Implémentation de la RFC 1086
Mise au format XML de fichiers bancaires

Felip Manyé i Ballester

19 mai 2009

Plan

La société AFSOL

- SAS basée à Tecnosud, Perpignan, Catalogne Nord
- créée suite à la disparition de l'antenne de MoneyLine
- fondateurs : MM. Alain Maravitti et Philippe Carreras
- 5 à 6 personnes

Historique

Novembre 1987	première version de l'application de télécollecte STAP
Mars 2005	création de la société AFSOL
Juin 2005	achat de la branche d'activité STAP à MoneyLine
Décembre 2005	entrée du GICM et de Lyra Network dans le capital

Monétique

Définition

Traitement informatisé des transactions financières : carte bancaire, mais aussi chèque

support chèque,
CB : ordinateur spécialisé inviolable

émetteur organisme financier qui met un support à disposition du porteur

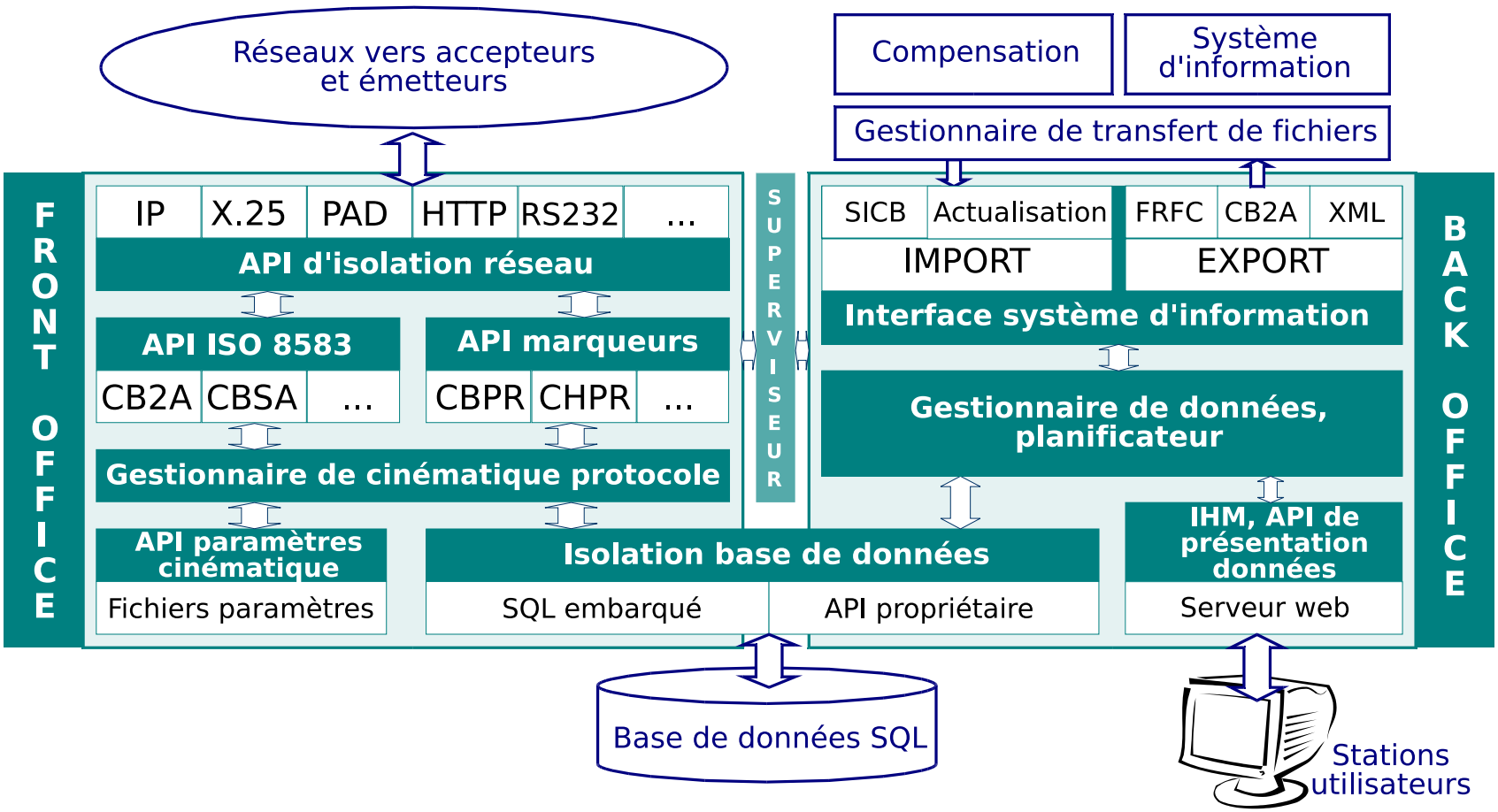
porteur personne en possession d'un support

acquéreur banque domiciliaire de l'accepteur

accepteur entreprise acceptant un moyen de paiement
TPE : Terminal de Paiement Électronique

STAP (1/2)

Serveur Télécollecte Acquéreur Paiement



STAP (2/2)

Serveur Télécollecte Acquéreur Paiement

Domaine d'intervention : « front office »

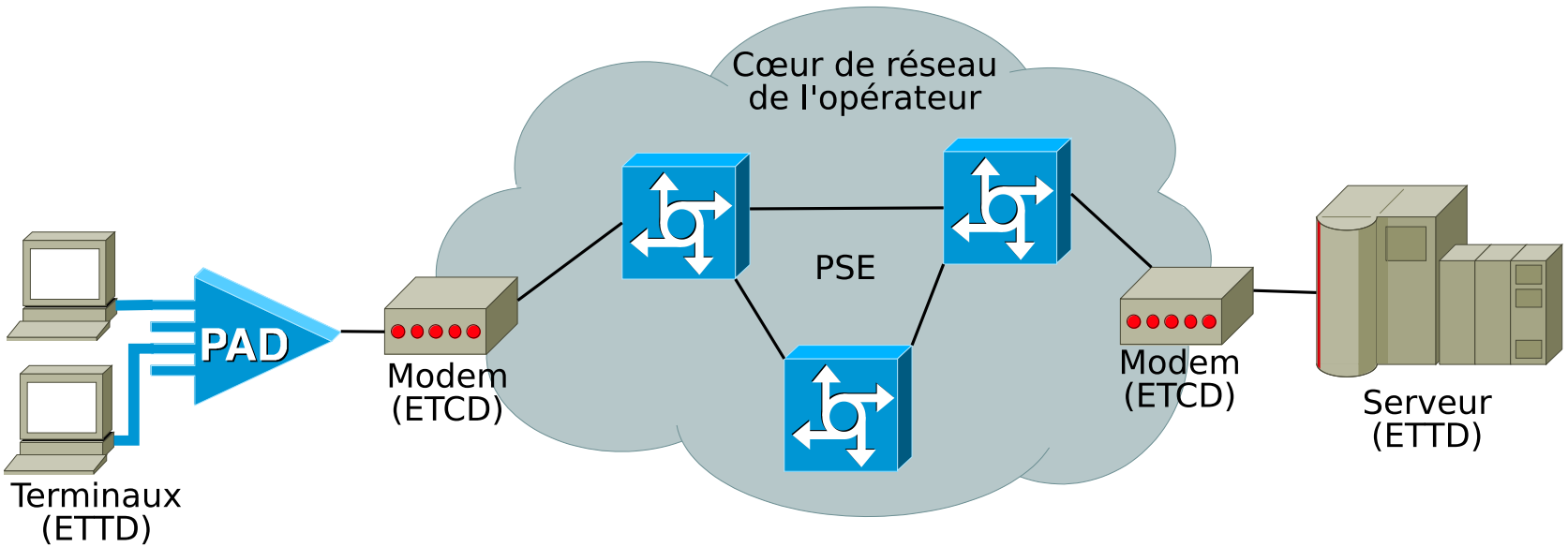
Couche réseau Implémentation de la RFC 1086 : protocole de conversion entre réseaux X.25 et TCP/IP
Étude préalable de la technologie SSL

Fichiers bancaires Mise au format XML de fichiers bancaires, les « bruts », contenant des remises financières (ensemble de transactions)

Réseaux X.25 (1/2)

- X.25 : protocole réseau par commutation de paquets en mode point à point, normalisé par l'UIT et ISO
- définit l'interface entre ETTD (Équipement Terminal de Traitement de Données) et ETCD (Équipement Terminal de Circuit de Données)
- le réseau français Transpac date de 1978 ; largement utilisé pour les applications financières depuis cette date (Distributeurs de billets, TPE...)

Réseaux X.25 (2/2)



Quel futur pour X.25 ?

Une technologie obsolescente

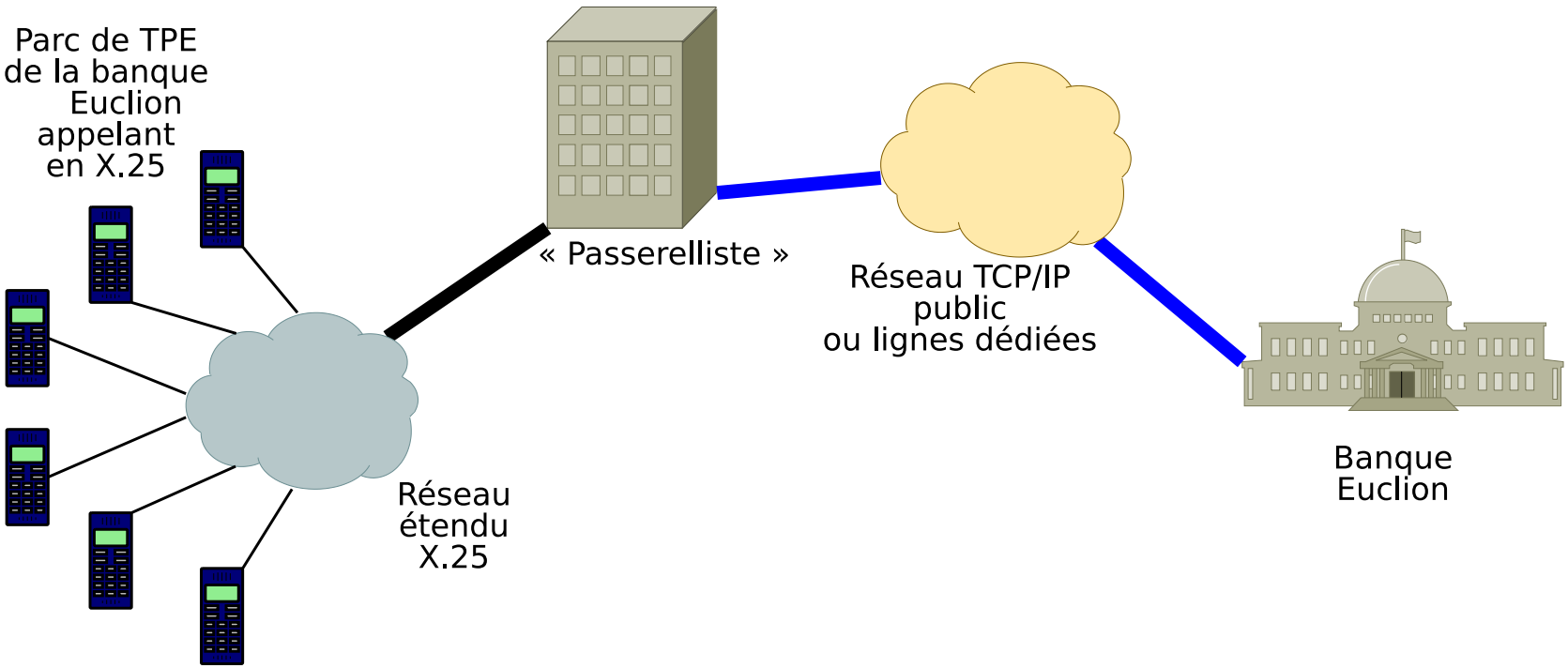
- débit limité, mode de tarification
- pensé pour des terminaux passifs, l'intelligence se trouve au centre du réseau

Conversion X.25/TCP

Deux solutions envisageables pour l'interconnexion le temps de la migration

- « les passerellistes » se chargent d'acheminer les transactions
- certains dispositifs réseau et protocoles de conversion permettent de rediriger soi-même les flux

« Passerelliste »

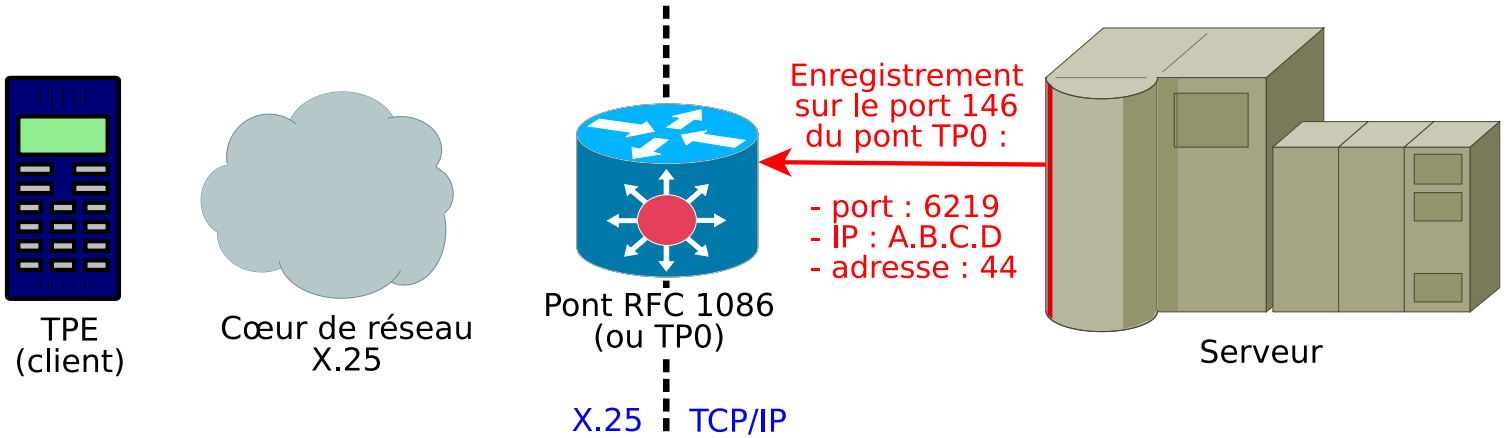


Qu'est-ce que la « RFC 1086 » ?

- un protocole ouvert décrit par l'IETF dans un document intitulé *ISO-TP0 bridge between TCP and X.25*, faisant suite à la RFC 1006, *ISO Transport Service on top of the TCP*
- elle décrit un « pont TP0 » permettant la conversion entre réseaux TCP/IP et réseaux X.25, et est implémentée sur quelques routeurs hybrides TCP/X.25
- dans l'esprit de la RFC 1006, le but est de permettre l'utilisation des couches ISO sur TCP/IP en attendant le développement des couches basses ISO
- depuis la fin des protocoles ISO (1996), le but est au contraire d'abandonner les protocoles ISO

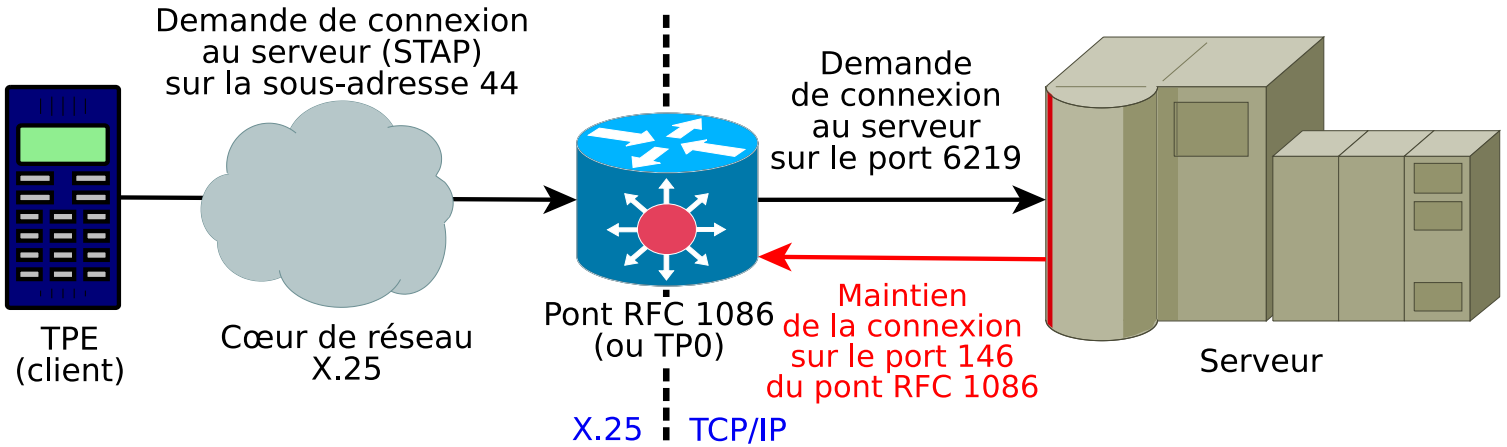
La RFC 1086, comment ça marche ? (1/2)

- conversion entre les deux réseaux assurée par un « pont »
- dialogue à l'initiative de l'hôte TCP : phase d'« enregistrement » sur le port TCP 146 du pont
  - → couple adresse IP/port TCP
  - → mode client ou serveur
  - → sous-adresse X.25, *etc.*



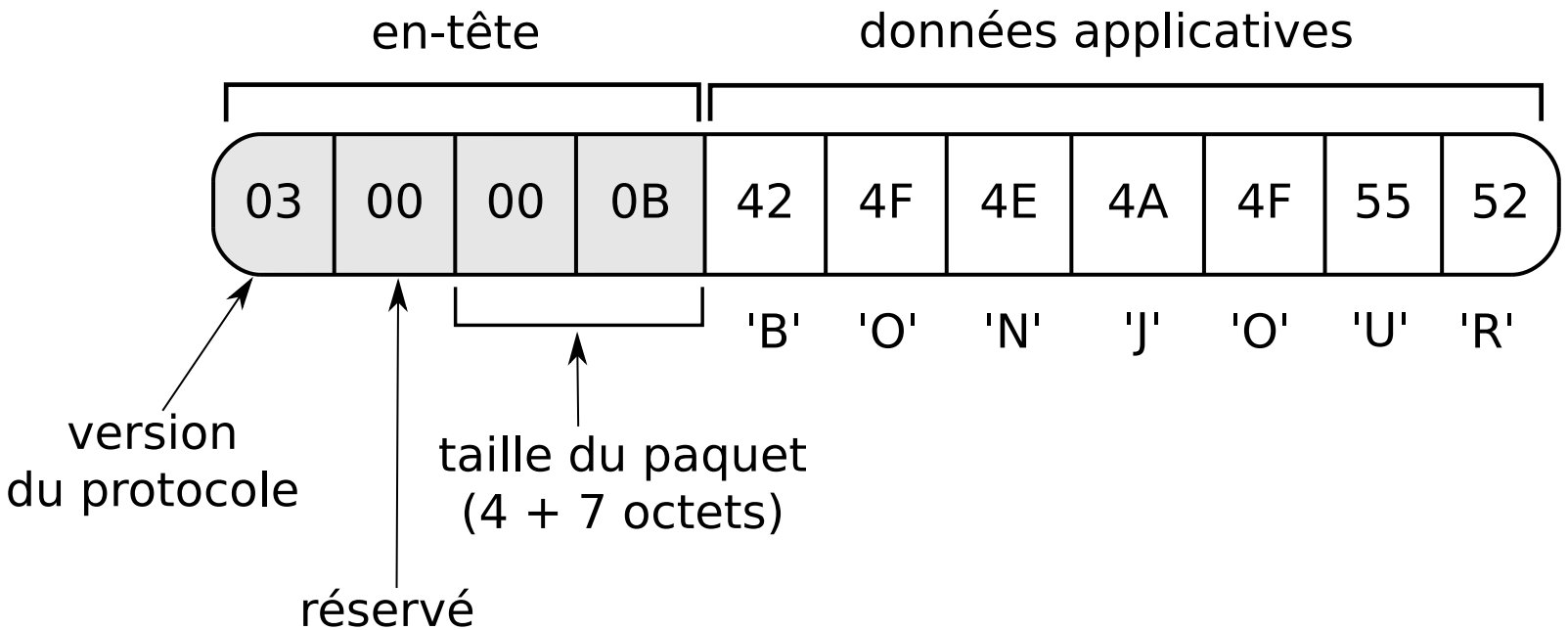
La RFC 1086, comment ça marche ? (2/2)

- translation effectuée par le pont *tant que la connexion sur le port 146 est maintenue*
- en mode serveur, les données à destination de la sous-adresse 44 sont extraites du paquet X.25, puis encapsulées dans TCP, adjointes d'un en-tête décrit dans la RFC 1006
- ces données sont renvoyées sur le couple adresse/port déclaré



Format des paquets

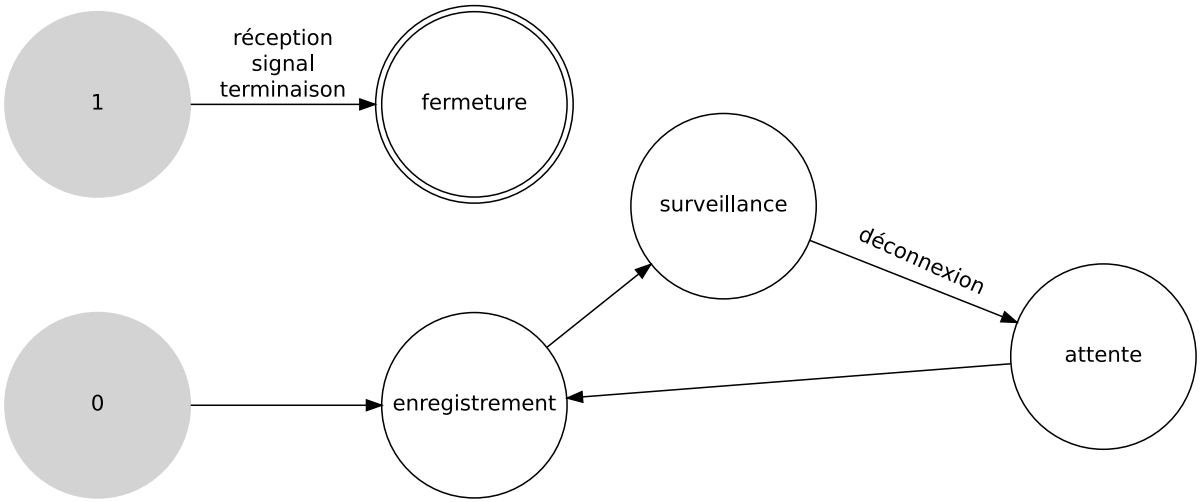
La RFC 1006, dont s'inspire la RFC 1086, impose un format de paquets particulier côté TCP, quel que soit le mode (client ou serveur).



Mode RFC1086

L'hôte TCP héberge le serveur de télécollecte (STAP)

- phase d'enregistrement sur un pont TP0
- puis gestion du format des paquets selon la RFC 1006
- maintien de la connexion sur le port 146 : appel aux primitives « TCP Keepalive » du système



Mode RFC1086 : limitations

Données d'appel

En X.25, le paquet de demande d'ouverture de session peut comporter 16 octets de données. Ce comportement n'admet pas de correspondance en TCP/IP.

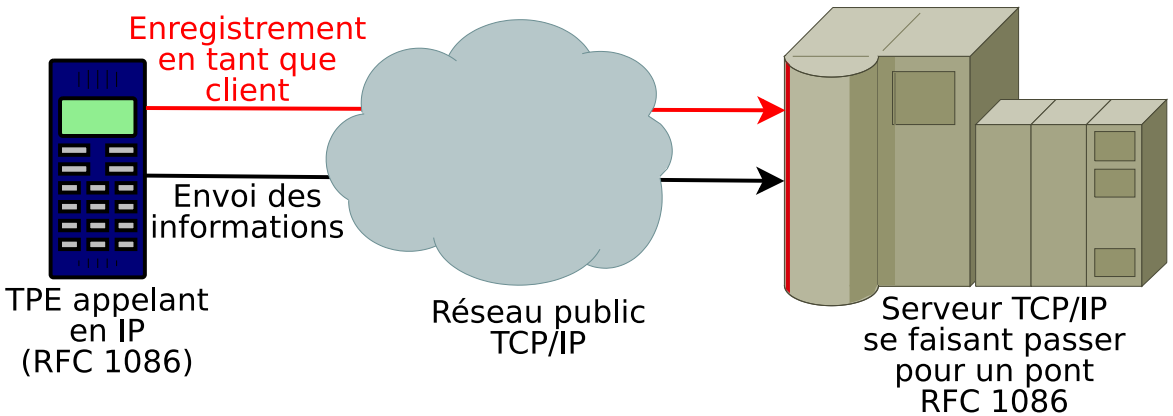
- les données d'appel sont perdues, ce qui rend inutilisable le protocole bancaire CB2A avant sa version 1.2
- CBPR et CB2A $\geq$ 1.2 restent utilisables
- aucun dispositif réseau ne sait les traiter, ni aucun autre protocole ouvert

Mode EMULRFC1086

Tout est en IP, mais le serveur se fait passer pour un pont TP0 auprès d'un terminal de paiement TCP/IP utilisant le dialogue RFC 1086 :

- réception du paquet d'enregistrement du TPE
- gestion du format des paquets selon la RFC 1006

Mode également compatible avec la variation « RFC 1086 Concert » du GIE Cartes Bancaires.

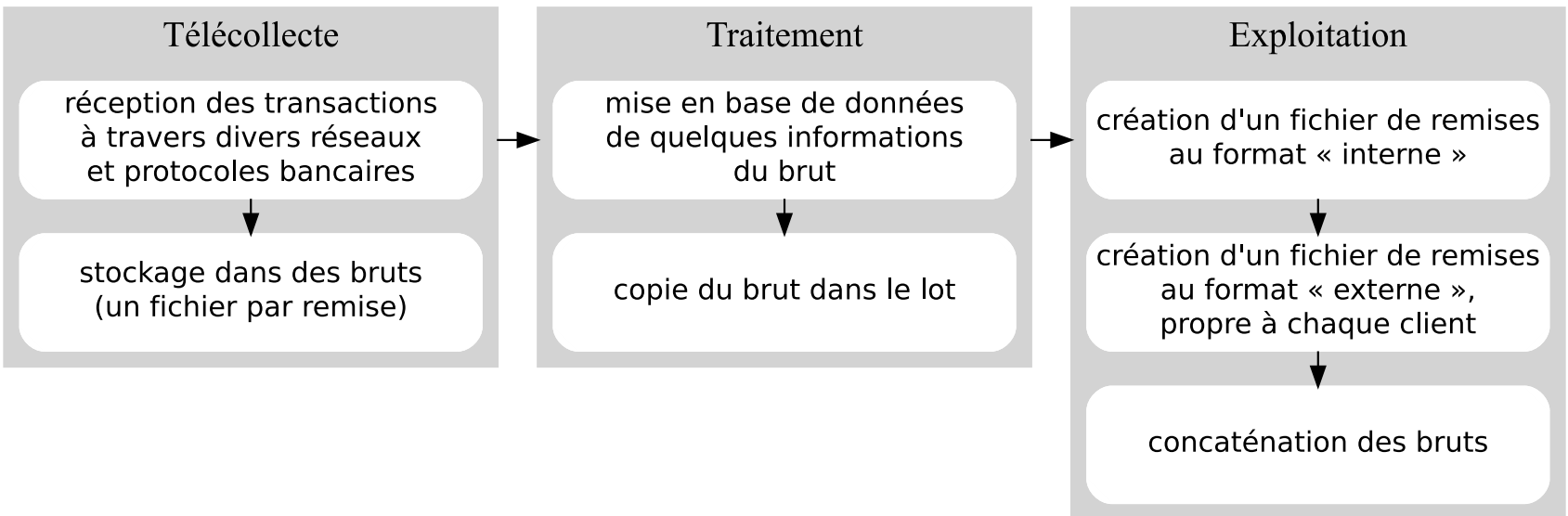


Quelques mots sur SSL

En mode EMULRFC1086, les données sont échangées sur un réseau TCP/IP *public* (à travers le FAI du commerçant). L'emploi du système cryptographique SSL (Secure Socket Layer) est recommandé, ce qui a donné lieu :

- à un travail de recherche sur ce protocole, suivi d'une présentation auprès de nos collègues (notions cryptographiques, certificats, protocole SSL)
- aux premiers essais de communication chiffrée et de déploiement de certificats sur un terminal de test
- à une première vue d'ensemble du matériel et des logiciels nécessaires (stunnel, modules matériels de sécurité, infrastructure à clefs publiques)

Chaîne de traitement des transactions



Protocoles bancaires

Protocoles en production

CBPR protocole très simple, obsolète depuis l'an 2000, mais encore très utilisé pour des applications privatives

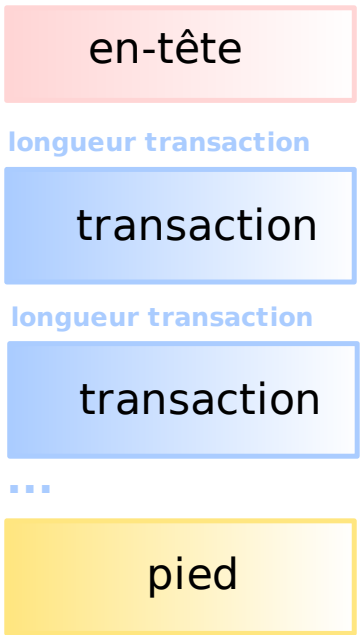
CB2A le plus courant actuellement, basé sur ISO 8583 et maintenu par le GIE CB

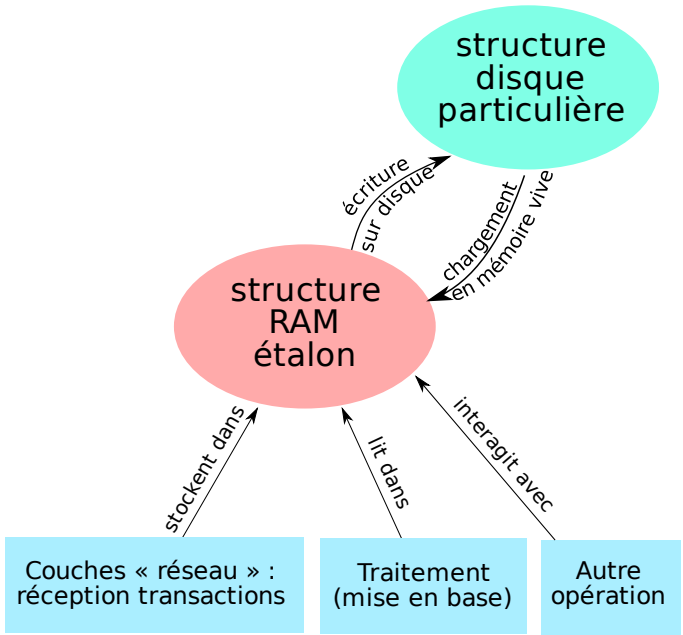
EPAS : un protocole européen en gestation

- basé sur ISO 20022, utilisera XML
- élaboré par divers acteurs du SEPA (Single Euro Payment Area)
- une fois achevé, les messages seront publiés sous la forme d'un schéma XML dans le catalogue UNIFI, maintenu par SWIFT
- son arrivée motive le passage à XML

Ancienne méthode d'écriture des bruts

- fichiers ASCII fixes et positionnels
- en mémoire vive, une transaction est stockée dans une structure C de référence (la même pour tous les protocoles)
- sur disque (dans les bruts), on enregistre une structure C propre à un protocole donné



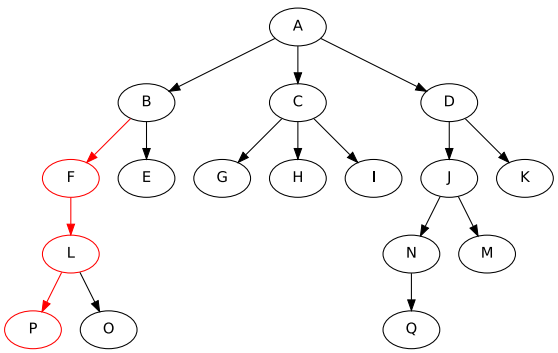


Mode de compatibilité

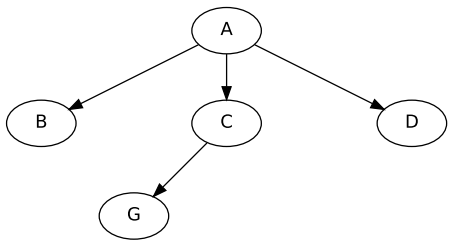
- on peut être amené à générer un brut XML à partir d'éléments issus d'anciens protocoles
- pour assurer la compatibilité et permettre une migration progressive de l'application STAP, on conserve la structure de référence (la structure particulière, elle, disparaît)
- un dispositif permettant de concilier les deux modes d'écriture a donc dû être mis en place
- à l'avenir, tout sera traité directement en XML, en utilisant la partie purement XML de la « bibliothèque » logicielle développée

Génération des bruts

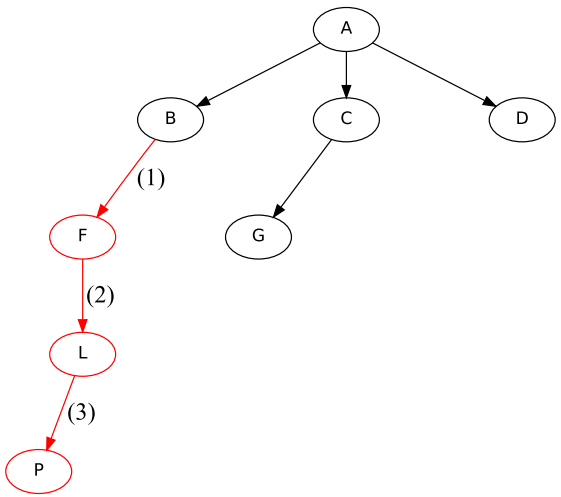
On considère le schéma W3C décrivant les futurs messages UNIFI comme le dictionnaire des données, au-delà du simple rôle de validation d'un schéma. Les bruts dérivent d'un gabarit obtenu par transformation XSLT de ce schéma.



gabarit issu d'une transformation XSLT du schéma XML



brut avant ajout de l'élément P



brut après recopie de l'élément

Lien avec la structure de référence

Comment concilier deux mondes

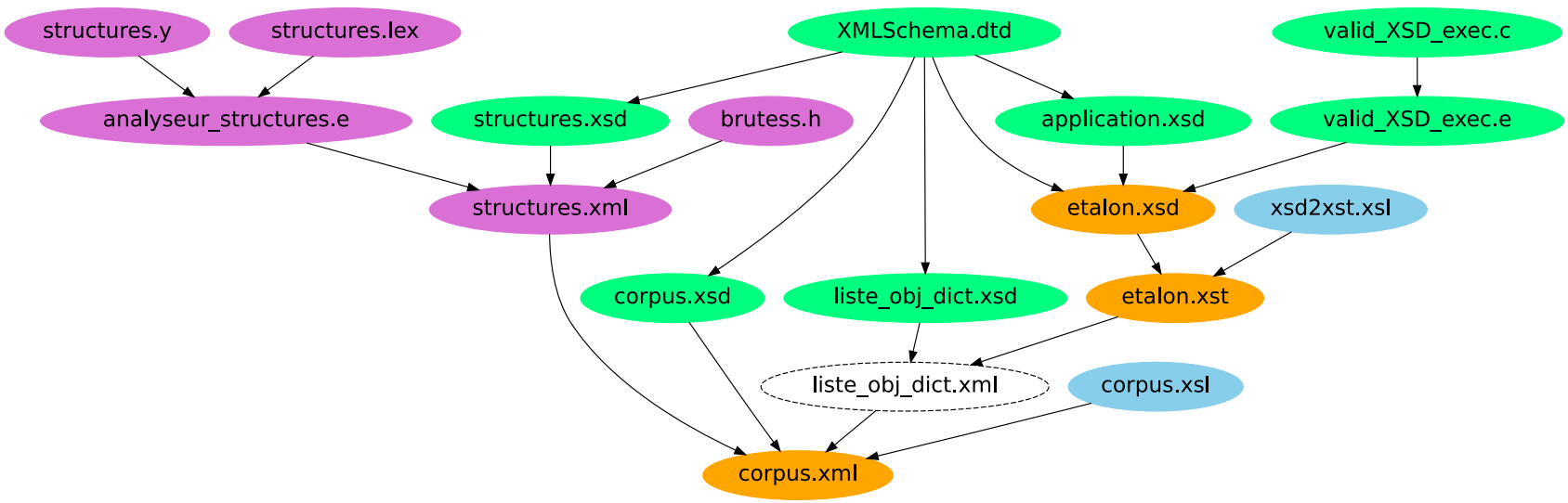
- la structure C est décrite dans un fichier XML (obtenu par analyse syntaxique du fichier source avec Flex/Bison)
- on place dans le schéma XML des annotations faisant référence à ce document (balise `xs:annotation`)
- le gabarit obtenu par transformation XSLT du schéma XML tient compte de tous ces éléments
- on utilise le langage XPath pour rechercher des éléments dans le gabarit ; en pratique, on « précompile » tous les dictionnaires dans ce que l'on nomme un « corpus »

Intégration à la chaîne de compilation (1/2)

On met à contribution le système de makefiles servant à compiler STAP ; plusieurs étapes aboutissent au corpus :

- on réalise une analyse syntaxique du fichier source contenant la structure C ⇒ `structures.xml`
- à partir du dictionnaire annoté, on obtient un gabarit par transformation XSLT ⇒ fichier `.xst`
- par transformation XSLT de chacun des gabarits et de `structures.xml`, on arrive au corpus ⇒ `corpus.xml`
- toutes ces étapes sont contrôlées (validation de chacun des fichiers)

Intégration à la chaîne de compilation (2/2)



Conclusion

- un TFE ayant permis d'aborder de nombreux concepts
- les deux implémentations de la RFC 1086 abouties (malgré les limitations du mode `RFC1086` induites par les données d'appel)
- de bonnes bases jetées pour le déploiement de SSL
- un système d'écriture des bruts au format XML permettant une migration progressive, ouvrant la voie au protocole EPAS
- l'adoption de PCI DSS facilitée

Merci de votre attention !