

TRANSPORT PATIENTS INTERFACE SOCIETES PRIVEES

Spécifications fonctionnelles détaillées

GEO SOFT AQUITAINE

09/11/2010

Créé par : J. Crézé, L. Pautal, R. Givernaud
modifié le 04/05/2011

Interface Sociétés Privées

Spécifications fonctionnelles détaillées

I. Table des matières

1. OBJECTIFS DE L'INTERFACE SOCIETES PRIVEES	3
2. DESCRIPTION GENERALE DE L'INTERFACE	3
A. PRINCIPE :	3
B. EMBLACEMENT DES FICHIERS ET MODALITES D'ECHANGE	4
C. NOM DE FICHIERS	5
D. FORMAT DES FICHIERS	5
E. MODALITES	6
F. CAS D'ERREUR DETECTE PAR LE SYSTEME DE LA SOCIETE PRIVEE	6
G. CAS D'ERREUR DETECTE PAR LE SYSTEME DE PTAH	7
3. SPECIFICATION DU FORMAT DES FICHIERS D'ECHANGE	7
A. FICHIERS MISSIONS	7
B. MODIFICATION D'ETATS DE MISSION	9
C. SUIVI DE MISSION	9
4. PROTOCOLE D'ECHANGE DE L'INFORMATION	10
A. PRINCIPES GENERAUX	10
B. DESCRIPTION DU PRINCIPE D'ECHANGE	11
C. PROTOCOLE DE COMMUNICATION DU CANAL « PUSH »	11

Versions

V1.00 03/02/2010 Création de ce document

V2.00 02/04/2010 Ajouts et modifications

V3.00 09/11/2010 Ajouts sur le protocole de communication

V3.01 12/06/2012 Correction des exemples de réponses

Introduction

Ce document a pour but de spécifier les échanges de données avec les applications de gestion des sociétés privées de transports sanitaires :

- description des fichiers de missions ou de modification d'état qui seront envoyées aux sociétés Privées par PTAH
- description des fichiers de suivi de mission (informations en retour) des sociétés privées et adressées à l'application PTAH

1. Objectifs de l'interface Sociétés privées

L'objectif de cette interface est de générer des transports intégrables dans les différents logiciels de gestion de transports sanitaires et de renvoyer des acquittements de réception. et de traitement.

Il s'agit de :

- Envoyer les données essentielles concernant les transports de patient pour une réalisation adaptée par les Sociétés de transports sanitaires
- Connaître la réponse de la société : acceptation / refus du transport.
- Récupérer des status de réalisation des transports.

2. Description générale de l'interface

A.

La réalisation de cette interface se fera par l'envoi et la réception de fichiers de données respectant les formats précisés dans ce document.

a. Principe :

- Dès affectation d'un transport à une Société privée, un fichier texte sera généré par PTAH (fichier de mission)
- PTAH dépose le fichier généré dans un répertoire de sortie du serveur en DMZ prévu à cet effet.
- Le fichier devra être récupéré par la société privée à l'endroit indiqué.
- Un compte rendu est réalisé dans un fichier de log

- Le régulateur peut envoyer une modification intervenu dans un transport, un fichier de modification d'état de la mission pourra être généré et suivra le même processus que pour une nouvelle mission.
- Un changement de statut sur le mobile de la Société privée génèrera un fichier de suivi qui sera déposé dans un répertoire d'entrée
- PTAH scrutera régulièrement ce répertoire pour récupérer les éléments de suivi de la mission. Ces statuts (Refusé, accepté, Démarré, Terminé, Annulé) seront récupérés dans le logiciel de régulation de PTAH et traités en conséquence.

b. Emplacement des fichiers et modalités d'échange

L'emplacement de ces fichiers est soit sur le serveur PTAH, soit sur un serveur distant (par exemple, un serveur en zone DMZ si les mobiles utilisent Internet).

- Le répertoire de Sortie, est généralement nommé to_MDT (du Système de Régulation PTAH vers Système Mobile MDT) sont localisés sur le Serveur PTAH-DMZ.

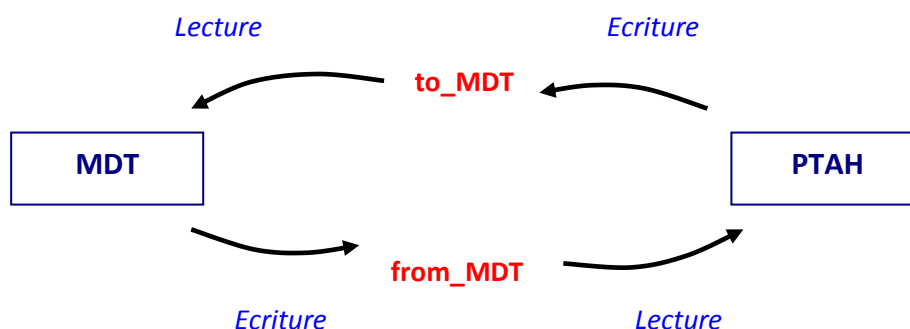
Règle "Interdiction des doublons sortants" :

Avant de déposer un fichier, PTAH doit s'assurer qu'il n'existe pas de fichier de même nom dans aucun le répertoire de sortie.

- Les répertoires d'entrée, sont généralement from_MDT (du Système Mobile MDT vers Système de Régulation PTAH) sont localisés également sur le Serveur PTAH-DMZ et que PTAH va scruter fréquemment.

Dans la plupart des applications, le système Mobile dépose tous les messages envoyés spontanément par les mobiles dans le répertoire « NORMAL ».

Le répertoire « ACK » est utilisé par le système Mobile pour donner le résultat des transactions déposées par le Système PTAH.



Chaque partie doit optimiser ses propres algorithmes de recherche et d'accès aux fichiers, de façon à ne pas pénaliser les accès effectués par l'autre partie.

Les fichiers déposés sont écrits dans le répertoire de destination avec l'extension .TMP de façon à être ignorés par l'autre partie durant les accès en écriture. Une fois que la session d'écriture est totalement terminée, le système termine son écriture en renommant le fichier par l'extension .TXT

À partir de cet instant, le fichier appartient à l'autre partie. La partie qui vient de le déposer n'a plus le droit de le modifier, de l'effacer, ni même de l'ouvrir en lecture.

c. Nom de fichiers

Les fichiers déposés par les intervenants ("Système PTAH" d'une part, "Système Mobile/Passerelle" d'autre part) ont pour extension ".TXT" afin d'être aisément contrôlés à l'aide d'un éditeur de texte standard (Type NOTEPAD de Windows).

Les noms de fichier incluent systématiquement :

- **<Fonction>** : Abréviation de la fonction à traiter (sur 3 caractères) : MIS, MOD, SUI.
- **<Mobile>** : Adresse du mobile (sur 4 caractères, comblés à gauche par des zéros), la Société privée sera considérée comme un mobile par le système PTAH
- **<Extension>** : .TMP puis .TXT

Exemple : MIS_0071.TXT (Mission de transport pour du mobile ou société 71).

d. Format des fichiers

Les Fichiers d'échange respectent le format ascii des fichiers. INI de Windows.

```
[RUBRIQUE1]
clef1=valeur
clef2=valeur
clef3=valeur
...
clefn=valeur
[RUBRIQUE2]
...
...
```

Remarques :

- Les majuscules ne sont pas significatives.
- Il n'y a pas d'espace significatif entre les clefs, le signe Egal et les valeurs.
- Une valeur vide est signalée par le signe "=".
- Les " " ne sont pas obligatoires.

Rubrique d'identification : [PASSERELLE] ou [GESTION]

Pour faciliter la vérification de cohérence, tout fichier débute par cette rubrique, qui identifie l'origine du message.

La clef **objet** précise la rubrique principale

La clef **version** précise le nom du client et le numéro de version en cours.

Important : La clef **Objet** est toujours la première de la rubrique d'identification. D'une manière générale on s'efforcera de respecter l'ordre des clefs, tel qu'il est décrit dans ce document.

Rubrique Principale : [objet]

La rubrique principale, porte le nom de la clef **objet**.

e. Modalités

- Chaque fichier doit être déposé sur le serveur de la manière suivante :
 - o Il est préalablement déposé par l'émetteur dans le répertoire cible sous le nom : **Fonction_Mobile.tmp**
 - o Une fois entièrement déposé et fermé, il sera renommé par l'émetteur en **Fonction_Mobile.txt**
 - o Une fois renommé, l'émetteur s'interdit de le supprimer ou de le remplacer.
- La suppression de ce fichier par l'interface est l'indicateur de réalisation de la prise en compte.

f. Cas d'erreur détecté par le système de la société privée

Nom de fichier non reconnu

To_mdt\ POUBELLE

L'interface Radio de PTAH déplace dans ce répertoire tous les fichiers trouvés dans les répertoires de sortie, avec un **NOM ou une EXTENSION incohérente**.

Exemple de cas d'erreur :

ABC_0001.TXT Code fonction ABC inconnu
ATT_1.TXT Adresse radio dans un format illégal (pas complété à gauche par des zéros)
MSG-0001.TXT Le séparateur "_" n'est pas trouvé
INF_abcd.TXT Adresse radio dans un format illégal
PRO_0001.DOC Extension illégale

S'il existe déjà un fichier de ce nom dans le répertoire POUBELLE, la transaction d'écriture est abandonnée (On considère que l'archivage d'un seul fichier par type d'erreur, suffira pour diagnostiquer sa cause).

Attention : les fichiers à l'extension. **TMP** ne doivent pas provoquer d'erreur

Contenu de fichier non reconnu

Le répertoire d'acquittement "From_mdt\ ACK". Le nom du fichier est complété avec le suffixe **@ERR** (au lieu des suffixes habituels @ACK ou @NAK).

From_mdt\ ACK \ nomoriginal **@ ERR** .TXT

g. Cas d'erreur détecté par le système de PTAH

Nom de fichier non reconnu

Le système de Gestion dépose dans ce répertoire tous les fichiers trouvés dans les répertoires d'entrée, avec un **NOM** ou une **EXTENSION** incohérente.

Exemple de cas d'erreur :

- **ABC_0001-9.TXT** Code fonction ABC inconnu
- **STA_1-9.TXT** Adresse radio dans un format illégal
- **AVC-0001-9.TXT** Le séparateur "_" n'est pas trouvé
- **DCO_abcd-9.TXT** Adresse radio dans un format illégal
- **PRO_0001-9.DOC** Extension illégale
- **STA_0001.TXT** Absence de compteur

S'il existe déjà un fichier de ce nom dans le répertoire POUBELLE, la transaction d'écriture est abandonnée.

Attention : les fichiers à l'extension **.TMP** ne doivent pas provoquer d'erreur

Contenu de Fichier non reconnu

Lorsque le système de Gestion détecte un contenu de fichier non conforme, le fichier est déplacé "intact" dans le répertoire d'acquittement "**From_mdt\ ERREUR**

**From_mdt\ ERREUR ** nomoriginal

S'il existe déjà un fichier de ce nom dans le répertoire ERREUR, la transaction d'écriture est abandonnée.

3. Spécification du format des fichiers d'échange

II.

a. Fichiers missions

Les fichiers de mission sont déposés par le Système PTAH dans le répertoire **<NORMAL>** et respectent le format suivant :

Exemple : Fonction MISSION -> MIS
Société 70 -> 0070
-> **MIS_0070.txt**

Avant de déposer un fichier, le Système PTAH veille à ce que les répertoires de sortie ne comportent pas déjà un fichier du même nom (l'écrasement est interdit).

Description :

[GESTION]

Objet = MISSION
Heure = 15:30
Date = 25/02/2004
Version = Privé V3.08

[MISSION]

Numero = "01234/0123" 10 caractères alphanumériques (**obligatoire**)
Etat = "XXX" NUL = Etat inchangé (réservé pour le changement de texte)
ATT = Attente d'acquittement de la société
EXP = Refus automatique par expiration du délai imparti
REF = Refus par la société
ACC = Acceptation par la société
REJ = Refus après avoir accepté
DEB = Mission débutée par l'ambulancier (prise en charge du patient)
PPR = Patient pas prêt indiqué par l'ambulancier
SLL = Sur le lieu de prise en charge
FIN = Mission terminée par l'ambulancier
ANN = Annulation par le régulateur (Système PTAH)
EFF = Effacement par le régulateur (Système PTAH)

TypeDti = I /ou E, ou S, ou M (transport interne /externe/matière) 1 car alphanum
RangTpt = 1 /pour RDV1 ou 2 pour RDV2 ou 3 pour RDV3 ou R pour Retour
DateRdv = "15/11/2004 14:00"
ModTpt = Brancard /Fauteuil ou Lit ou Marche
Partic = "Oxygène" ; "Perfusion" ; "Lettre" ; "Dossier" (pas de maximum)
Isolement = "Contact" 20 caractères alphanumériques
BMR = "O" /ou N
Accompagnant = "Famille" ; "Infirmière Privée"
NbAcc = 2 Nombre total d'accompagnant
ObsObs = "malade agité" 60 caractères alphanumériques
ObsExa = "+ rx en suivant" 60 caractères alphanumériques
// Si RangTpt="R"
ObsRet = "Prendre patient salle 12" 60 caractères alphanumériques

[Patient]

NumIpp = 743513
NumDos = 135144762
Titre = Mr /Mme /Mlle
NomMal = "DUPOND" 20 caractères alphanumériques
Prenom = "Jacques" 20 caractères alphanumériques
Categ = Adulte /Enfant ou Bébé
Poids = 120 en Kg
Age = 29
DateNai = 01/04/1981
NumCh = 200
NumLit = 200P

[DEPART]

CodEtb = "42" 10 caractères alphanumériques
NomEtb = "X ROUSSE" 10 caractères alphanumériques
LibEtb = "HOPITAL CROIX ROUSSE" 40 caractères alphanumériques
Adr = "126, rue..." 70 caractères alphanumériques
CPost = 69000 6 caractères alphanumériques

Ville	=	"LYON"	40 caractères alphanumériques
CodBat	=	"000"	10 caractères alphanumériques
NomBat	=	"XR Y"	10 caractères alphanumériques
CodSrv	=	"24457"	10 caractères alphanumériques
NomSrv	=	"SCANNER XR"	10 caractères alphanumériques
LibSrv	=	"SCANNER CROIX ROUSSE"	40 caractères alphanumériques
EtgSrv	=	"1"	6 caractères alphanumériques
TelSrv	=	"123456"	16 caractères alphanumériques
Date	=	"15/11/2004 13:15"	Date et Heure de départ prévue

[ARRIVEE]

CodEtb	=	"42"	idem que pour le service départ
NomEtb	=	"X ROUSSE"	
LibEtb	=	"HOPITAL CROIX ROUSSE"	
Adr	=	"126, rue..."	
CPost	=	69000	
Ville	=	"LYON"	
CodBat	=	"000"	
NomBat	=	"XR A"	
CodSrv	=	"24114"	
NomSrv	=	"DELORE XR"	
LibSrv	=	"Service DELORE CROIX ROUSSE"	
EtgSrv	=	"1"	
TelSrv	=	"123456"	
Date	=	"15/11/2004 13:50"	Date et Heure d'arrivée prévue

b. Modification d'états de mission

Les fichiers de modification d'état de mission sont déposés par le Système PTAH dans le répertoire <NORMAL> et respectent le format suivant :

Exemple :	Fonction MODIFETAT	-> MOD_
	Société 70	-> 0070
		-> MOD_0070.txt

Ce message est déposé par le système PTAH lorsque la régulation souhaite modifier l'état d'une mission.

Description :

[GESTION]

Objet	=	MODIFETAT
Heure	=	15:30
Date	=	25/02/2004
Version	=	Privé V3.08

[MODIFETAT]

Etat	=	"XXX"	(voir Etat dans Mission)
Numero	=	"01234/0123"	10 caractères alphanumériques (obligatoire)

Le codage du numéro est le même que pour l'envoi de mission avec état.

c. Suivi de mission

Les fichiers de suivi de mission sont déposés par le Système Mobile/Passerelle dans le répertoire <**NORMAL**> et respectent le format suivant :

Exemple : Fonction SUIVI -> SUI
 Société 70 -> 0070
 -> **SUI_0070.txt**

Description :

[PASSERELLE]

Objet = ETAT_SUIVI
Heure = **15:30**
Date = **25/02/2004**
Version = **Privé V3.08**

[ETAT_SUIVI]

Etat = "xxx" (voir Etat dans Mission)
Numero = 01234/0123 (10 caractères alphanumériques)
Date = 09/11/2004
Heure = 19:30
Comment = "xxx : yyyy" (3 caractères alphanumériques pour le code de refus puis 128 caractères alphanumériques max pour le commentaire du refus)

4. Protocole d'échange de l'information

a. Principes généraux

Le protocole de communication choisi est http.

Il y a deux logiciels à considérer :

- **SrvPtah** : le logiciel de communication côté serveur Ptah
- **AppClient** : le logiciel de communication côté client

La sécurité des échanges est assurée par deux mots de passe :

- psw1 : utilisé pour vérifier l'identité du client
- ~~psw2 : utilisé pour crypter les échanges~~

Le principe de communication s'appuie sur l'utilisation de deux canaux de communications pour effectuer un "**push-pull**":

Le canal "Push" est un canal permanent, facultatif, utilisé par Ptah pour prévenir immédiatement le client qu'il a sa disposition de nouvelles données. Ce canal est ouvert à l'initiative de l'application cliente, qui se met en écoute permanente.

- 1ère phase : AppClient ouvre le canal Push et s'identifie auprès de SrvPtah.
- 2ème phase : SrvPtah utilise ce canal pour indiquer à AppClient qu'elle dispose d'informations nouvelles.

Le canal "Pull" est un canal volatile, obligatoire, qui sert à transmettre les informations, de Ptah vers le client ou inversement. Ce canal est ouvert à l'initiative de l'application cliente :

- dès que le client possède une nouvelle information qu'il veut transmettre à Ptah
- dès que le client est prévenu par le canal Push qu'il dispose d'une nouvelle information sur Ptah
- régulièrement pour vérifier si Ptah possède des informations pour le client

Le canal ne reste ouvert que pendant la durée des échanges, il est fermé dès qu'il n'y a plus rien à échanger.

A noter que le canal Pull est utilisé pour échanger plusieurs fichiers dans la même session, et qu'il ne doit pas être fermé après chaque échange. C'est la raison pour laquelle l'identification du client n'est nécessaire qu'une seule fois, et que cette identification reste valable tant que la session est ouverte.

Il ne faut pas en conclure que ce canal peut rester ouvert en permanence : dès que plus aucun échange de fichier n'est constaté, le canal sera fermé automatiquement par Ptah.

b. Description du principe d'échange

Le principe d'échange est le suivant :

- l'application cliente AppClient ouvre le canal "Push" pour se connecter et s'identifier sur le serveur SrvPtah.
- l'application cliente peut à tout moment interroger SrvPtah en ouvrant le canal "Pull" pour :
 - envoyer des informations vers SrvPtah
 - demander s'il y a des informations à recevoir de la part de SrvPtah, et récupérer ces informations.
 - une fois les échanges terminés, l'application cliente ferme le canal "Pull".
- l'application serveur utilise le canal "Push" pour indiquer à AppClient qu'il y a des informations à récupérer.

c. Protocole de communication du canal « Push »

(1) Identification : App_Listen

Syntaxe :

```
App_Listen
  mod      App                // modèle de communication
  num      0101              // numéro de société
  psw      MOUTON            // password 1
```

Exemple :

Requête client :

`http://localhost:4004/App_Listen?mod=App&num=0101&psw=MOUTON`

Réponse serveur :

"ok\n" : immédiatement
puis : **"WakeUP [...] \n"** dès qu'il y a des fichiers à récupérer
(boucle de surveillance de 5 sec)
puis : **"Continue [...] \n"** toutes les 5 minutes de silence

d. Protocoles de communication du canal « Pull »

(2) Demande de la liste des messages à récupérer : App_Get_Files

Syntaxe :

```
App_Get_Files :  
    mod      App           // modèle de communication  
    num      0101          // numéro de société  
    psw      MOUTON        // password 1
```

Exemple :

Requête client :

`http://localhost:4004/App_Get_Files?mod=App&num=0101&psw=MOUTON`

Réponse serveur :

"NbFic=0\n" si aucun message n'est disponible
rupture du canal "Pull"

Réponse serveur :

"NbFic=2\n" si des messages sont disponibles
"\to_MDT\Urgent\MIS_0101-1.app.txt\n"
"\to_MDT\Normal\MIS_0101-2.app.txt\n"

(3) Demande de récupération des messages : App_Get_File

Le principe consiste à :

- Read : lire un nouveau message
- Dele : effacer un message lu dans une récupération précédente
- Stop : arrêter la récupération

On ne peut lire qu'un message à la fois.

Syntaxe :

`App_Get_File :`

```

Read      \to_MDT\Urgent\MIS_0101-1.app.txt
           // nom du message à récupérer de SrvPtah vers AppClient
Dele      \to_MDT\Normal\MIS_0101-2.app.txt
           // nom du message à détruire sur SrvPtah
Stop
           // arrêt de récupération de message

```

Par exemple pour lire un message :

```
\to_MDT\Urgent\MIS_0101-1.app.txt
```

1ère requête :

```
http://localhost:4004/App_Get_File&Read=\to_MDT\Urgent\MIS_0101-1.app.txt
```

2ème requête :

```
http://localhost:4004/App_Get_File&Dele=\to_MDT\Urgent\MIS_0101-1.app.txt&Stop
```

Par exemple pour lire deux messages :

```
\to_MDT\Urgent\MIS_0101-1.app.txt et %Normal%\MIS_0101-2.app.txt
```

1ère requête :

```
http://localhost:4004/App_Get_File&Read=\to_MDT\Urgent\MIS_0101-1.app.txt
```

2ème requête :

```
http://localhost:4004/App_Get_File&Dele=\to_MDT\Urgent\MIS_0101-1.app.txt&Read=\to_MDT\Normal\MIS_0101-2.app.txt
```

3ème requête :

```
http://localhost:4004/App_Get_File&Dele=\to_MDT\Normal\MIS_0101-2.app.txt&Stop=true
```

Exemple :

Requête client :

```
http://localhost:4004/App_Get_File?Read=\to_MDT\Normal\MIS_0101-2.app.txt&Dele=\to_MDT\Urgent\MIS_0101-1.app
```

Réponse serveur :

- . destruction du message \to_MDT\Urgent\MIS_0101-1.app.txt
- . envoi du message \to_MDT\Normal\MIS_0101-2.app.txt
 - sous forme cryptée (avec le password 2)
 - longueur du message dans http.ContentLength

(4) Envoi de message vers SrvPtah : App_Put_File

Syntaxe :

```

App_Put_File
  mod      App
  num      0101
  psw      MOUTON
  Writ     nom du message à envoyer

```

Exemple :

Requête client :

`http://localhost:4004/App_Put_File?mod=App&num=0101&psw=MOUTON&Writ=\from_MDT\Normal\MIS_0101-2.app.txt`

`http.ContentLength` : longueur du message cryptée par le password 2

Réponse serveur : `http 200 OK "ok\n"`