Ministère de l'**E**nseignement **S**upérieur et de la
Recherche **S**cientifique
Université de **M**onastir
--*-*-*
Institut **S**upérieur d'**I**nformatique et de **M**athématiques de **M**onastir

Projet de Fin d'Etudes

En vue d'obtention du

Diplôme de Mastère professionnel en génie logiciel

Sécurisation d'une application eHealth de suivi de patient à domicile

Réalisé par

Amira Kallel

Membres du Jury :

M. Mohamed Tounsi (Président)
Melle. Zouhour Ben Dhiaf (Rapporteur)
M. Christophe Ponsard (Encadrant - CETIC)
M. Nizar Ben Neji (Encadrant - ANCE)
M. Mohamed Taha Bennani (Encadrant - ISIMM)

Année universitaire : 2012/2013

Remerciements

Ce stage a nécessité tout au long de sa durée l'aide et le soutien de plusieurs personnes.

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique de l'ISIMM et les intervenants professionnels responsables de la formation mastère professionnel en génie logiciel, pour avoir assuré la partie théorique de celle-ci.

Je remercie également Monsieur Mohamed Taha BENNANI pour l'aide et les conseils concernant les missions évoquées dans ce rapport, qu'il m'a apporté lors des différents suivis.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance à Monsieur Christophe PONSARD, chef de département Software and System Engineering de CETIC, mon tuteur, pour m'avoir accordé toute sa confiance ; pour le temps qu'il m'a consacré tout au long de cette période, sachant répondre à toutes mes interrogations ; sans oublier sa participation au cheminement de ce rapport.

Monsieur Nizar BEN NEJI, chef de service à l'ANCE, pour son accueil, la confiance qu'il m'a accordée dès mon arrivée à l'agence et sa coopération professionnelle et pour m'avoir guidé et conseillé au cours de mes différentes missions.

Dédicace

A l'âme de mon très cher père Abdel Hamid

Aucune dédicace ne peut exprimer mon amour, mon respect et mon plus grand attachement que j'ai toujours eu pour toi.

A ma très chère mère Chadlia

Tu es toujours pour moi une source d'amour, de tendresse et d'espoir. Aucune dédicace ne peut exprimer les sacrifices que tu m'as donnés jusqu'à présent. Puisse Dieu, le Tout-Puissant, te protéger et te donner santé, longue vie et bonheur.

A ma sœur Anissa et mon beau-frère Hosni

A mon frère Aymen et ma belle-sœur Jennifer

A mon frère Wajdi

A qui je dois tout,

Qu'ils veuillent trouver dans ce modeste travail, résultat des encouragements incessants et des sacrifices qu'ils ont consentis pour mes études, l'expression de ma très grande affection et de mes infinies reconnaissances. Je leur souhaite tout le succès et le bonheur du monde.

A mes très chers amis

J'admirerai toujours votre gentillesse et votre humour.
J'espère que notre amitié sera éternelle.

A tous, je dis merci et je dédie le fruit de toutes ces années d'études.

Amira

Résumé

Ce projet a été proposé par le CETIC (un centre de recherche TIC en Belgique) en collaboration avec l'ANCE (Tunisie). Il entre dans le programme de l'amélioration de l'administration électronique belge. Plus précisement, il consiste à exploiter une technologie eGov (eID) pour sécuriser une application médicale de suivi de patient à domicile.

Tout d'abord, nous avons mis en place les fonctionnalités de base de l'eID (authentification, identification, signature) garantissant les principaux aspects de sécurité (intégrité , confidentialité , non-répudiation). Ensuite, nous avons procédé à l'intégration dans une application complexe dans le domaine de eSanté nécessitant ce type de fonctions.

Mots clés : eID , eHealth , sécurité.

Abstract

The project reported here was proposed by CETIC, a Belgian ICT research center in collaboration with the Tunisian ANCE. Its scope is the improvement of the Belgian electronic administration and more specifically it targets the exploitation of the eID eGovernment technology in order to secure a medical application for monitoring patient at home.

First, we have deployed basic eID functionalities (authentication, identification, signature) ensuring key security properties (integrity, confidentiality and non-repudation). Based on this, we have demonstrated how to integrate this technology on a complex eHealth application requiring this kind of features.

Keywords: eID , eHealth , security.

Liste des figures

Figure 1: Exemple de déploiement de la plateforme [7] 17
Figure 2 : Génération des clés 22
Figure 3 : Codage et décodage d'un message 22
Figure 4 : Principe d'authentification forte 25
Figure 5: Token OTP 27
Figure 6 : Exemple d'OTP synchrone 28
Figure 7:Organisation d'une PKI 30
Figure 8 : Certificat numérique 31
Figure 9: Ajout d'un certificat dans le navigateur Internet Explorer 33
Figure 10: Signature électronique (scénario de chiffrement) 33
Figure 11: Fonctionnement de base de la biométrie 35
Figure 12: Exemple de méthode d'usurpation 36
Figure 13 : Architecture globale eHealth 42
Figure 14: Vue d'ensemble de système OpemMRS 44
Figure 15 : Critique de L'existant _Architecture d'eHealth 45
Figure 16: Solution proposée _Architecture d'eHealth 46
Figure 17 : Etapes de la méthodologie Scrum 48
Figure 18: Diagramme de Gantt du projet 49
Figure 19:Diagramme cas d'utilisation -Médecin traitant 51
Figure 20 : Paradigme client-serveur 58
Figure 21: Architecture détaillée de l'application 59
Figure 22: Architecture de l'eID applet identification 60
Figure 23 : Diagramme de séquence Identification 60
Figure 24 : Authentification par certificat 61
Figure 25 : Diagramme de séquence authentification par certificat 61
Figure 26: Architecture de l'eID applet signature 62
Figure 27: Diagramme de séquence signature de document 63
Figure 28:Modèle base de données de la plateforme eHealth (modèle de référence) 65
Figure 29: Modèle base de données de système OpenMRS 67
Figure 30:Middleware Belgique-eID 87
Figure 31 : Middleware Client Classic Toolbox 88
Figure 32: Carte de test-eID 89
Figure 33 : Exemple de Carte eID virtuelle 89
Figure 34: Capture de création d'un certificat 95
Figure 35 : Test de SSL 96
Figure 36: L'interaction entre model (applet), la vue (jsp) et partie Controller (servlet) 98
Figure 37 : Menu principale du prototype 99
Figure 38 : Choix de service 100
Figure 39: Détecter le branchement du lecteur de carte 100
Figure 40: Détecter la présence de la carte 101
Figure 41: Autoriser l'utilisation des informations d'identification 101
Figure 42: Données en cours de transmission 102
Figure 43: Transmission avec succès 102

Figure 44: Réorientation à la page résultat et affichage de données 103
Figure 45 : Générer un fichier XML 103
Figure 46 : Choix de service 104
Figure 47 : Autoriser l'utilisation des données d'identification 104
Figure 48: Lecture de l'identité 105
Figure 49 : Retirer la carte eID 105
Figure 50 : Transmission des données d'identification 105
Figure 51 : Affichage des données d'identification 106
Figure 52 : Fichier PDF généré 106
Figure 53 : Choix de service 107
Figure 54 : Autoriser l'utilisation des données d'identification 108
Figure 55: Insertion du Code PIN 108
Figure 56: Authentification réussie et affichage de l' identifiant de l'utilisateur 109
Figure 57 : Choix de service 109
Figure 58 : Autoriser l'utilisation des données d'identification 110
Figure 59 : Insertion du code PIN 110
Figure 60 : Affichage de certificat d'authentification 111
Figure 61 : Démarrage du service Apache sous fedora et vérification du mot de passe 112
Figure 62 : Authentification par certificat 112
Figure 63: Choix de service 113
Figure 64: Saisie de texte et choix de l'algorithme de signature 114
Figure 65 : Demander l'accord de la signature du texte 114
Figure 66 : Insertion du code PIN 115
Figure 67 : Signature avec succès et affichage du certificat de signature 115
Figure 68 : Choix de service 116
Figure 69 : Choix de l'algorithme de signature 116
Figure 70 : Choix de l'algorithme de hachage 117
Figure 71 : Choix de fichier à signer 117
Figure 72 : Vérification de la présence de la carte 118
Figure 73 : Validation de signature du fichier 118
Figure 74 : Insertion code PIN 118
Figure 75 : Signature du fichier en cours 119
Figure 76 : Signature avec succès 119
Figure 77 : Vérification du branchement de lecteur carte 120
Figure 78 : Autoriser l'utilisation des données d'identification 120
Figure 79 : Insertion du code PIN 121
Figure 80 : Transmission avec succès 121
Figure 81 : Affichage de résultat d'authentification 122
Figure 82 : Le Dashboard technique de eID-Services 125
Figure 83 : Dashboard technique d'OpenMRS après l'intégration des fonctionnalités eID 126
Figure 84 : Les concepts de la gestion des risques 141
Figure 85.Le processus de gestion des risques 143
Figure 86: Les différentes zones de risque 145
Figure 87: Roue de Deming 147
Figure 88: Processus de gestion des risques -ISO/IEC 27005 :2008 [15] 148

Liste des tableaux

Tableau 1 : Token USB et carte à puce PKI 29
Tableau 2 : Avantages et inconvénients des différentes techniques d'authentification 38
Tableau 3 : Critères de sécurités assurés par technologie..... 38
Tableau 4 : Comparaison des modèles de BD eHealth et OpenMRS..... 80
Tableau 5 : Tableau comparatif des cartes eID et ANCE..... 93
Tableau 6 : Tableau récapitulatif des méthodes de gestion des risques SSI 149
Tableau 7 : Avantages et inconvénients des méthodes d'analyse de risque..... 150
Tableau 8 : Comparatif des méthodes d'analyse de risque –ENISA 151

Table des matières

Remerciements ..... **i**

Dédicace ..... **ii**

Résumé ..... **iii**

Liste des figures ..... **iv**

Liste des tableaux ..... **vi**

Table des matières ..... **vii**

Introduction générale **1**

I. Présentation générale **7**

I.1 Introduction 7

I.2 Présentation de l'organisme d'accueil 7

ANCE – Agence Nationale de Certification Electronique 7

CETIC – Centre d'Excellence en Technologies de l'Information et de la Communication 8

I.3 Présentation du projet 14

I.3.1 Problématique 14

I.3.2 Cadre du projet 15

I.4 Conclusion 19

II. Etat de l'art des techniques d'authentification **21**

II.1 Introduction 21

II.2 Rappel sur les principaux aspects de la sécurité informatique 21

II.2.1 La sécurité informatique 21

II.2.2 Cryptographie asymétrique 21

II.3 Moyens d'assurer la confidentialité d'une communication 22

II.3.1 Protocole SSL 22

II.3.2 Les fonctionnalités SSL 23

II.3.3 Champs d'application du SSL 24

II.4 Techniques d'authentification forte 24

II.4.1 Les Technologies de Type OTP (One Time Password): 26

II.4.2 La Technologie PKI 28

II.4.3 La technologie biométrique 34

II.4.4 Avantages et inconvénients des différentes techniques d'authentification 36

II.5 Conclusion 38

III. Analyse et spécification des besoins de sécurité **41**

III.1 introduction 41
III.2 Etude de Système Actuel 41
III.2.1 Architecture de la plateforme eHealth 41
III.2.2 OpenMRS 43
III.3 Critique de l'existant 45
III.4 Solution proposée 46
III.5 Gestion du projet 47
III.5.1 Choix de la méthodologie de développement 47
III.5.2 Diagramme de Gantt 49
III.6 Exigences non fonctionnelles relatives à la sécurisation de la plateforme 50
III.7 Scénarios détaillés 50
III.7.1 Sprint1 52
III.7.2 Sprint2 53
III.7.3 Sprint3 55
III.8 Conclusion 56
IV. Conception 58
IV.1 Introduction 58
IV.2 Architecture de l'application 58
IV.3 Conception détaillée de la sécurisation du poste du médecin traitant 59
IV.3.1 Diagrammes de séquence 60
IV.4 Analyse structurelle de la base de données 63
IV.4.1 Modèle de base de données du système eHealth 64
IV.4.2 Modèle de base de données de OpenMRS 66
IV.4.3 Comparaison des modèles de base des données 68
IV.5 Conclusion 80
V. Réalisation 82
V.1 Introduction 82
V.2 Outils utilisés 82
V.3 Sprint 0 – Familiarisation et comparaison technologique 90
V.3.1 Comparaison de deux cartes 90
V.3.2 Configuration du serveur web en mode sécurisé HTTPS 94
V.3.3 Interaction des couches et planification des sprints 98
V.4 Sprint 1 – Identification avec la carte eID 99
V.4.1 Identification-XML 100
V.4.2 Identification-PDF 104
V.5 Sprint 2 – Authentification 107

V.5.1 Authentification-Simple 107
V.5.2 Authentification-Certs 109
V.5.3 Authentification avec certificat –ANCE 112
V.6 Sprint 3 – Mise en place de la signature numérique. 112
V.6.1 Signature-simple d'un document textuel pur 113
V.6.2 Signature -XML 116
V.7 Sprint 4 - Intégration dans OpenMRS 120
V.8 Sprint 5 - Contrôle et amélioration de la qualité de code 122
V.9 Conclusion 127
Synthèse et conclusions 128
Bibliographie 131
Annexes A 134
Annexe B 139
Concepts de base et définitions 139
Concepts de la gestion des risques 141
Le processus de gestion des risques 143
Le processus de gestion de Sécurité 147
Méthodes d'analyse de risques 148

Introduction générale

Avec le développement des réseaux informatiques, à la fois au sein des hôpitaux, mais aussi au domicile des patients, le domaine des applications eHealth est en pleine expansion. Le développement de l'e-santé est susceptible de contribuer à la protection et au progrès de la santé publique. Disposer d'un dossier médical consultable par Internet, accessible aux acteurs de soins sur l'ensemble du territoire, représente un progrès attendu par de nombreux patients, en particulier ceux qui souffrent de pathologies chroniques. Le partage de l'information médicale est aujourd'hui reconnu par tous comme la garantie d'une meilleure prise en charge médicale des malades. Elle permet aussi de remédier aux déséquilibres croissants de la démographie médicale et de lutter contre les dérives des dépenses de santé [1].

L'intégration des technologies de l'information dans le domaine de la santé doit permettre de répondre à ces enjeux économiques, démographiques et de santé publique, au bénéfice des patients. C'est pourquoi, de nombreux pays se sont engagés dans la mise en place de dossiers électroniques partagés d'envergure nationale. Nous nous intéresserons en particulier au cas de la Belgique avec laquelle une collaboration a été menée dans le cadre de ce travail.

eHealth est la plateforme belge des technologies de l'information et de la communication TIC appliquées à la santé . Elle a pour but d'optimaliser la qualité et la continuité des prestations de soins de santé et la sécurité du patient, de promouvoir la simplification des formalités administratives .Ce dernier est assuré par des prestations de services et des échanges d'informations électroniques entre tous les acteurs des soins de santé, organisés avec les garanties nécessaires sur le plan de la sécurité de l'information et de la protection de la vie privée.

Néanmoins, le développement accéléré de la numérisation des données de santé de chaque citoyen est également porteur de nouvelles menaces pour la protection de ses données, compte tenu des risques de divulgation, d'utilisation détournée et de

déformation des données liés à l'utilisation d'Internet. Nous pouvons citer les risques suivants [1]:

- Chiffrement des flux - risques de vol session par absence de chiffrement des flux http.
- Phase d'authentification - faiblesse des mots de passe.
- Gestion de session - risque d'usurpation de l'identité – risque d'accès à des données ou fonctions non autorisées.
- Contrôle d'accès - un problème de contrôle d'accès va permettre à un attaquant authentifié d'accéder en lecture à des données non autorisées mais peut également permettre de modifier ou détruire ces données, de réaliser des opérations nuisibles au niveau de l'application, voire de prendre le contrôle de cette dernière.

Il est évident qu'avec la multiplication des transmissions de données médicales et l'accroissement du nombre de personnes susceptibles d'accéder aux réseaux informatiques, le déploiement de solutions de sécurités effectives et de haut niveau est une priorité renforcée.

Dans ce contexte , on peut soulever des questions d'ordre juridique (respect de la vie privée, droits du patient), technique (sécurité et confidentialité des données, authentification, droit d'accès) et médical (relation « médecin-patient », secret médical).

Cette plateforme doit répondre aux attentes du public en matière de sécurité et de confidentialité de leurs données à caractère personnel par les techniques les plus avancées de sécurisation et d'accessibilité des données, d'authentification et d'autorisation des praticiens mais aussi satisfaire aux exigences légales définies par le droit belge.

La loi belge de protection de la vie privée (loi du 11 décembre 1998 [2]), accompagnée de l'arrêté royal correspondant, stipule différentes obligations qui doivent être respectées dans le cadre de la protection des données personnelles et en particulier des données médicales. En résumant fortement les contraintes induites par cette loi, deux points sont à considérer :

- Les accès aux données personnelles doivent être compatibles avec la finalité pour laquelle les données ont été collectées.

- Les données collectées doivent être proportionnelles à la finalité.

Concrètement, cela signifie que les accès devront être aussi restrictifs que possible en fonction de la finalité de l'accès : l'accès aux informations fera l'objet d'un contrôle strict et vigilant et ne sera accordé qu'aux personnes dûment autorisées et tenues par le secret médical.

La plateforme eHealth suit de près les évolutions, tant au niveau national qu'au niveau européen, en ce qui concerne les TIC, la sécurité de l'information, la protection de la vie privée et la politique de santé nécessaires à l'exécution des missions de la plateforme. Les points suivants constituent les conditions essentielles au déploiement sécurité de l'e-santé :

- La confidentialité des données passe par une politique d'habilitation permettant de restreindre l'accès aux seules personnes habilitées et par une authentification forte de toute personne ayant accès au dispositif, qu'elle soit professionnelle de santé ou patient.
- Le contrôle de l'identité et l'authentification des personnes et des organisations qui participent au système de santé, vise la sécurité d'une communication assistée par ordinateur. Elle garantit aussi ce que l'on appelle la non-répudiation (d'où la nécessité d'une authentification qui garantisse ces exigences au moyen de certificats).
- Une autre préconisation concerne la journalisation des connexions et des actions effectuées sur les données qui doit, pour être efficace, être connue des professionnels et donner lieu à un contrôle organisé et effectif des traces, et à des alertes. La traçabilité doit donc s'accompagner d'une sensibilisation préalable des personnels.

Nous nous intéresserons dans ce travail à la carte d'identité électronique, aussi désignée par l'anglicisme electronic IDentity (eID), qui est un nouveau type de carte d'identité mise en place dans de nombreux pays. En particulier en Belgique, qui a été un des précurseurs en la matière avec un déploiement complet depuis déjà 2009.

Une eID est constituée d'une carte à puce qui contient toutes les informations nécessaires et apparentes sur les anciennes cartes, permettant une identification et capture informatique aisée des informations de l'utilisateur.

En outre, l'eID propose des fonctions d'authentification et de signature électronique reposant sur des mécanismes PKI (clef publique/privée, la clef privée étant confiée à la carte). Ceux-ci ouvrent de nouvelles perspectives d'applications plus sûres sur Internet.

Le présent travail, réalisé au sein de CETIC[1], en collaboration avec l'ANCE[2], s'inscrit dans le cadre des améliorations des outils existants pour la sécurisation de l'application eHealth complexe en mettant enœuvre des techniques eGov. Et plus particulièrement ,comparer une solution locale (ANCE) et étrangère (eID belge).

Le présent mémoire est constitué de cinq chapitres :

Le premier chapitre situe le projet dans son contexte général. Nous présenterons d'abord l'organisme d'accueil. Par la suite, nous décriverons la problématique, ainsi que le cadre applicatif du projet.

Le deuxième chapitre est consacré à l'étude des différents concepts abordés par notre projet. Tout d'abord, nous rappellerons les principaux aspects de la sécurité. Ensuite, nous présenterons les moyens d'assurer la confidentialité d'une communication. Puis, nous exposerons les techniques d'authentification forte.

Dans le troisième chapitre, nous procédons d'une part à une étude du système existant. Nous commencerons par exposer l'architecture de la plateforme eHealth et OpenMRS. Ensuite, nous critiquerons notre système pour prévoir les risques de sécurité. Puis proposons des solutions qui limitent ces derniers. Enfin, nous présenterons la méthodologie de développement et un plan pour notre travail.

D'autre part, ce chapitre aborde aussi la spécification des besoins. Nous présenterons les besoins identifiés pour les différents types d'utilisateurs et en particulier au niveau de l'encadrement médical. Nous commencerons d'abord par évoquer les besoins non-fonctionnels de

[1] CETIC : Centre d'Excellence en Technologies de l'Information et de la Communication (situé en Belgique francophone)

la plateforme afin de sécuriser les interactions entre le médecin traitant et cette dernière. Ensuite, les cas d'utilisation impactés par ces aspects seront exposés. Ces cas d'utilisation constituent les besoins fonctionnels qui seront organisés sous forme de sprints conformément à la méthodologie Scrum.

Dans le quatrième chapitre, nous détaillons la conception de la base de données et de l'application afin d'identifier les principaux éléments et les différentes fonctionnalités du système, essentiellement du point de vue de la sécurité. Nous présenterons tout d'abord l'architecture générale du système. Dans une deuxième étape, nous procéderons à la modélisation des diagrammes de séquence détaillant la sécurisation de poste du médecin traitant. Nous analyserons enfin la structure de données d'OpenMRS pour identifier les informations qui jouent un rôle au niveau de la sécurisation de la plateforme.

Le cinquième chapitre présente la phase de développement et réalisation. En premier lieu, nous présenterons l'environnement logiciel et les outils de développement du projet qui nous ont permis de réaliser cette application. Nous exposerons l'ensemble des technologies utilisées ainsi que les raisons qui ont motivé ces choix parmi d'autres. En second lieu, nous présenterons les différentes phases de réalisation de ce projet qui s'est articulé autour d'une méthodologie Agile.

[2] ANCE :Agence Nationale de Certification Electronique

CHAPITRE

I. Présentation générale

I. Présentation générale

I.1 Introduction

Nous présentons dans ce chapitre le contexte général du projet. Dans un premier temps, nous présentons l'organisme d'accueil. Ensuite,nous décrivons la problématique, ainsi que le cadre applicatif du projet.

I.2 Présentation de l'organisme d'accueil

ANCE – **A**gence **N**ationale de **C**ertification **E**lectronique

En Tunisie l'intérêt au commerce électronique a commencé depuis 1997, l'année qui a été marquée par la création d'une commission chargée de la mise en place des stratégies du commerce (e-commerce) et du gouvernement électronique (e-gouvernement). Il a été relevé la nécessité de bâtir une infrastructure à clé publique et un cadre légal [3].

En 1999, dans le même contexte, un conseil interministériel a été établi à propos de l'économie numérique.

En 2000, la promulgation de la loi n° 2000-83 du 09 août 2000 a mis en place un cadre de législation et de réglementation des échanges et du commerce électronique, dans lequel le régime des contrats écrits s'applique aux contrats électroniques et il s' en suit, en vue de favoriser un environnement de confiance, la création de l'Agence Nationale de Certification Electronique, une entreprise publique à caractère non administratif sous tutelle du ministère de l'industrie et de la technologie qui est l'autorité de certification-racine en Tunisie.

L'ANCE représente le plus haut niveau de confiance dans le domaine de la certification électronique et de la sécurité des transactions et des échanges électroniques tels que :

- L'e-commerce où le rôle de l'ANCE consiste à sécuriser les transactions de paiement et l'authentification des différents acteurs,
- L'e-banking,
- L'e-gouvernement .

Missions

- Sécurisation des transactions et des échanges électroniques dans les domaines de e-commerce, e-gouvernement, e-banking, e-finance…
- Etablissement des accords de confiance mutuelle avec les Autorités de Certification étrangères.
- Gestion des certificats électroniques (génération, révocation, publication et conservation de certificats).
- Recherche, formation et études dans le domaine de la certification et du commerce électronique.
- Spécification des exigences fonctionnelles et de sécurité pour les dispositifs de création et de vérification de la signature électronique (Dispositifs de signature électronique qualifiés).
- Spécification des exigences de signature électronique qualifiée.

Services

- Fourniture des certificats électronique.
- Développement des solutions de signature électronique.
- Développement des solutions de sécurité qui se basent sur la certification électronique : Authentification, identification, contrôle d'accès et chiffrement basés sur la certification électronique.
- Octroi d'autorisations d'activité de fournisseur de service de certification électronique.
- Autorisation/homologation des systèmes de cryptologie.
- Autorisation pour bénéficier d'un service de téléphonie sur IP.
- Formation en signature électronique, cryptologie et ICP.
- Consulting dans le domaine de signature électronique et cryptologie.

CETIC – **C**entre d'**E**xcellence en **T**echnologies de l'**I**nformation et de la **C**ommunication

Le CETIC en quelques mots… [4]

Le CETIC est le centre belge de recherche appliquée au service des entreprises dans le domaine des Technologies de l'Information et de la Communication (TIC).

Présent depuis 2001 sur le site de l'Aéropole de Charleroi, le CETIC y a été créé à l'initiative de l'Université de Namur (FUNDP), de l'Université catholique de Louvain (UCL) et de la Faculté Polytechnique de l'Université de Mons (FPMs-UMons). Grâce aux relations privilégiées avec les différents laboratoires universitaires, les équipes du CETIC sont au cœur des progrès de la recherche, en Belgique, en Europe et dans le monde.

L'expertise du CETIC à votre service

Le CETIC fait bénéficier les entreprises de son expertise en matière d'ingénierie logicielle, de technologies orientées services innovantes et de systèmes embarqués. Ce transfert d'expertise s'organise au travers de trois axes de recherche :

- **Software & System Engineering :** aider les entreprises à concevoir des produits et services de meilleure qualité, à en assurer la fiabilité, la sécurité, le respect des normes internationales, en leur apportant un soutien méthodologique.
- **Software & Services Technologies :** aider les entreprises à exploiter plus rapidement les nouvelles architectures informatiques réparties, dynamiques, orientées services, à accélérer le processus de transformation d'information en connaissance par les technologies sémantiques, à exploiter les réelles opportunités du logiciel libre, en mettant à leur disposition une expertise technologique de pointe.
- **Embedded & Communication Systems :** aider les entreprises à embarquer plus d'intelligence et plus de connectivité dans les systèmes qu'elles développent, mettre au point les démonstrateurs technologiques, les prototypes, en exploitant et intégrant les nouvelles technologies électroniques.

Un centre de calcul, un laboratoire « génie logiciel » et un laboratoire « technologies sans fil » forment les équipements de soutien à ces trois axes.

Missions

Le CETIC stimule la recherche, offrant un support IT aux entreprises en Région wallonne afin de leur permettre d'adopter les technologies avant-gardistes, d'innover plus rapidement, de gagner du temps et de l'argent, et

de répondre aux nouveaux besoins. Grâce à l'intégration des TIC dans leurs produits et services, les entreprises se positionnent de manière plus compétitive sur le marché.

Une mission internationale… pour un rayonnement régional

Investi dans les secteurs de pointe tels que l'eSanté, le Cloud Computing, l'Open Source, la sécurité, le transport/logistique, le sans fil et la sémantique. Le CETIC a établi des accords de partenariat stratégique avec des leaders technologiques ainsi que des industriels, tant au niveau régional qu'européen, accélérant le transfert technologique au profit des entreprises belges.

Les équipes d'experts du CETIC sont en effet actives dans des projets régionaux, en particulier dans le cadre du Plan Marshall pour le développement économique de la Wallonie. Ils ont par ailleurs positionné le CETIC en tant que leader de plusieurs projets internationaux et dans de nombreux projets des 6ème et 7ème programmes-cadres européens. Outre le partenariat dans le cadre de projets de recherche collaborative, le CETIC accompagne les entreprises dans leurs initiatives d'innovation, d'amélioration de produits, procédés ou services, et réalise notamment des prestations de conseils technologiques ou méthodologiques, d'élaboration de prototypes, démonstrateurs ou études de faisabilité.

En vue de renforcer la compétitivité et le leadership de la Région wallonne, le CETIC stimule donc la recherche, offrant un support IT aux entreprises afin de leur permettre d'adopter les technologies avant-gardistes, d'innover plus rapidement, gagner du temps et de l'argent, et répondre aux nouveaux besoins.

Département d'accueil : le département SSE [5]

Le département "Software and System Engineering" (SSE) met à jour continuellement son expertise en méthodes et outils permettant d'assurer la qualité des systèmes à base de logiciels. Cette expertise couvre tout le cycle de développement de systèmes à différents niveaux de criticité. Le département transfère activement son savoir-faire aux entreprises afin de leur assurer une ingénierie de qualité de leurs développements IT.

Un large panel d'expertise

Assurer la qualité d'un système informatique reste un défi pour de nombreuses entreprises. Le CETIC depuis sa création maintient activement une expertise en :

- l'ingénierie des exigences,
- la modélisation des processus-métiers,
- l'ingénierie de la sécurité des SI,
- la qualité du code en termes de maintenabilité, sécurité, fiabilité,
- la qualité des processus de développement, en particulier au sein des PME,
- la certification logicielle, y compris pour des systèmes critiques,
- l'estimation d'effort de développement,
- la gestion de la connaissance IT.

L'expertise du département est mise au service des besoins des entreprises wallonnes par des missions d'audit, de formation, d'accompagnement, de prestation de services, voire de recherche spécifique dans plusieurs domaines-clefs du développement tels que l'ingénierie des exigences, la qualité du code, les tests, la sécurité et la sûreté.

Comme notre sujet de mémoire traite la partie de sécurité des SI, on se limitera dans la suite à la présentation des services de l'ingénierie de la sécurité.

Ingénierie de la sécurité… l'ingénierie de la sécurité SI

Les systèmes d'information sont devenus les centres nerveux de notre société technologique et mondialisée. La sécurité de ces systèmes est dès lors devenue un enjeu majeur qui est encore trop souvent traité comme une qualité de second plan.

Le CETIC dispose d'une expertise couvrant un large spectre d'une part au niveau de l'analyse (depuis la sécurité du code jusqu'à l'analyse système) ; et d'autre part au niveau de la conception en particulier au niveau de la mise en œuvre de politiques de sécurité et de solutions basées sur l'eID.

Analyse de la sécurité du code

Dans le cadre du laboratoire de génie logiciel, le CETIC dispose de nombreux outils d'analyse de code orienté qualité (CAST, Sonar), fiabilité (Polyspace) et également d'outils spécifiques à la sécurité. Ces derniers permettent de réaliser des analyses pointues de la sécurité du code détectant des classes spécifiques de vulnérabilités (injection SQL, buffer overflow, utilisation

d'instructions risquées,...) afin de mettre à jour des vulnérabilités potentielles et avérées dans le code d'un applicatif. Il s'agit à la fois d'outils Open Source (PMD, FindBugs) ou commerciaux (Fortity) qui peuvent être combinés pour obtenir des résultats optimaux.

Les outils d'analyse de code du département sont à la disposition des entreprises et en particulier des PME selon diverses modalités : analyse éventuellement sur site, mise à disposition d'une instance (si la licence le permet), formation aux outils. Les analyses produisent un rapport détaillant les vulnérabilités en fonction de critères de criticité et accompagné de recommandations pour leur correction.

Analyse de la sécurité produit selon les "Critères Communs"

Les critères communs forment une norme internationale unifiée en matière de certification de produits logiciels critiques en sécurité. A l'heure actuelle, l'état belge n'a pas encore adopté de schéma de certification, les entreprises belges doivent donc recourir aux services de centres d'évaluation étrangers pour obtenir une certification de critères communs. Le CETIC dispose d'une connaissance approfondie de la norme et de plusieurs expériences de mise en œuvre sur des produits avec des entreprises belges.

Les services proposés consistent à accompagner les entreprises ayant un produit critique en sécurité à se préparer au processus complexe de certification. Cette aide couvre à la fois la production d'une série de documents suivant une structure très précise, la production d'une série de justifications démontrant des exigences de sécurité à la fois au niveau du produit mais également du processus de développement. Plus précisément, deux types d'accompagnements sont proposés :

- **Aide à la définition d'une cible de sécurité** : en identifiant des profils de protections existants pouvant servir de base, en aidant dans le choix de la portée et du niveau d'évaluation les plus adéquats, dans l'identification des menaces et objectifs de sécurité du produit à certifier, et enfin, dans la rédaction et justification des exigences de sécurité.

- **Pré-évaluation de cibles de sécurité :** une évaluation informelle des cibles de sécurité est menée afin d'y identifier les lacunes et problèmes permettant ainsi une revue de la cible avant l'évaluation officielle.

Analyse de sécurité système de type "micro-audit"

La certification selon des normes telles que l'ISO27000 ou les critères communs sont des processus très lourds qui ne correspondent pas aux besoins des PME ou d'entreprises peu matures en matière de sécurité. Pour ces dernières une approche légère combinant une analyse des risques et du retour sur l'investissement en matière de sécurité est proposée.

Cette évaluation est largement inspirée de la norme ISO27000, et consiste en une série d'entretiens avec le responsable de la sécurité de l'organisation auditée à l'issue desquels un état des lieux de la sécurité est dressé. Sur cette base une série de recommandations sont produites prenant en compte des aspects gestion de risque en fonction de divers impacts possibles.

Conception de politiques de sécurité et leur déploiement

De plus en plus les systèmes informatiques sont composés d'une fédération de services dans un environnement distribué de type SOA/Grid/Cloud. Afin de gérer l'authentification et le contrôle d'accès dans un tel contexte, une série de standards et de technologies ont été mis au point (token SAML, langage de contrôle d'accès XACML, etc.). Le CETIC dispose d'une connaissance approfondie de ces standards ainsi que des composants permettant de les mettre en œuvre. Le CETIC a notamment mis en œuvre des solutions dans le cadre de système de workflow d'images spatiales, de système eHealth et de télécom. Des extensions spécifiques, notamment pour le contrôle d'accès géographiques (geo-XACML) ont également été expérimentées sur projet.

Le CETIC aide les entreprises dans l'analyse, la conception, le choix technologique et le déploiement de solutions de gestion de l'authentification et du contrôle d'accès dans un contexte distribué. Typiquement des moteurs d'évaluation de politiques (PEP/PDP) sont mis en œuvre, configurés avec des politiques adéquates et couplés avec des mécanismes d'authentification globaux.

Conception et développement de solutions basées sur l'eID

Le CETIC a participé activement à la phase de déploiement de la carte d'identité belge en développant des applications pionnières, en contribuant à l'émergence d'outils de développement efficaces et en menant des actions de sensibilisation.

Le CETIC peut conseiller les entreprises en matière de mise en œuvre de l'eID dans des solutions locales ou connectées à Internet exploitant les fonctions de capture de données d'identification, d'authentification ou de signature électronique. Le CETIC conseille aussi sur les implications et démarches en matière de traitement des données relevant de la vie privée.

I.3 Présentation du projet

I.3.1 Problématique

L'information gérée par les systèmes d'information fait l'objet de convoitises. Elle peut être exposée à des attaques qui exploitent des éléments vulnérables du système d'information. La sécurité des systèmes d'information a pour objet de contrer ces menaces par des mesures proportionnées aux risques pouvant peser sur la confidentialité de l'information, son intégrité, sa disponibilité ou la possibilité d'en authentifier la source et de la signer.

Les menaces sont une réalité qu'on ne peut pas nier (indiscrétion des utilisateurs, furetage, appât du gain, modification de données, vente d'information, chantage informatique, espionnage industriel ou militaire), Ces menaces s'appliquent en particulier dans le domaine de la santé : on peut imaginer le cas d'une altération de dossier médical d'un patient conduisant à une mauvaise décision d'un médecin et potentiellement entraîner le décès du patient.

Favoriser la relation de confiance entre le système d'information de santé (eHealth) et les terminaux des utilisateurs est une priorité. Par conséquent, la plateforme eHealth doit répondre aux attentes du public en matière de sécurité et de confidentialité de leurs données à caractère personnel par les techniques les plus avancées de sécurisation et d'accessibilité des données, d'authentification et d'autorisation des praticiens mais aussi satisfaire aux exigences légales définies par le droit belge.

Dans le but de moderniser les services offerts par la plateforme eHealth en matière de sécurité, nous avons fixé comme objectif de développer un prototype de module gérant des fonctions d'authentification d'utilisateurs (patient/médecin/personnel infirmier) et de signature électronique au sein d'une plateforme médicale lié au suivi médical à domicile et intégré dans un système de réseau santé.

I.3.2 Cadre du projet

La Belgique commence à développer des projets eSanté dans le but de moderniser les soins de santé et les préparer à affronter les défis de vieillissement de la population et que le corps médical et le personnel paramédical seront en nombre insuffisant pour répondre à l'augmentation de ces besoins par les moyens traditionnels, sans parler des contraintes budgétaires [6].

En Wallonie, les experts sont d'ailleurs unanimes pour dire que les TIC représentent une composante essentielle de la solution à ce défi en favorisant la prévention, l'assistance et les soins à domicile. Les entreprises qui parviendront dès lors à se positionner en tant que fournisseur de technologies dans ce secteur, disposeront d'un potentiel de croissance important, non seulement au niveau belge, mais aussi au niveau européen et probablement mondial.

Le CETIC se positionne dans ce sens depuis quelques années déjà et développe des technologies eSanté orientées services offrant notamment la possibilité aux entreprises de composer à la demande et dynamiquement des services disponibles pour répondre aux besoins spécifiques et évolutifs des utilisateurs.

Dans le cadre de l'Objectif de Convergence de la Région wallonne, le CETIC et ses partenaires ont démarré en juillet 2009 un projet de recherche visant à implémenter une plateforme intégrée innovante de composition intelligente et dynamique de services eHealth. Cette démarche vise à apporter une réponse efficace au problème de la prise en charge et du suivi coordonnés et sécurisés des patients à domicile.

Au sein de ce projet, 4 activités principales ont été identifiées :

- le design d'interfaces utilisateurs ergonomiques et adaptatives,
- la composition dynamique de services électroniques,
- la communication électronique des données enregistrées par les appareils médicaux,

- l'interopérabilité et l'échange de données avec les systèmes d'information existants et à venir.

Au sein de ce projet, un ensemble d'actions de R&D ont été identifiées et sont réalisées afin de développer une plateforme pilote de services eSanté permettant la composition de services électroniques, incluant certains aspects de la prise en charge à domicile, le télémonitoring et la prévention de certaines pathologies.

Pour ce faire, les partenaires du projet adressent un ensemble de problématiques scientifiques et techniques bien précises :

- Tout d'abord, la composition de services électroniques nécessite des mécanismes poussés de raisonnement et d'inférence sur des déclarations de services électroniques. Avancer sur la problématique de composition permet de proposer de nouveaux services aux patients.

- Le public cible de cette plateforme étant un public très varié, des interfaces « homme- machine» ergonomiques et adaptatives doivent être conçues en intégrant la multimodalité des communications (interfaces tactiles, vocales, visuelles, …). L'ergonomie de la plateforme et plus particulièrement des appareils utilisés par les patients est un facteur crucial pour l'utilisation de cette plateforme. Des interfaces « homme-machine » mal adaptées rendraient la plateforme inutilisable.

- Par ailleurs, une part importante du matériel médical actuel ne répond pas aux standards de communication et d'interfaçage les plus récents (USB, Bluetooth, Zigbee, RFID). Rendre le matériel médical plus communiquant est donc un défi à adresser permettant ainsi l'interfaçage de ce matériel avec la plateforme. De même, afin de faciliter la connexion de nouveau matériel, une standardisation des interfaces et du protocole de communication sera proposée.

- La diversité des informations médicales constitue un autre défi. Intégrer ces données, les interfacer, les extraire et les qualifier sont autant de problématiques que la plateforme doit relever. L'exploitation des données est un facteur important pour la réussite du projet. En effet, la quantité de données à exploiter est énorme et les possibilités de les exploiter de manière à créer de nouveaux services sont très nombreuses. Il est donc important de traiter cette problématique.

- Enfin, cette plateforme doit rencontrer les attentes du public en matière de sécurité et de confidentialité de leurs données à caractère personnel, mais aussi satisfaire aux exigences légales définies par le droit belge. Cette plateforme devrait être conçue de manière à limiter les facteurs de rejet des futurs utilisateurs, notamment les patients. Les critères favorisant

l'acceptation sociale de la plateforme doivent donc être étudiés afin d'orienter correctement le développement de la plateforme technologique.

La figure suivante présente un exemple de déploiement de la plateforme eHealth.

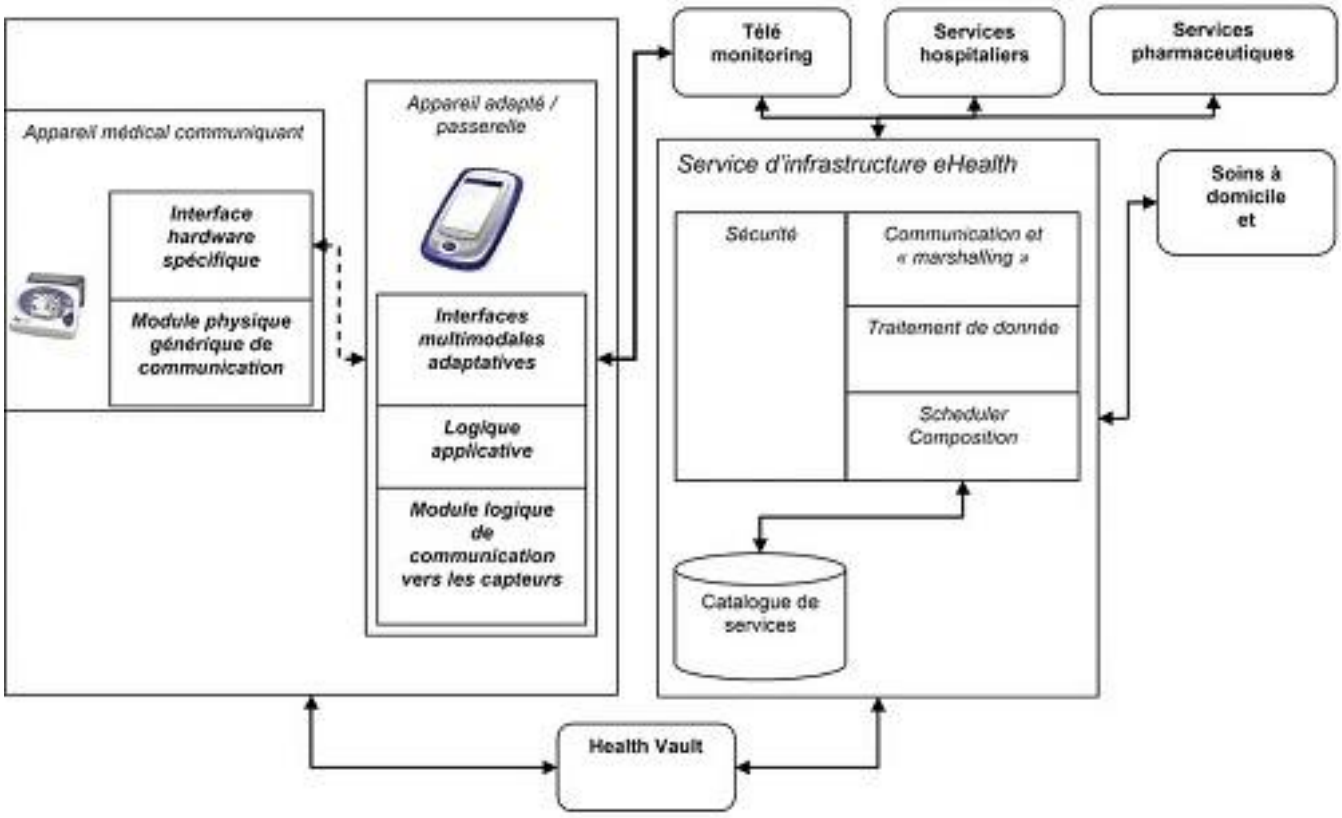


Figure 1: Exemple de déploiement de la plateforme [7]

Afin de démontrer la réussite du projet, des scénarios de démonstration de la plateforme seront mis en œuvre au travers de trois pilotes, définis avec les spécialistes de la santé, qui se renforceront mutuellement : la prise en charge de patients diabétiques dans une population variée, le suivi des personnes souffrant de la maladie de Parkinson et la prise en charge à domicile des personnes âgées souffrant de pathologies chroniques multiples. A cette fin, nous analyserons brièvement les interdépendances entre acteurs médicaux (médecins généralistes, hôpitaux, spécialistes, soins à domicile, …) dans la prise en charge de ces personnes face à des problèmes médicaux multiples tels que la maladie de Parkinson, le suivi d'encéphalogramme pour application en neurologie, l'hémostase, les maladies cardiaques, le suivi des pneumopathies,… Cette analyse succincte nous permettra de déterminer un scénario réaliste d'application de ce pilote. Ce dernier pilote sera construit en exploitant les éléments déjà élaborés dans les deux autres pilotes notamment en terme de prise en charge du diabète.

Rôles du CETIC …

Le CETIC en tant que chef de file du projet est responsable de la coordination et de la gestion du projet, est également responsable de plusieurs tâches techniques essentielles à l'élaboration de la plateforme eHealth :

- La collecte et de l'analyse des besoins auprès des différents acteurs du domaine eHealth et des utilisateurs potentiels.
- La définition et le design de l'architecture de la plateforme.
- L'implémentation de composants clés du système tels que l'architecture orientée service, les systèmes de communication et les protocoles pour appareils médicaux.
- L'implémentation des systèmes et des politiques de sécurité essentielles pour garantir la sécurité des données échangées et le respect de la vie privée.
- L'organisation et la gestion des projets pilotes visant à valider l'approche du projet et les développements réalisés.

Depuis la fin de l'année 2009, en Belgique, la totalité des cartes d'identité classiques ont été remplacées par l'eID. La carte d'identité électronique est devenue la preuve officielle de l'identité d'une personne, mais elle permet aussi l'identification électronique à distance. On doit ici faire la distinction entre l'identification simple et l'authentification qui livre une preuve effective de l'identité du titulaire.

Chaque titulaire dispose aussi d'une signature électronique à valeur légale lui permettant de signer des documents électroniquement. Il en résulte un service rapide et orienté client sans mettre les données privées du titulaire en danger.

L'outil informatique se fait chaque jour de plus en plus présent. Le citoyen belge ne pourra d'ailleurs bientôt plus échapper à la carte d'identité électronique qui matérialisera son existence obligatoire dans le monde cybernétique.

Qu'on le déplore ou qu'on s'en félicite, le monde de la santé n'échappe pas à cette évolution. La politique de la santé qui promeut l'échelonnement, l'intégration hospitalière au sein des bassins de soins et le raccourcissement des hospitalisations accroît chaque jour le besoin d'une communication immédiate et toujours plus poussée entre les professionnels de la santé [8].

I.4 Conclusion

Dans ce chapitre nous avons présenté l'ANCE et le CETIC comme organismes d'accueil. Puis, nous avons exposé la problématique de notre projet ainsi que la cadre applicatif de ce dernier.

Afin d'avoir une meilleure compréhension des concepts abordés dans ce travail, l'état de l'art des techniques d'authentification fera l'objet du prochain chapitre.

CHAPITRE

II. Etat de l'art des techniques d'authentification

II. Etat de l'art des techniques d'authentification

II.1 Introduction

Dans ce chapitre nous étudions les différents concepts définissant notre projet. Tout d'abord, nous rappelons les principaux aspects de la sécurité. Ensuite, nous présentons les moyens d'assurer la confidentialité d'une communication. Enfin, nous exposons les techniques d'authentification forte.

II.2 Rappel sur les principaux aspects de la sécurité informatique

Avec l'apparition des nouvelles technologies de l'information, les entreprises se sentent de plus en plus concernées, pour des raisons propres à chacune d'entre elles (la protection des biens et des informations personnelles ou pour des raisons stratégiques ou économiques), par la sécurité de leurs systèmes informatiques. Mettre en place une véritable politique de sécurité devient donc une obligation.

II.2.1 La sécurité informatique

Appliquer une protection optimale à un système informatique met en œuvre l'application des mécanismes fondamentaux de la sécurité informatique : l'authentification, la confidentialité, l'intégrité et la disponibilité et la non-répudiation.

- **L'authentification :** consistant à vérifier l'identité des partenaires (pas de fausse identité).
- **La confidentialité :** consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées (rendre l'information inintelligible à d'autres personnes que les seuls acteurs de la transaction).
- **L'intégrité :** c'est-à-dire garantir que les données sont bien celles que l'on croit être (pas d'altération accidentelle ou intentionnelle).
- **La disponibilité :** permettant de maintenir le bon fonctionnement du système d'information (garantir l'accès à un service ou à des ressources).
- **La non-répudiation :** permettant de garantir qu'une transaction ne peut être niée (aspect contractuel).

II.2.2 Cryptographie asymétrique

La **cryptographie asymétrique**, ou **cryptographie à clé publique**, est une méthode de chiffrement qui repose sur l'utilisation d'une clé publique (qui est diffusée) et d'une clé privée (gardée secrète), l'une permettant de coder le message et l'autre de le décoder. Ainsi, l'expéditeur peut utiliser la clé publique du destinataire pour coder un message que seul le destinataire (en possession de la clé privée) peut décoder, garantissant la confidentialité du

contenu. Inversement, l'expéditeur peut utiliser sa propre clé privée pour coder un message que le destinataire peut décoder avec la clé publique ; c'est le mécanisme utilisé par la signature numérique pour authentifier l'auteur d'un message .Voici comment se déroulent les choses :

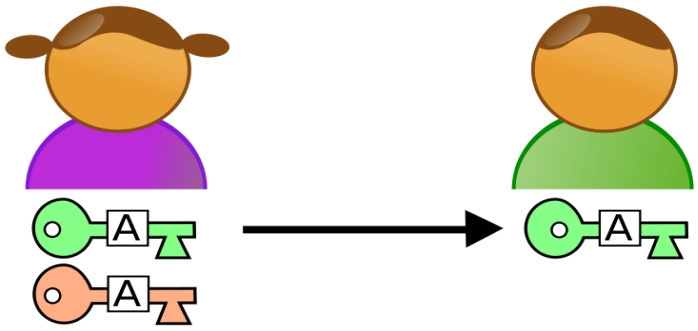


Figure 2 : Génération des clés

1° étape : Alice génère deux clés. La clé publique (verte) qu'elle envoie à Bob et la clé privée (rouge) qu'elle conserve précieusement sans la divulguer à quiconque

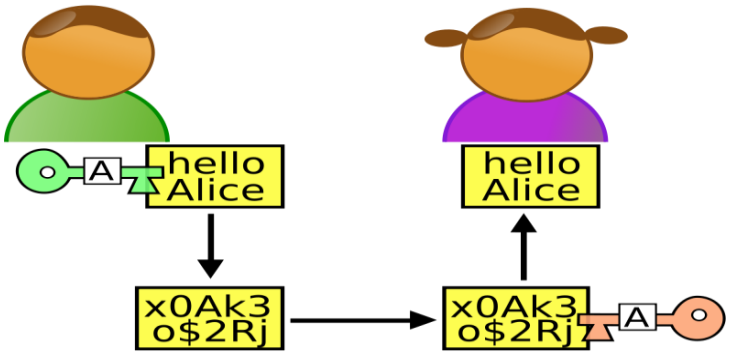


Figure 3 : Codage et décodage d'un message

2e et 3e étapes : Bob chiffre le message avec la clé publique d'Alice et envoie le texte chiffré. Alice déchiffre le message grâce à sa clé privée.

II.3 Moyens d'assurer la confidentialité d'une communication

II.3.1 Protocole SSL

SSL (Secure Socket Layer) est un protocole de sécurisation des échanges, développé par Netscape. Il a été conçu pour assurer la sécurité des transactions sur Internet (notamment entre un client et un serveur). Intégré aux principaux navigateurs et serveurs Web, SSL utilise des techniques de cryptage qui s'appuient sur un système de clés publiques/privées initialement développé par RSA. L'établissement d'une connexion SSL

nécessite l'installation d'un certificat numérique sur le serveur Web. Ce certificat utilise alors les clés publiques et privées pour le cryptage, et identifie le serveur de manière unique et définitive. Les certificats numériques s'apparentent à une forme de carte d'identité électronique qui permet au client d'authentifier le serveur avant l'établissement d'un canal de communication crypté.

II.3.2 Les fonctionnalités SSL

On associe généralement SSL au cryptage. Or, un certificat SSL remplit quatre fonctions bien distinctes, toutes indispensables à la protection de la confidentialité et de la sécurité des clients et utilisateurs : cryptage, intégrité, authentification et non-répudiation.

- **Cryptage**

Le cryptage utilise des algorithmes mathématiques pour transformer les données et les rendre exclusivement lisibles par les parties concernées. Dans le cadre d'une transaction sécurisée par SSL, les clés privées et publiques fournies avec le certificat numérique du serveur jouent un rôle déterminant dans la sécurisation des données envoyées et reçues par le navigateur Web.

- **Intégrité**

En cryptant les données pour les rendre exclusivement lisibles par les parties concernées, les certificats SSL assurent également leur intégrité. En d'autres termes, ces données ne pouvant être lues par aucun tiers, leur modification en cours de transit s'avère par conséquent impossible. Si les données cryptées étaient modifiées, elles seraient de fait rendues inutilisables – ce qui ne saurait échapper aux parties concernées. La moindre tentative d'interférence est donc automatiquement repérée.

- **Authentification**

L'émission d'un certificat numérique par une autorité de certification permet essentiellement de valider l'identité de l'organisme – ou de la personne – à l'origine de la demande de certificat. Les certificats SSL sont associés à un nom de domaine Internet. La vérification par l'AC de l'identité du propriétaire du domaine en question permet aux utilisateurs de savoir dans un premier temps à qui ils ont affaire. Ainsi, lorsque vous vous connectez sur un site sécurisé par SSL comme Amazon.fr, le certificat identifie le propriétaire comme étant la société Amazon, Inc. Vous avez ainsi l'assurance d'être sur le véritable site exploité par Amazon.

❖ **Non-Répudiation**

La non-répudiation se caractérise par l'association de fonctions de cryptage, de protection de l'intégrité et d'authentification. En clair, aucune partie prenante à une transaction sécurisée ne peut légitimement affirmer que ses échanges/communications provenaient d'une autre personne ou entité. Cette caractéristique supprime toute possibilité pour l'une des parties de répudier ou de « se rétracter » par rapport à des informations communiquées en ligne.

II.3.3 Champs d'application du SSL

Le protocole SSL peut être utilisé de diverses façons et à des fins différentes :

❖ **Communications « de navigateur à serveur » :**

La plupart du temps le SSL sert à sécuriser les communications entre un serveur Web et un navigateur, notamment dans le cadre de transmissions d'informations sensibles (achats en ligne, dossiers médicaux ou transactions bancaires). La technologie SSL permet de confirmer à l'utilisateur l'identité du destinataire de ses informations personnelles, tout en assurant que seule cette entité autorisée y aura accès.

❖ **Communications « de serveur à serveur » :**

Le protocole SSL peut également être utilisé pour sécuriser les communications entre deux serveurs, telles que les transactions entre deux entreprises. Dans ce scénario, les deux serveurs possèdent généralement un certificat qui leur permet de s'authentifier mutuellement et de sécuriser leurs communications bilatérales.

❖ **Respect des obligations réglementaires :**

De nombreuses réglementations juridiques et sectorielles exigent des niveaux d'authentification et de confidentialité que les certificats SSL permettent d'obtenir. Le standard PCI DSS (Payment Card IndustryData Security Standard) exige par exemple l'utilisation de technologies d'authentification et de cryptage pour tout paiement en ligne.

II.4 Techniques d'authentification forte

Qu'est- ce qu'une authentification forte ?

L'authentification est une procédure qui vise à s'assurer de l'identité d'un individu ou d'un Système Informatique(SI).C'est un préliminaire indispensable à l'activation de mécanisme d'autorisation et donc, à la gestion des droits d'accès à un système d'information donné.

Les méthodes classiques pour identifier une personne physique sont au nombre de trois :

1. **Quelque chose que l'on connaît : un mot de passe ou un Code PIN.**
2. **Quelque chose que l'on possède : une carte à puce, un « Token », etc.**
3. **Quelque chose qui nous caractérise : un attribut biométrique tel qu'une empreinte digitale.**

Le choix de telle ou telle technique dépend en grande partie de l'usage que l'on souhaite en faire : authentification de l'expéditeur d'un email, authentification d'un utilisateur qui se connecte à distance, authentification d'un administrateur au système, authentification des parties lors d'une transaction de B2B (Business to Business), etc.

On parle d'authentification forte dès que deux de ces méthodes sont utilisées ensemble. Par exemple une carte à puce avec un code PIN ou une carte à puce couplée à un lecteur biométrique.

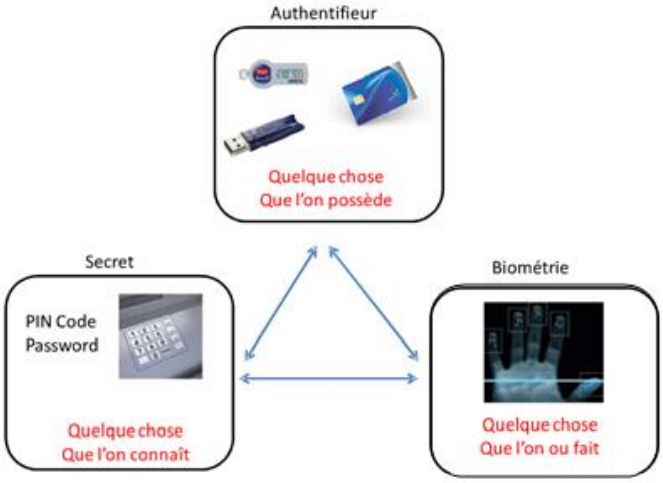


Figure 4 : Principe d'authentification forte

Les techniques d'authentification forte :

Il existe aujourd'hui un grand nombre de solutions d'authentification forte sur le marché. On peut principalement en distingués trois :

- Les Technologies de Type OTP.
- La Technologie PKI.
- Les Technologies biométriques.

II.4.1 Les Technologies de Type OTP (One Time Password):

Les mots de passe pris dans leur ensemble sont le moyen d'authentification le plus répandu à ce jour. On en distingue deux catégories **: les mots de passe statiques et les mots de passe dynamiques.**

Les mots de passe statiques sont des mots de passe qui restent identiques pour plusieurs connexions sur un même compte. Ce type de mot de passe est couramment rencontré sous Windows NT ou Unix. Cette technique d'authentification est la plus utilisée dans les entreprises mais aussi la moins robuste. En fait, les Entreprises devraient restreindre l'usage des mots de passe statiques à une authentification locale d'un utilisateur car les attaques qui permettent de capturer un mot de passe qui circule sur un réseau sont nombreuses et faciles à mettre en pratique. On recense trois grandes catégories d'attaques informatiques :

- **Attaque de « Force brute» :** il s'agit d'un attaque qui vise à deviner un (ou plusieurs) mot de passe en utilisant des dictionnaires et en testant toute les combinaisons possibles. Partant de principe que la majorité des mots de passe sont « faibles » (Combinaisons simples du type année de naissance, prénom de son fils, etc.), les découvrir rapidement se relève très facile.
- **Ecoute du Réseau :** la plupart des applications comme « TELNET », «FTP », « HTTP », «IDAP », etc., n'ont pas recours au chiffrement (encryption) lors du transport du mot de passe sur le réseau. « Ecouter » le trafic et extraire le mot de passe devient un jeu d'enfant dès lors qu'on dispose d'un logiciel d'écoute appelé « sniffer ».
- **Ecoute du Clavier :** imaginons que le trafic sur réseau soit chiffré et que l'utilisateur ait pris soin de choisir un mot de passe extrêmement « solide ». Une méthode d'attaque révèle malgré tout imparable : l'écoute du clavier ou **« key logger »** qui permet de « capturer » l'ensemble des touches tapées sur un clavier. Les logiciels utilisés dans ce cas sont généralement installés sur le poste de travail, à l'insu de l'utilisateur. La « compromission » du système informatique étant le plus souvent effectuée par le biais d'un virus de type « Cheval de Troie».

Les mots de passe dynamiques : pour palier les faiblesses de l'usage des mots de passe statiques, sont apparues des solutions d'authentification combinant deux facteurs (« ce que je possède » et « ce que je sais ») afin d'obtenir une authentification forte.

Les mots de passe sont obtenus par des générateurs de mots de passe activés à l'aide d'un code d'identification personnel ou PIN (**Personal Identification Number**).

La mise en place d'un tel mécanisme d'authentification forte rend la capture du mot de passe en cours d'aucune utilité puisque, dès que le mot de passe dynamique a été utilisé, celui-ci devient caduc.

La technologie de type OTP est très facile à déployer et offre une mobilité qui est particulièrement appréciée par des utilisateurs nomades. Généralement, les moyens d'authentification de type OTP sont embarqués sur des « Token » physiques. D'où le niveau d'authentification forte qui est obtenu en combinant un « Token »physique et un code PIN connu par l'utilisateur.

Figure 5: Token OTP

Un exemple d'OTP synchrone

Le client et le serveur doivent rester synchronisés. L'élément de synchronisation peut être soit le temps, soit un compteur d'événement, soit les deux à la fois. En plus, les deux parties partagent la même clé symétrique. Par conséquent, l'attaque devient plus complexe puisque l'information personnalisée est double.

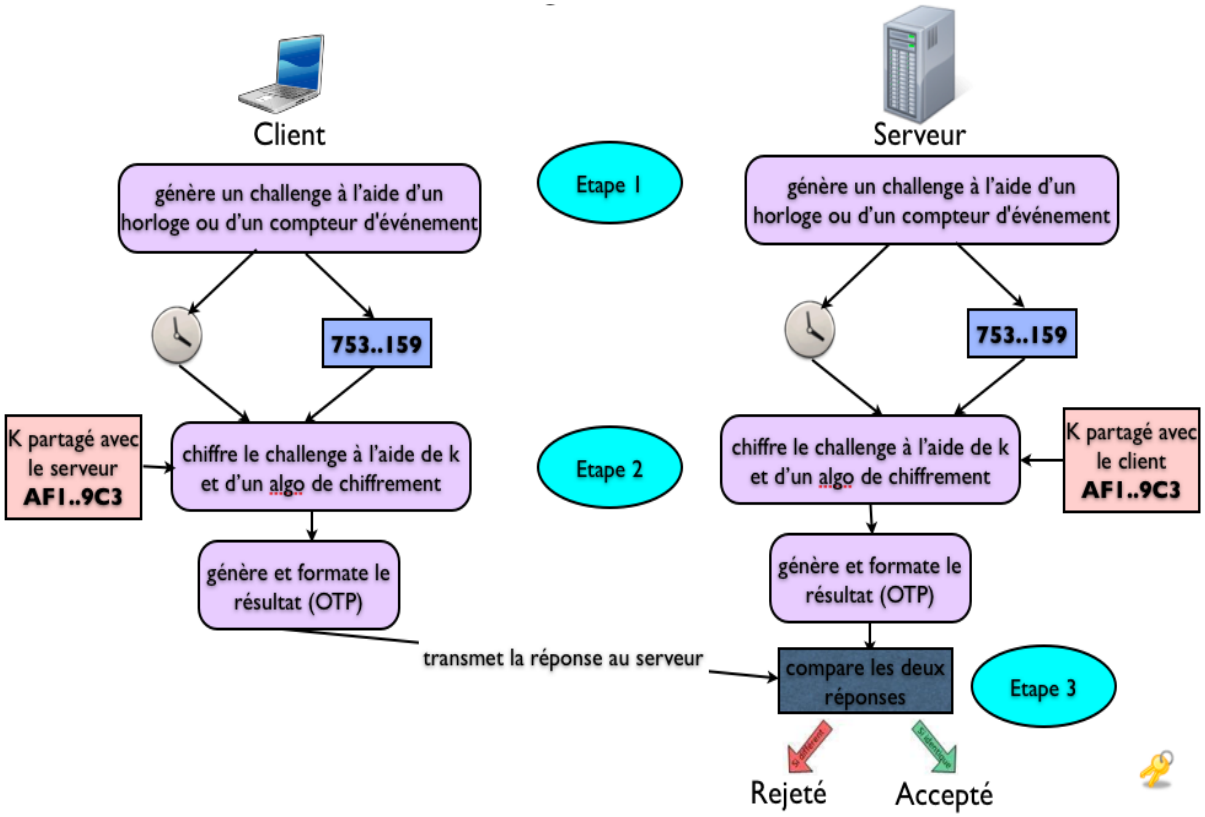


Figure 6 : Exemple d'OTP synchrone

II.4.2 La Technologie PKI

Les PKI (Public Key Infrastructure) [8] ou infrastructures à clé publique (ICP) constituent la meilleure réponse technique, organisationnelle et juridique au problème de la sécurité des échanges électroniques : messagerie, paiement sur les sites marchands, échanges B-to-B, téléprocédures administratives, etc.

Fondées sur des techniques de cryptographie asymétrique (clés publique et privée), les PKI contribuent à garantir l'authentification, l'intégrité, la confidentialité et la non-répudiation des documents échangés entre partenaires.

Dans le cadre de l'authentification forte, chaque utilisateur possède une paire de clés, ainsi que d'un certificat numérique attestant de l'identité de l'utilisateur.

Afin de monter le niveau de sécurité, le certificat numérique et la paire de clés sont stockés sur un support physique telle une carte à puce ou un Token USB

Exemples de Token USB et carte à puce PKI

eToken PRO carte à puce	**SafeNet eToken 4100**	**eToken PRO USB**
permet l'authentification forte des utilisateurs, la gestion de mot de passe, signatures numériques sécurisées, et de données de solutions de sécurité	la carte de SafeNet permet l'authentification basée sur les certificats à puce, offre hautement sécurisé (authentification forte à deux facteurs). l'authentification assurant l'accès des utilisateurs autorisés à des réseaux d'entreprise sensibles et applications professionnelles en ligne, ainsi qu'un soutien pour des solutions avancées telles que la signature numérique et d'authentification pré-boot.	eToken PRO est un authentificateur USB qui fournit l'authentification forte des utilisateurs et de gestion de mot de passe rentable.

Tableau 1 : Token USB et carte à puce PKI

Comme son nom l'indique, une ICP[3] est un ensemble de moyens matériels, de logiciels, de composants cryptographiques, mis en œuvre par des personnes, combinés par des politiques, des pratiques et des procédures requises, qui permettent de créer, gérer, conserver, distribuer et révoquer des certificats basés sur la cryptographie asymétrique.

Cette définition empruntée à l'IETF[4]montre qu'une ICP est bien plus qu'un système technique. Les politiques, pratiques et procédures décrivent le cadre de mise en œuvre des moyens dédiés à l'ICP de façon sûre, et qui correspondent à l'attente, en termes de confiance, de ses utilisateurs.

[3] **ICP : I**nfrastructure à **C**lé **P**ublique
[4] **IETF**: **I**nternet **E**ngineering **T**ask **F**orce

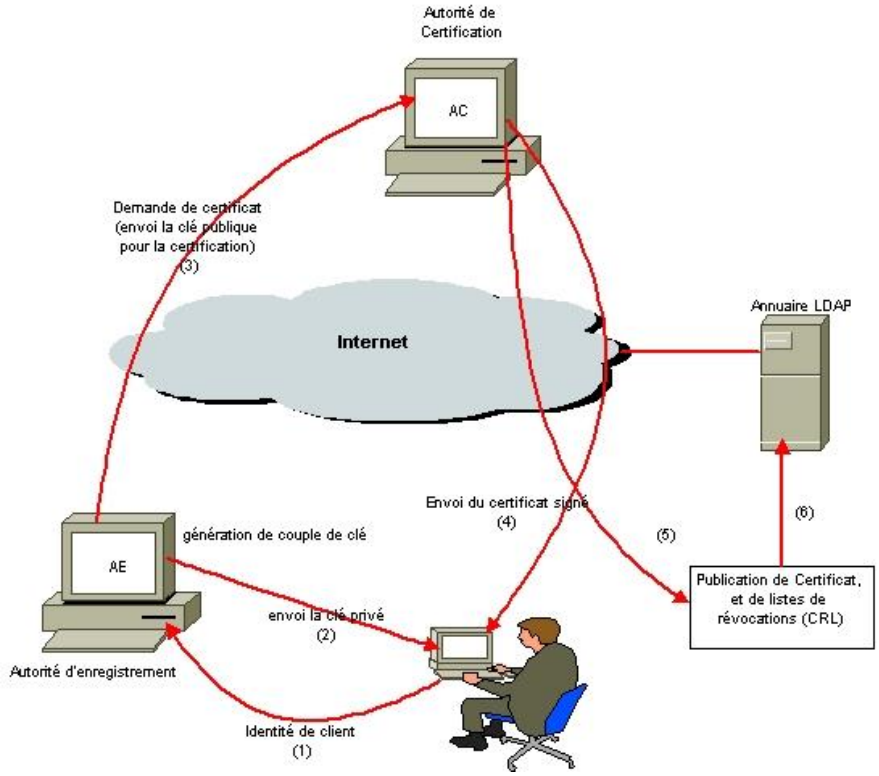


Figure 7:Organisation d'une PKI

Dans une infrastructure à clé publique, pour obtenir un certificat numérique, l'utilisateur fait une demande auprès de l'autorité d'enregistrement. Celle-ci génère un couple de clés (clé publique, clé privée), envoie la clé privée au client, applique une procédure et des critères définis par l'autorité de certification qui certifie la clé publique et appose sa signature sur le certificat parfois fabriqué par un opérateur de certification [9].
Nous allons maintenant aborder ce qui constitue l'objet premier de la présence d'une ICP, à savoir le « certificat de clé publique ».

Pourquoi un certificat de clé publique ?

La cryptographie à clé publique peut être utilisée pour chiffrer des mots de passe. En outre, elle peut également être employée pour signer des données, qu'il s'agisse d'un contrat afin que les parties qui l'ont signé ne puissent pas en répudier le contenu a posteriori, ou qu'il s'agisse d'une valeur aléatoire pour assurer l'authentification.
Les certificats de clés publiques sont l'une des techniques d'authentification les plus usitées à ce jour.

La cryptographie asymétrique :

La cryptographie asymétrique fait intervenir deux éléments qui sont mathématiquement liés entre eux : la clé privée et la clé publique.

❖ **La clé publique est :**

- Disponible pour tout le monde.
- Utilisée par une personne qui souhaite authentifier l'émetteur d'un document électronique signé avec la clé privée. Le contrôle de la signature permet aussi de s'assurer de l'intégrité du fichier.
- Utilisée par une personne souhaitant chiffrer un message afin que ce dernier soit uniquement lisible par le possesseur de la clé privée associée.

❖ **La clé privée est :**

- Conservée secrète par son possesseur.
- Utilisée par son possesseur pour signer un document électronique (message, contrat ou autre).
- Utilisée par son possesseur pour déchiffrer un message chiffré à son attention.

Mais connaissant la clé publique, est-il possible de deviner la clé privée associée ??

À ce jour, il est admis qu'en connaissant la clé publique , il est impossible en un temps raisonnable avec des moyens raisonnables de deviner la clé privée associée : l'usage de clés asymétriques est jugé sûre.

Le certificat numérique :

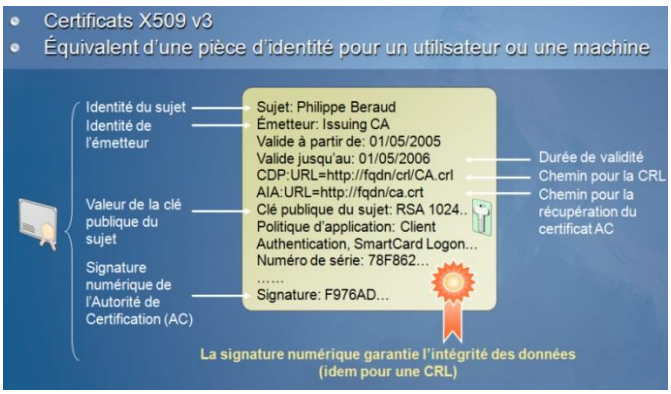


Figure 8 : Certificat numérique

Comment peut-on être sûr que cette signature (cf. modèle ci-dessous) est bel et bien celle de notre correspondant ? [10]

BEGIN SIGNATURE

QCVAwUBMARe7gvyLNSbw6ZVAQF6ygP/fDnuvdAbGIDWaSMXUIs&»%k3MuNHYzdZOOcqkDh/Tc2+DubuEa6GU03AgZY8K9t5r9lua34E68pCxegUz009b10cjNt6+o+704Z3j1yy9ijYM8BWNaSp9L2W4nUuWBdlWyel82PjjRVNZEtqtSRQuPEpJ2IHtx9tGevH10

END SIGNATURE

Cette certitude naît de la relation de confiance liant l'expéditeur en question à son bi-clé (couple constitué d'une clé privée et d'une clé publique). Ce lien appelé certificat est, en fait, un fichier contenant les paramètres cryptographiques (algorithme utilisé, taille des clés, etc.) et les données d'identification de son possesseur, le tout signé par un tiers qui en a validé le contenu. Ce tiers est appelé autorité de certification (AC).

Maintenant que l'on peut avoir la garantie qu'une personne est bien celle qu'elle prétend être.

Exemple :

La signature électronique de quelqu'un est bien de cette personne et que nous savons. Il faut pour cela, vérifier le certificat X.509 (cf. RFC 2459 Internet X.509 Public Key Infrastructure, Certificate and CRL Profile pour plus de détails sur son contenu) associé à la clé de signature et/ou remonter la chaîne de certification jusqu'au certificat de l'AC racine (certificat auto signé).

Il reste à savoir où trouver ce(s) certificat(s). Dans les entreprises, les certificats de clés publiques sont stockés dans des annuaires **LDAP** (*Lightweight Directory Access Protocol*). Pour un usage personnel de **MAC** ou autres applications utilisant la cryptographie asymétrique, les certificats des utilisateurs ainsi que de leur AC (et éventuellement, hiérarchie de certification jusqu'à l'AC racine) sont souvent stockés dans les **Navigateurs** (Internet Explorer, Firefox, Safari, Chrome)

Ceux-ci comportent déjà, par défaut, un certain nombre de certificats d'AC racines (cf. Outils/Options internet/Contenu/Certificats).

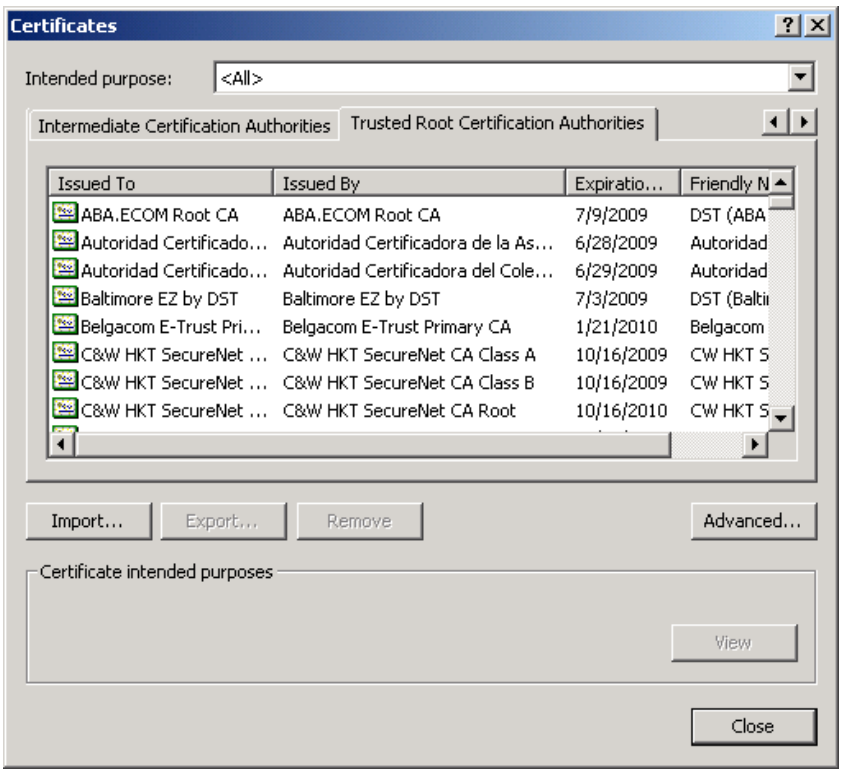


Figure 9: Ajout d'un certificat dans le navigateur Internet Explorer

Exemple d'authentification par carte à puce :

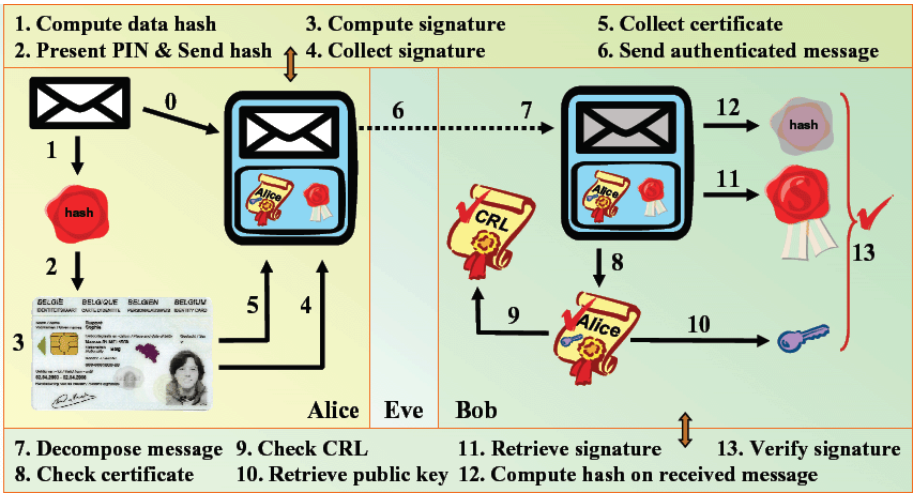


Figure 10: Signature électronique (scénario de chiffrement)

Le principe de l'authentification de l'expéditeur d'un message grâce à l'usage de la signature numérique est le suivant : l'expéditeur calcule le **M**essage **A**uthentication **C**ode ou **MAC**[5] à l'aide d'une fonction « hash » puis,

[5] **MAC** : **M**essage **A**uthentication **C**ode

l'expéditeur signe le MAC avec sa clé privée. Le **MAC** signé est joint au message et l'ensemble est envoyé.

Le destinataire fait le « hash » du message reçu en utilisant le même algorithme que celui de l'expéditeur et compare ce **MAC** au **MAC envoyé** avec le message. S'ils ne sont pas identiques, cela signifie que le message a été altéré. La modification d'un seul bit pendant la transmission fait échouer le contrôle et permet d'alerter le destinataire. Plus qu'un mécanisme d'authentification de l'expéditeur d'un message, le **MAC** permet donc aussi de garantir l'intégrité du message.

II.4.3 La technologie biométrique

Il existe actuellement un grand nombre de solutions biométriques. La majorité des solutions est déployée dans le cadre de la sécurisation des bâtiments. Cependant depuis quelques années un certain nombre de solutions voit le jour pour la sécurisation des environnements IT pour les entreprises soucieuses de leur sécurité du SI.

Les technologies sont très diverses telles que :

- Reconnaissance des empreintes.
- Reconnaissance de l'iris.
- Reconnaissance de la voix.
- ADN, etc.

Dans le cadre de l'environnement IT, il est montré que la solution basée sur les empreintes des doigts est actuellement la plus utilisée.

Le principe de fonctionnement de la biométrie :

Le principe de fonctionnement de la biométrie est basé sur la transformation d'un trait physique d'une personne en une information numérique. Le résultat de cette transformation s'appelle une « Biometric Pattern». Une fois cette information capturée et vérifiée, elle doit être stockée. Ce processus s'appelle « l'enrôlement ».

Une fois le système initialisé, l'utilisateur peut effectuer une procédure d'authentification afin de s'authentifier. Le système biométrique compare la « Biometric Pattern» stockée à celle de l'utilisateur, si elles sont identiques alors l'utilisateur est authentifié [11] [12].

Ce schéma explique le fonctionnement de base de la biométrie :

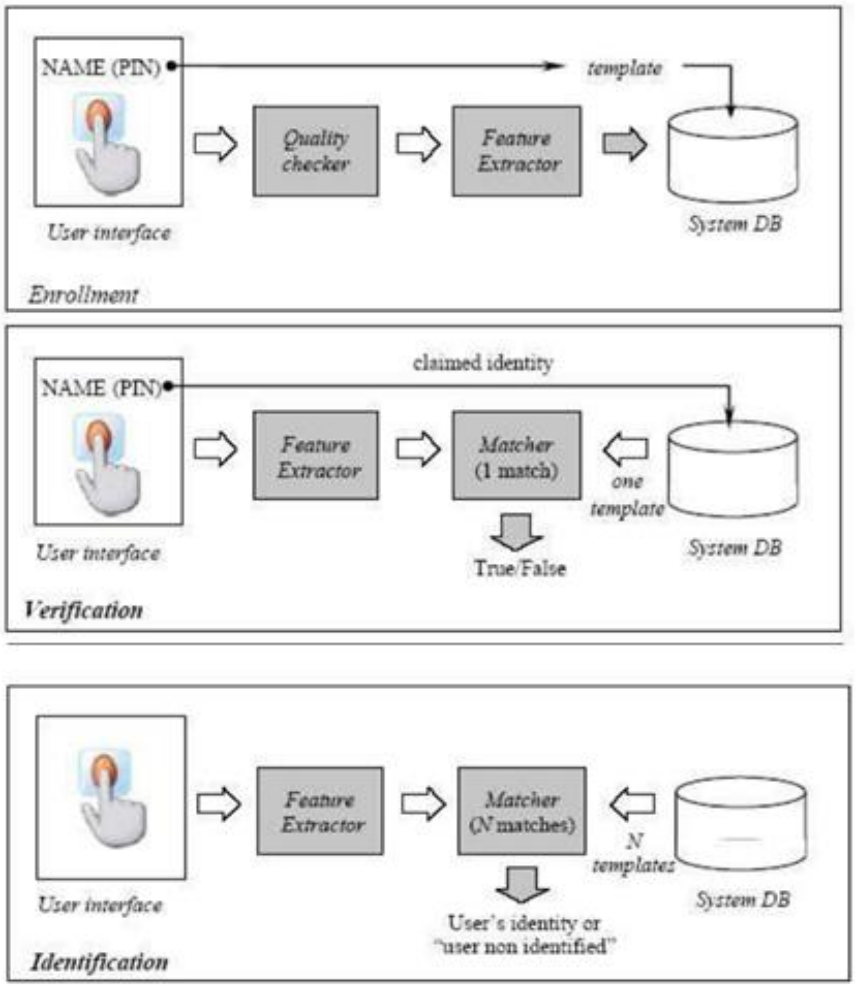


Figure 11: Fonctionnement de base de la biométrie

Niveau de sécurité des solutions biométriques :

De nombreuses études prouvent aujourd'hui qu'il est possible d'usurper l'identité d'un utilisateur par le biais de plusieurs moyens. Il existe à l'heure actuelle un grand nombre de produit biométrique offrant plus au moins de sécurité. Cependant, dans le cadre d'un projet IT pour une entreprise, le coût d'investissement est une composante très importante. Les études prouvent que sur ce genre de produit(IT), il est possible de contourner ces systèmes.

Donc, il n'est pas recommandé d'utiliser la biométrie comme facteur unique d'authentification. Par contre, la biométrie est un très bon moyen de remplacement du code PIN ou du mot de passe [13] [12].

Exemple de méthode d'usurpation :

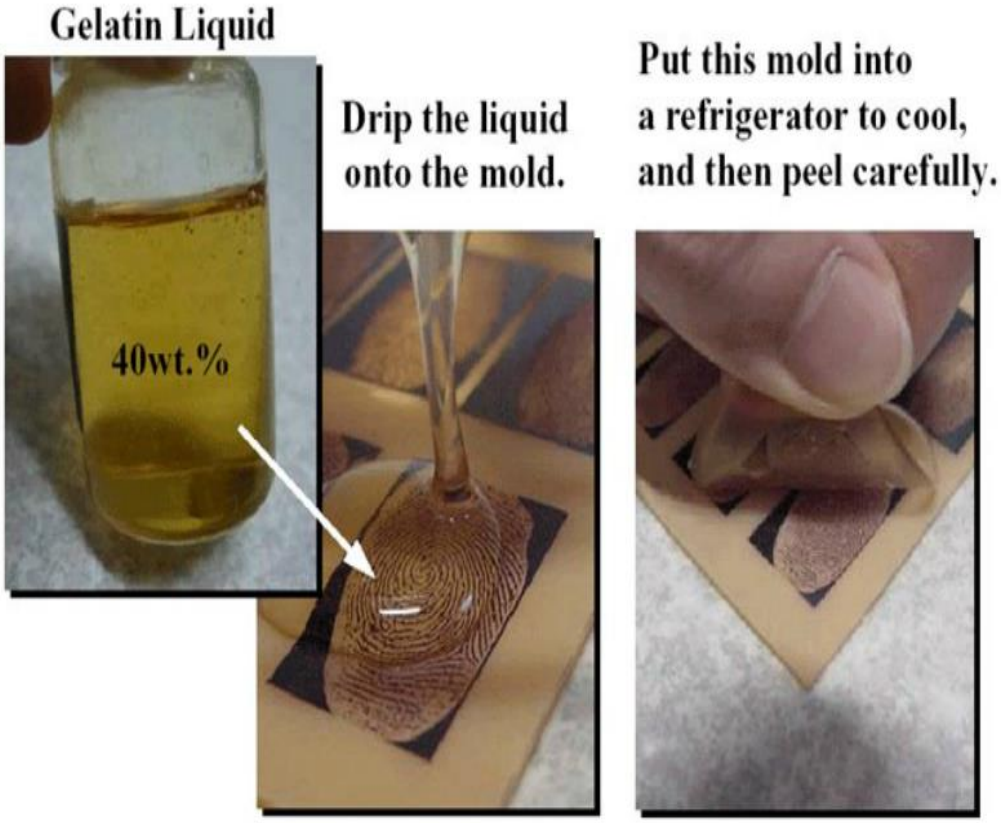


Figure 12: Exemple de méthode d'usurpation

II.4.4 Avantages et inconvénients des différentes techniques d'authentification

Le tableau suivant recense les principaux avantages et inconvénients de chacune des techniques d'authentification décrites précédemment [10]

	Avantages	Inconvénients
Mot de passe statique	+ peu coûteux + facile à mettre en œuvre + facile à utiliser	- vol du mot de passe en regardant par-dessus l'épaule - oubli du mot de passe - peu robuste (facilement devinable ou « craquable ») - partageable, et trop souvent partagé - mot de passe rejouable par une personne malveillante

Mot de passe statique stocké dans une carte magnétique activée par code **PIN**	+ robustesse du mot de passe (possibilité de choisir un mot de passe aléatoire et comprenant des caractères spéciaux) + pas de nécessité de mémoriser le mot de passe	- vol, perte ou oubli de la carte - carte partageable - durée de vie du mot de passe souvent trop longue : nécessité de renouveler régulièrement l'enregistrement dans la carte - mot de passe rejouable
Mot de passe dynamique généré par un outil logiciel	+ robustesse du mot de passe (mot de passe souvent aléatoire et comprenant des caractères spéciaux) + confort d'utilisation pour l'utilisateur (absence de mémorisation du mot de passe)	- peu de confort d'utilisation (nécessité d'utiliser un logiciel à chaque nouvelle connexion) - protection de l'utilisation du logiciel par une personne non autorisée
Mot de passe dynamique généré par un outil matériel	+ confort d'utilisation pour l'utilisateur (absence de mémorisation du mot de passe) + robustesse du mot de passe (usage unique)	- vol, perte ou oubli du générateur de mot de passe - le générateur peut se désynchroniser avec le serveur qui contrôle la vérification du mot de passe
Certificat X.509 dans le navigateur de l'ordinateur	+ multi-usage + robustesse de la méthode d'authentification + confort d'utilisation pour l'utilisateur	- vol ou utilisation frauduleuse de l'ordinateur et copie de la clé privée associée au certificat
Certificat X.509 dans un token USB	+ multi-usage + robustesse de la méthode d'authentification + attitude similaire à la possession de clés (maison, voiture)	- vol, perte ou oubli du token
Certificat X.509 dans une carte à puce	+ multi-usage + robustesse de la méthode d'authentification + attitude similaire à la possession d'une carte bancaire	- vol, perte ou oubli de la carte à puce
Biométrie et caractéristiques de référence dans une base de données en réseau	+ pas d'oubli ou de vol possible	- technologie encore immature - facilement falsifiable

Biométrie associée à une carte magnétique		- technologie encore immature - vol, perte ou oubli de la carte magnétique - coût élevé
Biométrie associée à un certificat X.509 dans un token USB	+ multi-usage + attitude similaire à la possession de clés (maison, voiture)	- technologie encore immature - vol, perte ou oubli du token USB - coût élevé
Biométrie associée à un certificat X.509 dans une carte à puce	+ multi-usage + attitude similaire à la possession d'une carte bancaire	- technologie encore immature - vol, perte ou oubli de la carte à puce - coût élevé

Tableau 2 : Avantages et inconvénients des différentes techniques d'authentification

Donc, quelle technologie choisir ? [11] [12]

	OTP	PKI (HW)	Biométrie
Authentification forte	■	■	* ■
Chiffrement	■	■	■
Signature numérique	■	■	■
Non répudiation	■	■	■
Usurpation d'identité	■	■	■

■ **possible**

■ **impossible**

* **Biométrie type Fingerprinting**

Tableau 3 : Critères de sécurités assurés par technologie

II.5 Conclusion

Une étude de concepts de base de sécurité et des moyens qu'ils assurent nous facilitera le choix de la solution qui s'adapte à sécuriser notre application.

Sur base des concepts de gestion de risque de l'information (voir annexe B), nous approfondirons dans la prochaine section les besoins de système en matière de sécurité.

CHAPITRE

III. Analyse et Spécification des besoins de sécurité

III. Analyse et spécification des besoins de sécurité

III.1 introduction

Dans cette section nous procédons d'une part à une étude sur le système existant. Nous commençons par exposer l'architecture de la plateforme eHealth et Open MRS. Ensuite, nous critiquons notre système pour prévoir les risques de sécurité. Puis, nous proposons des solutions qui limitent ces derniers. Enfin, nous présentons la méthodologie de développement et un plan pour notre travail.

D'autre part, étant la première dans le cycle de développement du projet, la spécification est la phase la plus importante. En effet, c'est au cours de celle-ci que les besoins de l'utilisateur sont identifiés et précisés. Nous commençons d'abord par évoquer les besoins non fonctionnels de la plateforme afin de sécuriser les interactions entre le médecin traitant et cette dernière. Ensuite, les cas d'utilisation impactés par ces aspects seront analysés . L'adaptation de ces cas d'utilisation est présentée ici comme des besoins fonctionnels qui seront exposés sous forme de sprints conformément à la méthodologie Scrum.

III.2 Etude de Système Actuel

III.2.1 Architecture de la plateforme eHealth

La plateforme eHealth permet le suivi des patients âgés à domicile. Comme l'illustre la figure ci-dessous, il est constitué :

- d'un partie « postclient » destinée au patient et qui est particulièrement simplifiée (terminal tactile sans clavier). Ce terminal est aussi appelé « compagnon digital ».
- d'une partie « portail médecin » qui permet un suivi à distance. Il constitue aussi une excellente base de simplification administrative pour les dispensateurs de soins en leur permettant de consacrer au maximum leur temps disponible à l'administration de soins.

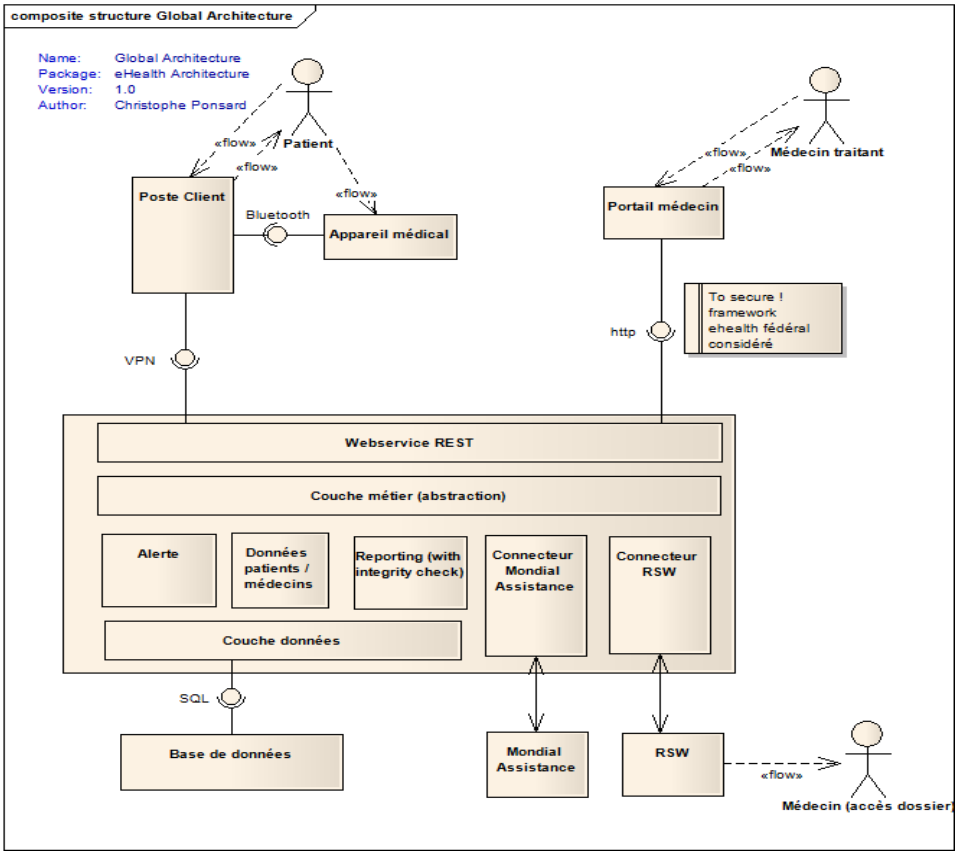


Figure 13 : Architecture globale eHealth

Le système considéré est une plateforme de suivi de patients à domicile. Les acteurs principaux de notre système sont le patient et le médecin traitant.

❖ Le patient

Le patient est responsable de son traitement mais est aidé et surveillé par la plateforme de la manière suivante:

- rappel des médications à prendre,
- prise de mesures liées à sa ou ses pathologies,
- collecte d'informations sur l'état du patient.

Il se connecte de sa machine cliente (personnalisée pour récupérer les valeurs de mesures par Bluetooth) aux services de la plateforme eHealth de suivi médical à distance par l'intermédiaire du réseau sécurisé VPN[6] qui favorise la transmission de données sécurisées

❖ Le médecin traitant

Un médecin traitant dispose d'un accès aux données d'un ensemble de patients qu'il suit et uniquement de ceux-ci. Le système aide le médecin traitant en lui remontant des alarmes en cas de détection de situations

[6] VPN : Virtual Private Network

anormales sur base d'un ensemble de règles qu'il peut personnaliser au profil de chaque patient.

Les informations collectées sont remontées au médecin traitant pour les analyser et en cas de mesures anormales prendre les décisions qu'il juge nécessaires (prise de rendez-vous pour ajuster un traitement, planification d'examens complémentaires jusqu'à l'intervention d'urgence, si nécessaire).

Par ailleurs, la plateforme est interconnectée via le réseau santé wallon. Ce dernier ne gère pas de banques de données ni d'infrastructures réseau de fond et ne développe pas lui-même d'applications de fond. En revanche, il offre des services que tous les acteurs des soins de santé et leurs prestataires de services ICT peuvent utiliser.

III.2.2 OpenMRS

OpenMRS est un système open-source de dossiers médicaux électroniques qui permet de suivre les dossiers des patients. Le système permet le suivi à distance d'un patient par son médecin traitant , où ce dernier a le droit de consulter le dossier médical de son patient suite à la présence d'une relation patient- docteur, le médecin peut adapter le traitement , demander une analyse , fixer une rencontre selon le besoin [16] [17].

Un modèle de données flexible qui détient les dossiers médicaux.

- Une interface de programmation d'application (API) qui encapsule le modèle de données.
- Une application Web qui représente l'interface-utilisateur de l'ensemble du système. Il accède au modèle de données à travers la couche API.

Le projet de base peut être étendu en utilisant OpenMRS modules pour répondre aux exigences spécifiques. Ces modules sont branchés sur le système de base de deux manières :

- Programmation Orientée Aspect (AOP) permet d'accéder aux événements des appels d'API.
- Des points d'extension permettent d'accéder à la couche d'interface-utilisateur qui est similaire à la prise en des points d'extension Eclipse.

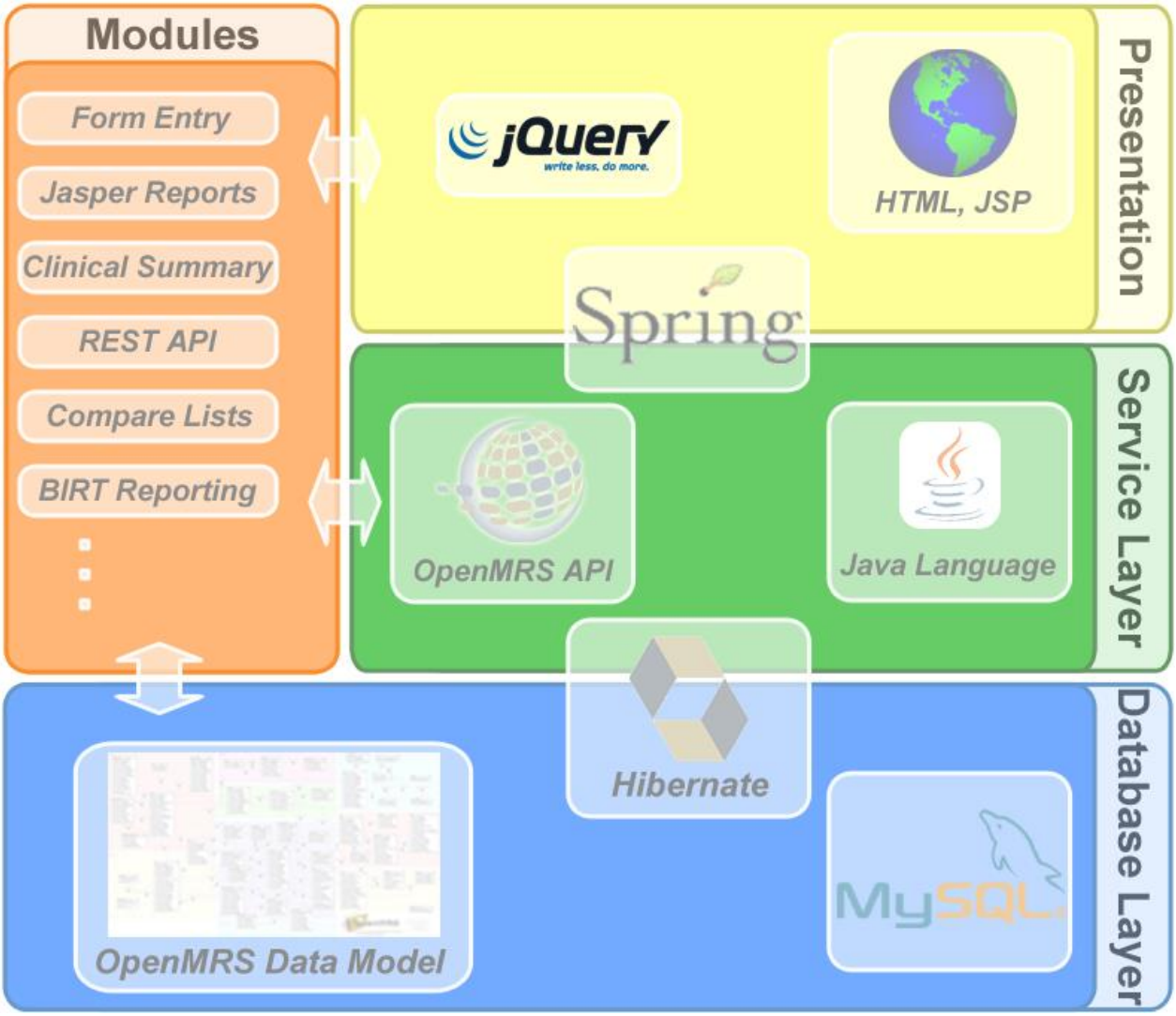


Figure 14: Vue d'ensemble de système OpemMRS

III.3 Critique de l'existant

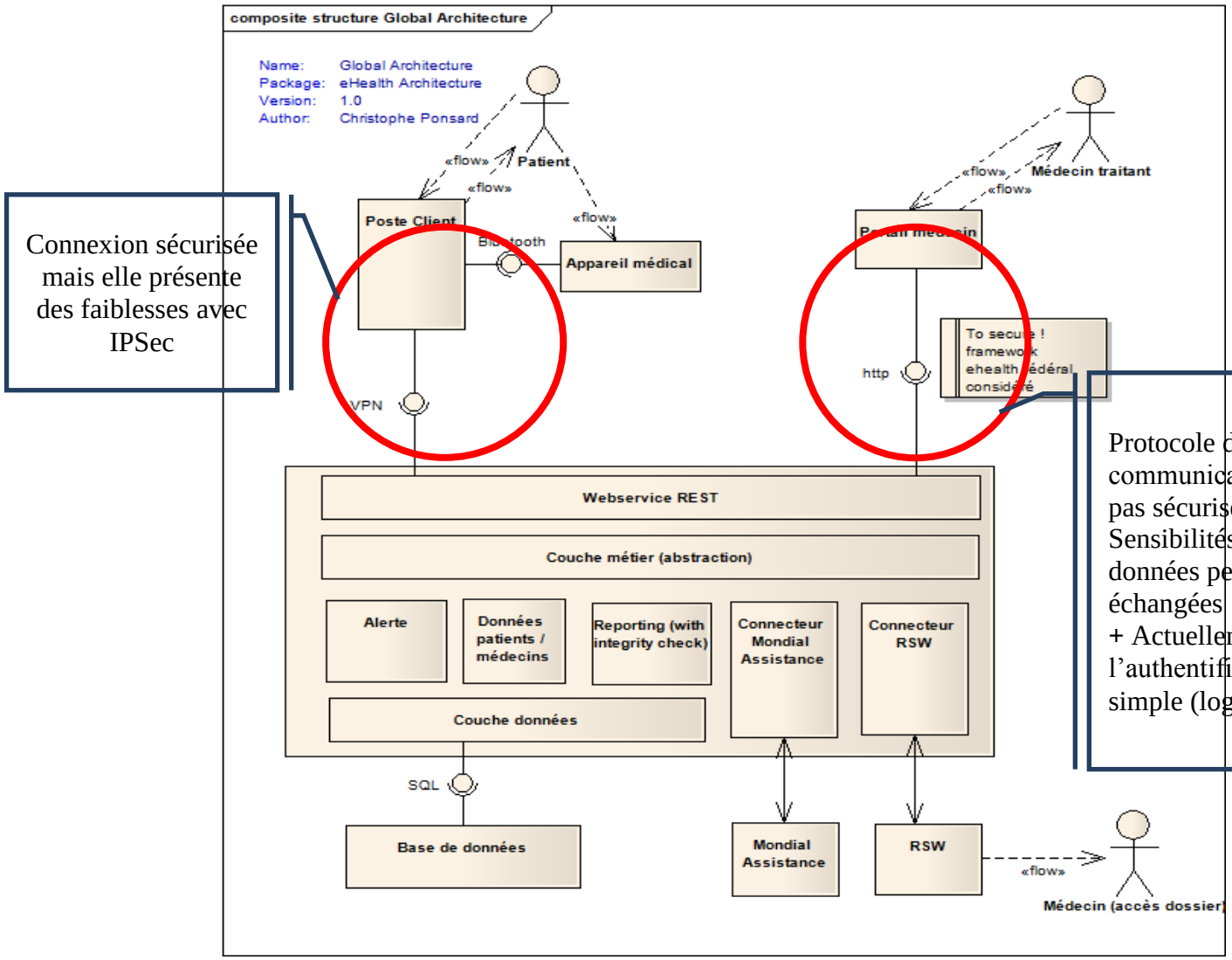


Figure 15 : Critique de L'existant _Architecture d'eHealth

❖ Limitations de système

Le système par ailleurs est conçu avec un souci particulier par rapport aux exigences non-fonctionnelles suivantes:

- ergonomie: le terminal sera simple d'utilisation et spécialement adapté aux personnes, notamment en évitant l'usage du clavier (terminal tactile), prévoyant des caractères de grande tailles, interface explicite et intuitive...
- sécurité: assurer l'authentification des acteurs (patients, médecin), l'intégrité des données du patient, ainsi que la confidentialité des données (en particulier par rapport au respect de la législation en matière de vie privée).

L'exigence de sécurité n'est actuellement que partiellement réalisée et doit être consolidée dans la perspective d'une exploitation.

Dans notre projet on se limitera à la sécurisation des interactions entre le médecin traitant et la plateforme.

III.4 Solution proposée

Développement d'un prototype qui gère les fonctions d'authentification des utilisateurs (intégration de l'eID) et de signature électronique au sein de notre plateforme de suivi médical à distance .Dans ce mémoire on se limitera à la partie droite de la figure ci-dessous, celle du médecin traitant.

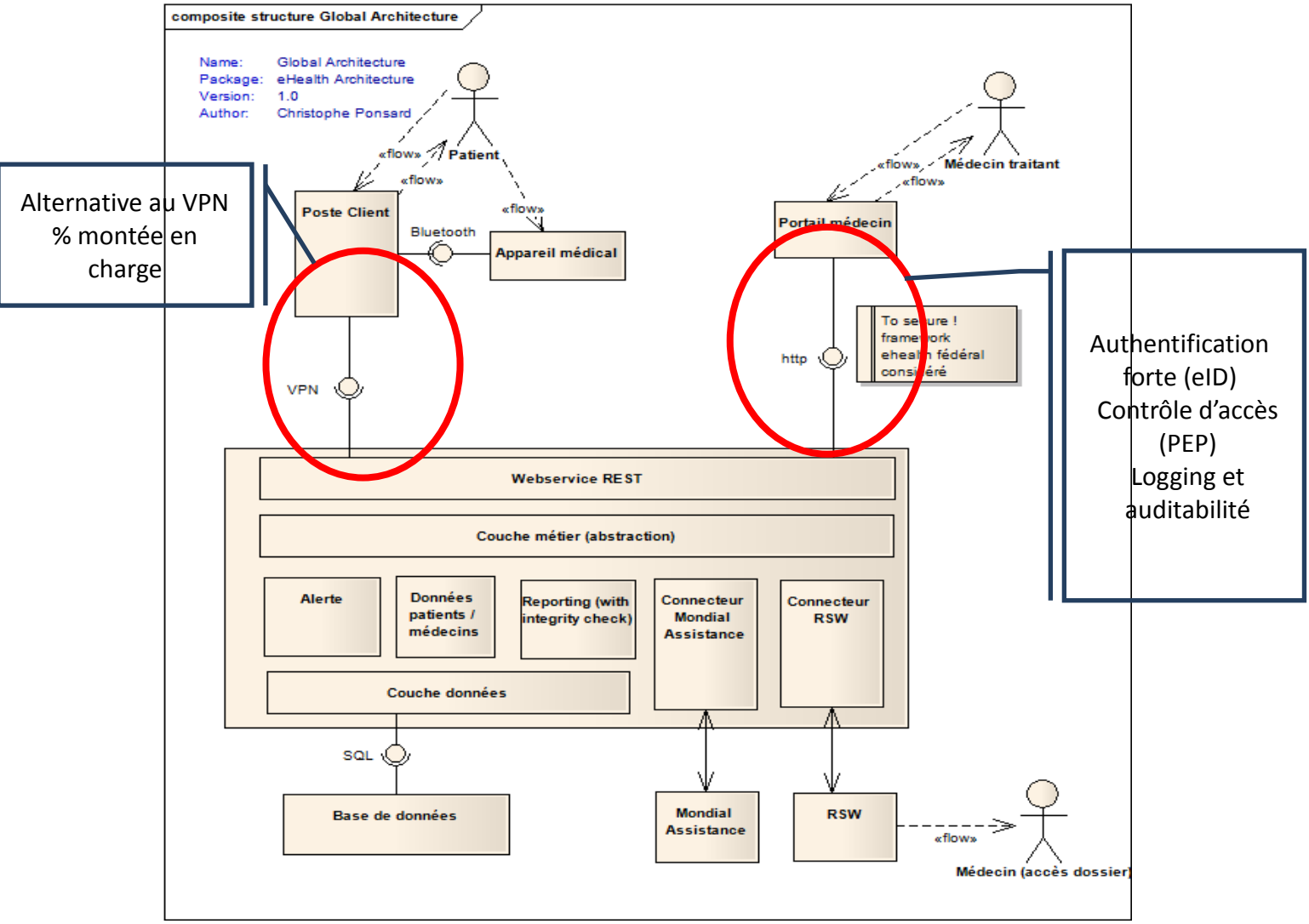


Figure 16: Solution proposée _Architecture d'eHealth

III.5 Gestion du projet

III.5.1 Choix de la méthodologie de développement

Le choix du modèle de cycle de vie, du langage ainsi que de l'outil de modélisation est initialement indispensable. Pour cela nous survolons la méthode Agile qui nous permet de bien mener notre projet tout en organisant notre plan de travail. Le choix de la méthode Agile était basé sur cinq facteurs principaux qui sont pris en considération pour valider la possibilité d'application de cette méthode.

Ces facteurs sont les suivants :

- Besoin rapide de mise à disposition du produit.
- Imprévisibilité des besoins du client.
- Nécessité de changements fréquents.
- Besoin de visibilité du client sur l'avancement des développements.
- Présence de l'utilisateur assurant un feedback immédiat.

❖ **Méthodologie AGILE**

Cette méthodologie favorise l'interaction du client et permet une grande réactivité à ses demandes. Elle vise la satisfaction réelle du besoin du client et non les termes d'un contrat de développement. Une méthode Agile est donc avant tout itérative sur la base d'un affinement du besoin mis en œuvre dans des fonctionnalités en cours de réalisation et même déjà réalisées. Cet affinement, indispensable à la mise en œuvre du concept adaptatif, se réalise en matière de génie logiciel sous deux aspects :

- **fonctionnellement,** par adaptation systématique du produit aux changements du besoin détecté par l'utilisateur lors de la conception-réalisation du produit (notion de validation permanente de l'utilisateur avec RAD[7] et notion de conception émergente avec XP [8])
- **techniquement,** par remaniement régulier du code déjà produit.

Une méthode Agile est aussi incrémentale, la production des fonctionnalités d'un projet s'effectue en plusieurs incréments.Les valeurs fondamentales de la méthode Agile sont :

- **L'équipe** : les individus et leurs interactions plus que les processus et les outils.
- **L'application** : des logiciels opérationnels plus qu'une documentation exhaustive.

[7] RAD : Rapid Application Development
[8] XP : Extreme Programming

- **La collaboration**: la collaboration avec les clients plus que la négociation contractuelle.
- **L'acceptation du changement** : l'adaptation au changement plus que le suivi d'un plan.

❖ Méthode Scrum

Scrum est une méthode Agile dédiée à la gestion de projets de développement de produits. Elle s'appuie sur le découpage d'un projet en incréments, nommés "**sprint**", ainsi que l'auto-organisation de l'équipe de développement

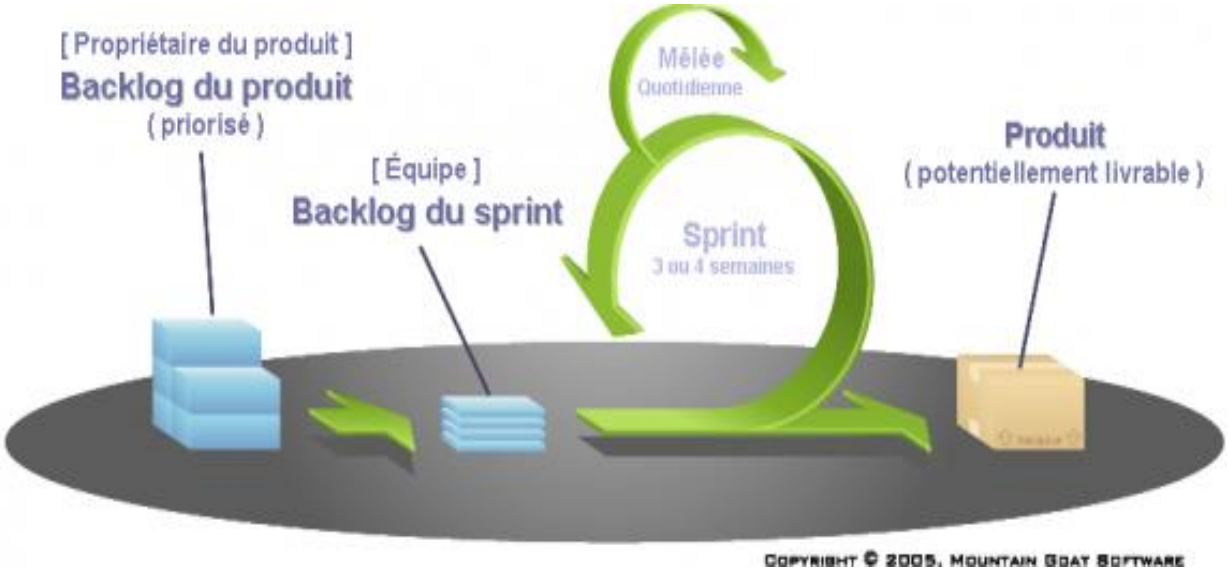


Figure 17 : Etapes de la méthodologie Scrum

Les sprints peuvent durer entre quelques heures et un mois (avec une préférence pour deux semaines). Chaque sprint possède un but et on lui associe une liste de fonctionnalités à réaliser. Aussi chaque sprint commence par une estimation suivie d'une planification opérationnelle, se termine par une démonstration de ce qui a été achevé et contribue à augmenter la valeur d'affaire du produit comme le montre la Figure 17)

Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective : elle analyse ce qui s'est passé durant ce sprint, afin de s'améliorer pour le prochain. Scrum est issu des méthodes itératives, incrémentales (telles que le modèle en spirale) et adaptatives (telles que RAD). La version 2011 de Scrum implémente l'aspect itératif: les modifications des fonctionnalités durant le sprint y sont autorisées.

❖ **Backlog de sprint**

Lorsqu'on démarre un sprint, on choisit quels items du **backlog** de produit seront réalisés dans ce sprint. L'équipe décompose ensuite chaque item en liste de tâches élémentaires.

Chaque tâche étant estimée en heures et ne devant pas durer plus que deux jours. On constitue ainsi le backlog de sprint. Pendant le déroulement du sprint, chaque équipier s'affecte des tâches du backlog de sprint et les réalise. Il met à jour régulièrement dans le backlog du sprint le reste à réaliser de chaque tâche.

Les tâches ne sont pas réparties initialement entre tous les équipiers, elles sont prises au fur et à mesure que les précédentes sont terminées.
La somme des heures des items du backlog de sprint constitue le reste total à faire du sprint. Cela permet de produire un sprint **burn down chart** (graphiques d'avancement permettant de visualiser graphiquement l'avancement du travail. Une interprétation simple permet d'avoir une idée sur les échéances futures) qui montre les heures restantes à réaliser au fur et à mesure du sprint (11).

III.5.2 Diagramme de Gantt

La planification structurelle de notre projet se base essentiellement sur la méthode Scrum dans laquelle chaque Sprint est de 15 jours à 1 mois au maximum

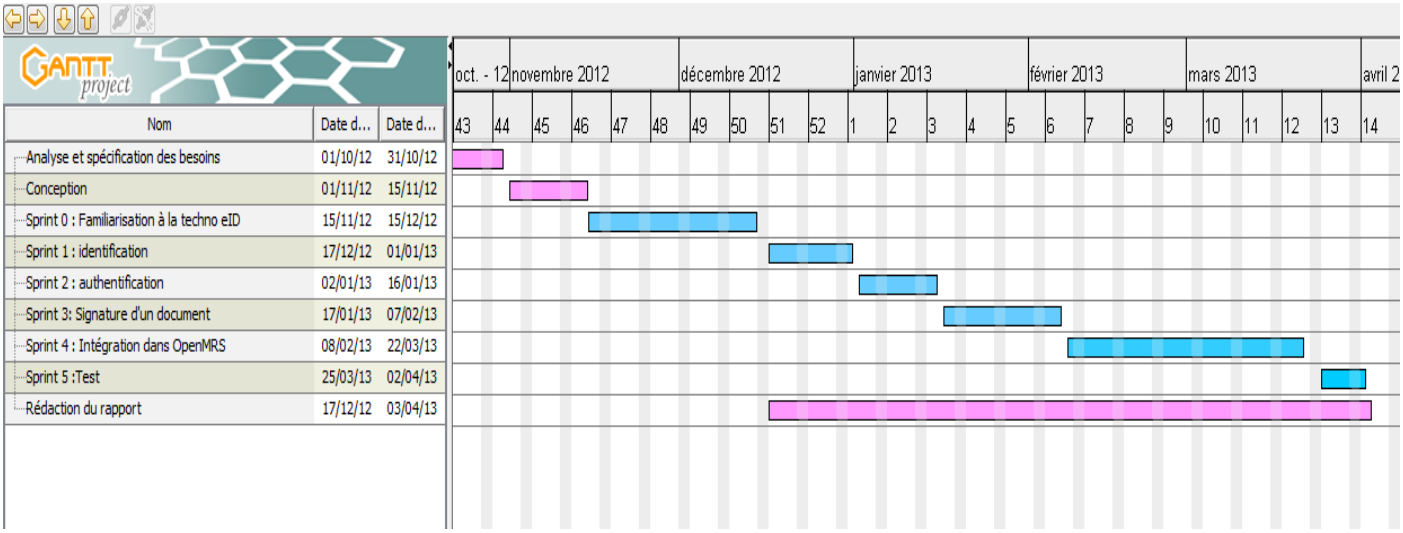


Figure 18: Diagramme de Gantt du projet

III.6 Exigences non fonctionnelles relatives à la sécurisation de la plateforme

❖ **NF1 - Authentification forte :**

La plateforme devra mettre en place un mécanisme d'authentification forte du médecin traitant pour se connecter à la plateforme.

❖ **NF2 - Contrôle d'accès :**

Un médecin traitant authentifié a accès uniquement aux données des patients qui lui sont affectés sauf en cas identifié comme urgence.

❖ **NF3 - Prise en charge :**

Un patient est toujours affecté à un médecin actif sur le système.

❖ **NF4 - Audit trace :**

Les accès de tout utilisateur du système aux dossiers médicaux sont tracés.

❖ **NF5 - Criticité:**

Les fonctions de suivi n'ont pas un caractère critique: le système n'a pas la finalité de surveiller des patients très instables.

❖ **NF6 - Signature digitale :**

Les rapports exportés vers le réseau santé wallon sont signés par la plateforme.

❖ **NF7 - Vie privée :**

La plateforme respectera les contraintes de vie privée locale (informations conservées, délais de conservation, droit de modification, etc.)

- **NF 7.1 -** les utilisateurs ont le contrôle de leurs données (ils peuvent les lire, les corriger, annuler leur inscription)
- **NF 7.2 -** le numéro RN ne peut pas être stocké comme identifiant primaire au niveau de la base de données de l'application

III.7 Scénarios détaillés

Étant donné que la méthodologie suivie au cours de notre processus de travail est Agile Scrum, cette partie sera organisée comme suit :
Nous commencerons par présenter le use case général de notre prototype. Ensuite nous présenterons chaque sprint, en spécifiant le diagramme de cas d'utilisation détaillé.

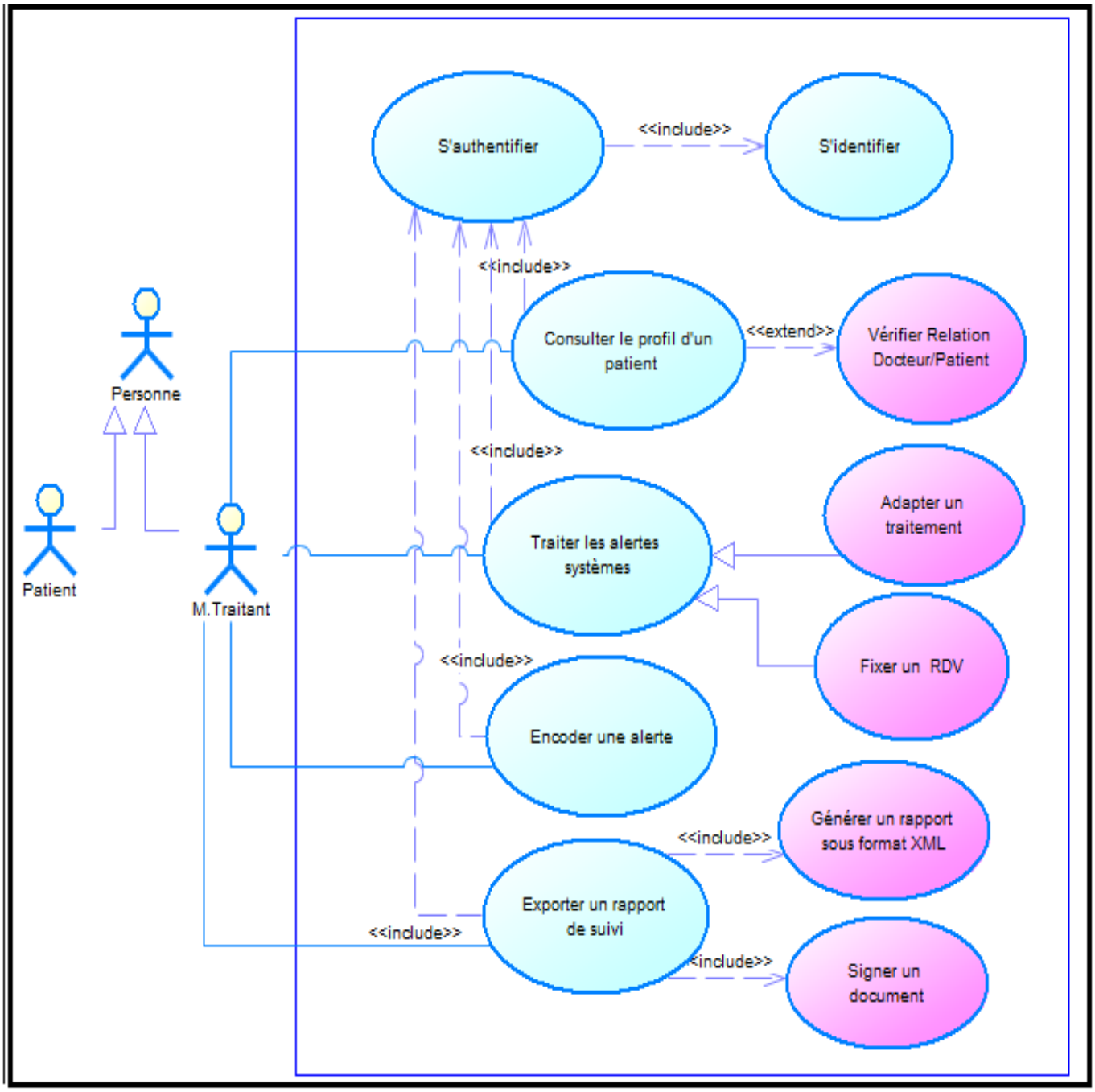


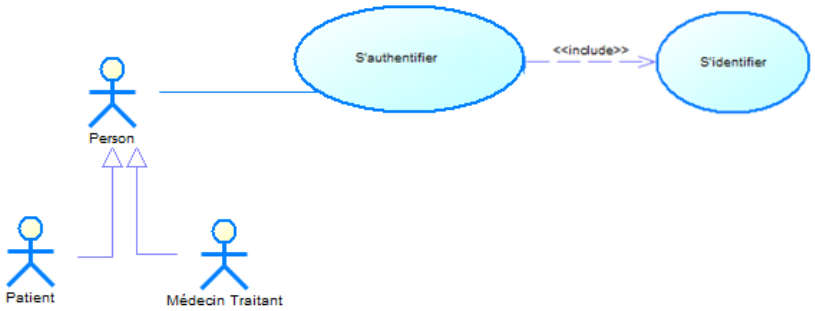
Figure 19:Diagramme cas d'utilisation -Médecin traitant

Pour mieux visualiser ces besoins, on a eu recours au diagramme cas d'utilisation d'UML. Le médecin traitant peut interagir avec le système après authentification.

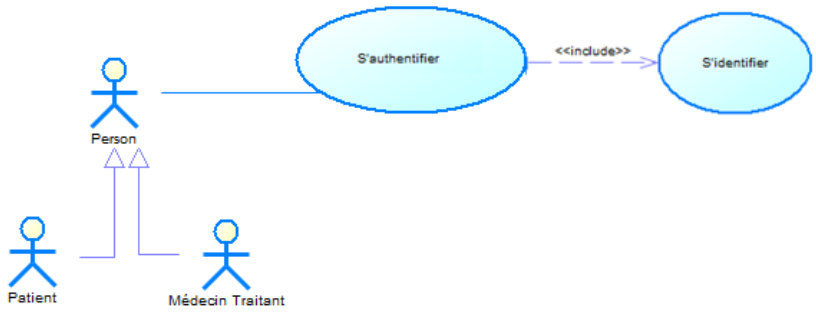
III.7.1 Sprint1

Généralement le premier sprint comprend toujours les besoins du client de criticité élevée, dans notre cas c'est la phase d'authentification dans laquelle en passe d'une authentification simple par login et mot de passe à une authentification forte par certificat.

❖ **UC1 - Connexion au système**

SOMMAIRE D'IDENTIFICATION	
Description	le médecin traitant s'authentifie via son token hardware (ex. carte eID, carte ANCE)
DESCRIPTION DES ENCHAINEMENTS	
Pré-condition	le médecin est connu de la plateforme, dispose d'un token valide et s'authentifie correctement
Post-condition	le médecin traitant est loggé sur le système. Il a accès : ○ à son profil ○ à la liste de ses patients ○ à une liste d'alerte relative à ses patients (on peut considérer des cas plus complexes s'il y a des exigences de réactivité)
Exception	rejet d'authentification - le médecin dispose de trois tentatives, ensuite son accès est bloqué. L'IP source est aussi loggée.
Diagramme cas d'utilisation	

❖ **UC2-Déconnexion du système**

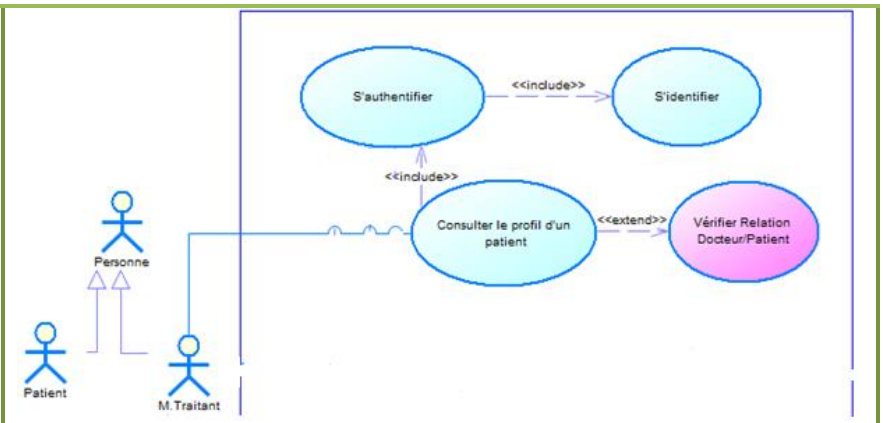
SOMMAIRE D'IDENTIFICATION	
Description	le médecin se déconnecte
DESCRIPTION DES ENCHAINEMENTS	
Pré-condition	le médecin est connecté
Exception	en cas de problème, le médecin est déconnecté et le problème est loggé
Diagrammes cas d'utilisation	

III.7.2 Sprint2

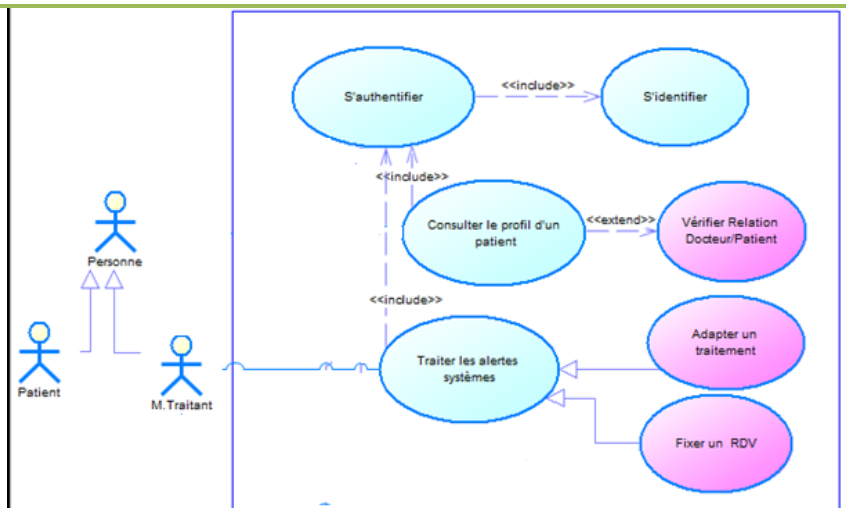
Dans le second sprint nous traitons l'aspect de contrôle d'accès ou le système vérifie la présence d'une relation de prise en charge côté médecin traitant pour autoriser l'accès aux dossiers médicaux des patients et leurs manipulations

❖ **UC3- Accès à un patient**

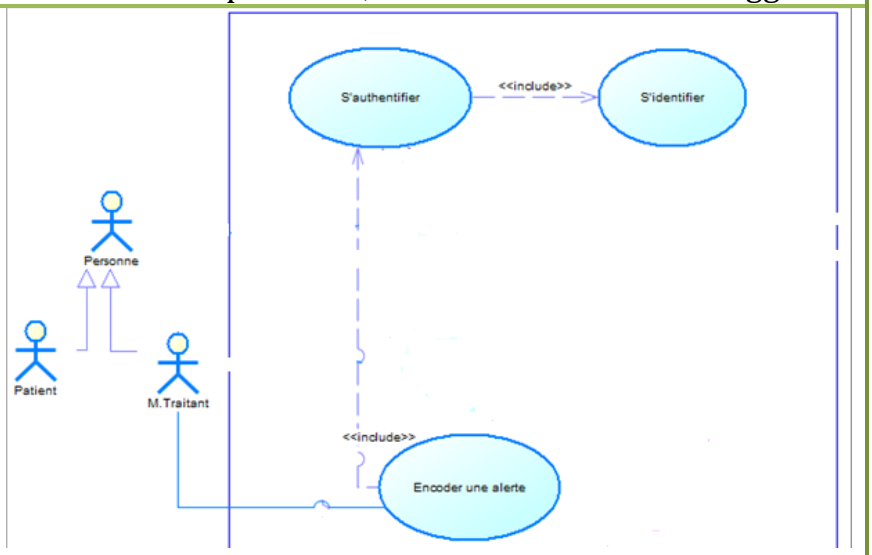
SOMMAIRE D'IDENTIFICATION	
Description	le médecin tente d'accéder à un dossier patient
DESCRIPTION DES ENCHAINEMENTS	
Pré-condition	le médecin est affecté au patient (relation docteur/patient)
Post-condition	le médecin a accès aux données du patient, l'accès est loggé
Exception	le médecin n'a pas accès, la tentative d'accès est loggée

Diagramme cas d'utilisation	

❖ **UC4- Traitement d'une alarme générée**

SOMMAIRE D'IDENTIFICATION	
Description	le médecin consulte une alarme générée par le système
DESCRIPTION DES ENCHAINEMENTS	
Pré-condition	l'alarme est relative à un patient du médecin
Post-condition	une action est prise: rejet justifié de l'alarme, demande de suivi médical non urgent, demande d'intervention...
Exception	le médecin n'a pas accès, la tentative d'accès est loggée
Diagrammes cas d'utilisation	

❖ **UC5 - Encodage d'une alarme par le médecin traitant**

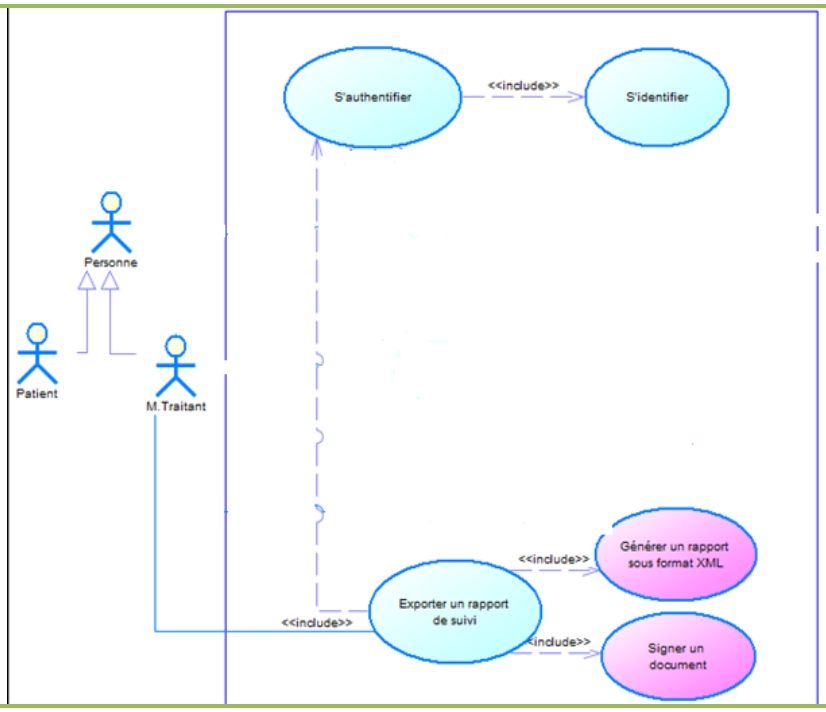
SOMMAIRE D'IDENTIFICATION	
Description	le médecin consulte le dossier et décide de lever une alarme lui-même
DESCRIPTION DES ENCHAINEMENTS	
Pré-condition	le médecin est affecté au patient
Post-condition	une action est prise: demande de suivi médical non urgent, demande d'intervention...
Exception	le médecin n'a pas accès, la tentative d'accès est loggée
Diagrammes cas d'utilisation	

III.7.3 Sprint3

Le troisième sprint traite l'aspect d'altération des données électroniques par la mise en place de la signature des documents par un certificat

- **UC6 - Export RSW**

SOMMAIRE D'IDENTIFICATION	
Description	génération d'un fichier sommaire du patient pour le RSW à une date donnée
DESCRIPTION DES ENCHAINEMENTS	

Pré-condition	utilisateur RSW authentifié
Post-condition	un export RSW (format XML) est généré. Le fichier est horodaté et signé par la plateforme.
Exception	rejet d'authentification - l'utilisateur RSW dispose de trois tentatives, ensuite son accès est bloqué. L'IP source est aussi loggée.
Diagrammes cas d'utilisation	

III.8 Conclusion

Dans ce chapitre, nous avons présenté, dans une première section, le système actuel. Puis, nous avons creusé ses limitations notamment en matière de sécurité. Ensuite, nous avons proposé une solution à ses dernières. La fin de cette section a été consacrée à la présentation de la méthodologie de travail.

Dans une seconde section, nous avons entamé la spécification des besoins qui nous a permis d'avoir une vision plus claire du sujet et une compréhension plus profonde des tâches à réaliser. Elle mène également à prévoir les problèmes à rencontrer et chercher les solutions permettant de les contourner. La conception fera l'objet du prochain chapitre dans lequel nous utiliser ons UML comme langage de modélisation.

CHAPITRE

IV. Conception

IV. Conception

IV.1 Introduction

Ayant achevé la phase d'analyse et de spécification, nous entamons dans ce chapitre la phase de la conception en se basant sur le travail déjà accompli. Cette phase de conception occupe une place très importante dans la réalisation des projets informatiques. La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

Dans cette partie, nous détaillons la conception de la base de données et de l'application afin d'identifier les principaux éléments et les différents fonctionnalités de système, essentiellement du point de vue de la sécurité. Nous présentons tout d'abord l'architecture générale du système. Dans une deuxième étape, nous procédons à la modélisation des diagrammes de séquence détaillant la sécurisation de poste du médecin traitant. Nous analyserons enfin la structure de données d'OpenMRS pour identifier les informations qui jouent un rôle au niveau de la sécurisation de la plateforme.

IV.2 Architecture de l'application

Une architecture client/serveur s'impose pour mettre en place notre système. La sécurité est de plus en plus considérée comme un service à part entier, c'est en particulier l'approche retenue par l'eID qui est appliquée ici afin de favoriser un environnement de confiance entre une machine cliente et un serveur distant.

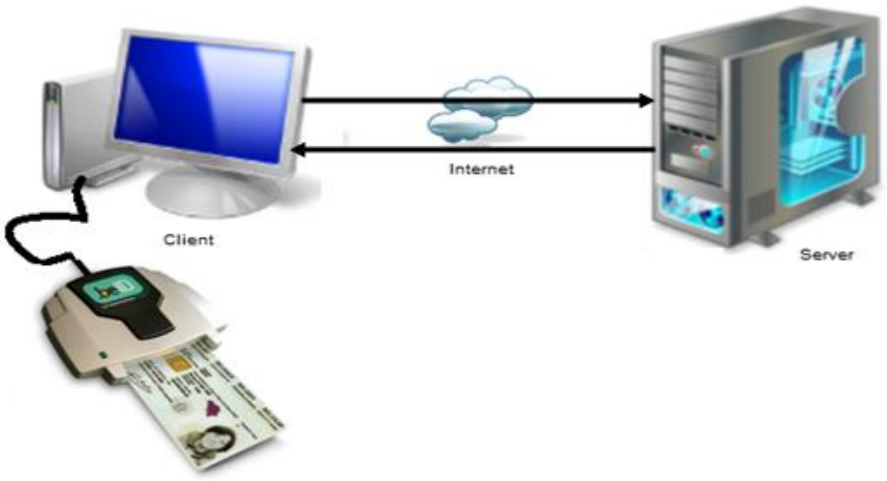


Figure 20 : Paradigme client-serveur

Détaillons un peu notre architecture

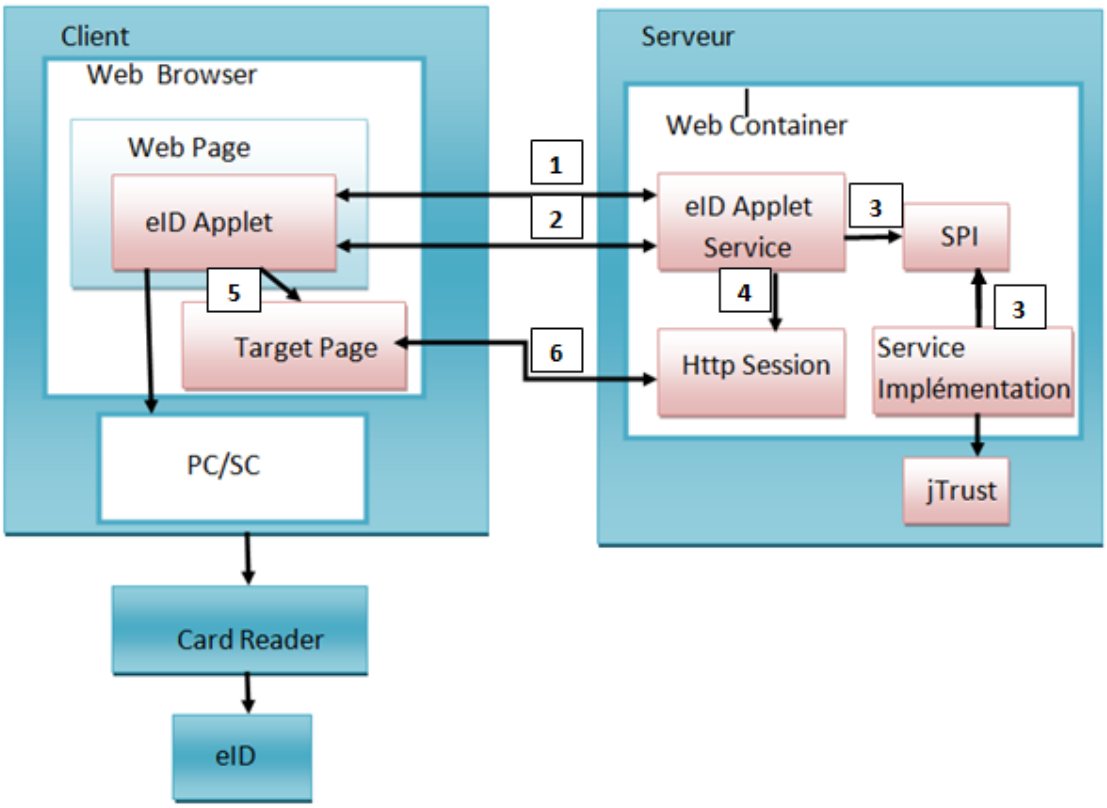


Figure 21: Architecture détaillée de l'application

1: le client envoi une requête au serveur pour bénéficier d'un service distant.
2:une applet qui prend comme paramètre le nom de service ainsi que la page résultat se lance dans le navigateur
3: dans l'eID applet service on trouve les prototypes de services (c'est un annuaire de services) disponibles sur le serveur, on utilise le design pattern SPI pour accéder à l'implémentation des services (OpenMRS dans notre cas)
4: ouvrir une nouvelle session
5/6: une réorientation vers la page web résultat et affichage de résultat d'exécution du service (par exemple la page d'un patient qu'on est autorisé à voir une fois authentifié)

IV.3 Conception détaillée de la sécurisation du poste du médecin traitant

Pour élaborer cette section, nous allons adopter l'UML comme une méthode de conception puisqu'il s'agit d'une interaction entre des objets. Nous détaillons chaque phase par un schéma puis, un diagramme de séquence.

IV.3.1 Diagrammes de séquence

❖ Identification

Soit le schéma explicatif de la phase d'identification qui permet d'identifier le client connecté pour exécuter un processus et visualiser ses coordonnées.

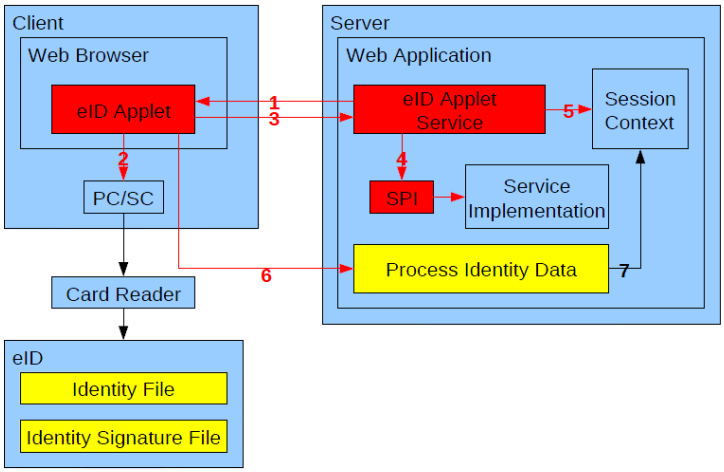


Figure 22: Architecture de l'eID applet identification

Ce diagramme décrit les différentes interactions entre le système et le client. Mais, il faut noter encore que ces différentes interactions ne constituent qu'un simple échange d'informations entre le système et le poste client afin d'identifier le citoyen par le serveur.

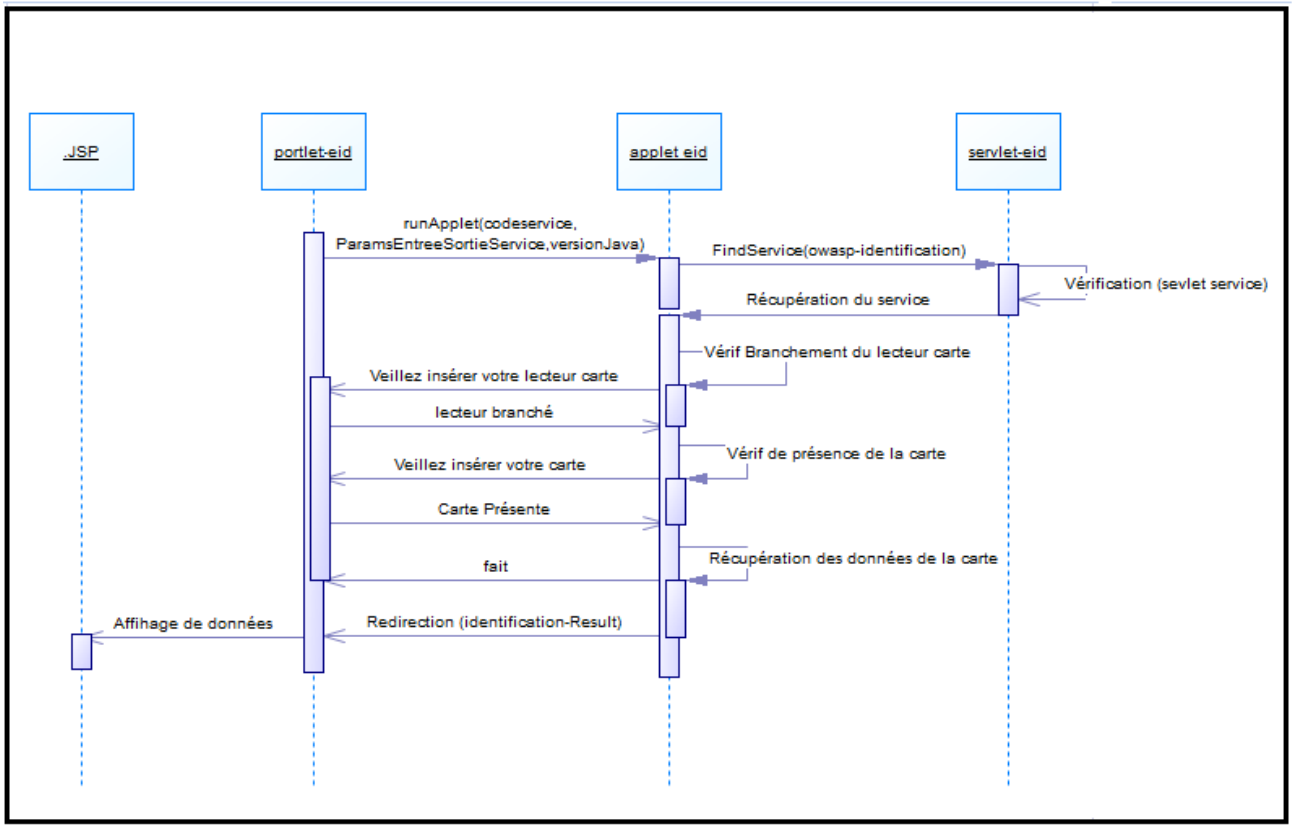


Figure 23 : Diagramme de séquence Identification

❖ Authentification

Soit le schéma explicatif de la phase d'authentification par certificat suivant:

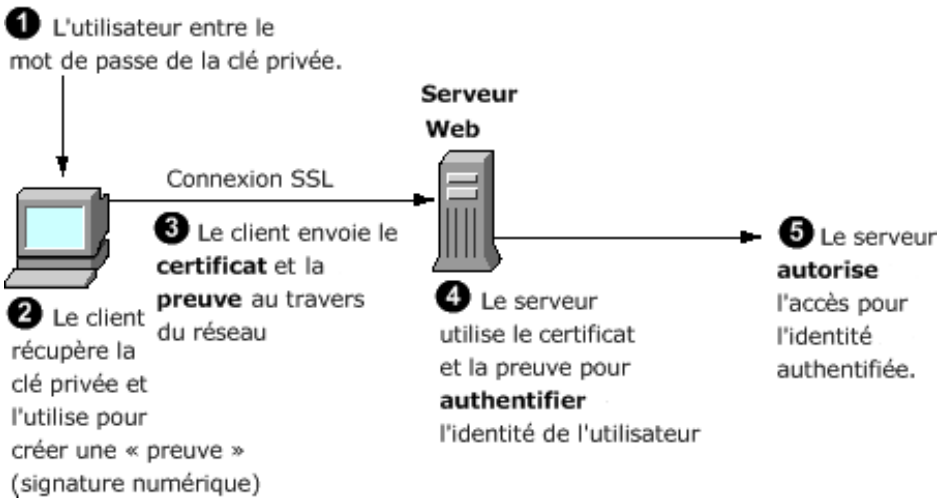


Figure 24 : Authentification par certificat

Le client possède un certificat valide qui peut être utilisé pour l'identifier auprès du serveur. L'authentification par certificat est généralement considérée comme préférable à l'authentification par mot de passe car elle est basée sur ce que l'utilisateur a (la clef privée) aussi bien que sur ce que l'utilisateur sait (le mot de passe qui protège cette clef privée).

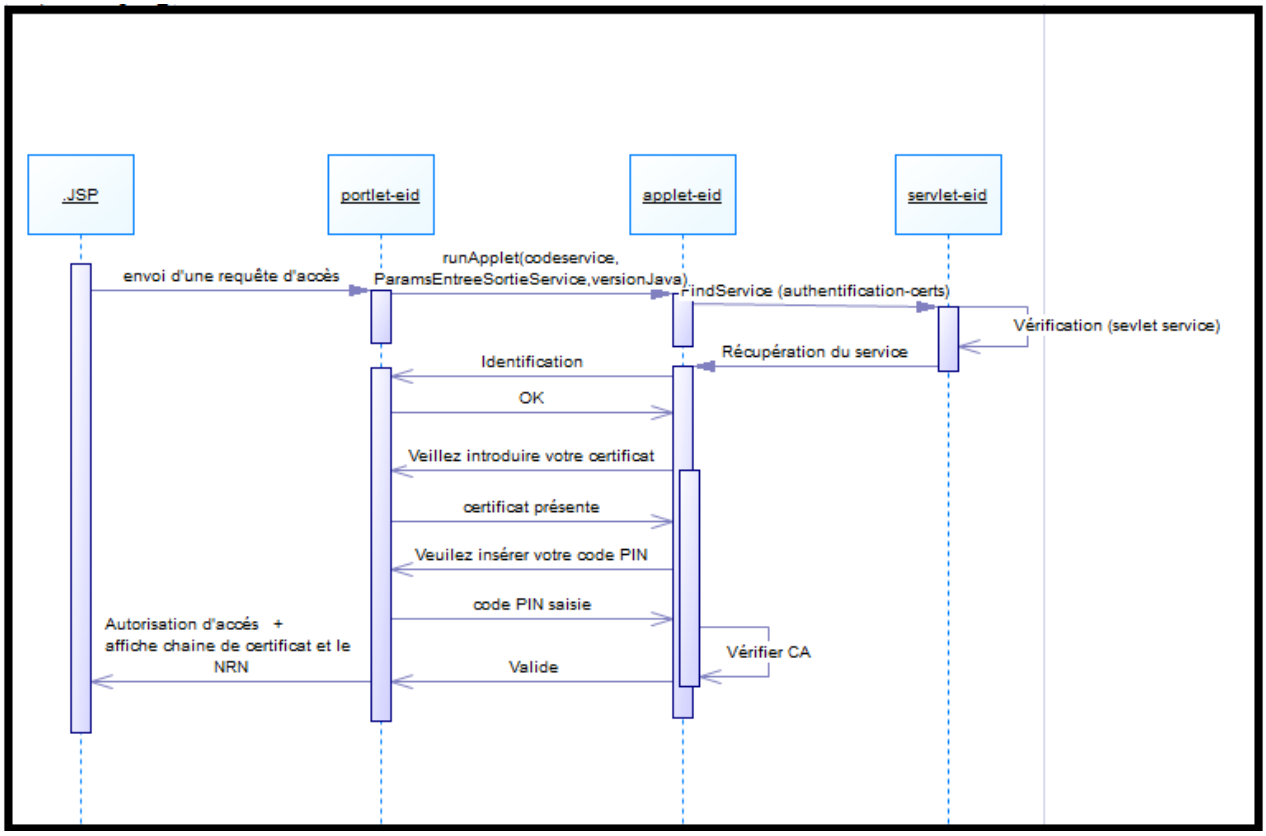


Figure 25 : Diagramme de séquence authentification par certificat

❖ Signature

Un simple échange d'informations qui n'impose comme moyen de sécurité que le privilège d'accès grâce à un nom d'utilisateur et un mot de passe ;alors que nous avons montré dans la partie de l'étude de l'art que ce moyen de sécurité ne suffit pas à cause de l'existence des logiciels d'espionnage qui ont permis à plusieurs pirates d'espionner les flux de données transmis vers le serveur web.

Comment pouvons-nous, alors, ajouter l'aspect sécurité à ces interactions et aux flux transmis ?

D'après l'analyse que nous avons effectuée lors de la phase d'étude préalable, nous avons conclu que la manière la plus efficace pour protéger notre flux de données est le cryptage de celles-ci.

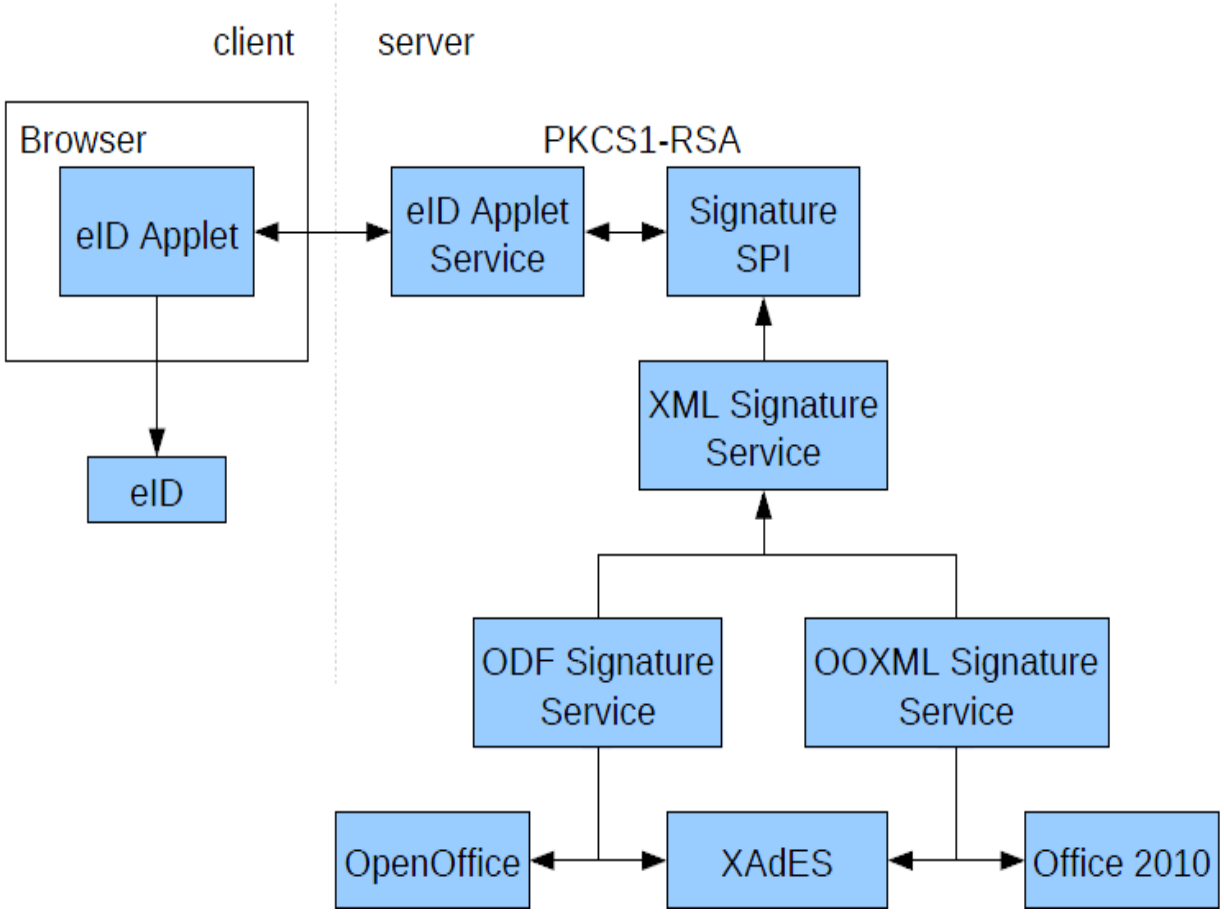


Figure 26: Architecture de l'eID applet signature

Ce diagramme montre l'interaction entre les objets lors de l'exécution de la signature

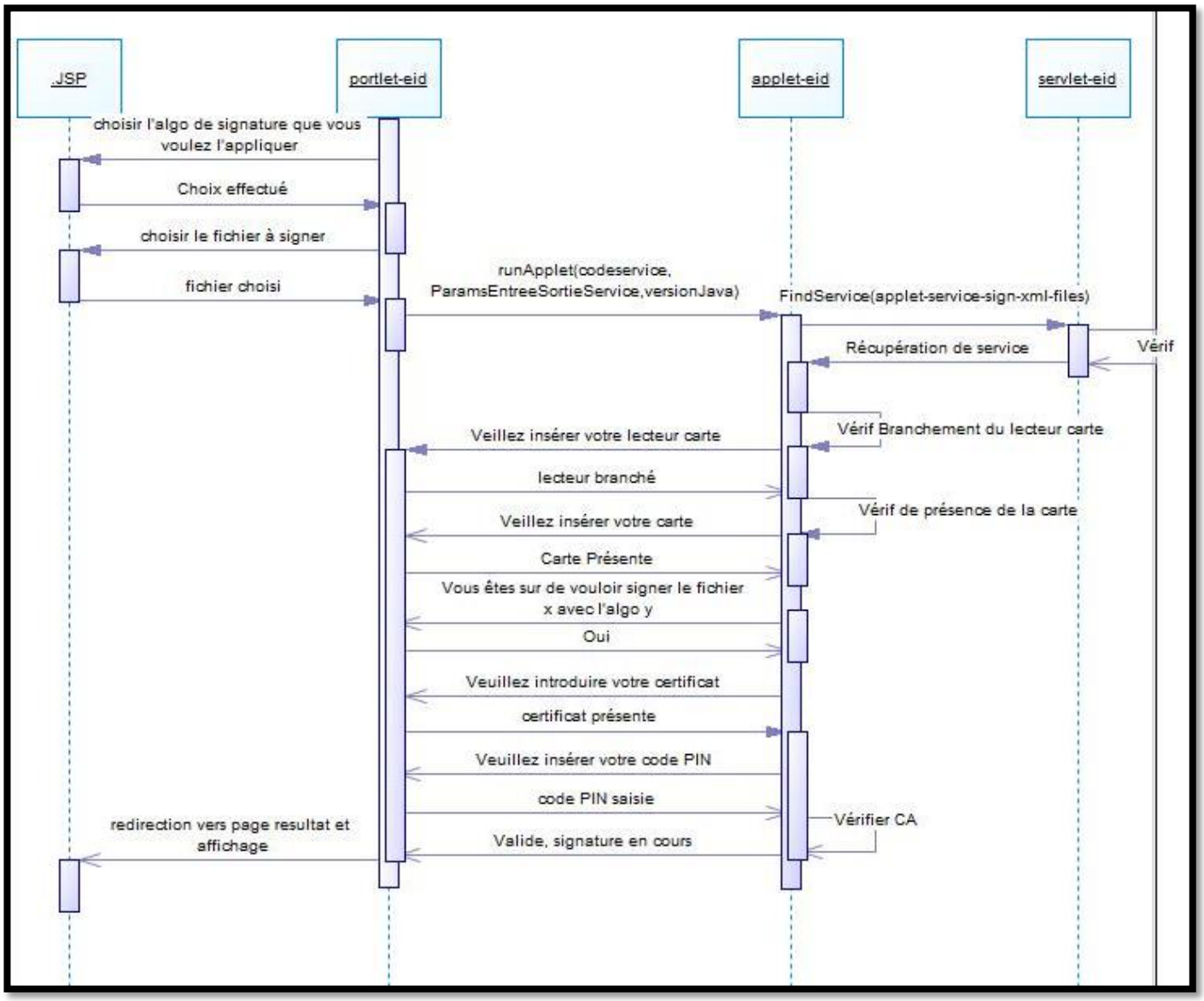


Figure 27: Diagramme de séquence signature de document

IV.4 Analyse structurelle de la base de données

Une étude approfondie de système OpenMRS et l'exploration de divers fonctionnalités qu'il supporte ont permis la classification de ces dernières en 2 catégories : souhaitée non supportée et souhaitée supportée (un tableau récapitulatif est présent dans l'annexe).OpenMRS est fortement complémentaire à la plateforme eHealth et a l'avantage d'être Open Source. Nous avons donc réalisé une analyse comparée de leurs modèles respectifs afin de permettre l'utilisation des deux solutions. Dans la suite du travail d'implémentation, nous nous concentrerons cependant sur OpenMRS.

IV.4.1 Modèle de base de données du système eHealth

Des raisons de confidentialité ont empêché le fait d'avoir accès à l'implémentation d'eHealth, dans ce cadre, le CETIC nous a fourni ce modèle de base de données comme étant référence sur le système.

Ce modèle est divisé en treize sous-modèles de base.

1. **Persons :** informations de base pour créer un profil d'une personne.
2. **Users :** sont les acteurs du système (docteur, patient, administrateur système, etc.) où les personnes possédant les mêmes permissions appartiennent à un groupe.
3. **Internationalisation :** ce sont des informations plus générales d'une personne.
4. **Devices :** une collecte d'informations sur les périphériques utilisés par une personne afin d'assurer le suivi médical à distance (appareil de mesure de glycémie).
5. **Measures :** selon le type de l'appareil, on liste l'ensemble de mesures qu'on peut aboutir (simple, complexe). Pour un type complexe on trouve une association de paramètres de mesure.
6. **Organisations :** un ensemble de services fournis par chaque organisation. Un contrat de service est effectué avec la personne souhaitant bénéficier de ces services.
7. **Prescriptions &Pathologies :** c'est un dictionnaire de données qui décrit les symptômes des pathologies et les traitements les plus courants pour chacun.
8. **Questionnaires :** la prescription impose des éléments-formes qui sont composés de sections dont chacune présente une question.
9. **Thresholds :** pour un paramètre de mesure on définit la valeur seuil et la condition (simple ou complexe) à vérifier.
10. **Constraints on persons :** un ensemble de privilèges et droits est affecté pour chaque type de contact entre les personnes.
11. **Logs tables :** journal des actions d'un utilisateur ou des évènements de devise.

12. **Reports :** la génération des rapports se fait selon le type de rapport, la période.
13. **External Communication :** la communication avec le RSW pour récupérer les dossiers médicaux des patients.

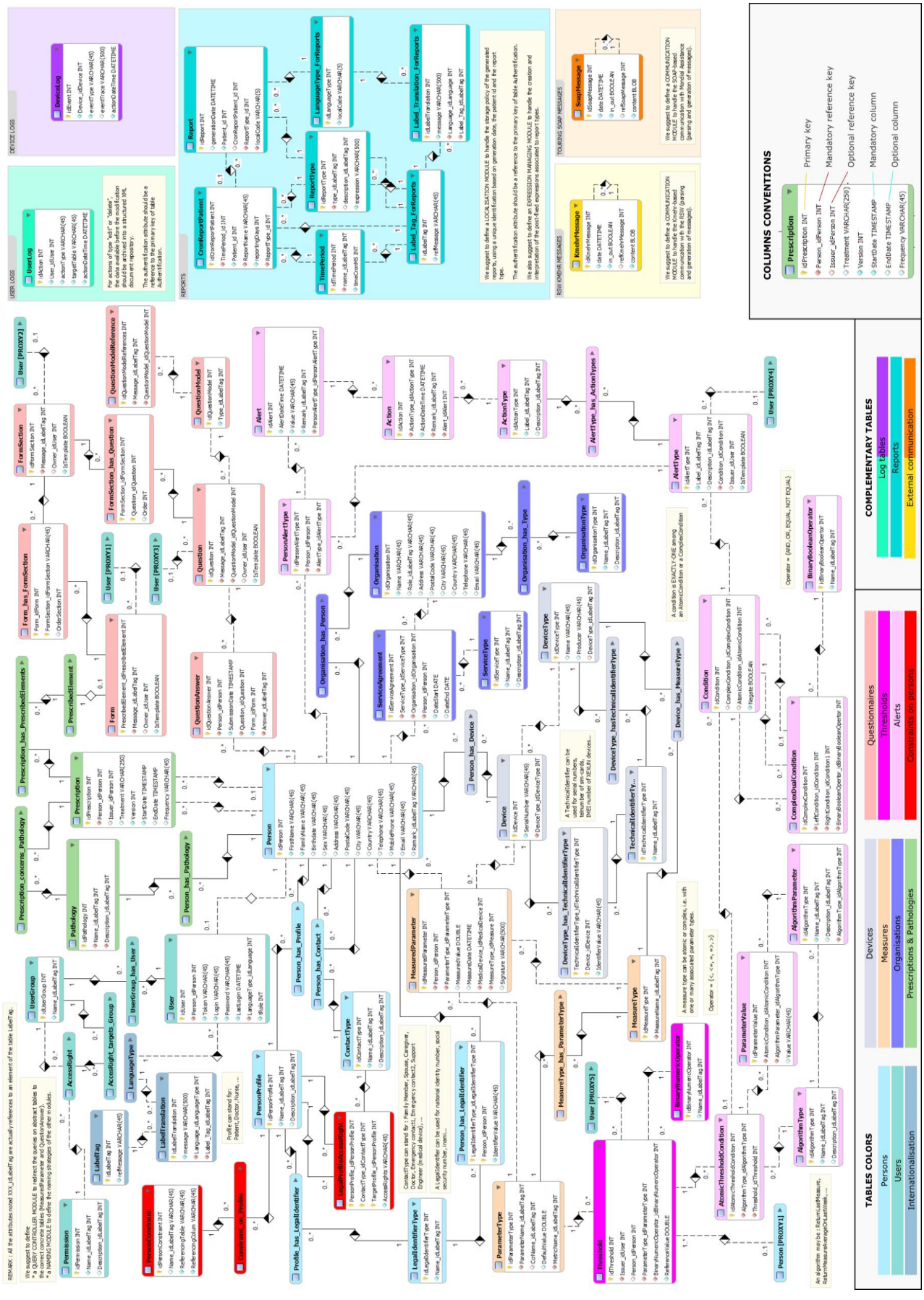


Figure 28:Modèle base de données de la plateforme eHealth (modèle de référence)

IV.4.2 Modèle de base de données de OpenMRS

La collaboration OpenMRS investit un effort continu dans l'élaboration du modèle de données OpenMRS. Plusieurs versions sont disponibles. La structure de ce modèle de données détermine l'évolutivité et la flexibilité d'un système. Le cœur de ce modèle de données porte sur le qui, quoi, quand, où, et le comment des rencontres médicales. Nous avons choisi la dernière version de ce modèle pour l'analyser ; la version 1.9.0

Ce modèle est divisé en dix sous-modèles de base. [16]

1. **Concept**: les concepts sont définis et utilisés pour étayer les données du système.
2. **Rencontre**: contient des métadonnées concernant les fournisseurs de soins de santé, des interventions avec un patient.
3. **Forme**: essentiellement, la description de l'interface utilisateur pour les différents composants.
4. **Observation**: c'est là que l'information sur la santé réelle des soins est stockée. Il existe de nombreuses observations par rencontre.
5. **Ordre** : choses / actions qui ont été invitées à se produire.
6. **Patient**: informations de base sur les patients dans ce système.
7. **Utilisateur**: informations de base sur les personnes qui utilisent ce système.
8. **Personne**: informations de base sur la personne dans le système.
9. **Entreprise**: données non médicales utilisées pour administrer OpenMRS.
10. **Groupes / Workflow**: flux de travail et les données de la cohorte.

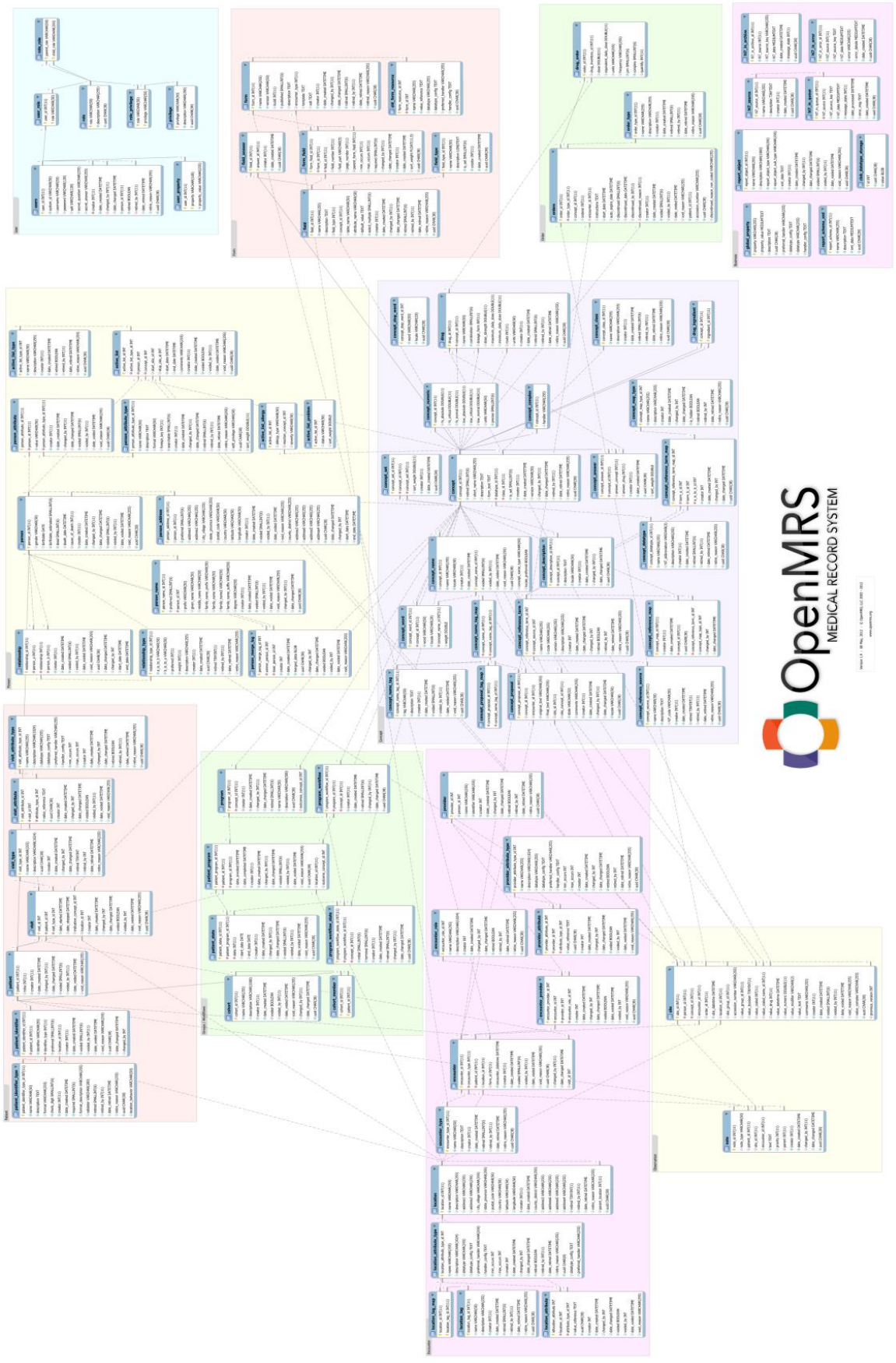
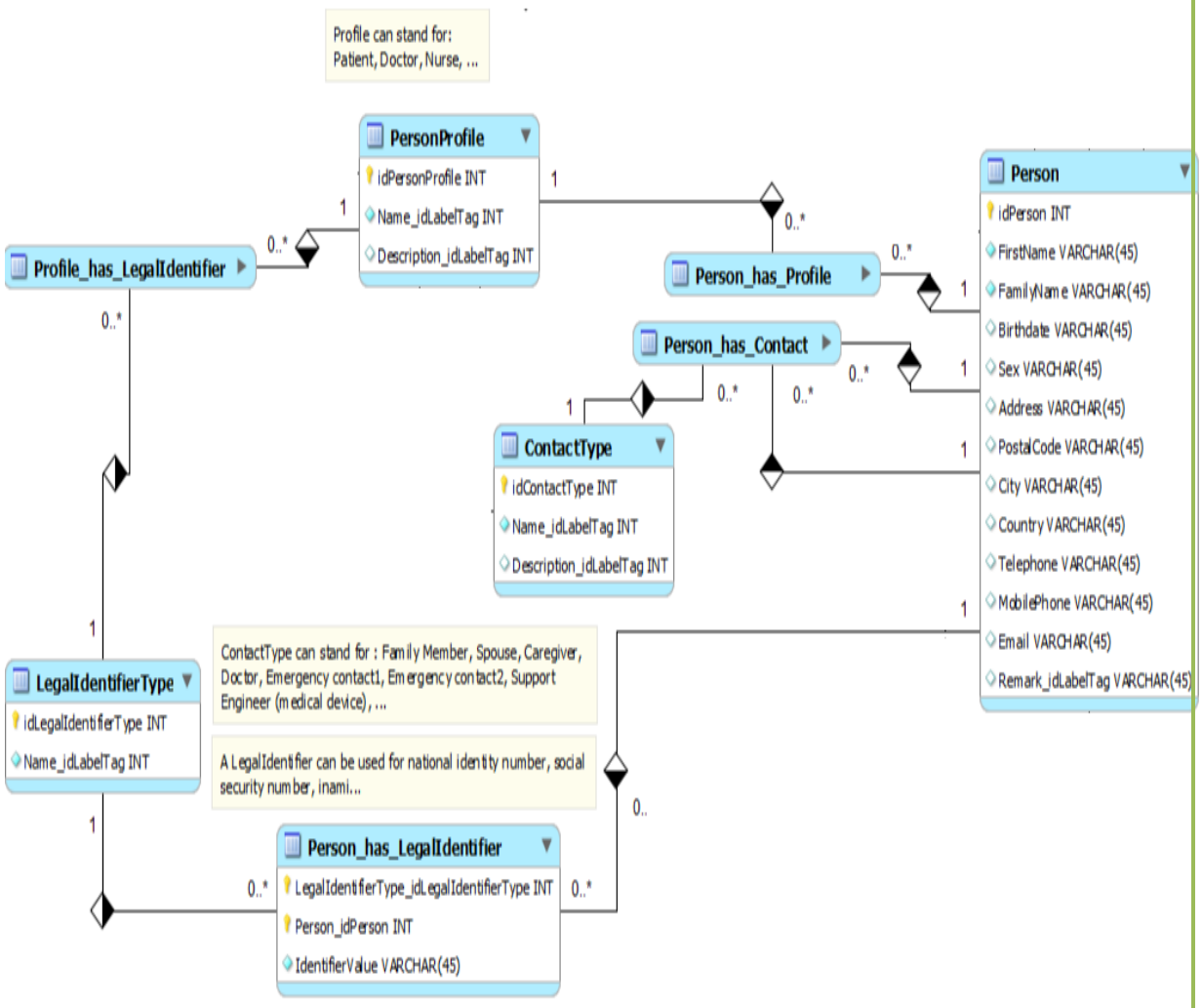


Figure 29: Modèle base de données de système OpenMRS

IV.4.3 Comparaison des modèles de base des données

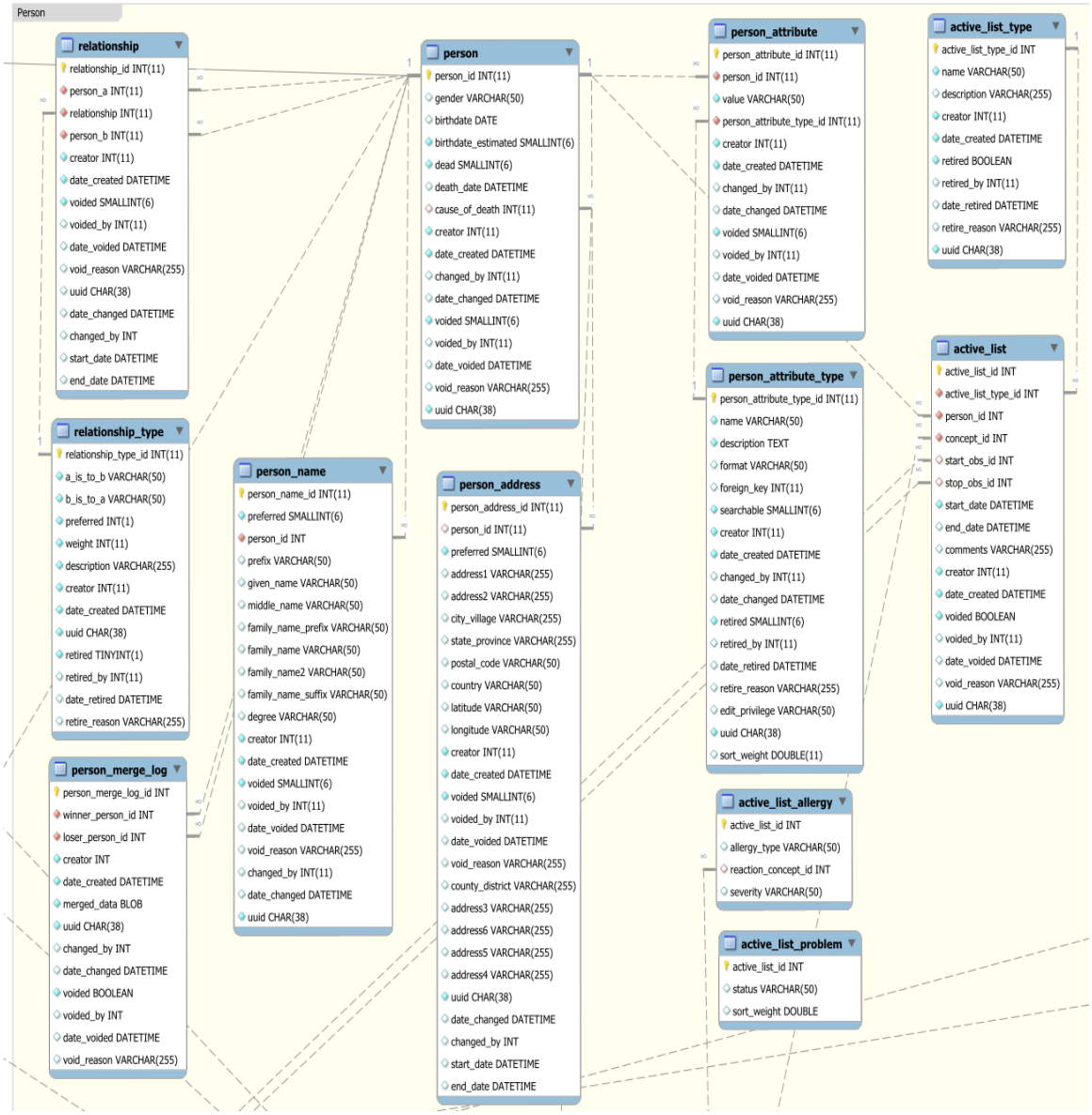
La comparaison des fonctionnalités souhaitées par le modèle de référence eHealth et supportées parfois partiellement par OpenMRS, a donné naissance au fait de creuser dans les modèles de BD pour sortir avec le ou les sous-modèles responsables à la mise en place de chaque fonctionnalité. Nous les détaillons ci-après avec une attention particulière au niveau de la sécurité.

Model_DB_e-Health(1) : Gestion des utilisateurs



Une Personne possède un profil utilisateur (Patient, docteur, etc.).Entre deux personnes, il peut exister une relation (Docteur, conjoint, etc.)

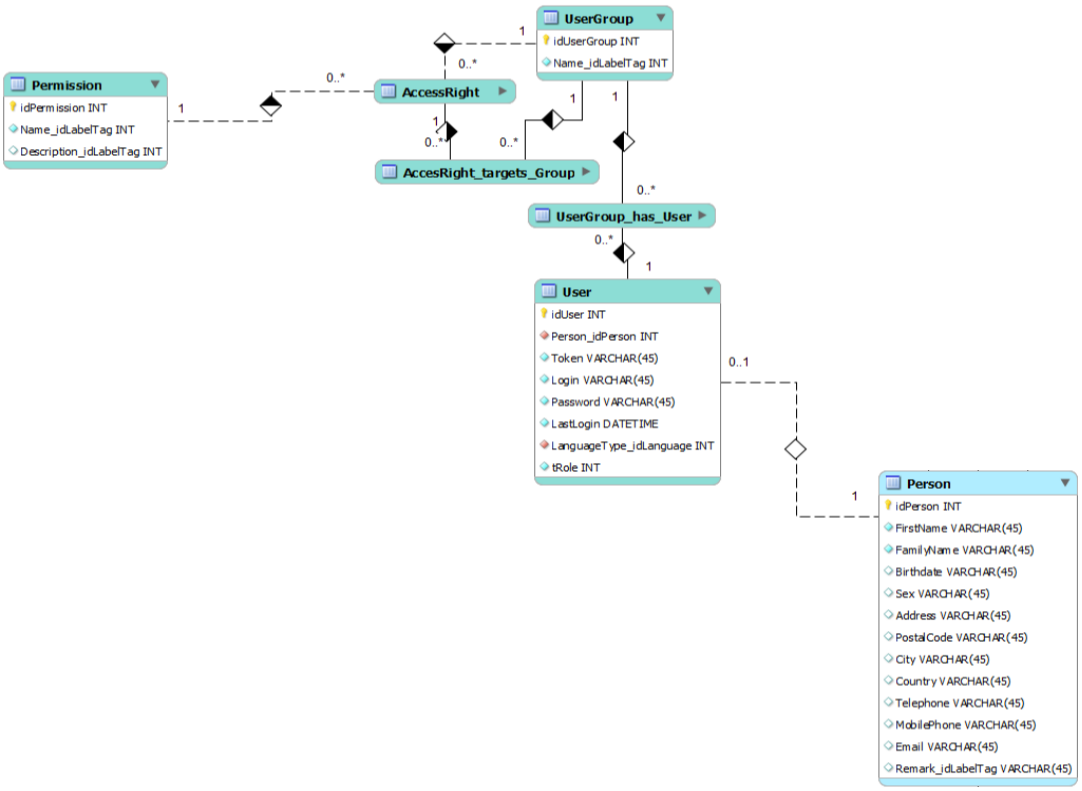
Model_DB_OpenMRS (2) : Gestion des utilisateurs



Synthèse

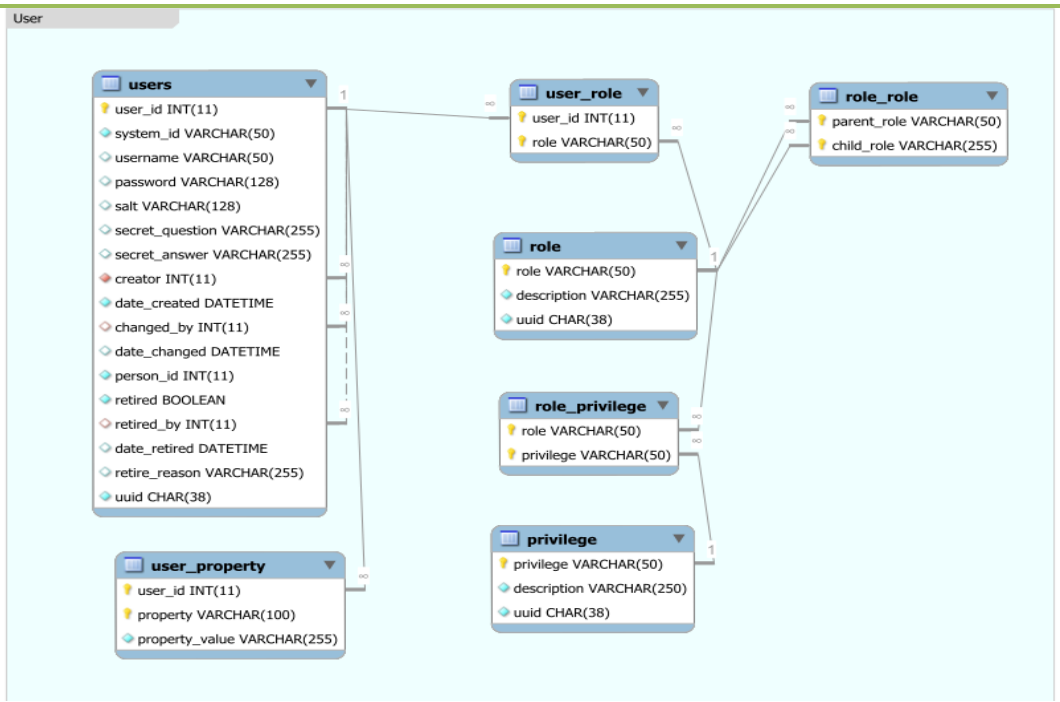
(1)	(2)
*les attributs sont statiques. *Contraintes dynamiques	*Une Personne peut avoir plusieurs PersonNames et Person Adresses *il y a certaines métadonnées générales qui définissent l'objet Personne (Date of Borth, Sexe, ...etc) *la plupart des attributs personne sont personnalisés
Table Contact_Type	Table Relationship
On peut faire l'analogie entre les 2 systèmes pour cette fonctionnalité	

Model_DB_e-Health(1) : Contrôles d'accès



À chaque groupe d'utilisateur est affecté un ensemble de permissions (privilèges)

Model_DB_OpenMRS (2) : Contrôles d'accès

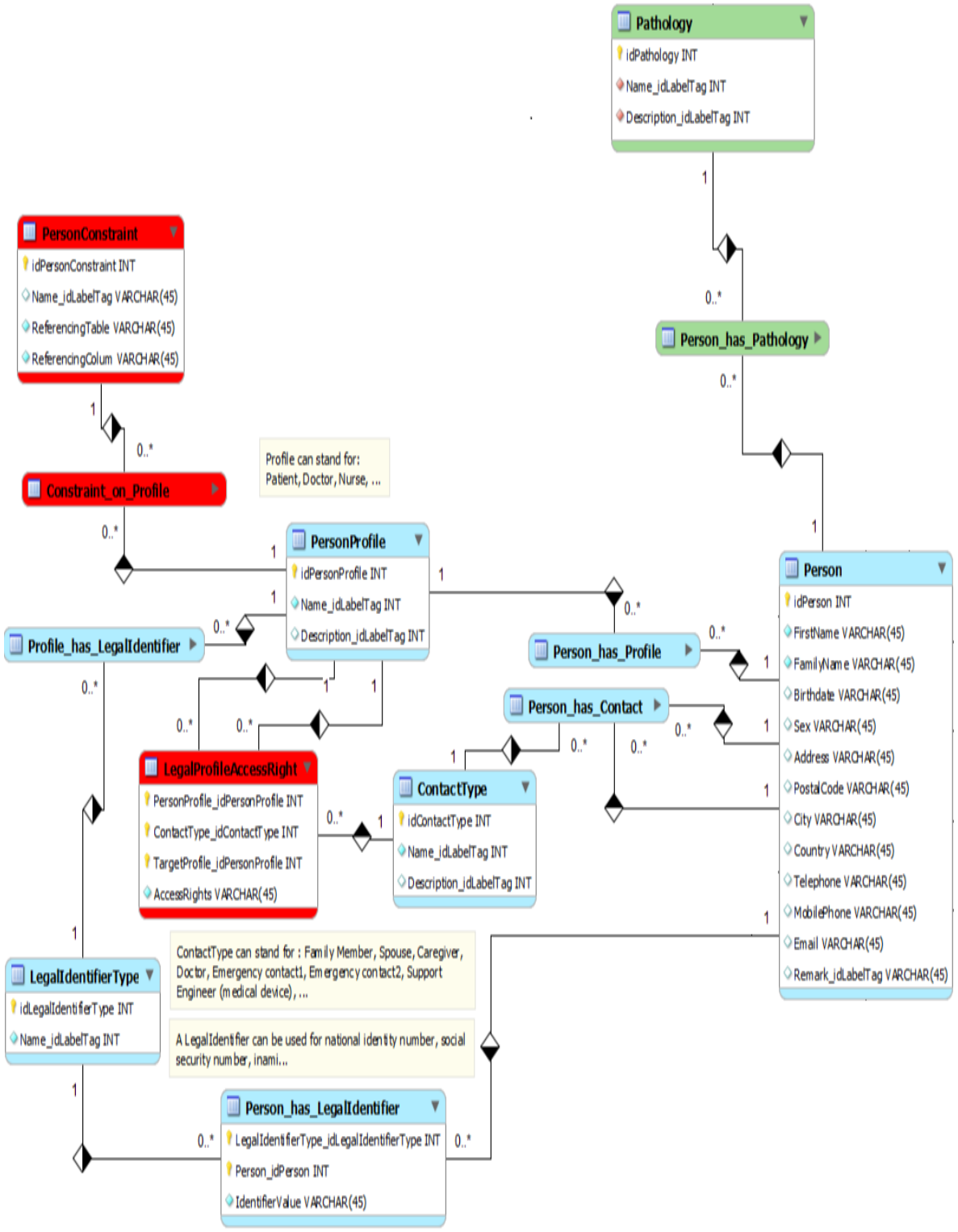


Synthèse

(1)	(2)
Table Permission (Gestion des droits d'accès dans la partie applicative)	Rôle Utilisateur (Privilège, rôle)

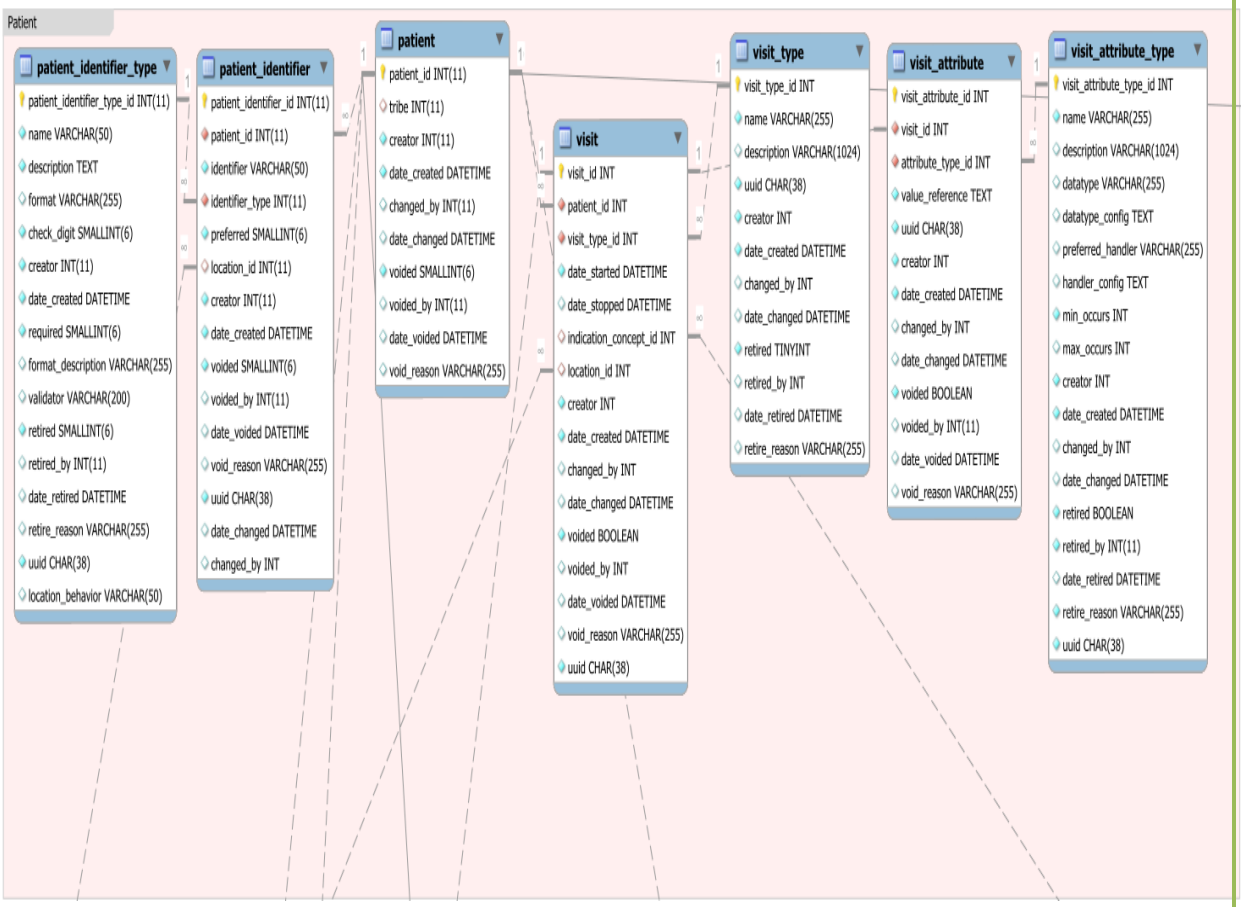
On peut faire l'analogie entre les 2 systèmes pour la gestion des droits d'accès des utilisateurs.

Model_DB_e-Health(1) : Suivi du Dossier Médical



Un patient est une personne souffrant d'une ou plusieurs pathologies. C'est le médecin traitant autorisé qui suit l'état de patient.

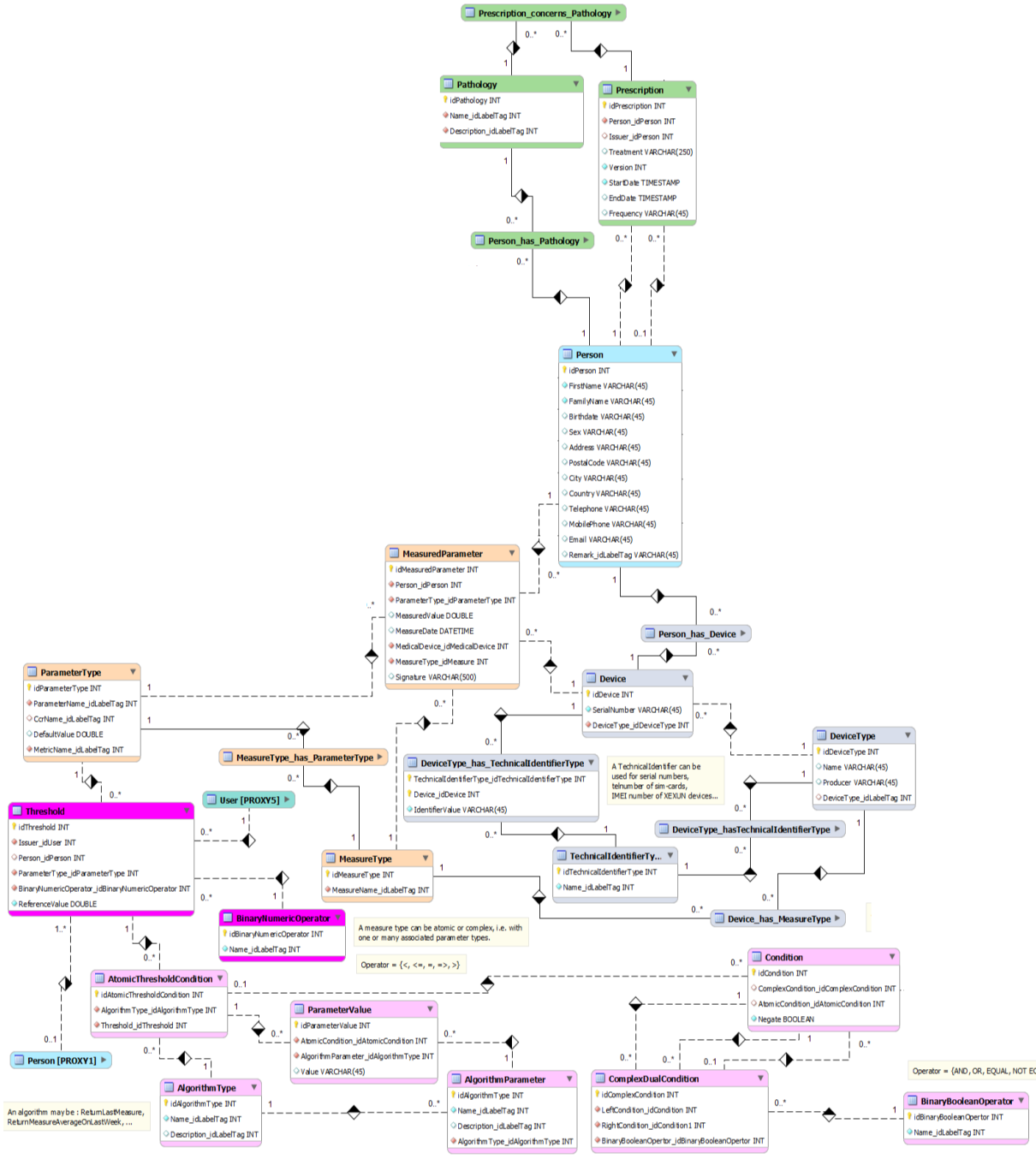
Model_DB_OpenMRS (2) : Suivi du Dossier Médical



Synthèse

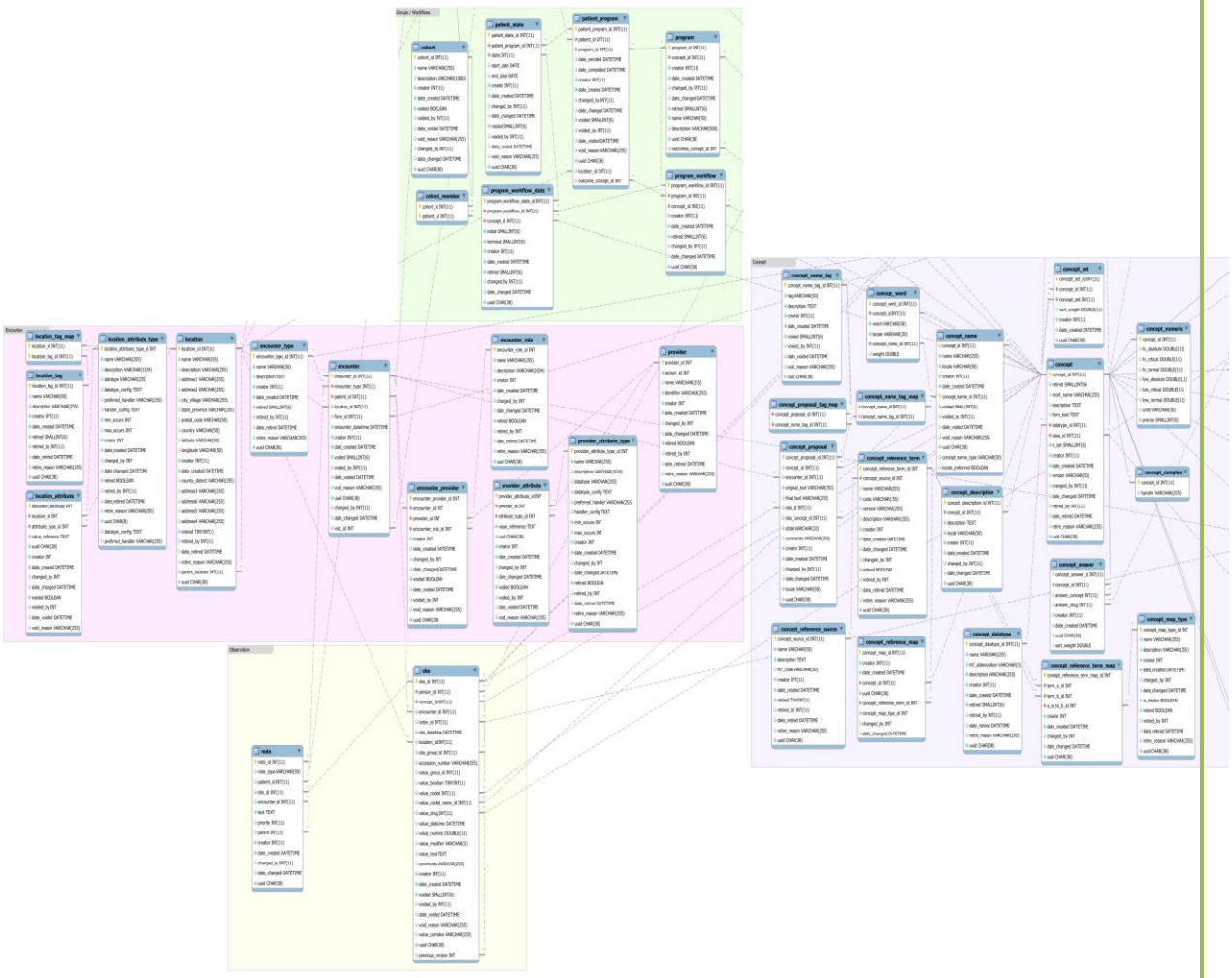
(1)	(2)
*Un patient est une personne qui a un identifiant logique légal pour lui. *Un patient souffre d'une pathologie	*Un patient est une personne qui a un identifiant logique qui peut être de type multiple *un patient doit passer des visites pour spécifier son état médical. *une visite englobe une ou plusieurs rencontres
On doit faire la personnalisation d'OpenMRS pour limiter l'accès uniquement au médecin traitant	

Model_DB_e-Health(1) : Prédéfinir les règles et les conditions



Un ensemble de règles et conditions sont prédéfinies pour automatiser la détection d'un état anormal d'un patient. La définition de celles-ci dépend de la pathologie.

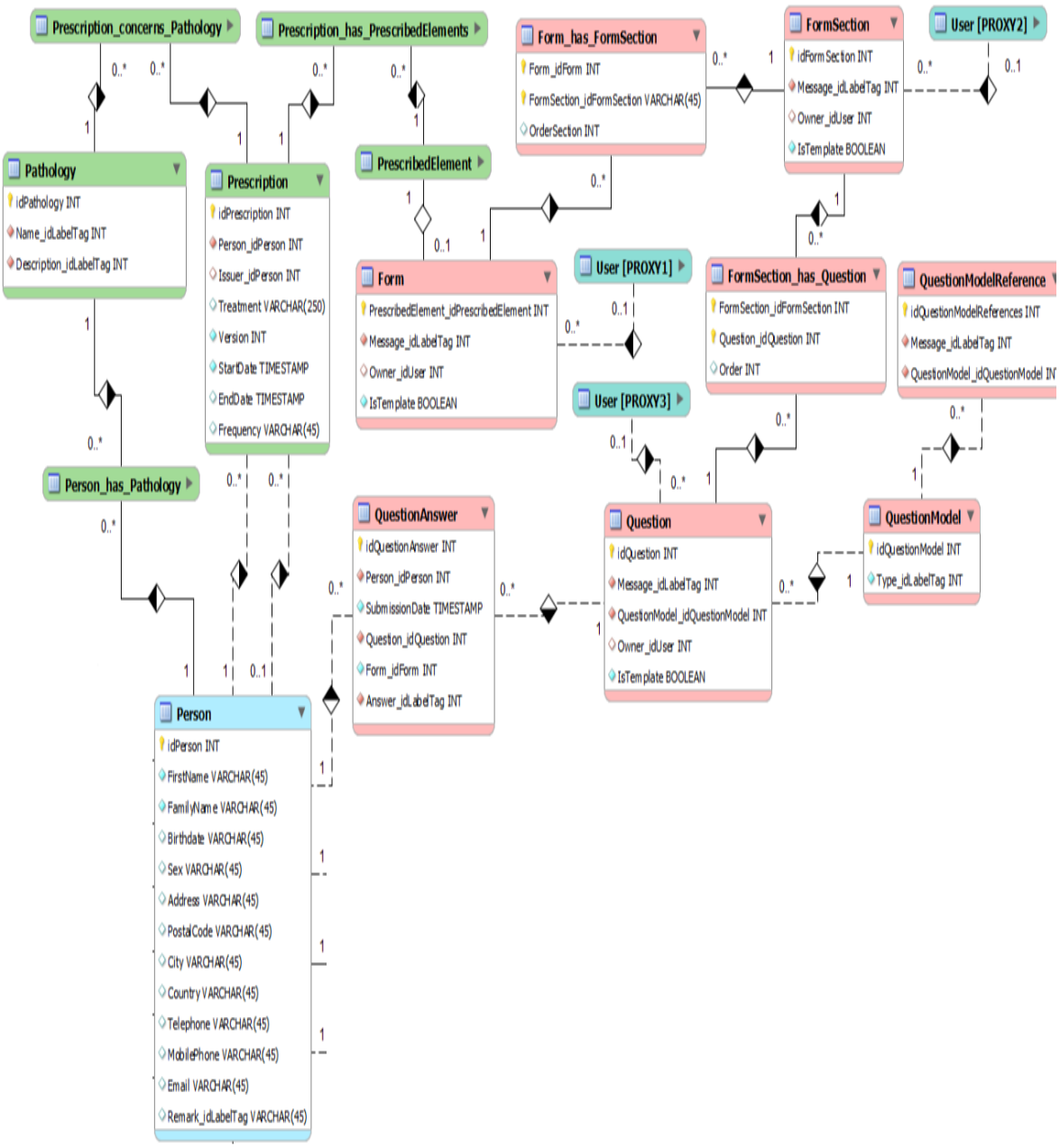
Model_DB_OpenMRS (2) : Prédéfinir les règles et les conditions



Synthèse

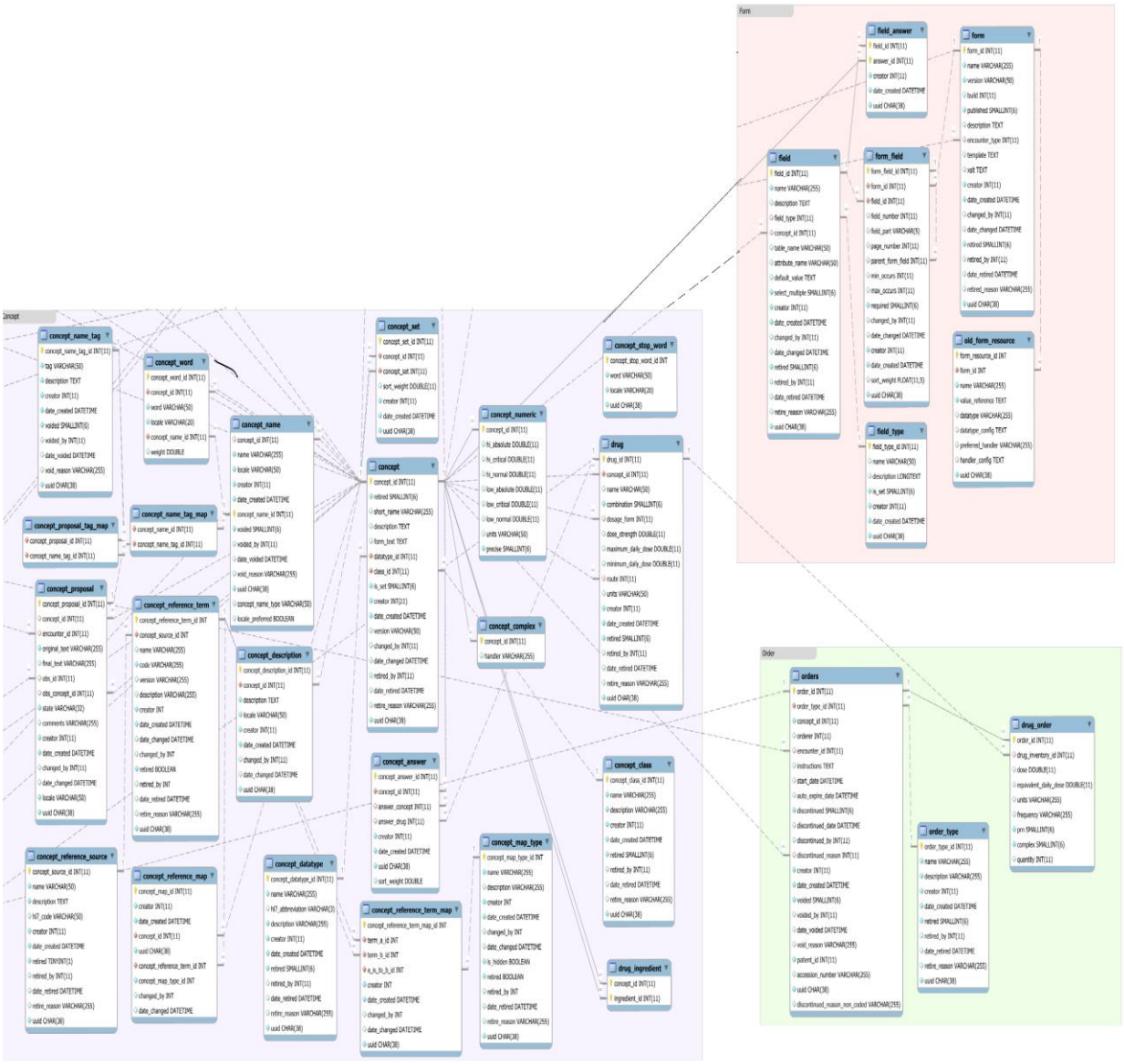
(1)	(2)
*Un traitement personnalisé est attribué à un patient (Vert : Pathology&Prescription) *Un patient interagit avec le système de soin afin d'évaluer son état (Exp : mesurer le taux de glycémie et vérification de condition) (Measures / Threshold /condition) *Un appareil a un type et un identifiant technique (Glucomètre, num de série) *Selon le type de l'appareil on liste l'ensemble de mesures (simple, complexe)	*un patient suit un programme de soins personnalisés (selon ses souffrances (maladie), allergies) *Un ou plusieurs Workflows peuvent être mis en place dans un programme (exp de workflow : l'état général d'un patient, état de grossesse) * une interaction spécifique entre le patient et le prestataire pour évaluer son l'état (des observations sont enregistrées) * un concept est représenté sur plusieurs classes(test , médicaments, diagnostic, question)
On peut faire l'analogie entre les 2 systèmes lors de la définition des règles et des conditions afin d'automatiser le suivi de patient à distance	

Model_DB_e-Health(1) : Interaction entre Patient et le Système de Suivi

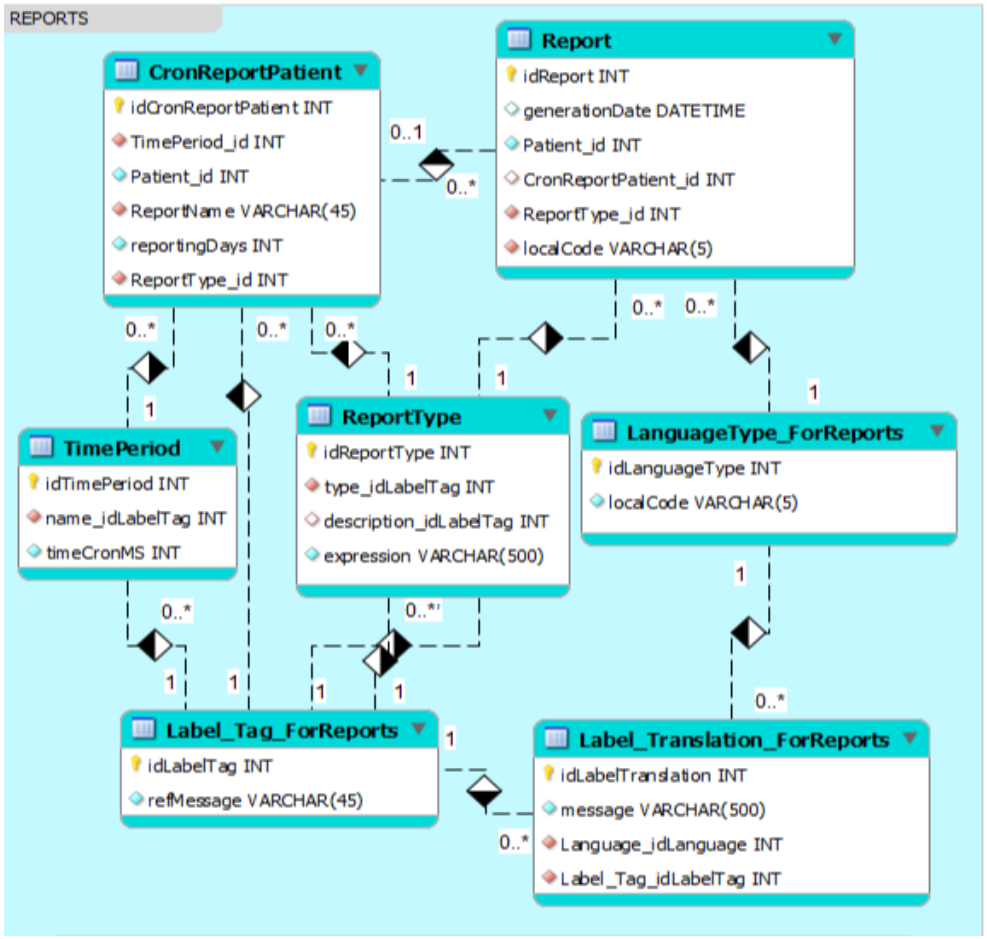


Un questionnaire est affiché par le système, selon la pathologie du patient, pour faciliter le suivi à distance.

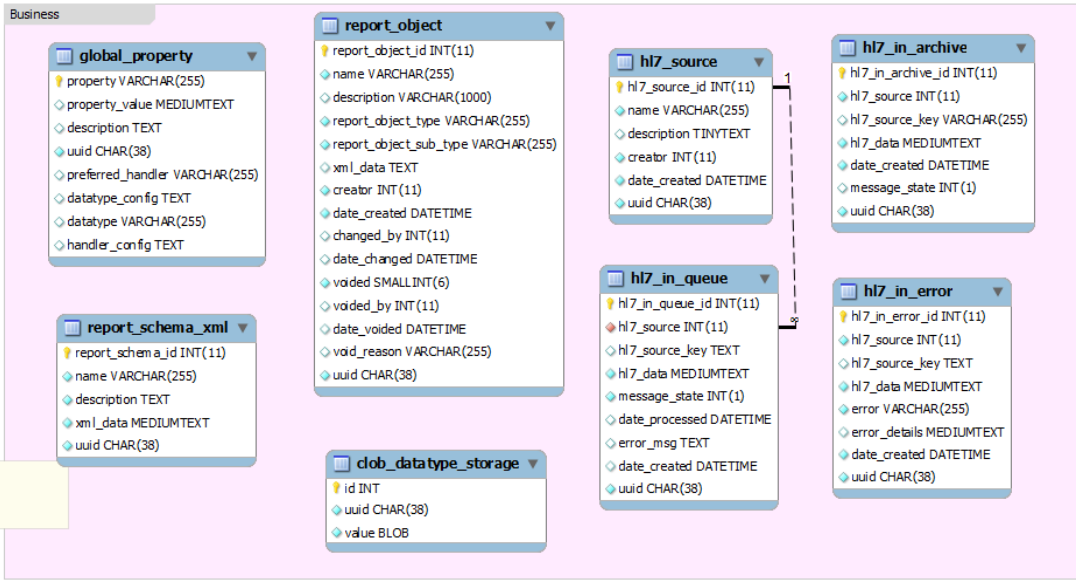
Model_DB_OpenMRS (2) : Interaction entre Patient et le Système de Suivi



Synthèse	(1)	(2)
	*Chaque traitement lui correspond un formulaire afin de faciliter l'interaction entre le patient et le système. Un formulaire(Forms) est composé de section dont chacune représente une question et qui suit un modèle donné * Sous-modèle Question	*Construire des formulaires est un processus qui implique la construction des concepts au sein de votre dictionnaire de concepts correspondant aux questions et aux réponses possibles sur votre formulaire *Sous-modèles Forms

	Personnalisation de l'interface utilisateur pour favoriser l'interaction avec le système médical de soin (Questions/Réponses)
Model_DB_e-Health(1) : Génération des rapports de suivi	
	Un rapport est généré par le système récapitulant l'état médical du patient

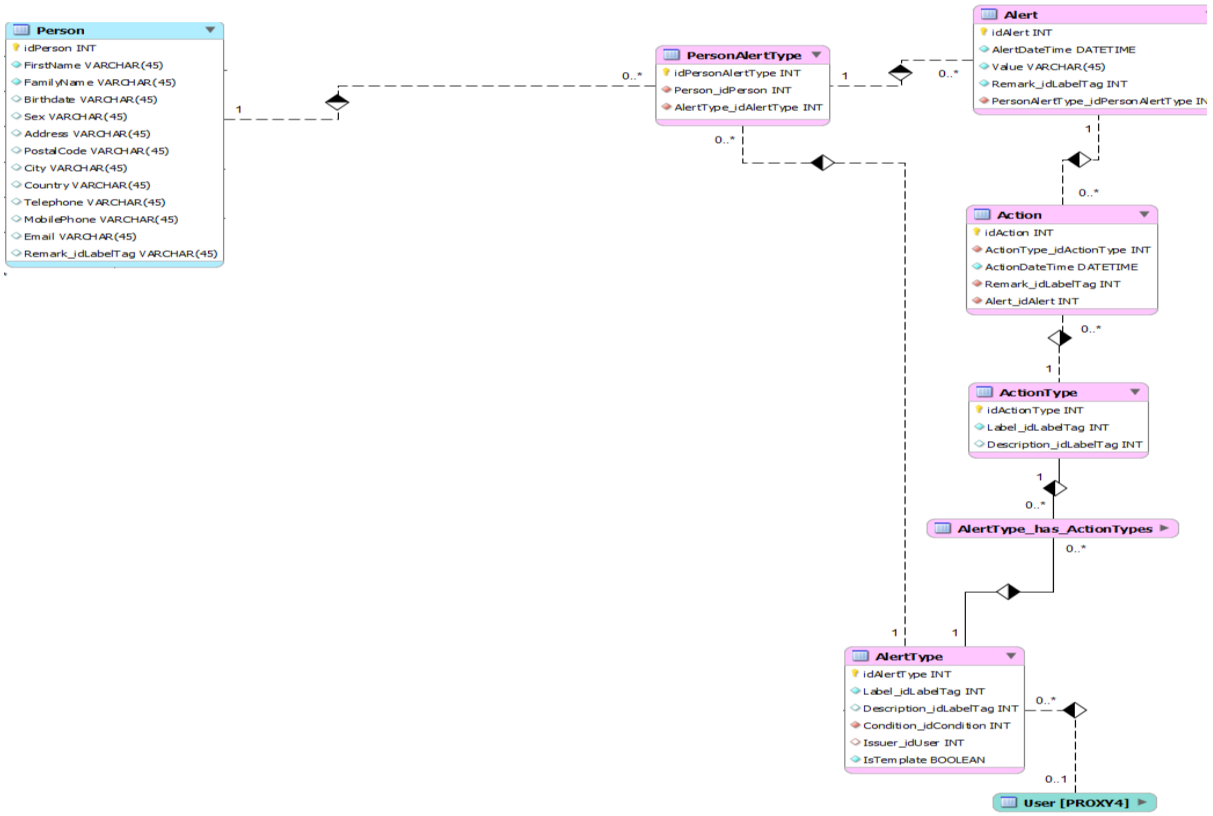
Model_DB_OpenMRS (2) : Génération des rapports de suivi



Synthèse

(1)	(2)
La génération des rapports personnalisés d'un patient est automatique périodique	Contient les données non–médicales qui spécifient les formats des rapports des utilisateurs
On peut faire l'analogie entre les 2 systèmes lors de la phase de génération des rapports médicaux	

Model_DB_e-Health(1) : Gestion des alertes



En cas d'un état anormal le système lance une alerte au médecin suiveur qui peut être un message, un bip sonore, etc.

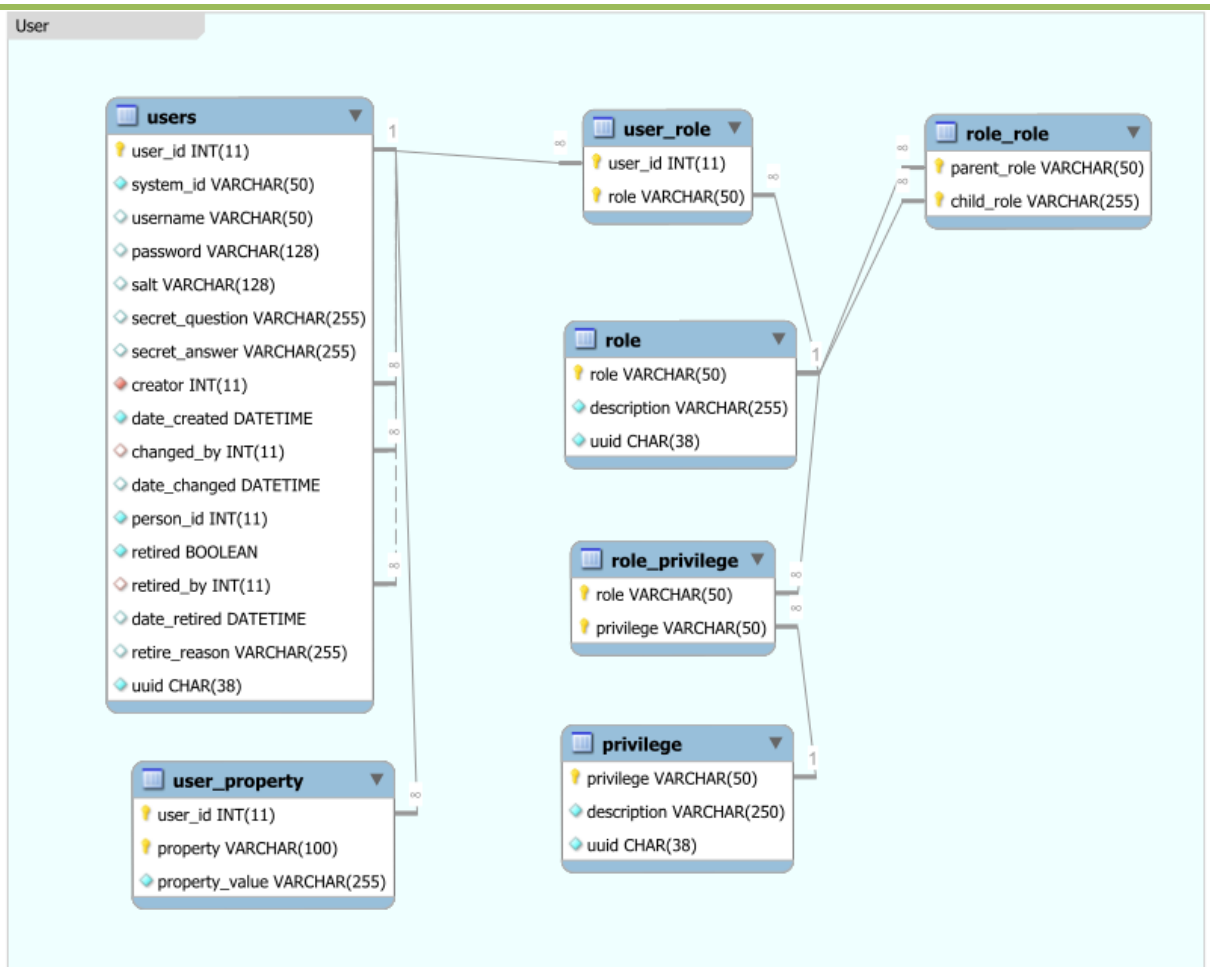
<table>
<tr><td>Model_DB_OpenMRS (2) : Gestion des alertes</td><td colspan="2"> 
</td></tr>
<tr><td rowspan="3">Synthèse</td><td>(1)</td><td>(2)</td></tr>
<tr><td>*Gestion des alertes
: sous-modèles _ alertes</td><td>*c’est dans la partie applicative,où la définition des alertes users et les actions (messages)</td></tr>
<tr><td colspan="2">Il faut personnaliser OpenMRS pour aboutir à l’analogie des 2 systèmes</td></tr>
</table>

Tableau 4 : Comparaison des modèles de BD eHealth et OpenMRS

IV.5 Conclusion

Dans cette section, nous avons abordé la conception de la base de données et de l’application afin d’identifier les principaux éléments et les différents fonctionnalités de système, essentiellement du point de vue de la sécurité.

La prochaine section est consacré à la mise œuvre du produit auquel nous exposons les différentes phases de réalisation prototype.

CHAPITRE

V. Réalisation

V. Réalisation

V.1 Introduction

La phase de réalisation est considérée comme étant la concrétisation finale du résultat de toute démarche de conception. En premier lieu, nous présentons l'environnement logiciel et les outils de développement du projet qui nous ont permis de réaliser cette application. Nous exposons l'ensemble des technologies utilisées ainsi que les raisons qui ont motivé ces choix parmi d'autres.

En second lieu, nous décrivons les différentes phases de réalisation de ce projet, sprint par sprint. :

- Sprint 0 : familiarisation et comparaison technologique,
- Sprint 1 : mécanisme d'identification,
- Sprint 2 : mécanisme d'authentification,
- Sprint 3 : mécanisme de signature numérique,
- Sprint 4 : intégration OpenMRS,
- Sprint 5 : contrôle et amélioration de la qualité de code.

V.2 Outils utilisés

Nous avons eu recours à un ensemble d'outils techniques pour préparer notre environnement de développement, présentons-les, ainsi que les versions utilisées :

❖ L'IDE Eclipse [18]

Eclipse est un projet de la Fondation Eclipse visant à développer tout un environnement de développement libre, extensible, universel et polyvalent.
Son objectif est de produire et fournir divers outils gravitant autour de la réalisation de logiciels, englobant les activités de codage logiciel proprement dites (avec notamment un environnement de développement intégré) mais aussi de modélisation, de conception, de test, de reporting , etc.
Son environnement de développement notamment vise à la généricité pour lui permettre de supporter n'importe quel langage de programmation.

La version d'Eclipse utilisée est : Eclipse –Juno for Java EE Developers version 4.2

❖ Le SMC[9] Maven 2 [19]

Maven est un outil qui peut être utilisé pour construire et gérer un projet basé sur Java. C'est une base de connaissances qui permet de faciliter la construction des projets, montrer une définition claire et standard des projets. Il offre aussi un moyen facile pour le partage des JAR[10] entre plusieurs projets et leurs publications.

L'objectif principal de Maven est de permettre à un développeur de compléter le développement dans la plus courte durée de temps. Afin d'atteindre cet objectif, Maven se concentre sur les domaines de préoccupation suivants :

- Faciliter le processus de compilation.
- Définir un système standard de compilation.
- Offrir des informations de qualité sur le projet.
- Donner un guide pour les bonnes pratiques de développement.

La version de Maven utilisée dans notre projet est : 2

❖ Le SCM Subversion [20]

Subversion (en abrégé **svn**) est un logiciel de gestion de versions (Source Code Management – SCM), distribué sous licence Apache et BSD. Il a été conçu pour remplacer CVS[11]. Ses auteurs s'appuient volontairement sur les mêmes concepts (notamment sur le principe du dépôt centralisé et unique) et considèrent que le modèle de CVS est le bon, et que seule son implémentation est en cause. Subversion a été écrit afin de combler certains manques de CVS. Voici les principaux apports :

- Les commits, ou publications des modifications sont atomiques. Un serveur Subversion utilise de façon sous-jacente une base de données capable de gérer les transactions atomiques (le plus souvent Berkeley DB) ;
- Subversion permet le renommage et le déplacement de fichiers ou de répertoires sans en perdre l'historique ;
- Les métadonnées sont versionnées : on peut attacher des propriétés, comme les permissions, à un fichier, par exemple.

Du point de vue du simple utilisateur, les principaux changements lors du passage à Subversion, sont :

---

[9] **SCM : S**oftware **C**onfiguration **M**anagement
[10] **JAR** : **J**ava **AR**chive
[11] **CVS** : **C**oncurrent **V**ersions **S**ystem

- les numéros de révision sont désormais globaux (pour l'ensemble du dépôt) et non plus par fichier : chaque patch a un numéro de révision unique, quels que soient les fichiers touchés. Il devient simple de se souvenir d'une version particulière d'un projet, en ne retenant qu'un seul numéro ;
- svn rename (ou svn move) permet de renommer (ou déplacer) un fichier ;
- les répertoires et métadonnées sont versionnés.

Une évolution plus récente est Git qui adopte un modèle décentralisé.

La version de SVN utilisée est : 1.6

❖ Plugin Maven Jetty [21]

Le plugin Maven Jetty est utile pour le développement rapide et le test. Vous pouvez l'ajouter à n'importe quel projet webapp qui est structuré selon les paramètres par défaut de Maven. Le plugin peut ensuite régulièrement analyser votre projet de changement et de redéployer automatiquement la webapp si vous en trouvez. Cela rend le cycle de développement plus productif en éliminant de construire et déployer des étapes: vous utilisez votre IDE pour apporter des modifications au projet, et le conteneur Web en cours d'exécution les sélectionne automatiquement, en vous permettant de les tester tout de suite.

La version de Plugin Jetty utilisée est : 6

❖ Le Server HTTP Apache [22]

Le logiciel Apache HTTP Server, mieux connu sous le nom d'Apache, est un serveur http, logiciel servant des requêtes respectant le protocole de communication client-serveur HyperText Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.
Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur Web.

Il existe plusieurs moyens de sécuriser l'accès aux ressources d'un serveur Apache :

- authentification des utilisateurs par les modules standards (mod_auth) et les modules tiers (LDAP, Smb, Kerberos, Radius, Oracle, etc.),
- sécurisation des requêtes par le protocole SSL (mod_ssl + certificat).

La version d'Apache utilisée pour ce projet est la : 7.012

❖ MySQL [23]

MySQL est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement)

La version de MySQL utilisée pour ce projet est : 5.5

❖ JAVA

La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. C'est la plateforme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisée par les informaticiens. Néanmoins, Java a été raffiné des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (Java Server Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Une **applet** Java est un programme Java utilisées pour fournir au sein de navigateur Web des fonctionnalités interactives qui ne peuvent pas être fournies par le langage HTML. Étant donné que le bytecode Java est multiplateforme, les applets Java peuvent être exécutées sur différentes plateformes, dont Windows, Unix, Mac OS et Linux.

Une **servlet** est une classe Java qui permet de traiter dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également l'être au format XML ou tout autre format destiné aux navigateurs web. Les servlets utilisent l'API Java Servlet (package javax.servlet).

L'utilisation de servlets se fait par le biais d'un conteneur de servlets côté serveur. Celui-ci constitue l'environnement d'exécution de la servlet et lui

permet de persister entre les requêtes des clients. L'API définit les relations entre le conteneur et la servlet. Le conteneur reçoit la requête du client, et sélectionne la servlet qui aura à la traiter. Le conteneur fournit également tout un ensemble de services standards pour simplifier la gestion des requêtes et des sessions.

Malgré que Java a été récemment assez controversé au niveau de la sécurité, une faille critique présente dans Java 7 a été détectée et permettrait l'installation de malwares, Oracle, l'éditeur de Java, a pu faire taire la polémique autour de la sécurité de son plugin avec un nouveau patch Java destiné à corriger 50 failles de sécurité, avec deux semaines d'avance sur le calendrier prévu.

La version de Java utilisée pour ce projet est : 7.0.13

Dans notre application nous allons utiliser la solution eGov eID pour l'identification, l'authentification et la signature électronique d'où le besoin de :

- Posséder une carte d'identité sur laquelle les fonctionnalités d'authentification et de signature sont actives (les personnes qui ont désactivé ces fonctions lors de la réception de la carte devront les faire activer dans leur commune au moyen du code PUK3).
- Avoir connaissance du code PIN de la carte (les personnes qui ont oublié leur code PIN devront se présenter dans leur commune pour le réinitialiser grâce au code PUK).
- Disposer d'un ordinateur.
- Disposer d'un lecteur de carte prévu à cet effet. Les modèles de carte à puce pouvant différer, vous ne pourrez pas lire la carte d'identité électronique sur un lecteur de carte bancaire qui n'a pas été prévu pour cela .
- Disposer d'une connexion Internet (si l'usage qui est fait de la carte n'est pas local).
- Installer le logiciel permettant à l'ordinateur de communiquer avec le lecteur de carte. Ce logiciel est appelé middleware.

❖ Middleware eID

Le middleware est un logiciel adapté à la carte d'identité électronique belge qui permet de communiquer avec le lecteur de carte. Il est mis à disposition par FEDICT. Il est parfois également appelé Run-time.

Le middleware se base sur le projet libre OpenSC qui a été adapté pour la carte d'identité électronique belge. Le code adapté a été publié et a été en partie inclus dans le projet standard. Les librairies OpenSC sont aussi utilisées pour les autres cartes d'identité électroniques adoptées par certains pays.

La version du middleware est téléchargeable, il est proposé pour 3 systèmes d'exploitation (Windows, Linux, Mac).Un guide d'installation y est fourni pour chacun des systèmes.

La version de Middleware utilisée pour ce projet est : 4.0.4

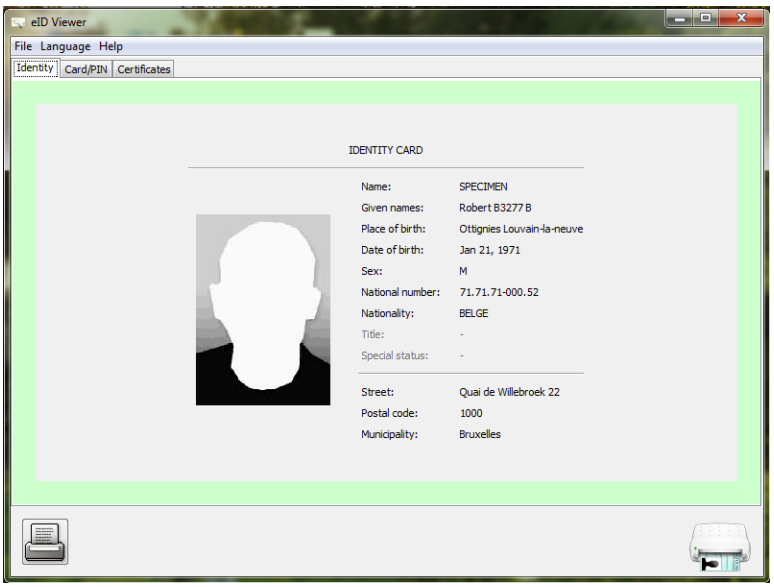


Figure 30:Middleware Belgique-eID

❖ Le Middleware Client Classic ToolBox [23]

Il s'agit du middleware utilisé pour la carte ANCE ou plus largement avec les cartes Classic TPC IM CC v3 est appelé « Classic Client ToolBox». Ce middleware supporte l'ensemble des cartes de la gamme « Classic TPC » qui fournit un certain nombre de fonctionnalités à l'utilisateur. Cet outil permet de :

- Administration de la carte : changement du PIN et déblocage (si autorisé).
- Visualiser le contenu de la carte : voir les informations sur la carte et les certificats stockés sur la carte (et choisir le cas échéant le certificat par défaut pour le logon par carte à puce).
- Réaliser des diagnostics (destinés aux interactions avec le support technique).

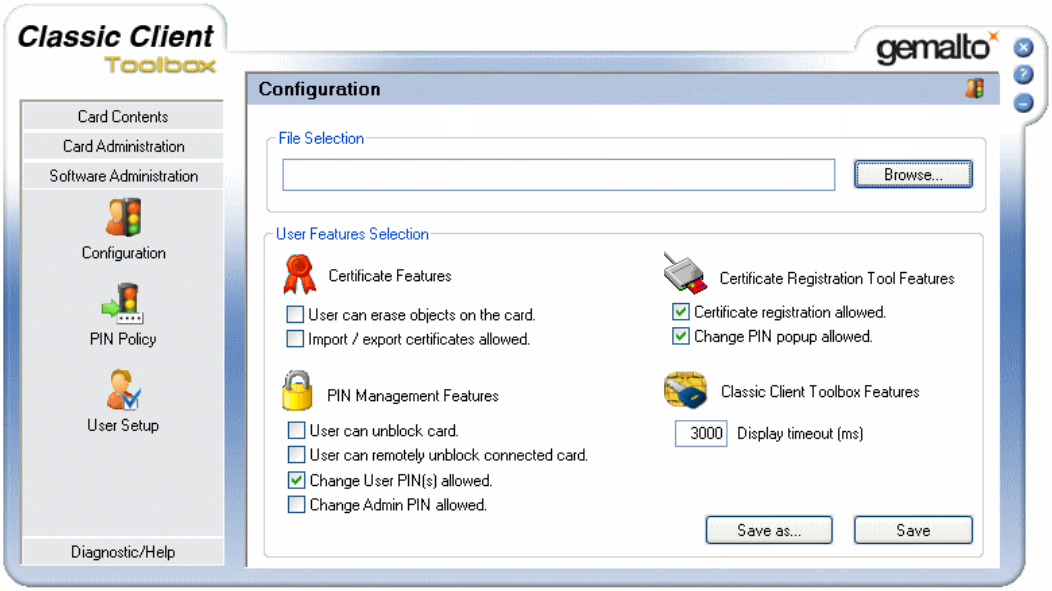


Figure 31 : Middleware Client Classic Toolbox

❖ Carte de test eID [24]

Afin de rendre plus accessible le développement d'applications pour l'eID, Fedict a déjà mis à disposition le middleware eID et le SDK[12] pour simplifier le développement ou l'intégration des solutions en se basant sur l'utilisation de l'eID. À côté des spécifications pour le matériel et les logiciels, la carte elle-même est devenue un outil de test permettant de contrôler le comportement d'une application en mode «end-to-end».

En 2009, une nouvelle méthodologie de test est proposée: elle comporte moins de limitations, engendre moins de frais, et repose largement sur une simulation de données identiques à celles utilisées en production. Cette solution consiste essentiellement en une carte (physique) de test à laquelle on peut lier plusieurs profils virtuels. Cette carte se comportera, selon le profil utilisé, comme une carte eID, une carte Kids-ID ou une carte pour étrangers. Nous pouvons créer nous-mêmes des profils au choix à l'aide du site de la carte d'identité belge électronique; cet environnement de test nous permet de créer jusqu'à 250 cartes eID virtuelles. Nous somme libre d'adapter la carte à nos propres besoins, bien qu'une sélection de profils standards sera présentée prête à l'emploi.

[12] SDK : Software Developement Kit

Figure 32: Carte de test-eID

Cette carte vierge avec ses certificats, codes PIN et codes PUK permet d'accéder au site de la carte d'identité belge électronique. Par carte, il est possible de personnaliser un profil ou de choisir un profil par défaut. Ces profils comprennent la vérification du statut des certificats (valide, suspendu ou révoqué). La validation des certificats peut alors être simulée et testée, à partir de votre application spécifique.

Dans l'environnement de test, nous aurons besoin d'un logiciel client PC/SC Proxy. Ce logiciel nous permet de tester nos développements en fournissant des cartes virtuelles à utiliser dans nos tests.

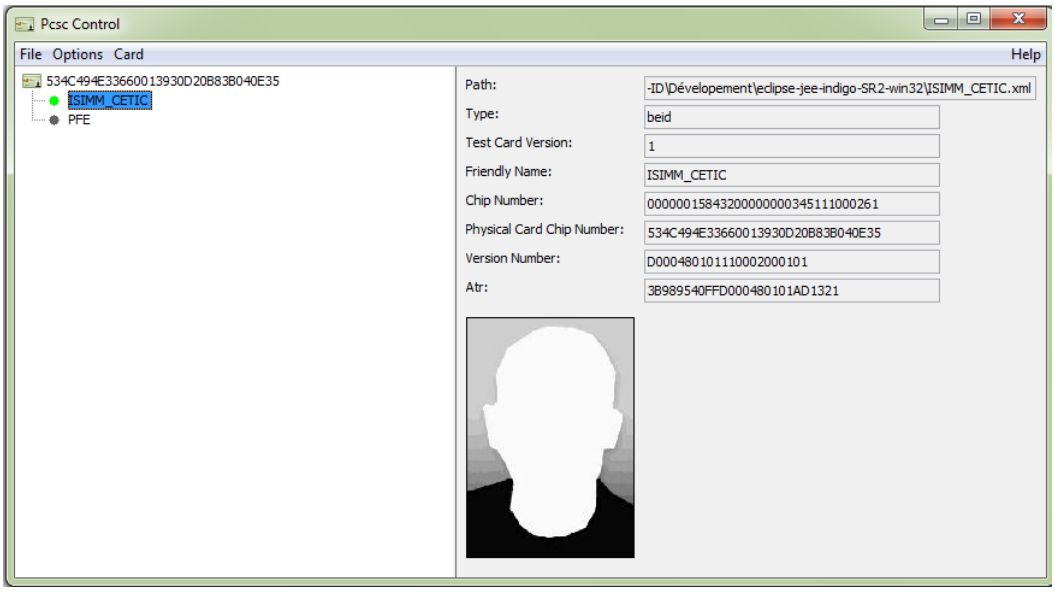


Figure 33 : Exemple de Carte eID virtuelle

❖ SONAR [25]

Sonar est un logiciel libre permettant de mesurer la qualité du code source sur les projets de développement Java. Le code source est analysé suivant sept axes :

- identification des duplications de code,
- mesure du niveau de documentation,
- respect des règles de programmation,

- détection des bugs potentiels,
- évaluation de la couverture de code par les tests unitaires,
- analyse de la répartition de la complexité,
- analyse du Design et de l'Architecture d'une application et en faire ressortir des métriques orientés objet.

L'ensemble de ces métriques qualités permet d'évaluer rapidement la dette technique de chaque projet. Une interface Web permet à la fois d'administrer l'outil (exclusion de code source, activation des profils qualité, définition des seuils d'alertes, …) et de consulter les résultats en croisant les indicateurs et en offrant plusieurs modes de restitution (clouds, treemap, hotspots, timemachine, etc.).

V.3 Sprint 0 – Familiarisation et comparaison technologique

Ce sprint a pour objectif de se familiariser avec la technologie. Commençons avec la comparaison de deux cartes l'une belge (eID) et l'autre tunisienne (ANCE). Puis, nous entamerons la manipulation des certificats pour configurer notre serveur pour fonctionner en HTTPS

V.3.1 Comparaison de deux cartes

Carte-eID

Carte-ANCE

	Propriétés des Cartes	
Contenu visible	Nom, prénom, lieu et date de naissance, sexe, nationalité, N° de la carte, période de validité, NRN, signature de titulaire, lieu de délivrance, autorité	Nom, prénom du propriétaire, date et heure d'expiration de la carte, adresse de l'autorité racine

Contenu Clés & Certificats	• 2 certificats : certificat d'authentification certificat de signature • 2 paires de clés • L'adresse (accessible électroniquement)	• Un seul certificat : Certificat Mail • 1 paire de clés (clé public + clé privée)
Format de Certificat	X.509 V3	X.509 (clé privée et clé publique)
Supports cryptographiques	carte à puce (Java Cards)	carte à puce (Smart Cards Gemalto)
Algorithme de Hachage	MD5,SHA_1,SHA_256	MD5, SHA_256
Algorithme de cryptages à clé public	RSA	RSA (algorithme de cryptographie asymétrique.)
Normes PKCS	PKCS#11, PKCS#15	PKCS#11
Taille de la clé	1024 ou 2048	2048

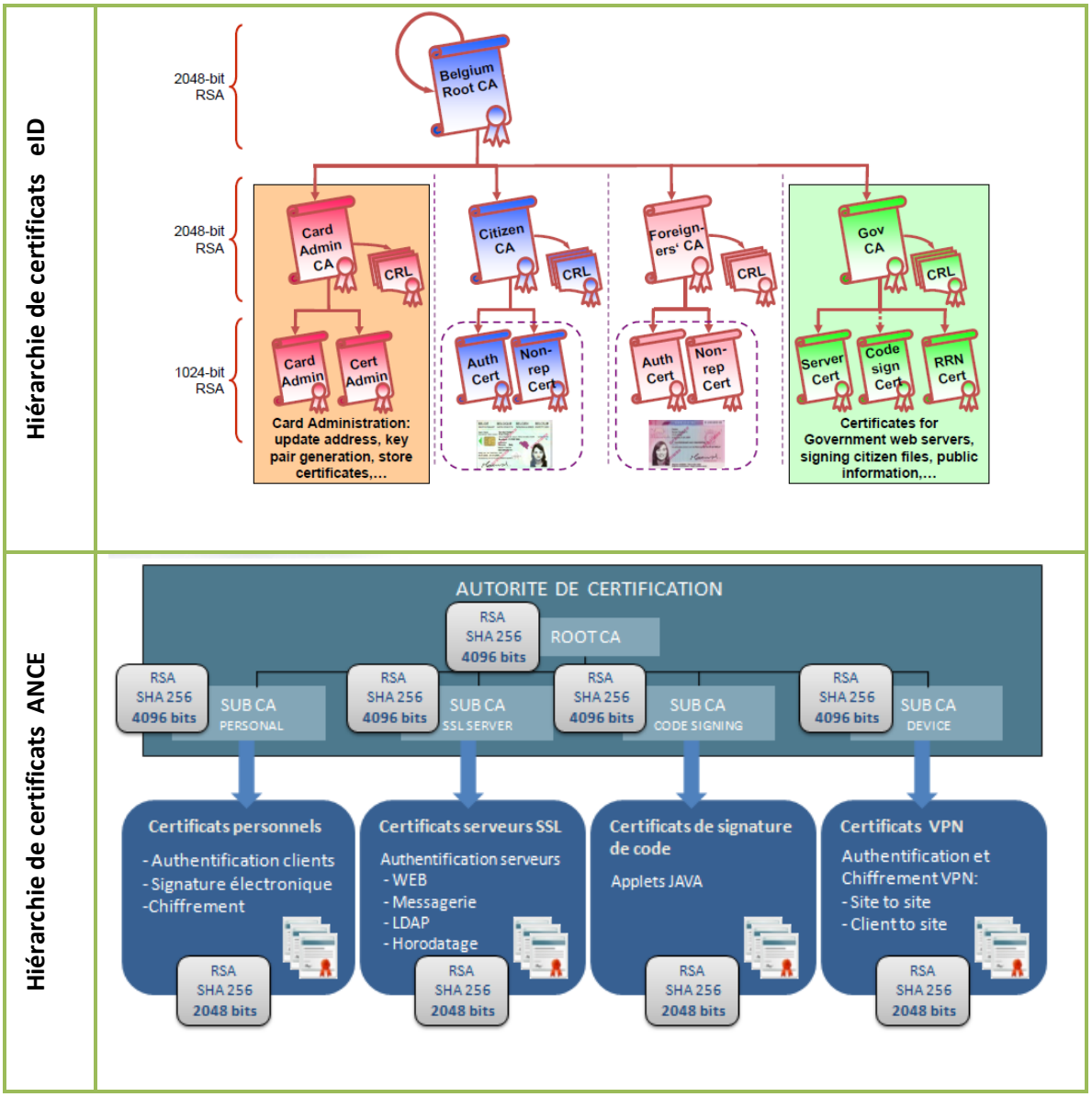
Type de Carte	3 types : Carte citoyen, Carte enfant, Carte étrangers 12-18 ans: carte eID avec uniquement le certificat d'authentification Signature activée à la majorité 18+ ans: carte eID complète Signature désactivable 0-6 ans: carte enfant sans certificat 6-12 ans: carte enfant avec uniquement le certificat d'authentification Étrangers: carte d'étranger .Elle est attribuée à tout étranger de plus de douze ans ayant obtenu un droit ou une autorisation de séjour en Belgique	un seul type : Carte-ANCE

De quoi avez-vous besoins ?	• Lecteur de carte (Un lecteur PC/SC) • Le logiciel eID (eIDViewer)	• Lecteur (un lecteur PC USB TR compatible PC/SC) • Le Middleware Classic Client de Gemalto
Domaines d'application	• e-Gov, • e-banking, • e-health, • e-commerce	• e-Gov, • e-banking, • e-health, • e-commerce
Prestataires de Services de Certification		
Prestataires de Services	• Fedict • Certipost • GlobalSign	• ANCE
Types de Certifications	• Citizen CA , Foreigner CA, Governement CA • Citizen CA , Foreigner CA, EtrustCA	• Certificat Mail, Certificat SSL, Certificat VPN, Certificat de signature de code
Fonctionnalités offertes		
Fonctionnalités	• L'identification • Saisie de données personnelles • L'authentification • La signature électronique	• L'authentification • La signature électronique

Tableau 5 : Tableau comparatif des cartes eID et ANCE

❖ **Détaillons l'architecture des certificats**

La provenance d'un certificat personnalisé est assurée de façon irréfutable par la signature du CA qui l'émet. L'authenticité du CA est à son tour assurée par la signature du Root CA qui émet les certificats des CA's. De cette façon une chaîne d'authenticité ou de certification se crée entre la racine du gouvernement et les certificats personnalisés qui se trouvent sur les cartes à puce électroniques.



V.3.2 Configuration du serveur web en mode sécurisé HTTPS

La couche de sécurisation est réalisée grâce à la technologie SSL, qui repose sur un procédé de cryptographie par clé publique, c'est-à-dire celui sur lequel sont basés les certificats.
Le principe de cette sécurisation consiste, après une étape d'authentification, à établir un canal de communication sécurisé (chiffré) entre le client et le serveur.

❖ Génération du Certificat Auto Signé

1. Ouvrez la console de commande sur n'importe quel système d'exploitation que vous utilisez et naviguez vers le répertoire où se trouve keytool.exe
2. Exécutez la commande suivante :

```
keytool -genkeypair -keyalg RSA -alias ClientCert -keystore "D:\ClientCert.crt" –storepass password  -validity 360 -keysize 2048
```

3. Remplissez les instructions pour votre information de l'organisation. Quand on vous demande votre nom et prénom, entrez le nom de domaine du serveur auquel les utilisateurs vont entrer pour vous connecter à votre application

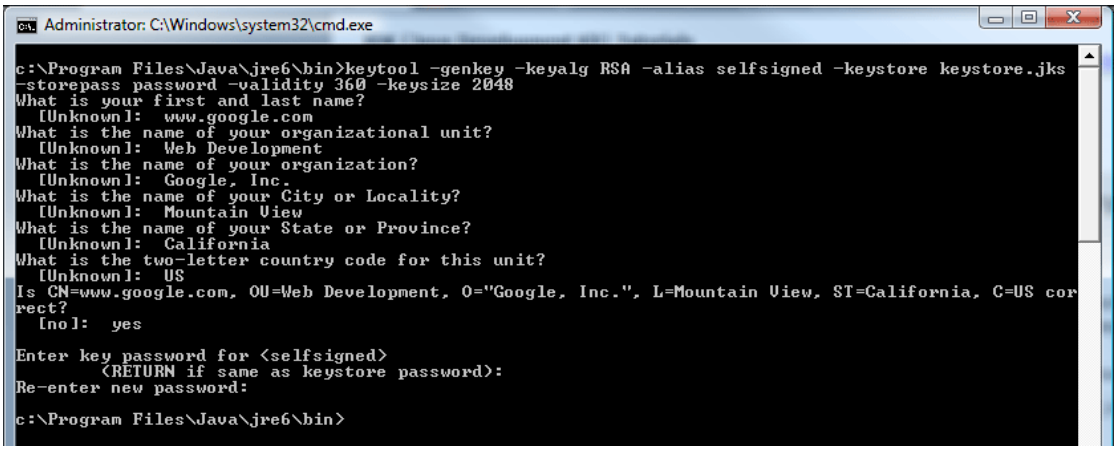


Figure 34: Capture de création d'un certificat

4. Editez le fichier server.xml de tomcat et il suffit d'ajouter le connecteur de SSL

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
SSLEnabled="true"       maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS" keystoreFile="C:\AmiraCert.crt"

keystorePass="password " />
```

Pour tester la mise en place du canal sécurisé SSL, essayons d'accéder à notre application web sécurisé via une adresse URL HTTP : il n'est désormais possible d'accéder à notre application qu'uniquement via le protocole sécurisé HTTPS.

Figure 35 : Test de SSL

L'accès au site HTTPS directement effectué depuis le serveur Web sécurisé nous montre plusieurs paramètres sur lesquels le certificat se base pour vérifier la validité de l'accès :

- L'autorité de certification privée : elle est bien connue du serveur Web.
- La date de validité du certificat émis par cet organisme : elle est bien valide et n'est pas périmée.
- Le nom du site sécurisé : il n'est pas correct. Pourquoi ?

  Tout simplement parce que lorsque la demande du certificat a été émise à l'organisme de certification privé, nous avions spécifié le Common Name (Nom usuel) comme étant égal à la valeur « amira service ». Comme nous tentons d'accéder à un site appelé localhost, il n'y a pas de correspondance entre les deux noms, d'où cet avertissement. Nous pouvons alors accepter quand même d'accéder à ce site : c'est à l'utilisateur final de décider de l'action à effectuer en toute connaissance de cause.

Côté ANCE nous avons utilisé openssl pour la génération des certificats :

```
<?php
$NAME                          = "Amira Kallel";
$ORGANIZATION                  = "ANCE";
$ORGANIZATION_UNIT             = "STAGE";
$MAIL                          = "kallel.amiraa@gmail.com";
$PKCS12_PWD                    = "123456";
$SERIAL                  = "9876543210";
$DAYS                          = 365;
$NAME_OUT_FILE = str_replace(" ", "-",$NAME."_".$ORGANIZATION);
$CLIENT_PATH                   = "certificates/".$NAME_OUT_FILE;
```

```
if(!file_exists($CLIENT_PATH)) mkdir($CLIENT_PATH);
$CA_PATH = "ca";
$CA_PATH = preg_replace('/\\\/','/', $CA_PATH);
$CONFIG = array( "config" => "$CA_PATH/openssl.cnf" );
$CLIENT_KEY = array(    'digest_alg'                    => 'sha256',
                        'private_key_type'          => OPENSSL_KEYTYPE_RSA,
                        'private_key_bits'          => 2048);
$CLIENT_DN = array(     "countryName"                       => 'TN',
                        "organizationName"          => $ORGANIZATION,
                        "organizationalUnitName"    => $ORGANIZATION_UNIT,
                        "commonName"                    => $NAME,
                        "emailAddress"                  => $MAIL);
$CA_CERTIFICATE         = file_get_contents($CA_PATH."/cacert.crt");
$CA_PRIVATE_KEY         = array(file_get_contents($CA_PATH."/cakey.pem"), "azerty");
$P12_ARRAY = array (    'extracerts'                    => $CA_CERTIFICATE,
                        'friendly_name'             => $NAME);
$CLIENT_PRIVATE_KEY         = openssl_pkey_new($CLIENT_KEY);
$CLIENT_REQUEST         = openssl_csr_new($CLIENT_DN,
$CLIENT_PRIVATE_KEY);
$CLIENT_CERTIFICATE         = openssl_csr_sign($CLIENT_REQUEST,
$CA_CERTIFICATE, $CA_PRIVATE_KEY, $DAYS, $CONFIG, $SERIAL);
$CLIENT_CERTIFICATE_RS      = openssl_x509_export($CLIENT_CERTIFICATE,
$CLIENT_CERTIFICATE_OUT);
file_put_contents($CLIENT_PATH."/".$NAME_OUT_FILE.".pem",
$CLIENT_CERTIFICATE_OUT);
openssl_pkcs12_export_to_file ($CLIENT_CERTIFICATE ,
$CLIENT_PATH."/".$NAME_OUT_FILE.".p12" ,$CLIENT_PRIVATE_KEY,
$PKCS12_PWD, $P12_ARRAY);
?>
```

Configurons Apache pour supporter une authentification par certificat !! dans le fichier ssl.conf du répertoire du /etc/httpd/conf.d nous ferons les modifications suivantes :

```
<VirtualHost localhost:443>
DocumentRoot "/var/www/html/repertoireSite"
ServerName www.service.tn
SSLCertificateFile /etc/httpd/conf.d/www.service.tn.crt
SSLCertificateKeyFile /etc/httpd/conf.d/www.service.tn.key
SSLCertificateChainFile /etc/httpd/conf.d/ANCETestCA.crt
SSLCACertificateFile /etc/httpd/conf.d/ANCETestCA.crt

SSLVerifyClient optional
SSLVerifyDepth 2
</VirtualHost>
```

Où

/var/www/html/repertoireSite : le chemin vers l' application web
www.service.tn.crt: le certificat du serveur
www.service.tn.key: la clé privée du serveur
ANCETestCA.crt: le certificat de l'autorité
www.service.tn: le nom de domaine du site à insérer au niveau du fichier /etc/hosts

V.3.3 Interaction des couches et planification des sprints

Tout est prêt !! Nous avons créé une application web Maven pour réaliser les 3 premiers sprints où l'interaction entre ses couches se récapitule dans ce schéma :

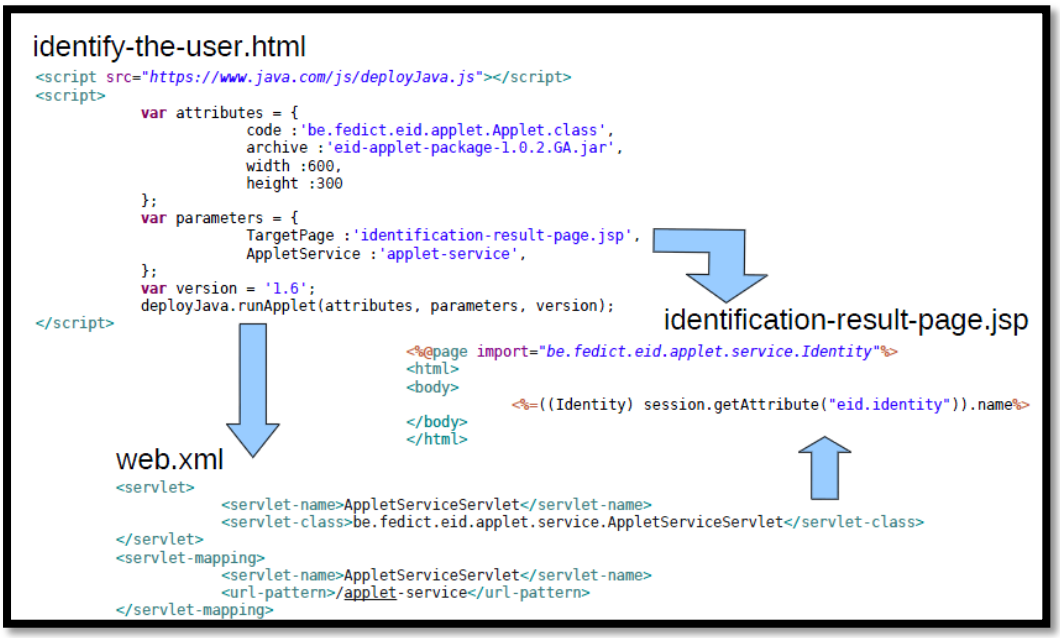


Figure 36: L'interaction entre model (applet), la vue (jsp) et partie Controller (servlet)

Nous allons suivre l'architecture MVC (Modèle-Vue-Controller) pour le prototype eID, où nous séparons les trois couches : présentation (vue), métier (contrôle) et gestion des données.

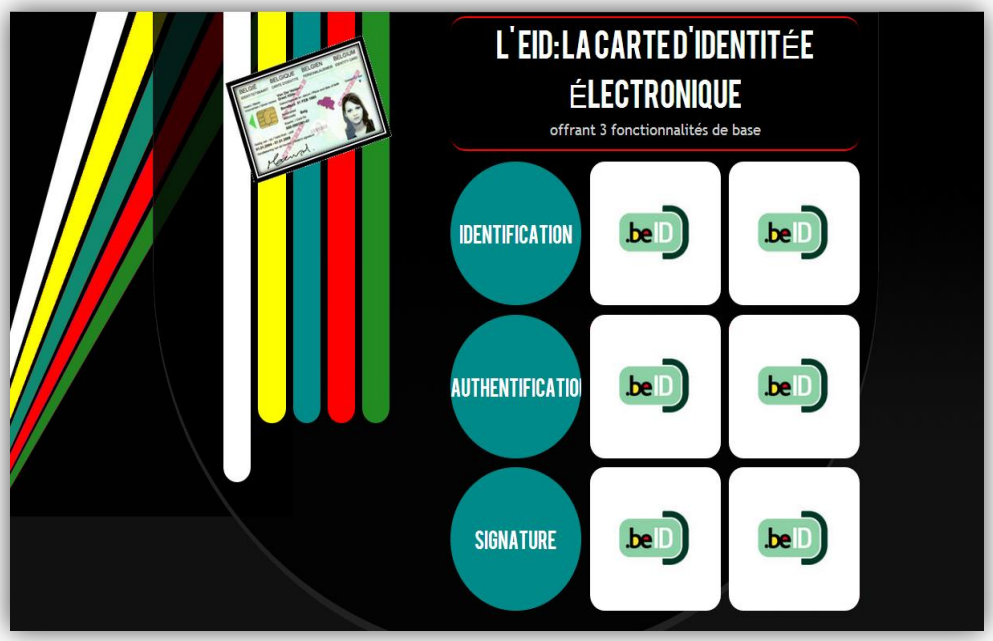


Figure 37 : Menu principale du prototype

Ce prototype présente les fonctionnalités de base offertes par une carte d'identité électronique eID qui sont : l'identification, l'authentification et la signature électronique.

V.4 Sprint 1 – Identification avec la carte eID

Ce sprint a pour objectif de mettre en place la phase d'identification où la récupération des informations d'une personne s'effectue à travers la carte eID afin d'éviter l'attaque d'écoute du clavier ou « **key logger** » qui permet de « capturer » l'ensemble des touches tapées sur un clavier.

La liste des figures suivantes récapitule l'exécution de service, d'une part pour la génération d'un fichier XML (destiné à un traitement par web-service) et d'autre part pour la génération d'un PDF (pour le traitement par un humain).

V.4.1 Identification-XML

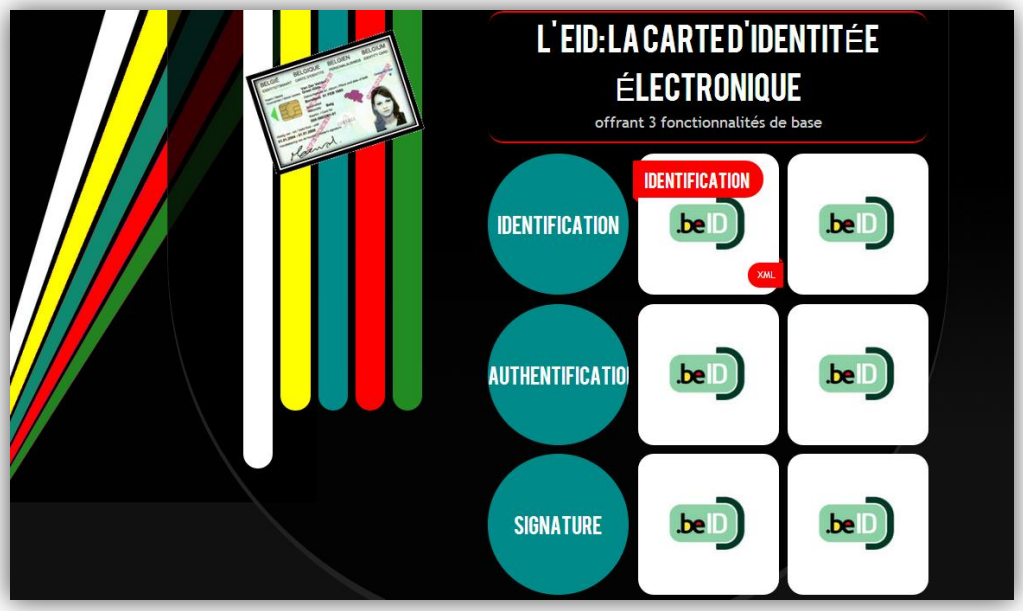


Figure 38 : Choix de service

L'utilisateur choisit l'exécution du service d'identification dont le résultat final est un fichier XML qui récapitule les informations fondamentales d'une personne.

Figure 39: Détecter le branchement du lecteur de carte

Le navigateur lance l'exécution de l'applet de service identification et vérifie le branchement du lecteur

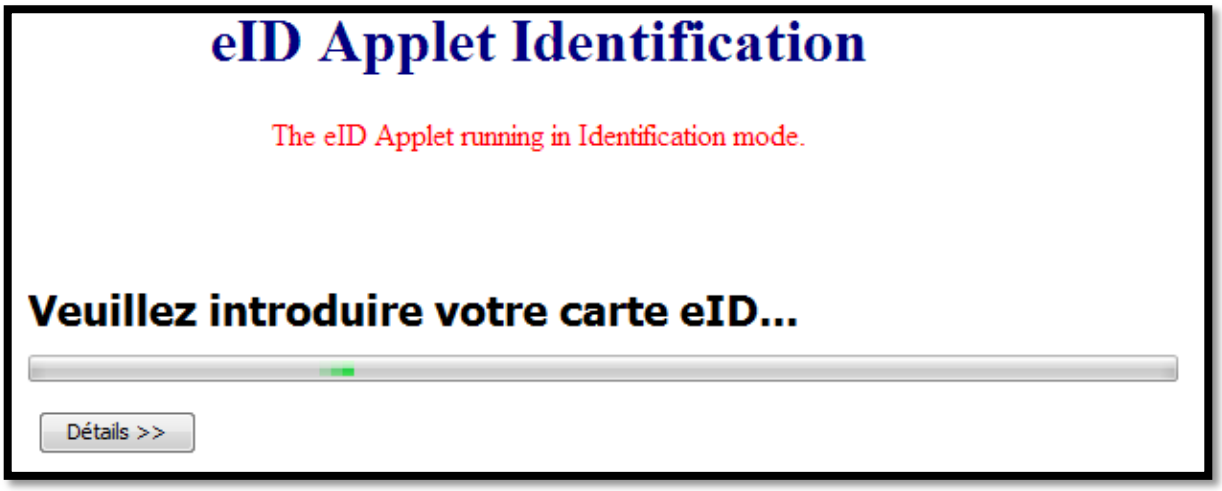


Figure 40: Détecter la présence de la carte

Une fois branché, nous passons à la vérification de la présence de la carte dans le lecteur

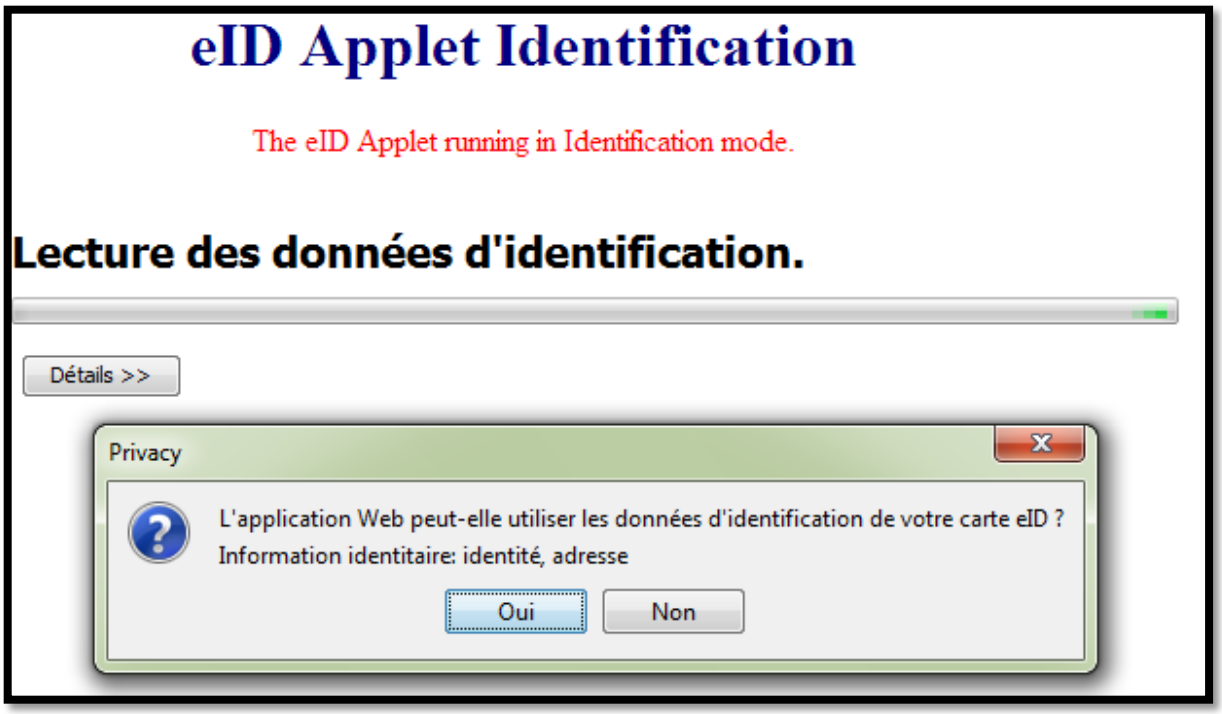


Figure 41: Autoriser l'utilisation des informations d'identification

Les données d'identification sont récupérées à travers la carte eID mais il faut avoir l'autorisation pour qu'elles soient utilisées dans l'application web.

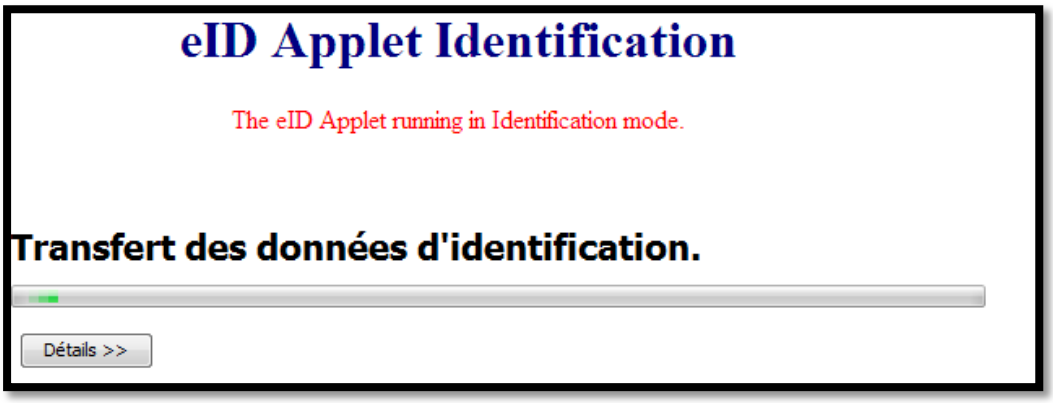


Figure 42: Données en cours de transmission

Suite à l'autorisation, l'application web a pu transmettre les données d'identification.

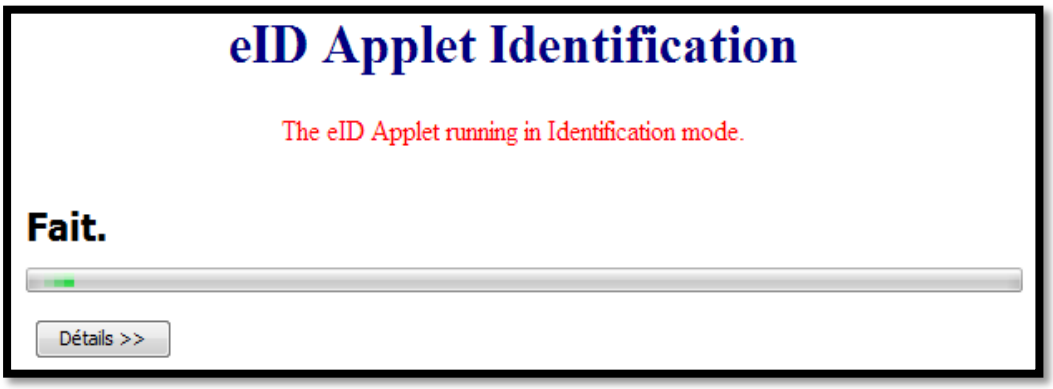


Figure 43: Transmission avec succès

La transmission est terminée avec succès.

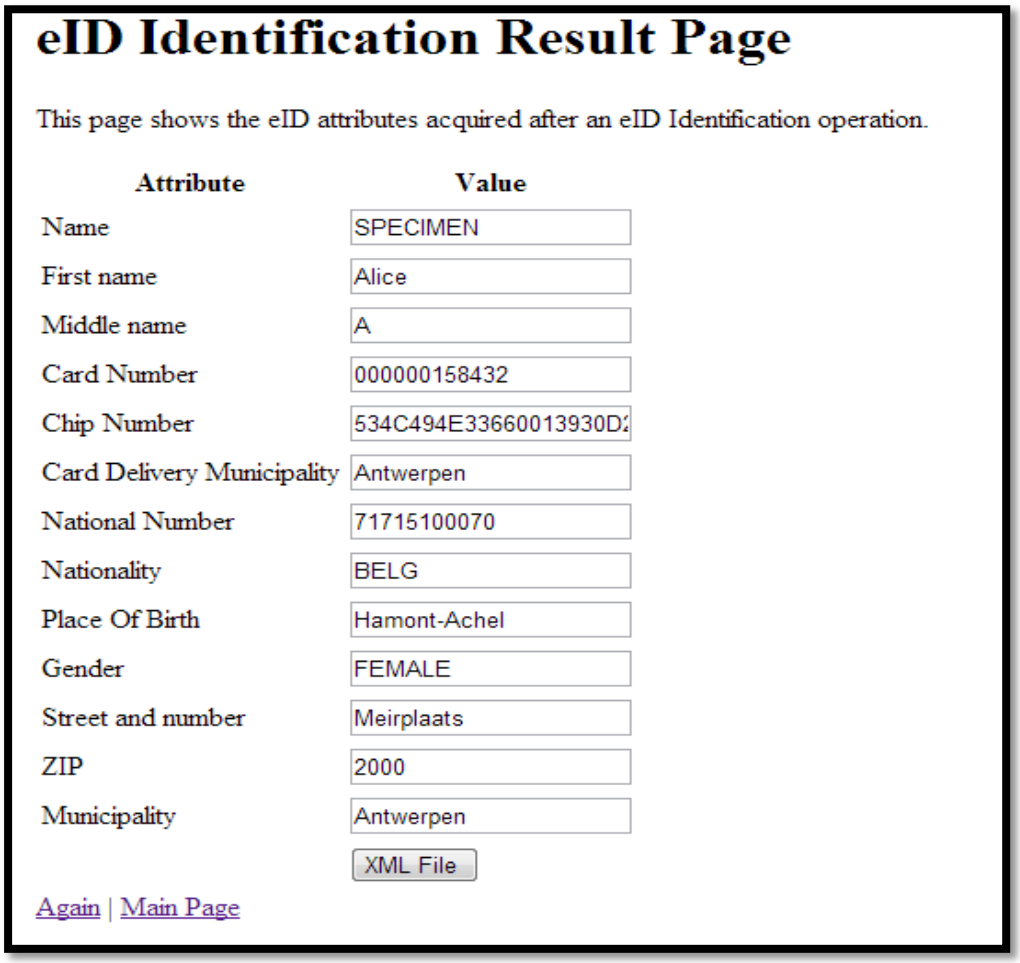


Figure 44: Réorientation à la page résultat et affichage de données

Affichage des données d'une personne pour l'identifier.

```
<?xml version="1.0" encoding="UTF-8"?><Personne>
<Name>SPECIMEN</Name>
<Firstname>Alice</Firstname>
<CardNumber>000000158432</CardNumber>
<ChipNumber>534C494E33660013930D20B83B040E35</ChipNumber>
<Gender>FEMALE</Gender>
<Nationality>BELG</Nationality>
<CardDeliveryMunicipality>Antwerpen</CardDeliveryMunicipality>
<NationalNumber>71715100070</NationalNumber>
</Personne>
```

Figure 45 : Générer un fichier XML

Cliquons sur le bouton File XML, un fichier XML est généré dont la racine est l'objet personne et les fils sont ses caractéristiques.

V.4.2 Identification-PDF

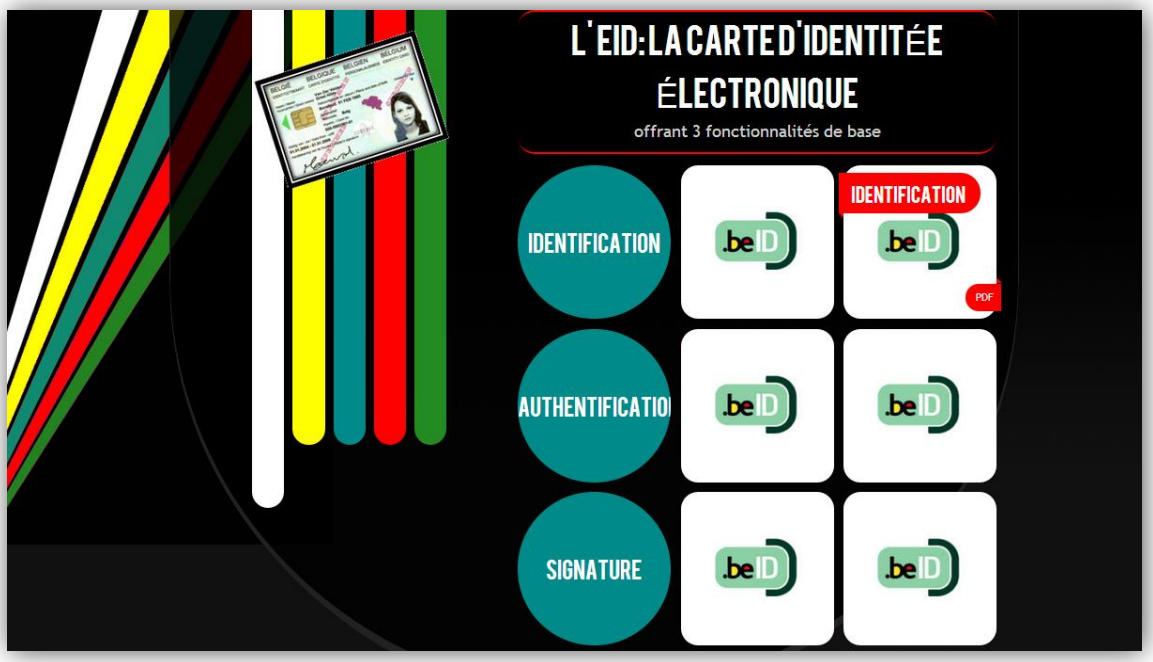


Figure 46 : Choix de service

L'utilisateur choisit l'exécution du service d'identification dont le résultat final est un fichier PDF qui récapitule les informations fondamentales d'une personne.

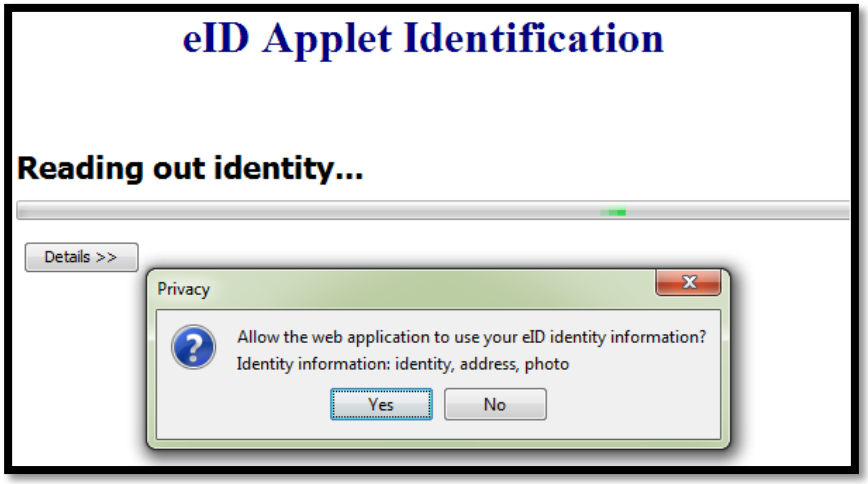


Figure 47 : Autoriser l'utilisation des données d'identification

Les données d'identification (identité, adresse et photo)sont récupérées à travers la carte eID mais, ici aussi, il faut avoir l'autorisation pour qu'elles soient utilisées dans l'application web.

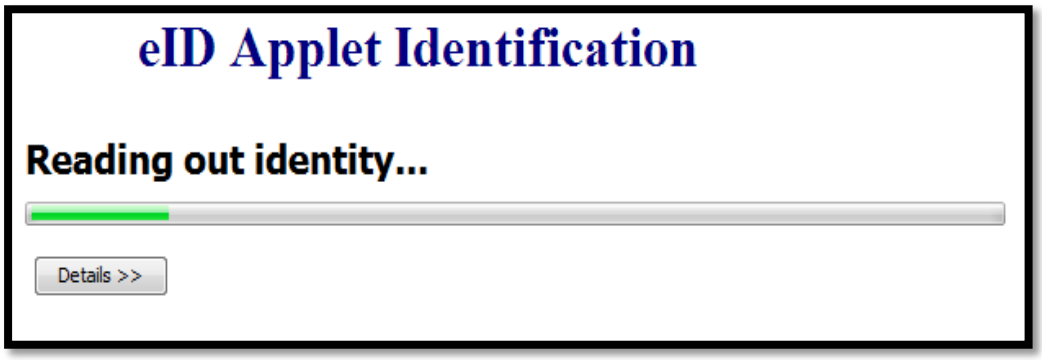


Figure 48: Lecture de l'identité

Lecture des données d'identification en cours.

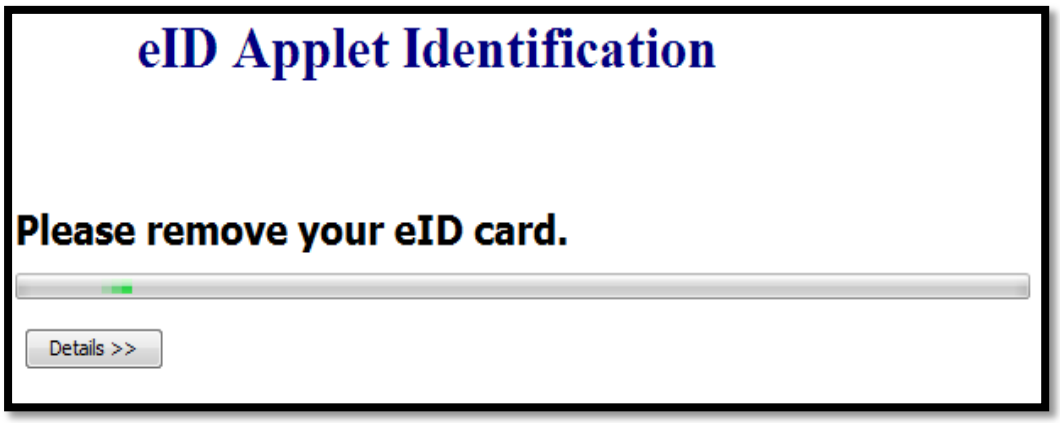


Figure 49 : Retirer la carte eID

Ici, nous avons ajouté l'option de récupérer la carte avant de transmettre les données.

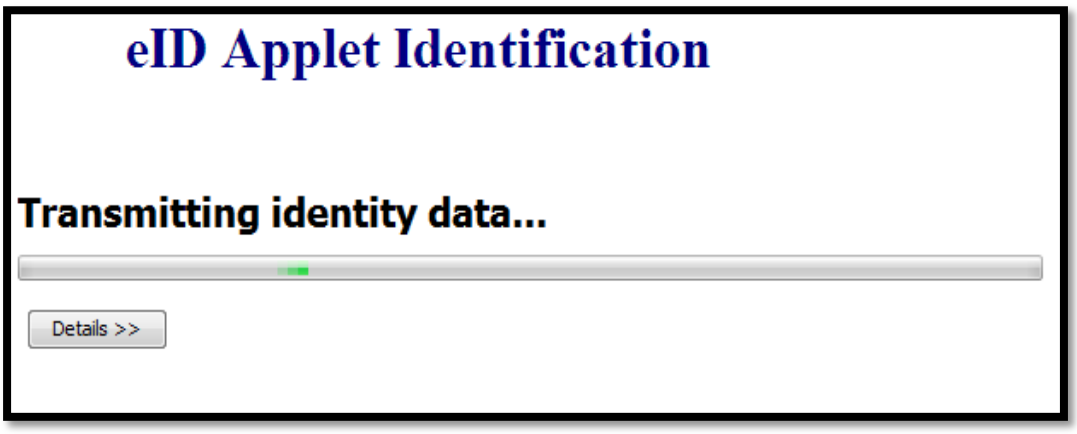


Figure 50 : Transmission des données d'identification

Transmission des données d'identification en cours.

Figure 51 : Affichage des données d'identification

Affichage des données d'une personne pour l'identifier.

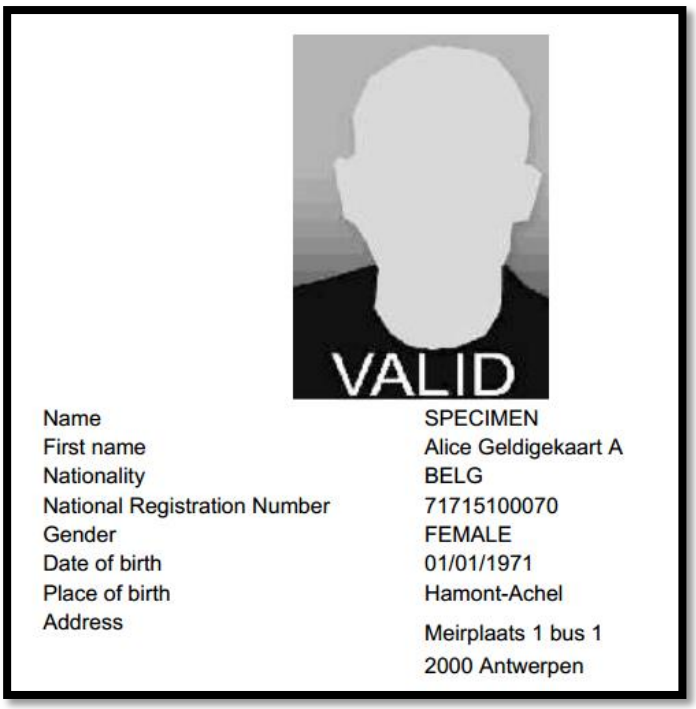


Figure 52 : Fichier PDF généré

Cliquons sur le bouton PDF File, un fichier PDF est généré dans lequel nous trouverons un récapitulatif des données qui permettent d'identifier une personne (voir figure ci-dessus).

V.5 Sprint 2 – Authentification

Ce sprint a pour objectif de mettre en place la phase d'authentification, où le certificat d'authentification est récupéré à travers la carte eID pour authentifier l'utilisateur. Elle est basée sur ce que l'utilisateur possède (la clé privée) aussi bien que sur ce que l'utilisateur sait (le mot de passe qui protège cette clé privée) afin d'éviter l'utilisation de l'authentification par login et mot de passe qui cause plusieurs risques d'attaques.

La liste des figures suivantes récapitule l'exécution de service, d'une part pour une authentification simple et d'autre part pour une authentification poussée qui procède à la vérification complète de la chaîne des certificats.

V.5.1 Authentification-Simple

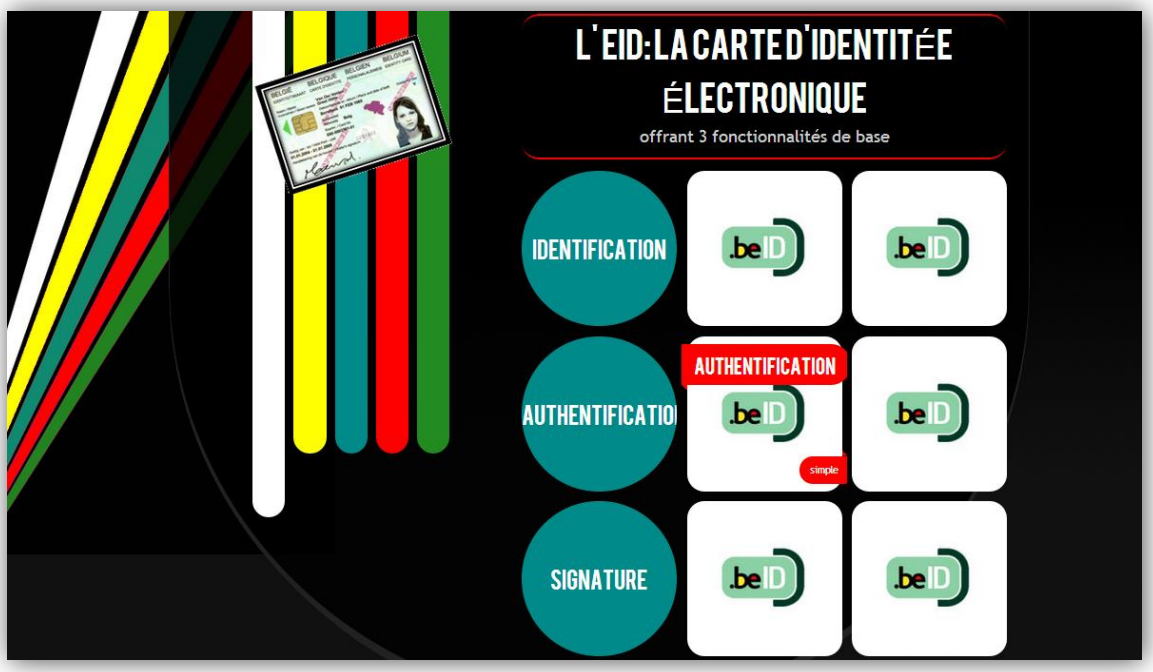


Figure 53 : Choix de service

L'utilisateur choisit l'exécution du service d'authentification où il prouve l'identité d'une personne en utilisant un code PIN.

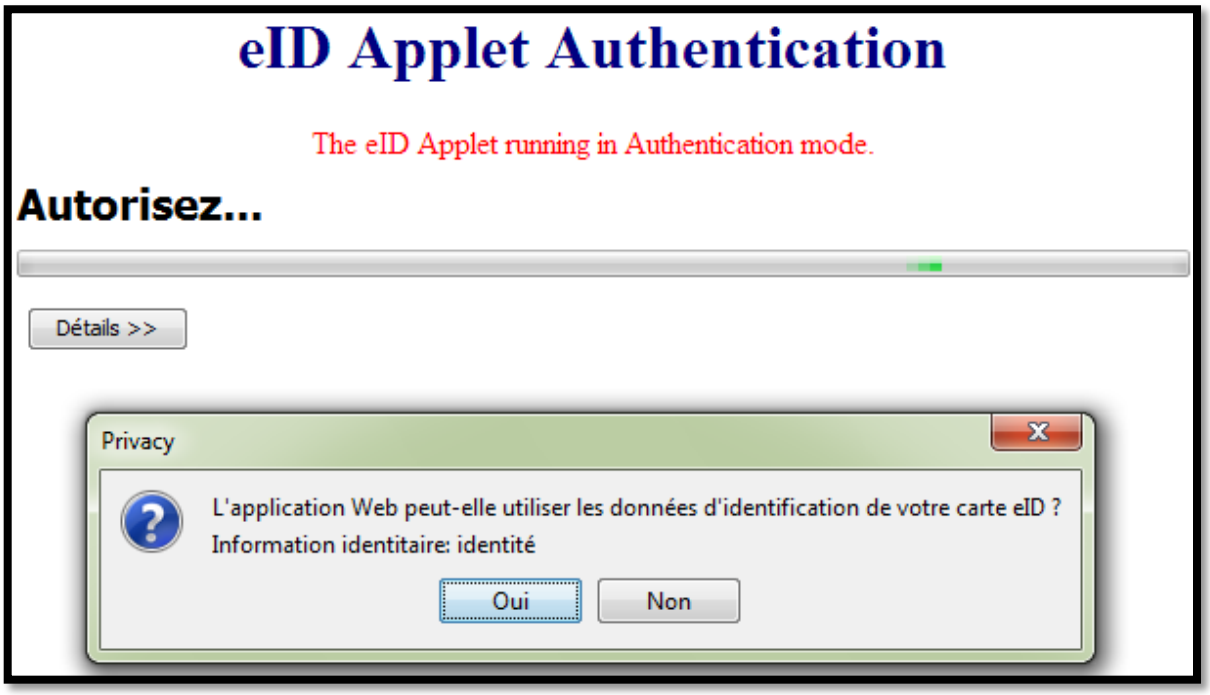


Figure 54 : Autoriser l'utilisation des données d'identification

Une fois que notre lecteur est bien branché et que la carte eID est présente, une validation d'utilisation des données d'identification dans l'application web est demandée.

Figure 55: Insertion du Code PIN

Afin d'utiliser le certificat d'authentification, protégé par une clé privée, l'utilisateur doit présenter le mot de passe valide de cette dernière (PIN).

Figure 56: Authentification réussie et affichage de l' identifiant de l'utilisateur

L'authentification est réussie pour notre utilisateur et réorienté vers une page résultat qui affiche l'identifiant d'utilisateur ainsi que son nom complet.

V.5.2 Authentification-Certs

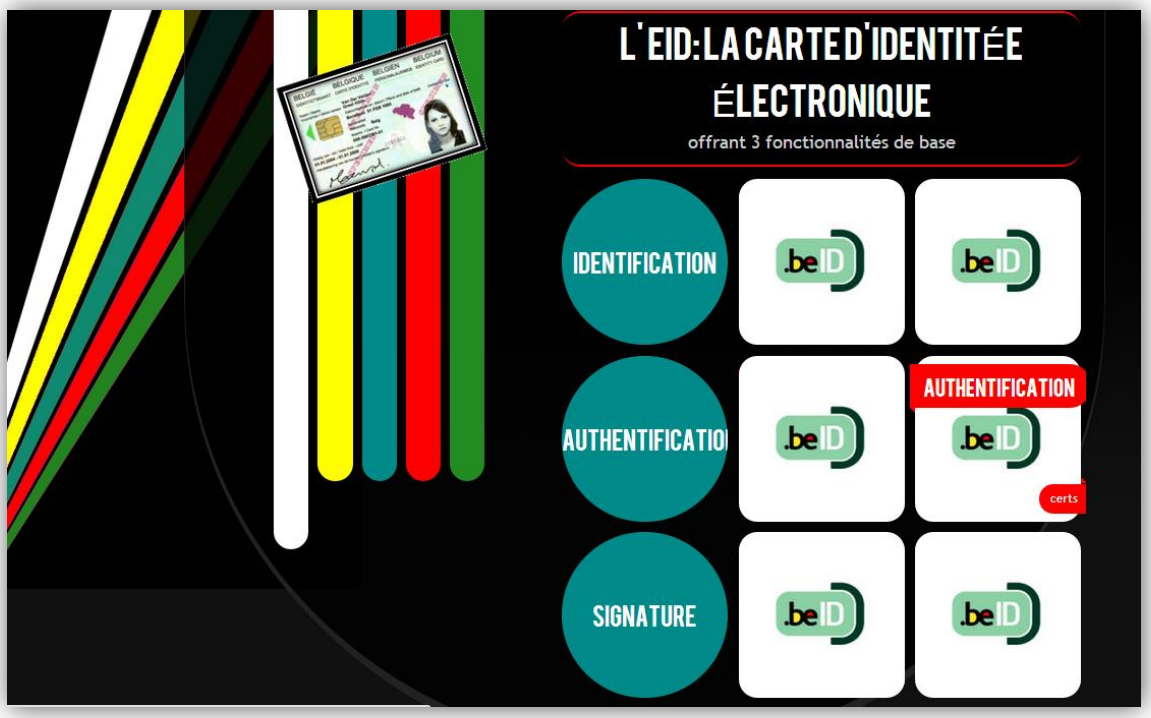


Figure 57 : Choix de service

Choisissons le service d'authentification-certs, l'utilisateur s'identifie puis prouve son identité par un code PIN qui est le mot de passe de la clé privée d'un certificat d'authentification. Comme résultat, nous allons retourner les

certificats qui existent dans notre carte et plus particulièrement celui de l'authentification.

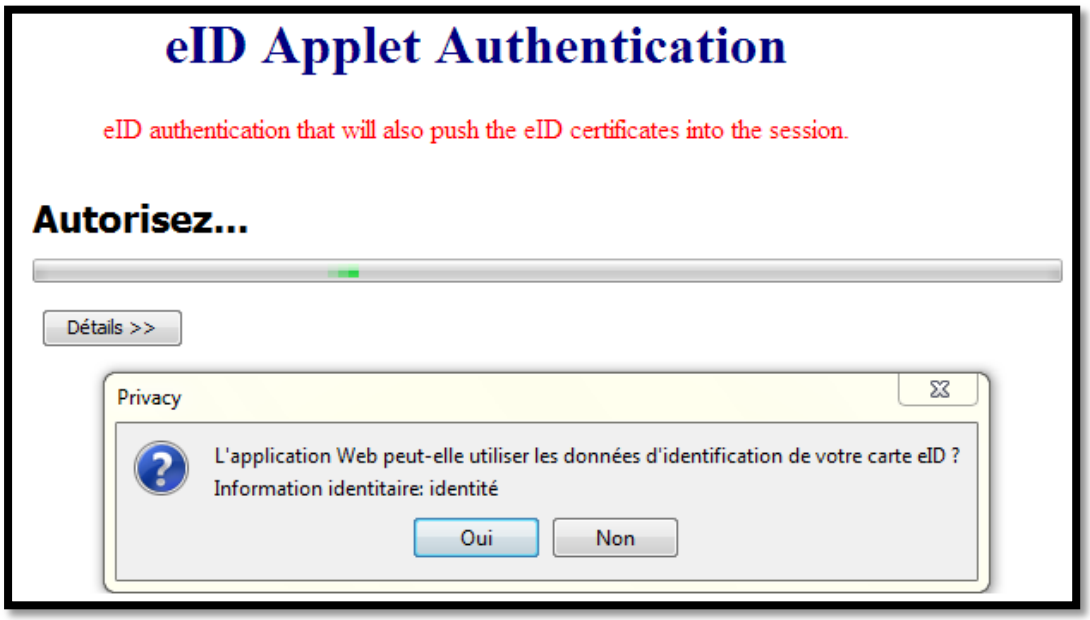


Figure 58 : Autoriser l'utilisation des données d'identification

Une fois que notre lecteur est bien branché et que la carte eID est présente, une validation d'utilisation des données d'identification dans l'application web est demandée.

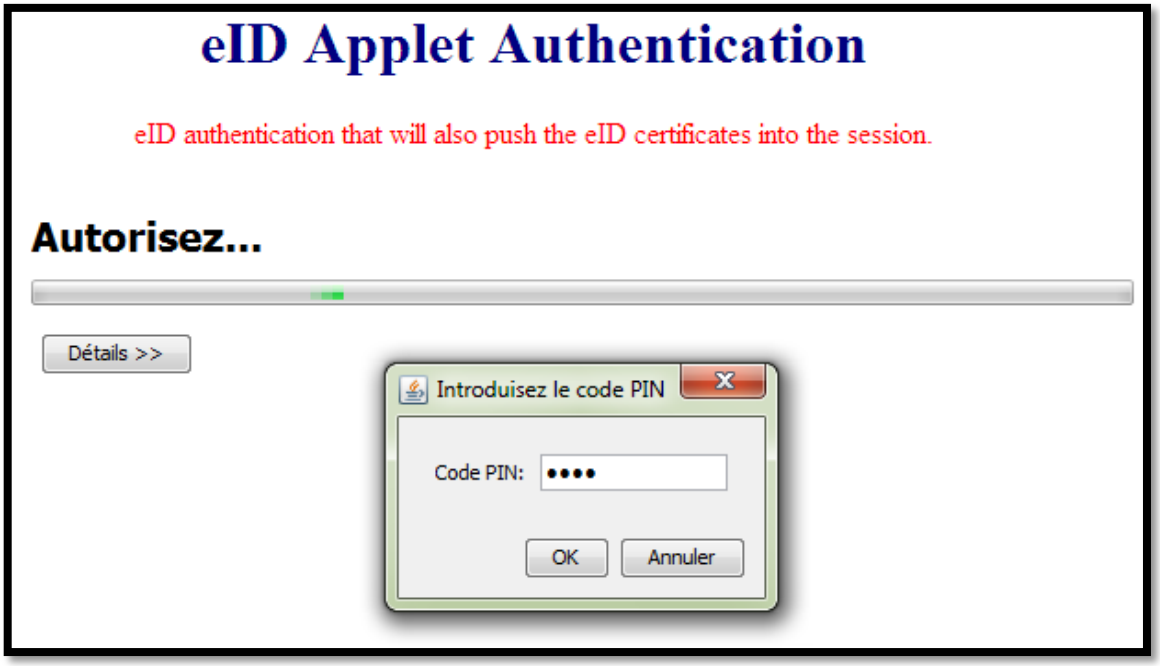


Figure 59 : Insertion du code PIN

Afin d'utiliser le certificat d'authentification, protégé par une clé privée, l'utilisateur doit présenter le mot de passe valide de cette dernière (PIN).

Authentication Result Page

Authenticated User Identifier: 71715100070

Certificates

Authentication Certificate:

```
[
[
  Version: V3
  Subject: SERIALNUMBER=71715100070, GIVENNAME=Alice Geldigekaart, SURNAME=SPECIMEN, CN=Alice SPECIMEN (Authentication), C=BE
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key:  Sun RSA public key, 1024 bits
  modulus: 128462547073996418214060203989525356091093087514983497971547156625302930762106389277316115449475652753204053887968197102264786415637313276874399913
  public exponent: 65537
  Validity: [From: Mon Jan 30 10:20:54 CET 2012,
              To: Mon Jan 30 10:20:54 CET 2017]
  Issuer: CN=eID test Citizen CA, C=BE
  SerialNumber: [    01000000 0001352d eac1ce00 00cec003]

Certificate Extensions: 6
[1]: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [
  [
   accessMethod: 1.3.6.1.5.5.7.48.2
   accessLocation: URIName: http://certs.dev.eid.belgium.be/root.crt,
   accessMethod: 1.3.6.1.5.5.7.48.1
   accessLocation: URIName: http://ocsp.dev.eid.belgium.be]
]

[2]: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
```

Figure 60 : Affichage de certificat d'authentification

L'authentification est réussie, l'identité de la personne est prouvée avec succès et une réorientation vers une page résultat est effectuée pour visualiser les détails du certificat d'authentification.

V.5.3 Authentification avec certificat –ANCE

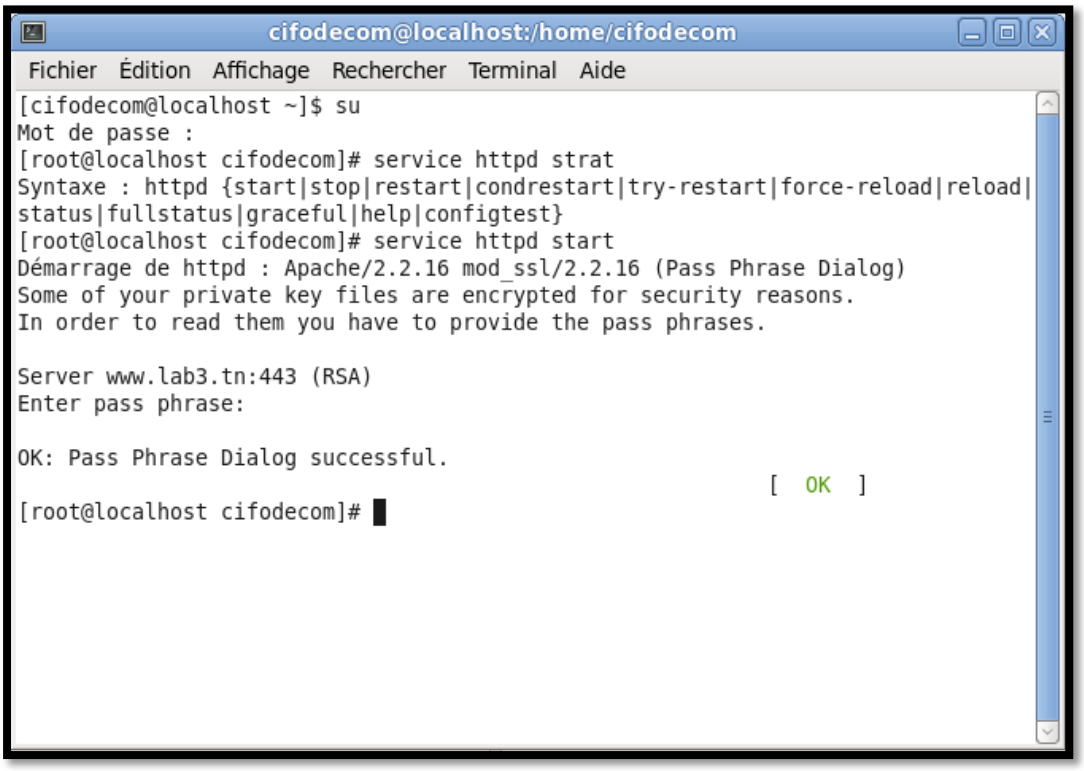


Figure 61 : Démarrage du service Apache sous fedora et vérification du mot de passe

Une fois que le certificat client est installé dans le navigateur, la vérification du mot de passe est demandée lors du démarrage du service Apache.

Figure 62 : Authentification par certificat

L'authentification par certificat est bien réussie.

V.6 Sprint 3 – Mise en place de la signature numérique.

Ce sprint a pour objectif de mettre en place la phase de signature. Pour pouvoir signer un message ou des données, l'utilisateur doit connecter sa carte à puce dans laquelle sa clé privée est confinée et protégée en confidentialité. L'application de création de signature calcule sur l'ordinateur

un condensat des données à signer et le transmet au dispositif de création de signature (la carte) par activation du code PIN.

Le calcul cryptographique générant la signature électronique des données à signer, est ainsi réalisé dans le dispositif de création de signature. Le condensat signé, intitulé signature électronique, est ensuite retourné à l'application.

La liste des figures suivantes récapitule l'exécution de service, d'une part pour une signature-simple d'un document textuel pur et d'autre part pour une signature au format XML permettant un traitement automatisé de certains types de fichiers (notamment PDF).

V.6.1 Signature-simple d'un document textuel pur

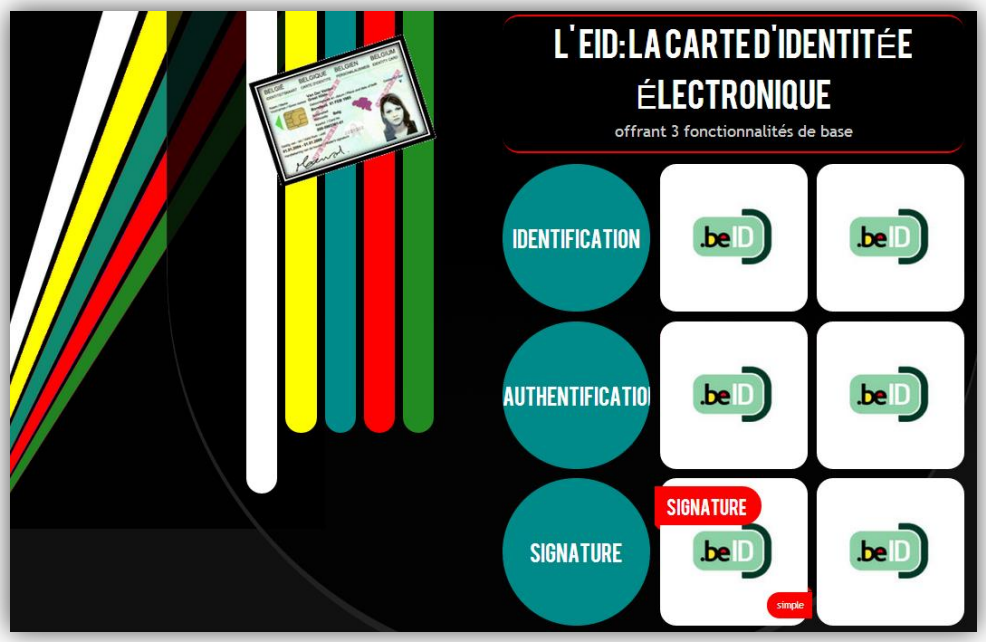


Figure 63: Choix de service

L'utilisateur choisit l'exécution du service de signature. Nous allons utiliser l'applet eID pour créer des signatures numériques à l'aide du certificat eID de non-répudiation. Les algorithmes de signature pris en charge sont : SHA-1, SHA-256, SHA-384, et SHA-512. Ici, la valeur de l'empreinte à signer provient uniquement de la mise en œuvre de composant de service Signature Service.

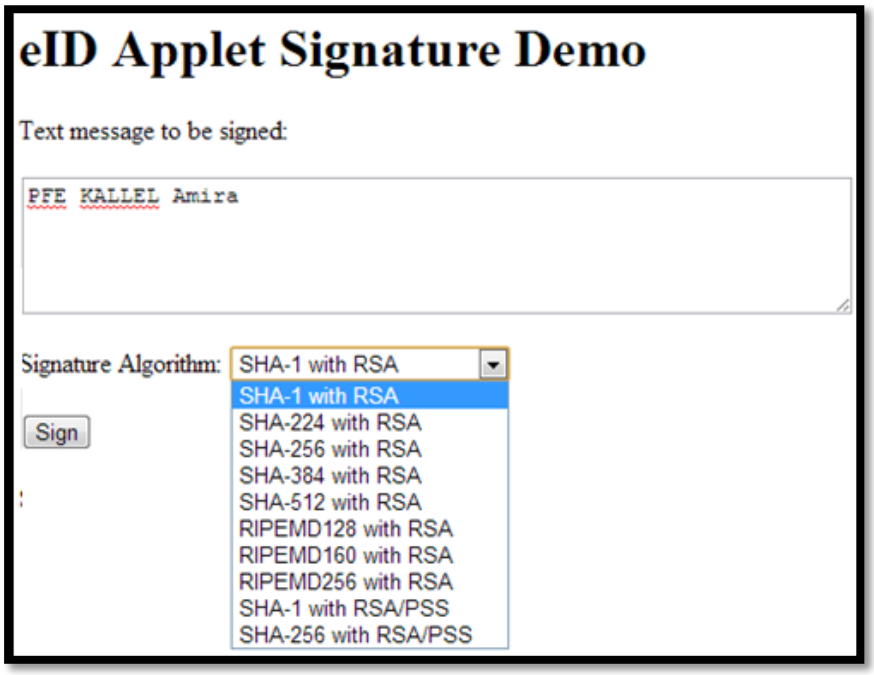


Figure 64: Saisie de texte et choix de l'algorithme de signature

Ici, nous allons saisir le texte et choisir l'algorithme de signature que nous allons lui appliquer

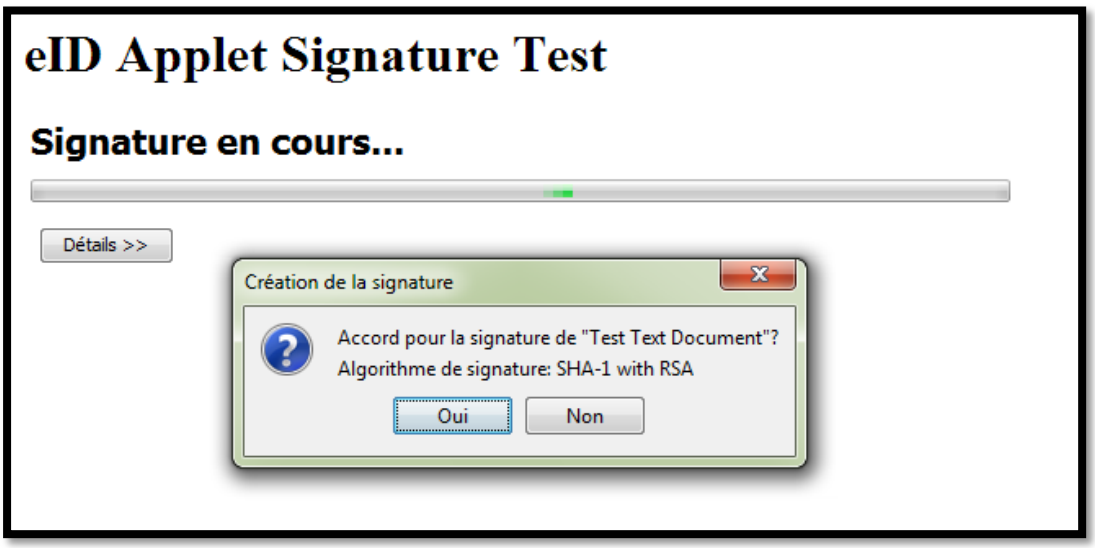


Figure 65 : Demander l'accord de la signature du texte

L'utilisateur est invité à confirmer qu'il est sûr de vouloir signer le bout du texte entré par l'algorithme choisi ou de refuser dans le cas contraire.

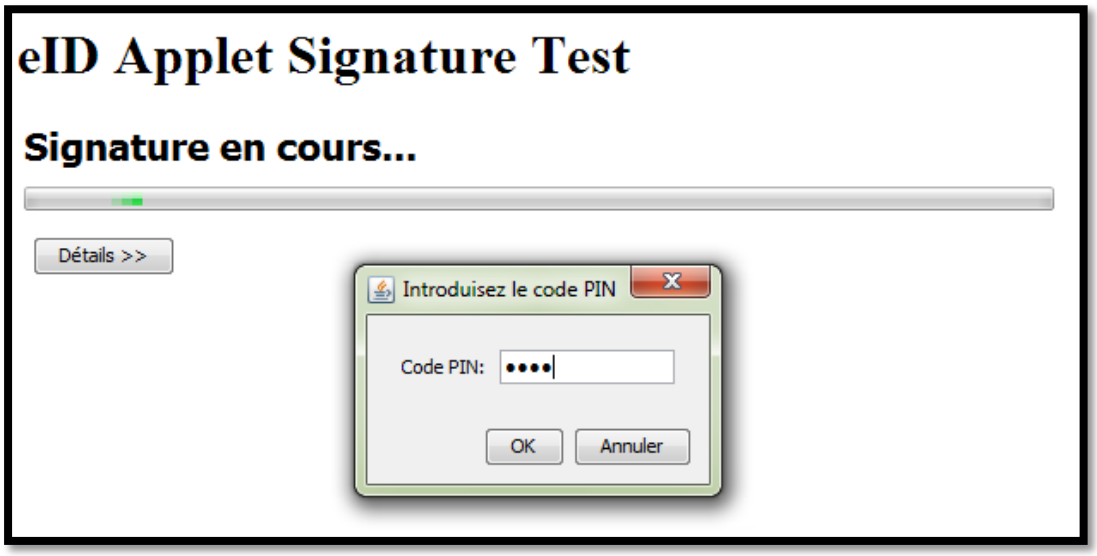


Figure 66 : Insertion du code PIN

Afin d'utiliser le certificat de signature, protégé par une clé privée, l'utilisateur doit présenter le mot de passe valide de cette dernière (clé).

Figure 67 : Signature avec succès et affichage du certificat de signature

La signature est créée avec succès. Au niveau de cette démonstration, nous affichons la valeur de la signature ainsi que la chaîne de certificat de signature.

V.6.2 Signature -XML

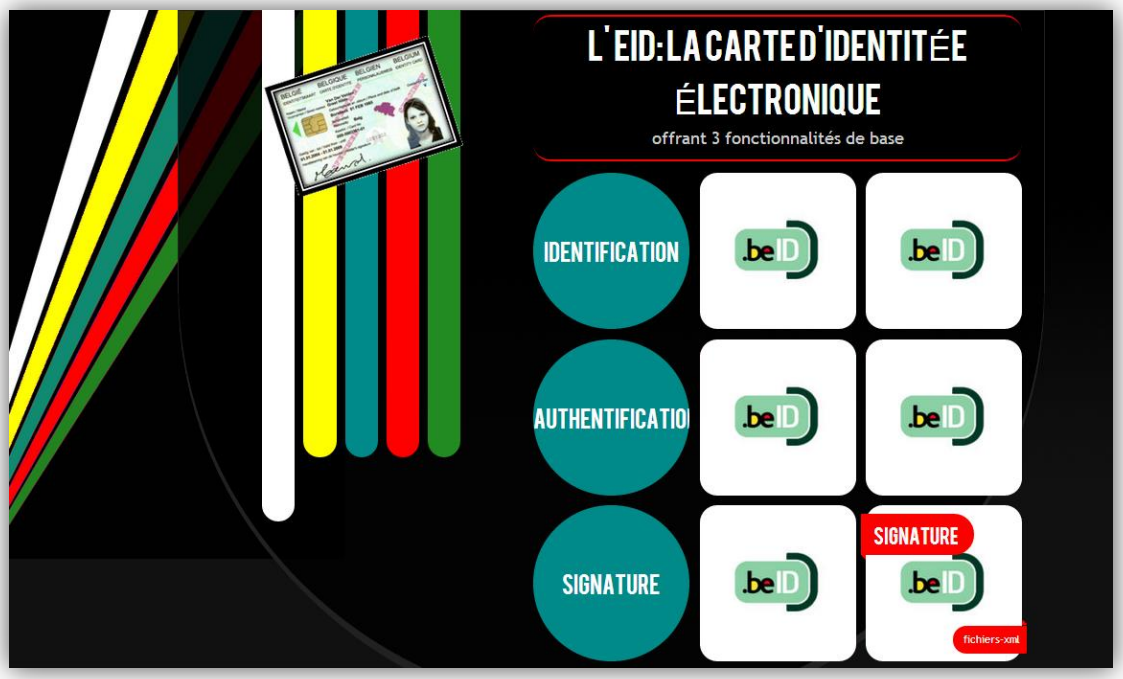


Figure 68 : Choix de service

L'applet eID envoie d'abord un ensemble de valeurs d'empreintes calculées à partir des fichiers locaux. Ces fichiers sont sélectionnés par le citoyen via une interface utilisateur de fichier de l'applet eID. Les algorithmes de hachage des fichiers pris en charge sont SHA-1, SHA-256, SHA-384 et SHA-512. Ce type d'opération de signature numérique peut être utilisé pour construire par exemple des signatures XML, les signatures XAdES ou les signatures PDF.

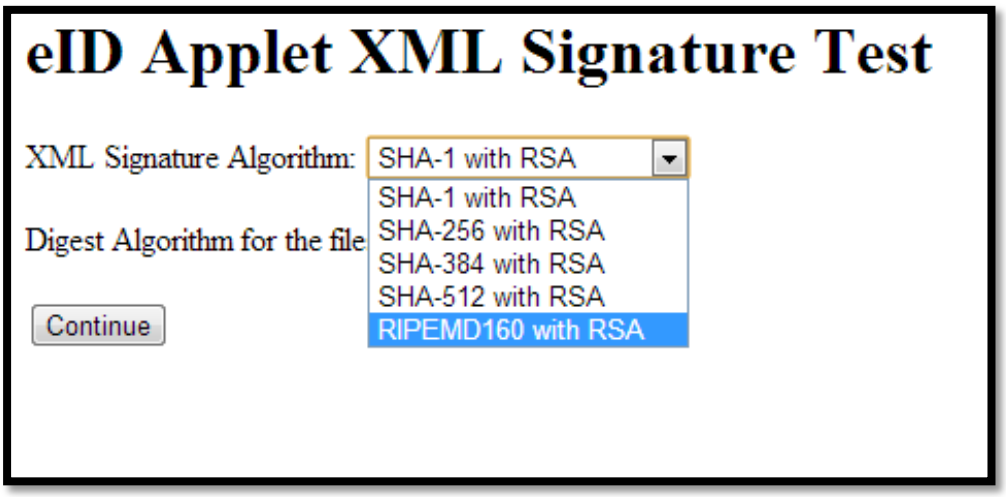


Figure 69 : Choix de l'algorithme de signature

Commençons par le choix de l'algorithme de signature XML.

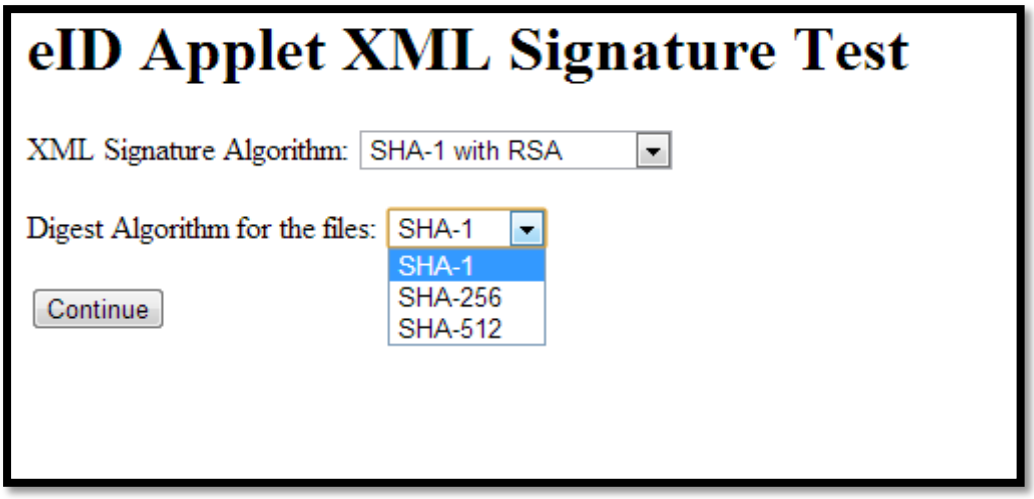


Figure 70 : Choix de l'algorithme de hachage

Ainsi que l'algorithme de hachage à appliquer.

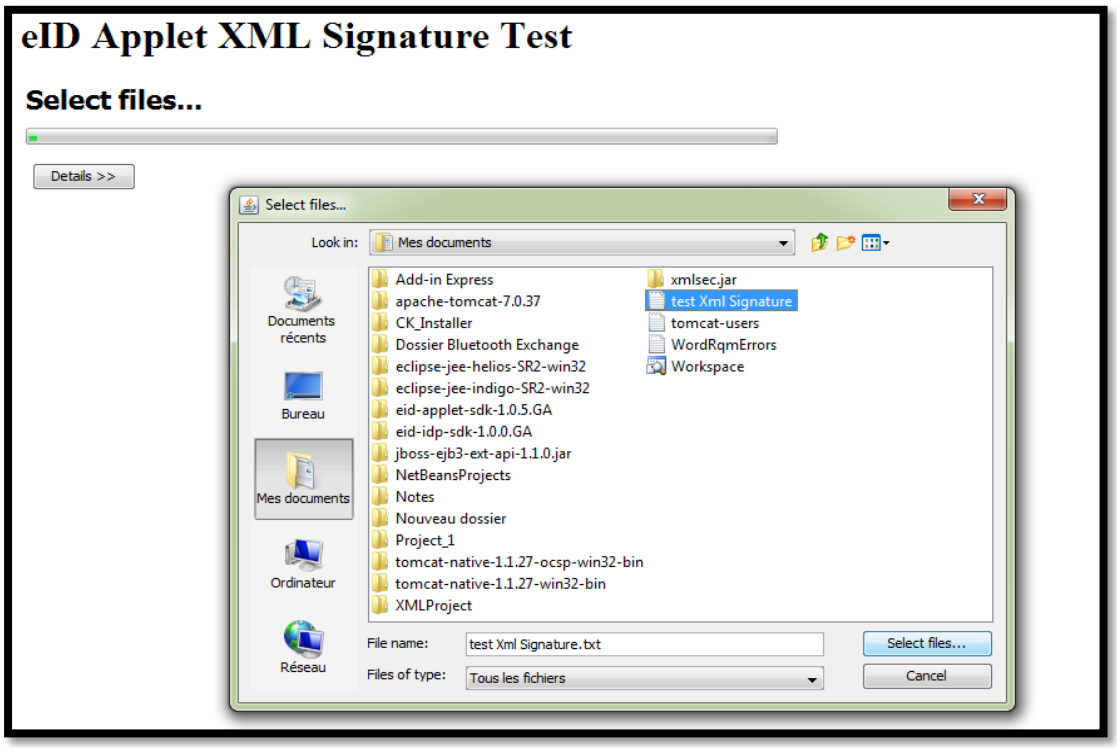


Figure 71 : Choix de fichier à signer

Ici, nous allons choisir le document local que nous souhaitons signer.

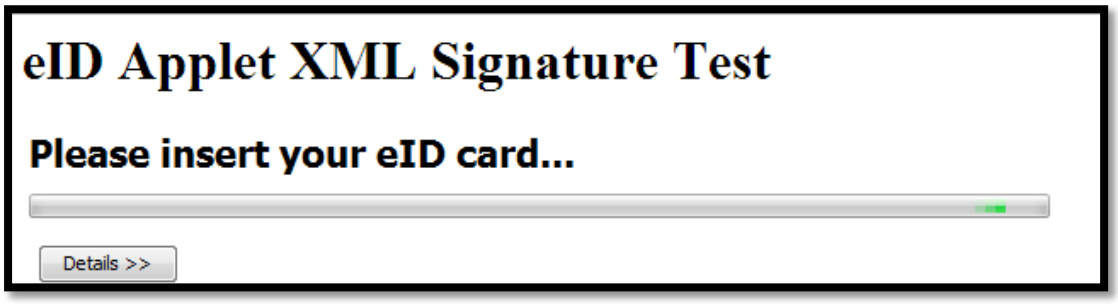


Figure 72 : Vérification de la présence de la carte

Une applet est lancée par le navigateur. Tout d'abord elle vérifie la présence de la carte eID.

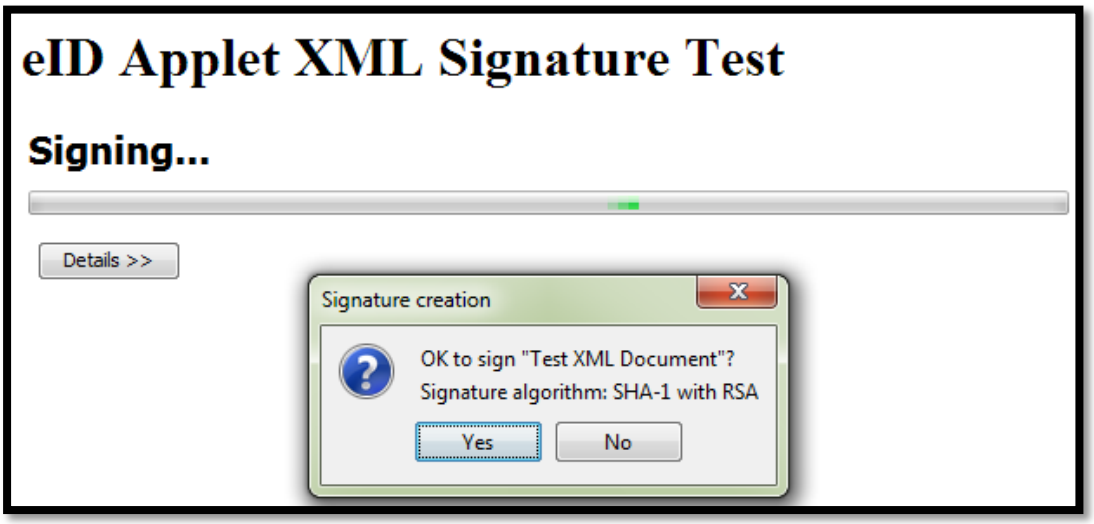


Figure 73 : Validation de signature du fichier

Ici, l'utilisateur est invité à confirmer qu'il est sûr de vouloir signer le fichier par l'algorithme choisi et de refuser dans le cas contraire.

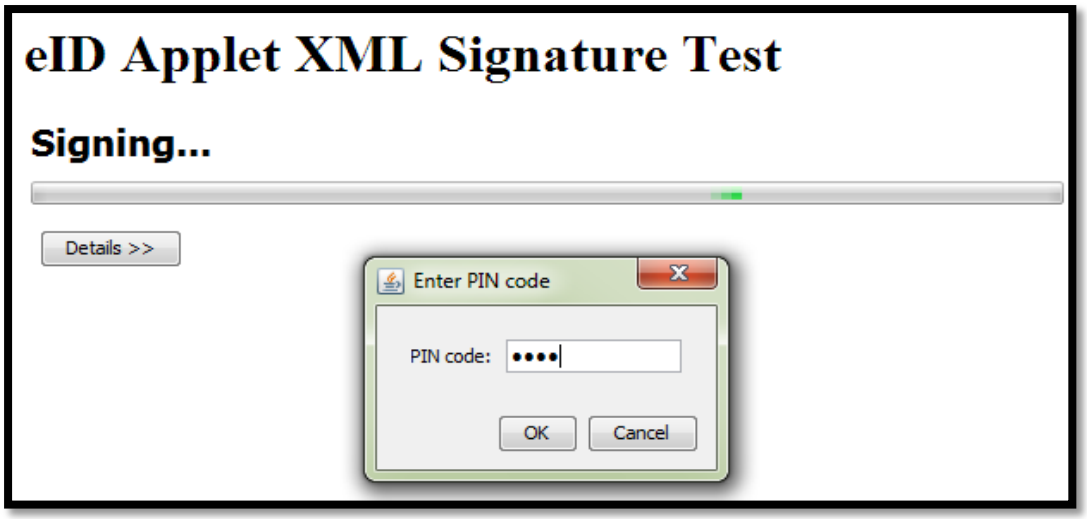


Figure 74 : Insertion code PIN

Afin d'utiliser le certificat de signature protégé par une clé privée, l'utilisateur doit présenter le mot de passe valide de cette dernière (PIN).

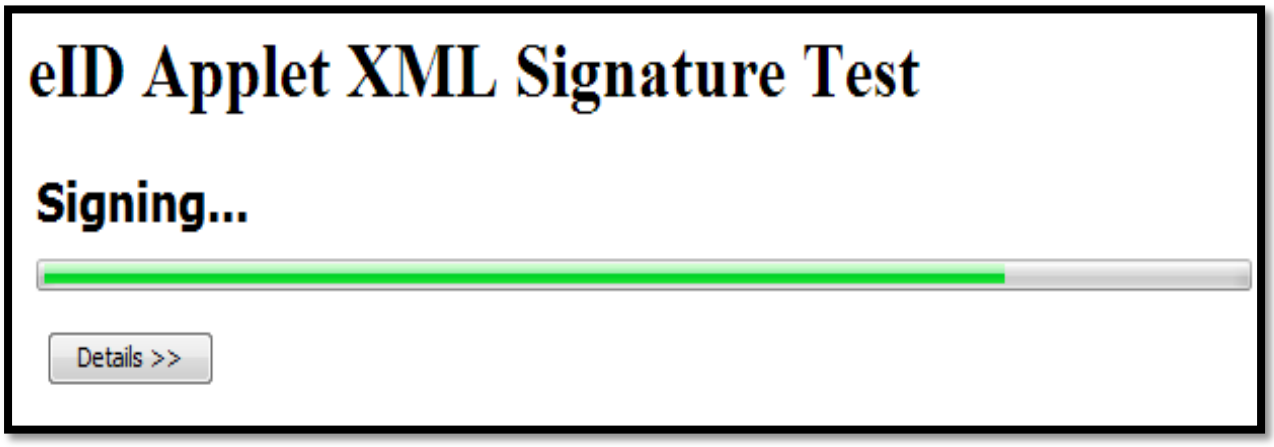


Figure 75 : Signature du fichier en cours

Le fichier est en cours de signature avec l'algorithme SHA-1.

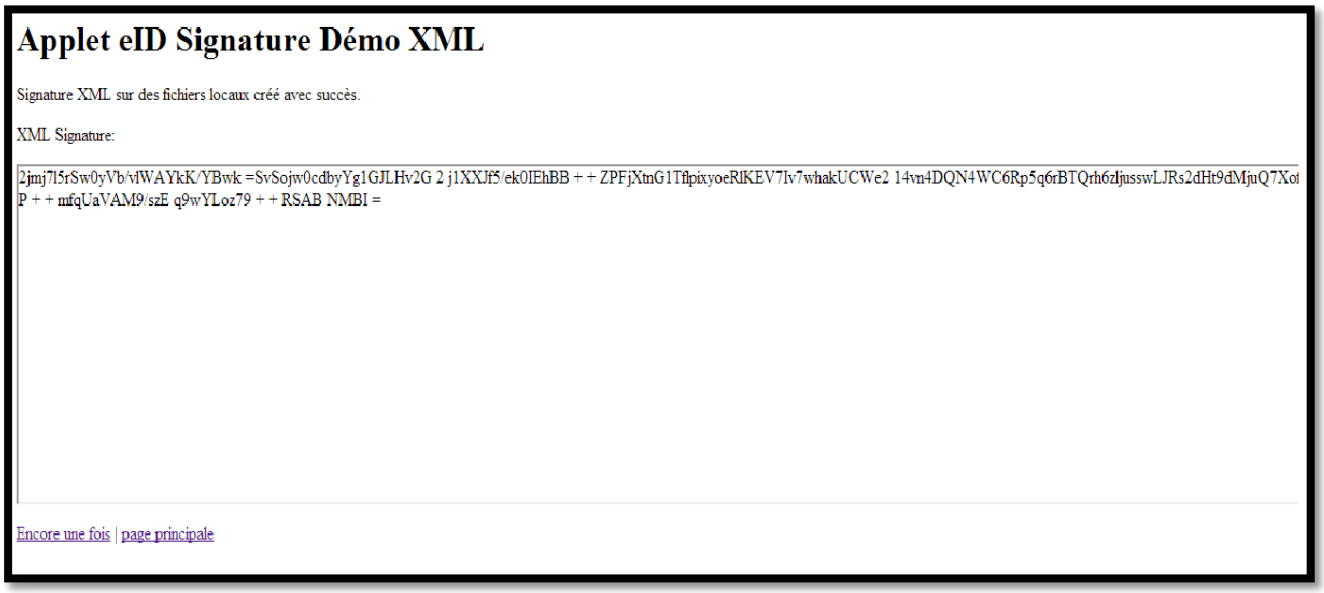


Figure 76 : Signature avec succès

Affichage de la signature XML suite à la signature réussie du fichier.

V.7 Sprint 4 - Intégration dans OpenMRS

Ce sprint a pour objectif l'intégration de ces fonctionnalités dans l'application complexe OpenMRS dont nous avons déjà présenté précédemment l'architecture en 3 couches, lors de l'étude de système existant.

En se référant à cette architecture, nous avons tout d'abord commencé par la mise en place de l'authentification. Au niveau de la page de login, nous avons remplacé la fonction d'authentification faible par login/mot de passe par l'applet eID connectée au serveur d'authentification.

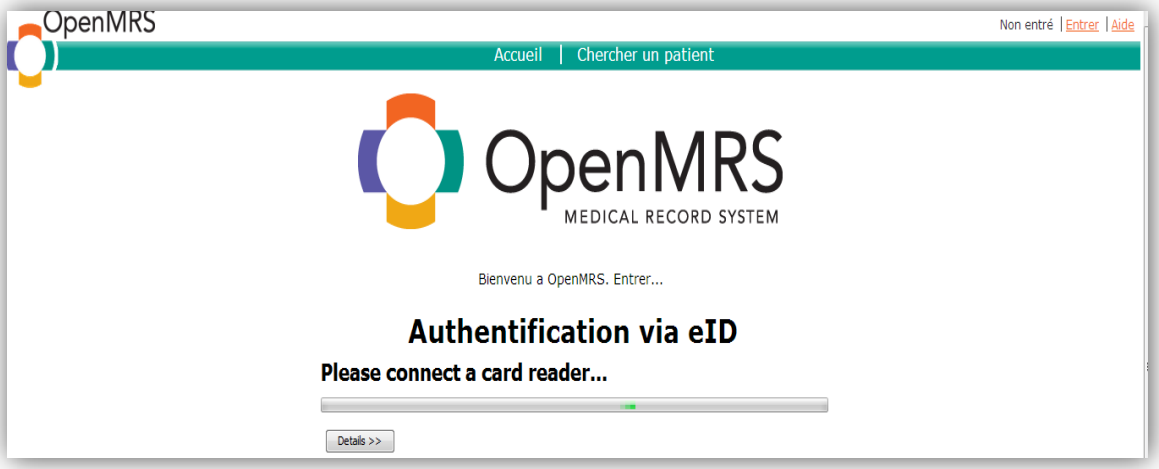


Figure 77 : Vérification du branchement de lecteur carte

Dans la page d'accueil d'OpenMRS une applet se lance pour authentifier l'utilisateur. Ici, une vérification du branchement de lecteur carte est demandée.

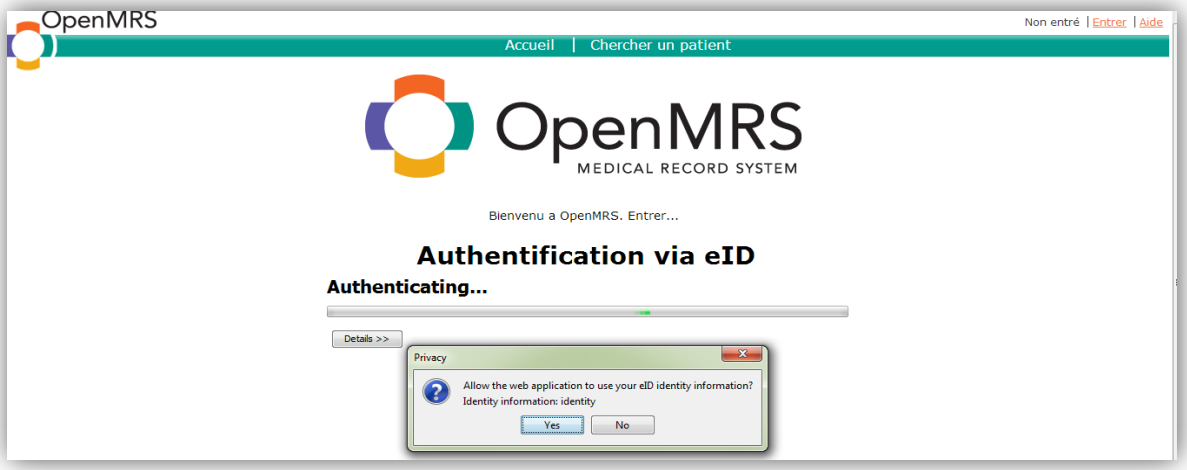


Figure 78 : Autoriser l'utilisation des données d'identification

Une fois que notre lecteur est branché et que notre carte est insérée, une autorisation est initiée.

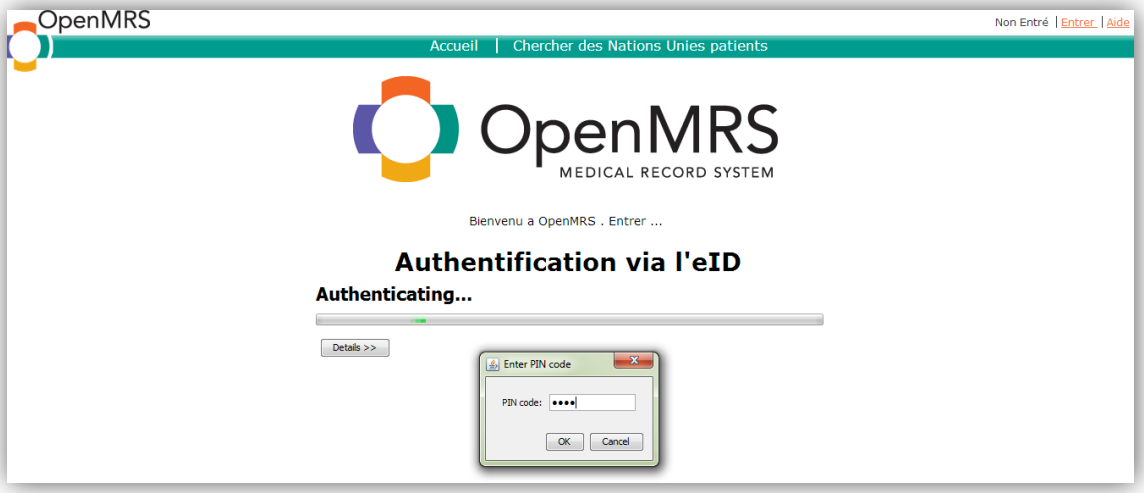


Figure 79 : Insertion du code PIN

Afin d'utiliser le certificat d'authentification, protégé par une clé privée, l'utilisateur doit présenter le mot de passe valide de cette dernière (PIN).

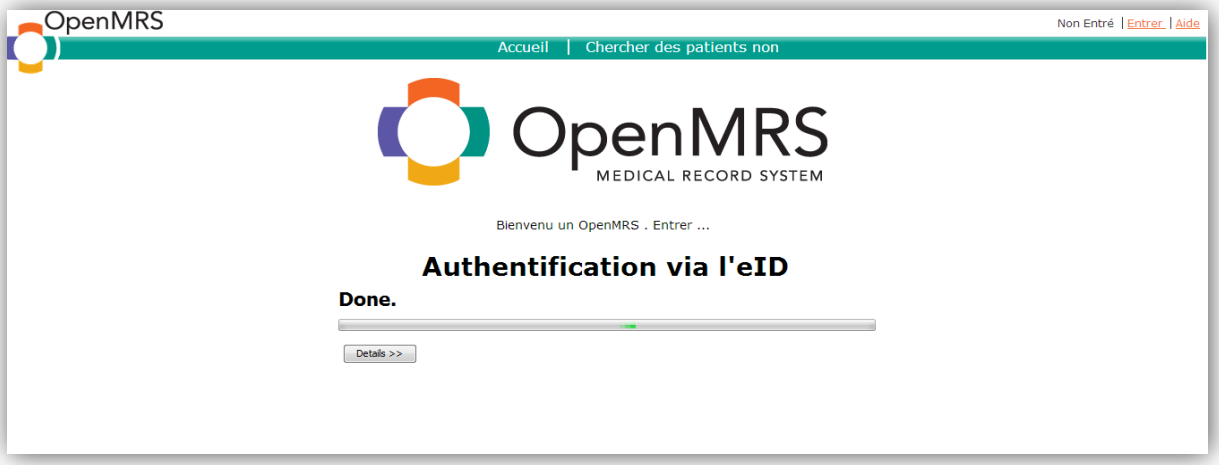


Figure 80 : Transmission avec succès

La transmission est terminée avec succès.

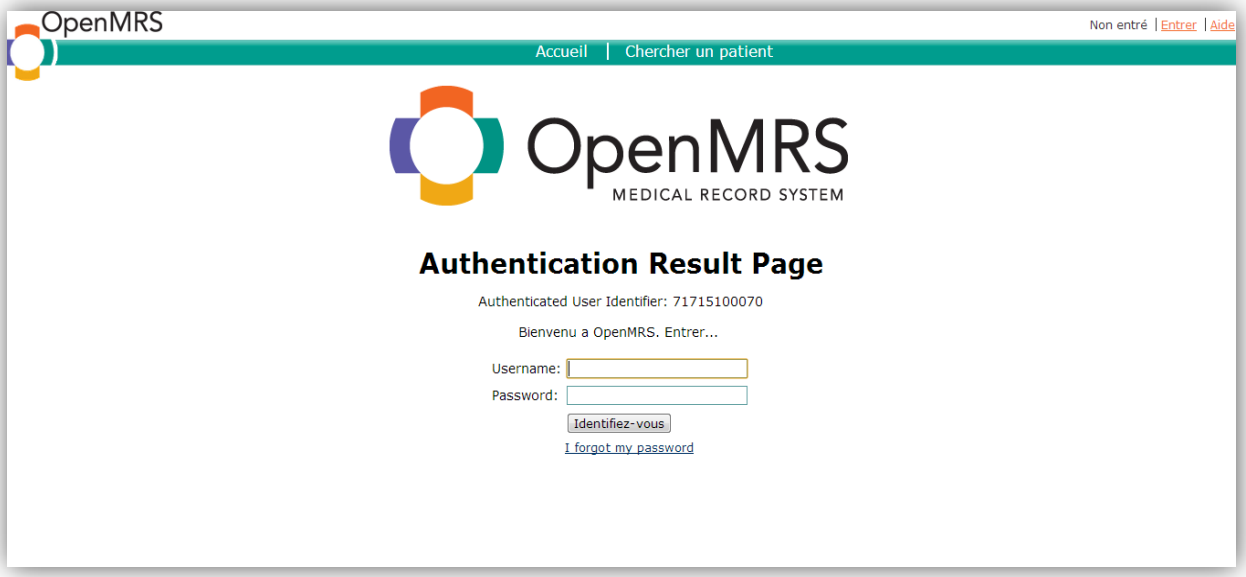


Figure 81 : Affichage de résultat d'authentification

La vérification du certificat d'authentification est réussie, l'utilisateur est bien authentifié. Au niveau de la démonstration, nous affichons le numéro de registre national. Dans le système réel, ce dernier est utilisé pour dériver un identifiant au niveau de la base de données.

V.8 Sprint 5 - Contrôle et amélioration de la qualité de code

Pour que notre produit fini soit de meilleure qualité possible, nous devons faire attention à la qualité de notre code source. Pour mesurer la qualité du code nous avons, en premier lieu, besoin de créer des indicateurs (nombre de lignes de code, nombre de commentaires, taux de couverture de code). En second lieu, nous avons besoin d'un outil analysant et synthétisant ces indicateurs. Sonar que nous avons présenté précédemment nous permet de faire cela de manière assez intuitive et ergonomique.

Il s'agit d'une aide technique : il permet de détecter les erreurs commises par les développeurs [26].

L'interface de Sonar se compose des écrans suivants :

- Le Dashboard technique qui affiche les indicateurs pour chacun des péchés.
- Le Dashboard de management « SQALE » qui calcule la dette technique (ce module est payant), une version simplifiée Open Source est également disponible.
- Le nuage « Cloud » de mots présente les noms des classes de l'ensemble du projet : les noms les plus gros et les plus rouges sont ceux à corriger en premier.
- La page « Design » affiche les dépendances cycliques entre packages.

- La page « Hotspot » priorise les maintenances à effectuer.
- La page « Time machine » permet de voir la progression de la qualité du code dans le temps.

Limitons-nous, dans ce sprint, à la visualisation du Dashboard technique. Voici les principaux péchés du développeur :

Duplication de code

De tous les péchés, la duplication de code est la plus dangereuse. En effet, il est difficile de maintenir deux codes similaires en parallèle. Il faut penser à apporter la même correction à deux endroits. De plus, la recopie d'une correction d'un endroit à l'autre peut être mal effectuée et entraîner un effet de bord qui n'aura pas forcément été testé.

C'est pourquoi, il est primordial de factoriser le plus possible le code. L'écriture de tests unitaires permet de définir le contrat actuel du code dupliqué. Nous factorisons ensuite ces deux codes. Les tests unitaires permettent d'indiquer que le fonctionnement précédent est conservé.

Mauvaise distribution de la complexité

La complexité dite « complexité cyclomatique » se mesure d'après le nombre de choix et de branches dans une méthode. C'est-à-dire le nombre de « if », « else », « for », « while », « switch / case » dans une méthode.

Il est conseillé d'avoir au maximum une complexité de 12 dans une méthode.

Plus la complexité d'une méthode est importante et plus elle sera jugée trop complexe et trop dangereuse à modifier. Une modification de ce type de méthodes présente un risque important d'effets de bord et de régression.

Pour éviter une complexité trop importante, les deux règles d'or suivantes sont à respecter :

- « Une méthode ou une classe n'est jamais trop petite »
- « Une et une seule responsabilité par classe. »

En effet, il est nécessaire de séparer les préoccupations : une classe n'est responsable que d'un traitement. Elle délègue les autres traitements à d'autres classes.

Mauvais design

Le design se mesure par les dépendances entre les couches de l'application. Ainsi les classes d'un package « A » ne doivent pas utiliser les classes d'un autre package « B » dans le cas où ces dernières utilisent les classes du package « A ». Il s'agirait alors d'un cycle de dépendance entre packages. Dans ce cas, les deux packages « A » et « B » sont interdépendants et ne pourraient exister ni être livrés séparément.

Manque de tests unitaires

Un test unitaire définit le contrat d'une méthode. Nous nous assurons de cette façon que la méthode répond bien à ce qui a été spécifié dans le test unitaire.

Ainsi, le test unitaire est le garant technique de la bonne tenue de l'application.

Manque de respect des standards

Un standard correspond par exemple à la gestion des exceptions, au formatage du code source.

Le respect de ces standards facilite la prise en main de la maintenance et de l'évolution de l'application par de nouveaux développeurs.

Bugs potentiels

Les bugs potentiels correspondent par exemple à la présence d'un risque de « NullPointerExpression » ou d'expressions ambiguës.

Ils sont détectés à l'aide de l'outil « FindBugs ». Intégré à Sonar, les résultats de l'analyse de « Findbugs » sont présentés d'une façon synthétique et hiérarchique.

Pas ou trop de commentaires

Il est conseillé de définir les commentaires sur les API publiques car elles permettent de décrire les intentions des classes et méthodes à une autre couche de l'application.

Cependant, trop de commentaires peuvent devenir contre-productifs car ils peuvent polluer le code et leur maintenance deviendrait coûteuse.

Il est plus important de bien nommer les méthodes et les variables, et de définir le contrat de chaque méthode à l'aide de tests unitaires.

Finalement, ce sont les tests unitaires qui commentent les méthodes en indiquant leurs intentions. Les tests unitaires deviennent ainsi la documentation technique du code source.

Dette technique

La dette technique est le coup de non qualité. Elle s'accumule chaque fois que vous faites quelque chose de la mauvaise façon en laissant la qualité du code se détériorer. Tout comme les dettes financières, agir ainsi est plus facile sur le court terme. Mais au fil du temps, les "intérêts" que vous payez sur cette dette deviennent énormes, jusqu'à atteindre un point où une réécriture complète de l'application devient plus facile que de maintenir ou de modifier le code existant.

Une fois que nous avons effectué l'intégration de Sonar dans notre serveur tomcat, nous procédons à l'analyse de nos produits, mais comment ?

1. installer le war dans le répertoire webApp de tomcat (par défaut une base de données embarquées est utilisée, pas très performant mais OK sinon il faut se connecter à un MySQL),
2. éditer le pom.xml et ajouter des prospérités, notamment l'URL de sonar

```
<sonar.host.url>http://localhost:8080/sonar</sonar.host.url>
<sonar.skipDesign>true</sonar.skipDesign>
<sonar.dynamicAnalysis>false</sonar.dynamicAnalysis>
```

   (les deux dernières lignes sont rédigées pour éviter certaines parties: les tests unitaires et l'analyse de couverture),

3. enfin, lancer via la commande maven : `mvn sonar:sonar`,
4. attendre la fin dans la console puis aller sur la page http://localhost:8080/sonar pour voir les résultats.

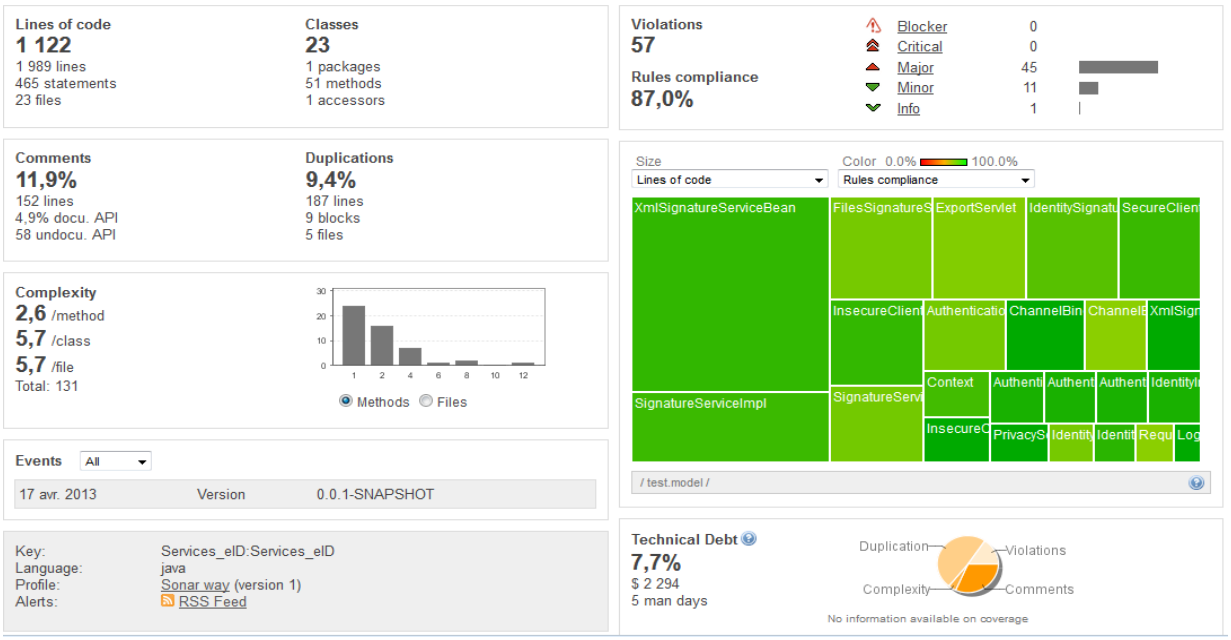


Figure 82 : Le Dashboard technique de eID-Services

Le prototype Services-eID comptait 1122 lignes de code, 57 violations et avait un respect élevé des règles de 87.0 % et homogène sur le code (voir le treemap entièrement dans le vert). La complexité cyclomatique pour les méthodes est (de 2.6 << 12) faible, donc pas dangereuse. La documentation est un peu faible. Le détail des violations montre qu'il s'agit du code Open Source non modifié et pas celui que nous avons développé.

La dette technique est de 7.7%, soit relativement faible d'autant plus que le code est de petite taille. Résorber la dette coûterait 5 jours de travail et est réalisable dans la durée d'un seul sprint. La duplication est le plus gros défaut mais il faut le relativiser car le code contient de la redondance naturelle dans la formulation des services.

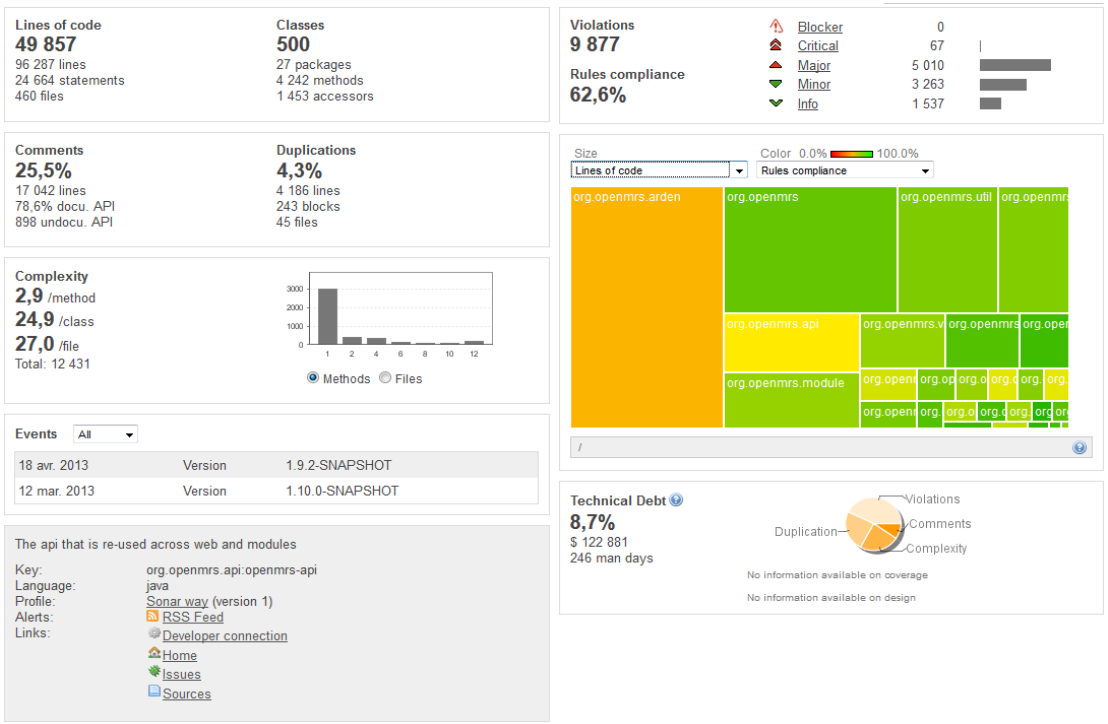


Figure 83 : Dashboard technique d'OpenMRS après l'intégration des fonctionnalités eID

La version utilisée d'OpenMRS compte presque 50.000 lignes de code. Les presque 10.000 violations représentent un niveau de respect des règles plus faible que pour les services eID, soit 62.6 %. Elle représente une grande partie de la dette qui se monte à 8.7%. Cette dette reste globalement acceptable mais n'est pas homogène : certains modules sont de moins bonne qualité et mériteraient une étude plus détaillée. Vu la taille du code, la dette représente aussi beaucoup plus de travail : 246 homme/jours de correction. Comme pour l'autre analyse, la complexité cyclomatique pour les méthodes est (de 2.9 << 12) faible, donc pas dangereuse. La documentation est par contre meilleure.

En conclusion, si les deux applications ont une dette technique relative assez similaire, leur coût de correction est très différent à cause de leur taille très différente. La répartition des problèmes est aussi fort différente. A noter que seul un sous-ensemble du code d'OpenMRS a dû être maîtrisé dans le cadre de ce travail.

V.9 Conclusion

Ce chapitre a illustré les travaux réalisés suivant les principales fonctionnalités de notre solution. Nous avons d'abord commencé par présenter les outils utilisés. Ensuite, nous avons présenté les différents sprints qui ont menés à l'obtention de notre produit final, en commençant par la réalisation d'un prototype présentant les fonctionnalités de base de l'eID, puis nous avons réalisé l'intégration de cette dernière dans OpenMRS. Enfin, nous avons testé la qualité de notre code avec Sonar.

Synthèse et conclusions

Avec le développement des réseaux informatiques à la fois au sein des hôpitaux mais aussi au domicile des patients, le domaine des applications eHealth est en pleine expansion. On ne compte plus les projets de dossiers médicaux informatisés. Les développements les plus récents consistent en une interconnexion croissante des intervenants de la chaîne de soin, y compris les patients eux-mêmes puisque des applications de suivi à domicile commencent à voir le jour.

Cette évolution pose bien le problème de la sécurité dans toutes ses dimensions : qu'il s'agisse de la confidentialité des données (protection de la vie privée et du secret médical), de leur intégrité (une donnée médicale altérée pouvant potentiellement conduire à des erreurs médicales) et de leur disponibilité (notamment d'un dossier médical partagé).

Dans la cadre de ce mémoire, nous nous sommes intéressés à une application de suivi de patient à domicile et en particulier d'un lien sensible à sécuriser : la liaison du docteur vers le portail afin d'en assurer le suivi à distance.

L'application belge utilisée est le projet eHealth. Dans le cadre du travail nous avons utilisé l'application OpenMRS qui est un équivalent Open Source pour les fonctionnalités considérées et qui permet de rendre disponible les résultats de ce mémoire de manière ouverte. La sécurisation exigée doit s'appuyer sur des mécanismes de sécurisation fiable, exploitant la PKI et sur des solutions disponibles sur le marché : notamment l'eID en Belgique et la carte ANCE en Tunisie.

Le travail s'est déroulé comme suit. Nous avons effectué tout d'abord une étude comparative des solutions eGov eId et ANCE. Ce travail s'est divisé en une partie théorique où nous avons présenté le contenu, la hiérarchie des certificats, les propriétés et les fonctionnalités offertes de chaque carte; suivie d'une partie pratique où nous avons manipulé les certificats électroniques et assuré l'authentification à travers ces derniers.

Ensuite, nous avons procédé à une étude comparative des systèmes eHealth et OpenMRS où nous avons exploré au début les fonctionnalités en les classant en deux catégories : la première, nous souhaitons l'améliorer dans eHealth et qui est déjà offerte OpenMRS ; tandis que la seconde présente celles qui sont souhaitées et non supportées. Puis, nous avons réalisé une analyse comparée de leurs modèles respectifs.

Sur cette base, nous avons élaboré un prototype des fonctions d'identification, d'authentification et de signature. La partie identification a consisté en la récupération des informations d'une personne. La partie d'authentification a mis en œuvre les mécanismes d'authentification forte proposés par la carte eID pour authentifier l'utilisateur. Elle est basée sur ce que l'utilisateur possède (la clé privée) aussi bien que sur ce que l'utilisateur sait (le mot de passe qui protège cette clé privée) afin d'éviter l'utilisation de l'authentification par login et mot de passe trop faible qui cause plusieurs risques d'attaques inacceptables dans le contexte médical considéré. La partie de signature visait à signer un message ou des données via le mécanisme d'insertion d'un hash codé par la clef privée ainsi que le certificat public.

Enfin, nous nous sommes intéressés à l'intégration de ces technologies dans le système OpenMRS servant de support à notre travail.

Ces diverses étapes ont été réalisées selon une démarche itérative de type Agile, divisée en cinq grandes étapes (sprint).

Au cours de la réalisation de ce projet nous avons bien sûr dû faire face à quelques limitations et difficultés. Le travail sur le code eHealth s'est avéré difficile à distance pour des raisons de confidentialité, c'est une des raisons pour laquelle le travail sur un équivalent Open Source a été réalisé.

La phase d'intégration sur un projet de grande envergure tel qu'OpenMRS a également demandé un investissement important au niveau de la mise en place des outils, de l'appropriation du code, un préalable est nécessaire à toute modification. Malgré qu'un sprint fût dédié à ceci, plusieurs difficultés d'intégration ont dû être surmontées.

Au final, le travail produit a pu démontrer la faisabilité de l'approche de sécurisation. En particulier l'architecture retenue est particulièrement modulaire puisqu'elle s'intègre sous forme d'une portlet très légère dans l'application OpenMRS et que la majeure partie du service est déployée sur un service séparé. Celui-ci peut d'ailleurs lui-même être géré par un tiers de confiance et concrétiser ainsi la vision de la sécurité comme service « security as a service ».

Le résultat de ce travail reste un prototype qui devrait être consolidé dans plusieurs directions afin de pouvoir être utilisé de manière opérationnelle, ou même être diffusé comme module du projet OpenMRS. Tout d'abord au niveau des tests, bien que l'on dispose d'une carte de test pouvant émuler de nombreuses configurations (carte adulte/enfant/étrangers valides/invalides),

les batteries de test ont été limitées et devraient par conséquent être enrichies et automatisées. Ensuite, les cartes eID et ANCE ne sont pas totalement fonctionnellement équivalentes et interchangeables. Une API unique d'abstraction n'a pas été développée mais pourrait l'être, en particulier sur base de la récente norme ISO/IEC 24727 traitant de l'interopérabilité des cartes eID. Au niveau du packaging du code, on pourrait aussi imaginer de créer des modules spécifiques (par exemple ANCE, eID ou autre) implémentant cette API et qui pourraient être « pluggés » dans OpenMRS afin de le sécuriser via le module adéquat, selon le contexte. Dans l'optique « Security as a service », on pourrait aussi imaginer une architecture ANCE sur le modèle des services eContract belge.

Bibliographie

[1] 1, «Sécurité et Confidentialité des Systèmes d'Information,».

Available: http://www.lecissbretagne.org/IMG/pdf/110517_SanteNumeriqueEtSecurite_TribuneCS-p35-36_1_.pdf.

[2] «protection de la vie privée,». Available: http://www.ejustice.just.fgov.be/cgi_loi /change_lg.pl?language=fr&la=F&table_name=loi&cn=1992120832.

[3] «presentation-de-l-ance,».Available: http://www.certification.tn/fr/content/presentation-de-l-ance.

[4] «Le-CETIC-en-quelques-mots,». Available: http://www.cetic.be/Le-CETIC-en-quelques-mots.

[5] «Ingenierie-de-la-securite,». Available: http://www.cetic.be/Ingenierie-de-la-securite.

[6] «CETIC_DossierPresse_Innovation_eSante,». Available: http://www.cetic.be/IMG/pdf /CETIC_DossierPresse_Innovation_eSante_Charleroi_09062011.pdf.

[7] «espace patient,». Available: https://www.abrumet.be/fr-fr/espacepatient/default.aspx.

[8] Sécuriser ses échanges électroniques avec une PKI, 2002.

[9] «pki,» [. Available: http://www.securiteinfo.com/crypto/pki.shtml.

[10] «Authentification_de_A_a_Z,». Available: http://www.thehackademy.net/madchat/sysadm /unix.seku/Authentification_de_A_a_Z.pdf.

[11] «lauthentification-forte-concept-et-technologies-,». Available: http://webcache.googleusercontent.com/search?q=cache:http://fr.slideshare.net/ifall/ lauthentification-forte-concept-et-technologies-12078001.

[12] «Biométrie,». Available: http://www.ossir.org/paris/supports/2009/2009-10-13/ Sylvain_Maret_Biometrie.pdf.

[13] «cryptographie,». Available: http://groups.csail.mit.edu/cis/crypto/classes/6.857/ papers/gummy-slides.pdf.

[14] 8, «Etude comparée des méthodes de gestion des risques».

[15] 7, «http://www.enigma-services.fr/pres/PRES_ENIGMA_Methodes%20de%
20Gestion%20des%20Risques%20SSI_Avril2010_V1.0.pdf,».

[16] 7, «OpenMRS- Documentation techniques,». Available: https://wiki.openmrs.org/
display/docs/Technical+Overview.

[17] «wiki-openmrs,». Available: https://wiki.openmrs.org/display/docs/Home.

[18] «Eclipse,». Available: http://fr.wikipedia.org/wiki/Eclipse_(logiciel).

[19] «Maven 2,». Available: http://dcabasson.developpez.com/articles/java/maven
/introduction-maven2/.

[20] «Subversion,». Available: http://fr.wikipedia.org/wiki/Apache_Subversion.

[21] «Jetty,». Available: http://www.eclipse.org/jetty/documentation/current/jetty-maven-plugin.html.

[22] «Apache,». Available: http://www.techno-science.net/?onglet=glossaire&definition=790.

[23] «Mysql,». Available: http://fr.wikipedia.org/wiki/MySQL.

[24] «classic-client-software,». Available: http://www.nis-infor.com/fr/p/logiciels-cartes-a-puce
/idgo-300-classic-client-software.

[25] «Sonar,». Available: http://fr.wikipedia.org/wiki/Sonar_(logiciel).

[26] «Patient-Numerique-WP-v40.pdf,». Available: http://www.patientnumerique.com/media/
Patient-Numerique-WP-v40.pdf.

[27] O. [5]Derrouazi, «Étude comparée des méthodes de gestion des risques,» 2009.. Available:
http://www.cases.public.lu/fr/publications/recherche/r2sic/wp11_2_2009.pdf.

[28] 8, «La gestion des risques pour les systèmes d'information,». Available:
http://www.nmayer.eu/publis
/NMA-JPH_MISC24.pdf.

[29] L. g. d. r. p. l. s. d'information, «1,» Avril-Mai 2006),. Available: http://www.google.tn
/url?sa=t&rct=j&q=la%20s%C3%A9curit%C3%A9%20des%20syst%C3%A8mes%20d%E2%80%99inf
ormation
%20en%20entreprise&source=web&cd=1&ved=0CGwQFjAA&url=http%3A%2F%2Fwww.ibimapubl

ishing.com

%2Fjournals%2FCIBIMA%2Fvolume9%2Fv9n19.pdf&ei=qlrsT7i-GMzP4.

[30] «eHealth,». Available: http://www.cetic.be/eHealth-for-Citizens.

[31] «Middleware Client Classic,». Available: http://www.nis-infor.com/fr/p/logiciels-cartes-a-puce /idgo-300-classic-client-software.

[32] Available: http://www.banque-finance.ch/numeros/88/59.pdf.

[33] Scrum et XP depuis les Tranchées (Comment nous appliquons Scrum), C4MEDIA Inc.

[34] «scrum-en-moins-de-10-minutes,». Available: http://www.thierry-pigot.fr/gestion-de-projet/292/ scrum-en-moins-de-10-minutes.html.

Annexes A

Cas d'utilisation	Supportée	Open MRS
UC 1 : Connexion du médecin traitant au système **(Authentification via son token hardware)**	Non	Authentification par login/mdp
Un médecin traitant a accès à : **Profile user**	Oui	Un médecin traitant est une personne dans le système qui possède des rôles (ensemble de privilèges) qu'on peut les personnalisés (gestions des accès utilisateurs)
Profile de ses patients	Oui	Association entre personnes (ajout de relations dans le profil d'un utilisateur)
Liste d'alertes relatives à ses patients	Oui/Non	Gestion des alertes (spécifier la destination d'une alerte : nom médecin traitant)
le médecin traitant est loggé sur le système		-----

rejet d'authentification	Non Non	------
UC2 - Accès à un patient **(le médecin tente d'accéder à un dossier patient)** **le médecin est affecté au patient** **l'accès est loggé**	Oui Oui Non	Le médecin peut consulter le Tableau de Bord du patient. Parmi les caractéristiques qu'on peut visualiser (programmes (traitement) , concept (pathologie) , allergies , visites , …etc) Relation Docteur / Patient ---- Vérification des modules
UC3 - Traitement d'une alarme générée **(le médecin consulte une alarme générée par le système)** **rejet justifié de l'alarme, demande de suivi médical non urgent, demande d'intervention...**	Oui /Non	Une notification d'alerte système (traitement de l'alerte n'est pas mis en œuvre !! à vérifier par une démo)
UC4 - Encodage d'une alarme par le médecin traitant **(le médecin consulte le dossier et décide de lever une alarme lui-même)**	Non	------

UC5 - Export RSW(**génération d'un fichier de sommaire patient pour le RSW à une date donnée)**	Oui/Non	On peut générer un rapport du patient qui récapitule ses visites (ensemble d'observation, le programme, …etc)
UC6 - Déconnexion du système	Oui	Oui

Etude comparative entre MERISE et UML :

MERISE (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise) est une méthode d'analyse et de réalisation des systèmes d'information qui est élaborée en plusieurs étapes: schéma directeur, étude préalable, étude détaillée et la réalisation.

Alors qu'**UML (Unified Modeling Langage**), est un langage de modélisation des systèmes standard, qui utilise des diagrammes pour représenter chaque aspect d'un système exemple : statique, dynamique,....en s'appuyant sur la notion d'orienté objet qui est un véritable atout pour ce langage.

Merise ou UML ?

Les "méthodologues" disent qu'une méthode, pour être opérationnelle, doit avoir 3 composantes:

- une démarche (les étapes, phases et tâches de mise en œuvre),
- des formalismes (les modélisations et les techniques de transformation),
- une organisation et des moyens de mise en œuvre.

Merise s'est attachée, en son temps, à proposer un ensemble "cohérent" sur ces trois composantes. Certaines ont vieilli et ont dû être réactualisées (la démarche), d'autre "tiennent encore la route" (les modélisations).

UML se positionne exclusivement comme un ensemble de formalismes. Il faut y associer une démarche et une organisation pour constituer une méthode.

Merise se positionne comme une méthode de conception de SI organisationnel, plus tournée vers la compréhension et la formalisation des besoins du métier que vers la réalisation de logiciel. En ce sens, Merise se réclame plus de l'ingénierie du SI métier que du génie logiciel. Jamais Merise ne s'est voulu une méthode de développement de logiciels ni de programmation.

UML, de par son origine (la programmation objet) s'affirme comme un ensemble de formalismes pour la conception de logiciel à base de langage objet.

MERISE est encore tout à fait valable pour:

- la modélisation des données en vue de la construction d'une base de données relationnelles, la modélisation des processus métiers d'un SI automatisé en partie par du logiciel,
- la formalisation des besoins des utilisateurs dans le cadre de cahier des charges des utilisateur, en vue de la conception d'un logiciel adapté.

UML est idéal pour :

- concevoir et déployer une architecture logicielle développée dans un langage objet (Java, C++, VB.net). Certes, UML dans sa volonté "unificatrice" a proposé des formalismes,
- pour modéliser les données (le modèle de classe réduit sans méthode et stéréotypé en entités), mais avec des lacunes que ne présentait pas l'entité relation de Merise,
- pour modéliser le fonctionnement métier (le diagramme d'activité et de cas d'utilisation) qui sont des formalismes très anciens qu'avait, en son temps, amélioré Merise...

Après cette étude comparative, il est certain que je vais adopter UML comme langage de modélisation puisque je vais utiliser le concept de l'orienté objet ainsi que JAVA comme langage.

Dans ce travail, on va essayer de développer deux vues (trois diagrammes)

- **Vue statique :**
  - diagramme de cas d'utilisation : décrit les besoins utilisateurs.
  - diagramme des classes : définit la structure statique.
- **Vue dynamique** :
  - diagramme de séquence : scénarios et flots de messages

Annexe B

Gestion des risques de la sécurité de l'information

Pour bien appréhender la gestion des risques, ses objectifs et ses limites, il est nécessaire de comprendre en premier lieu les concepts sous-jacents et le processus employé.

Concepts de base et définitions

Les concepts de méthode de sécurité et de norme de sécurité portent souvent à confusion. Nous allons tenter de définir chacun de ces concepts et de préciser la différence qui existe entre eux [14].

❖ La norme

Une norme est un standard à suivre. Les entreprises se font certifier pour prouver qu'elles suivent les recommandations de la norme. Pour être certifiés, il faut, dans un premier temps acheter la norme. Les normes appliquées à la sécurité des systèmes d'information sont généralement éditées par l'organisme ISO (Organisme International de Normalisation). Ensuite, l'entreprise doit mettre en pratique les recommandations décrites dans la norme. Généralement, celle-ci va faire appel à un consultant spécialisé pour l'aider dans la mise en place de la norme. Enfin, une fois qu'elle applique les recommandations, elle peut faire appel à un organisme certificateur qu'elle va payer pour qu'il vérifie si elle respecte bien les recommandations émises par la norme. Si oui, l'entreprise est alors certifiée pour une durée déterminée, et elle devra se rendre disponible pour des audits (écoutes) durant sa période de certification. À la fin de sa période de certification, elle peut se refaire certifier de nouveau.

Une entreprise peut se faire certifier pour trois raisons :

- Pour une raison commerciale, pour certaines entreprises, être certifiées par des normes de qualité par exemple est un gage de qualité pour les clients et est donc un atout commercial.
- Par obligation. En industrie aéronautique par exemple, les constructeurs exigent de leurs sous-traitants qu'ils soient certifiés par certaines normes.
- Il y a aussi des entreprises qui se certifient pour elles-mêmes, pour optimiser leur processus en interne.

Généralement ces entreprises ne se font certifier qu'une fois, et lorsque les procédures de travail normalisées sont bien intégrées dans l'entreprise, elles ne renouvellent pas la certification qui est parfois onéreuse, mais gardent les nouvelles méthodes de travail.

❖ La méthode

Une méthode est une démarche, un processus ou un ensemble de principes qui permettent d'appliquer une norme au système d'information de l'entreprise.
La méthode sert aussi à faire un audit (une écoute) qui permet de faire, par exemple, un état de la sécurité du système d'information. Une méthode est souvent accompagnée d'outils afin d'appuyer son utilisation. Ils peuvent être disponibles gratuitement auprès des organismes qui les ont produits. Par exemple la méthode MEHARI, que nous verrons plus loin, propose un outil (fichier Microsoft Excel). Le fichier contient un ensemble de questions et de scénarios.
Cette base de connaissance permet de ressortir toutes les vulnérabilités du système d'information et émet des recommandations pour y remédier.
La plupart des méthodes sont appliquées par des experts en gestion des risques (EBIOS, MEHARI). Par contre, peu de méthodes (OCTAVE) peuvent être utilisées par des personnes spécifiques de l'entreprise.

❖ Norme versus méthode

Une norme fournit des recommandations à suivre par une entreprise avant de mettre en place une politique de certification. Ensuite l'entreprise peut prouver avec sa certification qu'elle suit toutes les recommandations présentes dans la norme. La méthode est une démarche qui permet d'arriver efficacement à un résultat souhaité.
Elle commence par faire un état des lieux (ou diagnostic) du système d'information de l'entreprise, et en fonction des résultats, fournit des préconisations adaptées, en conséquence. Ainsi, on ne peut pas opposer norme et méthode mais plutôt les associer et considérer une méthode comme un outil utilisé pour satisfaire à une norme. Ainsi pour mettre en œuvre efficacement la norme ISO/IEC 27005, il faut s'appuyer sur une méthode de gestion des risques telle que MEHARI, EBIOS, OCTAVE /OCTAVE-S.

❖ Définitions

Actif (Asset): élément du SI qui a de la valeur pour l'organisation (données papier/informatique, personnel, infrastructure, …).
Menace (Threat) : évènement qui peut mettre en défaut la sécurité d'un actif.
Vulnérabilité (Vulnerability): faille du système permettant à une menace de se réaliser en impactant la sécurité d'un actif.

Risque (Risk): la probabilité qu'une menace exploite une vulnérabilité en mettant en défaut la sécurité d'un actif et provoque un impact pour l'entreprise (R = P x I).

Concepts de la gestion des risques

La gestion des risques se compose de trois blocs interdépendants. Nous distinguons [14]:

- l'organisation cible de l'étude, définie par ses assets et ses besoins de sécurité,
- les risques pesant sur ces assets,
- les mesures prises ayant pour but de traiter les risques et donc d'assurer un certain niveau de sécurité.

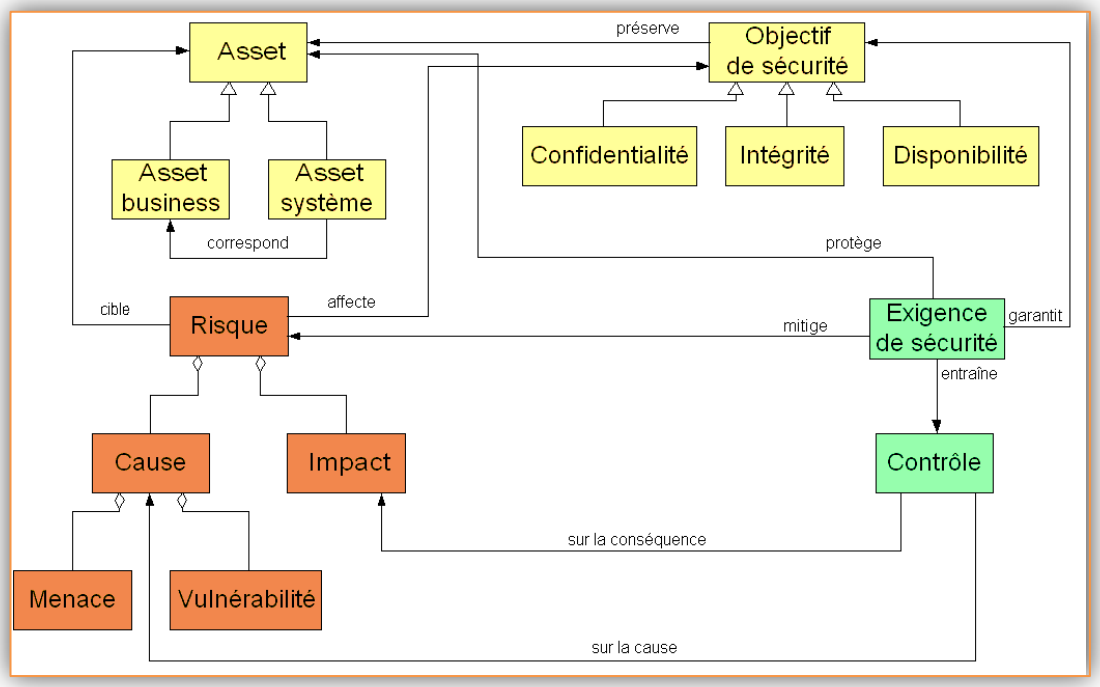


Figure 84 : Les concepts de la gestion des risques

❖ **Actif(Asset)**

Les assets sont définis comme étant l'ensemble des biens, actifs, ressources ayant de la valeur pour l'organisme et nécessaires à son bon fonctionnement. On distingue ici les assets du niveau business, des assets liés au SI.

Du côté des assets business

On retrouve principalement des informations (par exemple des numéros de carte bancaire) et des processus (comme la gestion des transactions ou l'administration des comptes).

Du côté des assets système

Les assets business de l'organisme sont bien souvent entièrement ou partiellement gérés au travers du SI, ce qui entraîne une dépendance de ces assets vis-à-vis de ce dernier. C'est ce que l'on appelle les « assets système ». On retrouve dans les assets système les éléments techniques, tels que les matéraux, les logiciels et les réseaux, mais aussi l'environnement du système informatique, comme les utilisateurs ou les bâtiments. C'est cet ensemble qui forme le SI.

Le but de la gestion des risques est donc d'assurer la sécurité des assets, sécurité exprimée la plupart du temps en termes de confidentialité, intégrité et disponibilité, constituant les objectifs de sécurité.

Ces assets à protéger sont soumis à des risques de sécurité. Le guide 73 de l'ISO définit un risque par la combinaison de la probabilité d'un événement et de ses conséquences. Cette définition est généralement étendue et on définit un risque à l'aide de ce que l'on nomme « l'équation du risque » :

RISQUE = MENACE * VULNÉRABILITÉ * IMPACT

Cette équation est celle qui est la plus couramment utilisée et la plus reconnue dans le domaine de la gestion des risques. Elle joue un rôle fondamental dans l'identification et l'évaluation du risque. Pour bien comprendre la notion de risque, il est important de se pencher sur chacune de ses composantes.
Tout d'abord, **MENACE** ,la source du risque, est l'attaque possible d'un élément dangereux pour les assets. C'est l'agent responsable du risque. Ensuite, **LA VULNERABILITE** est la caractéristique d'un asset constituant une faiblesse ou une faille au regard de la sécurité. Enfin, **L'IMPACT** représente la conséquence du risque sur l'organisme et ses objectifs. La menace et la vulnérabilité, représentant la cause du risque, peuvent être qualifiées en termes de potentialité. L'impact peut, quant à lui, être qualifié en termes de niveau de sévérité.
Afin de mitiger ces risques et de protéger les assets, une politique de traitement des risques est mise en place. Elle sera constituée **D'EXIGENCES DE SECURITE** permettant de répondre aux risques. Ces exigences de sécurité vont ensuite entraîner la mise en place de contrôles (ou contre-mesures) de sécurité à implémenter, afin de satisfaire aux exigences. **LES CONTROLES** sont de deux types :

- Sur la menace ou la vulnérabilité, afin de limiter la cause du risque.
- Sur l'impact, afin de limiter la conséquence du risque.

Le processus de gestion des risques

Après avoir mis en évidence les concepts intervenant dans la gestion des risques, on peut identifier un processus de haut niveau couvrant ses activités. Ce processus est presque toujours appliqué dans les méthodes pratiques de gestion des risques, comme nous le verrons par la suite.

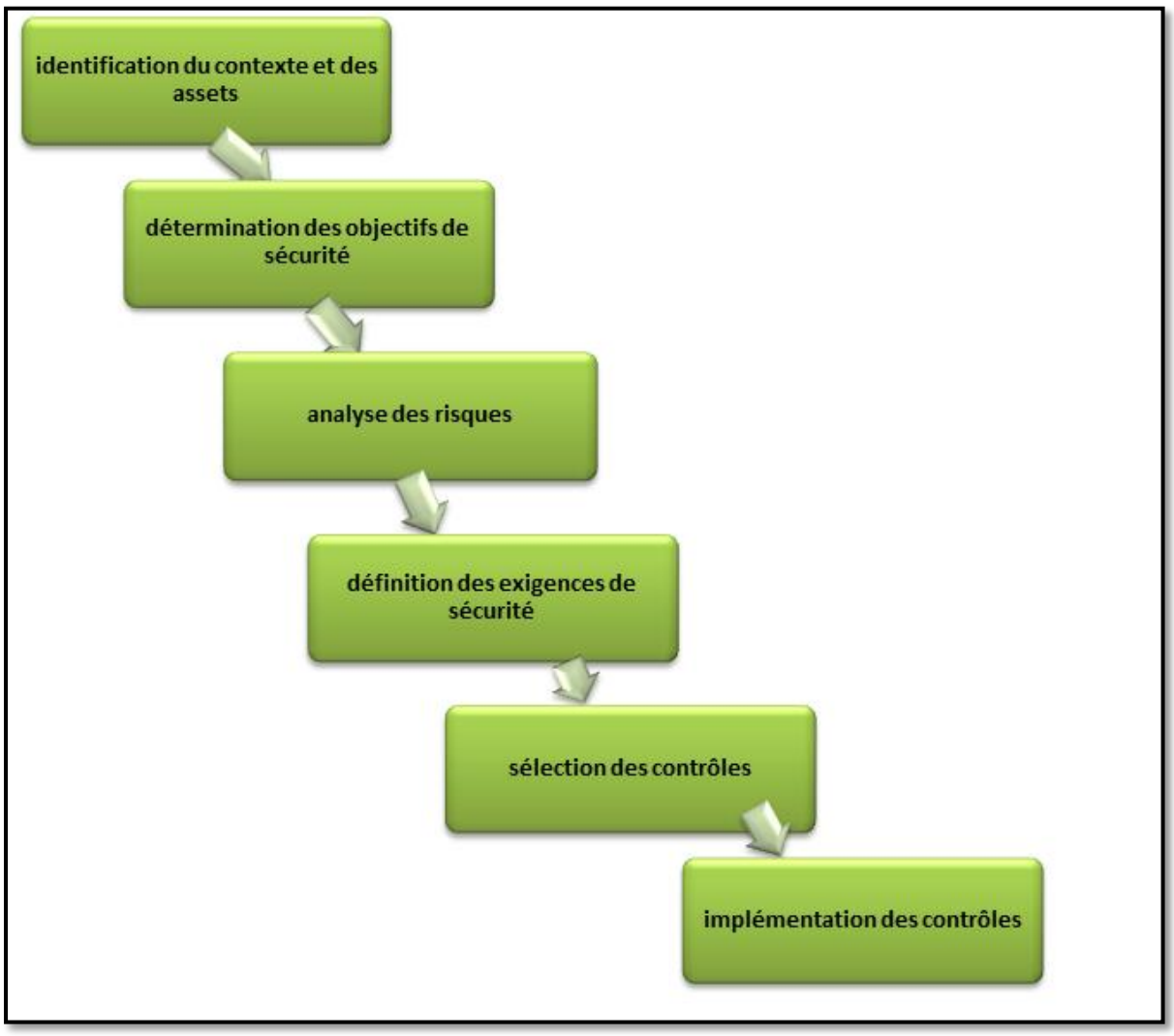


Figure 85.Le processus de gestion des risques

❖ L'identification du domaine et des assets

Prendre connaissance avec l'organisation, son environnement, son SI et de déterminer précisément les limites du système sur lequel va porter l'étude de gestion des risques. Une fois notre système borné, on procède en premier lieu à l'identification des assets business constituant la valeur de l'organisation. Ensuite, le lien sera fait entre ces assets business et les assets système, sur lesquels on identifiera et corrigera les risques d'un point de vue technique et organisationnel.

❖ La détermination des objectifs de sécurité

Vise à spécifier les besoins en termes de confidentialité, intégrité et disponibilité des assets, en particulier au niveau business. Le lien entre les assets business et les assets système étant fait en amont, on retrouve donc les besoins en sécurité au niveau du système.

Exemple : La base de données clients a donc un fort besoin de confidentialité, lié à la sensibilité des informations qu'elle contient.

❖ L'analyse des risques

Constitue le cœur de la démarche de gestion des risques. Elle a pour finalité l'identification et l'estimation de chaque composante du risque (menace/vulnérabilité/impact), afin d'évaluer le risque et d'apprécier son niveau, dans le but de prendre des mesures adéquates (parfois, cette étape est également appelée « appréciation du risque ».

Il y a deux grandes écoles pour l'identification des risques :

- En réalisant un audit du système et de ses différents acteurs.
- A partir de bases de connaissances prédéfinies.

Pour l'estimation des risques, il est possible en théorie de les quantifier à l'aide de distributions de probabilités sur les menaces et les vulnérabilités, ainsi qu'en estimant les coûts occasionnés par les impacts.

En pratique, il se révèle difficile de donner des valeurs absolues et on se contente bien souvent d'une échelle de valeurs relatives, par exemple, allant de 1 à 4. Cette estimation permet de faire un choix, représenté sur la figuresuivante, dans le traitement du risque, avant de passer à la détermination des exigences de sécurité.

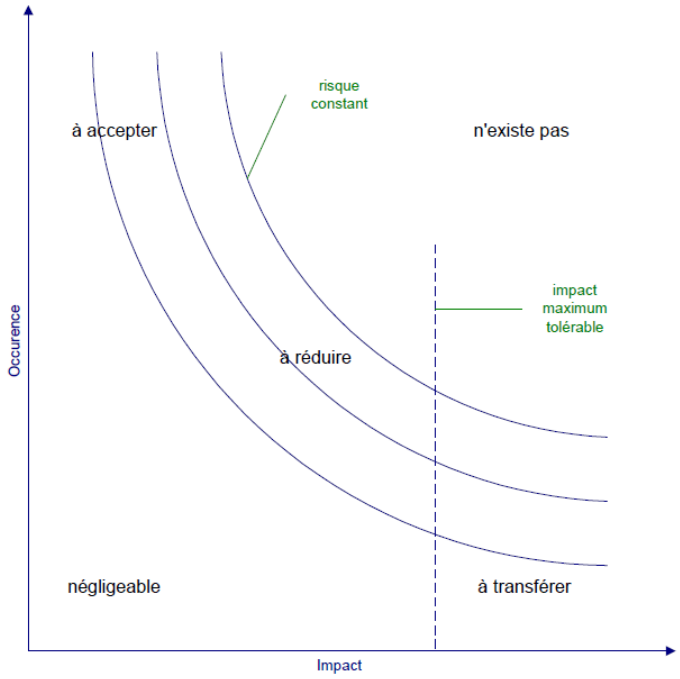


Figure 86: Les différentes zones de risque

D'une manière générale, on considère que :

- Les risques ayant une occurrence et un impact faible sont négligeables.
- Les risques ayant une forte occurrence et un impact important ne doivent pas exister, autrement une remise en cause des activités de l'entreprise est nécessaire (on évite le risque, avoidance en anglais).
- Les risques ayant une occurrence forte et un impact faible sont acceptés, leur coût est généralement inclus dans les coûts opérationnels de l'organisation (acceptation du risque).
- Les risques ayant une occurrence faible et un impact lourd sont à transférer. Ils peuvent être couverts par une assurance ou un tiers (transfert du risque).
- Enfin, les autres risques, en général majoritaires, sont traités au cas par cas et sont au centre du processus de gestion des risques ; l'objectif, étant de diminuer les risques en les rapprochant au maximum de l'origine de l'axe (mitigation du risque à l'aide de contrôles).

Exemple : L'analyse des risques montre un certain nombre de scénarios ayant un niveau de risque inquiétant pour la base de données clients. Un des risques identifiés est l'accès aux données par un utilisateur non-autorisé à travers le réseau. Cela aurait pour conséquence de nuire à la confidentialité des données. Ce risque nécessite d'être traité par des contrôles de sécurité.

Une fois l'analyse des risques effectuée, la définition des exigences de sécurité permettra de réduire les risques identifiés. Comme précédemment, en fonction des méthodes, cette étape pourra être effectuée avec l'assistance de référentiels [9,10], ou aiguillée par la connaissance d'experts système/sécurité. La définition des exigences de sécurité, de par son

importance et sa complexité, est souvent effectuée de manière incrémentale et par raffinement successif. En effet, on conseille bien souvent de débuter par des exigences générales, qui définiront l'intention de contrer les risques identifiés (de niveau stratégique), pour les raffiner ensuite en des exigences plus précises (vers le niveau opérationnel). Toutefois les exigences sont censées être génériques et applicables à tout SI. Il faut également rappeler que ces exigences de mitigation des risques porteront à la fois sur le système informatique (comme le besoin d'encryption des mots de passe), mais aussi sur son environnement (par exemple, « l'utilisateur du système ne doit pas dévoiler son mot de passe à un tiers »).

Exemple : Le serveur doit être protégé, dans sa globalité, des intrusions éventuelles. De même, on décide de mettre en place un contrôle d'accès adéquat sur la base de données. Une authentification à distance, plus forte que l'authentification actuelle par login/password, est nécessaire. Toutefois le coût et la simplicité d'utilisation de la solution choisie doivent rester à un niveau acceptable.

Le dernier niveau de raffinement est constitué par la sélection des contrôles (ou contre-mesures) de sécurité. Les contrôles sont l'instanciation des exigences de bas niveau pour le système cible étudié. Ici sont définis les choix techniques des solutions de sécurité, influencés par le système déjà en place, les compétences disponibles, les coûts de mise en œuvre…

Exemple : On sélectionne l'ajout d'un IDS au firewall déjà en place. De plus, on décide de mettre en place une politique de password plus exigeante constituée d'un login/password ainsi que d'un code de contrôle distribué à l'utilisateur lors de son inscription. Une authentification par carte à puce est trop contraignante.

Une fois les contrôles sélectionnés, il reste alors à les implémenter dans le SI et à éventuellement les tester et les évaluer. Il subsiste alors indéniablement une part de risques traités partiellement ou non, qui constitue ce que l'on appelle le risque résiduel.

Exemple : Le service informatique de la société mettra en place l'IDS et le prestataire chargé de la maintenance du site Web développera un nouveau portail supportant l'authentification désirée.

Ce processus est communément admis par les différentes méthodes de gestion des risques. Par contre, la terminologie est souvent très différente, d'une méthode ou norme à une autre. La comparaison du processus de plusieurs méthodes nécessite alors une bonne analyse, mais dans l'ensemble, le schéma présenté précédemment est suivi. Toutefois, quelques méthodes se distinguent en présentant une trame quelque peu différente ou étendue (tout en gardant bien souvent comme base l'inamovible processus générique présenté). Citons notamment le BS 7799-2:2002 qui voit la gestion des risques comme un processus suivant le paradigme PDCA de ou d'autres méthodes utilisant la gestion des risques dans une démarche de conception de système

Le processus de gestion de Sécurité

Basé sur la roue de Deming afin de repérer avec simplicité les étapes à suivre pour améliorer la qualité dans une organisation.
La méthode comporte quatre étapes, chacune entraînant l'autre, et vise à établir un cercle vertueux. Sa mise en place doit permettre d'améliorer sans cesse la qualité d'un produit, d'une œuvre, d'un service, etc.

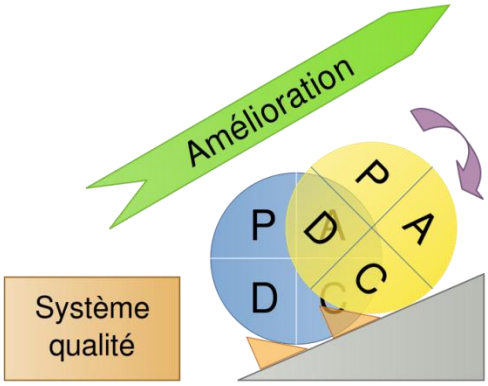


Figure 87: Roue de Deming

La première étape **Plan,** consiste à planifier la réalisation. Elle se déroule généralement en trois phases :

1. Identification du problème à résoudre ou du processus à améliorer.
2. Recherche des causes racines (par exemple à l'aide d'un diagramme de Pareto, d'un diagramme d'Ishikawa ou de la méthode des 5 pourquoi).
3. Recherche de solutions avec écriture du cahier des charges et établissement d'un planning.

L'étape **Do** (en français « faire ») est la construction, le développement, la réalisation de l'œuvre.

Elle est suivie de l'étape **Check** (en français « vérifier »), qui consiste à contrôler l'aptitude de la solution mise en place à résoudre le problème ou à améliorer le processus. Sont employés à cet effet des moyens de contrôle divers, tels que les indicateurs de performance.

Puis l'étape **Act** consiste à agir et réagir, c'est-à-dire corriger et améliorer la solution mise en place, voire à standardiser cette solution. L'étape Act amène donc un nouveau projet à réaliser, donc une nouvelle planification à établir. Il s'agit donc d'un cycle que l'on représente à l'aide d'une roue. À chaque étape, la roue avance d'un quart de tour. Cette avancée représente l'action de progresser.

De plus, pour éviter de « revenir en arrière », on représente une cale sous la roue, qui empêche celle-ci de redescendre et qui symbolise par exemple un système qualité, ou un système d'audits réguliers, ou un système documentaire qui capitalise les pratiques ou les décisions.

Méthodes d'analyse de risques

❖ L'avènement de ISO/IEC 27005:2008 [15]

L'ISO/IEC 27005 est la norme internationale qui traite de la gestion de risques en matière de Sécurité de l'Information. Elle se repose sur les concepts généraux spécifiés dans la norme ISO /IEC 27001(relative au Système de Management de la Sécurité de l'Information (SMSI)) et participe à la mise en œuvre d'un niveau de sécurité de l'Information satisfaisant.

Aussi, il est important de connaître les concepts, modèles, processus et termes exposés dans l'ISO/IEC 27001 et l'ISO/IEC 27002 (code de bonnes pratiques pour la gestion de la sécurité de l'Information). Cette norme internationale est applicable à tous les types d'organisations (par exemple, des entreprises commerciales, des organismes gouvernementaux, des organisations à but non lucratif) qui ont l'intention de gérer les risques qui pourraient compromettre l'organisation de la sécurité de l'information.

❖ Processus de gestion des risques SSI

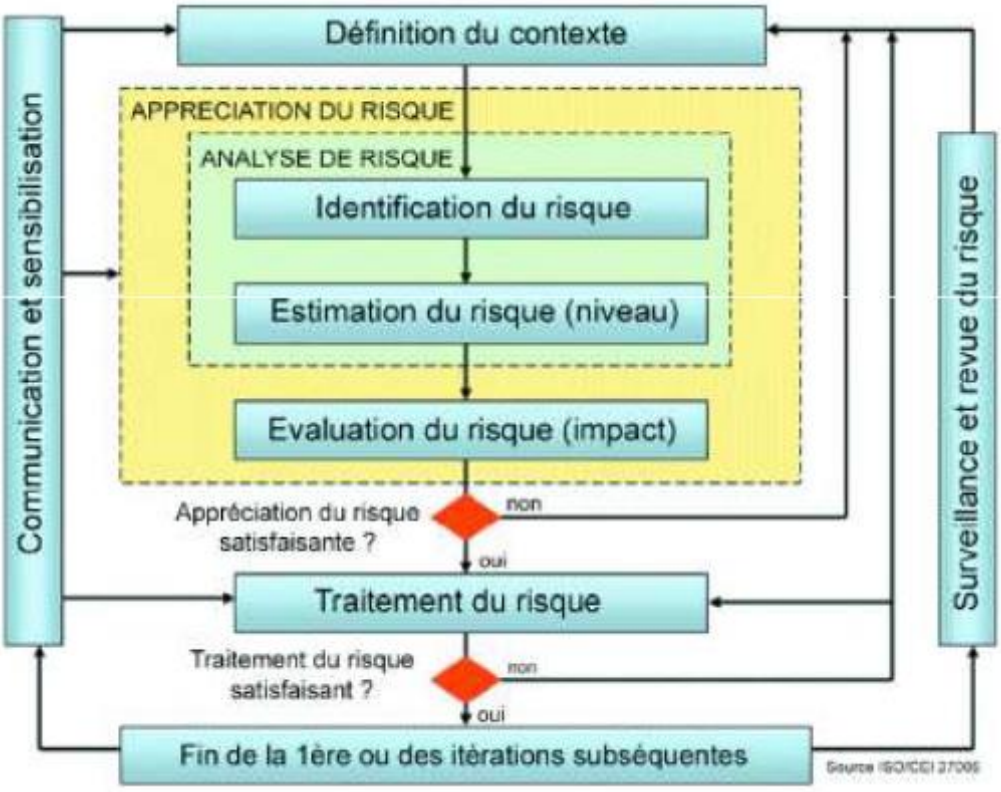


Figure 88: Processus de gestion des risques -ISO/IEC 27005 :2008 [15]

❖ **Bref historique des méthodes de gestion des risques SSI**

Méthode/ Révision	Titre	Auteur	Clientèle cible	Utilisateur cible	Principales étapes	Base de connaissances	Norme
MEHARI / 2007	Analyse et de management de risques	CLUSIF	Organismes et entreprises	Responsable de la Sécurité des Systèmes d'information et Risk manager	diagnostics de sécurité, analyse des enjeux et analyse des risques	Gratuite La base peut être enrichie	ISO 17799 ISO 27001
Cramm / 2005	Évaluation des risques	Siemens et Gouv. UK	Organismes et entreprises	Risk manager	établissement des objectifs, évaluation des risques et identification sélection des contre-mesures	Payant Expert et Express	ISO17799
EBIOS / 2004	Appréciation des risques	Conseil de la DCSSI et club EBIOS	Organismes (administra-tions de l'État)	Expert en sécurité de l'organisme	étude du contexte, expression des besoins, étude des menaces, identifier les objectifs et déterminer les exigences de sécurité	Gratuite Utilise plutôt des guides.	ISO 17799
OCTAVE / 2003	Évaluation des vulnérabilités et des menaces sur les ressources et informations opérationnelles	CERT - université de Carnegie Mellon	Organismes	Personnes de l'organisation connaissant ses pratiques et processus internes.	Vue organisationnelle, vue technique et stratégie de sécurité	Gratuite OCTAVE, OCTAVE-S	ISO 17799

Tableau 6 : Tableau récapitulatif des méthodes de gestion des risques SSI

Les méthodes de gestion des risques sont nombreuses et leurs approches peuvent être différentes. Il n'existe pas de bonnes ou de mauvaises méthodes de gestion des risques, puisqu'il y a une grande diversité dans les activités, dans les enjeux et dans les approches de la sécurité

Un Récapitulatif publié par le Centre de Recherche Public TUDOR HENRI en 2009 qui présente [14]:

• La norme internationale ISO/IEC 27005 qui traite de la gestion des risques des SI (Système d'Information).

• Les méthodes les plus utilisées à savoir EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité), MEHARI (Méthode Harmonisée d'Analyse de Risques), OCTAVE/OCTAVE-S (Operationally Critical Threat, Asset, and Vulnerability Evaluation/Small).

Pour chacune d'elles, les objectifs, les processus et les outils associés aux méthodes sont présentés de manière à en faciliter la comparaison.

Le tableau suivant met en évidence les avantages et inconvénients de chaque méthode.

	Avantages	Inconvénients
ISO/CEI 27005	• Démarche flexible	• Ne constitue pas un guide

EBIOS	• Démarche très logique • Traitement des exigences • Logiciel • Bases de connaissances • Soutenue par la DCSSI • Très reconnue	• Distingue peu l'opérationnel du reste de l'organisation • Difficile d'utilisation pour un novice de la sécurité
MEHARI	• Logiciel • Conçu par le CLUSIF • Reconnue	• Méthode potentiellement lourde
OCTAVE /OCTAVE-S	• Orientée métier • Accessible à tous • Documentation très complète • Catalogue de pratiques • Variante –S pour les PME • Soutenue par le CERT /CC (Computer Emergency Response Team/ Coordination Center)	• Méthode potentiellement lourde

Tableau 7 : Avantages et inconvénients des méthodes d'analyse de risque

Attributes / Methods	Risk identification	Risk Analysis	Risk Evaluation	Risk assessment	Risk treatment	Risk acceptance	Risk communication	Languages	Price (method only) (Information assessed in June 2006)	Size of organization	Skills needed[6]	Licensing	Certification	Dedicated support tools
Austrian IT Security Handbook	••	•	•	•••	•••	•••	•••	DE	Free	All	**	N	N	Prototype (free of charge)
Cramm	•••	•••	•••					EN, NL, CZ	Not free	Gov, Large	***	N	N	CRAMM expert, CRAMM express
Dutch A&K analysis	•••	•••	•••					NL	Free	All	*	N	N	
Ebios	•••	•••	•••	•••	•••	•••	•••	EN, FR, DE, ES	Free	All	**	Y	N	EBIOS version 2 (open source)
ISF methods	•••	•••	•••	•••	•••	•••	•••	EN	For ISF members	All except SME	* to ***	N	N	Various ISF tools (for members)
ISO/IEC IS 13335-2 (ISO/IEC IS 27005)	••	••	••	••	•••	•••	•••	EN	Ca. €100	All	**	N	N	
ISO/IEC IS 17799	•				•			EN	Ca. €130	All	**	N	Y	Many
ISO/IEC IS 27001					•	•		EN, FR	Ca. €80	Gov, Large	**	Y	Y	Many
IT-Grundschutz	•••	•••	•••	•••	•••	•••	•••	EN, DE	Free	All	**	Y	Y	Many
Marion (replaced by Mehari)	•••	•••	•••					EN, FR	Not free	Large	*	N	N	
Mehari	•••	•••	•••					EN, FR	€100-500	All	**	N	N	RISICARE
Octave	••	••	••	••	••	••	••	EN	Free	SME	**	N	N	
SP800-30 (NIST)	•••	•••		•••	•••	•••		EN	Free	All	**	N	N	

Tableau 8 : Comparatif des méthodes d'analyse de risque –ENISA