

# Sommaire

- I. Terminologie OSI
- II. La couche Session
- III. La couche Présentation
- IV. La couche Application**
- V. Les applications normalisées

## IV. La couche Application

### ➤ Rôle de la couche Application

- Elle est la seule couche à fournir des services directement aux Processus d'Application (APs)
- Elle fournit nécessairement tous les services OSI directement utilisables par des APs
- Les éléments constitutifs de cette couche bénéficient
  - ✓ Des facilités de représentation de l'information offertes par la couche Présentation
  - ✓ Des mécanismes de contrôle des interactions procurés par la couche Session

## IV. La couche Application

### ➤ Architecture de la couche Application

- La diversité des applications constitue un obstacle à leur intégration au sein de la couche Application
  - ✓ La norme ISO9545 propose une architecture modulaire et unifiée de la couche application appelée **Application Layer Structure** (ALS) considérée comme un modèle de référence pour la normalisation des applications
  - ✓ L'ALS décrit la composante **communication des APs**, son but c'est d'assurer la propriété **ACID** pour toute application dans un contexte réparti

## IV. La couche Application

### ➤ Propriétés ACID

- **Atomicité** = un ensemble d'opérations doit être vu comme un tout indivisible
- **Cohérence** = si un ensemble d'opérations s'exécute à partir d'un état cohérent, il conduit le système dans un état cohérent (vérifiant les contraintes d'intégrité de l'application)
- **Isolation** = c'est une propriété qui exige que les effets d'un ensemble d'opérations soient masqués par rapport à un autre ensemble d'opérations s'exécutant concurremment
- **Durabilité** = cette propriété exige que les effets des opérations ne doivent pas être altérés par une défaillance applicative ou de communication

## IV. La couche Application

### ➤ Modèle générique

#### ▪ Processus d'Application (AP)

- ✓ C'est la représentation abstraite des éléments d'un système ouvert réel qui réalisent, sur différents systèmes, le traitement de l'information pour une application répartie et qui **communiquent entre eux** pour exécuter collectivement leur travail
- ✓ Un AP peut être
  - manuel, lorsqu'il correspond au travail d'un opérateur sur un terminal
  - automatique, lorsqu'il correspond à l'exécution d'un programme
- ✓ Plusieurs AP peuvent collaborer à une même tâche si on assure
  - interconnexion : fonctions de mise en relation des AP, organisation du dialogue
  - interfonctionnement : procédures à suivre, types d'information (syntaxe de transfert commune)
  - transport des données : transfert de bout en bout

## IV. La couche Application

### ➤ Modèle générique

#### ▪ Processus d'Application (AP)

- ✓ L'AP représente l'ensemble des ressources d'un système ouvert réel qui permettent le traitement de l'information
  - il peut être considéré comme un objet statique
- ✓ L'exécution d'un traitement défini par un AP est appelée **Instance de Processus d'Application (API)**
  - à un instant donné, il peut exister plusieurs invocations d'un même AP sur un même système
  - chacune est repérée par un identificateur **APIID**

## IV. La couche Application

### ➤ Modèle générique

#### ▪ Entité d'Application (AE)

- ✓ Un AP peut être divisé en 2 parties
  - la première est locale, spécifique d'un système donné
  - la seconde est la partie communication, elle gère les relations avec d'autres AP
- ✓ Les AP comportent un ou plusieurs éléments gérant les communications, ce sont les **Entités d'Application**
  - les AP accèdent aux services OSI par l'intermédiaire des AE
- ✓ Chaque AE est mise en correspondance avec la couche Présentation par un ou plusieurs PSAP à des services de présentation

## IV. La couche Application

### ➤ Modèle générique

#### ▪ Entité d'Application (AE)

- ✓ L'interface entre l'AP et l'AE est l'**Élément Utilisateur (UE)**
  - une AE n'est attachée qu'à un seul AP
  - MAIS, un AP peut compter plusieurs AE
- ✓ Une activité (exécution) particulière de l'AE est appelée **Instance d'Entité d'Application (AEI)**
  - elle modélise les fonctions de communication d'une activité de communication particulière d'une API

## IV. La couche Application

### ➤ **Modèle générique**

#### ▪ **Élément de Service d'Application (ASE)**

- ✓ Les diverses fonctionnalités d'une AE sont réparties en classes
  - chaque classe est un **Élément de Service d'Application**
  - un ASE peut utiliser les services d'autres ASE
- ✓ Les ASE utilisés par une AE ainsi que les règles définissant les interactions entre ces ASE constituent le **contexte d'application**

## IV. La couche Application

### ➤ **Modèle générique**

#### ▪ **Association d'Application (AA)**

- ✓ La relation établie entre 2 AEI pour leur communication et leur coordination est appelé **Association d'Application**
  - une AA ne peut être créée qu'après l'activation de 2 AE correspondantes
  - un **Identificateur d'Association d'Application (AAI)** détermine de façon unique une association et fournit des informations sur chacune des AEI participant à l'association

## IV. La couche Application

### ➤ **Modèle générique**

#### ▪ **Fonction de Contrôle d'Association Unique (SACF)**

- ✓ Les ASE utilisés par une AE pendant une AA donnée appartiennent à un **Objet d'Association Unique (SAO)** qui comprend
  - l'**Élément de Service de Contrôle d'Association (ACSE)**
  - un ou plusieurs autres ASE
  - une **Fonction de Contrôle d'Association Unique (SACF)** assurant la coordination entre les différents ASE utilisés
- ✓ La durée de vie d'un SAO est celle de l'association (AA)

## IV. La couche Application

### ➤ **Modèle générique**

#### ▪ **Fonction de Contrôle d'Associations Multiples (MACF)**

- ✓ Un même AP peut utiliser simultanément plusieurs AA grâce aux règles suivantes
  - un AP peut utiliser simultanément plusieurs AE différentes
  - un AP peut utiliser plusieurs instances de la même AE
  - un API peut utiliser simultanément plusieurs AA différentes (cas d'une application répartie)
    - nécessité de synchroniser le travail réalisés par les différentes AA
- ✓ Cette coordination est assurée par une Fonction de Contrôle d'Associations Multiples (MACF)

## IV. La couche Application

### ➤ Adressage au niveau de la couche Application

- L'accès à une application distante nécessite la connaissance de son adresse qui comprend les éléments suivants
  - ✓ les adresses relatives aux couches 3 à 6
    - l'adresse du système : NSAP
    - l'adresse de Transport : TSAP
    - l'adresse de Session : SSAP
    - l'adresse de Présentation : PSAP
  - ✓ les critères de sélection propres à la couche application (facultatifs si pas d'ambiguïté)
    - le titre d'entité d'application composé de 2 champs
      - le titre de processus d'application : identifie l'AP et doit être précisé quand plusieurs AP partagent la même adresse de présentation
      - le qualificateur de l'entité d'application : identifie l'AE et doit être précisé si l'AP utilise plusieurs AE.
    - l'identificateur de l'API : identifie une invocation particulière du AP
    - l'identificateur de l'AEI : identifie une invocation particulière d'une AE de l'AP

## IV. La couche Application

### ➤ Les ASE de base

- Chaque ASE est en fait une syntaxe abstraite définissant un service normalisé et un protocole permettant de dialoguer avec l'ASE distante
- Deux types d'ASE
  - ✓ des ASE de base communs aux différentes applications
  - ✓ des ASE fonctionnels spécifiques à des applications particulières
- ASE de base
  - ✓ ACSE
  - ✓ RTSE
  - ✓ ROSE
  - ✓ CCRSE

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Contrôle d'Association (ACSE)

- ✓ Cet élément joue un rôle essentiel parmi les outils généraux
  - permet d'établir et de terminer les AA
  - point de passage obligé de toute AE avec la connexion de Présentation
- ✓ Chaque AE comprend un ACSE qui, lors de l'établissement d'une association, règle la négociation et la définition du contexte d'application
- ✓ Une AA s'appuie sur une connexion de Présentation, l'ACSE s'appuie alors sur les 4 services suivants
  - la connexion de Présentation
  - la terminaison de Présentation
  - la coupure par l'utilisateur du service de Présentation
  - la coupure par le fournisseur du service de Présentation

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Contrôle d'Association (ACSE)

- ✓ ACSE fournit 4 services
  - Établissement d'une association
    - service confirmé comportant 4 primitives (A-ASSOCIATE.xxx)
  - Terminaison d'une association
    - service confirmé comportant 4 primitives (A-RELEASE.xxx)
  - Coupure d'une association par l'utilisateur
    - service non confirmé comportant 2 primitives (A-ABORT.xxx)
  - Coupure d'une association par le fournisseur de service
    - service non confirmé comportant 1 primitive (A-P-ABORT.xxx)



## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Contrôle d'Association (ACSE)

##### ✓ Établissement d'une association

- La primitive A-ASSOCIATE.request : définit l'AE distante par son adresse et permet de négocier les options des couches Présentation et Session. Elle comporte les paramètres
  - nom du contexte d'application
  - titre de l'AP (appellant, appelé)
  - qualificateur de l'AE (appelante, appelée)
  - identificateur de l'API (appellant, appelé)
  - identificateur de l'AEI (appelante, appelée)
  - données de l'utilisateur
  - adresse de présentation (appellant, appelé) (P)
  - liste des contextes de Présentation (P)
  - nom du contexte de Présentation par défaut (P)
  - QoS (S)
  - unités fonctionnelles de présentation (P), de session (S)
  - numéro de série (S)
  - attribution des jetons (S)
  - identificateur de la connexion de session (S)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Contrôle d'Association (ACSE)

##### ✓ Établissement d'une association

- La primitive A-ASSOCIATE.indication : contient les mêmes paramètres que la requête
  - le fournisseur du service ACSE y ajoute le paramètre *liste de résultats de définition des contextes de présentation* récupéré dans la primitive d'indication de connexion de présentation
- La primitive A-ASSOCIATE.response : permet d'accepter ou de refuser une association et de fixer les valeurs des paramètres négociables des couches Présentation et Session
  - résultat de l'association (acceptation, refus temporaire, refus permanent)
  - diagnostic (présent qu'en cas de refus)
- La primitive A-ASSOCIATE.confirm : contient les mêmes paramètres que la réponse

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Contrôle d'Association (ACSE)

##### ✓ Terminaison d'une association

- La primitive A-RELEASE.request comporte les paramètres
  - raison : 2 raisons sont prévues (terminaison normale, terminaison urgente), 3 valeurs possibles du paramètre (normale, urgente, définie par l'utilisateur)
  - données de l'utilisateur
- La primitive A-RELEASE.indication reprend les paramètres de la requête
- La primitive A-RELEASE.response permet d'accepter ou de refuser une demande de terminaison d'association et comporte les paramètres
  - raison : les valeurs possibles du paramètre (normale, non terminée, définie par l'utilisateur)
  - données de l'utilisateur
  - résultat (acceptation ou refus)
- La primitive A-RELEASE.indication reprend les paramètres de la réponse

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Contrôle d'Association (ACSE)

##### ✓ Coupure d'une association par l'utilisateur

- La primitive A-ABORT.request : permet à l'utilisateur de mettre fin à l'association, comporte comme paramètre
  - données de l'utilisateur
- La primitive A-ABORT.indication comporte comme paramètres
  - origine de la libération: 2 valeurs possibles (utilisateur du service ACSE, fournisseur du service ACSE)
  - données de l'utilisateur

##### ✓ Coupure d'une association par le fournisseur du service

- La primitive A-P-ABORT.indication : permet au fournisseur, à la suite d'une erreur signalée par le service de présentation, de mettre fin à l'association, comporte comme paramètre
  - la raison fournie par le service de présentation

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ Cet élément permet l'échange d'APDU avec reprise et contrôle de flux entre 2 AE
  - cet élément garantit que chaque APDU est transmise entièrement et une seule fois
  - cet élément assure la reprise en cas d'incident de communication et minimise la quantité d'information à retransmettre après incident
  - cet ASE est particulièrement utile pour la messagerie X.400 qui peut avoir à transférer de grands volumes de données à travers des circuits de communication complexes
- ✓ RTSE assure
  - l'établissement et la terminaison de l'AA par l'intermédiaire de ACSE
  - le transfert de données en utilisant la gestion d'activités
    - le transfert d'une APDU est associé à une activité

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ Quand un contexte d'application contient RTSE (donc ACSE), les fonctions SACF doivent imposer les règles suivantes
  - le fournisseur de RTSE doit être le seul utilisateur des services de ACSE
    - ceci permet à RTSE d'être prévenu en cas d'incident, de tenter de récupérer la connexion de présentation pour établir une nouvelle association et de reprendre le traitement de manière transparente
  - le fournisseur de RTSE doit être le seul utilisateur des services de présentation suivants
    - gestion des connexions
    - transfert de données
    - demande de jetons
    - passation de contrôle
    - gestion d'activités
    - pose de points MI
    - signalisation d'anomalies

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ Pour minimiser la quantité d'information à retransmettre en cas d'incident, l'APDU peut être segmentée
  - la transmission d'un segment peut alors être suivi par la pose d'un point MI avec demande de confirmation
  - ce mode de transmission est contrôlé par 2 paramètres
    - la période des points de reprise en unités de 1024 octets
    - la taille de la fenêtre limitant le nombre de MIP non acquittés à un instant donné
  - pas de segmentation par défaut (valeur nulle du paramètre *checkpoint size*)
    - si la segmentation est autorisée, la fenêtre est par défaut égale à 3

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ Le transfert de données peut être unidirectionnel ou bidirectionnel à l'alternat
  - ceci nécessite l'utilisation du jeton de données
- ✓ En fait, le fonctionnement de RTSE nécessite qu'à chaque instant, le jeton de données, le jeton MI et le jeton MA et d'activité soient détenus par le même utilisateur
  - l'ensemble de ces trois jetons constitue le tour
- ✓ La terminaison normale de l'association ne peut être demandée que par l'utilisateur qui l'a établie
  - il doit être en possession du tour

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ RTSE fournit 6 services
  - Établissement d'une association
    - service confirmé comportant 4 primitives (RT-OPEN.xxx)
  - Terminaison d'une association
    - service confirmé comportant 4 primitives (RT-CLOSE.xxx)
  - Coupure d'une association par l'utilisateur
    - service non confirmé comportant 2 primitives (RT-U-ABORT.xxx)
  - Coupure d'une association par le fournisseur de service
    - service non confirmé comportant 1 primitive (RT-P-ABORT.xxx)
  - Demande de transfert
    - service confirmé comportant 3 primitives (RT-TRANSFER.xxx)
  - Gestion du tour
    - service non confirmé de demande de tour comportant 2 primitives (RT-TURN-PLEASE.xxx)
    - service non confirmé de cession du tour comportant 2 primitives (RT-TURN-GIVE.xxx)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ Établissement d'une association
  - La primitive RT-OPEN.request : comporte les paramètres
    - mode du dialogue : unidirectionnel, bidirectionnel à l'alternat
    - tour initial : peut être accordé au demandeur ou à l'accepteur de l'association
    - données de l'utilisateur
    - nom du contexte d'application (A)
    - titre de l'AP (appellant, appelé) (A)
    - qualificateur de l'AE (appelante, appelée) (A)
    - identificateur de l'API (appellant, appelé) (A)
    - identificateur de l'AEI (appelante, appelée) (A)
    - adresse de présentation (appellant, appelé) (P)
    - liste des contextes de Présentation (P)
    - nom du contexte de Présentation par défaut (P)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

##### ✓ Établissement d'une association

- La primitive RT-OPEN.indication : reprend les paramètres de la requête. Le fournisseur du service y ajoute les résultats des définitions des contextes de Présentation
  - liste des résultats de définition des contextes de Présentation (P)
  - résultat du contexte de Présentation par défaut (P)
- La primitive RT-OPEN.response : reprend les paramètres de l'indication
  - résultat de l'association (A) : acceptation, refus temporaire ou refus permanent
  - diagnostic (A) : le contenu du champ dépend de l'origine du refus (refus par l'utilisateur du service RTSE, refus par le fournisseur du service RTSE)
- La primitive RT-OPEN.confirm : reprend les paramètres de la réponse, indique l'origine du résultat (utilisateur du service RTSE, fournisseur du service RTSE, fournisseur du service de Présentation)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

##### ✓ Terminaison d'une association

- La primitive RT-CLOSE.request : comporte les paramètres
  - raison (A)
  - données de l'utilisateur (A)
- La primitive RT-CLOSE.indication : reprend les paramètres de la requête
- La primitive RT-OPEN.response : permet d'accepter ou de refuser la terminaison d'une association, comporte les paramètres
  - raison (A)
  - données de l'utilisateur (A)
- La primitive RT-OPEN.confirm : reprend les paramètres de la réponse

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ Coupure d'une association par l'utilisateur
  - La primitive RT-U-ABORT.request : permet à l'utilisateur de mettre fin à une association en cas d'anomalie, peut comporter comme paramètre
    - données de l'utilisateur
  - La primitive RT-U-ABORT.indication : comporte éventuellement les données de l'utilisateur
- ✓ Coupure d'une association par le fournisseur du service
  - La primitive RT-P-ABORT.indication : ne comporte pas de paramètre

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

- ✓ Demande de Transfert :
  - RTSE utilise ACSE uniquement pour établir ou libérer une association
    - les autres opérations de RTSE sont effectuées par accès direct aux services de Présentation avec mise en correspondance entre les primitives de présentation et les primitives RTSE
  - RTSE est un service confirmé comportant 3 primitives
    - pas de primitive de réponse car la confirmation est émise directement par le fournisseur du service RTSE distant
  - La primitive RT-TRANSFER.request : comporte les paramètres
    - APDU à transmettre
    - délai maximal autorisé pour la transmission
  - La primitive RT-TRANSFER.indication : comporte comme paramètre l'APDU
  - La primitive RT-TRANSFER.confirm : comporte comme paramètres
    - APDU transférée
    - résultat de l'opération

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Transfert Fiable (RTSE)

##### ✓ Gestion du tour :

- avec RTSE, les transferts de données ne peuvent être effectués qu'avec une interaction *unidirectionnelle*, ou le plus souvent, *bidirectionnelle à l'alternat*
- La primitive RT-TURN-PLEASE.request : comporte comme paramètre la priorité
  - priorité diminue quand la valeur du paramètre augmente (valeur 0 correspond à la priorité maximale, utilisée pour demander le tour en vue de terminer l'association)
  - si non précisée, la valeur par défaut est 0
- La primitive RT-TURN-PLEASE.indication : comporte un seul paramètre, la priorité
- La primitive RT-TURN-GIVE.request : pas de paramètre
- La primitive RT-TURN-GIVE.indication : pas de paramètre

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

- ✓ Cet élément permet à une AE de faire exécuter des opérations par une autre AE située le plus souvent sur un autre système
  - ROSE est destiné à couvrir des besoins où la communication est interactive (mettre en œuvre un dialogue de type requête/réponse)
  - ROSE est utilisé en particulier pour la gestion de réseaux par l'intermédiaire de CMISE
- ✓ La réalisation d'opérations distantes nécessite l'établissement d'une AA entre deux AE
  - celle qui demande l'opération (invoker)
  - celle qui l'effectue (performer)
  - le transfert des données se fait le plus souvent par accès direct au service de Présentation
    - possibilité d'inclure RTSE dans le contexte d'application



## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

- ✓ Chaque opération peut faire l'objet d'un compte rendu, plusieurs possibilités sont prévues
  - fourniture d'un CR positif (**result**) en cas de succès, ou négatif (**error**) en cas d'échec
  - fourniture d'un CR uniquement en cas d'échec (CR du type **error**)
  - fourniture d'un CR uniquement en cas de succès (CR du type **result**)
  - aucun CR
- ✓ Les opérations peuvent être effectuées de manière
  - synchrone = nécessité d'attendre la fin d'une opération avant d'en soumettre une autre
  - asynchrone = plusieurs opérations peuvent être soumises en parallèle

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

- ✓ Une opération synchrone nécessite la fourniture d'un compte rendu dans tous les cas
  - la réception de ce CR indique la fin de l'opération
- ✓ Cinq classes d'opération
  - classe 1 : OP synchrone avec CR en cas de succès ou d'échec
  - classe 2 : OP asynchrone avec CR en cas de succès ou d'échec
  - classe 3 : OP asynchrone avec CR en cas d'échec
  - classe 4 : OP asynchrone avec CR en cas de succès
  - classe 5 : OP asynchrone sans CR

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

- ✓ Possibilité de distinguer les rôles des 2 AE (celle qui a demandé l'association, celle qui l'a acceptée)
  - trois classes d'association
    - classe 1 : seule l'AE qui a demandé l'association peut demander l'exécutions d'opérations
    - classe 2 : seule l'AE qui a accepté l'association peut demander l'exécutions d'opérations
    - classe 3 : les deux AE peuvent demander l'exécutions d'opérations
- ✓ Possibilité d'effectuer des opérations liées
  - une AE demande l'exécution d'une opération (**mère**) à une AE distante
  - l'AE distante peut être amenée à faire exécuter par une autre AE (par exemple la première AE) une ou plusieurs opérations (**filles**)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

- ✓ Une opération est définie par les informations suivantes
  - la nature des paramètres à transmettre à l'AE distante
  - la classe de l'opération (synchrone, asynchrone)
  - les conditions de transmission d'un CR (en cas de succès, en cas d'échec...)
  - la possibilité d'effectuer des opérations liées
  - des niveaux de priorité associés à l'exécution de l'opération et à la transmission du CR
  - un identificateur unique (*invokeId*)
- ✓ Les opérations définies dans une utilisation particulière de ROSE sont regroupées en ensembles (*operation package*), pouvant être
  - symétrique : les opérations sont disponibles pour les 2 AE
  - asymétrique : il contient 2 sous-ensembles (un par AE) d'opérations disponibles; les AE jouent alors des rôles distincts (fournisseur, consommateur)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

- ✓ L'établissement et la terminaison de l'association peuvent être assurés indépendamment de ROSE, par utilisation directe de ACSE ou de RTSE
- ✓ On peut également effectuer ces opérations de manière dynamique par l'utilisation de services spécifiques de ROSE
  - bind : lors de l'établissement de l'association
  - unbind : lors de la terminaison de l'association
  - ces opérations particulières constituent l'ensemble de connexion (*connection package*)
- ✓ Les primitives de ROSE concernent
  - les services liés à l'établissement et la terminaison de l'association (bind, unbind)
  - les services permettant de transmettre les requêtes et les comptes rendus des opérations distantes

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

- ✓ ROSE fournit 7 services
  - Établissement d'une association
    - service confirmé comportant 4 primitives (RO-BIND.xxx)
  - Terminaison d'une association
    - service confirmé comportant 4 primitives (RO-UNBIND.xxx)
  - Appel de fonction
    - service non confirmé comportant 2 primitives (RO-INVOKE.xxx)
  - Transmission du résultat d'une opération réussie
    - service non confirmé comportant 2 primitives (RO-RESULT.xxx)
  - Transmission de l'erreur associée à une opération qui a échoué
    - service non confirmé comportant 2 primitives (RO-ERROR.xxx)
  - Rejet par l'utilisateur
    - service non confirmé comportant 2 primitives (RO-REJECT-U.xxx)
  - Rejet par le fournisseur du service
    - service non confirmé comportant 1 primitive (RO-REJECT-P.xxx)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

##### ✓ Établissement d'une association

- La primitive RO-BIND.request : comporte 3 paramètres
  - nom du contexte d'application
  - type de l'opération (synchrone, asynchrone)
  - paramètre de l'opération bind
- La primitive RO-BIND.indication : reprend les paramètres de la requête. Le fournisseur du service y ajoute les résultats des définitions des contextes de Présentation
  - nom du contexte d'application (repris depuis la requête)
  - paramètre de l'opération (si présent dans la requête)
  - indicateur précisant si le refus de terminaison de l'association est possible

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

##### ✓ Établissement d'une association

- La primitive RO-BIND.response : indique l'acceptation ou le refus de l'association et comporte les paramètres
  - résultat de l'association (acceptation, refus)
  - paramètre associé à l'acceptation
  - paramètre associé au refus
- La primitive RO-BIND.confirm : reprend les paramètres de la réponse

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

##### ✓ Terminaison d'une association

- La primitive RO-UNBIND.request : peut contenir un paramètre type de l'opération
- La primitive RO-UNBIND.indication : reprend le paramètre éventuel de la requête
- La primitive RO-UNBIND.response : comporte un ou 2 paramètres
  - résultat de l'opération (succès ou échec de l'opération)
  - paramètre associé au succès
  - paramètre associé à l'échec
- La primitive RO-UNBIND.confirm : reprend les paramètres de la réponse

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

##### ✓ Appel de fonction

- La primitive RO-INVOKE.request : comporte les paramètres
  - nature de l'opération
  - type de l'opération (synchrone, asynchrone)
  - argument (à passer à l'opération)
  - identificateur de l'appel
  - identificateur de l'appel lié (cas d'une opération fille)
  - priorité (gestion du tour par RTSE)
- La primitive RO-INVOKE.indication : comprend les paramètres
  - nature de l'opération
  - argument (si présent dans la requête)
  - identificateur de l'appel
  - identificateur de l'appel lié

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

##### ✓ Transmission du résultat d'une opération réussie

- La primitive RO-RESULT.request : comporte les paramètres
  - nature de l'opération
  - résultat
  - identificateur de l'appel
  - priorité (utilisé par RTSE)
- La primitive RO-RESULT.indication : comprend les paramètres
  - nature de l'opération
  - résultat
  - identificateur de l'appel

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

##### ✓ Transmission de l'erreur associée à une opération qui a échoué

- La primitive RO-ERROR.request : comporte les paramètres
  - nature de l'erreur
  - paramètre de l'erreur
  - identificateur de l'appel
  - priorité (utilisé par RTSE)
- La primitive RO-ERROR.indication : comprend les paramètres
  - nature de l'erreur
  - paramètre de l'erreur (si présent dans la requête)
  - identificateur de l'appel

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service d'Opérations Distantes (ROSE)

##### ✓ Rejet par l'utilisateur

- La primitive RO-REJECT-U.request : permet à un utilisateur de rejeter une opération ou une réponse, comporte les paramètres
  - raison du refus
  - identificateur de l'appel
  - priorité (utilisé par RTSE)
- La primitive RO-REJECT-U.indication : comprend les paramètres
  - raison du refus
  - identificateur de l'appel

##### ✓ Rejet par le fournisseur du service

- La primitive RO-REJECT-P.indication : comprend les paramètres
  - identificateur de l'appel
  - raison du rejet

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrency et Recouvrement (CCRSE)

##### ✓ C'est l'élément de service définissant la façon de coordonner des actions atomiques (possédant les propriétés ACID) dans le déroulement d'applications réparties

- utilisé principalement pour régler les conflits d'accès à l'information et pour maintenir le système dans un état cohérent malgré les pannes

##### ✓ CCRSE est utilisé par les applications

- RDA (Remote Database Access)
- TP (Transaction Processing)
- JTM (Job Transfer and Manipulation)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrence et Recouvrement (CCRSE)

##### ✓ Principe de fonctionnement du CCRSE

- Le CCRSE contrôle l'atomicité des opérations par un mécanisme d'engagement en 2 temps (2PC/OSI-TP)
  - un processus maître (ou supérieur) demande à tous ses subordonnés s'ils sont prêts à s'engager dans une opération et attend toutes les réponses
  - en fonction des réponses obtenues, il ordonne la validation ou l'abandon de l'opération

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrence et Recouvrement (CCRSE)

##### ✓ Principe de fonctionnement du CCRSE

- Une action atomique peut concerner un nombre quelconque d'utilisateurs du service CCRSE
- L'utilisateur qui démarre l'action en est le maître
  - le maître envoie une requête à un ou plusieurs autres utilisateurs
  - ceux-ci vont pouvoir à leur tour soumettre des requêtes à d'autres utilisateurs



## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrency et Recouvrement (CCRSE)

##### ✓ Principe de fonctionnement du CCRSE

- L'action atomique va entraîner la création d'une structure arborescente, l'arbre d'action atomique
- Chaque branche de l'arbre part du supérieur vers le subordonné
- Cet arbre comporte 3 types de nœuds
  - un nœud racine (unique) qui correspond au maître de l'action
  - des nœuds intermédiaires qui jouent à la fois le rôle de subordonné et de supérieur
  - des nœuds feuilles (au moins un) qui jouent uniquement le rôle de subordonné

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Principe de fonctionnement du CCRSE

##### ✓ Une action atomique est effectuée en deux étapes :

- le maître demande à ses subordonnés s'ils sont en mesure d'exécuter l'opération et attend l'ensemble des réponses (entre dans la phase 1)
- chaque subordonné vérifie qu'il peut effectuer l'opération, et s'il s'agit d'un nœud intermédiaire, transmet la demande à ses subordonnés et attend leurs réponses (entre dans la phase 1)
  - transmission de proche en proche de la racine vers les feuilles de l'arbre
- quand un nœud intermédiaire a terminé ses vérifications et a obtenu toutes les réponses attendues, il transmet à son supérieur
  - soit une réponse positive (sauvegarde des données liées dans l'état initial et dans l'état final)
  - soit une réponse négative (retour en arrière en laissant les données liées dans l'état initial)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Principe de fonctionnement du CCRSE

✓ Une action atomique est effectuée en deux étapes :

- le maître, après avoir collecté les réponses de ses subordonnés, décide
  - si les réponses sont positives, d'exécuter l'action : il demande la validation et attend les confirmations de validation (il est dans la phase 2)
  - sinon d'annuler l'action : il demande le retour en arrière
- un nœud, autre que le maître, qui reçoit l'ordre de validation entre dans la phase 2
  - met ses données liées dans l'état final
  - transmet l'ordre à ses subordonnées et attend leur confirmation de validation
  - confirme à son tour la validation à son supérieur

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Incidents au court de l'action atomique

✓ Certains incidents survenants durant l'action atomique peuvent remettre en cause la propriété d'atomicité de l'action

- le cas le plus fréquent est l'absence de réponse d'un supérieur après qu'une offre de validation lui ait été faite
  - le subordonné ne reste pas indéfiniment dans la période de doute, il va prendre une décision heuristique conduisant à la libération des données liées
  - si cette décision est en contradiction avec celle du maître, elle conduira à une situation mixte qui devra être corrigée par des actions de compensation
- une application utilisant le CCRSE doit fixer les règles concernant la prise de ces décisions heuristiques et définir des actions de compensations spécifiques

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Incidents au court de l'action atomique

- ✓ Les autres incidents (défaillance d'une communication ou d'une application) sont simples à résoudre par le mécanisme de retour en arrière
- ✓ Ce mécanisme définit quel est l'utilisateur qui est chargé des reprises
  - avant l'offre de validation : l'utilisateur en cas d'incident doit être capable de remettre les données liées dans l'état initial
  - pendant la période de doute : le subordonné qui a offert la validation est responsable des reprises sur la branche concernée
  - pendant la phase 2, le supérieur qui a ordonné la validation est responsable des reprises sur la branche concernée

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrence et Recouvrement (CCRSE)

- ✓ Les primitives de CCRSE sont regroupées en 4 classes
  - les primitives concernant la 1ère phase (préparation)
  - les primitives concernant la 2e phase (validation)
  - les primitives de retour en arrière
  - les primitives de reprise
- ✓ Les paramètres suivants sont obligatoires pour CCRSE (utilisés pour créer des ID uniques)
  - titre de l'AP appelant
  - qualificateur de l'AE appelante
  - titre de l'AP en réponse
  - qualificateur de l'AE en réponse
- ✓ Par ailleurs, chaque branche d'action atomique
  - doit utiliser une association spécifique
  - est identifiée d'une manière non ambiguë (le nom du supérieur, un suffixe unique)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrency et Recouvrement (CCRSE)

- ✓ CCRSE utilise les services
  - transfert de données de Présentation
  - transfert de données typées de Présentation
  - pose de points de synchronisation majeure
  - pose de points de synchronisation mineure
  - resynchronisation
- ✓ CCRSE a besoin de certaines unités fonctionnelles
  - transfert de données typées
  - synchronisation majeure
  - synchronisation mineure
  - resynchronisation

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrency et Recouvrement (CCRSE)

- ✓ CCRSE fournit 6 services
  - Primitives relatives à la 1ère phase
    - création d'une branche d'action atomique avec un subordonné : service confirmé ou non comportant 4 primitives (C-BEGIN.xxx)
    - sollicitation de l'offre de validation par le supérieur : service non confirmé comportant 2 primitives (C-PREPARE.xxx)
    - offre de validation par le subordonné : service non confirmé comportant 2 primitives (C-READY.xxx)

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrency et Recouvrement (CCRSE)

✓ CCRSE fournit 6 services

#### ➤ Primitives relatives à la 2ième phase

- service confirmé comportant 4 primitives (C-COMMIT.xxx)
- lorsque le subordonné reçoit la primitive (indication) , il libère les données liées ainsi que les autres ressources utilisées pour l'action atomique
- lorsque le supérieur reçoit la réponse, il n'est plus responsable de la reprise et termine cette branche

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Élément de Service de Validation, Concurrency et Recouvrement (CCRSE)

▪ CCRSE fournit 6 services

✓ Primitives de retour en arrière

#### ➤ service confirmé comportant 4 primitives (C-ROLLBACK.xxx)

#### ➤ service utilisé par le supérieur ou le subordonné

- permet au supérieur d'annuler une action atomique tant qu'il n'a pas demandé la validation
- permet au subordonné de refuser une action atomique tant qu'il n'en a pas offert la validation

#### ➤ l'utilisation de ce service provoque une resynchronisation de la connexion de Session au niveau du début de l'action atomique

## IV. La couche Application

### ➤ Les ASE de base

#### ▪ Éléments de Service de Validation, Concurrence et Recouvrement (CCRSE)

✓ CCRSE fournit 6 services

#### ➤ Primitives de reprise

- service confirmé ou non confirmé comportant 4 primitives (C-RECOVER.xxx)
- quand il est utilisé par le supérieur, le service doit être confirmé
- quand il est utilisé par le subordonné, le supérieur peut confirmer la reprise ou reprendre lui même la procédure de reprise

## Sommaire

- I. Terminologie OSI
- II. Rôle des couches hautes
- III. La couche Session
- IV. La couche Application
- V. **Les applications normalisées**

## V. Les applications normalisées

- Certains types d'applications sont normalisés
  - Les fichiers virtuels (FTAM)
  - La soumission de travaux à distance (JTM)
  - L'accès aux bases de données distantes (RDA)
  - Les terminaux virtuels (VT)
  - La gestion des réseaux
- Les logiciels conformes à ces normes peuvent se présenter
  - soit sous forme d'applications toute faites
    - ✓ émulation de terminal
    - ✓ transfert de fichiers (FTAM)
    - ✓ messagerie X400, annuaire X500
  - soit sous forme d'API
    - ✓ accès à des fichiers distants (FTAM)
    - ✓ accès à des bases de données distantes
    - ✓ accès à l'annuaire X.500
    - ✓ gestion de réseaux ...

## V. Les applications normalisées

- **File Transfer Access and Management (FTAM)**
  - L'objet de la norme définissant FTAM est de permettre une gestion de fichiers indépendante des systèmes d'exploitation
  - FTAM permet le transfert de fichiers mais assure également l'accès en lecture et en écriture à des fichiers distants
  - FTAM a un fonctionnement dissymétrique
    - ✓ l'utilisateur du service FTAM (initiateur) crée une association avec un processus distant
    - ✓ ce processus distant gère l'accès au système de fichiers conformément au protocole FTAM
    - ✓ l'utilisateur local accède aux fichiers locaux en utilisant directement les primitives du système d'exploitation

## V. Les applications normalisées

### ➤ File Transfer Access and Management (FTAM)

- FTAM utilise
  - ✓ l'élément de service ACSE pour établir une association avec le système distant
  - ✓ puis, les services de la couche Présentation
- Le protocole FTAM se base sur le concept du **système de fichiers virtuel**
  - ✓ modèle décrivant les fichiers et leurs attributs sous une forme utilisable par FTAM
  - ✓ le système de fichiers virtuel comprend un nombre arbitraire de fichiers
    - zéro
    - un fichier unique
    - une arborescence de fichiers dont la racine est un catalogue

## V. Les applications normalisées

### ➤ File Transfer Access and Management (FTAM)

- **Le fonctionnement de FTAM**
  - ✓ Le traitement du fichier virtuel est orienté connexion et passe par plusieurs phases imbriquées, chacune d'elle permettant un certain nombre d'opérations
  - ✓ Les différentes phases (9) sont les suivantes
    - phase d'initialisation : elle correspond à l'établissement de l'association (négociation de paramètres, transfert d'informations pour le contrôle d'accès, ...)
    - phase de gestion du système de fichiers : elle permet l'accès à un catalogue et la réalisation des opérations de copie, déplacement, liste d'attributs, sélection, suppression et modification d'attributs
    - phase de sélection : elle correspond au choix d'un objet existant ou à la création d'un nouvel objet (fichier, catalogue ou référence) auquel s'appliqueront les opérations



## V. Les applications normalisées

### ➤ File Transfer Access and Management (FTAM)

#### ▪ Le fonctionnement de FTAM

- ✓ Les différentes phases (9) sont les suivantes
  - phase de gestion : elle permet la lecture ou la modification des attributs d'un objet ainsi que la copie ou le déplacement d'un objet
  - phase d'ouverture de fichier : elle met en place les éléments nécessaires au transfert des données
  - phase d'accès aux données : cette phase gère l'accès aux données et comprend 2 types d'opérations
    - des opérations de localisation de nœuds
    - des opérations de transfert de données durant lesquelles les données sont transférées sous forme de FADU (File Access Data Unit)
  - phase de fermeture du fichier : elle termine le processus associé à l'ouverture de fichier
  - phase de désélection de l'objet : elle termine le traitement de l'objet
  - phase de terminaison : elle termine l'association établie

## V. Les applications normalisées

### ➤ La messagerie électronique – X400

- Le système de messagerie *Message Handling System* (MHS) assure un service de courrier électronique qui vise à offrir aux utilisateurs les mêmes facilités que le courrier postal ordinaire, mais avec une transmission quasi instantanée des lettres
- Les systèmes de messagerie fonctionnent selon 5 phases principales
  - ✓ la **composition** : elle permet de construire les messages et les réponses
  - ✓ le **transfert** : il désigne la phase d'acheminement de l'information de l'émetteur vers le récepteur
    - utilisation des services de l'ACSE pour établir/fermer une connexion
    - utilisation des services de la couche Présentation pour envoyer le message
  - ✓ la phase d'**information** : elle permet de dire à l'émetteur ce qu'il est advenu de son message
  - ✓ la **conversion** de données : elle peut être nécessaire pour un affichage correct des informations sur le terminal ou l'imprimante du destinataire
  - ✓ la **remise** : c'est la dernière étape qui permet au destinataire de traiter le message reçu

## V. Les applications normalisées

### ➤ La messagerie électronique – X400

- Les systèmes de messagerie offrent souvent quelques autres facilités
  - ✓ faire suivre le courrier ou envoyer automatiquement une réponse à tous les émetteurs (i.e. indiquer une date de retour)
  - ✓ gérer des boîtes aux lettres pour stocker les messages lors de leur arrivée
    - ajouter ou détruire une boîte aux lettres
    - insérer ou extraire des données d'une boîte
    - consulter son contenu
  - ✓ gérer des listes de diffusion qui sont des listes d'adresses de messagerie
    - lorsqu'un message est envoyé à une telle liste, une copie est délivrée à chaque correspondant
  - ✓ fournir un service de courrier recommandé (notification de remise ou non), un service de courrier urgent, de messages cryptés, de destinataires secondaires...

## V. Les applications normalisées

### ➤ La messagerie électronique – X400

- On peut distinguer 3 types de messages
  - ✓ les **messages utilisateurs** : ils contiennent les informations échangées entre abonnés au service
  - ✓ les **réponses** : ce sont des messages générés automatiquement par le système pour indiquer à un émetteur si son message est correctement arrivé ou non
  - ✓ les **essais** : ce sont des messages de test constitués d'enveloppes vides, leur rôle consiste à voir si le destinataire est accessible ou pas
    - grâce aux informations collectées, il est possible de reconstituer le chemin emprunté et le temps de traitement utilisé par chaque nœud intermédiaire

## V. Les applications normalisées

### ➤ Le service d'annuaire électronique – X500

- Cette norme propose une méthode pratique de représentation et d'organisation des informations
  - ✓ elle permet de manipuler les informations à l'aide de leurs identifiants (i.e. nom machine) au lieu de leurs valeurs (i.e. adresse machine)
- Ce service nécessite
  - ✓ une gestion centralisée des informations facilitant les opérations de mise à jour
  - ✓ un système d'interrogation souple dans le choix des critères et dans la nature des comparaisons (rechercher des sous-chaînes, orthographe proche...)
- L'ensemble des informations sur les objets contenu dans l'annuaire forme la base d'information de l'annuaire **Directory Information Base (DIB)**
  - ✓ l'annuaire se présente comme une base de donnée hiérarchique dont le gestion est répartie sur différents systèmes
  - ✓ chaque entrée de cette base est un ensemble d'attributs définis par
    - identificateur : nature de l'attribut
    - une ou plusieurs valeurs affectées à cet attribut

## V. Les applications normalisées

### ➤ Le service d'annuaire électronique – X500

- Le domaine couvert par l'annuaire est une réunion de sous-arbres disjoints dont la gestion est déléguée à divers pays ou organisme
  - ✓ chacun de ces sous-arbres est une *aire administrative*
  - ✓ la gestion d'une aire administrative comporte 3 aspects
    - la définition des règles liées à la nature des données (des entrées du type pays, organisation, personne)
    - la sécurité (protection des informations de l'annuaire contre les opérations non autorisées)
    - la définition d'attributs communs à un ensemble d'entrées (attributs collectifs)
  - ✓ un administrateur peut découper l'aire dont il a la charge en plusieurs sous-ensembles disjoints
    - *aires administratives spécifiques*

## V. Les applications normalisées

### ➤ Le service d'annuaire électronique – X500

- ✓ une aire administrative spécifique peut regrouper des entrées ayant des caractéristiques communes
  - ce regroupement est appelé *aire administrative interne*
  - ces aires peuvent se recouvrir et une entrée peut appartenir à un nombre quelconque d'aire administrative interne
- ✓ une aire administrative appartient à un même sous-arbre défini par sa racine qui est un point administratif
- ✓ on distingue 3 types de points administratifs, associés aux 3 types d'aires
  - point administratif autonome (AAP)
  - point administratif spécifique (SAP)
  - point administratif interne (IAP)

## V. Les applications normalisées

### ➤ Le service d'annuaire électronique – X500

#### ▪ Organisation de la structure hiérarchique

- ✓ Les entrées de l'annuaire constituent un arbre
  - **Directory Information Tree** (DIT) permet de représenter les relations de dépendance existant entre les objets réels associés aux entrées de l'annuaire
  - chaque sommet de l'arbre contient les informations décrivant un objet particulier
    - la racine de l'arbre (le point d'entrée à l'arbre) est un nœud particulier qui ne décrit aucun objet
    - chaque nœud intermédiaire de l'arbre est la racine d'un sous-arbre qui représente une entité du type pays, société, service, etc...
- ✓ La base de données peut contenir 3 sortes d'entrées
  - des entrées associées à des objets réels
  - des entrées correspondant à des alias
  - des entrées représentants des points administratifs