

SERVICES COMMUNS D'APPLICATION

ACSE - RTSE

ROSE - CCR

ACSE

 Association Control Service Element **Service utilisé par PRESQUE TOUTES** les entités d'application Sauf messagerie X400 : mode sans connexion Permet d'établir les associations entre ENTITES d'application
de rompre ces associations Identification des ASE qui constituent l'entité options retenues informations pour obtenir l'interopérabilité des applications à associer

Ces informations constituent un **CONTEXTE d'APPLICATION**

👉 Standards OSI 8649 (Service) et 8650 (Protocole)

👉 Permet à TOUS les SERVICES SPECIFIQUES (ou autres services communs) d'**ETABLIR** et de **ROMPRE** les **ASSOCIATIONS** entre 2 Processus d'Application

👉 4 Services :

👉 A-ASSOCIATE

établissement d'association

Permet de définir le contexte d'association

👉 A-RELEASE

relachement (rupture) d'association

👉 A-ABORT

Rupture anormale d'association
(sur défaut) par une entité d'application

👉 A-P-ABORT

Rupture d'association par le service inférieur

👉 Les 2 premiers services : A-ASSOCIATE et A-RELEASE sont confirmés.

 **Service PRESENTATION DE BASE** P-CONNECT P-RELEASE P-ABORT L'ASSOCIATION d'application est GROUPEE
avec la CONNEXION de Présentation Il en est de même pour la RUPTURE Si on dispose d'UN SEUL CONTEXTE de PRESENTATION
Il inclura de SYNTAXE de TRANSFERT de ACSE
Sinon Fonctionnalité CONTEXTES MULTIPLES est requise Pour rompre l'association il faut le JETON de SESSION correspondant

 Ils sont au nombre de 5

 AARQ	Etablissement
 AARE	Réponse à AARQ
 RLRO	Rupture
 RLRE	Réponse à RLRO
 ABRT	pour rupture anormale

 Ils sont directement liés au primitives de service

 A-ASSOCIATE	AARQ, AARE
 A-RELEASE	RLRO, RLRE
 A-ABORT	ABRT

👉 Structure et paramètres de AARQ

👉	AARQapdu ::=	[APPLICATION 0]	IMPLICIT SEQUENCE	
👉	protocolVersion	[0]	IMPLICIT BIT STRING	{Version1 (0)}
👉	calledAETitle	[1]	ApplicationTitle	OPTIONAL,
👉	callingAETitle	[2]	ApplicationTitle	OPTIONAL,
👉	applicationContextName	[3]	ApplicationContextName	
👉	userInformation	[4]	AssociationData	OPTIONAL
👉	ApplicationTitle ::=		OBJECT IDENTIFIER	
👉	ApplicationContextName ::=		OBJECT IDENTIFIER	
👉	AssociationData ::=		EXTERNAL	

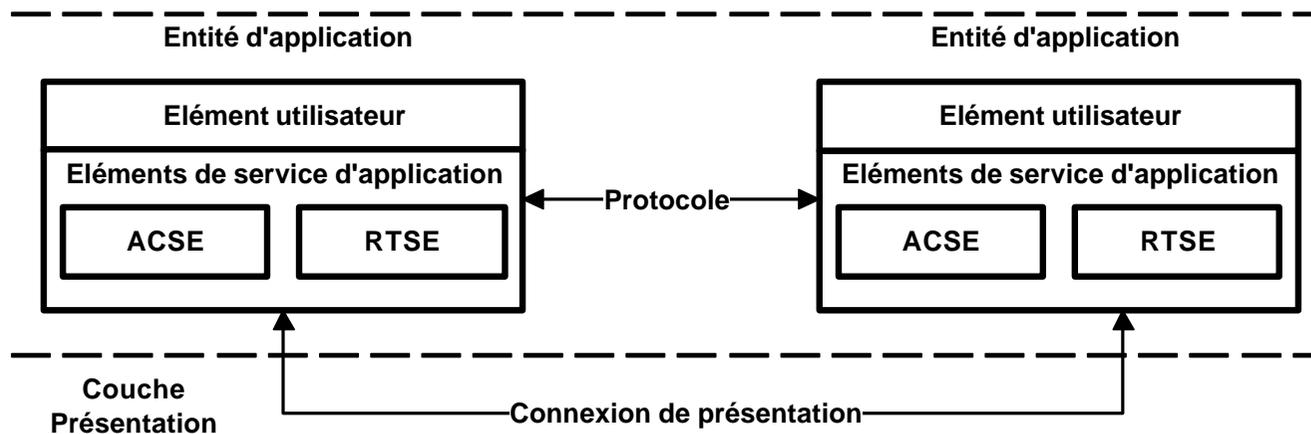
RTSE

Service de transfert fiable

- Reliable Transfert Service Element
- Standard OSI : 9066 Standards ITU (CCITT) X.218 et X.228
- Origine : Messagerie X.400 mais utilisables par d'autres services (ex. FTAM)

Transfert de données entre entités d'application avec **reprise en cas de défaillance**

- Pas de garantie : anomalie peut ne être surmontée ...



 **2 modes : normal et X410-1984**

 Dans ce cas : restrictions - Interfonctionnement avec anciennes applications

 **3 sous-ensembles:**

 **Établissement et Rupture d'association**

»»» **s'appuie sur ACSE**

»»» **Établissement d'association**

»»» **Fermeture d'association**

 terminaison négociée

»»» **Coupure d'association (par utilisateur ou fournisseur)**

 rupture brutale non confirmée

 **Transfert de données**

»»» **2 modes : « monologue » ou « bidirectionnel à l'alternat »**

 **Changement de Tour**

»»» **Demande et Cession de « Tour » : droit d'envoyer des données**

»»» **basé sur Jetons de Session**

Etablissement

-  Service confirmé
-  permet d'établir le mode de fonctionnement et le mode de transfert
-  permet de négocier les points de synchronisation et la taille de la fenêtre

Fermeture

-  Négociée donc service confirmé
-  Demande possible seulement si on bénéficie du « Tour »
 -  Voir Jeton de Terminaison dans la Session

Coupure

-  En cas de défaillance
-  Service simple (non confirmé)
 -  demandé par le Fournisseur (RTSE) ou l'utilisateur de RTSE
-  généralement suivi d'une Reprise
 -  Voir Reprise d'Activité de la Session

Service avec confirmation (automatique) engendrée par le fournisseur (RTSE)

-  pas de primitive Réponse
-  Confirmation Positive si message a pu être délivré dans le temps spécifié
 - »»» sinon confirmation négative**
-  La requête transfert contient un paramètre « temps de transfert »
 -  laps de temps pendant lequel le fournisseur doit transférer avec succès la pDU à l'autre utilisateur
-  Conditionné par la possession du « Tour »
-  la confirmation contient un paramètre « résultat »
 - »»» exceptionnellement ce résultat peut être négatif alors que les données ont été bien transférées ...**

Basé sur le Transfert de Jeton

 Service simple (non confirmé)

Demande de changement de Tour

 Transfert la priorité de l'action à entreprendre à l'autre utilisateur

Cession de Tour

 peut être faite n 'importe quand

 Pas forcément après une demande...

» responsabilité reste à celui qui déteint le Tour

Tour initial défini lors de l'établissement

Assez complexe

-  malgré faible nombre de primitives et de PDU
-  Problèmes posés par les reprises
 -  peuvent intervenir n 'importe quand

6 PDU

-  RTAB : Rupture (brutale) d'association
 -  RTORQ : ouverture d'association
 -  RTOAC : Acceptation d'ouverture
 -  RTORJ : refus d'ouverture
 -  RTTR : transfert de données
 -  RTTP : demande de Tour (pas pour cession..)
-  Pas de PDU pour terminaison normale ou Cession de Tour

Sauf exception : paramètres optionnels

 seul paramètre obligatoire : données utilisateur dans RTTR

RTORQ

 taille du point de reprise, taille de fenêtre, mode de dialogue, identificateur de Connexion de Session (sur reprise), données utilisateur (Établissement)

 mode de fonctionnement SI X410-1984

RTOAC

 taille du point de reprise, taille de fenêtre, identificateur de Connexion de Session (sur reprise), données utilisateur (Établissement)

RTORJ

 Raison et données utilisateur (Si mode X410)

RTTP

 Priorité

RTAB

 Raison, paramètre renvoyé, données utilisateur (explications ...)

ROSE

Établissement

-  Réalisé grâce à ACSE
-  Premier filtre par ACSE. Si Indication venant de ACSE examen des paramètres
-  Alors Acceptation ou Refus

Transfert

-  En cas d'interruption : Reprise d'Activité au niveau Présentation

Changement de Tour

-  Demande projetée sur Demande de Jeton de Présentation
-  Cession indiquée grâce à Indication de Passation de Commande de Présentation

Fermeture

-  Réalisée par ACSE

Rupture

-  Cas d'erreur grave où REPRISE impossible

 **Service d'Opérations distantes**

 Remote Operation Service Element

 **Permet à une entité d'application de demander le traitement à distance d'une opération définie**

 **Base d'un système client-serveur**

 **Créé pour messagerie interpersonnelle (X400)**

Support du service RPC de l'ISO

☞ Une opération est TOTALE si elle inclu TOUS LES CAS POSSIBLES de

RESULTATS
ERREURS

☞ Si toutes les opérations d'un type sont totales ce type est ROBUSTE

☞ 3 classes :



EXECUTION UNE FOIS EXACTEMENT

garantie la plus forte



EXECUTION AU MOINS UNE FOIS

Opérations "idempotent"

par exemple LECTURE



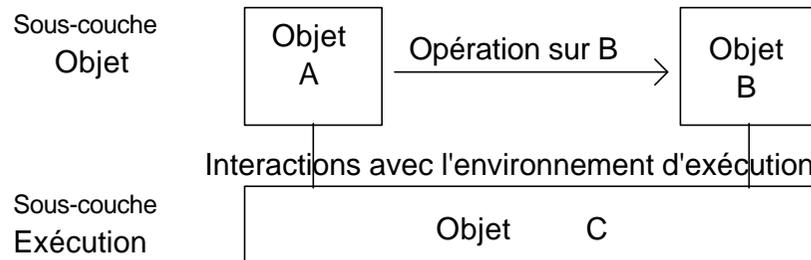
EXECUTION AU PLUS UNE FOIS

👉 Opération sur un objet est réalisée dans un ENVIRONNEMENT d'EXECUTION

👉 Celui-ci peut être modélisé par un OBJET INTERMEDIAIRE médiateur

👉 on peut utiliser

- 👉 opération distante basée sur un transfert en file
- 👉 un appel de procédure locale ou distante (synchrone)
- 👉 un passage de message asynchrone (retour immédiat)



👉 Nommage :

- 👉 NOM d'OPERATION
- 👉 NUMERO d'OPERATION

☞ **RELATION** entre les **INSTANCES** d'OBJETS
qui lie les objets entre eux

☞ Cette **LIAISON** (**BINDING**) peut

☞ n'exister que durant l'exécution de l'opération

☞ persister pour des opérations multiples

☞ Si la liaison subsiste : **CONTEXTE**

☞ on échange des "**TITRES**" (handle =
poignée qui référencent les opérations déjà
identifiées

☞ Une demande d'opération peut provoquer
d'autres demandes

☞ On dit que ces opérations sont **RELIEES** (linked)

— Opération parent →

← Opération enfant

← Opération enfant

👉 BIND pour établir les liaisons entre objet demandeur et exécutant

👉 UNBIND



Ces macros utilisent ACSE ou RTSE

👉 ERROR

👉 OPERATION pour définir un ensemble d'opérations exécutables

Ces macros utilisent le service ROSE ci-dessous

👉 Pour demander une opération une ASE utilise la primitive

RO-INVOKE

👉 Le "résultat" est fournit par

RO-RESULT ou RO-ERROR

👉 En cas d'anomalie

RO-REJECT-U ou RO-REJECT-P

CCR

Engagement, Concurrence et Reprise

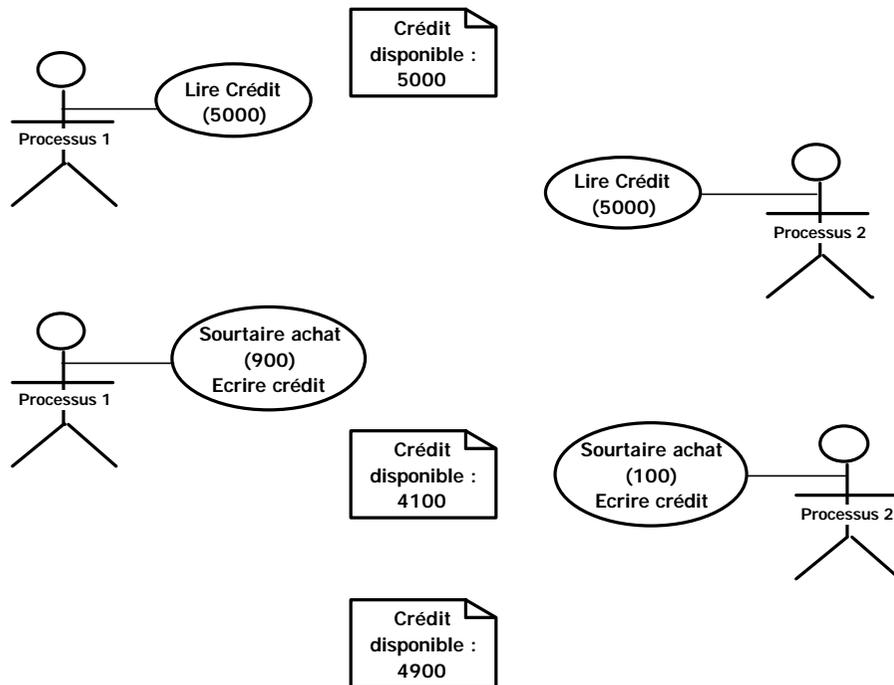
-  Commitment, Concurrency and Recovery
-  Validation, Annulation et Reprise
-  Engagement, Concomitance et rétablissement
-  Standards OSI : IS 9804 Standard ITU (CCITT) : X861, X851, X852

Défi posés par applications réparties

-  Possibilité de conflit pour des accès concurrents à une ressource
-  Possibilité d'échec partiel si plusieurs processeurs impliqués : incohérences
 -  Certains continuent à travailler quand d'autres sont en défaut
-  Peu fréquent mais possible et très gênant si cela se produit

Dispositions à prendre

-  Engagement (ou validation)
-  Concurrence (ou concomitance)
-  Reprise (ou rétablissement)



👉 Mise à jour incohérente d'un compte bancaire suite à 2 retraits quasi-simultanés.

👉 Seul le retrait qui fait la dernière mise à jour est pris en compte pour le crédit disponible

👉 Autre difficulté

👉 rupture d'activité et reprise par un processeur

👉 Concept central

👉 **ACTION ATOMIQUE**

- 👉 Action atomique exécutée par processus transactionnel
- 👉 Transaction initiée par un client et exécutée par un serveur
- 👉 Définition : Transaction = opération qui respecte les propriétés :
 - 👉 A : Atomicité
 - » opérations toutes exécutées ou aucune
 - 👉 C : Consistance
 - » Opérations correctes et toutes validées
 - » Passage d 'un état consistant à un autre
 - 👉 I : Isolation
 - » pas de mélange de résultats partiels entre transactions différentes
 - » sérialisation si données communes
 - 👉 D : Durabilité
 - » Résultats de transaction terminée ne peuvent être altérés même en cas de panne - Synchronisation et sauvegarde des données décrivant la transaction

2 phases

Phase 1 :

-  serveur reçoit une description de l'action atomique et prend arrangements
-  Serveur peut exécuter la transaction mais aussi revenir à état précédent
-  Serveur peut enregistrer l'action désirée en attendant instruction finale pour la réaliser (log)

Phase 2

-  Serveur « engage » (commit) l'exécution si cela est possible
 -  sinon avorte la transaction et rend compte au client
 -  **Quand serveur engagé il DOIT terminer les opérations quelques soient les problèmes rencontrés**
 -  si panne reprise de la transaction au redémarrage
 -  Pour cela transaction doit être enregistrée au préalable (journalisation, log)
-  Si transaction engagée ni serveur ni client ne peuvent la faire avorter

 3 sous-ensembles Description/Représentation du service

 C-Begin : Donne la liste des opérations à effectuer par le serveur

»»» confirmé ou non

 C-Prepare

»»» si liste pas clairement indiquée

»»» fin de définition de transaction et demande d'engagement

 Réponse à une demande de Transaction

 C-Ready ou C-Refuse : Acceptation ou Refus

»»» Acceptation : possibilité d'effectuer la transaction ou de revenir à l'état précédent

 Compléter ou Annuler la Transaction

 C-Commit ou C-Rollback : services (confirmés) pour engagement ou annulation

 C-Restart : pour tenir compte de différents cas d'incidents

 Protocole simple Permet de convoier les informations de COORDINATION entre un client et un serveur Repose sur le service Présentation pour définir un contexte de transfert de données Repose sur des fonctions de Session (à travers Présentation) Points de synchronisation Données typées (C-Prepare et C-Ready) pas besoin du jeton de données ... Toutes les actions sont traitées localement dans les processus client et serveur

Service d'Annuaire : X500

■ if ■ G.Beuchot 361

Services communs d'Application : ACSE - RTSE - ROSE - CCR

☞ Service d'annuaire

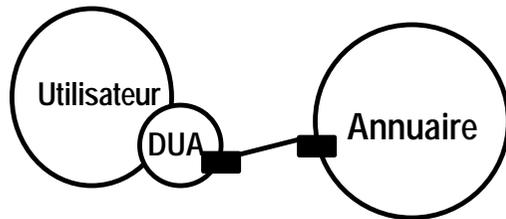
- ☞ Directory
- ☞ Standard ITU (CCITT) : série X500

☞ Associer une adresse à un **nom de ressource ou d'utilisateur**

- ☞ Application de base de données avec des contraintes de mise à jour faibles

☞ Maintenir une base de données logique portant sur un ensemble d'objets du monde réel.

- ☞ DUA : Directory User Agent
 - » représente chaque utilisateur ou ressource
 - » Point d'accès à l'annuaire
- ☞ DIB : Directory Information Base
 - » Information dans la base



 4 sous-ensembles Service de qualification

-  requêtes pour imposer des limites à l 'usage des ressources
-  paramètres de sécurité : service d'authentification forte
-  Requêtes peuvent être appliquées sur ces paramètres grâce à des filtres.

 Service d 'interrogation

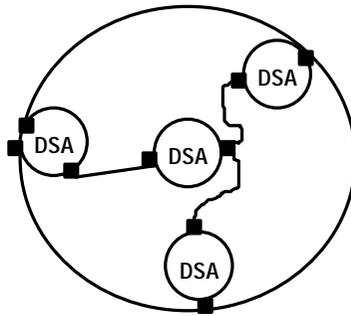
-  fonctions de lecture en entée, de comparaison d 'attribut, de lisatge de subordonnés d 'une entrée, de recherche d 'information par filtre
-  Exemple : ressemble à pages jaunes

 Service de modification

-  ajouter, retrancher modifier une entrée
 -  en particulier identificateur

 Autres fonctions

-  Traitement d 'erreur - renvoyer une référence (referral) - chaîner une requête



Base de données (DIB) répartie

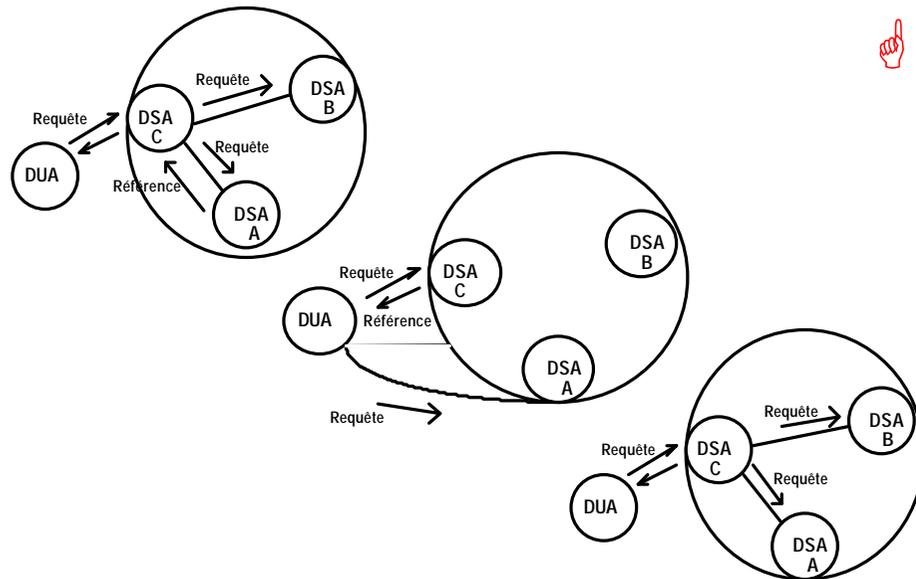
sur systèmes interconnectés

Bases locales

dépendent de implantation

accessible par agent : DSA

Directory System Agent



DSA peut

utiliser la base locale

interagir avec d'autres DSA

renvoi de référence (referral)

Chaînage

diffuser des requêtes