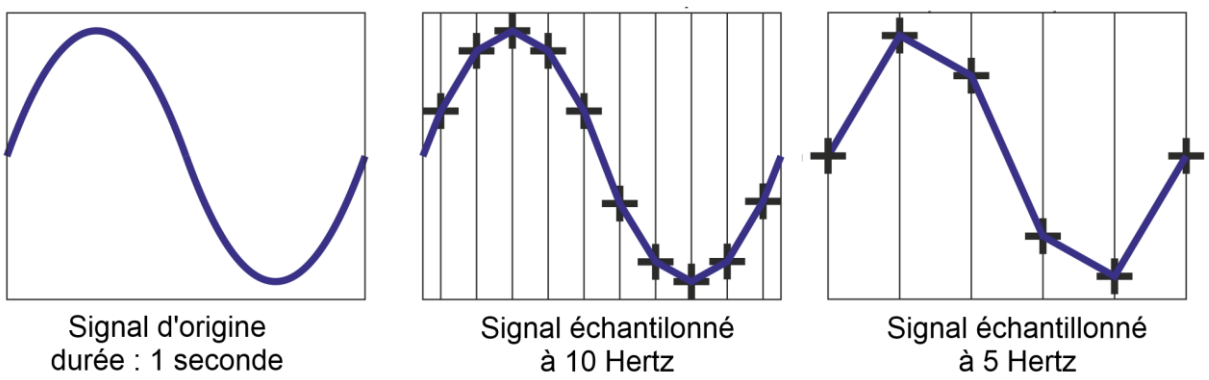


Illustration 13 : Impact du taux d'échantillonnage sur un signal

La compression permet donc de réduire facilement la taille d'une donnée. Cependant, elle en réduit aussi sa qualité et sa précision. Cette perte de précision provoque évidemment une perte d'efficacité du système, empêchant ainsi de reconnaitre un objet sur une image ou un phonème sur un son.

b. Partie serveur

Lorsqu'on utilise une partie serveur dans notre architecture, il faudra tenir compte de différents paramètres qui peuvent empêcher le système de fonctionner correctement.

Tout d'abord, avec une telle architecture, il faut penser que tous les utilisateurs peuvent envoyer une requête en même temps. Le serveur peut donc être vite surchargé et être dans l'incapacité de répondre aux demandes. Par ailleurs, les serveurs devant fonctionner 24 heures sur 24 et 7 jours sur 7, il est fréquent qu'il y ait des pannes. Dans ces cas-là, même si l'application mobile continue de fonctionner, le serveur ne pourra répondre aux requêtes des utilisateurs. De plus, des données peuvent être perdues suite à la panne.

Afin de pallier à tous ces problèmes, on peut déjà prévoir une architecture en conséquence du nombre d'utilisateurs attendu et de la puissance de calcul nécessaire au système. En outre, des systèmes de sauvegarde régulière et de répartition de charge peuvent être utilisés pour réduire les dysfonctionnements en cas de panne. Seulement tous ces systèmes ont des coûts non négligeables ce qui peut être un frein à leur intégration.

C. Intelligence Artificielle

1. Définition

L'intelligence artificielle (IA) est une branche de l'informatique qui étudie et développe des technologies apportant l'intelligence aux machines et aux logiciels. Elle est définie par les chercheurs comme étant l'étude et la conception d'agents intelligents, où ces agents sont des systèmes qui perçoivent leur environnement et prennent des décisions afin de maximiser leurs chances de succès. L'intelligence artificielle tend à reproduire le plus fidèlement possible l'intelligence humaine. L'IA a d'ailleurs été fondée sur l'affirmation que l'intelligence humaine peut être décrite tellement précisément qu'elle peut être simulée par une machine.

Ses principaux domaines sont le raisonnement, la connaissance, la planification, l'apprentissage, la communication, la perception et la capacité à déplacer et manipuler des objets.

On retrouve l'intelligence artificielle dans différente application de notre monde actuel, parmi lesquelles :

- La finance pour investir en bourse et organiser les opérations.
- Les hôpitaux et la médecine pour aider les diagnostics médicaux et organiser l'affectation des chambres.
- La gestion d'entreprise avec la business intelligence pour aider au choix stratégique de l'entreprise
- Les jeux vidéo pour rendre les actions des ennemis et des alliés les plus réalistes possible.

2. Utilité pour un assistant

Grâce à l'IA, les assistants peuvent prendre conscience de leur environnement et du contexte dans lequel ils sont exécutés. Cela leur permet ainsi de mieux comprendre les informations qu'ils traitent afin de fournir un résultat plus pertinent. De plus, on va pouvoir rajouter une notion d'apprentissage afin de rendre les systèmes toujours plus performants. Le modèle d'intelligence artificielle le plus utilisé pour réaliser cela et le réseau de neurones.

Un réseau de neurones est un modèle d'IA inspiré du mode de fonctionnement des neurones biologiques. Ils ont des capacités d'apprentissage et de reconnaissance. On peut les représenter comme un système de neurones interconnectés qui, à partir de données d'entrées propagées dans le réseau de neurones, retourne une valeur de sortie (résultat). L'apprentissage est réalisé en modifiant la propagation des informations.

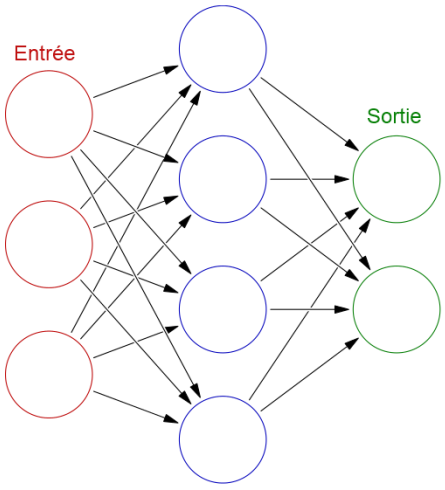


Illustration 14 : Représentation d'un réseau de neurones

Afin d'effectuer la propagation, chaque neurone aura une valeur de seuil et chaque connexion un poids. Lorsqu'on va initialiser les neurones d'entrées, si leur valeur dépasse leur seuil, ils vont propager l'information aux neurones suivant en effectuant un rapport avec le poids de la connexion. Les neurones vont effectuer une somme des valeurs qu'ils reçoivent et si cette dernière dépasse son seuil, il va propager à son tour l'information aux neurones suivants. Ceci va se répéter jusqu'à arriver aux neurones de sorties. Le neurone de sortie ayant une valeur supérieure à son seuil sera le résultat du système.

Concernant la reconnaissance vocale, ce système va permettre de contextualiser la phrase dictée par l'utilisateur permettant ainsi de faire un choix de mots plus pertinent. L'IA va agir sur deux niveaux : le choix des phonèmes et le choix des mots. Pour un phonème, il sera choisi en fonction de sa probabilité d'apparition avec le phonème précédent. Par exemple, le son « ch » sera rarement suivi du son « r », mais plutôt du son « a » ou « o ». Les mots sont aussi choisis en fonction de leurs probabilités d'apparition par rapport au mot précédent, mais aussi suivant le contexte de la phrase et sa signification. Le réseau de neurones permet aussi, grâce à l'apprentissage, d'optimiser le système avec les paramètres de la voix et de lecture de l'utilisateur (accent, intonation, vitesse de lecture, etc.).

Pour la reconnaissance d'image, le réseau de neurones va, d'une part, permettre de simplifier l'algorithme* de classification et surtout, d'autre part d'apporter l'apprentissage de nouveaux objets. En effet, pour savoir quel objet représente une forme, il y aura juste à initialiser les neurones d'entrée avec le paramètre de cette forme puis de propager les informations pour obtenir notre objet à partir des neurones de sorties. Enfin lorsque le système ne parvient pas à reconnaitre la forme, le système demandera à l'utilisateur de quel objet il s'agissait pour ensuite mettre à jour le réseau de neurones avec ce nouvel objet.

3. Ses limites

Même si au début de l'intelligence artificielle, dans les années 1960, on pensait être capable de créer, d'ici une vingtaine d'années, une IA capable de rivaliser avec une intelligence humaine. On sait aujourd'hui que ça n'a pas était le cas et ça ne l'est toujours pas. En effet, apprendre les informations propres au sens commun de l'humain, qui représente une masse de données colossale, à une IA prendrait trop de temps.

Sans ce sens commun, les systèmes de reconnaissance ne pourront donc pas être infaillibles et retourneront toujours une certaine marge d'erreur.

De plus, les réseaux de neurones possèdent une autre contrainte concernant l'apprentissage. En effet, ces derniers ont besoin de plusieurs cycles d'apprentissage pour apprendre et intégrer les nouvelles données. Pour la reconnaissance vocale, cela se présente sous la forme d'une phase de « calibration » nécessitant de lire plusieurs textes différents. En ce qui concerne la reconnaissance d'image, il faudra que le système apprenne plusieurs fois l'objet avant de pouvoir le reconnaitre précisément.

D. Contraintes dues à la mobilité

1. Sécurité

Les téléphones portables contiennent et font transiter de plus en plus de données sensibles et personnelles. Ces données concerneront autant l'utilisateur, avec ses informations personnelles, que l'entreprise ayant développé l'application, avec ses algorithmes* et ses bases de données. La sécurité va donc intervenir à deux niveaux :

- Au niveau de la transmission des données
- Au niveau du stockage des données

a. Transmission des données

Les problèmes de sécurité de transmission des données se posent plus particulièrement pour les systèmes utilisant une architecture orientée serveur, mais aussi pour tous systèmes transférant des données à un autre dispositif. Comme les systèmes de paiement par NFC*.

En effet, pour le cas d'une architecture orienté serveur, les communications entre le mobile et le serveur peuvent être intercepté. Si tel est le cas, la personne malveillante peut alors récupérer les messages envoyés et en extraire les données. Pour cela, différentes techniques peuvent être utilisées telles que le « Man in the middle ». Cette technique consiste à se faire passer pour le serveur et ainsi faire un relai entre le mobile et le serveur. Chaque communication passant donc forcément par l'ordinateur du pirate informatique.

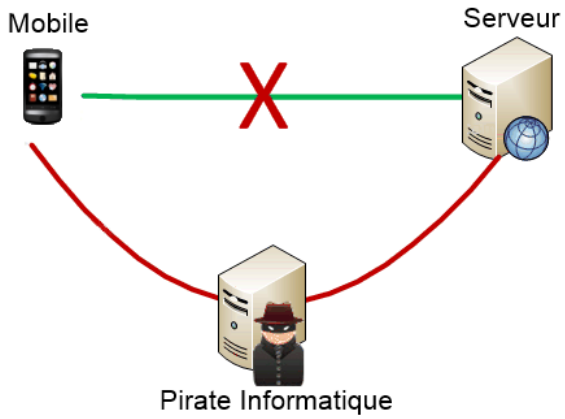


Schéma 18 : Schéma d'une attaque « Man of the Middle »

L'interception de données avec le NFC* et sa faible portée est plus compliquée à mettre en place. De plus, même si le pirate arrive à positionner un lecteur NFC* assez proche du téléphone de l'utilisateur, il ne pourra pas forcément récupérer les informations de l'utilisateur. En effet, pour cela il faut que ce dernier ait lancé le processus d'envoi d'information sur son téléphone, par exemple la procédure de paiement par NFC*.

Cependant, toutes les données et tous les systèmes ne sont pas concernés par ces problèmes de sécurité. En effet, il convient de sécuriser les informations proportionnellement à leur criticité. Ses informations bancaires devant, par exemple, être plus protégées que sa position GPS. De même que certaines informations n'ont pas besoin d'être sécurisées comme la recherche de points d'intérêts qui ne constitue pas une donnée sensible.

b. Stockage des données

Les Smartphones, proposant toujours plus de fonctionnalités, dont font partie les assistants mobiles, sont destinés à contenir d'autant plus d'informations. Seulement, en cas de vol, toutes ces informations seront à disposition du voleur. Il pourra simplement lancer les applications et ainsi accéder aux données contenues dans celle-ci. Cela lui permettra, par exemple :

- D'accéder au service de votre banque et de consulter votre compte
- De consulter vos itinéraires à partir de l'application de Navigation
- Ou encore d'effectuer un paiement via NFC*, s'il n'est pas protégé par code

Les données contenues dans un téléphone portable peuvent aussi être détournées à des fins d'espionnage industriel. Grâce au « reverse engineering », on peut effectivement récupérer les algorithmes* et les bases de données des concurrents représentant le cœur de leur savoir-faire. Les bases de données seront autant intéressantes pour leurs contenues que pour leurs organisations.

Les applications, les bases de données, ainsi que toutes autres données de l'application, vont pouvoir être récupérées grâce à des outils permettant de parcourir le contenu du téléphone.

Le « reverse engineering » que l'ont peur traduire en ingénierie inversée, consiste à retrouver le fonctionnement d'un logiciel à partir de son exécutable. Des outils sont d'ailleurs disponibles pour effectuer ce traitement sur toutes les plateformes mobiles. Une fois qu'on dispose d'un de ces outils, il suffit de copier l'application de son téléphone portable vers son ordinateur et de l'importer dans l'outil d'ingénierie inversé. Ensuite, l'outil va reconstituer le code source de cette application. Le code source reconstitué ne sera pas exactement identique à l'original, mais les algorithmes*, eux, seront identiques.

2. Environnement

L'environnement dans lequel évolue l'assistant mobile peut aussi présenter quelques contraintes. Tout d'abord, de par sa nature mobile, l'assistant est destiné à être utilisé dans divers environnements dont la plupart ne sont pas forcément adaptés. Par exemple, on peut se retrouver dans une zone non couverte par le réseau. Dans ce cas, tous les services en ligne ne seront pas accessibles. Cela pouvant restreindre :

- la réception d'information en direct sur les conditions de circulation sur l'application de Navigation GPS
- l'accès à ses informations bancaires
- ou encore la recherche de points d'intérêts à proximité.

On retrouve aussi des environnements particulièrement contraignants pour la reconnaissance vocale. On peut citer le passage dans une zone bruyante, tel qu'une zone de travaux, une gare ou bien dans un transport en commun.

Le bruit complique évidemment la tâche d'isolation de la voix et donc diminue la précision du système allant jusqu'à ne plus fonctionner. Une autre situation est son utilisation dans un lieu public, ou un endroit fréquenté. En effet, cela nous oblige à partager avec tout le monde ce qu'on dit à notre assistant et ce que l'ont fait avec. Ce qui n'est pas forcément agréable ni pour nous ni pour notre environnement.

Ensuite, certains assistants vont dépendre du déploiement d'une technologie. C'est par exemple le cas du paiement mobile par NFC*. Les constructeurs de téléphone portable et les commerçants ne sont pas encore prêts à investir tant qu'ils ne savent pas comment cette technologie va être reçue par le grand public. Ainsi on retrouve cette technologie principalement sur les téléphones haut de gamme et chez les commerçants ayant un partenariat avec un service de paiement bancaire. On comprend donc pourquoi ce sont des processus qui prennent du temps. Les investisseurs attendant un retour positif du public sur la technologie avant d'effectuer un déploiement et le public n'étant pas prêt à s'engager tant que la technologie n'est pas largement présente.

III. Solution et Évolution de l'assistant mobile

Dans cette partie, je propose des solutions à certaines limitations évoquées dans le chapitre précédent. J'ai décidé de me concentrer sur les solutions concernant les problèmes liés à la diversité des systèmes d'exploitation mobile, aux temps de réponse des architectures orientées serveurs et à la sécurité. À la suite de cela, je détaille une évolution de l'assistant mobile qui va redéfinir la vision d'un tel système.

A. Solutions techniques

1. Diversité des systèmes d'exploitation: Programmation multiplateforme

Comme on l'a vu, la diversité des systèmes d'exploitation contraint à développer une application pour chacun d'entre eux. Seulement, il existe une solution à ce problème : le développement multiplateforme. Cette dernière a pour but d'offrir la possibilité de développer une seule version de l'application qui pourra être ensuite exécutée sur toutes les plateformes. On distingue deux méthodes : les applications Web et le partage de code.

a. Les Applications Web

Les applications Web vont permettre de réaliser des applications multiplateformes grâce aux technologies Web. L'application sera donc écrite avec le trio HTML5*, CSS* et JavaScript*. Elle peut donc être utilisée sur n'importe quel appareil pouvant afficher une page Web. Seulement, en l'état, elle ne pourra pas accéder aux fonctionnalités spécifiques du téléphone telles que le récepteur GPS, le magnétomètre ou encore la liste des contacts. De telles applications peuvent quand même être utilisées, mais pour des taches simples ne nécessitant pas de solliciter les données ou composants du téléphone.

Afin de pallier à ce problème, il existe des Frameworks qui se spécialisent dans le développement d'application Web mobile. Ces Frameworks vont ainsi servir d'interfaces entre le mobile et l'application Web pour accéder aux différents composants du téléphone. Ces derniers génèrent de véritables applications. L'application Web sera donc embarquée dans l'application et le Framework faisant le lien entre les deux. Cela permet ainsi de pouvoir envoyer son application sur les magasins d'application des différentes plateformes.

L'application aura donc la même visibilité qu'une application mobile et pourra même être proposée payante pour générer du chiffre d'affaires.

Un des Frameworks les plus utilisés est PhoneGap. Ce dernier supporte iOS, Android, Windows Phone, BlackBerry, Symbian*, Bada*, HP WebOS*, Qt* et Tizen*. Il permet l'accès à différents composants et fonctionnalités du téléphone portable telles que : l'accéléromètre, l'appareil photo, la caméra, le microphone, le magnétomètre, le récepteur GPS ou encore la liste de contact. Ci-dessous, sur l'illustration 16, un exemple d'application ayant était développé à l'aide de PhoneGap pour les plateformes Android, iOS, Windows Phone et BlackBerry.

Illustration 15 : Application « Untappd » développée avec PhoneGap

Cependant, cette solution n'est pas adaptée à toutes les applications. En effet, il existe certaines limitations liées aux Frameworks et aux technologies Web. Tout d'abord concernant les Frameworks, ils ne supportent pas toutes les fonctionnalités avancées spécifiques à chaque plateforme. De plus, lorsqu'une nouvelle fonctionnalité est disponible pour une plateforme, il est nécessaire d'attendre que le Framework soit mis à jour pour pouvoir l'utiliser. Ensuite, les technologies Web ont quelques lacunes comparées aux technologies natives des plateformes. En effet, elles seront moins performantes concernant le traitement des données et l'affichage graphique avancé (2D et 3D) et ne permettent pas un fonctionnement multitâche.

b. *Le partage de code*

Le partage de code ne permet pas de se passer de développement spécifique à chaque plateforme, mais permet d'optimiser le processus de production. Pour cela, on regroupe au maximum le code qui sera commun à toutes les versions. On retrouve principalement toute la logique et les classes métier de notre application, la communication avec les serveurs et l'accès aux bases de données. Ce code se retrouvera alors en commun sur chaque application et évitera de refaire le même travail pour chaque plateforme. On pourra réaliser une librairie* à partir de ce code pour faciliter l'intégration et la maintenance de ce code. Il reste alors à effectuer le développement spécifique à chaque plateforme. C'est-à-dire, l'interface utilisateur et l'accès aux composants et fonctionnalités du mobile (récepteur GPS, liste de contact, caméra, etc.).

Le partage de code ne nécessite pas de Framework spécial, il suffit juste de bien organiser son code ou chaque classe à sa responsabilité. Si une version a déjà était développé sur une plateforme en suivant cela, l'extraction du code partagé ne pose pas de soucis. Cependant, pour que ce code soit exécutable par toutes les plateformes, il faut qu'elle soit développée dans un langage qu'elles supportent. Le choix se portera sur les langages C*/C++* supportés de façon native par la plupart des plateformes, à savoir : iOS, Windows Phone et BlackBerry. Pour Android, il y a deux solutions. Soit, on effectue un portage du code en Java* pour qu'ils soient compatibles. Soit, on utilise la bibliothèque JNI, pour Java Native Interface, qui permet de réaliser une interface entre le code Java* et celui d'un autre langage. On pourra ainsi réutiliser le code partagé en C*/C++* avec l'application Android.

Comme on l'a vu, cette technique ne permet pas de supprimer complètement le code spécifique à chaque plateforme, mais permet de réaliser un bon compromis. Certes, d'un côté, on doit passer plus de temps à développer les spécificités de chaque version, mais de l'autre on en tire plusieurs avantages. Tout d'abord, cela permet de proposer une interface utilisateur personnalisée et en adéquation avec chaque système d'exploitation*. On peut aussi tirer parti de tous les avantages de chaque plateforme permettant ainsi de rester à jour et de proposer des fonctionnalités toujours plus innovantes. Et surtout, on a une application disposant des meilleures performances possibles grâce à l'utilisation du code natif à chaque plateforme et du C*/C++*, reconnue pour leur efficacité, pour le code partagé.

2. Amélioration du temps de réponse des architectures orientées serveur : la 4G

Le problème majeur de l'utilisation d'une architecture orientée serveur est son impact sur le temps de réponse du système. En effet, le temps de communication entre le mobile et le serveur augmente d'autant le temps de réponse. Pour pallier à cela, il faudrait donc augmenter la vitesse de transfert, donc augmenter le débit. Seulement, on ne peut pas agir sur ce débit. Ce dernier dépend de la technologie de transmission de donnée téléphonique mobile. Cependant, on assiste, depuis la fin de l'année 2012, au déploiement de technologie mobile de 4ème génération : la 4G. Cette dernière devant, en théorie, grandement augmenter le débit du réseau téléphonique mobile.

a. La 4G

La 4G est la quatrième génération du standard des technologies de communication mobile pour téléphone portable. Elle ne représente donc pas une technologie à proprement parler. Ce standard est défini par l'« International Telecommunications Union-Radio communication sector » (ITU-R*) qui est le secteur des radios communications de l'union international des télécommunications. Cet organisme a donc défini, en mars 2008, une spécification pour le standard 4G nommé « International Mobile Telecommunications Advanced » (IMT-Advanced) que l'on peut traduire en télécommunication mobile internationale avancée. Les principales exigences de cette spécification sont :

- Une technologie complètement basée sur IP*, comme pour Internet, et donc de l'abandon du fonctionnement en mode commuté où un commutateur relié la ligne téléphonique de l'appelé à celui de l'appelant.
- Un pique de débit pouvant atteindre les 100 Mégabits par seconde en condition de grande mobilité, en voiture ou en train par exemple, et jusqu'à 1 Gigabit par seconde en condition de faible mobilité, stationnaire ou marche à pied.
- Une utilisation et un partage dynamique des ressources réseau afin de supporter plus d'utilisateurs simultanés par cellule*.
- Bascule fluide d'une cellule* à une autre sur un réseau hétérogène.
- La capacité d'offrir une haute qualité de service pour les supports multimédias de nouvelle génération (TV mobile haute définition, vidéoconférence, télévision 3D, etc.).

Malgré ces exigences, deux technologies : LTE et WiMAX, établies avant les spécifications IMT-Advanced et ne répondant pas totalement à ses exigences, ont reçu l'approbation pour l'utilisation commerciale du terme 4G. Cette exception est due au fait que malgré des débits inférieurs à ces exigences, elles apportent un niveau substantiel d'amélioration de performances comparées aux autres technologies de troisième génération. Cependant, une nouvelle version de ces technologies, LTE-Advanced et Gigabits WiMAX, supporte pleinement les exigences de la 4G.

Concernant le déploiement de ces technologies, le LTE-Advanced et le Gigabits WiMAX sont encore peu répandus, mais nécessitent uniquement une mise à jour des appareils LTE et WiMAX. La technologie WiMAX, avec seulement quelques téléphones compatibles, reste encore très peu présente pour la téléphonie mobile. Par contre, la technologie LTE est en plein déploiement sur la plupart des continents comme on peut le voir dans l'illustration 17. Bien qu'en France les premières offres LTE n'aient été disponibles qu'en début 2013, au niveau mondial 58 millions d'abonnés utilisaient déjà cette technologie à la fin 2012.

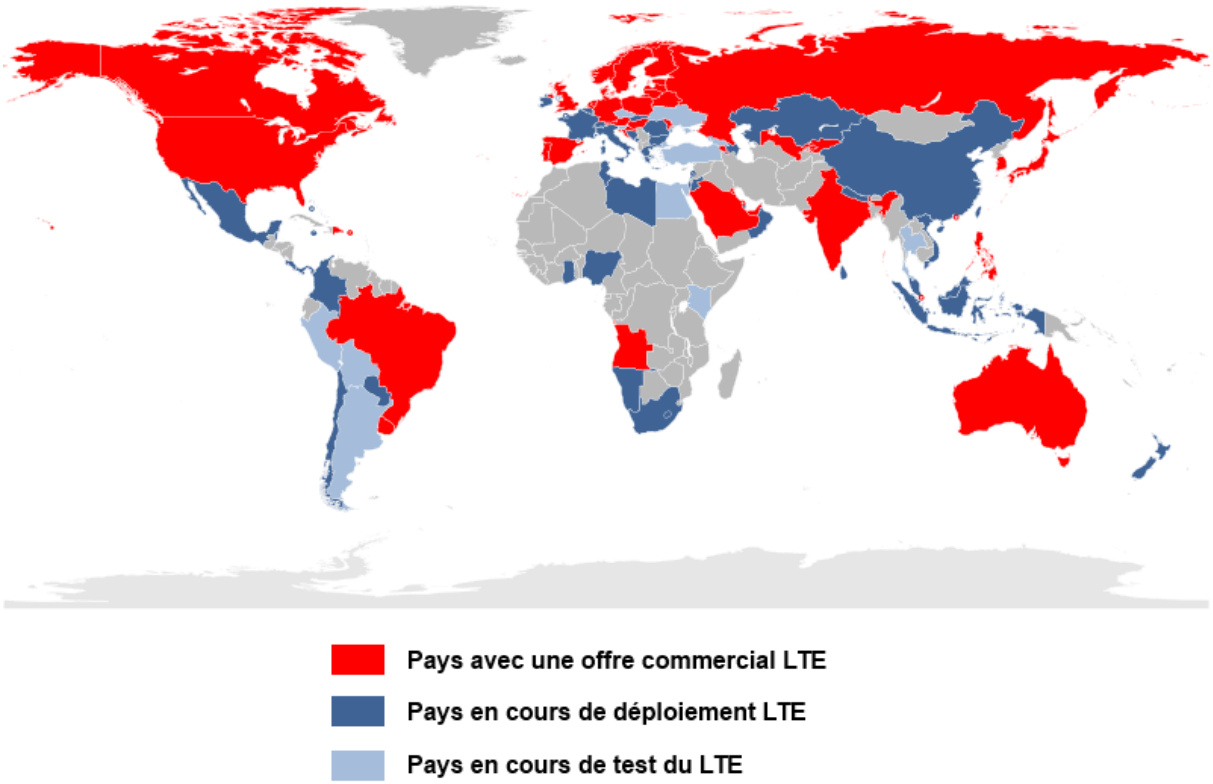


Illustration 16 : Carte de la couverture de la technologie LTE dans le monde

b. *Amélioration*

Maintenant qu'on a vu que ces nouvelles technologies de communication de quatrième génération offraient des débits élevés, je vais les comparer avec ceux des anciennes générations. On distingue deux types de débit. Le débit descendant qui correspond au débit de réception, dans notre cas cela correspond à une réponse du serveur. Et le débit ascendant correspondant au débit d'envoi, généralement inférieur au précédent, et qu'on associe à l'envoi de requête et de données au serveur.

Ci-dessous, un tableau comparatif des débits des différentes générations de technologie de communication mobile. Les débits présentés sont des débits théoriques. Je me suis basé sur ces derniers, car les débits réels sont complexes à déterminer et extrêmement variables suivant les conditions d'utilisation. Il faut noter que chaque technologie est affectée de la même manière par ces conditions.

	2G EDGE	3G HSPA	4G commercial	4G standardisé
Débit descendant	384 Kb/s	7,2 Mb/s	300 Mb/s	1 Gb/s
Débit ascendant	60 Kb/s	2 Mb/s	75 Mb/s	500 Mb/s
Temps téléchargement d'une photo (2 Mo)	41 sec	2 sec	0,05 sec	0,016 sec
Temps d'envoi d'une photo (2 Mo)	4 min 26 sec	8 sec	0,21 sec	0,032 sec

Tableau 1 : Comparatif des débits théoriques suivant la génération des technologies de communication mobiles

Comme on le voit dans ce tableau, le passage de la 3G à la 4G permet une augmentation majeure en termes de débit. Ainsi, on obtient un débit 40 fois plus rapide pour de la 4G commerciale, et 140 fois plus rapide avec la 4G standardisé comparé à la 3G. Cela permettant, avec les débits maximaux théoriques, de télécharger et d'envoyer une image de 2 mégaoctets soit une image de 5 à 8 mégapixels, en moins d'une demi-seconde.

Même s'il va falloir attendre que cette technologie soit plus largement déployée, autant sur les Smartphones que sur les infrastructures des opérateurs téléphoniques, cette dernière est une grande avancée pour les applications utilisant une architecture orientée serveur.

En effet, avec un débit réel attendu de quelques dizaines de mégabits par seconde, la plupart des communications entre le mobile et le serveur se feront en moins d'une seconde. Ceci permettant de grandement réduire le temps de réponse de ce type d'architecture.

B. Sécurisation des données

Dans cette partie, je vais présenter des solutions pour la sécurisation des données. On distingue deux types de données à protéger. Les premières sont celles qui sont transférées pendant les communications entre le mobile et le serveur. Et les deuxièmes sont celles présentes sur le mobile : les données et bases de données de l'application, ainsi que l'application elle-même, en particulier ses algorithmes*.

1. Données issues des communications mobile / serveur

Les données transférées entre le mobile et le serveur peuvent être très sensibles, en particulier les applications bancaires ou de commerce. Afin de protéger ces données, il faut donc les crypter. La meilleure solution et la plus sécurisée est l'utilisation de la technologie SSL ou plutôt son successeur TLS.

a. TLS

TLS, pour Transport Layer Security, que l'on peut traduire par sécurité de la couche transport, est un protocole de sécurisation des échanges par Internet. Développé à l'origine par Netscape, sous le nom de SSL (Secure Sockets Layer), il a était renommé en TLS suite au rachat du brevet par l'IETF* (Internet Engineering Task Force). Le SSL a été développé à l'origine afin de répondre aux problèmes de sécurité des débuts du paiement en ligne. Aujourd'hui, il est majoritairement utilisé pour toutes les transactions réalisées sur Internet en particulier via le protocole HTTPS, qui est du HTTP auquel on rajoute une couche SSL. TLS répond à plusieurs objectifs de sécurité :

- l'authentification du serveur pour que le client soit sûr de communiquer avec le bon serveur.
- la confidentialité des données échangées (ou session chiffrée) grâce au cryptage pour que personne d'autre ne puisse lire ces données.
- l'intégrité des données échangées afin de s'assurer que les données reçues sont identiques à celles envoyées.

- de manière optionnelle, l'authentification du client pour que le serveur soit sûr de communiquer avec des clients autorisés.
- la spontanéité, c'est-à-dire qu'un client peut se connecter de façon transparente à un serveur auquel il se connecte pour la première fois.
- la transparence : les protocoles de la couche d'application (du modèle OSI*) n'ont pas à être modifiés pour utiliser une connexion sécurisée par TLS. Par exemple, le protocole HTTP est identique, que l'on se connecte en http ou https.

Pour réaliser tous ces objectifs, TSL se base sur différent technique de cryptage. La partie authentification se fera grâce à un chiffrement asymétrique alors que toute la partie de cryptage des données est réalisée avec un chiffrement symétrique. Un code d'authentification de message, ou MAC (Message Authentication Code), basé sur des fonctions de hashage* est aussi utilisé pour contrôler l'intégrité et l'authenticité des données. Pour authentifier le serveur de manière sûre, il utilise un certificat qui représente l'identité du serveur.

Un certificat est un ensemble de données, dont au moins une clé publique, des informations d'identification, par exemple : noms, localisation, emails et au moins une signature. La signature sert à assurer que ce certificat a été validé par un tiers de confiance*, c'est-à-dire que le certificat correspond bien au serveur décrit. Dans notre cas, on signera nous-mêmes notre certificat, car nous maitrisons le serveur comme le client, on pourra donc vérifier nous-mêmes la validité du certificat sur l'application mobile.

i) Chiffrement symétrique

Le chiffrement symétrique est une des plus anciennes formes de chiffrement. Il se base sur une clé secrète qui (traitée par un algorithme*) permet de chiffrer et déchiffrer un message. Pour qu'il soit sûr, on doit pouvoir dévoiler l'algorithme* de chiffrement sans que le message puisse être déchiffré sans connaitre la clé. De plus, il faut que la clé puisse prendre suffisamment de valeur pour qu'un essai de toutes les clés soit trop long à réaliser.

Une méthode simple est d'effectuer un décalage de lettre sur le message original suivant la position des caractères de la clé dans l'alphabet. Prenons par exemple, le texte « message » à crypter avec la clé « crypto ».

La première lettre à crypter est le « m », qui correspond à la treizième lettre de l'alphabet, et le premier caractère de la clé est « c », la troisième lettre. Ainsi : 13 + 3 = 16, le premier caractère de notre message crypté sera le « p », la treizième lettre de l'alphabet. Lorsqu'on arrive à la fin de la clé, on repart du début. En continuant ainsi, on obtient :

Message	M	E	S	S	A	G	E
Clé	C	R	Y	P	T	O	C
Message crypté	P	W	R	I	U	V	H

Tableau 2 : Exemple de chiffrement symétrique simple

Ce n'est évidemment pas ce type d'algorithme* simple que l'on va utiliser, mais plutôt l'algorithme* de chiffrement AES (Advanced Encryption Standard ou standard de chiffrement avancé). C'est un standard du chiffrement symétrique pour le gouvernement américain, notamment utilisé pour les données top secrètes de la NSA. Cet algorithme* n'a pas encore était cassé et seule une recherche exhaustive pourrait le faire, mais trop longue a exécuté. L'algorithme* chiffre le message par bloc de 16 octets et utilise une clé de 128, 192 ou 256 bits. AES effectue plusieurs cycles suivant la taille de la clé, 10 cycles pour la 128 bits, 12 cycles pour la 192 bits et 14 cycles pour la 256 bits. Pour crypter un message, il copie les 16 octets dans une matrice de 4 par 4. Il effectue ensuite, pour chaque cycle, une série d'opérations sur cette matrice. Pour le décryptage, il effectue l'opération inverse.

ii) Chiffrement asymétrique

Le chiffrement asymétrique est une méthode de chiffrement utilisant deux clés pour son fonctionnement. Une clé, dite publique, qui sera diffusée et qui servira à crypter les messages. Et une clé, dite privée, qui sera gardée secrète et qui permettra de décrypter les messages. Les deux clés sont liées mathématiquement afin qu'avec la clé privée, et uniquement elle, on puisse décrypter un message crypté avec la clé publique. La clé privée ne pouvant être déterminée à partir de la clé publique. Le chiffrement asymétrique sert dans la plupart des cas à sécuriser l'échange de clés d'un algorithme* de chiffrement symétrique.

L'algorithme* de chiffrement asymétrique retenu est le RSA, des initiales de ses trois inventeurs : Rivest, Shamir et Adleman. C'est un des algorithmes* de chiffrement asymétrique les plus utilisé.

La taille des clés RSA est au libre choix de son utilisateur, même si la plupart varient entre 1024 et 2048 bits. Une clé de 1024 bits est considérée comme sûre étant donné qu'une telle clé n'a pas encore été cassée. Cependant par sécurité, il est couramment recommandé d'utiliser des clés de 2048 bits. RSA utilise des principes mathématiques comme la congruence, l'indicatrice d'Euler, les modulos, les exposants et les nombres premiers pour générer ses clés et effectuer le cryptage et le décryptage. Sa sécurité est basée sur un problème de factorisation. En effet, si on prend deux très grands nombres premiers (plusieurs centaines de chiffres) « p » et « q » et qu'on les multiplie entre eux pour obtenir un chiffre « n ». Il est alors impossible de retrouver « p » et « q » à partir de « n ».

iii) Code d'authentification de message

Un code d'authentification de message est un code accompagnant des données dans le but d'assurer l'intégrité de ces dernières, en permettant de vérifier qu'elles n'ont subi aucune modification, après une transmission par exemple. Il s'agit d'algorithmes* qui créent un petit bloc authentificateur de taille fixe se basant sur le message et sur une clé secrète. Les MAC n'ont pas besoin d'être réversibles. En effet, le récepteur exécutera le même calcul sur le message et le comparera avec le MAC reçu. Le MAC assure non seulement une fonction de vérification de l'intégrité du message, comme le permettrait une simple fonction de hachage, mais de plus authentifie l'expéditeur, détenteur de la clé secrète.

Pour le TSL, on utilise plus précisément le HMAC (keyed-Hash Message Authentication Code). Un HMAC est donc un type de code d'authentification de message calculé en utilisant une fonction de hashage* cryptographique en combinaison avec une clé secrète. Un code HMAC est calculé de la manière suivante :

$$HMAC(m, K) = h((K \oplus opad) \,||\, h(K \oplus ipad \,||\, m))$$

Avec :

- h : fonction de hashage*
- K : clé secrète
- m : le message
- ipad et opad sont définis par une répétition de valeur hexadécimale 0x36 pour ipad et 0x5c pour opad
- || : désigne une concaténation
- $\oplus$: désigne un ou exclusif

Une des fonctions de hachage les plus utilisé est le SHA-256. Une fonction de hashage* détermine, à partir d'une donnée en entrée, une empreinte unique à cette donnée. Une empreinte ne devant pas pouvoir correspondre à deux données différentes. Des plus, elle ne doit pas rendre possible le retour à la donnée initiale.

Le SHA-256 a était conçue par la NSA et répond à toutes les contraintes d'une fonction de hashage*. Il supporte des messages ayant une taille maximum de 2^{64} bits et retourne une empreinte de 256 bits. Le SHA-256 découpe le message en bloc de 512 bits et pour chaque bloc effectue 64 itérations de la fonction de compression. La fonction de compression effectue des opérations de « et » logique, de « ou » logique, de « ou » exclusif, de décalage de bits et d'addition entre les bits du bloc et des valeurs constantes prédéfinies.

b. Fonctionnement

La grande partie du protocole TLS s'effectue lors de l'initialisation de la connexion. C'est dans cette phase qu'ont lieu l'authentification et l'échange de clés. Une fois cela réalisé, la connexion sécurisée est établie et l'échange de messages crypté peut commencer. Voici la procédure d'initialisation d'une connexion TLS :

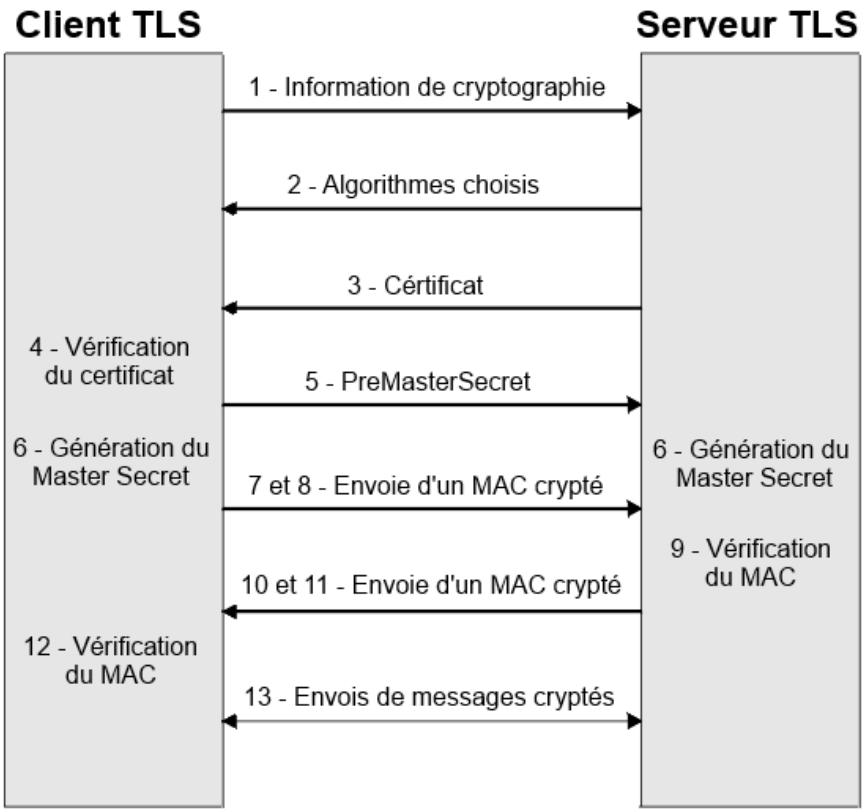


Schéma 19 : Procédure d'établissement d'une connexion TLS

1. Le client envoie au serveur, la plus haute version de TLS supporté, un nombre aléatoire et la liste des algorithmes* de chiffrement supporté pour l'échange de clés, la génération de la clé, le cryptage des données et le code d'authentification du message.
2. Le serveur répond avec la version de TLS choisi, un nombre aléatoire et la liste des algorithmes* sélectionnée.
3. Le serveur envoie son certificat et un message indiquant la fin de la phase de « négociation ».
4. Le client vérifie le certificat du serveur, s'il n'est pas valide il ferme la connexion.
5. Si le certificat est valide, le client répond avec un « PreMasterSecret » crypté avec la clé publique du serveur.
6. Le client et le serveur utilisent les nombres aléatoires envoyés au début de la communication et le « PreMasterSecret » pour générer la clé appelée « master secret ». Toutes les autres clés utilisées pendant cette connexion seront dérivées de cette « master secret ».
7. Le client envoie un message pour dire qu'à partir de maintenant, toutes les communications seront encryptées et authentifiées.
8. Le client envoie un message authentifié et crypté contenant un code d'authentification de message (MAC) de tous les messages précédents.
9. Le serveur essaye de décrypter le message et de vérifier le MAC. S'il n'y arrive pas, il ferme la connexion.
10. Le serveur envoie un message pour dire qu'à partir de maintenant, toutes les communications seront encryptées et authentifiées.
11. Le serveur envoie un message authentifié et crypté contenant un MAC de tous les messages précédents.
12. Le client effectue le décryptage et la vérification du MAC, s'il y a une erreur, il ferme la connexion.
13. La connexion est effectuée, le serveur et le client peuvent maintenant communiquer de manière sécurisée.

c. *Mise en place*

La mise en place d'un tel protocole nécessite une intervention sur le serveur et sur l'application mobile. Du côté du serveur, il faut tout d'abord installer un outil implémentant le protocole TLS, comme OpenSSL*. Ensuite, on génère les clés privée et publique ainsi que le certificat du serveur. Enfin, il suffira de configurer l'outil TLS pour qu'il s'interface avec l'application serveur.

Pour le mobile, des librairies implémentant le protocole SSL existent déjà sur toutes les principales plateformes. Il faut cependant effectuer quelques traitements lors de la réception du certificat du serveur. On va d'abord, vérifier que le certificat reçu est bien celui de notre serveur. Pour cela, la solution retenue est de stocker une empreinte du certificat du serveur, crypté en SHA-256, que l'on va comparer à l'empreinte SHA-256 du certificat du serveur. Si le certificat est valide, l'application devra enregistrer la clé publique contenue dans le certificat pour les futures communications.

Il faut noter qu'une connexion effectuée en TLS est très sécurisée. Un tel niveau de sécurité est utile pour les opérations bancaires, mais pas forcément nécessaire pour tous les assistants. Si l'assistant n'effectue pas d'opérations bancaires, mais qu'une protection est quand même nécessaire, on pourra crypter ses données avec un algorithme* RSA. Il faut cependant faire attention à l'impact sur le temps de réponse du système causé par le cryptage RSA et s'en servir uniquement pour les données critiques. L'utilisation d'un algorithme* à chiffrement asymétrique est à préconiser. En effet, avec un algorithme* à chiffrement symétrique, la clé devra être contenue dans l'application et donc sera potentiellement disponible pour tous.

2. L'Application et ses données

Comme vu précédemment, l'application et ses données sont des éléments importants à sécuriser pour que l'entreprise se protège de l'espionnage industriel. En effet, lorsque l'application est installée sur un mobile, elle en devient vulnérable. Le meilleur moyen d'éviter cette problématique est d'opter pour une architecture orientée serveur, lorsque cela est possible. Tous les éléments critiques de l'application se retrouveront donc sur le serveur et ne pourront donc pas être accédés aussi simplement que sur le téléphone.

Seulement cette solution n'est pas applicable à tous les systèmes. En effet, certains ayant des contraintes de temps de réponse ou de budget ne peuvent pas se permettre l'utilisation d'une telle architecture. Ainsi, je vais présenter, dans un premier temps, une solution pour la sécurisation de l'application, puis dans un second temps une autre pour les données.

a. Sécurisation de l'application

Sécuriser le code d'une application pour qu'il soit inaccessible pour une personne malveillante est impossible, du moins avec les technologies d'aujourd'hui. En effet, une personne ayant assez de compétences en reverse engineering et de volonté pourra, dans tous les cas, récupérer le code de l'application. Il existe seulement certaines techniques permettant de compliquées la tâche de ces personnes voir de les dissuader. La solution que je vais présenter permet d'obtenir un niveau satisfaisant de protection tout en nécessitant un investissement minime. Cette solution est l'obfuscation.

L'obfuscation d'un code informatique est un procédé ayant pour but de rendre le code illisible et incompréhensible par l'humain tout en restant compilable, et donc exécutable, par un ordinateur. C'est une technique très peu couteuse, facile à mettre en place et totalement transparente pour l'utilisateur et le développeur. Il existe des outils permettant l'obfuscation automatique de code, ils sont appelés obfuscateur. L'obfuscation agira principalement sur le style du code, les données écrites dans le code (constantes, valeurs d'initialisation, etc.) ou encore la structure de l'application.

Les modifications du style de code permettront de modifier l'aspect général du code. Pour réaliser cela, on pourra transformer les identifiants (nom des variables, méthodes et classes), supprimer les commentaires ou modifier les espaces et tabulations. Par exemple, le code suivant :

```
public synchronized void put(int key, Employee value) {
        Integer I = new Integer(key);
        super.put(I, (Object) value);
}
```

, deviendra, par transformation des identifiants :

```
public synchronized void a(int a, b c) {
        Integer d = new Integer(a);
        super.a(d, (Object) c);
}
```

La principale méthode pour obfuscer les données du code consiste à crypter de manière simple ces dernières. La clé et les fonctions de cryptage et de décryptage seront par la suite noyées dans le code obfuscé.

Enfin, pour modifier la structure de l'application, l'obfuscateur peut agir sur l'architecture des classes de notre application en scindant ou regroupant différentes classes ou encore en ajoutant des classes et méthodes inutilisées. Par exemple, comme on le voit dans l'illustration 18, les classes Personne et Employé pourront être découpé en plusieurs classes : PersonneDeBase, Habitant, HabitantTelecom, Internaute, Employé et Salarié.

Il sera évidemment intéressant de combiner toutes ces techniques afin d'obtenir un code le plus illisible possible. Pour plus d'exemples de méthode d'obfuscation voir l'Annexe I.

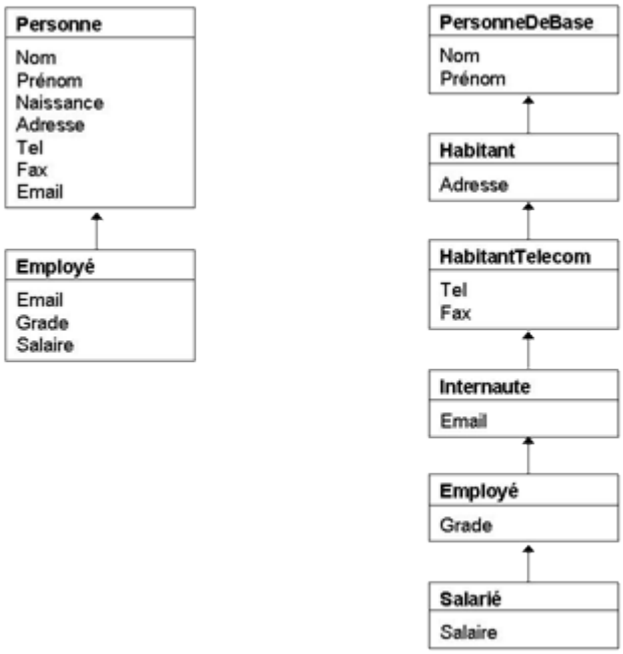


Illustration 17 : Exemple d'obfuscation par modification de structure de l'application

b. Sécurisation des données et base de données

Comme pour l'application, la sécurisation complète des données et base de données est impossible. On pourra malgré tout obtenir un niveau de sécurité convenable en cryptant ces dernières et en utilisant l'obfuscation pour noyer la clé et les méthodes de cryptage et décryptage dans le code. Cette technique n'est cependant pas assez sécurisée pour protéger de données de haute criticité. Pour ces dernières, leur stockage sur un serveur est le seul moyen proposant assez de sécurité.

Le cryptage sera basé sur un chiffrement symétrique étant donné que l'application devra chiffrer et déchiffrer les données avec la même clé. Le choix de l'algorithme* se fera suivant la criticité des données et le temps d'accès aux données minimums nécessaires au bon fonctionnement de l'application. Cependant, l'algorithme*, très sécurisé, AES pourra convenir à la plupart des applications. En effet, l'impact de ce dernier sur les temps d'accès aux données reste relativement faible, assez pour avoir l'air instantané aux yeux de l'utilisateur.

C. Nouvelle vision de l'assistant mobile avec l'agrégation de donnée

Dans cette partie, je présente une nouvelle forme d'assistant mobile qui se base sur l'agrégation de données. Pour cela, je vais d'abord définir la notion d'agrégation de données. Ensuite, je vais expliquer en quoi l'agrégation de données apporte une nouvelle vision de l'assistant. Et pour finir, je décrirais l'architecture d'un tel assistant en détaillant les composants principaux.

1. L'agrégation de données : définition

L'objectif de l'agrégation de données est de répondre à un problème d'interprétation des données. En effet, dans la plupart des cas, une donnée brute et isolée ne permet pas d'en extraire une signification. Par contre, si on place cette donnée dans son contexte, on va pouvoir transformer cette simple donnée en information signifiante. Seulement, ce contexte est souvent représenté par une masse de données importante qui ne permet pas une interprétation simple. En regroupant ces données selon des axes d'analyses pertinents, on va pouvoir réduire le volume de donnée et ainsi simplifier leurs interprétations. C'est ce processus que l'on appelle agrégation de données.

L'agrégation de données correspond donc à la recherche et au regroupement d'un lot de données afin d'en obtenir un résultat synthétique. On peut faire l'analogie avec la comptabilité d'une entreprise. Où toutes les opérations comptables seules n'ont que peu d'intérêt pour la direction. Seulement, en agrégeant ses opérations on va pouvoir obtenir le Compte de résultat et le Bilan qui eux sont des informations décisives pour la direction.

Il faut cependant noter qu'une fois l'agrégation réalisée, on ne peut pas revenir au niveau de détail des données de départ. En effet, pour reprendre l'exemple de la comptabilité, en ne possédant que le Bilan et le Compte de résultat on ne peut pas retrouver toutes les opérations comptables.

2. L'agrégation de données au service de l'assistant mobile

a. Principe de fonctionnement

L'agrégation de données au sein d'un assistant mobile va permettre à ce dernier de percevoir l'univers de l'utilisateur grâce à sa « vie virtuelle ». Cela lui permet de connaitre le contexte de son utilisation et ainsi de fournir les informations les plus pertinentes dans ce contexte.

C'est une nouvelle vision de l'assistant mobile, car ce dernier ne va pas se spécialiser dans un domaine précis et nécessiter une intervention de l'utilisateur pour accéder à l'information voulue. Au lieu de cela, ce dernier va proposer une multitude de services, mais uniquement lorsque vous en aurez besoin. De plus, cet assistant, connaissant le contexte dans lequel est utilisé le service, pourra l'initialiser avec les valeurs dont vous avez besoin, par exemple votre destination lors d'un trajet en voiture.

Pour cela, cet assistant va agréger d'une part les différentes données de la vie numérique de l'utilisateur et d'autre part celles présentes sur son mobile. C'est l'agrégation de ces deux types de données qui vont permettre de proposer un service le plus pertinent possible.

Ce type d'assistant va nécessiter une architecture orientée serveur (voir Schéma 20). La partie serveur effectuera, d'une part, la collecte et l'agrégation périodiques des données pertinentes de tous ses utilisateurs. Et d'autre part, lorsqu'un utilisateur lancera l'application sur son mobile, le serveur lui transmettra les informations et services à proposer après les avoir calculés en fonction des données agrégées. La partie mobile servira à envoyer au serveur les informations présentes sur le mobile et issue des capteurs et à afficher les informations et services reçus du serveur.

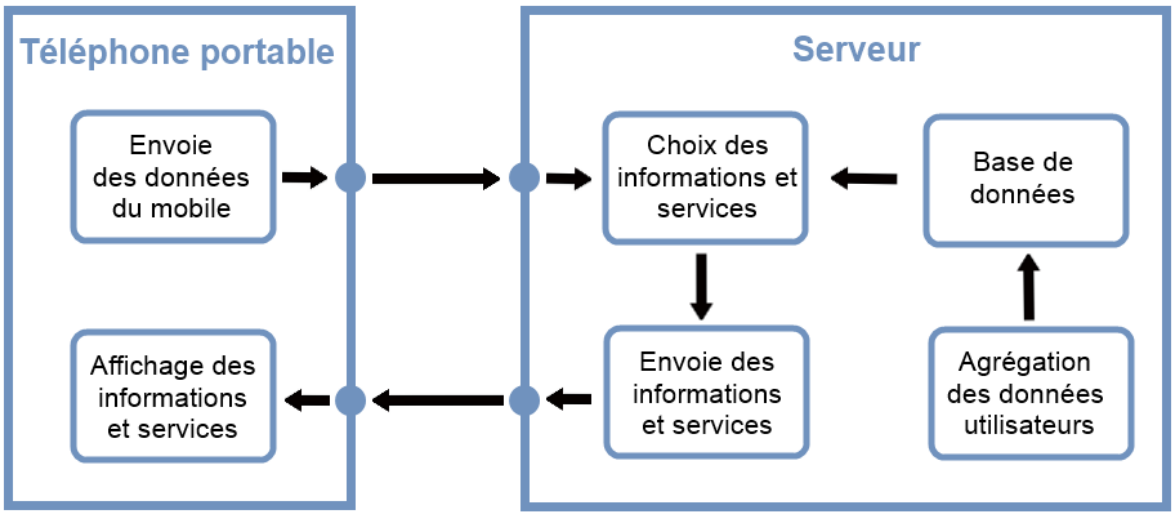


Schéma 20 : Fonctionnement d'un assistant mobile avec agrégation des données

b. Sources de données

On va donc distinguer deux types différents de source de données. Ceux récoltés par le serveur concernant la vie numérique de l'utilisateur. Et ceux provenant du mobile permettant de contextualiser en temps réel l'utilisation de l'application.

Les données récoltées par le serveur vont provenir de différentes sources de données : du navigateur Internet de l'utilisateur, sa boite mail, son calendrier en ligne et des réseaux sociaux. De plus, pour certains services, le serveur a besoin d'aller chercher certaines informations présentent sur Internet concernant les centres d'intérêt et l'environnement de l'utilisateur.

Les données intéressantes qui pourront être récupérées du navigateur Internet sont les recherches effectuées et l'historique de navigation. Ces données vont permettre à notre système de connaitre les derniers centres d'intérêt de l'utilisateur. Pour la boite mail, c'est évidemment le contenu de certains mails qui vont être pertinent. En particulier, les mails de confirmation de réservation de vols, de train, d'hôtels ou de restaurant où l'on va récupérer le code-barres, s'il y en a un, la date et lieu de la réservation. Le calendrier en ligne va permettre de récupérer tous les rendez-vous de l'utilisateur avec leurs dates et lieux. Concernant les réseaux sociaux, on va s'intéresser aux données du profil de l'utilisateur avec son adresse personnelle et celle de son travail et de ses centres d'intérêt, mais aussi à sa liste d'amis avec leurs anniversaires.

En plus de ces données sur la vie numérique de l'utilisateur, le système récupère aussi des informations sur Internet. Celles concernant les centres d'intérêt de l'utilisateur sont :

- les sorties de films, livres, musiques, séries télévisées ou jeux vidéo.
- les horaires de spectacles, concerts, pièces de théâtre et cinémas à proximité.
- les scores des rencontres sportifs
- les actualités de ces derniers.

Le système va aussi rechercher les informations concernant l'environnement de l'utilisateur. On va ainsi retrouver :

- les horaires des trains et des transports en commun lorsque l'on se trouve à proximité d'une gare, d'une station ou d'un arrêt.
- les magasins, musée, cinéma et autre endroit intéressant à proximité
- les informations sur le trafic routier, route fermée, bouchon, etc.

Enfin, concernant les données provenant du mobile, on retrouve principalement la position GPS de ce dernier permettant ensuite au serveur de choisir quels services et informations afficher.

c. Exploitation des données

Une fois toutes ces données récupérer le système est prêt à les exploiter pour proposer à l'utilisateur les services et informations dont il a besoin. Lorsque le serveur recevra la position GPS d'un utilisateur, il pourra croiser cette position avec toutes les données en sa possession et ainsi lui proposer un résultat.

Ce système peut ainsi proposer à l'utilisateur l'actualité pouvant l'intéresser, en se basant sur sa position GPS et ces centres d'intérêt récupérer grâce à ces recherches et son historique de navigation Internet et son profil de réseaux sociaux. Ça peut être le résultat d'un match d'un sport ou d'une équipe favorite, l'actualité d'un thème recherché, la sortie d'un film, livre, série, album ou jeu vidéo pouvant vous intéresser ou tout simplement la météo locale.

Toujours en utilisant la position GPS de l'utilisateur, cet assistant va pouvoir proposer des détails sur le lieu où il se trouve et ceux l'entourant. Il peut ainsi savoir s'il y a des événements pouvant l'intéresser, concert, séance de cinéma, etc., des lieux touristiques ou tout simplement des magasins ou restaurant à proximité. S'il a effectué des recherches immobilières sur Internet, il est même en mesure de lui présenter les biens immobiliers à proximité. Lorsque l'utilisateur se trouve dans un endroit particulier comme une gare ou un arrêt de transport en commun, le système proposera les horaires des prochains passages de bus ou train à cet endroit. De plus, il peut détecter lorsque ce dernier quitte le pays pour proposer des services de traduction et de conversion de devise.

En se basant sur les rendez-vous du calendrier de l'utilisateur et ses mails de confirmation de réservation, le système va afficher un rappel du rendez-vous ou de la réservation le jour de ces derniers. De plus, lorsqu'il sera temps de partir pour votre rendez-vous, l'assistant vous proposera le meilleur itinéraire pour vous y rendre. Il propose cette information en fonction du trafic routier et du temps estimer pour se rendre sur place avec les conditions de circulation actuelle. Ce même service est proposé lorsque l'utilisateur se rend au travail ou qu'il rentre chez lui. Pour cela, il utilise l'adresse du lieu de travail et son adresse personnelle qu'il croise avec la date, l'heure et sa position actuelle pour déterminer s'il travaille aujourd'hui et s'il va bientôt partir. Ainsi, il vous indiquera quand vous devrez partir en fonction des conditions de circulations afin d'arriver à l'heure.

Enfin, le système propose différents services comme le suivi de vos colis lorsque vous recevez des notifications sur votre boite mail ou encore l'affichage d'alerte public comme les alertes inondations, tempête ou enlèvement.

Cet assistant fonctionne donc sans la nécessité d'une action de l'utilisateur, si ce n'est le lancement de ce dernier, et permet d'anticiper ses demandes pour lui proposer les services dont il a besoin et quand il en a besoin.

3. Architecture et technologies de l'assistant

Comme on l'a vue dans le chapitre précèdent, un assistant se basant sur l'agrégation de données utilise différentes sources de données. Ces sources de données, les emails, les calendriers en ligne, les profiles des réseaux sociaux et les historiques de recherche et de navigation, produisent un volume considérable de données surtout lorsqu'on le multiplie par le nombre d'utilisateurs. Une architecture de gestion de base de données standard n'est pas adaptée à un si grand volume de données. Il faudra donc envisager des solutions de BigData qui permettront de gérer cette masse de données.

a. Big Data

Big Data est un terme anglophone désignant des ensembles de données si grand et complexe qu'ils en deviennent difficiles à exploiter par des outils classiques de gestion de base de données. Avec un tel volume de données, la capture, le stockage, la recherche, le partage, l'analyse et la visualisation des données doivent être redéfinis. Le Big Data est une problématique actuelle qui a pris de l'ampleur ces dernières années. En effet, de plus en plus de données sont générées dans le monde et on est passé de 1,2 zétaoctet (10^{21} octets) de données créées en 2010 à 2,8 zétaoctets en 2012. Ces données proviennent essentiellement des recherches scientifiques, d'Internet et des réseaux sociaux et des différents capteurs provenant de différentes plateformes (mobile, tablette, etc.).

Un rapport de recherche du groupe Gartner* définit le Big Data en trois enjeux appelé les « 3V » pour Volume, Variété et Vélocité. Le volume pour la masse de données stockée toujours plus élevée. La variété, car les données proviennent de source, de structures et de types différents et que leurs analyses portent de plus en plus sur le lien entre ces données. Et la vélocité correspond à la fréquence à laquelle sont générés, capturés et partagé ses informations et à leur vitesse de traitement. Ce modèle est d'ailleurs encore largement utilisé pour décrire le Big Data.

Les solutions de BigData vont donc permettre de résoudre les problèmes liés à ce grand volume de données. Ces solutions vont agir à deux niveaux différents. Tout d'abord sur la structure de la base de données elle-même, qui ne sera plus fondée sur l'architecture classique des bases relationnelles, mais qui sera adaptée à un tel volume de données.

Ensuite au niveau de l'exécution des requêtes, qui seront distribuées et massivement parallélisées afin de garantir des performances optimales. Pour détailler cela, je vais aborder la structure de base de données avec le NoSQL et l'exécution des requêtes avec MapReduce.

i) NoSQL

NoSQL signifie littéralement « Not Only SQL », que l'on peut traduire par « Pas seulement SQL* ». Ce terme désigne une catégorie de systèmes de gestion de base de données (SGBD*) qui ne se base plus sur l'architecture classique des bases relationnelles. NoSQL a était créé à l'origine pour les géants du Web qui ont vu leur besoin en terme de charge et de masse de donnés croitre de manière exponentielle. Ces organisations ont donc fait des compromis sur la notion relationnelle et ont simplifié le SGBD* afin de répondre à leurs besoins. Les systèmes de type de NoSQL sont donc plus performants, en particulier avec un grand nombre de données, plus évolutif et supporte mieux la monté en charge en augmentant simplement le nombre de serveurs. Il ne vise pas à remplacer les solutions SQL*, mais permet plutôt de répondre à des besoins qu'un SGBD* de type SQL* ne pourrait pas ou pas de manière aussi satisfaisante. Le NoSQL sera donc intéressant à utiliser dans un contexte ou la performance, l'évolutivité et la montée en charge sont des exigences incontournables.

On va distinguer quatre grandes familles de bases de données NoSQL. Tout d'abord les bases de données orientées clé/valeur. Elles sont constituées d'une liste de valeurs avec pour chacune d'elle une clé unique. Les données sont donc représentées par un simple couple clé/valeur. La valeur peut être de n'importe quel type, entier, décimal, chaine de caractère, etc. La communication avec une telle base de données se résumera à des commandes d'ajout/modification, de suppression et de récupération. Ci-dessous, un exemple de modélisation de données, à gauche sous la forme d'un modèle relationnel et à droite d'un modèle clé / valeur.

ID	Nom	Âge
1	David	21
2	John	28

Clé	Valeur
utilisateur1_nom	David
utilisateur1_age	21
utilisateur2_nom	John
utilisateur2_age	28

Tableau 3 : Exemple de modélisation de données orientée clé / valeur

Ensuite, nous avons les bases de données orientées document qui sont un dérivé des bases de données orienté clé/valeur. En effet, ces dernières reprennent le principe d'une liste de donnée représenté par un couple clé et valeur. Seulement ici, la valeur pourra être plus complexe et être représentée sous la forme d'une autre liste de clés/valeurs, appelées document. On manipulera donc une liste de documents associés à une clé unique. Cela permet donc, à partir d'une seule clé, de récupérer un ensemble d'information structuré de manière hiérarchique. Chaque document étant indépendant des autres, il ne contient pas forcément les mêmes clés. Voici un exemple de modélisation de données orientée document, basé sur les données du « Tableau 3 ».

Clé	Document
utilisateur1	nom = David âge = 21
utilisateur2	nom = John âge = 28

Tableau 4 : Exemple de modélisation de données orientée document

Un autre type de base de données sont les bases de données orientées colonne. Ces dernières ressemblent aux tables d'une base de données relationnelle sauf qu'ici le nombre de colonnes sera dynamique. En effet, les colonnes d'une base de données relationnelle sont fixées à la création de la table et chaque enregistrement contiendra le même nombre de colonnes. Par contre, avec une base de données orientée colonne, le nombre de colonnes peut varier d'un enregistrement à un autre, ceci évitant de se retrouvant avec des colonnes ayant des valeurs nulles. Ce modèle est très performant pour les requêtes interrogeant peu de colonnes. Il l'est en particulier lorsque l'on veut récupérer la liste des enregistrements suivant la valeur d'une colonne, par exemple toutes les personnes dont le nom est John. Ci-dessous, un exemple de modélisation de données orientée colonnes reprenant les données du « Tableau 3 ».

Nom	
ID	Valeur
1	David
2	John

Âge	
ID	Valeur
1	21
2	28

Tableau 5 : Exemple de modélisation de données orientée colonne

Enfin les bases de données orientées graphe se base sur la théorie des graphes. Il reprend donc les notions de nœud, de relation et de propriété. Les nœuds vont représenter les entités que l'on veut sauvegarder, par exemple une personne. Les propriétés seront les informations relatives au nœud, comme le nom ou l'âge d'une personne. Les liaisons représentent la connexion entre deux nœuds, cette connexion pouvant contenir des informations, par exemple une connexion entre deux personnes avec la nature de leur relation et depuis quand cette dernière existe. Ce modèle facilite la représentation du monde réel, ce qui le rend particulièrement adapté pour les réseaux sociaux. Avec ce type de base de données, on pourra facilement parcourir le graphe pour récupérer un nœud. Alors qu'avec une base de données standard une jointure, qui est très couteuse en termes de performance, sera nécessaire. Elle est particulièrement appropriée lorsqu'il s'agit d'exploiter les relations entre les données. L'illustration suivante présente un exemple de modélisation de données orientée graphe reprenant partiellement les données du « Tableau 3 ».

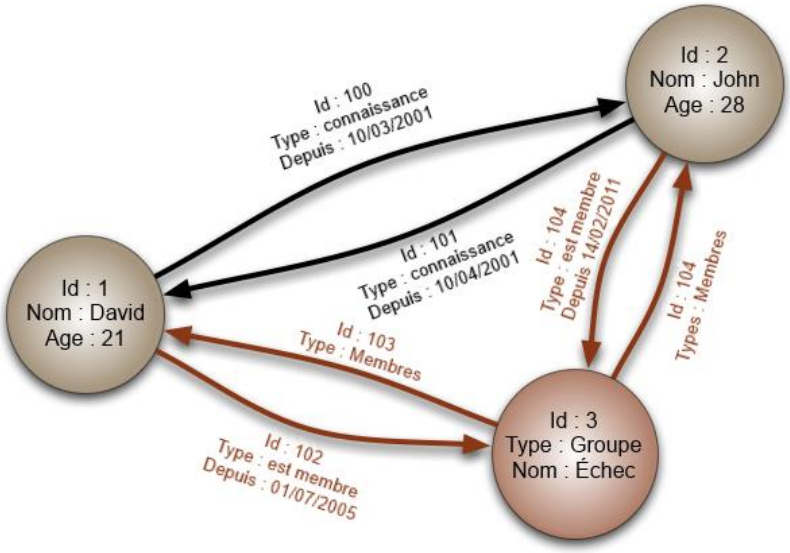


Illustration 18 : Exemple de modélisation de données orientée graphe

ii) MapReduce

MapReduce est un modèle de programmation popularisé par Google, qui permet d'effectuer des traitements parallèles, et souvent distribués, de données potentiellement très volumineuses. Il est constitué de deux fonctions : map et reduce.

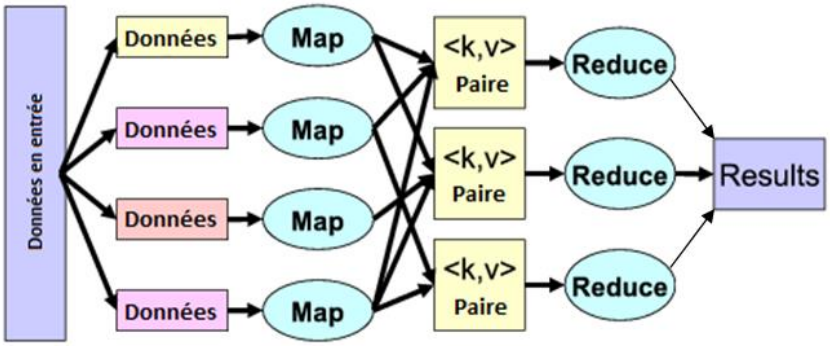


Schéma 21 : Fonctionnement de MapReduce

Dans l'étape Map, le nœud principal analyse un problème et le découpe en sous-problèmes qu'il va ensuite déléguer à d'autres nœuds (qui peuvent en faire de même récursivement). Chaque nœud va alors traiter le sous-problème qui lui est assigné à l'aide de la fonction map. Cette fonction va prendre en entrée un couple clé/valeur, le sous-problème, et y associer un ensemble de couples clé/valeur. Prenons un exemple où l'on veut compter le nombre d'occurrences de chaque mot dans un texte. Le nœud principal reçoit le texte à analyser. Il le découpe en différents paragraphes. Il délègue ensuite chaque paragraphe à différents nœuds. Ces derniers exécutent la fonction map avec en entrée le numéro du paragraphe en tant que clé, et le paragraphe lui-même en tant que valeur. La fonction va alors décomposer le paragraphe en ensemble de couples clé/valeur où les clés seront tous les mots du paragraphe avec pour valeurs 1, cette valeur servira par la suite à compter l'occurrence de chaque mot.

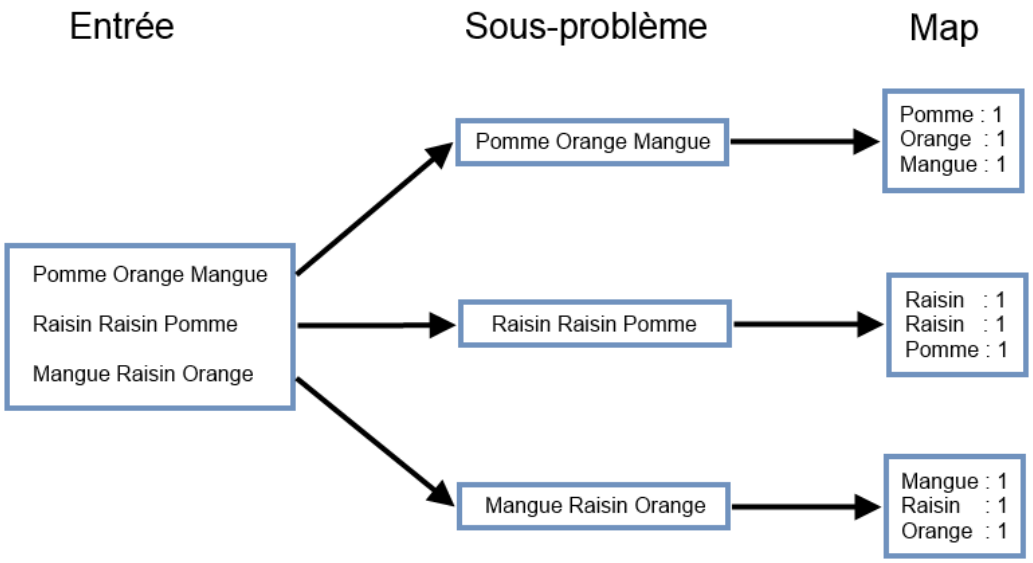


Schéma 22 : Modélisation de l'étape Map de MapReduce

Pour l'étape reduce, à partir des listes de clé/valeur retournées par les différents nœuds ayant exécuté la fonction map, le système MapReduce regroupe toutes les valeurs ayant les mêmes clés. Il transmet ensuite aux différents nœuds une clé avec la liste de valeur associée. Ces nœuds exécutent alors la fonction reduce qui va retourner au nœud principal un couple clé/valeur à partir de la clé et de la liste de valeur fournie. Le nœud principal va alors pouvoir recomposer la réponse finale à partir du résultat de la fonction reduce de chaque nœud. Si on reprend notre exemple, chaque nœud a donc extrait tous les mots du paragraphe qui lui était assigné en tant que clés et avec 1 pour valeur. Le système MapReduce va maintenant combiner les résultats afin de récupérer pour chaque clé, donc chaque mot, sa liste de valeurs, où chaque valeur correspond à une occurrence du mot. Ces mots et ces listes de valeur vont ensuite être redistribués à différents nœuds afin de calculer le nombre d'occurrences de chaque mot. Pour cela, la fonction reduce va effectuer une somme des valeurs présente dans la liste. Le résultat est alors retourné au nœud principal qui disposera donc du nombre d'occurrences de chaque mot.

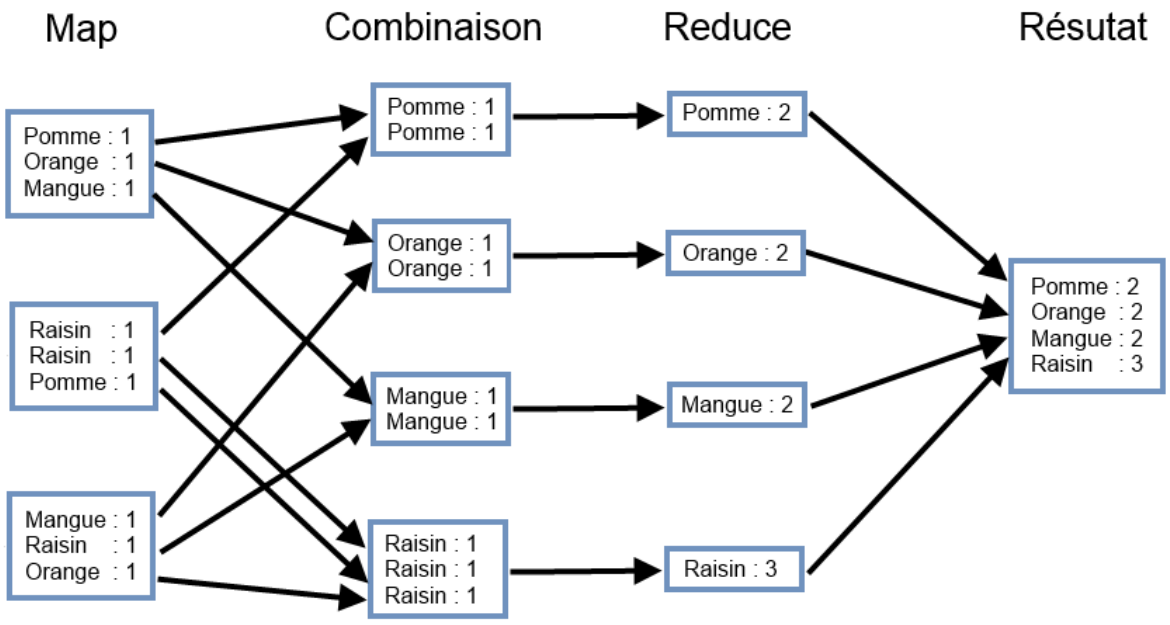


Schéma 23 : Modélisation de l'étape Reduce de MapReduce

b. *Architecture distribuée*

Comme on l'a vue dans le chapitre précédent, afin de pouvoir manipuler toutes les données utiles au fonctionnement de cet assistant mobile, il faut utiliser des solutions de BigData. Avec une base de données de type NoSQL et une solution de parallélisation grâce à MapReduce. Pour utiliser pleinement ces technologies, il faut mettre en place une architecture distribuée. Ce type d'architecture va permettre de répartir la base de données et de distribuer leurs traitements sur plusieurs serveurs.

Une architecture distribuée va donc permettre de faire travailler ensemble un réseau de serveurs. Ces machines vont alors travailler de manière coordonnée afin d'accomplir une tâche commune. Ce réseau de serveurs peut aussi bien se situer dans un même local que réparti géographiquement. Ce type d'architecture est essentiel lorsqu'on parle de BigData. En effet, le stockage et les traitements d'un tel volume de données nécessitent forcément l'utilisation de nombreux serveurs.

Pour mettre en place ce type d'architecture il sera nécessaire d'utiliser : un système permettant le traitement parallélisé et distribué, une base de données autorisant la répartition de ses données et enfin un système de fichier supportant la distribution d'un fichier sur plusieurs machines. La parallélisation et la distribution des traitements pourront être effectuées grâce au système MapReduce présenté précédemment. Concernant les bases de données, celle de type NoSQL supporte nativement la distribution des données. En effet, comme on l'a vu au chapitre précédent, les bases de données de type NoSQL ont la capacité de supporter la montée en charge en multipliant les serveurs. Enfin, il faudra utiliser un système de fichier, dit distribué, pour supporter la répartition des fichiers.

i) Système de fichier distribué

Un système de fichier distribué va donc permettre de répartir un fichier, souvent très volumineux, sur plusieurs machines. Cette répartition est totalement transparente pour l'utilisateur du système. Ce type de système de fichier fait, en effet, abstraction de l'architecture physique du stockage et permet ainsi d'être manipulé comme si c'était un seul disque dur.

Pour réaliser cela, le système, va découper le fichier en plusieurs parties et stocké chacune de ces dernières sur un serveur différent. Afin de garantir la fiabilité, chacune de ces parties sera répliquée sur un serveur différent.

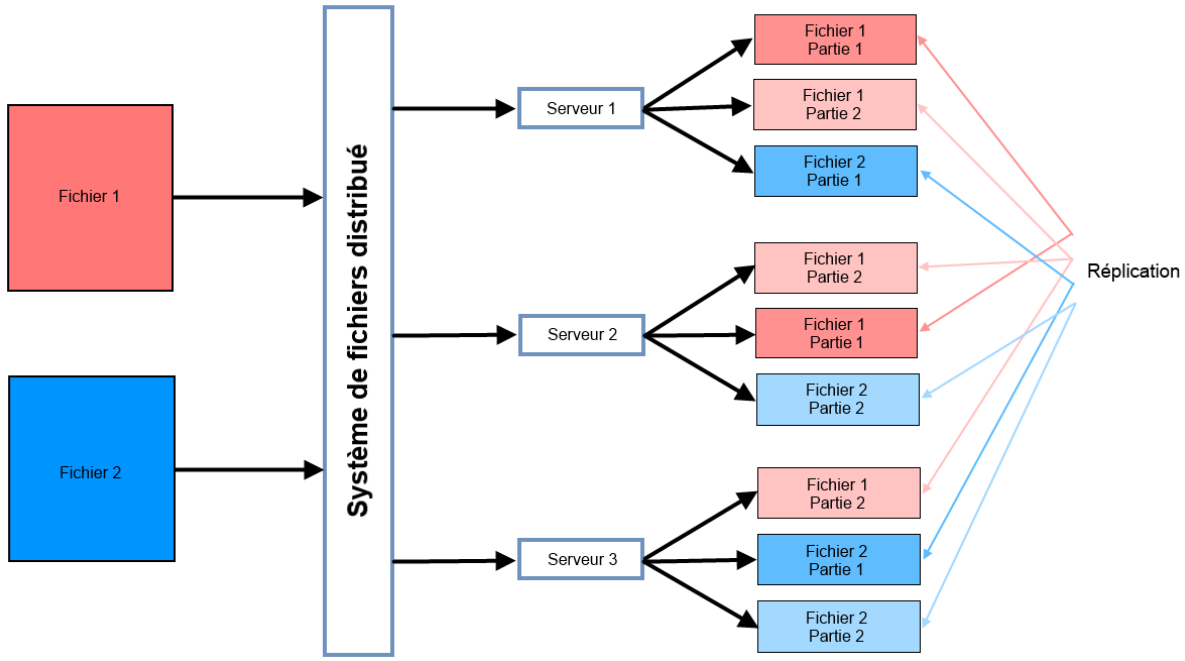


Schéma 24 : Système de fichier distribué

ii) Framework pour les architectures distribuées

Afin de faciliter la mise en place d'une architecture distribuée dans une utilisation de BigData, certains Frameworks regroupant tous les outils nécessaires ont été développés. Parmi les plus populaires, on retrouve Hadoop. Ce dernier est un framework Java* open source développé par la fondation logicielle Apache*. Il implémente un système MapReduce pour la parallélisation et la distribution des traitements, propose une base de données NoSQL : HBase et utilise un système de fichier Hadoop Ditributed File System (HDFS).

HBase est un système de gestion de base de données non relationnelle distribué développé en Java*. C'est un projet open source utilisant une base de données orientée colonne. Il s'inspire d'un produit Google : BigTable*. Il est optimisé pour les systèmes de fichiers distribués. En effet, chaque colonne de la base de données est découpée en plusieurs parties qui pourront être réparties sur différent serveur. Ces parties de base de données pourront être compressées afin de réduire la taille de la base. La localisation dans le système de fichier de chacune de ces parties et enregistré dans une colonne spécifique.

Le système HDFS, inspiré de GoogleFS*, développé par Google, repose sur deux composants majeurs :

- NameNode : ce composant gère l'espace de nom et l'arborescence du système de fichier. Il centralise la localisation des blocs de données répartie sur les serveurs. Il est unique dans l'architecture, mais il existe un NameNode secondaire qui va gérer l'historique des modifications du système de fichier. Il servira donc de sauvegarde et pourra prendre le relai du NameNode principal en cas de panne.
- DataNode : il stocke et restitue les blocs de données. Les DataNodes communiquent de manière périodique au NameNode la liste des blocs de données qu'ils hébergent. Si certains ne sont pas assez répliqués, ils sont copiés sur d'autre DataNodes.

Lors du processus de lecture d'un fichier, le NameNode est interrogé pour localiser l'ensemble des blocs de données. Pour chacun d'entre eux, le NameNode renvoie l'adresse du DataNode le plus accessible, c'est-à-dire le DataNode qui dispose de la plus grande bande passante.

4. Mise en place

La mise en place d'un tel assistant nécessite un investissement conséquent. Il faudra tout d'abord être en mesure de récolter toutes les données nécessaires et donc d'avoir accès aux différentes sources de données : boite mail, calendrier, navigateur et réseaux sociaux. Même si cela est relativement facile d'accéder aux données de la boite mail, du calendrier et des réseaux sociaux de l'utilisateur en lui demandant ses identifiants. Il faudra quand même prévoir le support des différentes solutions existantes de ces logiciels n'utilisant pas forcément les mêmes technologies et donc les mêmes méthodes de communication. Le plus compliqué reste l'accès à l'historique des navigations et des recherches de l'utilisateur. Ces données ne sont pas forcément sauvegardées en ligne, et lorsqu'elles le sont, sont réservées à l'utilisation par l'éditeur du navigateur.

De plus, l'architecture matérielle à mettre en place est plutôt conséquente. Il faudra en effet prévoir assez de serveurs pour le bon fonctionnement de l'assistant.

Tout d'abord pour stocker toutes les données voulues, qui peuvent être très volumineuses avec la multiplication des sources de données et des utilisateurs. Mais aussi pour supporter la charge, le nombre de requêtes simultané, de tous les utilisateurs.

Ce n'est donc pas n'importe qui et n'importe quelles entreprises qui vont pouvoir mettre en place ce type d'assistant. Cependant, certaines grandes entreprises vont pouvoir se permettre de mettre en place un tel assistant. En effet, si on prend de grandes entreprises comme Google ou Microsoft, ces dernières possèdent déjà toutes les données nécessaires via leurs différents logiciels et services en ligne. De plus, leurs systèmes informatiques possèdent déjà les architectures et technologies nécessaires (architecture distribuée et solution de BigData). Le développement d'un tel assistant par ces derniers est donc envisageable sans pour autant nécessiter un investissement trop important. D'ailleurs, Google a sorti, fin 2012, un assistant reprenant le concept d'agrégation et de croisement de données. Ce dernier est disponible gratuitement sur Android et iOS sous le nom de Google Now.

Conclusion

Ce mémoire a donc pour but de montrer en quoi un téléphone portable peut être un bon assistant. C'est ainsi qu'on a vu que les mobiles étaient très répandus aujourd'hui : deux tiers de la population mondiale en possède un, et dans 66% des cas, il s'agit d'un Smartphone*. Ce qui représente une base d'utilisateurs potentielle conséquente. L'accès à l'assistant mobile en est d'ailleurs simplifié. Les utilisateurs désirant se procurer ce dernier n'auront qu'à télécharger l'application sur leur mobile. Pas besoin de se rendre dans un magasin pour acheter un nouvel appareil. Il faut aussi prendre en compte, le prix d'une application mobile, qui, étant largement inférieur à celui d'un appareil complet, permet d'intéresser et de toucher un plus grand public.

Ensuite, nous avons vu certaines capacités du Smartphone* permettant une utilisation d'assistant. Avec, tout d'abord les systèmes de reconnaissances vocales et d'image. On peut distinguer deux types d'utilisation de reconnaissance vocale. Dans le premier cas, elle va servir d'interface entre l'utilisateur et l'assistant. Permettant d'interagir et de communiquer de manière plus naturelle. L'assistant va ainsi répondre à un certain nombre de commandes vocales prédéfinies. Dans le second cas, cela va permettre à l'utilisateur de dicter des notes, des SMS, des mails ou une recherche, directement à l'assistant. La reconnaissance d'image sert, quant à elle, à reconnaitre un objet sur une image, le plus souvent une photo prise depuis le Smartphone*. Cela permet ainsi à l'assistant de proposer des informations concernant l'objet reconnu comme le détail du lieu ou le nom d'un produit et son prix.

De plus, les mobiles vont pouvoir se positionner dans l'espace. D'une part, grâce à la puce GPS permettant de géolocaliser l'utilisateur. La position GPS va servir à tous les assistants proposant des services géolocalisés, allant de la simple recherche à proximité de centres d'intérêt, à la navigation GPS. Les Smartphones* intègrent aussi un magnétomètre permettant de déterminer l'orientation sur trois dimensions du téléphone. Cette capacité sert principalement aux assistants proposant une partie de réalité augmentée* pour visualiser des informations localisées.

Enfin, les mobiles vont aussi être capables de restituer certains services. J'ai évoqué plus particulièrement les services bancaires, commerciaux et de paiements. Ces derniers permettent, respectivement, d'accéder à ses services bancaires, de réaliser ses achats depuis n'importe où et d'effectuer un paiement simplement en passant son mobile sur une borne. Ce dernier service, le paiement mobile, utilise une des dernières technologies apparues sur les mobiles, le NFC*.

Les terminaux mobiles possèdent donc déjà de nombreuses fonctionnalités permettant de les qualifier d'assistants. Mais il faut aussi noter que ces derniers sont en constante évolution et intègrent toujours plus de technologie permettant de proposer des assistants toujours plus pertinents.

Cependant, ils souffrent aussi de quelques limitations. En effet, le mobile possède des ressources limitées et ne permettra pas d'exécuter des assistants demandant trop de puissance. De plus, si un assistant à besoin de données provenant des capteurs du téléphone, sa précision dépendra de la qualité de ces derniers. Or, de tels capteurs, ayant un coût non négligeable, ne sont pas favorisés par les constructeurs. Il existe, par ailleurs, sur le marché, différents systèmes d'exploitation* mobiles nécessitant chacun sa propre version de l'assistant. Cela entraine évidemment une multiplication de la charge de développement.

Même si on décide d'utiliser une architecture orientée serveur pour pallier aux contraintes du mobile, on va faire face à d'autres problèmes. Le principale étant l'augmentation du temps de réponse du système dû au transfert des données entre le mobile et le serveur.

On retrouve aussi des limitations sur les assistants nécessitant l'utilisation d'intelligence artificielle. Cette dernière n'est, à ce jour, pas assez développée pour reproduire une réflexion humaine. Cela affecte particulièrement les systèmes de reconnaissance qui n'auront pas cet aspect de contextualisation, naturel pour les humains, permettant, par exemple, de comprendre une phrase même si elle est mal articulée.

La mobilité représente aussi une contrainte dans certains cas. Tout d'abord pour la sécurité des données. En effet, comme tout échange sur Internet, les transferts de donnée entre le mobile et le serveur peuvent être interceptés pas des personnes malveillantes.

Cela peut être particulièrement risqué surtout lorsqu'il s'agit de données sensibles comme des données privées ou des informations bancaires. Les données présentes sur le téléphone peuvent aussi poser des problèmes de sécurité. Tout d'abord en cas de vol du téléphone, le voleur aura accès à toutes vos données. Mais aussi dans le cadre de l'espionnage industriel. En effet, une fois l'application installée sur le téléphone, on peut facilement récupérer le code et les bases de données de l'application. Enfin, la mobilité entraine une utilisation dans différents types d'environnements pas forcément adaptés pour la réception GPS et Internet ou pour la reconnaissance vocale.

Il existe, toutefois, des solutions permettant de pallier aux problèmes évoqués dans la partie précédente. Tout d'abord concernant la diversité des systèmes d'exploitation* mobiles. Grâce à l'utilisation d'application Web, on va pouvoir exécuter notre application sur n'importe quelles plateformes. En effet, ces dernières se présentant sous la forme de page Web ont uniquement besoin d'un navigateur Internet pour fonctionner. Cette solution est cependant limitée à des tâches relativement simples et ne permet pas d'atteindre les performances des applications natives. Une autre solution est le partage de code. Ici, on va mettre en commun le maximum de code qui pourra être exécuté sur toutes les plateformes. On retrouvera quand même du code spécifique à chaque plateforme concernant l'interface utilisateur et l'accès à certains capteurs et service spécifique. Cette dernière permet cependant de conserver de bonne performance et de proposer des interfaces utilisateur fidèles à ceux des systèmes d'exploitation.

La démocratisation et le déploiement de la nouvelle génération de technologie de télécommunication mobile, la 4G, va quant à elle permettre de réduire le temps de communication entre le mobile et le serveur. Avec cette dernière, on va pouvoir obtenir des temps de communication jusqu'à 140 fois plus rapide qu'avec les technologies de 3ème générations. Cette technologie est en cours de déploiement depuis 2012 et regroupe déjà 58 millions d'abonnés.

Concernant les solutions de sécurisation des données, nous avons la technologie TLS pour protéger celles issues des communications entre le mobile et le serveur.

Cette technologie utilise différents algorithmes* de cryptographie afin de sécuriser l'échange des clés de décryptage et les communications en elles-mêmes. Cette technologie est très robuste, mais peut cependant être gourmande en ressource. Il faudra donc utiliser cette dernière uniquement lorsque cela est nécessaire, c'est à dire lorsque les données sont sensibles.

Pour les données stockées sur le téléphone, la meilleure solution reste de laisser les algorithmes* et données sensibles sur un serveur sécurisé. Cependant, si on doit toutefois protéger son application, on peut effectuer une opération d'obfuscation qui va rendre le code le plus illisible possible pour l'homme, mais qui sera toujours exécutable par une machine. Cette solution n'a pas vocation à sécuriser entièrement l'application, mais à rendre la tâche de reverse engineering difficile voir d'en dissuader ses auteurs.

Tout récemment, un nouveau concept d'assistant mobile a vu le jour. Ce dernier utilise l'agrégation de données afin de prédire les services et informations dont l'utilisateur va avoir besoin. Pour cela, il va croiser des données provenant de différentes sources d'information, tel que la boite mail, le calendrier, les réseaux sociaux et le navigateur, afin de déterminer ce dont a besoin l'utilisateur à ce moment précis.

L'utilisation de tant de données pose des problèmes de performance. Il est donc nécessaire de mettre en place des solutions de BigData avec une architecture distribuée. Cela passe par un système de fichier distribué permettant le stockage de fichier (très volumineux) sur différentes machines. Mais aussi par un système de gestion de base de données de type NoSQL qui sera capable de gérer un tel volume de donnée de manière performante. Et enfin, afin d'optimiser la manipulation de ces bases volumineuses, il sera nécessaire de paralléliser et de distribuer ses traitements avec l'utilisation d'un système comme MapReduce.

Ce type d'architecture nécessite un grand investissement, en particulier pour l'achat de serveur en nombre nécessaire pour stocker la masse de données et supporter la charge lorsque tous les utilisateurs utilisent le service. Cette solution est donc réservée aux grandes entreprises disposant déjà de telle architecture et ayant accès aux sources de donnée nécessaires.

Pour qu'un assistant personnel soit performant, il doit tout connaitre de la vie de l'utilisateur. Ceci est vrai autant pour un assistant virtuel qu'humain et plus particulièrement pour les assistants mobiles utilisant l'agrégation. En effet, la pertinence des résultats de ces systèmes dépend entièrement de la qualité et de la quantité des informations à disposition sur l'utilisateur. Si cela est globalement accepté lorsqu'il s'agit d'un assistant humain, du fait de la confiance accordée et de la confidentialité due au contrat, avec un assistant virtuel, on peut se poser la question de la vie privée. En effet, un tel assistant est capable de savoir où une personne se trouve, où elle va, quels sont ses centres d'intérêt, quelle recherche elle a effectuée sur Internet, etc.

On peut définir la vie privée comme étant l'ensemble des éléments relatifs à une personne relevant de son intimité. Ces éléments seront différents pour chaque personne suivant son travail, ses coutumes, son pays et, plus globalement, son envie de garder ces éléments privés. En informatique, les informations relatives à la vie privée sont toutes celles que l'on va pouvoir associer à une personne : son état civil, son adresse IP, son identifiant, tout type d'adresse (courriel, postale, téléphone, etc.), son numéro de Sécurité sociale et ses caractéristiques biométriques.

Le droit au respect de la vie privée est protégé par la Déclaration universelle des droits de l'homme. Ainsi, tout système stockant et utilisant des données personnelles doit respecter certaines obligations suivant le pays dans lequel il se trouve. En France, par exemple, il faudra tout d'abord faire une déclaration à la CNIL. Les données doivent être protégées de manière adaptée suivant leur nature et leur sensibilité. Les informations doivent être confidentielles et n'être communiquées qu'aux personnes autorisées. Les données doivent avoir une durée de conservation raisonnable en fonction de leur objectif. Chaque personne concernée par des données doit pouvoir exercer pleinement ses droits. Il doit pouvoir accéder aux données à des fins de consultation ou de modification, être informé de leur utilisation et des destinataires. Enfin, les données doivent avoir un objectif précis et être utilisées de manière cohérente par rapport à ce dernier.

Dans les faits, ces règles sont dans la plupart des cas respectées. Seulement, renseigner ses données personnelles sur Internet est devenu quelque chose d'habituel. Les conditions générales d'utilisations ne sont d'ailleurs que rarement lu, alors que c'est ici qu'est indiqué comment vont être utilisé vos données. Il faut avouer que, bien souvent, ces informations sont noyées dans la masse de texte. On se retrouve ainsi avec peu de personnes sachant réellement comment sont utilisées leurs données et quels risques elles encourent.

Avec des dossiers d'actualité telle que PRISM, on peut donc se demander comment sont utilisées nos données. Pour rappel, PRISM pour « Planning tool for Resource Integration, Synchronization, and Management », ou en français « Outil de planification pour l'intégration la synchronisation et la gestion des ressources » est un programme de surveillance électronique américain de collecte de renseignement à partir d'Internet et d'autres services électroniques. Ce programme classé de la NSA a été révélé par un analyste en 2013 ce qui a lancé beaucoup de débats sur le respect de la vie privée.

Tout d'abord, il faut rappeler que le but de toute entreprise est de faire des bénéfices. Ainsi, lorsque ces dernières proposent des services gratuits, c'est qu'ils font des profits autre part. Dans le cas où ces dernières manipulent des données personnelles, c'est souvent ces données qui vont leur permettre de générer des profits. Pour cela, ils vont utiliser les données collectées à des fins publicitaires et commerciales. Certaines vont ainsi revendre directement vos informations ou des statistiques sur l'ensemble des données d'une communauté. Mais la solution la plus répandue est l'utilisation de bandeau publicitaire ciblé. Cet dernier consiste à proposer à l'utilisateur de la publicité qui sera le plus en mesure de l'intéresser. Pour cela, elle se base sur ses centres d'intérêt obtenus à partir de ses données personnelles. Par exemple, lorsque vous recevez un mail, son contenu sera scanné pour proposer de la publicité en rapport. Ce système leur permet de vendre leurs emplacements publicitaires plus chers, car plus pertinents.

Ainsi, ces dernières années, les données sont devenues un enjeu économique majeur. En 2013, on considère que la collecte, le contrôle et l'exploitation des données représentent à eux seuls 312 milliards de dollars, soit 234 milliards d'euros. Ceci correspond à 38% du chiffre d'affaires mondial du commerce en ligne.

Renseigner ses informations à une entreprise afin d'utiliser un service comporte donc certains risques pour sa vie privée. Seulement, dans la plupart des cas, ses données, même si elles sont utilisées pour générer des bénéfices, sont nécessaires au fonctionnement du service. Partager ses informations sera donc nécessaire si on veut en bénéficier. Mais dans ce cas, comment être sûr que ses données ne sont pas utilisées à des fins publicitaires ou commerciales.

Une des solutions consiste à l'utilisation de systèmes répartis ou distribués. Comme nous l'avons vu, un système distribué permet de faire réaliser une même tâche à différentes machines. Dans le cas du BigData, toutes les machines appartenaient à l'entreprise et faisaient partie du même réseau local, et cela servait à améliorer les performances. Ici, il en sera tout autre, les appareils de chaque utilisateur du service feront partie du système réparti et seront reliés entre eux par Internet. Ainsi, les données ne seront plus transmises aux serveurs de l'entreprise et chacun pourra les conserver. Une telle organisation est appelée grille informatique ou Grid en anglais. Les grilles informatiques sont, comme pour les systèmes distribués, le plus souvent utilisées afin d'optimiser les performances en se servant de chaque ressource inutilisée. Cependant, certains projets utilisant ce type d'organisation ont vu le jour afin de permettre aux utilisateurs de garder leurs informations localement ou de les rendre anonymes.

Un tel système, dit décentralisé, va donc se passer des serveurs de l'entreprise qui représentaient un tiers de confiance*. Afin de garder une sécurité optimale, ces systèmes vont se baser sur des solutions cryptographiques de types asymétriques. La vérification du certificat ne sera pas effectuée par une autorité de certification, mais par le réseau d'utilisateurs lui-même. Pour cela, chaque utilisateur va choisir les personnes à qui il fait confiance. Ensuite, lorsqu'il recevra un message, l'utilisateur vérifiera qui a signé ce certificat. Si ce dernier a été signé par une personne à qui il fait confiance, le certificat sera considéré comme valide. Ce concept de vérification de certificat est appelé toile de confiance.

Un exemple de l'utilisation d'un tel système est la monnaie électronique BitCoin. Il s'agit d'un système qui permet d'effectuer des transferts de monnaie BitCoin directement entre deux utilisateurs. Il va donc permettre de se passer d'un intermédiaire présent dans les systèmes standard sous la forme d'une banque, d'une entreprise ou d'un état.

Pour garantir la sécurité des transactions, il existe des utilisateurs spéciaux appelés « mineurs ». Ces derniers, après avoir téléchargé la liste publique de toutes les transactions sur le réseau BitCoin, vont vérifier, de manière chronologique, les transactions en attentes et les rajoutés aux transactions effectuées. Pour cela, le certificat, la clé privée et l'empreinte de la transaction sont vérifiés. Si cette dernière est considérée comme valide, une empreinte de cette transaction, basée aussi sur la précédente, sera générée. Cela va permettre d'empêcher la modification d'ancienne transaction, car cela nécessiterait de recalculer l'ensemble des empreintes des transactions suivantes. Une transaction est jugée comme valide par le système BitCoin, si une majorité des mineurs la considère comme valide. Ainsi, pour qu'une fausse transaction soit validée, il faudrait que le nombre de mineurs malveillants soit supérieur au nombre de mineurs honnêtes.

Pour utiliser BitCoin, l'utilisateur aura besoin d'installer un portefeuille BitCoin. Ce dernier permettra de gérer vos adresses BitCoin. Les adresses BitCoin vont servir à identifier l'émetteur et le destinataire de la transaction. Un portefeuille peut contenir plusieurs adresses BitCoin. Ces adresses sont personnelles et ne sont communiquées uniquement par son possesseur pour recevoir une transaction. Il est ainsi conseillé de créer une adresse différente pour chaque transaction, cela permettant de garder son anonymat. Vous trouverez en Annexe II, un schéma en anglais expliquant les étapes d'une transaction BitCoin.

Sujet de polémique, l'actualité est marquée par des cas de violation de vie privée qui mettent en évidence le flou entourant l'utilisation des données. Cela met en avant, par la même occasion, le fait que les grandes entreprises de l'informatique sont capables, aujourd'hui, de tracer l'ensemble des activités, déplacements et agissements, que nous réalisons. La question de la vie privée est donc devenue un enjeu majeur alliant des questions juridiques, techniques et surtout d'éthiques.

Glossaire

ALGORITHME : suite d'opérations ou d'instructions permettant de résoudre un problème.

APACHE : organisation à but non lucratif qui développe des logiciels open source sous la licence Apache. Elle a été créée en juin 1999 dans le Delaware aux États-Unis. La Fondation Apache est une communauté décentralisée de développeurs qui travaillent sur ses projets open source. Les projets Apache sont caractérisés par un mode de développement collaboratif fondé sur le consensus ainsi que par une licence de logiciel ouverte et pragmatique. Chaque projet est dirigé par une équipe de contributeurs auto-désignée et on ne devient membre de la fondation qu'après avoir contribué activement aux projets Apache.

BADA : système d'exploitation pour Smartphones de Samsung. Il est sorti début 2010 et est, au 27 août 2011, le 3e en part de marché en France et équipe plus d'un million de Smartphones. Le 25 février 2013, Samsung a annoncé qu'il va arrêter le développement de Bada, et se concentrer, à la place, sur le développement de Tizen*.

BIGTABLE : système de gestion de base de données compressée, haute performance, propriétaire, développé et exploité par Google. C'est une base de données orientée colonnes, dont se sont inspirés plusieurs projets libres, comme HBase, Cassandra ou Hypertable. Chez Google, BigTable est stockée sur le système de fichiers distribué GoogleFS*. Google ne distribue pas sa base de données, mais propose une utilisation publique de BigTable via sa plateforme d'application Google App Engine.

CAPTEUR D'ORIENTATION : mesure l'angle de l'appareil par rapport à un axe de référence (position verticale du téléphone).

CELLULE : zone géographique circulaire couverte par une station de réseau de l'opérateur.

COMPRESSION : opération informatique consistant à transformer une suite de données A en une suite de données B plus courte pouvant restituer les mêmes informations en utilisant un algorithme* particulier. La décompression est l'opération inverse de la compression.

C : un des langages de programmation les plus utilisés. Il est issu de la programmation système. C'est un langage de bas niveau, chaque instruction du langage est conçue pour être compilée en un nombre d'instructions machine assez prévisible en termes d'occupation mémoire et de charge de calcul. Ce langage est donc extrêmement utilisé dans des domaines comme la programmation embarquée sur microcontrôleurs, les calculs intensifs, l'écriture de systèmes d'exploitation et tous les modules où la rapidité de traitement est importante. De nombreux langages plus modernes comme C++* ou Java* reprennent des aspects de C.

C++ : un des langages de programmation les plus populaires. C'est un dérivé du C apportant certaines nouvelles fonctionnalités comme la programmation orienté objet, les fonctions virtuelles, la surcharge d'opérateur, l'héritage, les templates ou encore la gestion des exceptions. Comme le C, le C++ est un langage très performant.

C# : langage de programmation orienté objet créé par la société Microsoft. Il a été créé afin que la plateforme Microsoft .NET soit dotée d'un langage permettant d'utiliser toutes ses capacités. Il est très proche du Java* dont il reprend la syntaxe générale ainsi que les concepts (la syntaxe reste cependant relativement semblable à celle de langages tels que le C++* et le C). Un ajout notable à Java* est la possibilité de surcharge des opérateurs, inspirée du C++*.

CSS : langage informatique qui sert à décrire la présentation des documents HTML et XML.

FONCTION DE HASHAGE : fonction particulière qui, à partir d'une donnée fournie en entrée, calcule une empreinte servant à identifier rapidement, bien qu'incomplètement, la donnée initiale. Les fonctions de hachage sont utilisées en informatique et en cryptographie.

GARTNER : entreprise américaine de conseil et de recherche dans le domaine des techniques avancées. Elle mène des recherches, fournit des services de consultation, tient à jour différentes statistiques et maintient un service de nouvelles spécialisées.

GOOGLEFS : système de fichiers distribué propriétaire. Il est développé par Google pour leurs propres applications. Il ne paraît pas être publiquement disponible. GoogleFS a été conçu pour répondre aux besoins de stockage de données des applications Google, notamment pour tout ce qui concerne ses activités de recherche sur le Web. Il est optimisé pour la gestion de fichiers de taille importante (jusqu'à plusieurs giga-octets), et pour les opérations courantes des applications Google : les fichiers sont très rarement supprimés ou réécrits, la plupart des accès portent sur de larges zones et consistent surtout en des lectures, ou des ajouts en fin de.

GPRS : norme pour la téléphonie mobile dérivée du GSM, permettant un débit de données plus élevé. Le GPRS est une extension du protocole GSM : il ajoute par rapport à ce dernier la transmission par paquets. Cette méthode est plus adaptée à la transmission des données.

HP WEBOS : système d'exploitation mobile propriétaire fonctionnant grâce à un noyau Linux*. D'abord appelé « Palm webOS », il a été renommé HP webOS le 19 octobre 2010 suite au rachat intervenu quelques mois auparavant de Palm par HP. Le 18 août 2011, HP a annoncé qu'il allait cesser la production de tous les appareils webOS. Le constructeur annonce fin 2011 la prochaine libération du projet webOS et sa distribution en logiciel libre.

HTML5 : cinquième version du format de données conçu pour représenter les pages Web : l'HTML. C'est un langage de balisage permettant d'écrire de l'hypertexte. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du Web. Cette cinquième version facilite le développement d'application Web avec l'apport de nouvelles API permettant de dessiner en 2D, de jouer des vidéos ou des sons ou encore d'accéder à la géolocalisation.

IETF : Internet Engineering Task Force, littéralement traduit de l'anglais en « Détachement d'ingénierie d'Internet » est un groupe informel, international, ouvert à tout individu, qui participe à l'élaboration de standards Internet. L'IETF produit la plupart des nouveaux standards d'Internet.

IP : pour Internet Protocol est le protocole principal de la suite des protocoles d'Internet. Il permet un service d'adressage unique pour l'ensemble des terminaux connectés. Il propose aussi un système de routage permettant l'interconnexion des réseaux et ainsi de transférer des données à travers différents réseaux.

ITU-R : Secteur des radios communication de l'International Telecommunication Union (ITU). Son rôle est de gérer le spectre des radiofréquences et les orbites des satellites de télécommunications et de développer les standards des systèmes de radiocommunication afin d'assurer une utilisation efficace du spectre des radiofréquences.

JAVA : langage de programmation informatique orienté objet. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux*, avec peu ou pas de modifications. Pour cela, diverses plateformes et Framework associé visent à guider, sinon garantir, cette portabilité des applications développées en Java.

JAVASCRIPT : langage de programmation de scripts principalement utilisé dans les pages Web interactives, mais aussi côté serveur. C'est un langage orienté objet.

LIBRAIRIE / BIBLIOTHÈQUE : collection de fonctions, compilée et prête à être utilisée par des programmes. Les bibliothèques sont enregistrées sous la forme d'une collection de fichiers de code objet rassemblés accompagnée d'un index permettant de retrouver facilement chaque fonction. Les fonctions contenues dans les bibliothèques sont typiquement en rapport avec des opérations fréquentes en programmation : manipulation des interfaces utilisateurs, manipulation des bases de données ou les calculs mathématiques.

LINUX : noyau de système d'exploitation de type Unix. Le noyau Linux est un logiciel libre développé essentiellement en langage C par des centaines de bénévoles et salariés communiquant par Internet. Le noyau est le cœur du système, c'est lui qui s'occupe de fournir aux logiciels une interface pour utiliser le matériel.

LOTUS DOMINO : produit IBM qui fournit une plateforme de gestion électronique des documents développée en mode open source, qui inclut une messagerie électronique et des applications de travail collaboratif. Il est rebaptisé IBM Domino depuis la version 9.

MÉMOIRE VIVE : mémoire informatique dans laquelle un ordinateur place les données lors de leur traitement.

MICROSOFT EXCHANGE : un logiciel pour serveur de messagerie électronique créé par Microsoft, pour concurrencer Lotus Domino* d'IBM. Microsoft Exchange est très utilisé dans les entreprises, 65 % du marché de la messagerie professionnelle en France. Il est conçu pour la messagerie électronique, mais aussi pour la gestion d'agenda, de contacts et de tâches. Il assure le stockage des informations et permet des accès à partir de clients mobiles et de clients Web.

MODÈLE OSI : pour Open Systems Interconnection, est un standard de communication, en réseau, de tous les systèmes informatiques. C'est un modèle de communications entre ordinateurs qui décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions. Il se décompose en sept couches :

- La couche « physique » est chargée de la transmission effective des signaux entre les interlocuteurs.
- La couche « liaison de données » gère les communications entre 2 machines adjacentes, directement reliées entre elles par un support physique.
- La couche « réseau » gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.
- La couche « transport » gère les communications de bout en bout entre processus.
- La couche « session » gère la synchronisation des échanges et les « transactions », permet l'ouverture et la fermeture de session.
- La couche « présentation » est chargée du codage des données applicatives, précisément de la conversion entre données manipulées au niveau applicatif et chaines d'octets effectivement transmises.
- La couche « application » est le point d'accès aux services réseaux.

NFC : pour Near Field Contact ou Communication en champ proche, est une technologie de communication sans-fil fréquence, permettant l'échange à courte portée et haute d'informations entre des périphériques jusqu'à une distance d'environ 10 cm.

NOVELL GROUPWISE : logiciel de courriel, groupe de travail, et messagerie instantanée pour Linux* et Microsoft Windows côté serveur, et Linux*, Microsoft Windows, Mac OS et PDA côté client.

OBJECTIVE-C : langage de programmation orienté objet. C'est une extension du C, comme le C++, mais qui se distingue de ce dernier par sa distribution dynamique des messages, son typage faible ou fort, son typage dynamique et son chargement dynamique. Contrairement au C++, il ne permet pas l'héritage multiple. Aujourd'hui, il est principalement utilisé dans les systèmes d'exploitation d'Apple : Mac OS X et son dérivé iOS.

OPENSSL : outils de chiffrement comportant deux bibliothèques (libcrypto fournit les algorithmes* cryptographiques, libssl implémente le protocole SSL) et une interface en ligne de commande (openssl). Les bibliothèques (qui sont écrite en langage C) implémentent les fonctions basiques de cryptographie et fournissent un certain nombre de fonctions utiles. Les paramètres de l'outil en ligne de commande openssl sont très nombreux ; ils permettent entre autres de choisir l'un des nombreux types de chiffrement (Blowfish, DES ou Triple DES, DSA, RC4, RC5, RSA...), d'encodage (base 64...) ou de hachage (MD5, SHA-1...). Cet utilitaire et les bibliothèques associées sont disponibles pour la plupart des Unix dont Linux* et Mac OS X, mais aussi pour Microsoft Windows, DOS et OpenVMS.

PROCESSEUR : composant de l'ordinateur qui exécute les programmes informatiques.

QUADTREE : structure de données de type arbre dans laquelle chaque nœud a quatre fils. Les quadtrees sont le plus souvent utilisés pour partitionner un espace bidimensionnel en le subdivisant récursivement en quatre nœuds.

QNX : système d'exploitation UNIX commercial temps réel, conçu principalement pour le marché des systèmes embarqués. L'entreprise qui le développe appartient à RIM. Il est considéré à la fois comme léger, robuste, rapide et complet.

QT : API orientée objet et développée en C++ par Qt Development Frameworks, filiale de Digia. Qt offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. Qt permet la portabilité des applications qui n'utilisent que ses composants par simple recompilation du code source. Les environnements supportés sont les Unix (dont Linux*), Windows, Mac OS X, Tizen* et Symbian*.

RÉALITÉ AUGMENTÉE : désigne les systèmes informatiques qui rendent possible la superposition d'un modèle virtuel 3D ou 2D à la perception que nous avons naturellement de la réalité et ceci en temps réel.

SGBD : logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

SMARTPHONE : téléphone mobile disposant aussi des fonctions d'un assistant numérique personnel. Il fournit des fonctionnalités basiques comme : l'agenda, le calendrier, la navigation sur le Web, la consultation de courrier électronique, de messagerie instantanée, le GPS, la photographie numérique, etc.

SQL : pour Structured Query Language, en français langage de requête structurée, est un langage informatique normalisé servant à effectuer des opérations sur des bases de données relationnelles. La partie langage de manipulation de données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles. La partie langage de définition de données permet de créer, et de modifier l'organisation des données dans la base de données, la partie langage de contrôle de transaction permet de commencer et de terminer des transactions, et la partie langage de contrôle de données permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes.

SYMBIAN : système d'exploitation pour téléphones portables et PDA conçu par Symbian ltd. Il est né d'un consortium de différents constructeurs. Le 16 novembre 2006, 100 millions de téléphones mobiles ont été vendus avec cet OS. Il est adopté par différents fabricants de téléphones portables. Il est acheté en 2008 à 100 % par Nokia. À la suite de cet achat Nokia décide de changer la licence de Symbian OS et d'en faire un logiciel open source le 21 octobre 2009 (mais reviendra sur cette décision en avril 2011). Le 11 février 2011, le nouveau PDG de Nokia ancien cadre de Microsoft, annonce qu'il abandonne le système d'exploitation pour être remplacés par Windows Phone.

SYSTÈME D'EXPLOITATION : premier programme exécuté lors de la mise en marche de l'ordinateur. Il sert d'intermédiaire entre les logiciels applicatifs et le matériel informatique. Un système d'exploitation apporte commodité, efficacité et capacité d'évolution, permettant d'introduire de nouvelles fonctions et du nouveau matériel sans remettre en cause les logiciels.

TIERS DE CONFIANCE : personne physique ou morale mettant en œuvre des signatures électroniques reposant sur des architectures d'infrastructure à clés publiques.

TIZEN : système d'exploitation open source multiplateforme, conçu pour un usage sur smartphones, tablettes, TV connectées et les équipements automobiles. Les composants logiciels principaux sont Linux* et WebKit. Les applications Tizen sont principalement des applications Web, donc des applications HTML5 au sens large du terme, qui fonctionnent sans navigateur Web et hors ligne. Tizen fait partie de la Linux Foundation. Le développement technique est dirigé par Intel et Samsung, la partie commerciale par la Tizen Association.

UMTS : technologies de téléphonie mobile de troisième génération (3G). Une amélioration importante de l'UMTS par rapport au GSM consiste, grâce à une nouvelle technique de codage, en la possibilité de réutiliser les mêmes fréquences dans des cellules* radio adjacentes et en conséquence d'affecter une largeur spectrale plus grande à chaque cellule* (5 MHz). Cela permet en UMTS d'avoir plus de bande passante et donc plus de débit (ou plus d'abonnés actifs) dans chaque cellule*.

XAML : langage déclaratif développé pour les besoins des systèmes d'exploitation de Microsoft, Windows Vista, Windows 7, Windows 8 et Windows Phone qui permet la description de données structurées. Il s'agit d'un dialecte XML. Il permet de séparer le développement de l'interface utilisateur du code métier. Un designer pourra utiliser le logiciel Expression Interactive Designer pour générer un fichier XAML qui sera ensuite intégré par le développeur dans l'application.

Liste des tableaux, schéma et illustrations

Tableau

Illustrations

Schéma

Table des matières

Bibliographie

148apps.biz, « Application Category Distribution », avril 2013 (date de consultation).

Article publié par le site Web « 148apps.biz » qui est mis à jour chaque semaine listant la répartition des applications mobiles par catégorie. Cet article est accessible à l'adresse suivante :

http://148apps.biz/app-store-metrics/?mpage=catcount

Dmitry Zaykovskiy, « Survey of the Speech Recognition Techniques for Mobile Devices », juin 2009.

Étude de Dmitry Zaykovskiy du département des technologies de l'information à l'université d'Ulm en Allemagne datant de juin 2009. Cette étude présente les différentes approches possibles pour implémenter un système de reconnaissance vocale pour mobile. L'étude est disponible à l'adresse suivante :

http://www.eurasip.org/Proceedings/Ext/SPECOM2006/papers/014.pdf

eMarketer, « Consumers Spending More Time with Mobile as Growth Slows for Time Online », 22 octobre 2012.

Article publié par l'entreprise indépendante d'étude de marchés liés au marketing numérique eMarketer publié le 22 Octobre 2012 sur son site Internet : www.emarketer.com. EMarketer se base sur une méta-analyse des estimations issue des recherches d'autres sociétés, de la consommation des médias des utilisateurs et de la pénétration des appareils dans les foyers. L'article montre que le temps passé sur les mobiles, hors communication, a doublé durant les deux dernières années, passant de 37 minutes en 2010 à 82 minutes en 2012. L'article est disponible à l'adresse suivante :

http://www.emarketer.com/newsroom/index.php/consumers-spending-time-mobile-growth-time-online-slows/

Ericsson, « Ericsson Mobility Report », novembre 2012.

Rapport réalisé par l'entreprise de télécommunication suédoise Ericsson en novembre 2012. Ericsson a réalisé des mesures du trafic, depuis les premiers jours du haut débit mobile, sur une majorité des réseaux mondiaux. Ce rapport effectue une analyse de ses mesures montrant l'évolution du nombre d'abonnés, du trafic, de la couverture, de la vitesse et de l'utilisation du réseau mobile. Ce rapport est accessible à l'adresse suivante :

http://www.ericsson.com/res/docs/2012/ericsson-mobility-report-november-2012.pdf

Gene Roddenberry, Star Trek, 1960.

Univers de science-fiction créé par Gene qui regroupe six séries télévisées, douze longs métrages, des centaines de romans, de bandes dessinées et des dizaines de jeux vidéo. Dans l'univers Star Trek, l'humanité développe le voyage spatial suite à une période post-apocalyptique du milieu du 21ème siècle. Plus tard, l'homme s'unit à d'autres espèces intelligentes de la galaxie pour former la Fédération des planètes unies. À la suite d'une intervention extraterrestre et grâce à la science, l'humanité surmonte largement ses nombreux vices et faiblesses terrestres au 23ème siècle. Les histoires de Star Trek dépeignent souvent les aventures d'êtres humains et d'espèces extra-terrestres, ainsi que les nombreux contacts de ceux-ci avec d'autres civilisations.

George Lucas, Star Wars, 1977.

Épopée cinématographique de science-fiction créée par George Lucas en 1977. L'action se déroule « Il y a bien longtemps, dans une galaxie lointaine » et se fonde sur la lutte entre les chevaliers Jedi et les Sith. Le personnage central Anakin Skywalker cède à la tentation du côté obscur de la Force pour devenir Dark Vador puis connait sa rédemption grâce à l'action de son fils, Luke.

GiffGaff.com, « Mobile Maps, Navigation Applications & How GPS, WiFi, Network Positioning Works », 22 novembre 2011.

Article publié sur le site Web « GiffGaff.com » le 22 novembre 2011 expliquant le fonctionnement du GPS sur mobile. Lien de l'article :

http://community.giffgaff.com/t5/Blog/Mobile-Maps-Navigation-Applications-amp-How-GPS-Wi-Fi-Network/ba-p/2198631

Hugo Etiévant, « Obfuscation : protection du code source contre le reverse engineering », 8 octobre 2006.

Article publié par Hugo Etiévant sur le site développez.com. Ce document présente les principes, les différentes techniques et les limites de l'obfuscation de code. L'article est disponible à l'adresse suivante :

http://cyberzoide.developpez.com/securite/obfuscation/

IDC, « Android and iOS Combine for 92.3% of All Smartphone Operating System Shipments in the First Quarter While Windows Phone Leapfrogs BlackBerry », 16 mai 2013.

Communiqué de presse de l'entreprise IDC. IDC pour International Data Corporation est une société américaine d'étude, de conseil, d'analyse du marché spécialisé dans l'informatique, les technologies et les télécommunications. Ce communiqué presse publie les chiffres et l'évolution des commandes et des parts de marché des différents systèmes d'exploitation mobiles entre le premier trimestre 2012 et le premier trimestre 2013. Lien vers le communiqué de presse :

http://www.idc.com/getdoc.jsp?containerId=prUS24108913

Leena Marya et B. Yegnanarayana, « Extraction and representation of prosodic features for language and speaker recognition », Speech Communication volume 50, décembre 2008.

Publication de l'European Association for Signal Processing (EURASIP) et de l'International Speech Communication Association (ISCA) parue dans le journal Speech Communication volume 50. Dans cette publication, Leena Marya et B. Yegnanarayana proposent une nouvelle approche pour extraire les paramètres de la voix directement à partir du signal.

Michael Menne, « Development and Integration of a Navigation Component into an Automotive Embedded System », octobre 2007.

Thèse dont le sujet de cette thèse est le développement d'une solution de navigation pour voiture et de son intégration. Chaque composant de cette solution est expliqué, en particulier l'architecture de la base de données cartographique. Voici le lien vers la thèse :

https://www.fbi.h-da.de/fileadmin/personal/j.wietzke/mein_ordner/Studentenarbeiten/Menne_-_Masterarbeit_-_Navi_-_rv2.pdf

Neal Stephenson, L'Âge de diamant, 1995.

Roman de science-fiction de l'auteur américain Neal Stephenson paru en 1995. Il raconte l'évolution d'une jeune fille défavorisée qui vit dans un monde dont tous les aspects sont déterminés par les nanotechnologies. La thématique du roman exploite aussi bien des problèmes d'éducation ou de classes sociales que le tribalisme culturel, en passant par les possibles réponses sociétales à un monde en proie à de grands changements technologiques.

Satoshi Nakamoto, « Bitcoin: A Peer-to-Peer Electronic Cash System ».

Spécification originale du système BitCoin écrit par son créateur sous le pseudonyme de Satoshi Nakamoto. Ce document décrit le fonctionnement de chaque composant du système de monnaie électronique décentralisé BitCoin. Il est accessible à l'adresse suivante :

http://bitcoin.org/bitcoin.pdf

Sebastian OLSSON et Philip ÅKESSON, « DISTRIBUTED MOBILE COMPUTER VISION AND APPLICATIONS ON THE ANDROID PLATFORM », 2009.

Thèse soutenue par Sebastian OLSSON et Philip ÅKESSON décrivant la mise en place d'architecture embarquée et distribuée d'un système de reconnaissance vocale sur la plateforme mobile Android. Ils mettent en évidence les possibilités et la limite de ces systèmes sur des téléphones portables moderne. La thèse peut être consultée à l'adresse suivante :

http://www2.maths.lth.se/vision/publdb/reports/pdf/olsson-akesson-master-09.pdf

TechHive, « How It Works: Speech Recognition », 4 avril 2000.

Article publié par le site Web www.techhive.com. Cet article explique simplement le fonctionnement global d'un système de reconnaissance vocale avec ces principales étapes. Il liste aussi les principaux acteurs du marché ainsi que la configuration requise pour les faire fonctionner. L'article est accessible à l'adresse suivante :

http://www.techhive.com/article/16276/article.html

Annexes

Annexe I : Exemple d'obfuscation de code

Voici une liste d'exemple, non exhaustive, de technique d'obfuscation.

Style du code

Transformation des identifiants

Méthode aléatoire : une chaine de caractère aléatoire unique est générée pour chaque identifiant à obfuscer.

Méthode Overload Induction : la chaine la plus simple est donnée à chaque identifiant, par exemple : a(), puis b(), ... aa()... pour les fonctions.

Méthode d'invisibilité : une chaine de caractère comportant des caractères spéciaux interdits par le langage et les principaux décompilateurs est générée pour chaque identifiant.

Code non obfuscé :

```
public synchronized void put(int key, Employee value) {
        Integer I = new Integer(key);
        super.put(I, (Object) value);
}
```

Obfuscé par méthode aléatoire :

```
public synchronized void yrwla35rn3(int sbhc8wduot, k0j9y980ek 78nrx59777f)
{
        Integer f841593p5r = new Integer(sbhc8wduot);
        super.yrwla35rn3 (f841593p5r, (Object) 78nrx59777f);
}
```

Obfuscé par méthode Overload Induction :

```
public synchronized void a(int a, b c) {
        Integer d = new Integer(a);
        super.a(d, (Object) c);
}
```

Obfuscé par méthode d'invisibilité :

```
public synchronized void #~a(int @b, f# a~) {
        Integer #~b = new Integer(@b);
        super.#~a(#~b, (Object) a~);
}
```

Suppression du style de codage

Code Java respectant une convention de style :

```
public class BankImpl extends BankPOA {

   private HashMap entries;

   public void unregisterBranch(String number) throws ProblemException {

      try {

         Branch b = (Branch) (entries.get(number));

         if(b != null) {
             entries.remove(number);
             System.out.println("the branch " + number + " is deleted")
             return;
         } else {
             throw new ProblemException("BankImpl.unregisterBranch",
                                        "no branch reference found !");
          }

      } catch(ProblemException e) {
          System.out.println("BankImpl.unregisterBranch : error in
                              unregistering – " + e.getMessage());
      }

   }
}
```

Code Java dont le style a été détruit :

```
public class BankImpl extends BankPOA{private HashMap entries;public void
unregisterBranch(String number) throws ProblemException{try{Branch
b=(Branch)(entries.get(number));if(b!=null){entries.remove(number);System.o
ut.println("the branch "+number+" is deleted");return;}else{throw new
ProblemException("BankImpl.unregisterBranch","no branch reference found
!");}}catch(ProblemException
){System.out.println("BankImpl.unregisterBranch : error in unregistering -
" + e.getMessage());}}}
```

Données

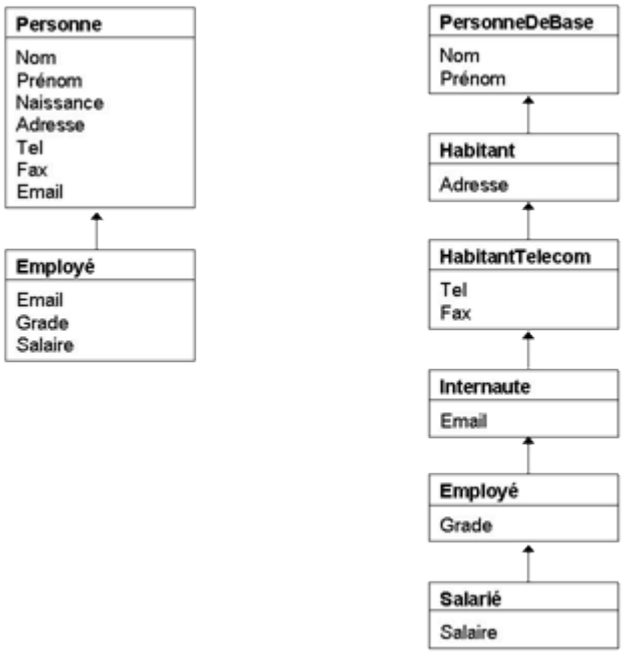
Cryptage simple de donnée

Cryptage d'une adresse email en JavaScript :

```
<script type="text/javascript">
<!--
   Ch=new Array(4);
   Res=new Array(4);
   Ch[0]='le_club_des_developeur';
   Ch[1]='ÙÆÈÏàä';
   Ch[2]='¬×ÄÇÍØÖÈÓÓ';
   Ch[3]='ÐÊÕÈØäÒÏÉß¡ÂÓÒ';
   for(y=1;y<4;y++){
      Res[y]="";
      for(x=0;x<Ch[y].length;x++)
         Res[y]+=String.fromCharCode(Ch[y].charCodeAt(x) –
                 Ch[0].charCodeAt(x));
   }
   document.write('<a href="'+Res[1]+':webmaster'+Res[2]+'– '
                  +Res[3]+'">webmaster'+Res[2]+' –'+Res[3]+'</a>');
//-->
</script>
```

Structure de l'application

Architecture des classes



Structures de contrôle

Différentes techniques :

- l'insertion de code mort (instructions n'impactant pas les traitements ni les données de l'application) :

```
private void InitializeComponent()
{
   this.button_crypter = new System.Windows.Forms.Button();
   this.NeRienFaire("APXS"); // code mort
   this.button_decrypter = new System.Windows.Forms.Button();
   this. FaireRienDuTout(new System.Windows.Forms.Button()); // code mort
   this.button1 = new System.Windows.Forms.Button();
   this.EncoreRien(50, -10); // code mort
   this.SuspendLayout();
   this.AutreRien(); // code mort
}
```

- l'augmentation de la quantité de tests et de structures de contrôle :

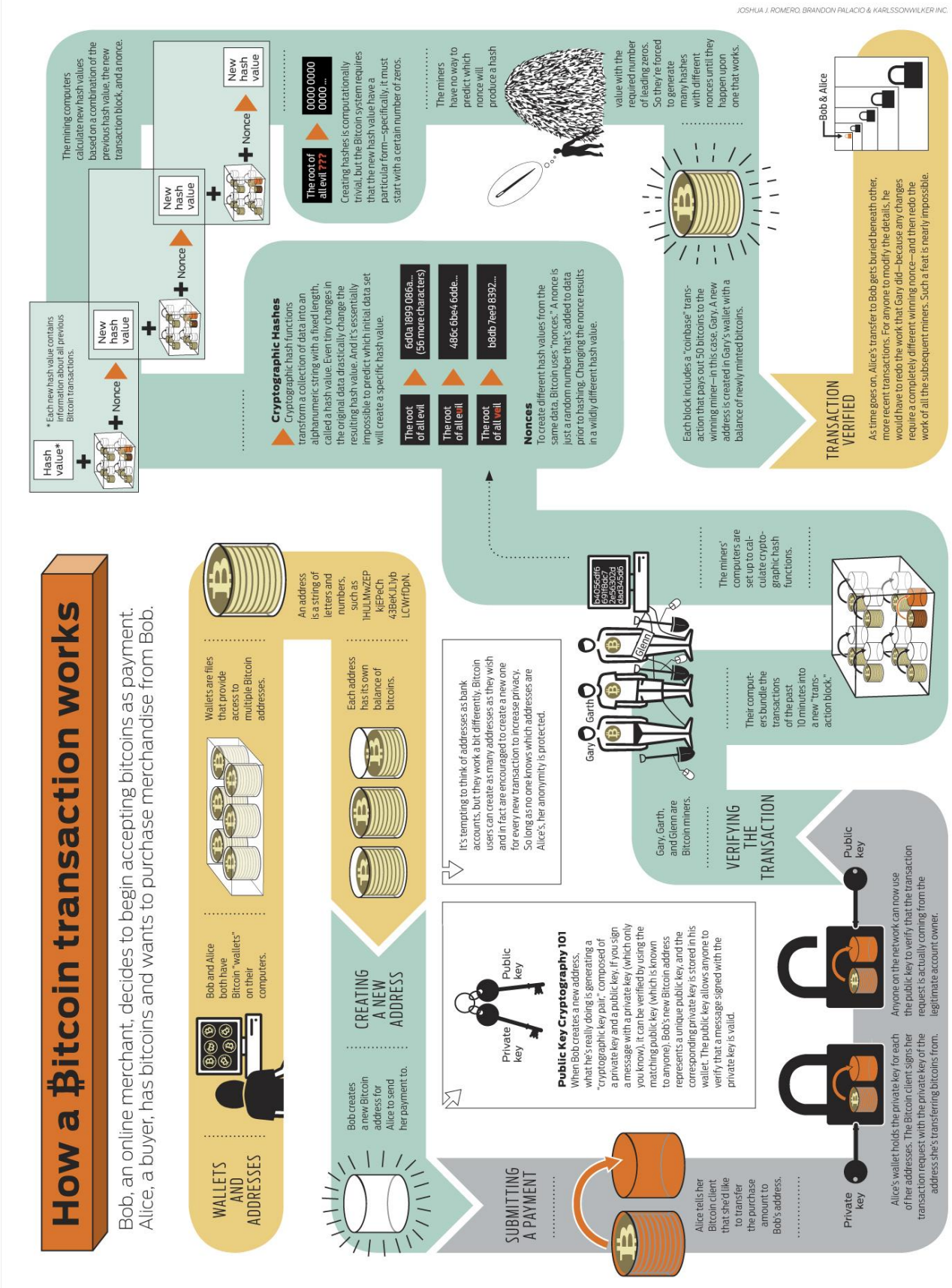
Code non obfuscé :

```
for(int i=1; i<=n; i++) {
   for(int j=1; j<=n; j++) {
      tab[i,j] = fct(i,j);
   }
}
```

Code obfuscé :

```
for(int I=1; I<=n; I+64) {
   for(int J=1; J<=n; J+64) {
      for(int i=I; i<=min(I+63,n); i++) {
         for(int j=J; j<=min(J+63,n); j++) {
            tab[i,j] = fct(i,j);
         }
      }
   }
}
```

Annexe II : Schéma d'une transaction BitCoin



Annexe III : Résumé en anglais

Today, mobile phone occupies a more and more important place in our daily life. Indeed, according to the « Ericsson Mobility Report » of November 2012, 61% of the world's population, or 4.1 billion of people, have a mobile, where 1.3 billion are Smartphones. Moreover, a study of the « eMarketer » website reveals that users spend 82 minutes per day on their phones, communication not included, in 2012, while they spent only 22 minutes in 2009. Mobile phones being always more present and used make him a perfect support for the development of a mobile assistant. In fact, such a market represents a large base of potential users. Users who want to get one will only have to download the assistant's application on their phone. No more need to go to the store and buy a new device. We must also pay attention to the price of a mobile application, far less than a complete device, which allows to interest and reach a wider audience.

The goal of this memory is, on one hand, to show the potential of mobile as a daily life assistant and the obstacles to its development. On the other hand, he proposes possible solutions to overcome these problems.

Smartphones have therefore capabilities allowing them to be used as an assistant. With first speech and image recognition systems. We can distinguish two types of voice recognition. The first one is used as an interface between the user and the assistant. It will allow to interact and to communicate in a more natural way. The assistant will meet with a number of built-in voice commands. The second one allows the user to dictate notes, SMS, mails or search directly to the phone. Image recognition serves to recognize object on an image, most often a picture taken with the smartphone. This allows the assistant to offer information about the recognized object like the details of a place or the product's name and price.

In addition, mobile will be able to position itself in space. First, thanks to a GPS receiver which will allow to locate the user. The GPS position will serve to all assistant offering location-based service, from the simple search of near points of interest (theatre, restaurant, shop, etc.), to GPS navigation. Smartphones also includes a magnetometer for determining the three-axis orientation of the device. This ability is mainly used by assistant who offer a part of augmented reality to visualize located information.

Finally, the mobile will be able to offer some services like commercial, banking or payments services. Banking services allow, through the application, to access to all its account information as well as some functions like fund transfer or set an alert. Commercial services include applications which allow to consult an online catalog and to purchase items. Lastly, the mobile payment use one of the latest technologies appeared on mobiles, the NFC for Near Field Contact, to make payments directly on a NFC terminal.

Mobile phones already have features which can be used by daily-assistant's application. Moreover, these features are constantly evolving and include always more new technologies allowing them to offer an assistant always more relevant.

However, they also have some limitations. Indeed, the mobile has limited resources which don't allow running an assistant requiring too much power. This mainly concerned the CPU (Calcul Processing Unit or processor) which provide computing power, the storage memory which allow to save data and the RAM used to store application's data during their execution. In addition, if an assistant needs data from phone's sensors, its accuracy will depends of the quality of these sensors. However such sensors with a non-negligible cost are not favoured by manufacturers. Moreover, there are different mobile operating systems on the market requiring each its own version of the assistant. This obviously leads to an increase in workload.

Even if we use a server-oriented architecture to avoid mobile phones limitations, we will deal with other problems. The main being the increase of the system's response time due to the data transfer between mobile and server.

There are also limitations for assistant which need to use artificial intelligence. This is, to date, not enough developed to reproduce a human reflection. This particularly affects recognition systems that should not have this aspect of contextualization, natural to humans, allowing, for example, understanding a sentence even if it is poorly articulated.

Mobility also represents a constraint in some cases. First, for the data security. Indeed, like all communications on Internet, the data transfer between mobile and server can be intercepted by a malicious person.

This can be especially risky when it concern sensitive data such as personal ones or banking information. Data stored on the phone can also present some security issues. Foremost, in case of the theft of the mobile, the thief will have access to all your data. But it also concerns the industrial espionage. In fact, once the application is installed on the smartphone, we can easily get its source code and its database. At last, the mobility condition leads to use the assistant in different types of environments. These are not really suited to some uses like those requiring GPS or internet reception or voice recognition.

Nevertheless, there are solutions to overcome problems mentioned in the previous section. First of all, regarding the diversity of mobile operating systems. Through the use of web application, we will be able to run the same application on any platform. In fact, these are web pages which only need a web browser to run. However, this solution has some limits. It can be uses only for simple task and doesn't achieve the performance of native applications. Another solutions is the code sharing. Here, we will share as much code as can be run on every platform. We will still have some specific code for each platform regarding the user interface and the access to some specific services and sensors.

Democratisation and deployment of the new generation of mobile telecommunication technology, the 4G, will allow to reduce the communication time between mobile and server. With this, we will be able to get airtime up to 140 times faster than third generation technologies. The 4G is being deployed since 2012 and already has 58 million subscribers.

For the data security solutions, we have first the TLS, for Transport Layer Security, which will protect data exchanged between mobile and server. This technology use different encryption algorithm - symmetric and asymmetric encryption and message authentication code with hash function – to secure, first, the encryption key exchange, and then the communication themselves. TLS is very robust but can be resource-intensive. We must therefore use it only when necessary, i.e. with sensitive data.

For the data stored in the phone, the best solution remains to leave algorithms and sensitive data on a secure server. However, if you really need to protect your application, you can use obfuscation.

It’s an operation which will make the source code as unreadable as possible but will always be executable by a computer. This solution is not intended to entirely secure the application but rather to make the reverse engineering task very difficult and if possible to dissuade them.

Recently, a new concept of mobile daily assistant has emerged. It uses data aggregation to predict which services and information the user will need. To achieve this, it cross data from different source, such as mailbox, online calendar, web browser or social networks, to determine what the user will need at this precise moment. Thanks to this, this kind of assistant will be able, for example, to warn the user when he should leave to get at time to his work or an appointment following current road traffic.

Nevertheless, such an assistant need to store and manipulate a huge amount of data which will cause some performance issues with standard solution. It’s therefore required to implement Big Data solutions. These solutions mainly consist to setting up a distributed architecture. This kind of architecture will allow distributing database and their treatments on multiple servers.

First, this requires a distributed file system which enables the storage of very large files on different machines. This kind of file system divides files in chunk, and stores each of them on different server. For more security, each chunk will be replicated several times to be uses as backup or for multiple accesses.

Then, we need a database management system (DBMS) which allows handling of large amount of data in an efficient way. For this, we will not use the traditional relational database management system but contrary a non-relational DBMS with NoSQL, for Not only SQL, database. NoSQL database use simplicity of design to offer a more efficient data manipulations.

Finally, in order to optimize the handling of these large databases, it’s necessary to parallelize and distributes its process. To do it, we can use an implementation of MapReduce which will divide the request in sub request and make them executed by different computers.

The architecture for this kind of assistant required an important investment, especially for the purchase of servers. If fact, it required enough server to store the amount of data and to carry the load when all users use the service. This solution is therefore reserved for large companies that already have such architecture and having access to the necessary data sources.

For a daily assistant to be efficient, he needs to know the entire user's life. This is true as much for a virtual assistant than a human one, and especially for those using aggregation. Indeed, the relevant of these systems' result depends entirely of the quality and the quantity of available information about the user.

This raises the question of privacy when such assistant is able to know where a person is, where it goes, what are his interests, what research she has done on the internet, etc. In fact, the goal of any business is to make profits. Thus, when they offer free service, its means that they earn money in a different way. In most cases, they monetize the personal information of their users mainly through targeted advertising.

This way, in recent years, the monetization of data has become a major economic stake. In 2013, we consider that collection, control and exploitation of data represent 312 billion dollars, or 234 billion euros. This corresponds to 38% of the world's online business turnover.

Subject of debate, the news is marked by private life's violation cases. This highlights the vagueness surrounding the data use. At the same time, it underlines the fact that, today, major IT companies are able to track each activity, move and action make by their users. The issue of privacy thus became a major stake combining legal, technical and especially ethics matter.