CONSERVATOIRE
NATIONAL
DES ARTS
ET METIERS

Systèmes Distribués :

Introduction à DCE

Avril 2000

René J. Chevance

Contenu

- Pourquoi un système distribué?
- Besoins en matière de système distribué
- Caractéristiques communes des systèmes distribués
- Composants d ’une architecture générique de système distribué
- Historique de DCE
- Communication entre programmes
  - Appel de procédure à distance (Remote Procedure Call - RPC)
  - Middleware orienté message (Message Oriented Middleware - MOM)
- Kerberos

Cet exposé est fondé sur les ouvrages suivants :

- Understanding DCE
- A Guide to Writing DCE Applications

Pourquoi un système distribué?

- **Faciliter la communication des applications tout en maintenant l'autonomie locale**
- **Améliorer la disponibilité**
  - **Redondance des composants et des informations**
  - **Détection des défaillances, reconfiguration et reprises**
- **Augmenter l'efficacité:**
  - **Parallélisme**
  - **Serveurs spécialisés**
- **Partage de ressources (statique et dynamique)**
- **Faciliter l'évolution**
- **Adaptation de l'architecture technique aux besoins du système d 'information de l'entreprise**
- **« Universalité » de l'interface homme/machine**

Caractéristiques des systèmes distribués

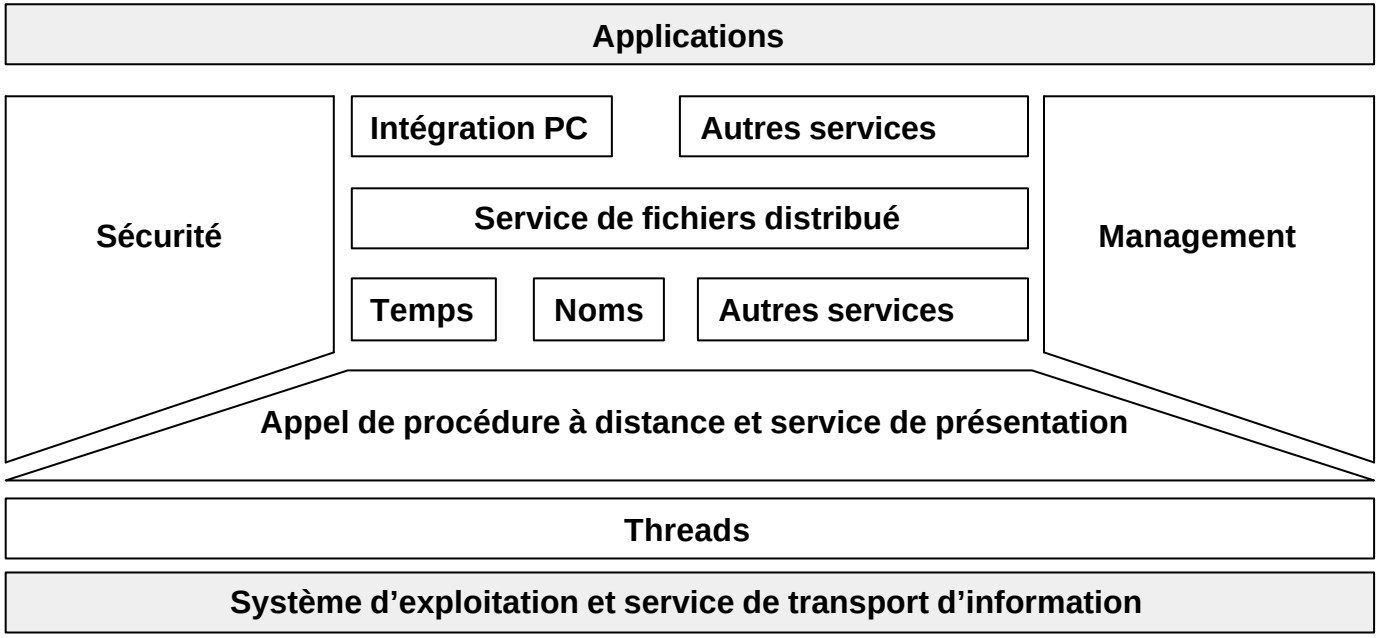
- **Contrôle décentralisé (absence de hiérarchie)**
- **Absence de mémoire partagée**
  - **Impossibilité de reconstituer (dans le cas le plus général) un état global du système**
- **Pas de référence temporelle commune**
- **Exploitation du parallélisme (communication et synchronisation)**
- **Potentiel de haute disponibilité mais l 'augmentation du nombre de constituants augmente le risque de défaillance**
  - **Mécanismes de détection des erreurs et de reprise**
- **Vulnérabilité du système**
- **Potentiel de performance mais le coût des communications et de la synchronisation peut l'annuler**

Besoins en matière de système distribué

- **Service de désignation ou annuaire (Directory Services)**
- **Sécurité**
- **Service de fichiers distribué**
  - **NFS - Network File System (d'origine Sun)**
  - **DFS - Distributed File System (OSF/DCE d 'origine Carnegie Mellon)**
- **Service d'appel à distance (communication entre composants d'une application distribuée)**
  - **RPC - Appel de procédure à distance (Remote Procedure Call)**
  - **RMI - Remote Method Invocation**
- **Service de temps**
- **Service transactionnel**
- **Gestion de système**
- **Support des stations de travail**

Composants d'une architecture générique

- **Modèle DCE (Distributed Computing Environment) de l'OSF (Open Software Foundation)**

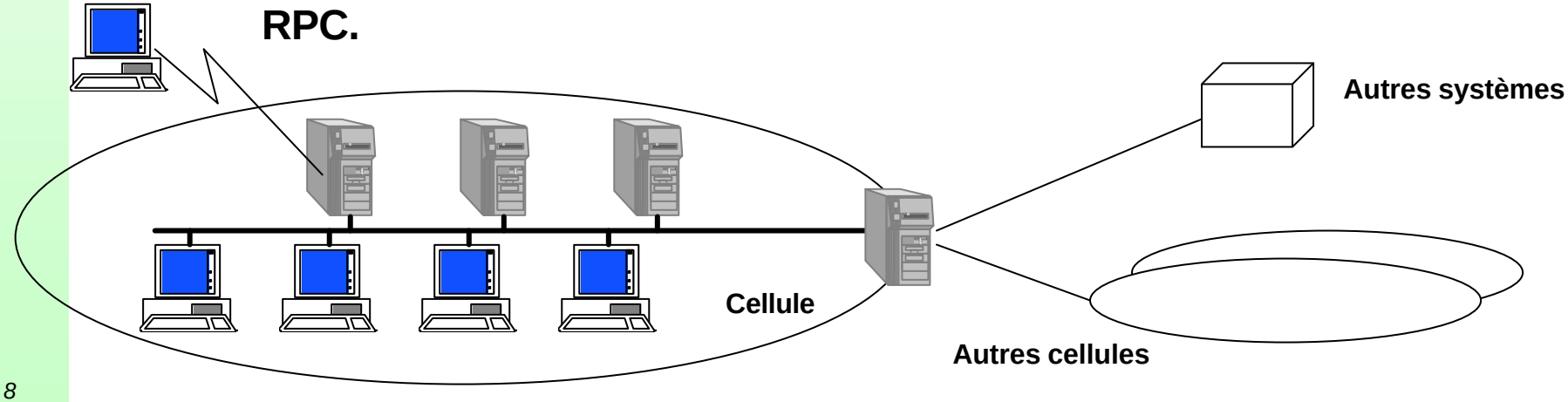


Historique de DCE

- **OSF (Open Software Foundation) a été créée à la fin des années 1980 par un groupement de constructeurs de systèmes (dont Bull, DEC, HP, IBM, …) avec l 'objectif de développer un système UNIX indépendamment d'AT&T (qui venait de prendre une participation dans Sun)**
- **OSF a, parallèlement avec le système d 'exploitation, cherché a promouvoir des standards pour la programmation des systèmes distribués :**
  - **Motif : standard de présentation « multi-fenêtres » fondé sur X-11 (MIT)**
  - **DCE pour le développement et le support des applications distribuées**
- **OSF opérait au moyen de RFT/RFP (Request for Technology/Request for Proposals)**
- **OSF appartient maintenant à l'Open/Group**

DCE : La Cellule

- **Cellule : entité organisationnelle groupant des utilisateurs ayant des centres d 'intérêt communs**
- **Cellule : pas de contrainte géographique mais souvent supportée par un réseau local (RLE ou LAN)**
- **Transparence de la distribution au sein de la cellule :**
  - **Annuaire de la cellule (Cell Directory Service)**
  - **X500 à l 'extérieur de la cellule**
- **Accès aux services :**
  - **Coopération au moyen du RPC;**
  - **Serveur de communication avec l 'extérieur accessible via RPC.**

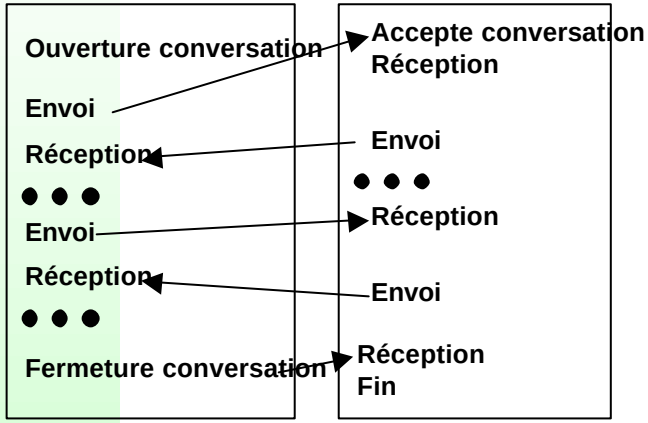


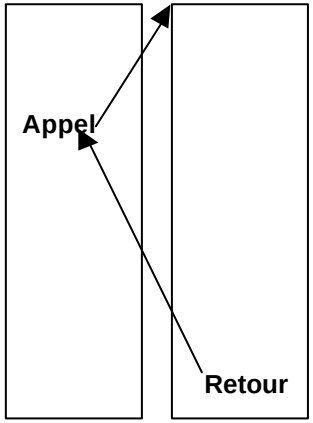
Communication entre programmes

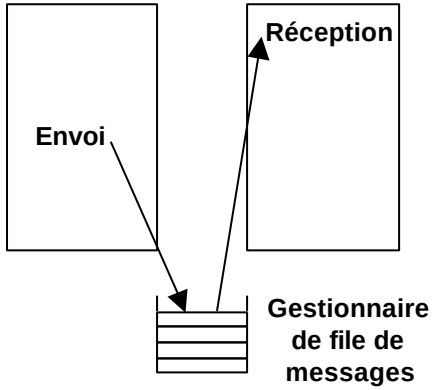
- **Conversationnel**
- **Requête/Réponse (Appel de procédure à distance - RPC)**
- **Échange de messages - Trois modes**
  - **Passage de message**
    - **Unidirectionnel**
    - **Non bloquant**
  - **File de messages (Message Queuing - MOM)**
  - **Publish-and-Suscribe**
    - **Communication de 1 vers N**

Communication entre programmes(2)

- **Illustration de quelques modes de communication entre programmes**







(1) - Conversationnel

Égal à égal

(2) - Appel de procédure
Appel de procédure à distance
(RPC Remote Procedure Call)

Synchrone

(3) - Message
Middleware Orienté Message
(MOM Message Oriented Middleware)

Asynchrone

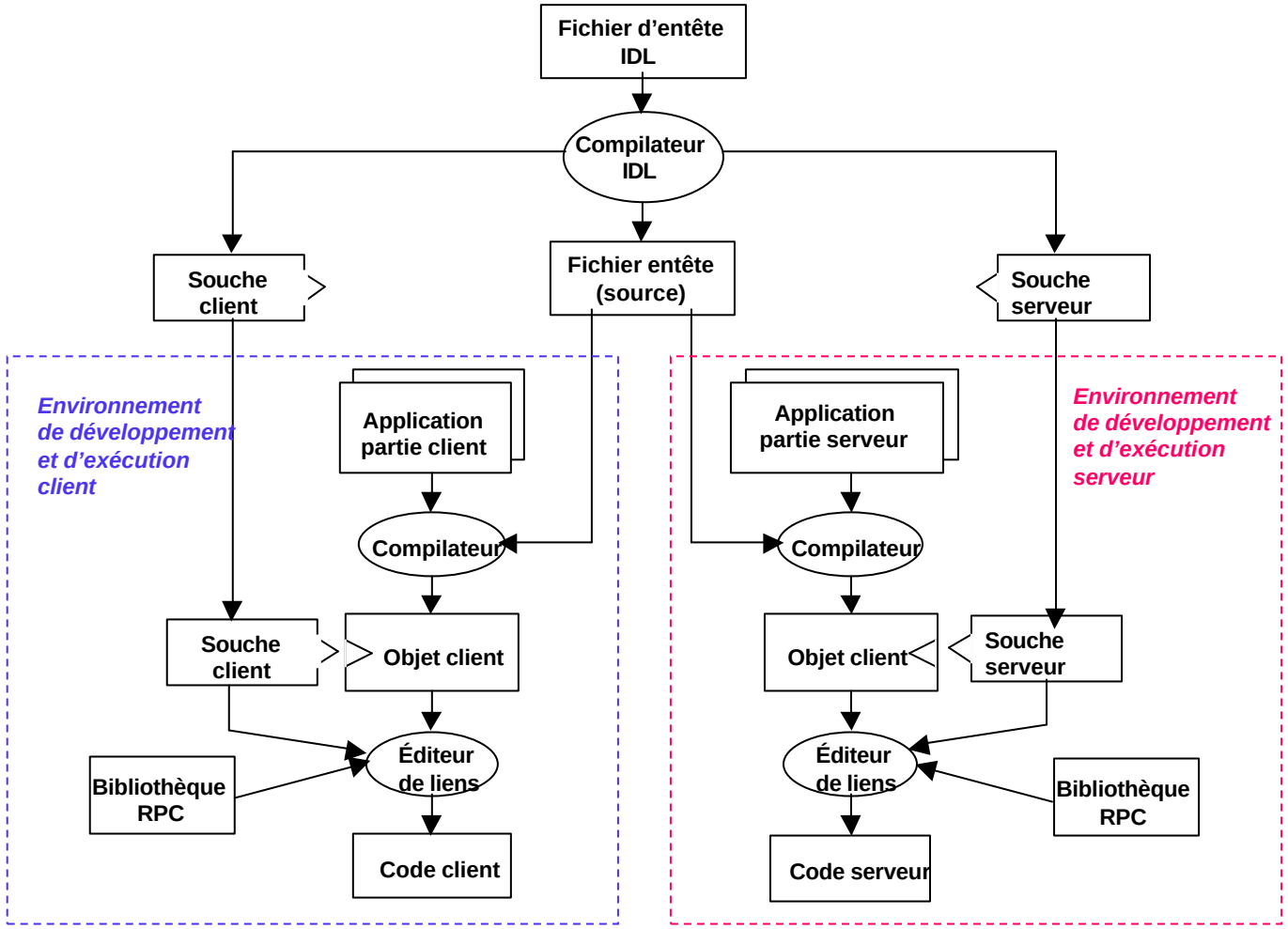
Caractéristiques comparées RPC - MOM

Caractéristique	MOM	RPC
Métaphore	**Courier**	**Téléphone (sans répondeur)**
Relation temporelle entre client et serveur	**Asynchrone**	**Synchrone**
Nature du dialogue	**File d'attente**	**Requête - Réponse**
Etat opérationnel du serveur	**Pas nécessaire**	**Obligatoire**
Equilibrage de charge	**Politique d'extraction des messages (système de priorités)**	**Au moyen d'un moniteur transactionnel**
Support des transactions	**Dépend du produit**	**Dépend du produit (nécessité d'un RPC transactionnel)**
Filtrage des messages	**Possible**	**Non**
Performances	**Lent en cas de sécurisation des messages par écriture sur disque**	**Plus efficace que MOM car pas de sauvegarde**

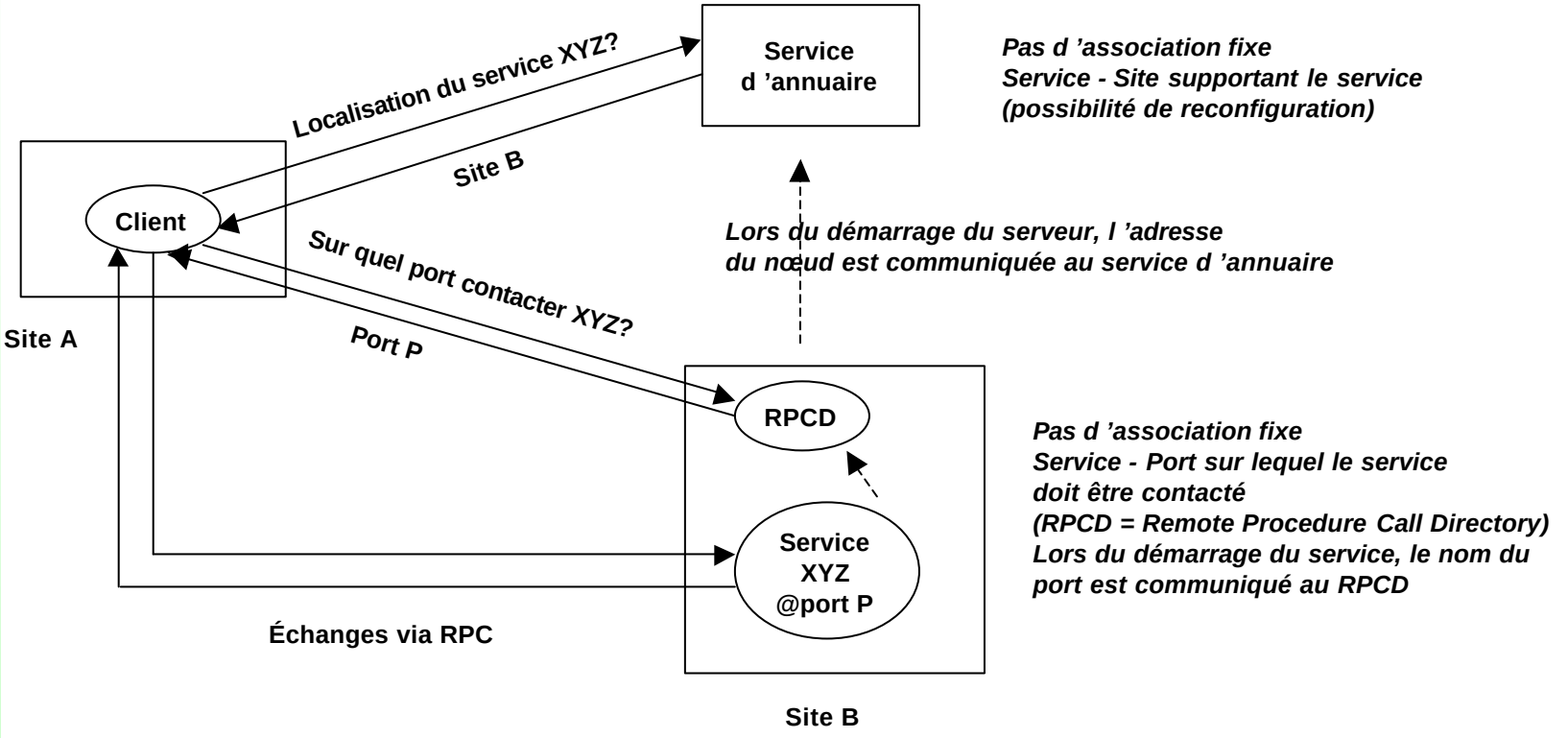
DCE : Appel de procédure à distance RPC

- **D'origine Apollo (Société reprise par HP)**
- **Se fonde sur l'appel de procédure mais autorise que la procédure appelée soit à distance (sur une autre système)**
- **Contrainte : archivage de l 'entête de procédure pour permettre de faire certaines vérifications lors de la compilation et d 'engendrer les souches**
- **Entêtes décrites en langage IDL (Interface Definition Language)**
- **Rôle des souches (Stubs) :**
  - **Client :**
    - **Encapsulation des données à transmettre**
    - **Appel de la bibliothèque RPC et attente de la réponse**
  - **Serveur**
    - **Extraction des paramètres**
    - **Appel de la procédure « serveur »**
    - **Encapsulation des résultats**
    - **Appel de la bibliothèque RPC**

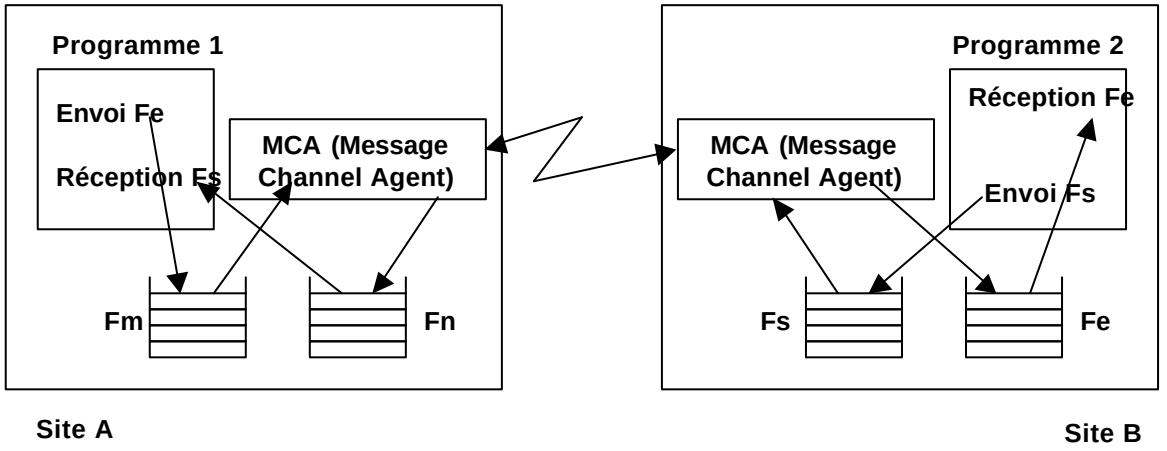
RPC - Développement d'une application



Fonctionnement du RPC



Communication au moyen de MQSeries



Communication - Cas distant

Types de files de messages :

- **Files locales**
- **Files distantes**
- **Alias**
- **Modèle**
- **Rebut**

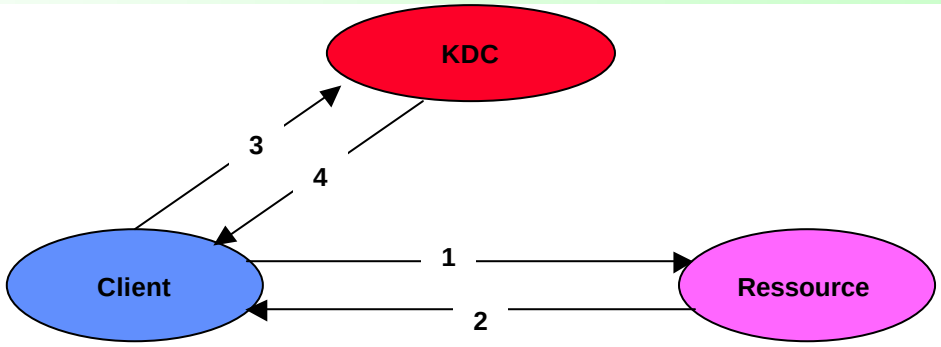
Programmation du serveur :

- **Scrutation**
- **Mécanisme de déclenchement (trigger)**

Sécurité distribuée : Kerberos

- **Dérivé du projet Athena du MIT**
- **Permet l ’authentification dans un système distribué**
- **Largement utilisé (e.g. infrastructure de sécurité de Windows 2000)**
- **Fondé sur DES (Data Encryption Standard - Clés privées) et ses extensions**
- **Kerberos se fonde sur les entités suivantes**
  - **Client (utilisateur ou application cliente)**
  - **Ressource (doit s ’assurer que son client est bien celui qu’il prétend être)**
  - **Centre de distribution des clés (KDC Key Distribution Center) entité centrale de confiance qui connaît la totalité des clés (clients et ressources)**

Kerberos : Exemple de fonctionnement



Authentification d'un client vis-à-vis d'une ressource

- **Le client, désirant un service, s 'adresse à la ressource qui lui envoie un ticket de vérification (TGT Ticket Granting Ticket)**

- **Le TGT provient du KDC, il contient notamment la clé de session, le nom de l'utilisateur et un temps de validité du ticket.**
  Le TGT est chiffré avec une clé que ne connaît pas le client (elle est partagée entre KDC et la ressource)

- **Pour son authentification, le client envoie le ticket au KDC qui déchiffre le ticket, extrait la clé de session et déchiffre**
  le nom de la ressource à l'aide de cette clé, la vérification de la somme de contrôle (checksum) permet de s'assurer
  que la même clé a été utilisée pour chiffrer et déchiffrer.

- **La date de validité du ticket permet d'éviter qu'un intrus ayant " écouté " la conversation la rejoue.**

- **Le client peut aussi demander que le serveur soit authentifié. Un scénario " symétrique " est alors mis en œuvre.**