

### ### 1. **\*\*Protocole Telnet\*\***

- **\*\*Telnet\*\*** est un protocole qui permet d'établir une connexion en ligne de commande (CLI) avec une machine distante. Il permet de **\*\*faire du déport d'affichage\*\***, c'est-à-dire que l'utilisateur peut exécuter des commandes sur un serveur distant comme s'il était physiquement devant celui-ci.
- Ce protocole a été principalement utilisé dans les années **\*\*1970\*\*** et **\*\*1980\*\***. C'était l'un des premiers moyens de communication à distance entre les ordinateurs.
- **\*\*Telnet\*\*** est cependant non sécurisé, car les informations sont envoyées en clair (non cryptées).

### ### 2. **\*\*X Window System (X11)\*\***

- Le **\*\*X Window System** (ou X11) est un protocole développé à partir de 1984, qui permet de réaliser du **\*\*déport d'affichage graphique\*\*** (ce qui inclut les éléments visuels, comme les fenêtres ou les boutons). Il est principalement utilisé dans les environnements **\*\*UNIX\*\*** et **\*\*Linux\*\***.
- Contrairement à Telnet, X11 gère des **\*\*requêtes graphiques\*\*** : le serveur (la machine puissante) envoie les informations d'affichage au client (l'utilisateur). Le client fait ainsi fonctionner l'interface graphique distante.
- Ce protocole a été utilisé par de grandes entreprises comme **\*\*IBM, BULL, HP, SUN, SGI\*\***.
- **\*\*Date clé\*\*** : Le **\*\*projet X\*\*** a débuté en **\*\*1984\*\*** et la version **\*\*X11\*\*** a été publiée en **\*\*1987\*\***.

### ### 3. **\*\*Dumb Terminals\*\***

- Les **\*\*dumb terminals\*\*** (terminaux passifs ou "muets") sont des ordinateurs très simples qui servent uniquement à afficher ce que le serveur leur envoie. Ils ne font aucun traitement en local, mais simplement du **\*\*déport d'affichage\*\*** (affichage distant).
- Ils permettent de gérer l'affichage graphique avec des interactions simples comme les clics de souris, mais toute la puissance de traitement est sur le serveur.

### ### 4. **\*\*Windows NT (WNT)\*\***

- **\*\*Windows NT\*\*** est un système d'exploitation conçu par **\*\*Microsoft\*\*** au début des années **\*\*1990\*\***. Il a été développé par une équipe qui venait de DEC (Digital Equipment Corporation), ayant travaillé sur **\*\*VMS\*\***, un ancien système d'exploitation.
- NT signifie "New Technology", et l'une de ses principales innovations a été de permettre une architecture plus **\*\*puissante et modulaire\*\***, ce qui a favorisé l'évolution vers des **\*\*PC puissants\*\*** capables de remplacer les terminaux passifs, devenant ainsi des **\*\*clients lourds\*\***.
- **\*\*Date clé\*\*** : **\*\*Windows NT 3.1\*\*** a été lancé en **\*\*1993\*\***.

### ### 5. **\*\*Client lourd vs Client léger\*\***

- **\*\*Client lourd\*\*** : Un client lourd est un ordinateur qui exécute localement toutes les applications, y compris les bases de données. C'était typiquement utilisé dans des secteurs comme les banques et les assurances, où les requêtes SQL étaient gérées localement mais les données étaient stockées sur des serveurs comme ceux d'**\*\*Oracle\*\***.
- **\*\*Oracle\*\*** est une entreprise fondée en **\*\*1977\*\***, spécialisée dans les bases de données relationnelles, et elle a dominé le marché dans les années **\*\*1980\*\*** et **\*\*1990\*\***.
- **\*\*Client léger\*\*** : Dans ce modèle, le serveur fait la plupart des traitements, et le client se contente de recevoir et d'afficher les résultats. C'est le modèle de **\*\*terminal server\*\*** utilisé par **\*\*Microsoft\*\*** pour remplacer l'approche du client lourd.

### ### 6. **\*\*Terminal Server et VMware\*\***

- **\*\*Microsoft Terminal Server\*\*** : Il permet de faire du déport d'affichage sur des PC modernes, en utilisant des flux vidéo pour envoyer les interfaces graphiques au client.
- Ce système est similaire à des technologies comme **\*\*VMware\*\***, qui permettent de faire fonctionner des machines virtuelles et de virtualiser les serveurs. Le but étant de centraliser les ressources tout en déportant l'affichage.
- **\*\*Date clé\*\*** : **\*\*VMware\*\*** a été fondée en **\*\*1998\*\*** et a lancé sa première version en **\*\*1999\*\***.

### ### Résumé des dates importantes :

- **\*\*1970s - 1980s\*\*** : Utilisation courante de **\*\*Telnet\*\***.
- **\*\*1984\*\*** : Début du projet **\*\*X Window System (X11)\*\***.
- **\*\*1987\*\*** : Version **\*\*X11\*\***.
- **\*\*1990s\*\*** : Début de **\*\*Windows NT\*\***, lancé en **\*\*1993\*\***.
- **\*\*1977\*\*** : Fondation d'**\*\*Oracle\*\***.
- **\*\*1999\*\*** : Première version de **\*\*VMware\*\***.

Oui, **Windows Terminal Server** a effectivement commencé à être déployé par **Microsoft** en se basant sur **Windows NT**.

### Détails :

- **Terminal Server** est une extension de **Windows NT** qui permet de faire fonctionner plusieurs sessions distantes sur un même serveur. Cette technologie permet aux utilisateurs de se connecter à un serveur Windows à distance et d'utiliser un bureau ou des applications comme s'ils étaient physiquement présents sur la machine.

- La première version de **Terminal Server** est apparue avec **Windows NT 4.0 Terminal Server Edition**, lancée en **1996**. C'était une solution idéale pour remplacer les architectures de client lourd en proposant un **client léger**, où le serveur gère la plupart des traitements, et les clients se connectent simplement pour afficher et interagir avec les applications.
- **Windows NT** servait donc de base à cette technologie, qui utilisait le **Remote Desktop Protocol (RDP)** pour gérer l'affichage à distance.

#### **RDP (Remote Desktop Protocol) :**

- RDP est un protocole de communication développé par Microsoft, permettant à un utilisateur de se connecter à un ordinateur distant et d'utiliser son interface graphique. Il transmet non seulement les entrées de l'utilisateur (clavier, souris) vers le serveur, mais aussi l'affichage graphique de l'interface du serveur vers le client.

### **### 1. \*\*Qu'est-ce que RPC ?\*\***

- **Remote Procedure Call (RPC)** est un protocole qui permet à un programme (le client) d'exécuter une procédure (une fonction) sur un autre programme (le serveur) qui peut être situé sur une machine différente dans un réseau.
- L'idée principale est de **déporter** les processus coûteux d'un client local vers un serveur plus puissant, optimisant ainsi les performances des applications.
- L'idée remonte à 1976 décrit dans le RFC 707, et utilisé la première fois en 1981 par XEROX sous le nom de courrier, la plus populaire est par SUN.

### **### 2. \*\*Fonctionnement du RPC**

- **Prototypes et Stubs** : Le client doit définir des prototypes de fonctions qui seront exécutées sur le serveur. Cela se fait généralement dans un fichier d'interface.
- Le **stub** côté client est un code généré qui contient les appels de fonctions que l'utilisateur souhaite exécuter sur le serveur.
- Le **skeleton** côté serveur est le code qui reçoit l'appel du stub, exécute la fonction appropriée et renvoie les résultats au client.
- **Fichiers .rpc** : Les prototypes des fonctions sont généralement définis dans un fichier d'interface avec l'extension `.rpc`` (ou `.x`` dans certaines implémentations). Ce fichier est utilisé pour générer le code stub et skeleton.

### **### 3. \*\*Communication**

- **Connexion TCP** : Le RPC utilise des connexions **TCP** pour établir la communication entre le client et le serveur. Les données sont échangées via des **sockets**.

- **Port** : Un port spécifique est utilisé pour écouter les connexions entrantes sur le serveur, permettant au client d'envoyer des requêtes à la bonne destination.

#### **### 4. Fonctionnement des Appels**

- Lorsque le client appelle une fonction via le stub, le stub :
  1. Prépare les arguments pour l'appel distant.
  2. Envoie une requête via le réseau au serveur.
  3. Le serveur reçoit la requête, exécute la fonction correspondante, puis renvoie la réponse au stub.
  4. Le stub traite la réponse et la renvoie à l'utilisateur.

#### **### 5. API**

- Le RPC peut être considéré comme une **API** (Application Programming Interface), car il définit comment les fonctions peuvent être appelées à distance, et comment les données sont échangées entre le client et le serveur.

#### **### Conclusion**

- **RPC** est un moyen efficace de déporter les traitements lourds vers un serveur, améliorant ainsi les performances des applications client-serveur.
- Les concepts de **stub** et **skeleton** sont cruciaux pour le fonctionnement du RPC, facilitant la communication entre le client et le serveur.

#### **### Points à retenir :**

- **RPC** permet d'exécuter des fonctions à distance.
- Utilisation de fichiers `.rpc` pour définir les prototypes.
- Communication via TCP et sockets.
- Concepts de stub (côté client) et skeleton (côté serveur).
- Les sockets durent indéfiniment, de plus lorsqu'il y a plusieurs tentative de connexion alors il y a un buffer qui va stocker les requêtes de connexion généralement elles ne dépassent pas plus de 1min30 pour éviter la saturation sur les serveurs et les attaques DDOS.

#### **Distribution des Ressources**

- En utilisant RPC, les applications peuvent :
  - **Déporter** les processus coûteux sur un serveur puissant, libérant ainsi des ressources sur le client.

- **Accéder** à des services ou à des données qui sont hébergés sur d'autres machines dans le réseau, permettant ainsi une architecture distribuée.