Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PREMIER MINISTRE

Agence pour
le Développement
de l'Administration
Electronique

juin 2005

Protocole d'échanges pour l'administration électronique

Appel à commentaires

Version 1.0

Références :

ID	Titre	Version	Auteur	Description
[COMP]	Etude comparative des normes FAST, IDA eLink et ebMS2	2	ADAE	Etude comparative des protocoles à disposition de l'ADAE
[FAST]	Norme d'échange FAST	1.1	CDC	Définition du protocole d'échange utilisé par la plate-forme FAST
[OSCI]	OSCI Transport : Design Principles, Security Objectives and Mechanisms	05/02	OSCI Leitstelle	Description des principes et mécanismes de la version 1.2 du protocole transport OSCI
[ELINK]	IDA eLink Specification V2	1.0	European Dynamics	Spécification de la deuxième version du protocole eLink
[EBMS2]	Message Service Specification Version 2.0	2.0	ebXML OASIS	Spécification de la deuxième version du protocole d'échange de ebXML
[MTOM]	SOAP Message Transmission Optimization Mechanism (W3C Proposed Recommendation)	01/05	W3C	http://www.w3.org/TR/soap12-mtom/
[XOP]	XML-binary Optimized Packaging (W3C Proposed Recommendation)	01/05	W3C	http://www.w3.org/TR/xop10/
[TEDECO]	Site Internet	-	GIE TEDECO	http://www.admiroutes.asso.fr/action/theme/pouvoirs/tedeco/index.htm
[SANTE]	Les protocoles de communication dans le secteur socio-sanitaire	-		Etude du ministère de l'emploi et de la solidarité

SOMMAIRE

1 INTRODUCTION

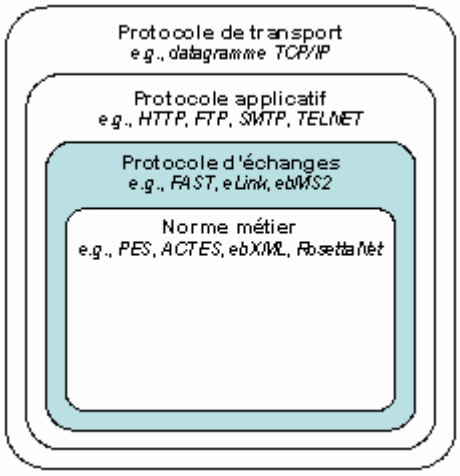
1.1 PRESENTATION GENERALE

Le présent appel à commentaires a pour objet de recueillir des observations et des recommandations en matière de protocoles d'échanges de messages informatiques entre applications. Il s'agit de poser les bases de ce qui deviendra, à terme, le protocole d'échanges de l'administration électronique (programme ADELE).

Ce protocole véhiculera des messages entre différents intervenants, tels que les ministères, les établissements publics, les collectivités locales, certains partenaires privés, les administrations européennes… Il concerne donc les échanges entre acteurs de l'administration électronique et n'a donc pas vocation à régir les échanges internes de chaque système d'information.

Ce protocole devra fournir un certain nombre de fonctionnalités, énumérées dans la suite de ce document, tout en respectant les normes et standards actuels et en devenir, de manière à assurer sa pérennité et son interopérabilité.

A noter enfin que ce protocole est totalement indépendant des données métier qu'il véhicule, aussi bien en termes de contenu que de format. Il n'est donc pas comparable à des protocoles tels que PES de la DGCP ou ACTES de la DGCL. Il est également indépendant du protocole de transport sous-jacent (couches basses OSI). Il n'est donc pas comparable à des protocoles tels que HTTP, FTP, SMTP, TELNET, etc.



1.2 OBJET DE L'APPEL A COMMENTAIRES

L'objet du présent appel à commentaires est de faire un point sur les fonctionnalités supposées attendues d'un protocole d'échange de données ADELE et d'en dégager une cible à atteindre à horizon de 18 mois en essayant de concilier :

- l'attente des organismes qui puisse refléter l'expression des besoins et la stratégie de migration ou d'évolution de leur système d'information,
- la politique industrielle et commerciale des opérateurs et offreurs de solution dans le domaine des réseaux de transports ou des réseaux à valeur ajoutée,
- la capacité des systèmes existants à s'adapter à cette évolution.

Aucun des protocoles candidats mis en « compétition » ne répond complètement aujourd'hui aux besoins identifiés pour le programme ADELE. Chacun nécessite des adaptations ou comporte des fonctionnalités qui semblent inutiles dans les cas d'usages identifiés pour le programme ADELE.

Il a donc paru plus opératoire de présenter en restant factuel, une comparaison des fonctionnalités par besoin en s'appuyant sur le protocole européen eLink en cours de validation.

Une première partie (chapitre 3) fait le point sur les motivations fonctionnelles qui sous-tendent la mise en place de ce protocole. Il s'agit donc d'affiner et de recenser les enjeux et besoins fonctionnels.

Puis (chapitre 4) une transcription technique des besoins identifiés propose différentes pistes quant à l'implémentation du protocole. Pour chaque besoin fonctionnel identifié, une ou plusieurs questions sont posées et nécessitent une expertise technique quant à la pertinence de la réponse.

Ensuite (chapitre 5) une brève présentation des fonctionnalités techniques nécessaires à la mise en œuvre du protocole est effectuée. Il s'agit de déterminer les critères d'aptitude que doit remplir une plate-forme logicielle et matérielle pour mettre en œuvre le protocole d'échanges.

Enfin (chapitre 6) différentes perspectives en rapport avec le protocole d'échanges sont exposées de manière à anticiper les besoins et contraintes futurs.

1.3 PUBLIC VISE

Le chapitre 3, enjeux et contextes, s'adresse en priorité aux responsables d'applications qui doivent échanger des données avec des organismes externes ainsi qu'aux urbanistes qui définissent des politiques d'échanges inter-applicatifs.

Le chapitre 4, description du protocole, s'adresse en priorité aux responsables techniques des différentes administrations et partenaires en vue d'interopérer ainsi qu'aux architectes techniques travaillant à la mise en place d'infrastructures d'échanges.

Le chapitre 5, hub d'échanges, s'adresse en priorité à ces mêmes architectes ainsi qu'aux éditeurs de solutions logicielles et aux sociétés de conseil compétentes sur le sujet.

Le chapitre 6, perspectives, s'adresse quant à lui à l'ensemble des publics précités.

2 CONTEXTE

2.1 RAPPEL DE LA PROBLEMATIQUE CONCERNANT LES ECHANGES

Dans le cadre d'une démarche d'informatisation et d'urbanisation des échanges entre les différents acteurs de l'administration électronique, l'ADAE souhaite disposer pour le programme ADELE d'un protocole d'échange de données qui puisse répondre à la majorité des cas d'usage.

A titre d'exemple, parmi les projets concernés par les échanges de documents et qui sont candidats à ce besoin, on peut citer les projets de dématérialisation Hélios (dématérialisation des pièces comptables et fiches de paie) pour le ministère des Finances et de l'Industrie ou encore Actes (contrôle de légalité dématérialisé) pour le ministère de l'Intérieur. Ces deux projets s'appuient sur le protocole d'échanges FAST développé par la CDC.

Les projets de ce type produisent des flux comprenant une grande quantité d'échanges de données de tout type. Ainsi, les documents transmis peuvent être des documents XML, PDF, des images JPEG, etc. Par ailleurs, ces échanges sont soumis à des contraintes de sécurité importantes : les documents doivent être signés (identification de l'émetteur et intégrité du contenu) voire chiffrés (confidentialité des données).

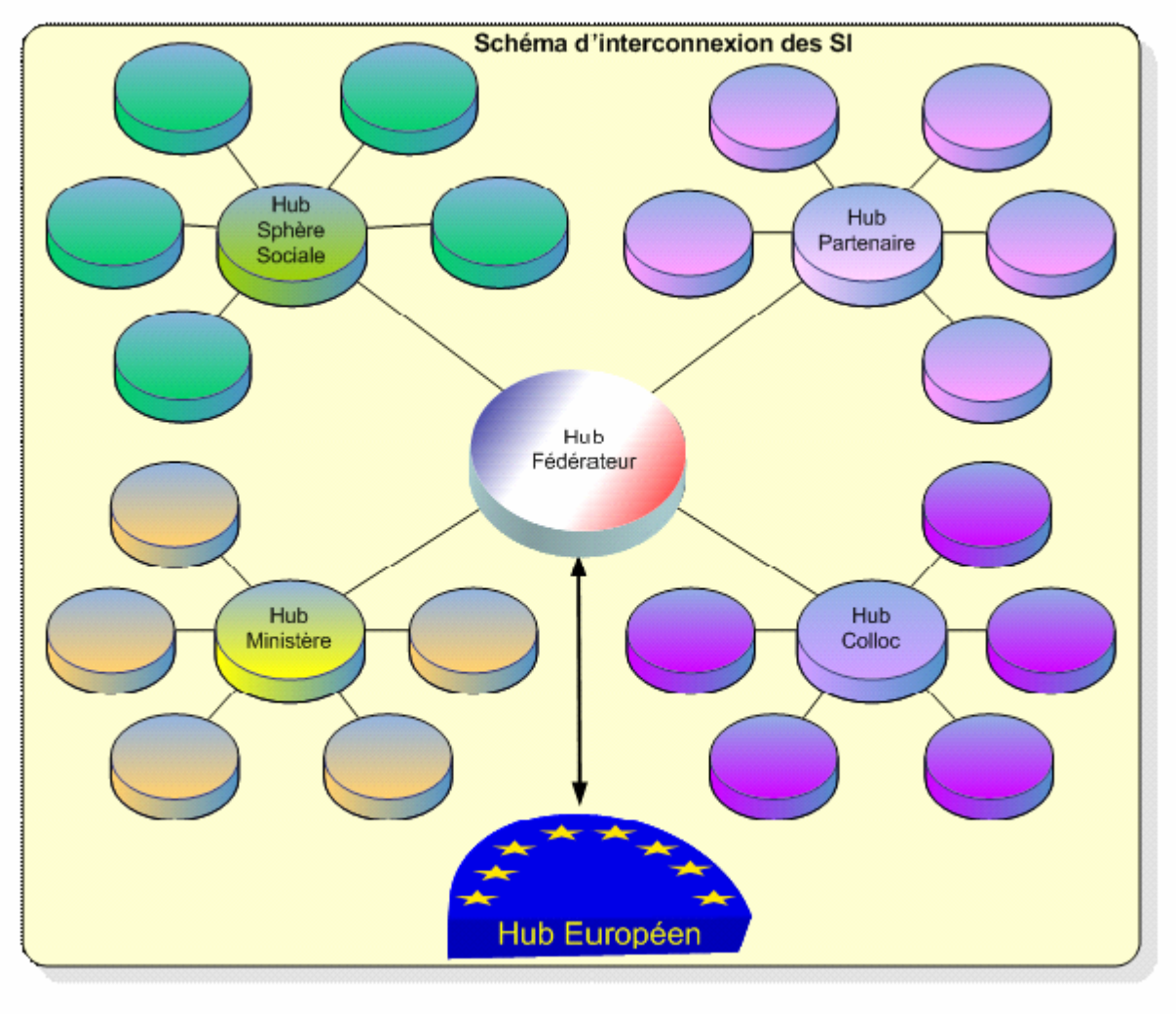
2.2 ARCHITECTURE GENERALE DES ECHANGES ENTRE LES SI DE L'ADMINISTRATION ET PARTENAIRES AFFILIEES

Ce chapitre a pour but de décrire l'architecture générale d'articulation des SI de l'administration et des partenaires affiliés. Cela comprend :

- les ministères,
- les collectivités locales,
- la sphère santé-social,
- les différents autres partenaires rendant un service public ou assimilé.

Dans le cadre de l'administration électronique, à l'heure d'Internet, les échanges de fichiers ou de services entre administrations ou avec les partenaires tiennent un rôle essentiel et grandissant. Il en résulte une multiplication exponentielle du nombre de partenaires, ce qui complexifie d'autant la maintenance de ces échanges. De plus, chaque partenaire ne dispose pas toujours des mêmes outils de communication, ni n'utilise les mêmes formats d'échanges ce qui ne fait qu'augmenter l'entropie générale. L'architecture proposée a pour but de faciliter les échanges d'informations et de pouvoir, par la suite, mettre en place des systèmes de services distribués facilement accessibles et intégrables.

Elle doit aussi servir de base aux communications avec les partenaires européens voire être ouvert à d'autres partenaires extranationaux.



Cette architecture s'articule autour de points d'échanges autrement dénommés hub qui, à l'image des hubs dans le domaine du transport aérien, fédèrent chaque entité cohérente, comme un ministère, un service de collectivité, un partenaire social…

De ce fait, le nombre de partenaires en contact direct diminue fortement et un changement sur un partenaire est automatiquement répercuté sur l'ensemble car, il n'est nécessaire de reconfigurer qu'une seule liaison, celle vers le hub.

Ainsi, par exemple, une mairie pourra communiquer aisément avec un ministère ou encore avec un partenaire social sans avoir à connaître l'ensemble des paramètres de communication sous-jacents. Il lui suffira d'envoyer un message formaté à son hub de collectivité locale avec le destinataire final. Ce hub poussera le message vers le hub fédérateur qui le transférera au hub du ministère. Le hub du ministère acheminera alors le message vers le service concerné. Il est à noter que la mairie peut par exemple utiliser FAST alors que le ministère utilise FTP. Les hubs assurent donc la conversion au passage des protocoles de transport.

Ces conversions de protocoles sont effectuées en périphérie de chaque hub grâce à des connecteurs – ou passerelles – qui assurent, d'une part, la connectivité entre deux hubs (e.g., SOAP/HTTP, FTP, etc.) et, d'autre part, la conversion des enveloppes permettant l'acheminement des messages. L'enveloppe du Hub Fédérateur est le protocole d'échanges qui fait l'objet du présent document. Les autres hubs sont libres d'utiliser leur propre protocole d'échanges. Toutefois, pour accroître l'interopérabilité, il serait préférable de généraliser l'utilisation d'un unique protocole ou, a minima, s'assurer de la « compatibilité » des différents protocoles d'échanges. En effet, le passage d'un protocole à un autre peut engendrer une perte d'information car les champs prévus en entête des messages peuvent être plus ou moins riche d'un protocole à l'autre (certains protocoles gèrent l'horodatage des messages d'autres non, certains protocoles gèrent la segmentation des messages d'autres non, certains protocoles gèrent des clés de corrélation d'autres non, etc.).

A noter enfin que cette architecture n'exclut pas la communication directe entre entités dans les cas spécifiques ou cela s'averre nécessaire comme par exemple : un échange synchrone entre partenaires, de fort besoins en communication, un cadre sécuritaire particulier… Dans de tels cas, ni les hubs, ni les protocoles d'échanges, dont il est question dans ce document, ne seront utilisés.

2.3 ETUDE COMPARATIVE DES PROTOCOLES FAST, ELINK ET EBMS2

En préalable à l'établissement de cet appel à commentaires, l'ADAE a lancé une étude comparative [COMP] entre différents protocoles d'échanges qui pourraient répondre à tout ou partie des besoins concernant le futur protocole d'échange du programme ADELE. Les protocoles à ce jour étudiés par l'ADAE sont :

- IDA eLink : ce projet du programme IDA (Interchange of Data between Administrations) vise à définir les caractéristiques d'un nouvel intergiciel de communication qui sera utilisé par l'UE et par les administrations des États membres [ELINK]. Il est fondé sur le projet Government eLink, élaboré en Suède, et la German Online Services Computer Interface, élaboré en Allemagne. Le protocole eLink est basé en grande partie sur le protocole allemand OSCI [OSCI].
- CDC FAST : ce protocole a été défini par la CDC pour les besoins de sa plate-forme d'échanges transactionnelles sécurisées qui propose, entre autres, des fonctions avancées de gestion de preuves électroniques [FAST].
- ebMS2 : ce protocole utilisé dans le cadre d'échanges ebXML (electronic business XML) a été élaboré sous l'égide du consortium OASIS (Organization for the Advancement of Structured Information Standards). La version 3, en gestation, devrait se rapprocher des standards émergeants liés aux services Web [EBMS2].

Le tableau suivant synthétise les résultats de l'étude comparative :

	FAST	**eLink**	**ebMS2**
3.1 Format des données	Utilisation du format XML pour structurer le message. Gestion de rôles. Distinction entre deux parties : la partie transport propre au middleware et la partie métier propre aux applications qui constituent une double enveloppe.		
	Un seul type de contenu à la fois. Le document est encodé dans le message XML.	Utilisation des attachements (Soap With Attachments) : les documents binaires sont placés en tant que pièces jointes (MIME).	
3.2.1 Protocole de transport	HTTP(S)	HTTP(S) ou MOM	
3.2.2 Routage	Routage explicite (point à point) ou routage sur le contenu.	Routage explicite (point à point), mode *publish and subscribe*, mode requête/réponse. Notion de nœuds intermédiaires (routage multiple).	
			Possibilité de masquer les intermédiaires.
3.2.3 Messages de services	Gestion des accusés, envoi d'erreurs, récupération d'informations sur le transport.	Gestion des accusés, envoi de messages d'erreurs via la norme SOAP Fault.	Gestion des accusés, envoi de messages d'erreurs via la norme SOAP Fault, récupération d'informations sur les échanges, *ping* de service.
3.2.4 Corrélation	Possibilité de lier des messages entre eux via un identifiant de message.	Les messages sont identifiés par un ensemble de champs techniques qui renseignent l'identifiant d'application et l'identifiant de connexion. Il est également possible de découper les messages de grande taille.	Utilisation d'un champ identifiant de message et identifiant de corrélation. Possibilité d'ordonnancer les messages.
3.2.5Synchronisme	Gestion des modes d'échanges synchrones et asynchrones.		
3.3.1 Chiffrement	Support de HTTPS, possibilité de chiffrer les messages via XML Encryption.		
3.3.2 Signature	Utilisation de XAdES : extension de XML Signature avec archivage sur le long terme.	Utilisation de XML Signature.	
3.4 Modularité BPM	FAST s'appuie sur XML/HTTP et propose une extension BPM.	L'utilisation des standards SOAP et UDDI permet d'intégrer eLink et une solution BPM.	ebXML peut être orchestré par des solutions compatibles ebXML. Il existe également une norme ebBPSS concernant les processus ebXML.

Remarque : D'autres protocoles d'échanges existent mais n'ont pas fait l'objet d'une étude. Parmi les protocoles plus anciens et spécialisés dans d'autres secteurs, on peut citer les protocoles TEDECO [TEDECO] ou X400/P-EDI [SANTE]. Parmi les protocoles en construction, on peut citer les protocoles émanant des extensions orientées messages des services Web. Ces derniers sont issus d'organismes tel que le W3C, OASIS ou encore WS-I et ont généralement fait l'objet de contributions de la part des éditeurs du marché.

3 ENJEUX ET BESOINS

Les questions suivantes doivent permettre de préciser et de prioriser les besoins fonctionnels qui doivent être couvert par le futur protocole d'échange de l'administration électronique.

Il est souhaitable que les réponses apportées soient illustrées d'exemples concrets et représentatifs des besoins à court terme.

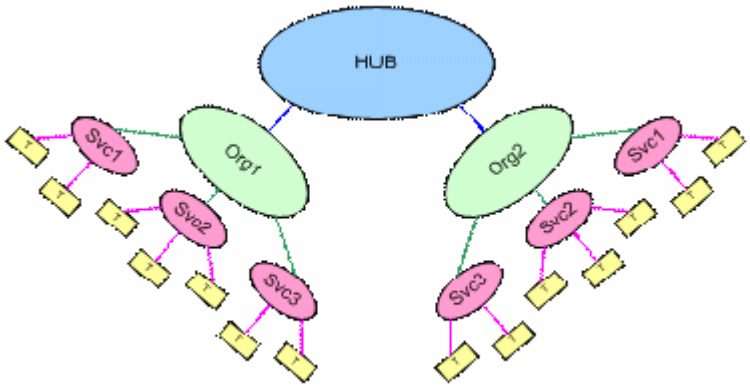
Réf.	Question
Données transportées	
1.1	Avez-vous un besoin fort concernant l'acheminement de pièces jointes binaires ? De quels types (PDF, RTF, GIF, JPEG, etc.) ?
1.2	Quelle est la taille de vos plus volumineux messages (pièces jointes comprises) ?
1.3	Pensez-vous qu'il puisse être utile de vérifier la structure des données transportées ?
1.4	Pensez-vous qu'il soit utile de pouvoir vérifier les signatures électroniques apposées sur les données transportées ?
1.5	La possibilité de détecter les doublons vous semble-t-elle utile ? Si oui, comment les identifier, suivant quels critères ?
Transport et connectivité	
2.1	Vous semble-t-il nécessaire de pouvoir diffuser un message vers plusieurs destinataire ? Si oui, qui doit avoir la liste des destinataires : l'émetteur ou le hub d'échange ?
2.2	Disposez-vous d'un hub d'échanges ou d'une plate-forme B2B ? Utilise-t-il un format d'échanges particulier ?
2.3	Quels sont les protocoles de transport et de connectivité que vous utilisez le plus (FTP, NFS, HTTP, SMTP, JMS, interface propriétaire, etc.) ?
2.4	Quels sont les technologies que vous utilisez le plus (J2EE, .NET, Web Services, interface propriétaire, etc.) ?
Sécurité	
3.1	Authentification de l'application : faut-il s'assurer que le flux d'information provient bien d'une application donnée ?
3.2	Authentification de l'utilisateur : faut-il s'assurer de l'identité de l'utilisateur finale émetteur du flux d'information ?
3.3	Le protocole d'échange véhiculera les informations nécessaires à la signature des messages, mais pensez-vous qu'il soit utile de gérer les signatures multiples sur même document, sur des parties de documents, sur certaines pièces jointes ?
3.4	Le protocole doit-il prévoir des informations de non-répudiation, c.-à-d. de traces électroniques à valeur probante (signature de l'émetteur, du récepteur, d'un tiers de confiance et horodatage des messages) ?
3.5	Le protocole doit-il prévoir les informations nécessaires à l'archivage sécurisé des données échangées ?
Qualité de service	
4.1	Quelle est la fréquence attendue ?

4.2	La gestion des priorités pour acheminer des messages urgent vous semble-t-elle nécessaire ?
4.3	Les émetteurs doivent-ils pouvoir demander qu'un accusé de livraison leur soient envoyé lorsqu'un message est remis au destinataire final ?
4.4	Les émetteurs doivent-ils pouvoir positionner des alertes spécifiques telles qu'une alerte en cas de non réception d'un accusé de réception dans des délais paramétrables ?
4.5	Les émetteurs doivent-ils être notifiés dans le cas ou une anomalie surviendrait au cours d'un échange (signature invalide, destinataire injoignable, etc.) ?
Orchestration de service	
5.1	Les messages doivent-ils pouvoir être pilotés par un processus ou un workflow ? En d'autres termes, un message doit-il pouvoir stocker des informations sur son état d'avancement, générer des messages informatifs de suivi ou encore être corrélé avec d'autres messages ?

Notion d'enveloppe multiple

Le périmètre de mise en œuvre des échanges du programme ADELE concerne principalement le niveau « organisation » : par exemple, échanges entre ministères, administration européennes, une mairie, un partenaire privé…

Le transport des documents s'appuie sur un mécanisme d'enveloppe multiple : une enveloppe de transport contient les informations techniques nécessaires pour la transmission, une enveloppe métier contient le document lui-même.



Dans le cadre des échanges du programme ADELE, il peut être souhaitable de mettre en œuvre, au sein de chaque « hub local», un routage fonctionnel : le destinataire fonctionnel est déterminé selon des informations contenues dans l'entete de l'enveloppe fonctionnelle. Plutôt que d'extraire les données du document qui peut être chiffré, le routage fonctionnel utilise un entête dédié. Cet entête fonctionnel contient un ensemble de valeurs clés nécessaires et suffisantes pour le routage.

Ce routage fonctionnel est utile par exemple pour les demandes administratives sur les statuts des sociétés répondant à un marché public dans un contexte européen : le routage est effectué à partir du champ Pays contenu dans l'entête fonctionnel ; il consiste à positionner la valeur des champs Organisation, Service et Traitement pour que le message soit transmit au hub du pays concerné, puis au service traitant (via d'autres hub locaux au dit pays si nécessaire) ou, s'il s'agit de la France, de l'organisme traitant. Dans le cas de la France, ce routage ne s'appuie pas sur un annuaire, mais plutôt sur la connaissance précise du destinataire.

Réf.	**Question**
Routage	
6.1	Quelles informations doivent être véhiculées par le routage pour permettre au document d'être routé vers le bon service et le bon traitement ? Quelles sont les informations qui vous sont nécessaires pour déterminer le destinataire de vos messages ?

4 DESCRIPTION DU PROTOCOLE

Le protocole d'échanges eLink est donc le protocole retenu comme étant le plus représentatif des besoins pour les échanges du programme ADELE. Certaines fonctionnalités de eLink ne sont pas utiles dans le cadre de ces échanges, par contre d'autres semblent nécessaires à ajouter.

Ce document va présenter pour chaque besoin ces fonctionnalités.

4.1 FORMAT DES DONNEES

4.1.1 Transport des données binaires

XML ne permet pas de transporter nativement des données binaires. Deux possibilités sont envisageables pour pallier cette contrainte : soit on encode les données binaires avant de les inclure dans le document XML, soit on les transporte de façon dissociée sous leur forme originale.

4.1.1.1 Le protocole eLink

Le format de eLink s'appuie sur le format SOAP 1.2 With Attachments (SWA) qui consiste à placer les données binaires en pièce jointes (utilisation de MIME, Multipurpose Internal Mail Extensions). Ce format est préférable pour le transport des données binaires (fichiers PDF, JPEG, etc.) mais présente des contraintes certaines :

- utilisation de plusieurs formats : XML n'est pas le format unique, l'utilisation exclusive d'outils de sécurité XML (Signature et Encryption) n'est pas possible,
- contrôle du contenu : les pièces jointes peuvent être altérées sans que le transport de s'en rende compte, les documents joints devront être signés et chiffrés individuellement.

SWA est donc un mécanisme lourd mais non limitant car chaque pièce jointe peut être chiffrée et signée séparément.

4.1.1.2 Le protocole FAST

Le protocole FAST propose une alternative qui consiste à placer le fichier binaire encodé (base 64) dans le message XML. Cette solution n'utilise donc aucune pièce jointe et a pour avantage de n'avoir qu'un unique format d'échange (XML).

En contrepartie, cette approche est considérée comme coûteuse en complexité de document et en performance de traitement : la taille d'un document encodé en base 64 est augmentée de 33 %.

4.1.1.3 Le protocole ebMS2

Le protocole ebMS2 propose la même solution que le protocole eLink.

4.1.1.4 Les orientations

Les technologies MTOM (Message Transmission Optimisation Mecanism) et XOP (Xmlbinary Optimised Package) sont sensées clore le débat en combinant les différentes techniques. Ces technologies sont des recommandations W3C et poussées par les principaux éditeurs. Le principe est d'encoder les données binaires dans le document XML sous la forme de données base64 qui permet de bénéficier des normes XML de sécurité et de Signature, et de laisser à l'implémentation l'extraction de ces données et leur placement en pièce jointe. Cette technologie serait parfaitement adaptée au transport d'un protocole tel que FAST : l'augmentation de taille liée à l'encodage serait résolue.

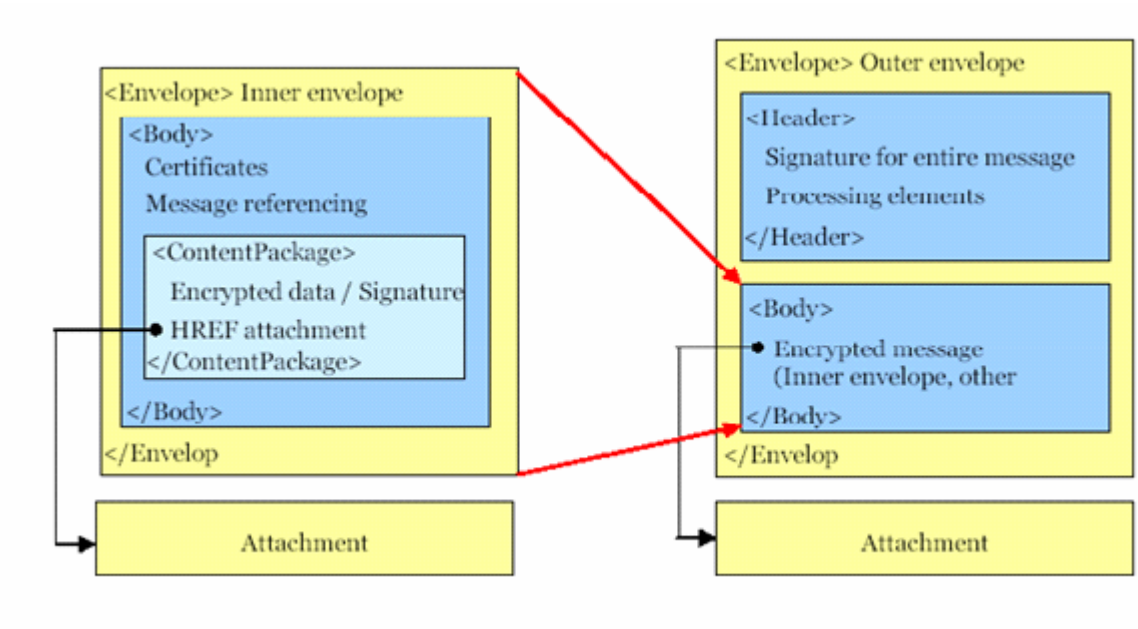
Cependant, MTOM ne semble pas une solution assez mature à ce jour mais pourrait se présenter comme un compromis idéal entre les deux approches vues précédemment. Le protocole devra donc prévoir une évolution vers la technologie MTOM. A ce titre, on peut citer le cas de FAST car MTOM est compatible avec les options prises par FAST. Les impacts d'une migration sur l'existant seraient donc extrêmement limités.

Réf.	Question
Protocole	
7.1	Que préconisez-vous : SWA ou MTOM (ou autre comme, par exemple WS-Attachments/DIME) ?
7.2	Voyez-vous d'autres limitations à l'utilisation de SWA comme mécanisme d'échange des pièces jointes ?
7.3	La technologie MTOM vous parait-elle une solution envisageable ? A quelle échéance ?

4.1.2 Notion d'enveloppe multiple

4.1.2.1 Le protocole eLink

Le mécanisme d'enveloppes eLink utilise Soap With Attachments.

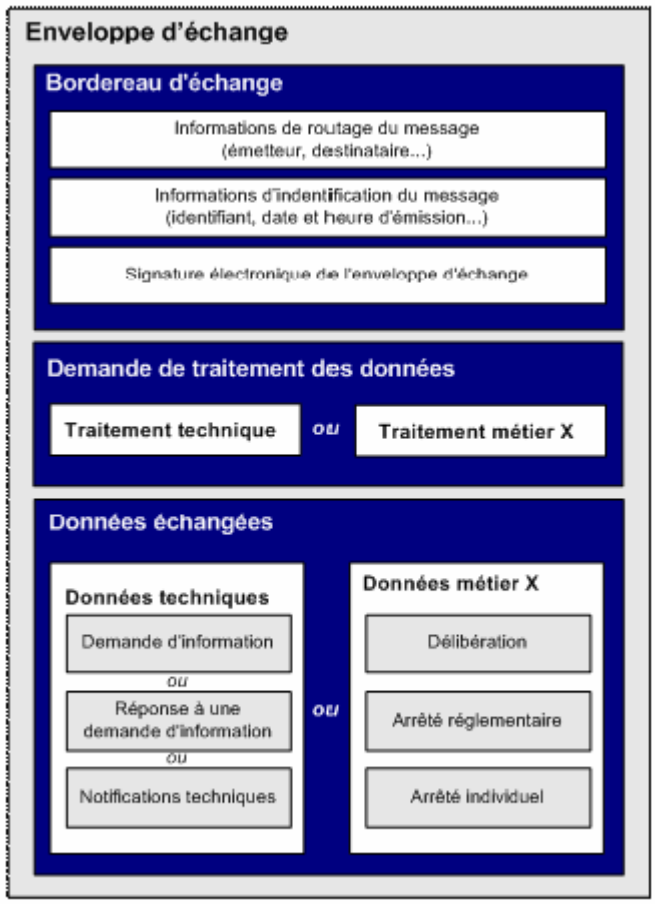


Le message à transférer est placé en pièce-jointe d'un message dit de « transport » qui permet sa diffusion entre éléments eLink. Le message n'est donc pas directement manipulé par eLink. La norme prévoit d'ailleurs qu'il soit chiffré, ce qui a pour conséquence d'interdire tout traitement sur le contenu. A noter enfin que seule la partie XML du message est placé dans le corps de l'enveloppe de transport. Les pièces en attachement ne sont pas modifiées.

4.1.2.2 Le protocole FAST

Une enveloppe d'échange est constituée :

- D'un bordereau d'échange (routage, identification, signature),
- D'une demande de traitement des données échangées (i.e., un type de message),
- Des données échangées elles-mêmes (exemple du contrôle de légalité).



L'intégralité du message, pièces-jointes incluses, est compris dans un unique document XML. La manipulation des messages s'en trouve simplifiée (transformation, vérification de structure, signature et chiffrement par partie, etc.).

4.1.2.3 Le protocole ebMS2

Un message ebXML, défini par la norme ebMS2, est constitué d'une enveloppe de message composée de plusieurs attachements MIME. De même que eLink, cette enveloppe est structurée en accord avec la spécification de SWA (SOAP With Attachement).

Réf.	Question
Protocole	
7.4	Quel principe de double enveloppe préconisez vous parmi ceux proposés ?
7.5	Quelles informations doivent être véhiculées par le routage pour permettre au document d'être routé vers le bon service et le bon traitement ?

4.1.3 Standards Web Services

Les organisations WS-I, W3C et OASIS travaillent sur les définitions des normes et standards autour de XML et des services Web. Un ensemble de normes émergentes devraient en grande partie s'imposer comme standard :

- *UDDI (version 3)* : annuaire de services, les services (mais aussi les documents XML, schémas XSD et transformations XSL) sont classés suivant une taxonomie et peuvent être dynamiquement localisés par une interface d'accès normalisée qui est elle-même un service Web.
- *WS-Security* : ensemble de règles de sécurité à l'usage des services Web comprenant XML-Signature, XML-Encryption, Token Authentication (basé sur identifiant-mot de passe, certificats, SAML), etc. Cette norme est déjà implémentée sur certaines solutions, eLink en supporte une partie (XML-Signature et WML-Encryption).
- *WS-Addressing* : standard de routage des requêtes asynchrones basé sur des entêtes particuliers (<To>, <From>, <MessageID>, <ReplyTo>, <FaultTo>). Cette norme est déjà implémentée sur certaines solutions du marché.
- *WS-Policy* : méta-données permettant de décrire les contraintes et les préférences liés à l'utilisation d'un service Web (par exemple, la nature de l'encodage, le langage naturel, la version, le ou les mécanismes d'authentification, etc.).
- *WS-Reliability / WS-ReliableMessaging* : garantie la livraison des messages.
- *WS-Eventing / WS-Notification* : mise en place de mécanismes publication/abonnement, envoi par *topic*. Ces normes s'appuient sur WS-Addressing.
- WS-*Coordination, WS-Transaction, WS-Context, WS-ResourceFramework, WS-Federation…* : de nombreuses études sont en cours pour répondre à d'autres problématiques telles que la gestion des transactions, la fédération d'identité, etc.

4.1.3.1 Les orientations

Les standards à prendre en compte en priorité sont WS-Security, WS-Addressing, WS-Reliability / WS-ReliableMessaging et UDDI v3.

Réf.	Question
Protocole	
7.6	Quel est selon vous le calendrier réaliste que l'on peut attendre pour une implémentation opérationnelle de ces standards ?
7.7	Faut-il aller vers ces standards (aujourd'hui pas mature et demain…) ?

4.2 ARCHITECTURE D'ECHANGE DES MESSAGES

4.2.1 Protocole de transport

Les trois sont indépendant des protocoles de transport sous-jacent.

4.2.1.1 Le protocole eLink

Le protocole d'échange doit être indépendant du transport. L'utilisation du protocole eLink qui s'appuie sur SOAP 1.2 permet notamment de s'appuyer sur les transports suivants :

- **HTTP(S)** : ce transport est le plus utilisé pour les services Web. Il possède l'avantage indéniable de passer la plupart des *firewall* (le protocole HTTP est souvent autorisé et ne nécessite l'ouverture que d'un seul port) et l'ajout de SSL permet d'en sécuriser le contenu au niveau transport. L'utilisation d'accélérateurs SSL, de compresseurs HTTP est possible. En contrepartie, HTTP ne garanti pas la livraison de message, est synchrone et n'est pas transactionnel (XA) : l'émetteur est bloqué tant que le message n'est pas traité.
- **MOM (solution propriétaire ou JMS)** : l'utilisation d'un *middleware* de message asynchrone (MOM) pour les services Web est une tendance émergente et permet d'utiliser ces derniers

dans un mode plus robuste. L'utilisation du mode asynchrone permet de découpler émetteur et récepteur. Le *middleware* de messages assure de plus une qualité de service avec la persistance des messages qui transitent et (selon le produit retenu) une durée de vie, le support du mode transactionnel (XA), la reprise sur incident, etc. Le principal inconvénient est l'absence de standard (à l'exception de JMS qui est standard au sein du monde J2EE mais n'a pas d'équivalent par ailleurs).

A noter qu'il n'y a, a priori, pas de limitations quant à l'utilisation d'autres protocoles tels que SMTP ou FTP.

4.2.1.2 Le protocole FAST

Aujourd'hui, il n'y a que des implémentations HTTPS mais le protocole permet de fonctionner sur d'autres couches de transport, par exemple FAST utilise JMS en interne.

4.2.1.3 Le protocole ebMS2

Ici aussi, il n'y a pas, a priori, de limitations. Cependant la norme [EBMS2] ne décrit que deux « binding » : HTTP (version 1.1 minimum) et SMTP (support de MIME requis).

4.2.1.4 Les orientations

Le support de plusieurs transports est important car il laisse la possibilité à terme de changer de mode d'échange. Les protocoles HTTP et MOM permettent de s'appuyer sur un grand nombre de solutions du marché.

Réf.	Question
Protocole	
7.8	Le support d'autres transports (SMTP, FTP, SFTP, SSH, etc.) vous parait-il important dans le cadre des échanges entre organisations concernées par le programme ADELE ?

4.2.2 Routage des messages

Le tableau suivant présente différentes topologies d'échanges envisageables. Un ordre de priorité indique leur importance pour les échanges de l'administration électronique.

Topologie	Description	Priorité
Point à point 1 → 1	L'émetteur envoi un message vers un destinataire identifié. Aucune réponse n'est attendue.	1
Point à point 1 → N	L'émetteur envoi un message vers un groupe de destinataires (utilisation d'un alias ou d'une liste de destinataires). Aucune réponse n'est attendue.	2
Requête/Réponse	L'émetteur envoi un message vers un destinataire. Une réponse est attendue (synchrone).	3
Conversationnel	L'émetteur et le(s) destinataire(s) échangent des messages. Une session est maintenue.	3

Publication/Abonnement	L'émetteur publie un message dans un *topic*. Zéro ou plusieurs destinataires peuvent être abonnés à ce *topic* et recevoir le message. L'émetteur ne connaît pas les destinataires. Aucune réponse n'est attendue.	4

Les protocoles eLink, FAST et ebMS2 supportent le mode « point à point 1 → 1 ». eLink et ebMS2 supportent de plus le mode « publication/abonnement ».

4.2.2.1 Les orientations

Le mode point à point est le mode prioritaire pour les échanges des projets ADELE.

4.2.3 Gestion de la corrélation

4.2.3.1 Le protocole eLink

Le mécanisme de gestion de corrélation du protocole eLink permet de retourner la réponse à l'émetteur en se basant sur un système de persistance des requêtes.

Le protocole eLink utilise plusieurs indicateurs de corrélation :

- **MessageId** : identifiant unique du message,
- **MessageType** : type de message (requête, réponse, publication/abonnement…),
- **Part** : numéro de message dans le cas d'une séquence de messages,
- **TotalParts** : nombre total de parties de la séquence,
- **ConnectionId** : identifiant de connection,
- **ApplicationId** : identifiant d'application.

4.2.3.2 Le protocole FAST

Le protocole FAST utilise 2 indicateurs de corrélation :

- **MessageId** : identifiant unique du message courant. Ce champ est obligatoire, qu'il s'agisse d'un message seul ou d'un message conversationnel, c'est-à-dire lié à d'autres messages. Ce champ est généré par FAST pour assurer l'unicité.
- **RefToMessageId** : identifiant unique du message auquel le message fait suite. Si le message ne fait suite à aucun message, cet élément ne doit pas être présent.

A noter qu'une évolution en cours doit permettre segmenter un message suivant le même principe qu'eLink (champs Part et TotalParts).

4.2.3.3 Le protocole ebMS2

Le protocole ebMS2 propose dans son en-tête un identifiant unique de message (**MessageId**), un identifiant unique de conversation (**ConversationId** : fait référence à un ensemble d'échanges donné entre deux tiers) et une référence de message (**RefToMessageId** : utilisé entre autres pour faire référence à un message donné en cas d'erreur).

4.2.3.4 Les orientations

Le mécanisme de corrélation est indispensable dans le cadre des échanges asynchrones.

Réf.	Question
Protocole	
7.9	Qui, selon vous, doit générer les identifiants uniques : la plate-forme d'échanges ou l'application émettrice ?

4.2.4 Gestion des messages de service

4.2.4.1 Le protocole eLink

Le protocole eLink spécifie un certain nombre de types de messages permettant d'obtenir des informations de service. Pour plus de détails sur les formats de message eLink, se référer au document de spécifications eLink [ELINK].

- *Connect* : message de connexion à un nœud eLink,
- *Disconnect* : message de déconnexion à un nœud eLink,
- *SendMessageId* : récupération d'un identifiant de message,
- *GetMessage* : envoi d'un message,
- *SendMessageList* : récupération d'une liste de message,
- *SendMessage* : récupération d'un message,
- *Subscribe* : abonnement à un service,
- *Unsubscribe* : désabonnement à un service,
- *Publish* : publication vers un service,
- *SubscriptionStatus* : récupération de l'état d'un abonnement,
- *Acknowledgement* : envoi d'accusé de réception,
- *Agreement*[1] : déclaration des informations de communication entre applications.

Concernant la gestion des erreurs, eLink s'appuie sur le protocole SOAP qui définit une balise d'erreur appelée SOAP Fault. Il propose la liste d'erreurs suivante :

- le message reçu est invalide,
- le fournisseur ne peut être détecté,
- signature invalide,
- certificat de chiffrement invalide,
- erreur interne,
- message expiré (*timeout*).

4.2.4.2 Le protocole FAST

Le protocole FAST spécifie les types de messages suivants :

- Transmission d'un message métier.
- Récupération d'un message en attente :
  - Un message métier,
  - Un message technique contenant des notifications asynchrones.

[1] Agreement n'est pas un type de message mais est un élément important concernant le service eLink.

- Demande d'informations techniques permettant aux applications de récupérer des informations pouvant être utiles à leur bon fonctionnement. Ces requêtes sont définies par application :
    - Liste des politiques de signature : il s'agit de la liste des politiques de signature pouvant être utilisées lors des opérations de signature dans le cadre de l'application (e.g., politique de signature pour un agent, pour un officier d'Etat Civil, pour une application métier, etc.). Pour chacune des politiques sont en particulier fournies son OID, son empreinte, ainsi qu'une URL d'où elle peut être téléchargée.
    - Liste des autorités de certification référencées : chaque autorité de certification peut être référencée pour un ou plusieurs usages parmi les suivants : signature, chiffrement de données et/ou échange.
    - Liste des certificats serveur d'authentification : il s'agit de la liste des certificats serveurs utilisés par la plateforme FAST pour s'authentifier auprès des entités souhaitant communiquer avec elle. En temps normal cette liste ne contient qu'un unique certificat. Il est cependant utile de pouvoir en préciser plusieurs afin de gérer les périodes de recouvrement.
    - Liste des acteurs connectés aux services : Tout acteur accessible au travers de la plateforme FAST est référencé dans cette liste. La requête permet ainsi de récupérer le type de l'acteur (collectivité, organisme social…), son numéro de SIREN, les applications qu'il prend en charge ainsi que son certificat de chiffrement (permettant d'assurer la confidentialité de bout en bout des échanges).
    - Liste des habilitations des utilisateurs chez l'acteur : la liste des utilisateurs inscrits chez un acteur peut être transmise à ce dernier (et à ce dernier uniquement). Chaque utilisateur est caractérisé par ses nom et prénom, son courriel, son rôle métier et son groupe d'appartenance, ainsi que par la liste des ressources applicatives qu'il est autorisé à manipuler.
    - Liste des anomalies : il s'agit ici de fournir aux applications une table de correspondances entre les codes et libellés d'erreurs techniques et des libellés plus haut niveau pouvant être directement utilisez pour informer un utilisateur.
    - Liste des enveloppes mises à disposition : cette requête permet à un acteur de demander la liste des enveloppes qui sont mises à sa disposition de façon asynchrone. A partir de cette liste il est alors possible de récupérer chaque enveloppe pour la traiter.
    - Etat d'avancement dans le traitement d'une enveloppe transmise : il est possible de connaître à tout moment l'état d'avancement du traitement (l'enveloppe a-t-elle bien été reçue, traitée, transmise…).

Concernant la gestion des erreurs, FAST utilise, tout comme ebMS2, un champ ErrorList en entête de message et une API permet aux applications de récupérer cette liste.

La plateforme FAST assure une traçabilité maximum des informations échangées. Celles-ci passent par trois états principaux :

- Information reçue par la plateforme FAST : l'information a bien été reçue par la plateforme FAST, mais n'a pas encore été transmise à son destinataire. Elle est en cours d'horodatage, de validation…
- Information prête à être transmise par la plateforme FAST, mais non arrivée à destination : l'information a été mise à disposition du destinataire par la plateforme FAST, mais la plateforme reste en attente de la notification de réception.
- Information arrivée à destination : le document a bien été transmise à son destinataire.

Les émetteurs ont la possibilité de contrôler en permanence l'état de leur télétransmission et pourront s'ils le souhaitent être notifiés par la plateforme FAST à chaque évolution de l'état des informations transmises.

Les émetteurs peuvent de plus être notifiés dans le cas ou une anomalie surviendrait au cours d'un échange (signature invalide, destinataire injoignable, etc.).

Enfin, l'utilisateur pourra positionner lui-même des alertes relatives à son domaine métier : alerte en cas de non réception d'un accusé de réception dans des délais paramétrables, alerte en cas de non réception d'une réponse dans des délais paramétrables, etc.

4.2.4.3 Le protocole ebMS2

Le protocole ebMS2 prévoit les services et messages de service suivants :

- Accusé de réception,
- Détection de doublons,
- Mécanisme de *retry*,
- Durée de vie des messages,
- Interrogation de l'état d'un message,
- Test de connectivité (*ping* et *pong*).

Concernant la gestion des erreurs, un champ ErrorList en entête de message permet de stocker une liste d'erreurs (id, codeContext, errorCode, severity, location, description). Les erreurs suivantes sont supportées en standard :

- ValueNotRecognized,
- NotSupported,
- Inconsistent,
- OtherXML,
- DeliveryFailure
- TimeToLiveExpired
- SecurityFailure
- MimeProblem
- Unknown

4.2.4.4 Les orientations

La définition du protocole aura pour base les messages de services proposés par eLink. Elle devra être suffisamment simple et extensible.

Réf.	Question
Protocole	
7.10	La liste des erreurs vous paraît-elle suffisante dans le cadre des échanges des projets du programme ADELE ?
7.11	Quelles possibilités de suivi de message (cf. FAST : notifications d'avancement, positionnement d'alertes) vous semblent-elles pertinentes ?

4.2.5 Asynchronisme des échanges

4.2.5.1 Le protocole eLink

Le protocole eLink précise que l'utilisation d'un système de persistance est nécessaire pour faire office de tampon de gestion de messages asynchrones.

4.2.5.2 Le protocole FAST

Idem eLink.

4.2.5.3 Le protocole ebMS2

Idem eLink.

4.2.5.4 Les orientations

Les échanges électroniques concernés par ce protocole se font entre des systèmes d'informations variés (ministères, administration européenne, organismes privés…). Il est donc très important que le protocole ne soit pas intrusif et que le couplage avec les acteurs soit lâche (i.e., aucune dépendance entre SI distincts).

Pour cela, le mode asynchrone doit être privilégié. En effet, dans le cas d'un envoi de document sans attente de réponse, il est inutile de bloquer l'application émettrice.

Le mode asynchrone est fortement recommandé dans tous les cas d'échange de données, si une réponse doit être retourné, il faut que celle-ci soit transmise sous la forme d'un accusé de réception (avec un mécanisme de corrélation).

4.3 GESTION DE LA SECURITE

4.3.1 Chiffrement des données

Les messages transmis sont constitués de plusieurs éléments : les éléments XML qui peuvent être des données techniques (enveloppe de transport) ou métier et les éléments dits « binaires » qui peuvent être des fichiers PDF, des fichiers plats ou des images.

Les protocoles évalués proposent un chiffrement des données XML assuré par le standard XML-Encryption.

Le protocole eLink ne traite pas clairement le cas du chiffrement des données binaires jointes au message. La possibilité de chiffrer ces éléments est importante.

De même la possibilité de chiffrer par partie semble intéressante. Le protocole FAST permet ce type de chiffrement partiel.

Réf.	Question
Protocole	
7.12	Quel mécanisme de chiffrement des documents joints serait-il préférable d'utiliser ?

4.3.2 Signature des messages

4.3.2.1 Le protocole eLink

Le protocole eLink supporte le standard XML-Signature.

Le protocole eLink ne traite pas clairement le cas de la signature des données binaires jointes au message. La possibilité de signer ces éléments est importante.

4.3.2.2 Le protocole FAST

FAST supporte l'utilisation de XAdES. Les spécifications XAdES (ou XML Advanced Electronic Signature) prolongent celles de XML-Signature dans le domaine de la non-répudiation en définissant des formats pour les signatures électroniques qui doivent restées valides pendant de grandes périodes et être conformes à la « Directive 1999/93/EC du parlement Européen et du conseil du 13 décembre 1999 sur le cadre communautaire des signatures électroniques ».

Les éléments ajoutés par XAdES sont par exemple :

- la date et l'heure déclarées de signature ;
- le certificat de signature ou une référence non ambiguë au certificat de signature ;
- la politique de signature sous forme d'une référence non ambiguë à la politique (empreinte et/ou OID), d'une URL de téléchargement du document complet et d'une description littérale ;
- le lieu de la signature ;
- le rôle du signataire (e.g., Agent, Officier d'Etat-Civil, Signataire…).

Le format XAdES permet de plus d'insérer un jeton d'horodatage portant sur la signature, ainsi que les informations nécessaires à un archivage long terme des signatures.

4.3.2.3 Le protocole ebMS2

Idem eLink.

Réf.	**Question**
Protocole	
7.13	Quel mécanisme de signature des documents joints serait-il préférable d'utiliser ?
7.14	La technologie de signature XAdES est-elle, selon vous, pérenne et surtout utilisable sans contrainte ?

5 HUB D'ECHANGES

Ce chapitre à pour objet de présenter brièvement les fonctionnalités de la plate-forme cible qui mettrait en œuvre le protocole d'échanges.

Les mots DOIT, DEVRAIT, NE NECESSITE PAS doivent être interprétés comme suit :

- DOIT : la fonctionnalité est indispensable,
- DEVRAIT : la fonctionnalité est intéressante et serait un plus,
- NE NECESSITE PAS : la fonctionnalité ne semble pas utile.

5.1 CONNECTIVITE

5.1.1 API et kit de développement

Le hub d'échange DOIT fournir une API afin de permettre le développement de connecteurs spécifiques. Cette API DOIT permettre l'intégration d'applications réalisées dans différents langages (C/C++, Java, COM, PHP, .NET, etc.)

La plateforme DEVRAIT fournir un kit de développement pour accélérer la réalisation des connecteurs.

5.1.2 Connecteurs techniques

Le hub d'échanges DEVRAIT disposer d'un ensemble de connecteurs techniques permettant l'intégration des différentes organisations. Parmi les connecteurs techniques, on peut citer HTTP(S), Web Services (SOAP sur HTTP ou MOM), les principaux MOM du marché, .NET, Java, JMS, base de donnée (JDBC, ODBC), fichiers (NFS, FAT32, etc.)…

Réf.	Question
Hub d'échanges	
8.1	Quels sont selon vous les connecteurs techniques indispensables ?

5.1.3 Interopérabilité avec d'autres protocoles d'échanges

L'interopérabilité avec d'autres protocoles sera mise en œuvre en utilisant une passerelle fournie avec la plateforme ou en développant une passerelle spécifique à partir du kit de développement et des API.

La passerelle DOIT assurer la compatibilité entre le deux formats en s'appuyant sur des fonctionnalités de transformation (par exemple, XQuery ou XSLT), de validation (par schéma XSD) et éventuellement de transcodification.

Réf.	Question
Hub d'échanges	
8.2	Quels sont selon vous les protocoles d'échanges avec lesquels il sera nécessaire d'être interopérable ?

5.1.4 Connecteurs progiciels

Aucun besoin de connecteur progiciel n'est identifié.

5.2 TRANSPORT

Le hub d'échanges DEVRAIT s'appuyer en interne sur un *middleware* de messagerie (MOM) qui garantisse la persistance des messages, la reprise sur incident, la garantie de livraison, le respect du mode transactionnel...

Réf.	Question
Hub d'échanges	
8.3	Cette exigence vous semble-t-elle pertinente ? Quelles alternatives voyez-vous ?
8.4	Quel système de persistance (SGBD/R, MOM, système de fichier) préconisez-vous pour les échanges asynchrones ?

5.3 ROUTAGE ET TRANSFORMATION

5.3.1 Principe

Le hub d'échanges DOIT disposer d'un système de routage simple supportant les modes retenus par le protocole d'échanges (point à point synchrone et asynchrone, requête/réponse, publication/abonnement).

Le hub d'échanges NE NECESSITE PAS de fonctionnalités le routage sur le contenu. Ce mode n'est pas requis dans un premier temps mais un routage par en-têtes fonctionnels DEVRAIT être envisagé.

Les transformations, transcodifications, gestion des références croisées et enrichissement des données ne s'appliqueront que sur l'enveloppe de transport mais jamais sur la partie métier du message (en tout cas, dans un premier temps). En effet, la partie métier ne doit pas être altérée par le hub d'échanges.

5.3.2 Base de routage

La plateforme d'échanges DOIT fournir une base de routage afin de contenir les informations de connexion de chaque destinataire.

La plateforme DEVRAIT permettre l'interrogation d'une base de routage externe via une interface d'accès LDAP, UDDI, SQL ou par une API Web Services.

Un mécanisme de synchronisation entre un référentiel externe et la base de routage de la plateforme DEVRAIT également être possible.

Réf.	Question
Hub d'échanges	
8.5	Comment sont propagées les mises à jour des tables de routage entre différents hubs interconnectés ? Faut-il imaginer un protocole de synchro du type de BGP (Border Gateway Protocol) ?

5.4 SUPERVISION

Une console de supervision des échanges DOIT permettre entre autres de visualiser :

- un historique des échanges (source, cible, heure…) avec un mécanisme de non répudiation,
- des statistiques sur les échanges,
- la visualisation des instances d'orchestrations en court, la possibilité d'intervenir sur les instances en erreur…

5.5 SECURITE

Le hub d'échanges DOIT permettre la mise en œuvre des mécanismes de sécurité du protocole : gestion des certificats, support de SSL, chiffrement, signature… soit nativement, soit par l'import d'API du marché.

5.6 ROBUSTESSE ET PERFORMANCES

La plate forme cible DOIT être apte à supporter la charge des mouvements de documents échangés dans le cadre des projets ADELE.

Elle DOIT proposer des mécanismes de répartition de charge et de reprise sur incident.

Il DOIT être possible de sauvegarder la solution afin de restaurer le système en cas de perte grave.

6 PERSPECTIVES

6.1 CONVERGENCE VERS LES SERVICES WEB

Réf.	Question
Perspectives	
9.1	Quelle est la convergence à prévoir avec les futures technologies de services Web ?
9.2	Plutôt que de redéfinir un protocole d'échanges, le programme ADELE ne devrait-il pas plutôt utiliser « tel quel » un standard existant (WS-*, ebXML, RosettaNet, EDI XML…) ?

6.2 ORCHESTRATION BPM

La plateforme cible assure un routage des documents entre applications. L'ajout d'une couche d'orchestration de type BPM (Business Process Management) pourra apporter une logique aux échanges de documents.

Une couche d'orchestration est envisagée afin d'ajouter une approche des échanges par « scénario ». L'orchestration sera responsable d'exécuter les échanges en transmettant les données aux différentes organisations selon des critères plus avancés qu'un simple routage.

L'orchestration de processus métiers fait l'objet de travaux de standardisation, notamment pour la notation avec BPMN, pour l'exécution avec BPML et BPEL. OASIS propose également, ebBPSS qui spécifie les processus métiers ebXML.

Le marché des outils de conception et d'exécution d'orchestrations s'oriente fortement vers la norme BPEL (Business Process Execution Language) et continuent d'en proposer des évolutions.

Le protocole et la plate-forme cible devront permettre la mise en œuvre à terme d'une solution d'orchestration de processus métiers.

Pour que le protocole d'échanges puisse s'intégrer dans une architecture d'orchestration, il doit permettre l'utilisation de mécanismes de corrélation exploitables par un processus. En effet, un moteur de BPM utilise un identifiant unique pour transmettre les messages aux instances d'orchestrations en cours d'exécution. En général l'identifiant unique correspond à l'identifiant de l'instance au sein du moteur (processId).

Réf.	Question
Perspectives	
9.3	Quels éléments doivent être prévus dans le protocole pour assurer la mise en œuvre future d'un outil d'orchestration ? Et d'un outil d'orchestration basé sur BPEL ?

6.3 PROJET RITA

Le futur projet RITA (Référentiel de l'Infrastructure Technique d'ADELE) a pour objectif de réaliser un référentiel des données techniques dans l'optique de mutualiser les informations nécessaires à la gestion des échanges d'information avec les différents partenaires: les communes, les ministères ainsi que d'autres entités administratives (au sens large) pour tous les téléservices.

Ce référentiel stocke aussi des informations de nature administrative (les gestionnaires par exemple,..) et également des données d'audit (logs, statistiques, erreurs, etc.).

RITA proposera une API d'interrogation par services Web.

La plateforme cible devra proposer un moyen de réutiliser les informations localisées dans RITA (interrogation dynamique ou synchronisation).