Spécifications des échanges de proximité avec l'appli carte Vitale

Référence : **APCV-NT-001** / Version : **1.6** / Date : **21/02/2024**
Rédigé par : **GIE SV** / Approuvé par :
Sécurité : **DIFFUSION RESTREINTE**

GIE SESAM-Vitale
5, boulevard Marie et Alexandre Oyon – 72019 LE MANS Cedex 2
www.sesam-vitale.fr

Contacts

Pour toute question technique ou fonctionnelle durant vos développements, contactez l'assistance technique du GIE SESAM-Vitale :

- par courriel : centre-de-service@sesam-vitale.fr

Date	Version	Évolution du document
15/12/2017	0.1	Création du document
29/12/2017	0.2	Ajout du BLE
08/01/2017	0.3	Mise à jour de l'AID
31/01/2018	0.4	Précision sur le NFC HCE (incompatibilité iPhone)
31/01/2018	0.5	Ajout d'un Glossaire
06/02/2018	0.6	Mise à jour addendum 8
16/07/2020	0.7	Mise à jour de la référence du document en ApCV-NT-001
05/10/2020	1.0	Version de référence avec mise à jour des titres du document
18/11/2020	1.1	Ajout des normes NFC et QRCode. Corrections sur des concepts restants de l'expérimentation.
03/12/2020	1.2	Correction mineure trame d'échange (AID)
05/07/2021	1.3	Suppression de l'ATR
02/11/2021	1.4	Ajout paragraphe 6 sur les recommandations de sécurité
23/02/2022	1.5	Mise à jour du message de proximité pour identifier un QR Code APCV
21/02/2024	1.6	Exemple de code de lecture d'un QR Code avec un lecteur en mode USB Com

Tables des matières

Tables des illustrations

GLOSSAIRE

QRCode	Type de code-barres en deux dimensions constitué de modules noirs disposés dans un carré à fond blanc. L'agencement de ces points définit l'information que contient le code.
NFC	*Near Field Communication :* technologie de communication sans fil à courte portée et à haute fréquence, permettant l'échange d'informations entre des périphériques jusqu'à une distance d'environ 10 cm dans le cas général

1 CONTEXTE

Dans le cadre du projet appli carte Vitale, les solutions logicielles des industriels doivent « lire » l'appli carte Vitale présentée par un utilisateur (assuré).

Cette lecture est une « authentification de proximité » : au-delà de récupérer les données des bénéficiaires, l'utilisateur de l'appli carte Vitale est authentifié en ligne.

Cette authentification est réalisée en 2 étapes :

1. **La solution logicielle récupère les données chiffrées de l'appli carte Vitale (NFC, QRCode)**
2. La solution logicielle authentifie en ligne l'utilisateur et récupère les données de l'appli carte Vitale

Ce document a pour objet de décrire comment réaliser l'étape 1 du processus.

2 OBJECTIF DU DOCUMENT

L'objectif de ce document est de spécifier les échanges de proximité entre une appli carte Vitale et une solution logicielle (un équipement homologué ou un logiciel agréé ou autorisé TLSi AMO)

- Récupérer les données en NFC « HCE » ou émulation de carte
- Récupérer les données via la lecture d'un QRCode

3 CHOIX DES MODES D'ECHANGE

L'appli carte Vitale sera disponible pour les assurés dans les systèmes d'exploitation Apple iOS et Google Android.

Du fait des limitations de l'iPhone d'Apple, l'appli carte Vitale sur iOS n'implémentera pas le mode NFC HCE (host card emulation).

↳ **Les solutions logicielles doivent implémenter obligatoirement les 2 modes d'échanges.**

Le choix du dispositif de lecture est à la charge de l'industriel et dépend de l'environnement de travail de l'utilisateur :

- Dispositif orienté vers l'utilisateur de l'appli carte Vitale (lecteur de QRCode sur pied, lecteur NFC type bancaire, ..)
- Dispositif à la main de l'utilisateur de la solution logicielle (par exemple douchette à la main du professionnel de santé, …)

Enfin le dispositif de lecture peut être bi mode : assurer à la fois le NFC et la lecture du QRCode.

4 LECTURE D'UNE APPLI CARTE VITALE EN NFC

4.1 Introduction

Avec Android, il est possible d'échanger des données en PC/SC avec un smartphone Android : Host-based Card Emulation (HCE).

Le smartphone se comporte alors comme une carte à puce sans contact.

4.2 Norme ISO 14443 et AID

La norme ISO 14443 définit le fonctionnement des cartes à puce sans contact, ou cartes de proximité, c'est-à-dire des cartes dans lesquelles est intégrée une antenne de radiofréquence (RF).

La norme décrit le protocole de communication sans fil utilisé au niveau de la couche de liaison entre une carte et un lecteur de carte qui fonctionne à 13,56 MHz (RFID HF).

Ici, le smartphone émule une carte à puce (« HCE », host emulation card »).

L'identifiant de l'application est l'AID.

L'AID de l'appli carte Vitale est : **D2 50 00 00 02 41 50 43 56**

4.3 Architecture

La solution logicielle a besoin d'un dispositif pour échanger en NFC.

Ce dispositif de **lecture NFC doit être compatible avec l'ISO 14443**.

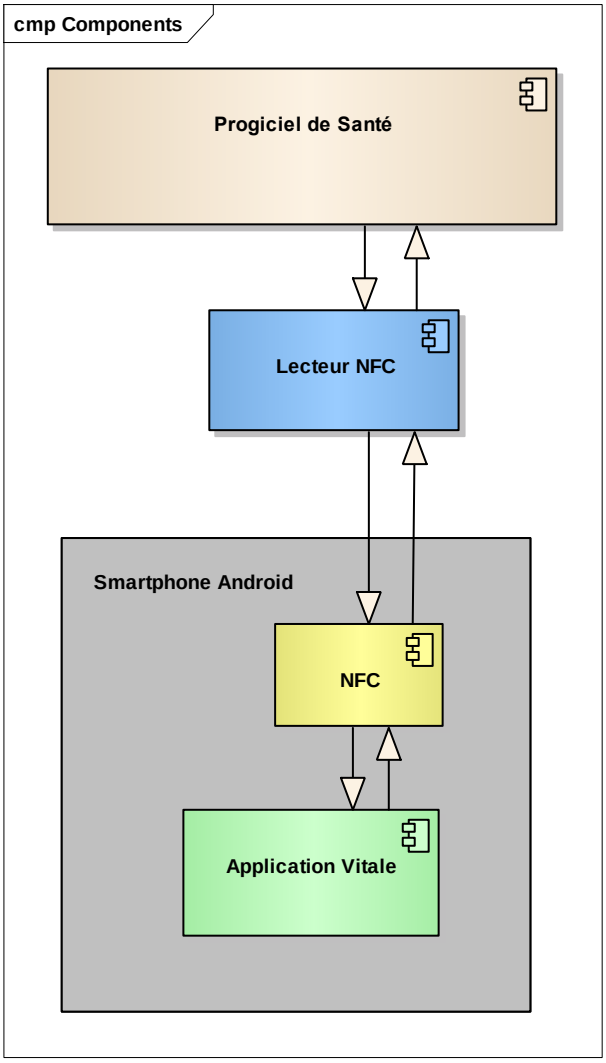

Figure 1 : ApCV mode Host Card Emulation

4.4 Spécifications des échanges PC/SC

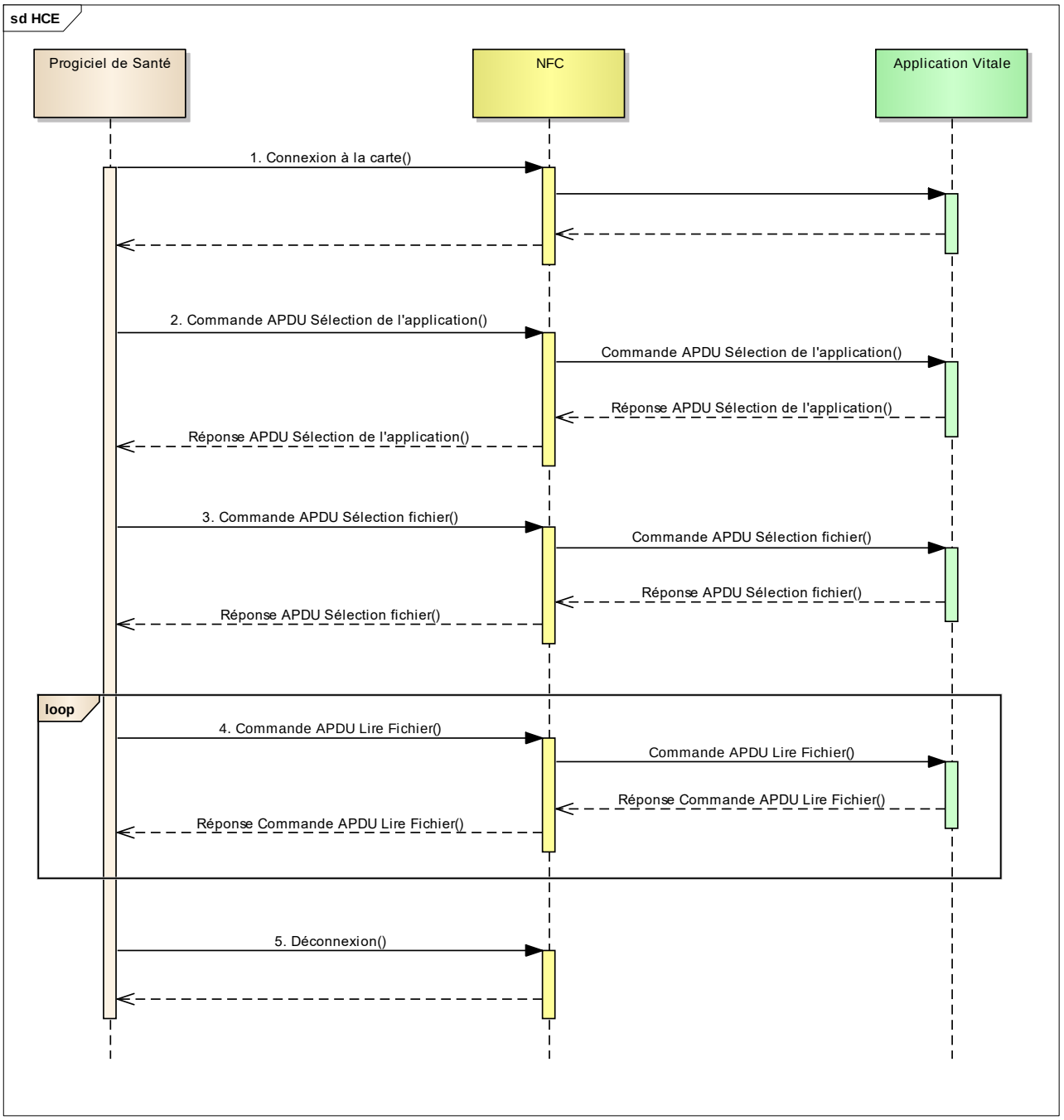

Figure 2 : Vue dynamique : Lecture de l'appli carte vitale en NFC

Pour lire une appli carte vitale en NFC, il faut enchainer les étapes suivantes :

1. Connexion à la carte

   - Pour se connecter à la carte, il faut utiliser le protocole SCardConnect T= *
   - Une fois la « carte » connectée, il faut ouvrir une exclusivité sur la « carte » SCardBeginTransaction.

     L'ATR n'est pas exposé par HCE. Une valeur peut être restituée mais sa valeur ne doit pas être analysée.

2. Commande APDU Sélection de l'application

Commande APDU	
Champ	Valeur
CLA	0
INS	A4
P1	04
P2	0C
L_c	09
Données envoyées	D2 50 00 00 02 41 50 43 56
Réponse APDU	
SW1-SW2	90 00

La donnée D2 50 00 00 02 41 50 43 56 correspond à l'AID de l'appli carte Vitale.

3. Commande Sélection fichier

Commande APDU	
Champ	Valeur
CLA	00
INS	A4
P1	02
P2	00
Ne	EF
Réponse APDU	
SW1-SW2	90 00

4. Commande lecture

La « carte » est lue par bloc de 255 octets.

- P1 se calcule comme suit : `(P1 = ((longueurLue >> 8) & 255)`
- P2 se calcule comme suit : `(P2 = ((longueurLue & 255)`

Commande APDU	
Champ	Valeur
CLA	00

INS	B0
P1	00
P2	00
Ne	FF

- Si 255 octets ont été lus la réponse est la suivante :

Réponse avec SW1=90 SW2=00 → 255 octets ont été lues	
SW1	90
SW2	00
Réponse	255 octets

- Si la longueur à lire est inférieure à 255 octets la réponse est la suivante :

Réponse avec SW1=90 SW2=00 → 255 octets ont été lues	
SW1	6C
SW2	Ne octets restant à lire

Dans ce cas il convient, de passer la dernière commande pour lire les Ne derniers octets :

Commande APDU	
Champ	Valeur
CLA	00
INS	B0
P1	
P2	
Ne	Ne octets de la commande précédente
Reponse APDU	
SW1	90
SW	00
Ne	Ne octets

5. Déconnexion
   - Fermeture de l'exclusivité avec la « carte »
   - Déconnexion.

4.5 Exemple de lecture d'une appli carte Vitale

L'appli carte Vitale lue comporte 440 octets.

Commande	Commande APDU	Réponse APDU
Sélection de l'application	00 A4 04 0C 09 D2 50 00 00 02 41 50 43 56	90 00
Sélection du fichier Vitale1	00 A4 02 00 EF	90 00
Lecture	00 B0 00 00 FF	90 00 et 255 octets lues

	00 B0 00 FF FF	6C **B9**
	00 B0 00 FF **B9**	90 00 et 185 (0x**B9**) octets

4.6 Exemple de lecture d'une appli carte Vitale en code en Java

```
package fr.sesamvitale.poc.qrcode.pcom;

import java.io.ByteArrayOutputStream;
import java.nio.charset.StandardCharsets;
import java.util.List;

import javax.smartcardio.Card;
import javax.smartcardio.CardChannel;
import javax.smartcardio.CardException;
import javax.smartcardio.CardTerminal;
import javax.smartcardio.CommandAPDU;
import javax.smartcardio.ResponseAPDU;
import javax.smartcardio.TerminalFactory;

@SuppressWarnings("restriction")
public class App {
  private static final int MAX_READ_SIZE = 255;

  static {
    // Définition des propriétés système du protocole de la carte
    // afin que "GET RESPONSE" soit appelé manuellement
    System.setProperty("sun.security.smartcardio.t0GetResponse", "false");
    System.setProperty("sun.security.smartcardio.t1GetResponse", "false");
  }

  /**
   * @param args
   * @throws CardException
   */
  public static void main(String[] args) throws CardException {
    Card card = null;
    CardTerminal terminal = null;
    try {
      // Récuperation de la liste des périphériques
      TerminalFactory factory = TerminalFactory.getDefault();
      List<CardTerminal> terminals = factory.terminals().list();

      // Sélection du périphérique (le premier de la liste dans cet exemple)
      terminal = terminals.get(0);

      // Attente de la présentation de la carte pendant 1 minute (60000 millisecondes)
      if (terminal.waitForCardPresent(60000)) {
        // Tentative de connexion à la carte
        card = terminal.connect("*");
        card.beginExclusive();
      }
      else {
        // Le timeout a expiré
        System.out.println("Carte non présente.");
        System.exit(1);
      }

      CardChannel channel = card.getBasicChannel();

      CommandAPDU cmdAPDU = null;
      ResponseAPDU answer = null;

      // Sélection de l'application ApCV D2 50 00 00 02 41 50 43 56
      byte[] aid = { (byte) 0xD2, 0x50, 0x00, 0x00, 0x02, 0x41, 0x50, 0x43, 0x56};
      cmdAPDU = new CommandAPDU(0x00, 0xA4, 0x04, 0x0C, aid);
      answer = channel.transmit(cmdAPDU);

      // Sélection du fichier Vitale1
      cmdAPDU = new CommandAPDU(0x00, 0xA4, 0x02, 0x00, 0xEF);
      answer = channel.transmit(cmdAPDU);

      ByteArrayOutputStream bos = new ByteArrayOutputStream();

      if (answer.getSW1() == 0x90) {
        int P1 = 0;
        int P2 = 0;
        int longueurAlire = MAX_READ_SIZE;
        int longueurLue = 0;
        boolean derniereLecture = false;
```

```
        while (true) {
          // Lecture des données
          cmdAPDU = new CommandAPDU(0x00, 0xB0, P1, P2, longueurAlire);
          answer = channel.transmit(cmdAPDU);

          int w1 = answer.getSW1();

          // Si le code retour est 0x90, il reste encore des données à lire
          if (w1 == 0x90) {
            byte[] data = answer.getData();
            bos.write(data, 0, data.length);

            longueurLue += data.length;

            if (derniereLecture)
              break;

            // renseignement de l'offset pour la prochaine lecture
            // l'octet de poids fort de l'offset dans le paramètre P1
            P1 = ((longueurLue >> 8) & 0xFF);
            // l'octet de poids faible de l'offset dans le paramètre P2
            P2 = (longueurLue & 0xFF);
          }
          else if (w1 == 0x6C) {
            // Si le code retour est 0x6C, la longueur des données restant
            //  à lire est inférieure à la longueur demandée,
            // on relance une dernière lecture avec la longueur restante
            longueurAlire = answer.getSW2();

            if (longueurAlire == 0)
              break;

            dernieereLecture = true;
          }
          else {
            break;
          }
        }
        byte[] bufRead = bos.toByteArray();
        System.out.println(bufRead.length + " octets lues : " +
                           new String(bufRead, StandardCharsets.UTF_8));
      }
    }
    catch (Exception e) {
      System.out.println(e.getMessage());
    }
    finally {
      if (card != null) {
        card.endExclusive();
        card.disconnect(false);
      }
    }
    System.exit(0);
  }
}
```

5 LECTURE D'UNE APPLI CARTE VITALE EN CODE 2D

5.1 Norme utilisée et lecture

Le code 2D affiché sur l'appli carte Vitale est un QR Code.

Ce QRCode répond à la norme **ISO/IEC 18004:2015**

La lecture est standard et le bloc chiffré est lu en une fois.

5.2 Lecteurs de QR Code

Le dispositif de lecture est un lecteur de codes **compatible 2D QRcode**.

- Soit implémentant une technologie LASER (douchettes classiques)
- Soit implémentant une lecture vidéo (caméra miniature HD) avec reconnaissance des codes 2D QRCode.

5.3 Identifier un QR Code ApCV

Un QR Code APCV commence toujours par les caractères **PB83N8 en base 45.**

5.4 Configurer un lecteur de QR Code en mode USB COM

Par défaut, les lecteurs de codes **compatible 2D QRcode** sont configurés en mode HID Clavier. Le GIE SESAM-Vitale recommande la configuration des lecteurs en mode USB COM. Ce mode de configuration permet une lecture performante et s'affranchit du problème de verrouillage Majuscule du clavier (CAPS LOCK).

Pour se faire, il suffit de scanner le code 2D fournit pas le fabricant du lecteur permettant la configuration dans le mode choisi. Ce code est propre à chaque équipement et est disponible dans le manuel utilisateur ou sur le site du fabricant.

Une fois le lecteur configuré dans ce mode, la lecture des données se fait sur le port COM associé au lecteur. Il est possible de connaitre sa valeur depuis le gestionnaire de périphérique dans la catégorie Ports (COM et LPT).

Note : il est aussi possible d'utiliser le lecteur en mode HID clavier (mode par défaut). Cette hypothèse de configuration requiert de paramétrer le lecteur en clavier Français pour s'affranchir du problème de verrouillage Majuscule du clavier. Ce mode de configuration n'est pas préconisé par le GIE SESAM-Vitale car moins performant par rapport à un lecteur configuré en mode USB COM.

5.5 Code exemple Java pour la lecture d'une appli carte Vitale sur le port COM

```
package fr.sesamvitale.poc.pcsc ;

import java.io.InputStream;
import java.io.UnsupportedEncodingException;
import com.fazecast.jSerialComm.SerialPort;
import com.fazecast.jSerialComm.SerialPortDataListener;
import com.fazecast.jSerialComm.SerialPortEvent;

public class LireQRCode {

    //fonction de lecture d'un qr code en port serie
      public void lireQRCodePcom2(String portCom){
          // recherche du nom de port com dans la chaine transmise
         SerialPort comPort = SerialPort.getCommPort(portCom);
```

```
            // ouverture du port
            boolean res = comPort.openPort();
            if (res == true) {
               // attente QRCode
                comPort.setComPortTimeouts(SerialPort.TIMEOUT READ BLOCKING, 1000, 0);
                int numRead = 0;
                byte[] readBuffer = new byte[5000]; // un QRCODe max 4296 caractères selon la norme

                // variables pour le timer
                long debutAttente = System.currentTimeMillis();
                long tempsAttente = 0;
                try {
                    // on attend 20 secondes le scan du QRCode
                   while (numRead == 0 && tempsAttente < 20000)
                   {
                       long suiteAttente = System.currentTimeMillis();
                       tempsAttente = suiteAttente - debutAttente;

                       numRead = comPort.readBytes(readBuffer, readBuffer.length);
                   }
                   if (numRead != 0)
                   {
                       String chaineLu = new String(readBuffer, "UTF8");
                      // test sur la chaine PB83N8
                       if (chaineLu.startsWith("PB83N8"))
                       {
                          // "Le QR Code correspond à l'application carte Vitale."
                          // vous pouvez utiliser le QR Code obtenu dans chaineLu
                       }
                       else {
                           // "le QR Code ne correspond pas à l'application carte Vitale."
                       }
                   }
                   else {
                       // "Aucun QR Code lu dans le délai de 20 secondes. Relancez la lecture."
                   }
                } catch (Exception e) {
                    // gerer l'exception obtenue  e.getMessage()
                  }
                finally {
                    comPort.flushIOBuffers();
                    comPort.closePort();
                  }
            }
            else {
                // " ouverture du port com n'a pas abouti "
            }
        }
}
```

6 RECOMMANDATION SECURITE

6.1 Gestion des correctifs de sécurité

Il est recommandé de mettre en place un processus de gestion des correctifs de sécurité : il s'agit mettre en œuvre des mesures organisationnelles et techniques permettant de garantir le maintien de la sécurité de la solution, incluant à minima :

- Une veille relative aux vulnérabilités de l'ensemble des composants logiciels et matériels de la solution, notamment des drivers permettant l'acquisition des données de l'ApCV (lecteur de QR-Code, lecteur NFC).
- La mise en œuvre des correctifs de sécurité (développement des correctifs, déploiement, assistance aux utilisateurs).

6.2 Supervision et gestion des incidents de sécurité

Il est recommandé de mettre en place l'organisation et les outils permettant de réaliser une supervision des anomalies et événements de sécurité liés à la solution et à son système d'information.

Cette supervision peut s'appuyer sur les journaux des systèmes (pare-feu, OS, outils de détection d'intrusions, contrôle d'intégrité, etc.) et des services liés à la solution.

6.3 Guide d'installation et conditions d'utilisation

Il est recommandé de mettre à disposition des guides et conditions d'utilisation afin de permettre une bonne intégration et un usage dans de bonnes conditions de sécurité de sa solution au sein du système d'information du professionnel de santé.

Le guide doit rappeler les bonnes pratiques et besoins de sécurité ainsi que les conditions d'utilisation de la solution au niveau du SI PS (mises à jour de sécurité du système, présence d'un anti-virus, etc.). Il doit également indiquer la responsabilité de l'utilisateur vis-à-vis de la sécurisation de son système d'information et notamment de son réseau local (filaire, wifi…), dans le cas où celui-ci est partagé avec l'équipement.

Ces éléments doivent figurer dans le contrat de mise à disposition de la solution aux clients / utilisateurs ou dans une de ces annexes (CGU…).

6.4 Formation et sensibilisation du personnel

Le personnel intervenant dans les phases de conception, développement, déploiement, maintenance doit être formé et sensibilisé à la sécurité de l'information et aux bonnes pratiques de sécurité.