

14 Introduction aux couches d'application

Objectifs

- ✧ Comprendre la raison d'être des couches de haut niveau.
- ✧ Comprendre les raisons qui ont conduit à la normalisation.
- ✧ Connaître les différentes catégories de services d'application : communs, domaines génériques et de gestion.

14.1 Changement de perspective

La réseautique peut être scindée en deux activités successives : acheminer des données d'un point à un autre, traiter des informations sans égard à la localisation (ni du traitement ni des informations). La ligne de démarcation entre ces deux activités sépare la couche 4 (transport) de la couche 5 (session) au sein du modèle OSI. Il en va de même dans la famille des protocoles TCP où la cinquième et dernière couche (application) est dédiée au traitement de l'information.

La première activité met uniquement en cause les équipements, la seconde est le plus souvent accomplie à l'initiative d'un humain en fonction d'un besoin à combler. La programmation de l'application permettant de combler ce besoin est grandement simplifiée si le transfert des données est transparent. Le modèle OSI, tout comme la famille TCP, ont pris ce besoin en compte mais de façon fort différente.

La démarche planifiée ayant présidée à l'élaboration du modèle OSI a inévitablement conduit les concepteurs à isoler des besoins communs aux applications. L'élaboration, et la structuration, des services communs a donc mené à insérer deux couches entre la couche de transport et la couche d'application proprement dite : les couches session et présentation.

Deux décennies plus tôt les concepteurs de FORTRAN avaient cru, eux aussi, qu'en offrant un langage de programmation de haut niveau, ils affranchiraient les utilisateurs de la nécessité de passer par des informaticiens (des programmeurs) pour concevoir et développer des logiciels.

Pour les concepteurs originaux de la famille TCP, la question du transport étant résolue, toute latitude devait être laissée aux utilisateurs pour développer leurs propres applications. La prolifération des applications incompatibles, c'est-à-dire incapables de communiquer entre elles, a cependant rapidement conduit à la formation des groupes de travail ayant pour tâche de normaliser, souvent a posteriori, les protocoles d'application. Une partie importante des travaux de l'IETF (*Internet Exxxx Task Force*) y sont consacrés.

La nature ouverte et l'accès sans barrières aux travaux et aux documents de l'IETF a contribué considérablement aux succès de la famille TCP, lui procurant à la fois vitesse de réaction et main d'œuvre nombreuse, qualifiée et passionnée.

Au-delà des besoins communs généralement induits par la prestation de services d'application, plusieurs services se sont imposés comme ayant une portée générale, ce sont les services touchant aux domaines génériques. Leur utilité effective dépendant de leur universalité (le téléphone est utile parce que tout le monde a accès au téléphone), leur normalisation est impérative.

Finalement, la prolifération des réseaux et des équipements a naturellement suscité des besoins spécifiques à la gestion et à l'optimisation des réseaux, pour lesquels des services d'application se sont avérés nécessaires.

Les prochaines sections présentent donc ces trois catégories de services d'application :

- ✧ services communs;
- ✧ services de domaines génériques;
- ✧ services de gestion.

14.2 Services communs

Les services communs sont caractérisés par le fait qu'ils sont :

- ✧ nécessaires à plusieurs autres services;
- ✧ non compris dans le transport des données.

Le plus souvent, ces services ne sont pas directement sous la gouverne de l'utilisateur mais plutôt sous-jacents à un autre service ou induits par les modalités d'utilisation d'un autre service. Pour cette raison, ils seront donc regroupés, au sein du modèle OSI, en couches distinctes de la couche application.

Les principaux services communs sont les suivants :

- ✧ **Sécurité :**
Ce qui comprend le traitement de la confidentialité, de l'authenticité, de l'imputabilité (non répudiabilité) et de l'intégrité de l'information

transmise. Ces services sont présentés de façon approfondie ultérieurement. Le chiffrement, souvent nécessaire à la mise en oeuvre de la sécurité, est un service distinct.

✧ **Reprise :**

Pour transparent qu'il est devenu, le transport des données demeure cependant sujet aux défaillances. Le traitement des défaillances est réalisé à l'aide de points de reprise qui permettent tout d'abord le rétablissement d'une situation viable (données intègres, état stable et déterminé) voire une éventuelle reprise du service depuis un point proche du point d'interruption.

✧ **Facturation :**

Tout service induit une utilisation de ressources qu'il convient de quantifier, même s'il n'en résulte pas la production d'une facture en bout du compte. La planification d'un réseau ne peut être réalisée sans informations fiables quant à l'utilisation des ressources.

✧ **Traduction :**

Deux formes de traduction automatisée sont envisageables. La traduction formelle, généralement de niveau lexical et syntaxique, permet de modifier le format et l'agencement des informations. La traduction linguistique, dont les résultats sont souvent discutés, vise à diffuser plus rapidement, sans l'intermédiaire de traducteurs humains, une information désormais diffusée sur un réseau mondial. Le cours ne traite que de la traduction formelle.

✧ **Compression :**

La bande passante a beau être de plus en plus large, les coûts de transport de plus en plus faibles, la quantité d'information à traiter semble croître encore plus rapidement. La compression tire partie de la redondance interne de la plupart des informations pour en réduire le volume. De matérielle (comme celle mise en oeuvre par les modems modernes) et générique (telle qu'implantée par des logiciels tels Stuffit, WinZip, gz), l'évolution de la compression se poursuit pour devenir de plus en plus logicielle et spécifique afin de tirer partie du format et de la nature des informations à traiter. Elle se déplace donc de plus en plus vers le contexte applicatif pour bénéficier d'une caractérisation aussi complète que possible; d'autant plus qu'une compression hâtive et une décompression tardive en maximisent les bénéfices.

✧ **Chiffrement :**

La confidentialité des informations circulant sur les réseaux est au centre des préoccupations de nombreux utilisateurs. La multiplicité des algorithmes et standards (des, des3, idea, rsa, md5, pgp, pem, etc.) répond en partie à celle des besoins et des contraintes, il est donc impératif d'encapsuler ceux-ci dans des services communs universels, sans quoi le chiffrement risque d'empêcher l'utilisation légitime de l'information par un destinataire privé du mécanisme de déchiffrement. Tout comme pour la compression, un chiffrement hâtif et un déchiffrement tardif sont souhaitables.

La signalisation des erreurs n'a pas été incluse dans la liste précédente, bien qu'elle fasse souvent partie des services communs, alors que le traitement des erreurs (reprise) a été maintenu dans la liste. En fait le mécanisme général, les conventions régissant la signalisation des erreurs, a avantage à être modélisé de façon commune à tous les services.

Les services de traduction (formelle), de compression et de chiffrement sont liés par l'ordre dans lequel ils doivent intervenir et par la nécessité de choisir judicieusement les algorithmes de chacun en fonction des autres. L'ordre est imposé par la nature même des transformations : il est impossible de traduire après avoir compressé; il est inutile de compresser un (bon) chiffrement. Le choix de l'algorithme de compression en fonction du format de l'information est capital. Finalement, il est parfois possible de combiner les trois opérations en une seule pour un gain appréciable du temps de traitement, donc du débit pouvant être soutenu.

14.3 Domaines d'application génériques

Ces domaines d'application sont ceux pour lesquels la plupart des organisations décide de mettre en place un réseau privé, ou d'accéder à un réseau public.

Ils sont regroupés ainsi :

- ✧ **Échange d'information :**
La première et plus fondamentale prestation de service des réseaux.
- ✧ **Fourniture de services :**
Mise à disposition d'un service en plusieurs points distants du point de service d'origine tout en conservant le caractère unifié du service (ce qui peut se faire avec ou sans centralisation).
- ✧ **Intégration de services :**
L'approche intégrée permet la desserte de la totalité des fonctions.

14.3.1 Échange d'information

Tout service propose un échange d'information. On examine ici ceux qui ne proposent que le seul échange d'information à l'exclusion de tout autre traitement.

Ils se distinguent par le niveau d'interaction et la structure de l'information telle que perceptible pour le réseau. En effet, l'information contenue dans un fichier peut être très complexe par rapport à un hypertexte linéaire mais cette complexité est cachée au réseau, il n'a pas à la gérer ni même à la prendre en compte.

Soit du moins interactif au plus interactif et du moins structuré au plus structuré :

✧ **Fichiers :**

Non seulement désire-t-on échanger des fichiers mais désire-t-on également manipuler une structure (généralement arborescente) de gestionnaires de fichiers comprenant accessoirement la possibilité de repérage, d'indexation.

✧ **Messages :**

L'objet est ici déjà plus complexe puisqu'il comprend une information d'acheminement (incluant des possibilités de diffusion) et un contenu (pouvant être composé de plusieurs parties de formats variés); une part d'automatisation dans l'envoi et la réception des messages augmente l'interactivité par rapport à la simple transmission de fichiers.

✧ **Hypertexte :**

L'objet est a priori de même complexité que le message mais l'interaction permet de moduler (par le choix des liens et la navigation) le contenu effectif de l'objet. Comme la cible de ces liens est répartie, modifiable dans le temps, l'objet est réellement dynamique. La navigation constitue l'archétype de l'interaction en réseautique.

14.3.2 Fourniture de services

L'idée qui présida initialement au développement de ce groupe de services fut de rendre accessible un service disponible dans un rayon restreint du point de service (jadis via la console de l'ordinateur ou un nombre limité de terminaux) à l'ensemble des ordinateurs et des postes de travail reliés au réseau.

Les premières applications à être réparties de la sorte furent des systèmes comptables et administratifs exploitant une base de données par l'entremise d'un nombre de logiciels le plus souvent produits à l'aide d'environnement de développement spécialisés propres aux grands systèmes de gestion de base de données.

Par la suite, plusieurs applications furent développées en utilisant dès le départ une architecture dite client-serveur. En plus de permettre d'étendre arbitrairement la zone de disponibilité du service, plusieurs autres bénéfices résultent d'une telle architecture :

- ✧ découplage naturel des fonctions entre client et serveur par l'entremise d'un protocole d'application, ce qui permet par la suite de faire évoluer les composants indépendamment dans le respect du protocole;
- ✧ possibilité de mettre rapidement en place un serveur de relève par simple ré-aiguillage des transactions vers celui-ci.

On y distingue normalement deux types de clients : les clients légers et les clients lourds selon qu'ils participent ou non de façon significative au traitement. Plus un client est léger, plus ses besoins en bande passante sont susceptibles d'être élevés, plus un client est lourd plus sa capacité de traitement et de stockage est susceptible d'être grande. Plus un client est lourd, plus sa base de logiciels d'exploitation est complexe et la gestion de sa configuration exigeante. La diminution comparativement plus importante des coûts en bande passante par rapport aux coûts de capacité de traitement, de stockage et (surtout) de gestion de configuration tend à privilégier les clients légers. Ce changement de tendance, relativement récent, est l'enjeu d'une bataille économique qui force plusieurs entreprises à se recentrer, à fusionner ou à disparaître.

Finalement, nous assistons à l'émergence des applications réparties où non seulement les points d'utilisation sont répartis mais aussi les points de service. Typiquement les informations et les traitements requis sont fournis par plusieurs ordinateurs qui doivent alors se synchroniser. Une telle architecture a pour objectifs :

- ✧ d'accroître la fiabilité et la tolérance aux pannes par la redondance des composants;
- ✧ d'augmenter la capacité de traitement disponible par la multiplicité des ressources (n ressources de taille 1 coûtant souvent beaucoup moins chère qu'une ressource de taille n tout en étant moins vulnérable);
- ✧ d'offrir un moyen facile d'augmenter les ressources de façon incrémentielle, souvent sans interruption de service.

14.3.3 Intégration

Une approche intégratrice permet de :

- ✧ réduire la gestion de configuration et minimiser l'impact sur les clients de l'ajout d'un nouveau service;
- ✧ agrandir la palette des plates-formes matérielles susceptibles d'accéder à un service;
- ✧ généraliser la plus grande partie possible des applications afin de réduire le développement aux seules parties spécifiques et corollairement, diminuer le coût et le temps de développement.

14.4 Besoins en gestion de configuration

Le succès de la réseautique à rendre accessibles les services dans un vaste territoire comporte aussi des risques supplémentaires:

- ✧ Comment, à quel coût, avec quelle infrastructure, peut-on gérer le parc d'équipements, fournir le soutien technique, logistique et applicatif aux utilisateurs?
- ✧ Comment faire évoluer, corriger les erreurs, mettre à jour ou ajouter des composants logiciels sans devoir intervenir sur place?

La gestion et la configuration des équipements à distance, grâce à des protocoles généralement reconnus par les producteurs d'équipements, est devenu une nécessité incontournable à l'évolution de la réseautique.

La gestion et la configuration des logiciels à distance répondent à la deuxième préoccupation. Initialement, chaque fournisseur de système d'exploitation a mis de l'avant une solution fermée. L'hétérogénéité des parcs informatiques rend toutefois cette approche inapplicable dans les grandes organisations. Des protocoles normalisés, indépendants des fournisseurs, se sont donc imposés.

Finalement, la surveillance, l'optimisation, la configuration, la mise à niveau, et l'ajout de protocoles des réseaux doivent pouvoir être automatisés en grande partie. De même, la collecte automatisée de données sur l'utilisation des ressources afin d'en planifier l'évolution est nécessaire.

L'ensemble de ces besoins tend à être desservi par une panoplie de protocoles unifiés en un même cadre général : CMISE pour le modèle OSI et SNMP pour la famille TCP.

14.5 Modèle général

14.5.1 Pourquoi revenir sur le modèle en couches

Pourquoi affiner notre compréhension du modèle en couches ?

Jusqu'à présent la décomposition d'une couche en services, protocoles et points de service nous a surtout permis de faciliter l'exposition rationnelle des concepts, principes et mises en œuvre proposés par le modèle OSI. Peu de gens sont effectivement impliqués dans le développement de nouveaux services offerts dans les couches de transport et leur configuration peut être réalisée grâce à des outils de haut niveau mis à disposition des ingénieurs en réseautique ayant à superviser et exploiter au quotidien des réseaux. Une certaine stabilité caractérise ces couches et le développement de nouveaux services est l'apanage de laboratoires spécialisés dont les résultats sont commercialisés sous une forme prête à être utilisée, souvent encapsulée dans un produit.

Les services des couches de haut niveau (application, présentation, session) sont beaucoup plus nombreux, évoluent plus rapidement et interagissent beaucoup plus entre eux que les services de bas niveau

(transport, réseau, liaison, physique). Ils sont le sujet d'une gestion de configuration constante qui nécessite, de la part des ingénieurs en réseautique, la compréhension fine des mécanismes d'interaction, donc du modèle sous-jacent.

14.5.2 Retour sur le modèle en couche

Les services mettent à disposition des fonctions. Les fonctions sont mises en œuvre par des composants (*entities*). Une fonction représente un traitement offert par un système ouvert. Les services et les fonctions font l'objet de normes permettant à des systèmes ouverts d'interagir sur une base commune. Il appartient au fournisseur d'un système ouvert d'élaborer les composants.

Si l'appel provient du même système ouvert, il appartient aux couches basses (par exemple la couche liaison) de faire l'éventuelle optimisation; dans le cadre de la présente discussion, le système ouvert se comporte par rapport à lui-même, dans une communication horizontale, comme par rapport à système tiers

Le mode d'utilisation des composants est différent selon qu'il est vertical (appel provenant d'une couche supérieure du même système ouvert) et horizontal (appel provenant d'une couche de même niveau d'un autre système ouvert). Le service est en quelque sorte l'interface verticale d'un regroupement de fonctions et le protocole l'interface horizontale. Il est possible de définir plusieurs regroupements de fonctions et pour chaque regroupement plusieurs services et plusieurs protocoles.

Le composant est un processus (matériel, logiciel ou hybride) entièrement exécuté par un même système ouvert. Il met en œuvre une ou plusieurs fonctions soit directement, soit par l'intermédiaire de services de la couche sous-jacente (du même système ouvert). Pour établir un dialogue avec le composant correspondant d'un autre système ouvert, il doit :

- ✧ soit établir une connexion hors norme;
- ✧ soit transmettre aux services appropriés de la couche sous-jacente de son propre système ouvert les objets respectant l'encodage et le protocole établi.

Dans un contexte normalisé, seules les couches physiques entrent effectivement en contact.

Un ensemble de systèmes, ouverts ou non, peut être regroupé de telle sorte à constituer un seul système ouvert. Dans un tel cas, le composant pourrait en fait être réparti entre plusieurs sous-système et répondre à la définition.

Une couche de communication est formée de l'ensemble des composants de même niveau des différents systèmes ouverts susceptibles de communiquer entre eux. Dès lors, il importe de pouvoir adresser uniformément et sans ambiguïté chacun des composants. Le composant ne doit cependant pas être immédiatement accessible (ni même visible) à l'extérieur du système ouvert afin de faciliter l'évolution indépendante des systèmes. Un découplage est requis. Celui-ci a été fourni par le service qui a aussi l'avantage d'apporter des informations supplémentaires sur le contexte d'utilisation du composant et sur les protocoles susceptibles d'être utilisés.

Ce seul niveau d'indirection n'est toutefois pas suffisant. Il faut prendre en compte que plusieurs « dialogues » peuvent se dérouler simultanément, dans la même couche. C'est ce qui motive, notamment,

la nécessité d'ajouter un intermédiaire, le point de service, et d'inclure dans l'adressage la désignation du point de connexion.

Un service peut être impliqué dans plusieurs dialogues simultanément, il importe de pouvoir déterminer rapidement duquel lui provient une requête. Le point de service (*SAP, service access point*) permet de distinguer les communications verticales entre elles. Il est le truchement par lequel un composant de niveau n+1 obtient un service d'un composant de niveau n. Les caractéristiques suivantes sont maintenues en tout temps :

- ✧ un SAP et le composant qui le dessert appartiennent à la même couche du même système ouvert;
- ✧ tout SAP est desservi par au plus un composant;
- ✧ un composant peut desservir un nombre arbitraire de SAP;
- ✧ un SAP dessert au plus un composant de niveau supérieur;
- ✧ un composant peut être desservi par un nombre arbitraire de SAP de niveau inférieur.

Le point de connexion permet quant à lui de distinguer les communications horizontales entre elles.

[analogie : forum de discussion. plusieurs personnes se réunissent dans une salle pour établir une nouvelle norme (sur la couleur des fils téléphoniques). Chacune peut être vue comme un système ouvert dont un des composants est le module de conversation au sujet des fils téléphoniques. Des sous-groupes se forment par affinité, il s'agit ici d'un dialogue ; certaines personnes font partie de plus d'un sous-groupe et devront maintenir plus d'un fil de discussion].

L'établissement d'une communication bout en bout nécessite de convenir à la fois d'un service et d'un protocole, l'élément de service (*SE, service element*) est le résultat de la combinaison d'un service et d'un protocole. Les SE sont divisés en deux catégories : les éléments de service spécialisés (*SSE, specific service element*) et les éléments de service communs (*CSE, common service element*).

Les SSE sont les services d'intérêt général offert par une couche. De nouveaux services et protocoles sont introduits au fur et à mesure des besoins. Typiquement, ils sont nombreux et une mise en œuvre ou un équipement installé peut n'en offrir qu'une partie.

Les CSE sont des services d'intendance utiles autres services. Typiquement, ils sont peu nombreux mais offerts par toute mise en œuvre et présents sur la quasi-totalité des équipements installés.

En fait, un équipement installé omettra rarement un CSE à moins que son offre de service soit clairement limitée à des SSE dont aucun ne le requiert (ce qui limitera, par voie de conséquence, les choix offerts aux autres partenaires avec lesquels l'équipement installé est susceptible de communiquer).

Le modèle OSI dénote les composants par des titres, les points de service par des adresses et les points de connexion par des identificateurs. Ce sont les éléments constitutifs utilisés par les deux systèmes d'adressage du modèle : le système hiérarchique et le système tabulaire. Dans le présent cours nous désignerons par adresse complète les adresses issues d'un de ces systèmes afin de les distinguer des adresses des points de service.

Le système hiérarchique consiste à construire successivement les adresses de niveau en niveau en utilisant l'adresse de niveau inférieur à laquelle un suffixe (adresse de point de service ou identificateur de point de connexion) est ajouté. Ce système, simple à mettre en œuvre, est toutefois limité : seules des associations simples, n à 1 et permanentes peuvent être représentées.

Le système tabulaire est conçu afin de lever les limites du système hiérarchique. Il consiste à maintenir une table associative indépendante constituée de chaque paire de niveaux mitoyens. Cette technique est similaire à celle utilisée pour représenter les relations n à n dans les systèmes de gestion de bases de données relationnelles. Il est possible de modifier une liaison rapidement et efficacement tout en levant la restriction de cardinalité de n à 1. Elle ne permet toutefois pas de retracer, sans consulter la table, les adresses de niveaux inférieurs sur la simple base de l'adresse hiérarchique.

Dans les deux cas, un répertoire permet de conserver la liaison entre l'adresse complète et ses éléments constitutifs qui peuvent varier dans le temps. Cette variation est requise dans plusieurs contextes, notamment la ré-allocation des ressources et la récupération des erreurs. Pour cette raison, les adresses complètes doivent être considérées comme indivisibles.

L'architecture de traitement et d'adressage étant en place, il est temps de parler des données qui circuleront le long de cette infrastructure. Chaque niveau prescrit un certain nombre d'unités (*data unit*) construites par encapsulation de l'unité de niveau supérieur dans une enveloppe comprenant facultativement un préfixe et un suffixe. Diverses opérations peuvent intervenir lors de la composition de l'unité (*segmentation, blocking, concatenation*) en une transformation portant sur l'unité en provenance de la couche supérieure (notamment dans la couche présentation, cette transformation peut être importante : transcodage, compression, chiffrement).

Les unités sont classées en catégories génériques afin de faciliter leur acheminement et leur traitement (*protocol-control-information, user-data, interface-control-information, interface-data*).

14.5.3 La définition des services et la spécification des protocoles

Le modèle en lui-même n'est pas suffisant pour permettre d'établir une base commune d'interopérabilité entre systèmes ouverts. Il faut définir

les fonctions et les services, spécifier les protocoles. Un ensemble de conventions a été établi afin d'uniformiser la présentation des définitions et des spécifications. Une présentation synthétique est disponible dans JA 3, les normes applicables sont x.210 (iso 8509) et x.220 (iso 8xxx).

14.5.4 La spécification des données et de l'encodage

La grande variété des encodages ...

- ✧ le lexique,
- ✧ la syntaxe,
- ✧ la sémantique,
- ✧ la pragmatique,
- ✧ une convention asn.1, limitée au lexique et à la syntaxe.

Malheureusement les concepteurs n'avaient pas suivi de cours de langages formels d'où le monstre : le langage de description lui-même pose des problèmes d'analyse et d'interprétation. À la limite ADA eût fourni une meilleure base formelle qui fut proposée par les informaticiens. Malheureusement les ingénieurs gagnèrent et depuis les informaticiens sont pris à réparer les pots cassés!

14.5.5 Récapitulation

fonction

Un traitement mis en œuvre par un composant.

La spécification normalisée d'une fonction comporte normalement des antécédents et des conséquents.

service

L'ensemble des règles régissant la communication verticale entre deux composants.

La spécification normalisée d'un service comporte normalement une description d'interface et des conditions préalables d'utilisation.

protocole

L'ensemble des règles, notamment temporelles, régissant la communication horizontale entre deux composants.

La spécification normalisée d'un protocole comporte normalement des diagrammes de transition et des diagrammes de séquence.

point de service

Un intermédiaire par lequel un composant de niveau n+1 peut utiliser les services d'un composant de niveau n au sein d'un même système ouvert.

Les points de service sont désignés par préfixation de la couche : [x]SAP : *service access point* pour x parmi (Presentation, Session, Transport, Network, data Link, Physic).

point de connexion

Un intermédiaire par lequel un composant peut utiliser les services d'un composant d'un autre système ouvert; les deux composants étant de même niveau.

Les points de connexion sont désignés par préfixation de la couche : [x]CEP : *connection end point* pour x parmi (Presentation, Session, Transport, Network, data Link, Physic).

adressage

Il existe deux systèmes d'adressage dans le modèle OSI: hiérarchique et tabulaire.

Le système hiérarchique consiste à construire successivement les adresses de niveau en niveau en utilisant l'adresse de niveau inférieur à laquelle un suffixe est ajouté.

Le système tabulaire consiste à maintenir une table associative indépendante constituée de chaque paire de niveaux mitoyens.

données

L'unité de donnée (*data unit*) d'un niveau est construite par encapsulation de l'unité de niveau supérieur dans une enveloppe comprenant facultativement un préfixe et un suffixe. Des opérations sur l'unité en provenance de la couche supérieure qui peuvent la transformer de façon importante un

Diverses opérations peuvent intervenir lors de la composition de l'unité (*segmentation, blocking, concatenation*) en une transformation portant sur l'unité en provenance de la couche supérieure (notamment dans la couche présentation, cette transformation peut être importante : transcodage, compression, chiffrement).